"Análisis del comportamiento del consumidor mediante la base datos *Marketing Campaign*"

por

LUCÍA SÁNCHEZ SÁNCHEZ

Tesis presentada en conformidad con los requisitos del
Máster en Economía, Finanzas y Computación.

Universidad de Huelva & Universidad Internacional de Andalucía

HUELVA, 2023

"Análisis del comportamiento del consumidor mediante la base datos *Marketing Campaign*"

LUCÍA SÁNCHEZ SÁNCHEZ
Máster en Economía, Finanzas y Computación

Supervisado por:
JUAN JOSÉ ALBENDÍN MOYA
Universidad de Huelva

Abstract

This Final Degree Project, presented below, aims to study consumer behaviour using the Marketing Campaign database. The study will focus mainly on analysing it using machine learning techniques. To achieve this, an exploration of the database has been carried out with its corresponding data cleaning to be able to analyse it later. For the analysis, a decision tree classification algorithm will be implemented to predict whether or not the consumer will accept the marketing campaign. In addition, a k-means clustering algorithm will be applied to group consumers with similar characteristics into different homogeneous segments. Combining both techniques will help understand consumers and enable the development of specific and personalized strategies

JEL Classification C45, M31

Keywords Machine learning, decision tree, clustering, statistical analysis, marketing campaigns

Resumen

El Trabajo de Fin de Grado que se expone a continuación tiene como objetivo el estudio del comportamiento del consumidor utilizando la base de datos *Marketing Campaign*.El estudio se centrará principalmente en analizarlo mediante técnicas de aprendizaje automático. Para lograrlo, se llevará a cabo una exploración de la base de datos seguida de una limpieza de datos para luego poder analizarla. Para el análisis, se implementará un algoritmo de clasificación de árbol de decisión para predecir si el consumidor aceptará o no la campaña de marketing. Además, se realizará un algoritmo de *clustering k-means* para agrupar los consumidores que poseen características similares en diversos segmentos homogéneos. La combinación de ambas técnicas ayudará a comprender a los consumidores y permitirá desarrollar estrategias específicas y personalizadas.

Palabras claves Aprendizaje automático, árbol de decisión, clustering, análisis estadístico, campañas de marketing

Agradecimientos

Este trabajo no podría haberse llevado a cabo sin el apoyo de varias personas y, por ello, quiero mencionarlas:

En primer lugar, me gustaría agradecer a mi director, el Prof. Dr. Juan José Albendín Moya, por su labor académica y el ayudarme por segunde vez en la realización de un trabajo de investigación enfocado en el área de marketing.

En segundo lugar, agradezco a mis amigos por acompañarme durante todo el proceso. Su apoyo constante, su escucha y motivación han sido un factor clave para superar cada desafío que me he ido encontrando a lo largo del proyecto.

Por último, y no menos importante, me gustaría agradecer a mi familia. Han estado a mi lado a lo largo de todo el máster brindándome un apoyo incansable. Su influencia positiva y, sobre todo, sacrificio, han sido los pilares para mi éxito académico. Les estaré agradecida eternamente por su presencia en esta etapa de mi vida.

...

Índice General

Índice de Tablas

Índice de Figuras

1. INTRODUCCIÓN

La realización de este Trabajo de Fin de Máster tiene como fin poner de manifiesto diversos conocimientos adquiridos durante el máster sobre marketing y aprendizaje automático. Esto se ha conseguido gracias al uso del conjunto de datos *marketing campaign* [Parr-Rud, 2014]. En el área del marketing es imprescindible comprender y entender el comportamiento de los consumidores así como las necesidades que poseen para adaptar las estrategias de las empresas hacia ellos.

Por ello, es necesario conocer los procesos de compras, las distintas herramientas de comunicación que se pueden llevar a cabo para atraer la compra y los diversos factores que influyen en el consumidor para tomar decisiones. Se ha utilizado una base de datos que ayudará a analizar los patrones de los consumidores y poder realizar estrategias entorno a ellos. El conjunto de datos seleccionado cuenta con una amplia información de diversos aspectos importantes para llevar a cabo una campaña de marketing por parte de una empresa. Desde el tipo de compra, los ingresos de los consumidores hasta el estilo de vida que llevan.

Se tendrá en mente en todo momento los objetivos marcados a la hora de analizar los datos. Se buscará solución a cuestiones como el tipo de perfil que tienen los consumidores que aceptan las campañas de marketing de la empresa. También se comprobará si los modelos realizados serán capaces de clasificar a los consumidores entre los que realmente aceptan las campañas de marketing como los que no.

A fin de conseguir esto, primero se llevará a cabo una revisión bibliográfica de conceptos teóricos relacionados con el marketing y el aprendizaje automático. Además de agregando a lo anterior una utilización de una metodología por la cual se realizará un análisis del conjunto de datos examinando sus variables. En caso de encontrar algunos patrones extraños o valores nulos, se llevará a cabo una limpieza y tratamiento del conjunto de datos para poder utilizarlo de manera adecuada.

Tras esto, se realizará un estudio estadístico a raíz de los datos. Con ello, se puede observar aquellas características de los potenciales consumidores que más se repiten y empezar a sacar

las primeras conclusiones. Se aplicarán dos técnicas de aprendizaje automático seleccionadas, una paramétrica usando árboles de decisiones y otra no paramétrica usando *clustering*. Todo ello se hará para conocer si los modelos predicen bien y poder encontrar patrones para agrupar a los tipos de clientes.

Para finalizar el análisis empírico, se realizará una evaluación utilizando diversas métricas como la precisión, la exactitud o la distancia entre los datos. La utilización de estas métricas ayudarán a conocer cómo de bien se han ejecutado los distintos algoritmos empleados y así poder analizar los resultados para poder dar respuesta a los objetivos planteados.

En conclusión, en este trabajo de fin de máster se analizará la base de datos *marketing campaign* para poder conocer los consumidores y su actitud frente a las acciones de marketing emprendidas por la empresa. Así como predecir las respuestas que tendrá un consumidor ante los estímulos realizados por las campañas. Todo ello servirá para poder desarrollar campañas de marketing más eficaces enfocadas de manera correcta ante los diversos tipos de consumidores existentes.

2. OBJETIVOS

Para la realización de este trabajo se han planteado una serie de objetivos a conseguir. Se busca sacarle el mayor partido a la base de datos para obtener información de gran valor sobre diversos aspectos del marketing y las campañas realizadas por la empresa.

El primer objetivo marcado para este estudio es el análisis de las variables que influyen en el consumidor para aceptar o no la campaña de marketing. Para ello, evaluar variables como la edad del consumidor, su estado civil, los ingresos familiares o el nivel de estudio que poseen serán claves para determinar los patrones. Al tener conocimiento sobre lo que afecta a que una campaña tenga mayor o menor éxito, se podrá ajustar la estrategia seguida por la empresa de una manera más eficaz y dirigir el foco de atención a los segmentos correspondientes.

El segundo objetivo propuesto es la segmentación de los consumidores mediante la búsqueda de patrones que los determinen. Para realizar este proceso, se utilizarán técnicas de aprendizaje automático de agrupación para buscar características parecidas entre los consumidores y agruparlos juntos. La realización de este proceso ayudará a poder implementar campañas de marketing enfocadas en determinados segmentos y así lograr optimizarlas. De este modo, se podrá aumentar la posibilidad de que la campaña realizada tenga efecto en el grupo de consumidores elegido.

El tercer objetivo planteado será la optimización de los canales de distribución. A partir de los datos del conjunto analizados, se pretende identificar los canales de distribución más efectivos para que la empresa pueda poner el producto al alcance de los consumidores. Se examinarán tantos los canales físicos como los virtuales para conocer aquellos que más utilizan. Así, se podrá llevar a cabo una estrategia para optimizar el acercamiento de los productos a los clientes y adaptar las tácticas empleadas hasta ahora.

El cuarto, y último, objetivo es el desarrollo de modelos de aprendizaje automático. Estos modelos servirán para entrenar los datos actuales y validar su eficacia. Así, cuando en un futuro se posean nuevos datos, poder predecir si los clientes se verán afectados positivamente por las campañas

realizadas de marketing. También, servirán para poder identificar a qué tipo de grupo de consumidores pertenecen y utilizar una estrategia determina u otra.

2.1. Hipótesis

En este trabajo de investigación se han planteado diversos objetivos que van a marcar el rumbo y la línea de investigación. Para poder comprender el conjunto de datos, analizar lo que hace que el consumidor acepte la campaña o no, y lograr los objetivos marcados, se han propuesto diversas hipótesis. Estas ayudarán a dar respuesta a las preguntas plateadas a lo largo del trabajo.

La primera hipótesis que se ha planteado es conocer el éxito de la campaña de marketing realizada por parte de la empresa. Si hay más clientes que aceptan la oferta propuesta por la última campaña realizada o no. El planteamiento de esta hipótesis es fundamental ya que el conocer de forma analítica si la campaña de marketing ha tenido éxito o no, supondrá que en un futuro se siga la misma línea o se haga un cambio en las estrategias.

H1= Hay más consumidores que no aceptaron la oferta en la última campaña.

La segunda hipótesis que se plantea para este trabajo es comprobar si los ingresos anuales en los hogares de los consumidores afectan en menor o mayor medida a la hora de verse influenciados por las campañas de marketing. De este modo, se busca comprobar si los consumidores con distintos rangos salariales muestran patrones distintos a la hora de aceptar las campañas o no.

H2= Los sueldos influyen en la aceptación de la campaña de marketing.

En esta tercera hipótesis se ha señalado la búsqueda de correlación entre el gasto general realizado por los consumidores de la marca y el ingreso anual que tienen por familia para contrastar la relación entre ambas variables y ver si existen distintos patrones según el rango salarial.

H3= Existe correlación entre los diversos productos adquiridos por los consumidores.

La cuarta hipótesis que se analizará será la relación entre los canales de distribución y el gasto general que realizan los consumidores en los distintos productos de a marca. Se comprobará la correlación entre ambas variables para saber su relación. De este modo, se podrán analizar las estrategias de marketing y enfocarlas de manera adecuada a los respectivos canales.

H4= Existe correlación entre los diversos productos adquiridos por los consumidores y la forma de adquisición.

3. METODOLOGÍA

La realización de este trabajo de investigación se ha basado en el estudio de las hipótesis planteadas seleccionando los métodos de investigación más adecuados para dar respuesta a éstas.

En la primera parte de esta investigación, se ha llevado a cabo una revisión bibliográfica, apoyada en evidencias científicas para dar solución a las hipótesis planteadas. Para ello se realizó una búsqueda bibliográfica en diversas bases de datos científicos, fuentes de carácter oficial y libros, algunos de ellos genéricos para poder asentar algunos de los conocimientos necesarios para entender más adelante el análisis que se quiere llevar a cabo. De esta forma, se ha obtenido información suficiente para poder dar explicación al comportamiento del consumidor y el proceso que lleva a cabo para realizar una compra. Por otro lado, ha servido para definir las distintas estrategias de comunicación comercial para poder entender los datos que proporciona el conjunto seleccionado para este estudio.

En la segunda parte, se ha efectuado un estudio empírico a través del análisis de la base de datos *Marketing Campaign* [Parr-Rud, 2014]. Para ello, se ha realizado también una búsqueda en distintas plataformas de bases de datos hasta dar con la que se va a analizar en este trabajo, siempre buscando una relacionada con el ámbito del marketing. Una vez seleccionada, se ha buscado información sobre aprendizaje automático para poder dar solución a los objetivos marcados para esta investigación. Este conocimiento ha servido para conocer los distintos tipos de métodos existentes para así posteriormente escoger aquellos que mejor se adaptasen a los propósitos marcados. Así mismo, se ha consultado diversas fuentes de información para poder entender correctamente las diversas métricas y métodos necesarios para evaluar las técnicas de aprendizaje automático realizados.

En este estudio se ha decidido realizar el estudio empírico del conjunto de datos mediante el lenguaje de programación python para poder ejecutar diversos códigos y así analizar el conjunto de datos seleccionado. Además de utilizar este lenguaje, se han utilizado diversas librerías que han facilitado el estudio de los datos para poder conseguir las respuestas a los objetivos marcados. Por

un lado, se ha utilizado la librería pandas para poder hacer un tratamiento de los datos así como el manejo de ellos. Además, ha facilitado la creación de gráficas para poder evaluar y analizar de forma visual los resultados obtenidos [McKinney, 2012]. Por otro lado, se ha utilizado la librería sklearn para poder usar los diferentes algoritmos de aprendizaje automático. En el caso de este trabajo de investigación, se han utilizado los algoritmos específicos para la creación de programas de clasificación y agrupación. También ha aportado diversos códigos necesarios para evaluar o tratar de la mejor forma dichos algoritmos [Pedregosa et al., 2011].

3.1. MÉTODOS UTILIZADOS

Para poder dar respuesta a los objetivos planteados en el trabajo de investigación, se tienen que utilizar diversas técnicas. Estas se utilizan para construir un modelo que permita clasificar y agrupar las clases de las instancias analizadas en función de una serie de atributos de entrada. Por ello, se han seleccionado los siguientes métodos para aplicar al conjunto de datos en base de los objetivos marcados, los tipos de datos y sus propias características.

3.1.1. Árbol de decisión

El primer método que se ha utilizado es para construir un modelo de clasificación. Hay diversos métodos que se pueden aplicar como la regresión logística, las máquinas de vector soporte (SVM) o los árboles de decisión. En este trabajo de investigación se ha optado por realizar la técnica de árboles de decisión para construir un modelo de clasificación.

Un árbol de decisión tiene una estructura parecida a los árboles que se utilizan para realizar predicciones en base a diversas características. En este caso, es un modelo de aprendizaje automático que se utiliza para tomar decisiones y realizar predicciones en base a los datos del conjunto estudiado. El algoritmo consiste en ir dividiendo el conjunto de datos en funciones y escoger aquellas divisiones que separen de manera eficiente las instancias de cada clase [Quinlan, 1986].

Los árboles de decisiones están formados por distintas partes. La primera parte que lo compone es

un nodo raíz que es el nodo principal del cual parte el árbol y representa el atributo que iniciará la primera división. A partir de este, se crearán más nodos conocidos como nodos internos. Estos representan las características extras que se utilizarán para crear más divisiones en el árbol. Además de los nodos, existen otras componentes que forman los árboles como son las hojas y las ramas. Las hojas representan los nodos de salida, es decir, la decisión final sobre la clase a la que pertenece una instancia. Por otro lado, las ramas hacen referencia a las conexiones que existen entre los nodos y las posibles opciones que existen a la hora de elegir una opción [Breiman et al., 2017].

Existen diversos algoritmos para construir los árboles de decisión como el *iterative dichotomiser 3* o el CART, *classification and regression trees*. El primero consiste en ir seleccionando las características discriminatorias e ir construyendo el árbol de manera recursiva. En el segundo, se van construyendo arboles binarios donde se ramifica en función de las características. Luego existen otros tipos de árboles como los ensambles de estos mismo donde se encuentran los *random forest* y *gradient boosting* [Bouza and Santiago, 2014].

3.1.2. Validación cruzada

La validación cruzada, también conocida en inglés como *cross-validation*, es una técnica que se usa en el aprendizaje automático para crear modelos de una forma más precisa. Este tipo de técnica consiste en dividir los datos en *folds*, es decir, en diversas particiones y realiza varios entrenamientos combinando las distintas particiones obtenidas [Refaeilzadeh et al., 2009].

El proceso de esta técnica consiste en las siguientes fases. Primero se realiza la división de los datos estudiados en diversas particiones. Normalmente, estas particiones van desde 5 hasta 10 aunque dependiendo de los datos puede variar. Una vez se ha obtenido las particiones se realiza una iteración donde se selecciona una partición como conjunto de prueba. Tras esto se entrena el modelo y se evalúa comparando las predicciones del modelo con los datos reales. Por último, se recogen los datos obtenidos y se hace una media de los rendimientos obtenidos por cada partición [Zhang et al., 2010].

3.1.3. SMOTE

Cuando existe un desajuste de clases, es decir, una proporción desigual entre las instancias, pueden realizarse diversas técnicas para igualarlas. Una de las técnicas que se utiliza para tratar este desbalance es la técnica SMOTE (*Synthetic Minority Over-sampling Technique)*. Esta técnica de sobremuestreo realizada mediante interpolación consiste en generar nuevas instancias ficticias de la clase que tiene un menor porcentaje de datos para así lograr un equilibrio [Chawla et al., 2002].

El funcionamiento de esta técnica consiste en elegir una instancia que pertenezca a la clase menos representada. Tras esto, se buscan los k vecinos que se encuentren más cerca a la instancia elegida utilizando alguna técnica como la distancia euclídea. Una vez seleccionada aleatoriamente un k vecino, se crea una instancia ficticia nueva siendo un punto medio entre la instancia y el k vecino. Esto se realiza hasta conseguir el número deseado de instancias. [Bekkar and Alitouche, 2013].

La utilización de esta técnica ayuda a evitar un sesgo a la hora de ejecutar el algoritmo elegido. Hay que saber que la utilización de la técnica SMOTE se debe utilizar solamente con el conjunto de entrenamiento pues sino estaríamos modificando también el conjunto de prueba y no saldrían los resultados esperados a la hora de realizar la evaluación [Maldonado et al., 2022].

3.1.4. *Clustering*

El segundo método utilizado para analizar los datos del conjunto ha sido el *clustering*. Este tipo de método de agrupación consiste en reunir en grupos aquellos elementos que tengan en común alguna característica. Se busca que los datos agrupados sean homogéneos entre sí y que tengan más parecido entre los distintos elementos del grupo que con los elementos de otros grupos ajenos. La utilización de este tipo de técnica es muy común para aquellos trabajos donde se busca conocer patrones y estructuras propias de los conjuntos de datos [Estarelles et al., 1992].

Como se comentará más adelante en las métricas, este tipo de algoritmo busca tener un resultado máximo a la hora de analizar la situación intra-*cluster* de los datos y una mínima con respecto al

inter-*cluster*. Los datos de un mismo agrupamiento deben encontrarse cercanos entre sí mientras que deben encontrarse alejados de los datos de otros *clusters* [Ezugwu et al., 2022].

Al igual que en los árboles de decisiones, existen distintos algoritmos para realizar un estudio de *cluster*. Existen muchos tipos de *cluster* ya que según el tipo de dato analizado estos podrán tener una forma o tamaño distinto. Asimismo, los *cluster* realizados pueden ser lineales o no. Algunos de los algoritmos que más se usan son el *k-means* o el DBSCAN(*density-based spatial clustering of applications with noise*) [Prada Conde, 2022].

3.1.5. Técnica *one hot*

La técnica de one hot consiste en modificar las variables categóricas del conjunto de datos para convertirlas en vectores binarios. Esto se realiza para poder trabajar con diversos algoritmos de *machine learning*, ya que necesitan tener los datos de forma numérica para poder trabajar con ellos. De este modo, cada variable categórica se vuelve una variable binaria. Los valores que se asignan son 1 y 0, correspondiendo el valor 1 con la posición que corresponde a la variable categórica y en caso contrario, el 0 [Ul Haq et al., 2019].

3.1.6. Método codo

Como se ha comentado, el método de agrupación por *cluster* consiste en la creación de grupos con patrones parecidos. La selección de este número óptimo de agrupaciones se puede realizan mediante diversos métodos, uno de ellos es el método del codo. El método del codo que consiste en un análisis del conjunto de datos para decidir cuál es el número óptimo de divisiones que debe de tener el conjunto estudiado. Este método logra visualizar un punto de inflexión de la suma de las distancias cuadradas intra-*cluster* [Umargono et al., 2020].

Esta suma consiste en calcular las distancias cuadradas de las instancias al centroide más cercano de cada uno. Se busca que este resultado sea lo menor posible ya que supondría que los *cluster* son más

homogéneos y a la vez compactos. Mediante la representación gráfica que nos arroja este método, se puede comprobar un punto donde la gráfica empieza a ser más estable. Este punto de inflexión o mínimo, indica el número de *cluster* que le debemos dar al algoritmo. Hay que tener en cuenta que a veces ese cambio puede no verse claramente y deberá realizarse un estudio comparando diversas cantidades de *cluster* hasta dar con aquel que se adapte mejor a los datos [Nanjundan et al., 2019].

3.2. MÉTRICAS UTILIZADAS

A la hora de realizar el estudio del conjunto de datos, existen diversas métricas que sirven para medir el rendimiento de estos datos. Este tipo de métrica puede medir la bondad de ajuste del conjunto de datos que se obtiene a raíz de los aciertos y fallos que comete el clasificador.

3.2.1. Matriz de confusión

La primera, y sobre la que se sustentan las demás métricas, es la matriz de confusión. Esta matriz consiste en una tabla de contingencia que se utiliza como herramienta estadística para el análisis de valores que tienen un grado de semejanza entre las diversas observaciones emparejadas [Ariza-López et al., 2018]. Para la organización de los datos obtenidos por clase en un problema de clasificación binario tendría la siguiente estructura:

		Predicción	
		Positivos	*Negativos*
Observación	*Positivos*	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Cuadro 1: Representación de la matriz de confusión [Düntsch and Gediga, 2019]

La tabla relaciona la clasificación dada por el modelo. En los ejes verticales se encuentran los valores que corresponden con las predicciones. Mientras que en los ejes horizontales se encuentran

los valores relacionados con las observaciones. Por tanto, los valores en la diagonal principal corresponden a los aciertos, y el resto de valores a los errores.

3.2.2. *Accuracy*

Una vez realizada la matriz de confusión, se podrán realizar las demás métricas para evaluar nuestro modelo. El *Accuracy (%)* también se le conoce como instancias bien clasificadas a nivel global, tasa de aciertos, exactitud, etc. Esta métrica representa el porcentaje total de valores correctamente clasificados, tanto positivos como negativos. Se calcula con la siguiente fórmula a raíz de los resultados de la matriz de confusión. No es recomendable utilizarla cuando las clases analizadas están desbalanceadas. En caso de que el conjunto de datos se encontrase desbalanceado, otro tipo de métrica aportaría mayor información respecto al modelo [Hossin and Sulaiman, 2015].

$$\frac{(VP + VN)}{(VP + VN + FP + FN)} \tag{1}$$

3.2.3. Tasa de error

La tasa de errores (%) se conoce como la fórmula complementaria al *accuracy*. Esta fórmula ayuda a conocer la cantidad de predicciones que se ha clasificado de manera incorrecta con respecto al total de predicciones. Por ello, se busca minimizar lo máximo posible el resultado [Hossin and Sulaiman, 2015]. Esta fórmula se calcula de la siguiente forma:

$$\frac{(FP + FN)}{(VP + VN + FP + FN)} \tag{2}$$

3.2.4. Índice kappa de Cohen

El índice kappa de Cohen hace referencia al índice de bondad de ajuste del modelo completo. Este se utiliza para evaluar la concordancia de los instrumentos de medida cuyo resultado es categórico, siempre que tenga 2 o más categorías. Mide la consistencia en las respuestas y el valor que arroja va entre el 0 y el 1. Si el resultado obtenido es bajo, el clasificador no se considera apropiado para esos datos ya que el conjunto de datos tiene dificultades al presentar discordancia. Si el resultado obtenido fuese de 0, significa que esta concordancia es la misma que se esperaría al azar. Un índice por encima de 0.5 se consideraría bueno y cuanto más cercano a 1 se encuentre el

resultado, más concordancia perfecta se dará entre las variables [Brennan and Prediger, 1981].

3.2.5. Precisión (P)

La precisión (P) se utiliza para saber qué porcentaje de valores que se han clasificado como positivos son realmente positivos. Ese tipo de fórmula da la calidad de la predicción, pero hay que tener en cuenta que se debe complementar con otras fórmulas sobre todo si el conjunto de datos está desbalanceado [Rodríguez et al., 2016]. Se calcula de la siguiente forma:

$$\frac{VP}{(VP + FP)} \tag{3}$$

3.2.6. *Recall (R)*

La métrica *Recall (R)* se le conoce como el ratio de verdaderos positivos o exhaustividad. Se utiliza para conocer la cantidad de valores positivos han sido correctamente clasificados. Esta fórmula se considera importante cuando el objetivo del conjunto de datos es minimizar los falsos positivos [Borja-Robalino et al., 2020]. Se calcula de la siguiente forma:

$$\frac{TP}{(TP + FN)} \tag{4}$$

3.2.7. *F-Measure*

La métrica de *F-Measure* combina la fórmula precisión y el *recall*, para obtener un valor mucho más objetivo. El uso de esta fórmula permite comparar el rendimiento combinando las dos fórmulas ya mencionadas. Asimismo, asume que ambas importan de igual manera aunque no tiene por qué suceder por igual en todos los problemas. Se utiliza mucho en aquellos problemas donde los conjuntos de datos analizados están desbalanceados y donde se busca minimizar los falsos positivos y falsos negativos. Los resultados de la fórmula van desde 0 hasta 1, siendo el 1 un equilibrio perfecto entre la precisión y la exhaustividad. [Bekkar et al., 2013]. Se calcula como se expresa a continuación:

$$F1 = 2 * \frac{(recall * precision)}{(recall + precision)} \tag{5}$$

3.2.8. Curva ROC

El área bajo la curva ROC se representa mediante la curva ROC se representa en un eje de abscisas y ordenadas. Esta expone la relación entre la sensibilidad, que hace referencia a los verdaderos positivos entre el total de positivos, y especificidad, que son los verdaderos negativos entre el total de negativos, del modelo de clasificación realizado [Hoo et al., 2017].

La representación consiste en una curva para cada categoría tengan relación con el resto de categorías. Esto indica la capacidad que tiene el modelo para detectar los casos que pertenecen a la categoría de la variable criterio. Esta métrica indica, por lo tanto, la precisión que tiene el modelo para identificar correctamente a los sujetos de un grupo. Tiene en cuenta teniendo el porcentaje de acierto en la detección de la categoría así como el porcentaje de desaciertos al identificar los sujetos de esa categoría [Park et al., 2004].

Se puede utilizar para comparar varios algoritmos de aprendizaje automático. Por debajo de 0.5, significa que el modelo estudiado tiene un rendimiento peor que al realizar una clasificación aleatoria. Un modelo con un resultado por encima del 0.7 se puede considerar bueno ya que cuanto más cercano a 1, mayor capacidad tendrá el modelo de distinguir entre las clases [Bekkar and Alitouche, 2013].

3.2.9. *Silhouette*

Las métricas que se utilizan en *machine learning* para agrupar son distintas que las de clasificación, las comentadas hasta ahora. Existen diversas métricas que sirven para poder evaluar la agrupación realizada. Según el tipo de problema que se vaya a analizar y el tipo de algoritmo realizado, unas serán más recomendadas que otras. No existe una métrica que sea mejor que otra, es recomendable hacer uso de diversas métricas para poder realizar un análisis completo del *cluster* realizado [Estarelles et al., 1992].

La métrica utilizada para *cluster* es la métrica de la silueta o *silhouette*. Esta métrica se utiliza para evaluar la agrupación de datos obtenido mediante un algoritmo de *clustering*. Con ella, se

puede comprobar cómo de bien se separan las distintas clases y cómo de similares son los datos dentro de éstas [Thinsungnoen et al., 2015].

La fórmula está formado por dos parámetros. El primer parámetro hace referencia a la coherencia intra-*cluster*(a), que mide el promedio entre la distancia de un objeto y los demás dentro de un mismo agrupamiento. Se considerará que tiene buena coherencia si la distancia entre los objetos dentro del *cluster* es mínima. El segundo parámetro necesario para la fórmula es la coherencia inter-*cluster*(b). Este parámetro mide la media entre la distancia de un objeto y los objetos de otros agrupamientos cercanos. En este caso, cuanto mayor sea el resultado obtenido, mejor será la separación que hay entre los distintos *clusters* [Gaido, 2023].

$$s = \frac{b - a}{max(a, b)} \tag{6}$$

Los resultados obtenidos tienen valores comprendidos entre -1 hasta 1. Cuanto más cercano a 1 sea el resultado, mejor asignado estarán los datos en sus *clusters* correspondiente y mejor separados estarán de los demás. En cambio, si el resultado obtenido es cercano al -1, significará que los datos se han asignado mal al *cluster* y que están cerca de los demás [Mulaomerović-Šeta et al., 2023]

4. MARCO TEÓRICO

4.1. Comportamiento de los consumidores

Las estrategias de marketing están enfocadas a las 4p que hacen referencia al producto, precio, distribución y promoción. Además de esto, las campañas o estrategias de marketing se pueden realizar a través de dos modelos. Por un lado, se pueden realizar estrategias enfocadas a un producto en específico de la empresa, y por otro lado, se pueden realizar otras que estén enfocadas en la empresa en su conjunto total. A la hora de analizar el comportamiento del consumidor, se debe tener en cuenta que tipo de estrategia se está realizando [Berger et al., 2010].

Hay que tener en cuenta que cada consumidor es distinto. Cada uno tiene una cultura, una rutina, preferencias o estilo de vida que difiere del resto. Esto implica que una empresa nunca vaya a encontrar dos personas iguales en el mundo. A la hora de plantear y llevar a cabo una estrategia se tendrá que tener en cuenta que los consumidores reaccionarán de forma diferente ante estas [Vergara, 2022].

Así mismo, se debe conocer que si una persona se implica con una marca, será más fácil que ésta acabe aceptando las campañas y estrategias realizadas por la empresa. En el lado contrario, si la persona es un consumidor que no se implica con la marca o que es un cliente esporádico, no se verá tan influenciado por las campañas realizadas [De Matos and Veiga, 2004].

4.2. Proceso de compra

El consumidor, a la hora de realizar su proceso de compra, decide sobre la adquisición y las diversas circunstancias que la rodea. La conducta del cliente a la hora de realizar este proceso es algo complejo e incierto [Nebreda, 1992]. Hay que saber, que el proceso de compra de un consumidor pasa por diversas fases que consisten en darse cuenta de la necesidad, buscar información sobre esta necesidad, evaluación de las diversas alternativas, toma de decisión de compra y evaluación posterior de ésta [Humbría, 2010]. El comportamiento de los consumidores viene marcado por los diversos factores culturales, sociales y personales que rodean a la persona. Esto quiere decir que

un proceso de compra nunca será igual entre distintos consumidores [Kotler and Keller, 2006].

La primera fase de este proceso consiste en que el consumidor se dé cuenta de que tiene una necesidad insatisfecha. La necesidad puede ir desde de adquirir un servicio hasta adquirir un producto. El descubrimiento de esta necesidad carente puede ser promovida por factores internos del consumidor al no estar satisfecha alguna necesidad fisiológica o por factores externos mediante los estímulos de la publicidad o recomendaciones de personas cercanas [Blythe, 2004].

Una vez que el consumidor ha descubierto que tiene una necesidad que no ha cubierto, procede a recabar información acerca de posibles servicios o productos para dar solución a su problema. Actualmente existen diversos modos de informarse acerca de estos como ir presencialmente a la tienda a ver los productos que ofrecen, preguntar a algún familiar cercano sobre su opinión al respecto o buscar en internet comentarios de otros consumidores [De Benito, 2014].

Tras obtener el consumidor el conocimiento necesario del producto o servicio que va a adquirir, procede a evaluar las distintas opciones que tiene a su alcance. Desde el punto de vista del marketing, se debe tener en cuenta que el consumidor se basará en factores como la calidad-precio o su relación con la marca a la hora de decantarse por un producto u otro [com, 2021].

Finalmente, el consumidor toma la decisión de adquirir el bien o servicio por el que se ha decantado al final. En esta fase, se realiza un intercambio entre el consumidor y la empresa. Esta compra se puede realizar de forma presencial a través de la tienda propia de la marca o incluso en línea a través de sus portales web de compra [Kotler and Armstrong, 2012].

La última fase de compra del consumidor corresponde a la evaluación que se realiza tras haber adquirido el producto o servicio y haberlo probado. Si este satisface la necesidad por la cual surgió la compra, el consumidor puede plantearse en el futuro volver a adquirir el producto en concreto u otro relacionado con la marca. En caso contrario, podría optar por no adquirir nada relacionado [Ramón and Morán, 2014].

4.3. Estrategias según el canal de distribución

Cuando se selecciona un canal para llevar a cabo una estrategia de marketing se debe contemplar diversos factores. Cada canal tiene sus ventajas y desventajas, así como unas características que lo hace único. Por ello, se debe estudiar los medios disponibles para llegar al alcance de los consumidores y así poder promocionar los productos de la marca [Acosta, 2017].

Hay dos tipos de canales de distribución a grandes rasgos, los que son en físico y los que son en línea. Los primeros que se van a analizar son los tradicionales, los canales de distribución física. Estos canales se caracterizan por tener puntos de ventas físicos y ser más cercanos con el cliente. Una de las desventajas que poseen los puntos físicos es que se encuentran en una ubicación determinada y no llega a todos los posibles clientes. El tipo de estrategia de comunicación está enfocada en aspectos físicos como es la publicidad en los locales o *merchandising* en el punto de venta. Los locales físicos siempre buscarán crear una experiencia para los clientes a través del diseño de la tienda o del servicio que se le ofrece al consumidor. Por último, la medición de los resultados de las campañas de marketing en lugares físicos suelen ser más complicados de medir [Kotler and Armstrong, 2012].

Por otro lado, se encuentran los lugares en línea como son las páginas web de las empresas o las redes sociales. Gracias a internet, las empresas pueden tener un alcance geográfico más amplio ya que pueden acceder desde cualquier punto del planeta. Las estrategias de comunicación distan de las de los locales físicos pues al ser en línea, se enfocan en crear campañas en redes sociales u optimizar los motores de búsqueda para posicionar entre los primeros puestos la página de la empresa. Al igual que en una tienda, una web busca retener el cliente y que acabe comprando el producto. Por eso, el diseño de la web debe ser atractiva, con información clara y que la navegación sea accesible a cualquier persona. Para finalizar, la medición de los resultados de las estrategias de marketing implementadas puede evaluarse de una forma más sencilla pues se disponen de métricas como la tasa de conversión o el tráfico de la página para analizarlo [Lozano-Torres et al., 2021].

5. ESTUDIO EMPÍRICO

El conjunto de datos que se ha seleccionado para la realización de este Trabajo de Fin de Master es una base de datos llamada *Marketing Campaign* [Parr-Rud, 2014]. Esta consiste en una recopilación de datos sobre las intenciones de compra de distintos clientes. El objetivo que tiene este conjunto de datos es poder aplicarle diversas técnicas de clasificación para conocer qué tipo de cliente responde ante las diversas campañas de marketing que han sido creadas para distintos productos o servicios.

5.1. Estudio de las variables

Este conjunto seleccionado para el estudio está formado por 2240 observaciones. Cada una de las observaciones viene descrita por 29 atributos y una variable de salida o respuesta. El conjunto de datos posee valores nulos, por lo que para poder utilizar la base de datos hay que realizarle un proceso de detección de valores y limpieza de los mismos.

Como todo conjunto de datos, estos están formados por variables de salida y variables explicativas. En el caso de este conjunto de datos, la variable de salida está formado por datos utilizados para predecir si el potencial cliente ha realizado una compra o no en la página. Ésta variable objetivo es de tipo categórica ya que está formada por dos posibles resultados: 'comprador' y 'no comprador'. Por ello, este conjunto de datos se trata de un problema de clasificación binario.

Respecto a las variables explicativas, éste conjunto de datos está compuesto por un total de 29. De estas 29 variables explicativas, 23 son numéricas (Identificador del cliente, el año de nacimiento, ingresos del cliente, niños por casa, jóvenes por casa, inscripción, compra reciente, gasto total en vino, gasto total en carne, gasto total en pescado, gasto total en dulces, gasto total en productos de lujo, número de compras con descuentos, número de compras en la web, número de compras mediante catálogo, número de visitas a la página web, si acepta la oferta en la primera campaña, si acepta la oferta en la segunda campaña, si acepta la oferta en la tercera campaña, si acepta la oferta en la cuarta campaña, si acepta la oferta en la quinta campaña, quejas en los últimos dos

años, coste de contacto, ganancia y respuesta del cliente ante la oferta) y 3 son categóricas (Estado civil, nivel de estudios y país de procedencia del cliente).

Variable	Descripción
ID	ID del cliente
Cumpleaños	Año de nacimiento del cliente
Educación	Nivel de educación del cliente
Estado civil	Estado civil del cliente
País	País de procedencia del cliente
Ingresos	Ingreso familiar anual del cliente
Niños	Número de niños en el hogar del cliente
Adolescentes	Número de adolescentes en el hogar del cliente
Inscripción	Fecha de alta del cliente en la empresa
Última compra	Número de días desde la última compra
Total vinos	Cantidad gastada en vinos en los últimos 2 años
Total frutas	Cantidad gastada en frutas en los últimos 2 años
Total carnes	Cantidad gastada en carnes en los últimos 2 años
Total pescados	Cantidad gastada en pescados en los últimos 2 años
Total dulces	Cantidad gastada en dulces en los últimos 2 años
Total lujos	Cantidad gastada en productos de lujo en los últimos 2 años

Cuadro 2: Descripción de las variables del conjunto de datos

Variable	Descripción
Compras descuentos	Número de compras realizadas con descuento
Compras web	Número de compras realizadas a través de la web
Compras catálogo	Número de compras realizadas por catálogo
Compras tienda	Número de compras realizadas en tiendas
Visitas web mes	Número de visitas a la web en el último mes
Campaña 3	1 si el cliente aceptó la oferta en la 3ª campaña, 0 en caso contrario
Campaña 4	1 si el cliente aceptó la oferta en la 4ª campaña, 0 en caso contrario
Campaña 5	1 si el cliente aceptó la oferta en la 5ª campaña, 0 en caso contrario
Campaña 1	1 si el cliente aceptó la oferta en la 1ª campaña, 0 en caso contrario
Campaña 2	1 si el cliente aceptó la oferta en la 2ª campaña, 0 en caso contrario
Quejas	1 si el cliente se quejó en los últimos 2 años., 0 en caso contrario
Z Coste contacto	Variable constante del coste de contacto
Z ingresos	Variable constante de ingresos
Respuesta	1 si el cliente aceptó la oferta en la última campaña, 0 en caso contrario

Cuadro 3: Continuación descripción de las variables del conjunto de datos

5.2. Tratamiento de la base de datos

La base de datos inicial se llama *Marketing Campaign* [Parr-Rud, 2014]. Tras la investigación por los distintos repositorios de bases de datos, se descubrió que había variables que no se habían pasado de la fuente original. Se buscó otras bases de datos que se habían subido a los repositorios proveniente de la original y se encontró una llamada *Marketing Analytics* que contenía una variable categórica correspondiente al país de origen del consumidor, pero las otras variables categóricas como estado civil y educación venía en formato binario y no en formato categórico. Cómo es más interesante analizar estas dos variables de forma categórica y no de forma binaria, se realizó una concatenación de datos añadiendo a la base de datos original la variable categórica de la segunda base.

Tras revisar los diversos conjuntos de datos, en los que en cada uno faltaba alguna variable o estaba incompleto, se ha optado por completar manualmente el conjunto de datos fusionando datos de dos bases distintas tomando como clave primaria el identificador del cliente. Se han seleccionado los dos conjuntos de datos y se le ha añadido la variable "país de origen del cliente" al conjunto de datos.

Aun así, se ha tenido que realizar un tratamiento de datos ya que en estas dos variables nuevas se encontraban valores redundantes, que a pesar de significar lo mismo, se le habían asignado distinto nombre. En el caso del estado civil, existen 8 tipos de respuestas categóricas distintas. Se puede observar como existen diversas respuestas: absurdo, solo y solo se vive una vez, que hacen referencia a soltero. Por lo tanto, se han renombrado esos valores como soltero directamente.

Para la variable categórica estudios, se vuelven a encontrar respuestas redundantes donde se le dan dos nombres distintos a respuestas que significan lo mismo. Para este caso, se observan 5 posibles respuestas, teniendo la respuesta "segundo grado" el mismo significado que "máster". Como hay mayor cantidad de máster, se ha modificado aquellas que hacen referencia a segundo grado renombrándolos como máster directamente.

Tras esta limpieza previa, se ha continuado analizando la base de datos para saber si hay valores nulos entre ellos. Con un análisis se puede observar cómo se encuentran 24 valores nulos en la columna de ingresos y para solucionar esto, se ha optado por rellenar los valores nulos con la media de dicha columna en vez de eliminar los registros.

Al analizar estadísticamente y representativamente las distintas variables, se puede observar como hay valores en las variables "año de nacimiento" e "ingresos", que distan mucho de la media de los valores totales. En el primer caso, se puede observar como hay tres registros de personas que han añadido una fecha de nacimiento que corresponde a una edad mayor a 100 años. Ante la falta de contexto de estos datos, se ha decidido reemplazar los datos con la media de las edades antes que eliminarlos y perder datos. Con los ingresos sucede algo parecido ya que al realizar un estudio

estadístico, se puede comprobar que el salario máximo es de 666.666 dólares, lo cual vuelve a ser muy diferente a la media de los datos. Al no poseer contexto sobre este dato para saber si es una equivocación o no, se ha optado por reemplazarlo por la media y así no perder los demás datos del usuario.

Se ha seguido tratando el conjunto de datos para poder utilizarlo sin ningún problema. El siguiente paso en el apartado de tratamiento ha sido convertir la variable "fecha de compra" en tipo fecha, ya que los datos que se poseían estaban en formato objeto. Además de esta variable, también se ha convertido en tipo fecha la variable "año de nacimiento". Todo esto se ha realizado para poder sacar la edad del cliente con respecto a la fecha en la que realizó la última compra ya que es más sencillo analizar la edad del cliente que el año de nacimiento.

5.3. Estudio estadístico

5.3.1. Variables cualitativas

La primera parte analizada estadísticamente corresponde a las variables categóricas, es decir, aquellas variables que están formadas por valores cualitativos. Esta base de datos cuenta con tres variables de este tipo: educación, estado civil y país de origen del consumidor.

Analizando el conjunto de datos se observa como hay un total de 2240 instancias para todas las variables. La primera variable cualitativa es la de "educación" que cuenta con 4 valores cualitativos diferentes haciendo referencia al nivel de estudios completado que tiene el consumidor: básico, graduado, máster o doctorado. De estas cuatro, la que posee una mayor frecuencia correspondiéndose con la métrica de la moda, sería un consumidor con un nivel de estudio de graduado. Analizando la frecuencia de esta instancia se observa como del total, 1127 corresponde a los consumidores que poseen esta formación.

La variable cualitativa correspondiente a los distintos estados civiles, se cuenta con 5 instancias

distintas correspondiente a los consumidores: soltero, casado, divorciado, Viudo o pareja de hecho. De estas cinco, el estado civil que más se repite entre los distintos perfiles de los consumidores es casado. Del total de muestras que se posee, hay un total de 864 consumidores que se encuentran casados.

Por último, la variable categórica "país de origen", cuenta con 8 valores cualitativos distintos. Estos valores hacen referencia a los siguientes países: Alemania, Australia, Arabia Saudí, Canadá, España, Estados Unidos, India o México. De estos ocho países de origen, del que más compran es España con una frecuencia total de 1095 clientes.

	Educación	**Estado civil**	**País**
Total	2240	2240	2240
Único	4	5	8
Moda	Graduado	Casado	SP
Frecuencia	1127	864	1095

Cuadro 4: Cuadro estadístico de las variables categóricas

A continuación, se muestran unas series de gráficas correspondientes a las variables cualitativas creadas a partir del conjunto de datos para poder ver los datos de una manera visual y así, poder interpretarlos.

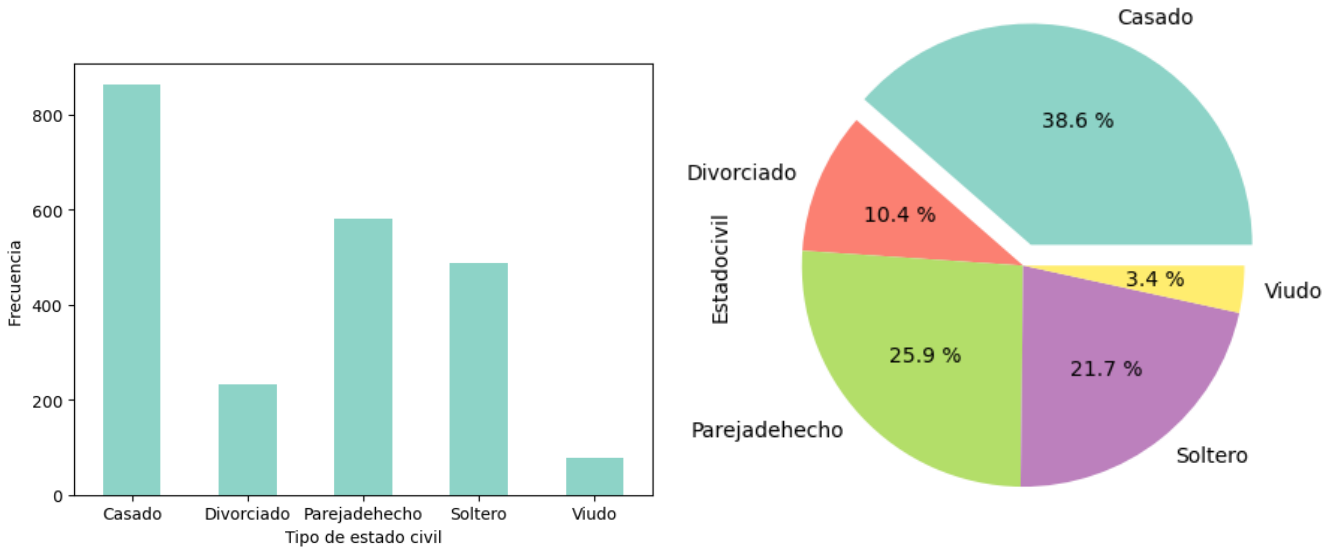


Figura 1: Gráficas representativas del estado civil. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Estas dos gráficas cualitativas representan la distribución de las instancias de la variable "estado civil". En la primera gráfica, en el eje y se representa la frecuencia indicando la cantidad de veces que se repite cada instancia. En la gráfica de la derecha se representa mediante porcentajes para poder ver la cantidad a la que corresponde.

Se puede observar como las instancias correspondientes a la convivencia con otra persona en el núcleo familiar de los consumidores predominan sobre las demás. Mientras tanto, aquellas que corresponden a hogares con solo una persona, tienen una menor frecuencia ya que no es algo tan habitual.

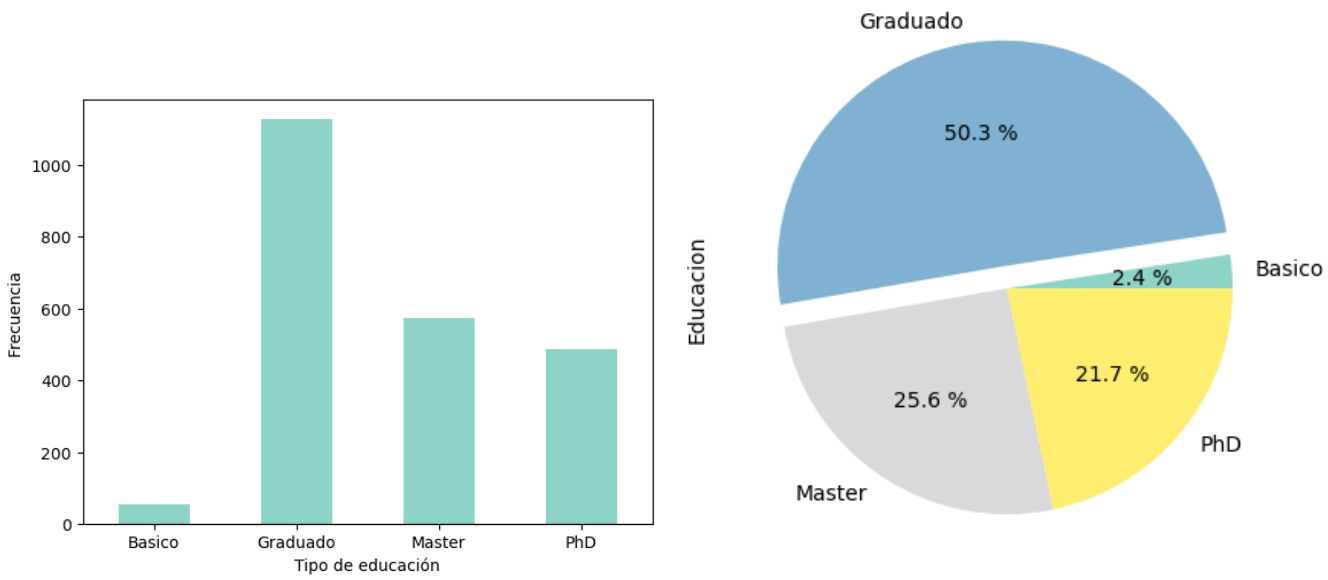


Figura 2: Gráficas representativas del nivel de estudio. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Las dos gráficas siguientes representan la variable cualitativa del nivel de educación que poseen los consumidores de la marca. La primera gráfica es una representación de barras donde se muestra la frecuencia que tiene cada instancia de la variable. En la segunda, se muestran los mismos datos pero recogidos de forma porcentual.

En ellas se observa como hay más consumidores que cuentan con una formación más elevada mientras que solo el 2,4 % de los consumidores solamente están formado con los estudios básicos. El tipo de estudio que más poseen los clientes de la marca es un graduado, que este tipo de formación corresponde a haberse formado en cualquier tipo de carrera universitaria y haberla terminado.

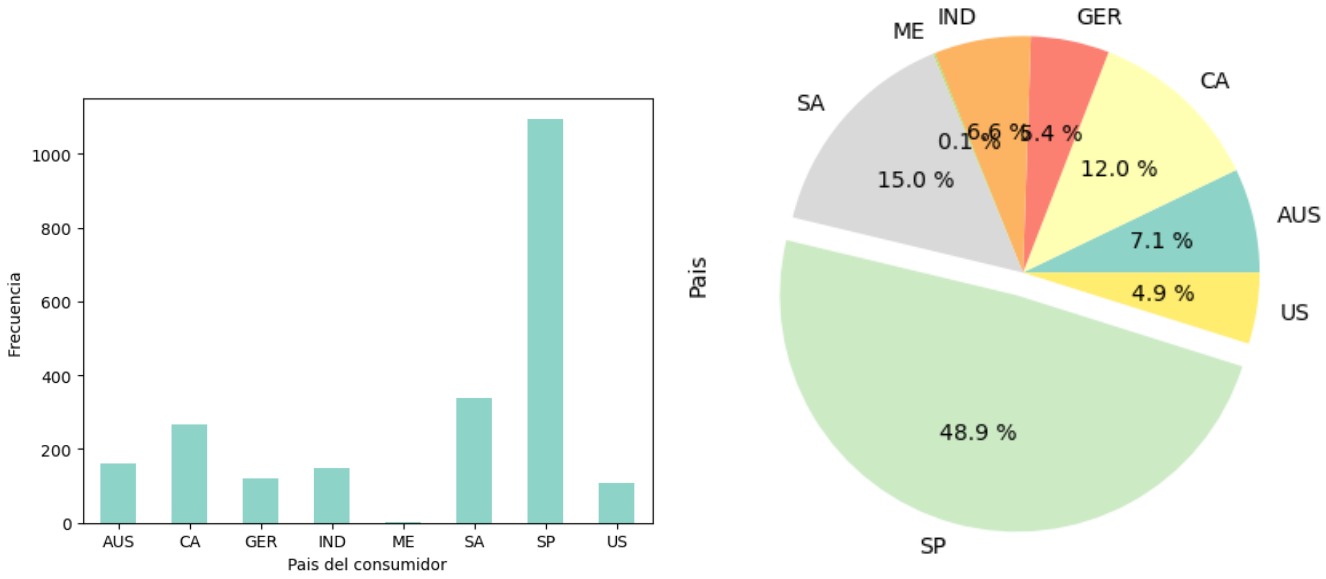


Figura 3: Gráficas representativas del país de origen del consumidor. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

En estas dos últimas gráficas de las variables cualitativas, se representan los distintos países a los que pertenecen los consumidores. Al igual que en los anteriores casos, la primera gráfica es una representación de barras donde se muestra la frecuencia que tiene cada instancia de la variable. En la segunda se muestran los mismos datos pero recogidos de forma porcentual.

Tras un análisis de las gráficas se observa como hay más de 1000 consumidores que pertenecen a España, siendo esta cantidad casi la mitad de las muestras que hay en el conjunto de datos. El resto de instancias se distribuyen entre los otros 7 países a los que pertenecen los clientes. Los clientes procedentes de México son los que menos representación tienen, llegando solo al 0,1 % del total.

Lo importante de analizar estas tres variables cualitativas es poder conocer el tipo de perfil que tienen los consumidores de la marca. De este modo, se pueden aplicar de una forma más eficaz las estrategias de marketing planteadas por la empresa. Si bien es importante conocer los grupos de consumidores que adquieren los productos de la empresa así como las características de estos, es

necesario analizarlo con otras variables para poder sacar más información que ayude a construir las estrategias de marketing.

La última campaña de marketing realizada no tuvo buenos resultados, ya que casi 2000 consumidores rechazaron la campaña creada por la empresa. Si se analiza por los países de origen de los clientes, se observa como México tiene un alto porcentaje de aceptación de las campañas. Es cierto que los consumidores de este país son residuales, pero en comparación con los otros países son los que valoran más positivamente la campaña. El siguiente país que tiene más porcentaje de consumidores que aceptan la campaña creada es España, que coincide con la nacionalidad más repetida en la muestra. Aun así, son muy pocos los clientes que la aceptan.

Con respecto al nivel de estudio, cuanto mayor es el nivel de estudio de los consumidores, más probable es que acepten las campañas de marketing incluso se puede ver que aquellos clientes que poseen un doctorado aceptan en mayor medida las campañas. Por el contrario, los que solo cuentan con unos estudios básicos, es decir, aquellos que no han estudiado ninguna carrera universitaria, son los que menos las aceptan. Es importante saber que a pesar de esto, el porcentaje de aceptación sigue siendo muy bajo pues en la mayoría predomina el rechazo a las campañas.

Por último, los consumidores que no conviven con nadie más son los más propensos a aceptar las campañas de marketing. A pesar de que es el estado civil con menos frecuencia, los viudos son los que más aceptan las campañas. Pasa todo lo contrario con los casados y las parejas de hecho, quienes son los que más se resisten a las estrategias llevadas a cabo por la empresa.

5.3.2. Variables cuantitativas

En esta segunda parte del análisis estadístico se centrará en el estudio de las variables cuantitativas, las cuales son la gran mayoría de este conjunto de datos estudiado.

	Total	**Media**	**Std**	**Min**	**25 %**	**50 %**	**75 %**	**Max**
ID	2240.0	5592.159821	3246.662198	0.0	2828.25	5458.5	8427.75	11191.0
Cumpleaños	2240.0	1968.901786	11.694076	1940.0	1959.00	1970.0	1977.00	1996.0
Ingresos	2240.0	51972.957270	21405.824379	1730.0	35538.75	51741.5	68275.75	162397.0
Niños	2240.0	0.444196	0.538398	0.0	0.00	0.0	1.00	2.0
Adolescentes	2240.0	0.506250	0.544538	0.0	0.00	0.0	1.00	2.0
última compra	2240.0	49.109375	28.962453	0.0	24.00	49.0	74.00	99.0
Total vinos	2240.0	303.935714	336.597393	0.0	23.75	173.5	504.25	1493.0
Total fruta	2240.0	26.302232	39.773434	0.0	1.00	8.0	33.00	199.0
Total carnes	2240.0	166.950000	225.715373	0.0	16.00	67.0	232.00	1725.0
Total pescados	2240.0	37.525446	54.628979	0.0	3.00	12.0	50.00	259.0
Total dulces	2240.0	27.062946	41.280498	0.0	1.00	8.0	33.00	263.0
Total lujos	2240.0	44.021875	52.167439	0.0	9.00	24.0	56.00	362.0
Compras descuentos	2240.0	2.325000	1.932238	0.0	1.00	2.0	3.00	15.0
Compras web	2240.0	4.084821	2.778714	0.0	2.00	4.0	6.00	27.0
Compras catálogo	2240.0	2.662054	2.923101	0.0	0.00	2.0	4.00	28.0
Compras tienda	2240.0	5.790179	3.250958	0.0	3.00	5.0	8.00	13.0
Visitas web mes	2240.0	5.316518	2.426645	0.0	3.00	6.0	7.00	20.0
Campaña 3	2240.0	0.072768	0.259813	0.0	0.00	0.0	0.00	1.0
Campaña 4	2240.0	0.074554	0.262728	0.0	0.00	0.0	0.00	1.0
Campaña 5	2240.0	0.072768	0.259813	0.0	0.00	0.0	0.00	1.0
Campaña 1	2240.0	0.064286	0.245316	0.0	0.00	0.0	0.00	1.0
Campaña 2	2240.0	0.013393	0.114976	0.0	0.00	0.0	0.00	1.0
Quejas	2240.0	0.009375	0.096391	0.0	0.00	0.0	0.00	1.0
Z Coste contacto	2240.0	3.000000	0.000000	3.0	3.00	3.0	3.00	3.0
Z Ingresos	2240.0	11.000000	0.000000	11.0	11.00	11.0	11.00	11.0
Respuesta	2240.0	0.149107	0.356274	0.0	0.00	0.0	0.00	1.0

Cuadro 5: Cuadro estadístico de las variables numéricas

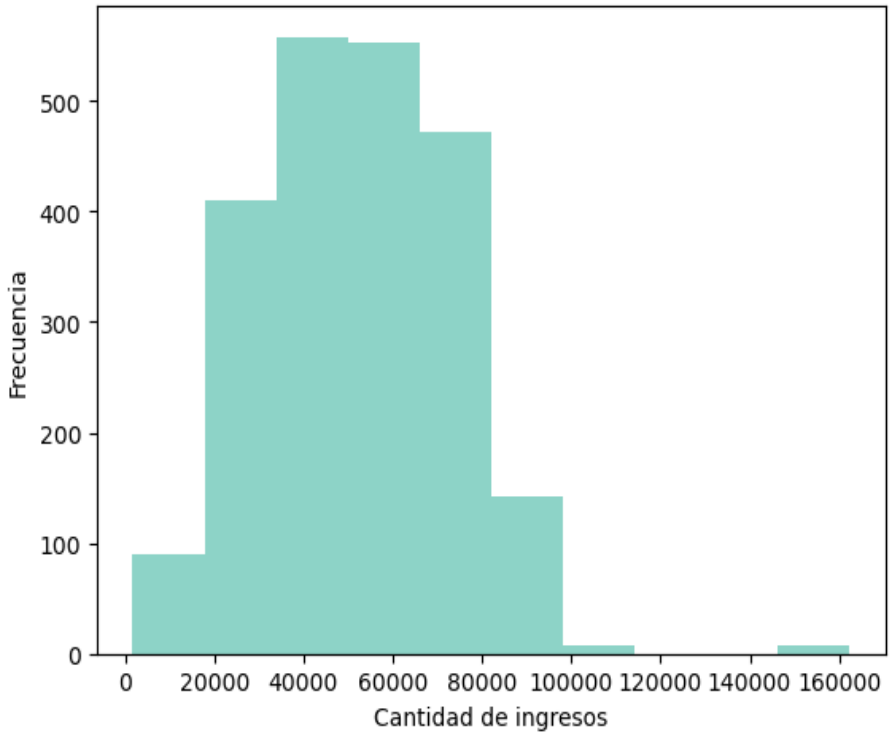


Figura 4: Gráfica representativa de la frecuencia de los distintos ingresos de los consumidores. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Analizar los ingresos que tienen los consumidores puede aportarnos gran información a la hora de realizar las campañas de marketing con sus respectivas ofertas. El ingreso es la base de todo, pues si un consumidor posee un alto salario se puede permitir consumir más productos de la marca.

El intervalo de salario donde se encuentran más consumidores es entre los 40.000 y los 60.000 dólares anuales. Como algo excepcional, se puede observar que existen algunos consumidores que tienen una mejor economía familiar disponiendo unos ingresos anuales aproximados entre 140.000 y 160.000. Conociendo el nivel de vida que hay en los países procedentes de los consumidores y sus salarios, se puede adaptar la oferta realizada teniéndolo en cuenta.

Si se analiza el salario con otras variables, se pueden sacar patrones para analizar. Por ejemplo, en el caso de la variable "estudios", se puede comprobar que cuanta más formación tenga el consumidor, más alto serán probablemente sus ingresos anuales. Por otro lado, analizando la variable "estado

civil", se descubre que aquellas que son menos representativas (viudo y divorciado) son las que tienen unos ingresos medios más altos. A pesar de ser una sola persona la que aporta dinero al hogar, tienen unos salarios más altos que aquellos que los conforman dos. Por último, con respecto a los países, México vuelve a liderar a pesar de ser el país del que provienen menos consumidores. Además, España, que es el país donde la empresa tiene una gran base de clientes, tiene un salario medio más bajo en comparación con México, siendo la diferencia de 6.000 dólares anuales.

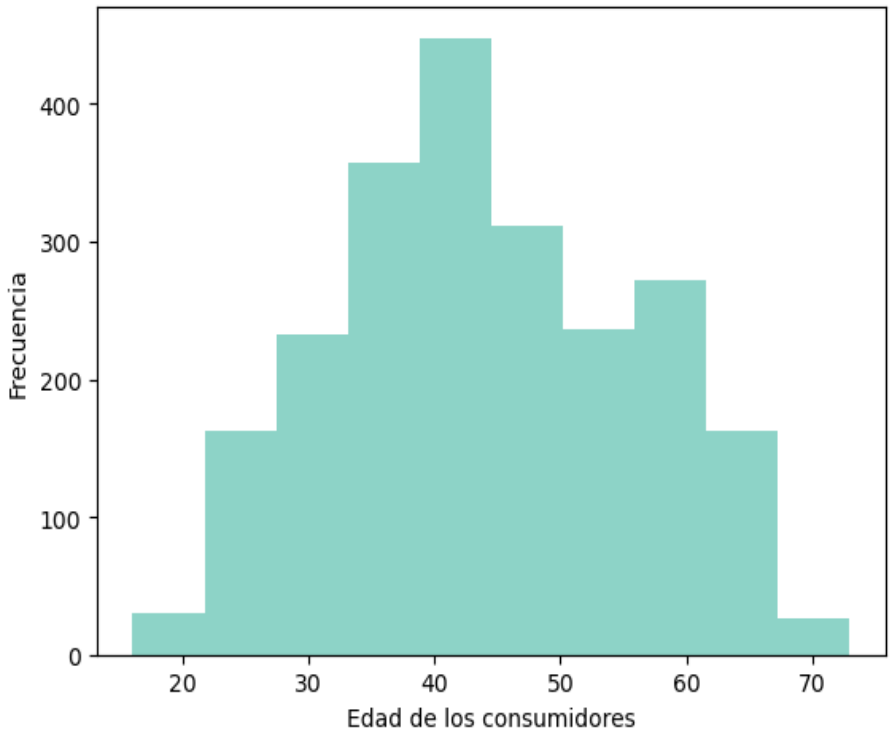


Figura 5: Gráfica representativa de la frecuencia de las diversas edades de los consumidores. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

La distribución de las edades los consumidores de la marca comprende entre los 16 hasta los 73 años. Las edades extremas del intervalo de edad son las que menos frecuencia tienen, mientras que el grueso se encuentra entre las edades centrales de la muestra. El conocer las edades de los clientes puede aportar información sobre el tono que debe tomar la campaña a la hora de realizarla pues no es lo mismo enfocarlo a un público juvenil que a uno adulto.

A pesar de que el grueso de consumidores se encuentra entre el intervalo de 35 a 50 años, si

se analiza en conjunto con otras variables se observan distintos resultados. Por ejemplo, ambos extremos, son los que más han aceptado las campañas de marketing realizadas y se observa observa un patrón parecido a la hora de analizar las edades de los consumidores junto con sus ingresos familiares anuales. Aquellas edades que se encuentran en los extremos son las que tienen de media mejores ingresos familiares en comparación con las edades que más frecuentan la marca.

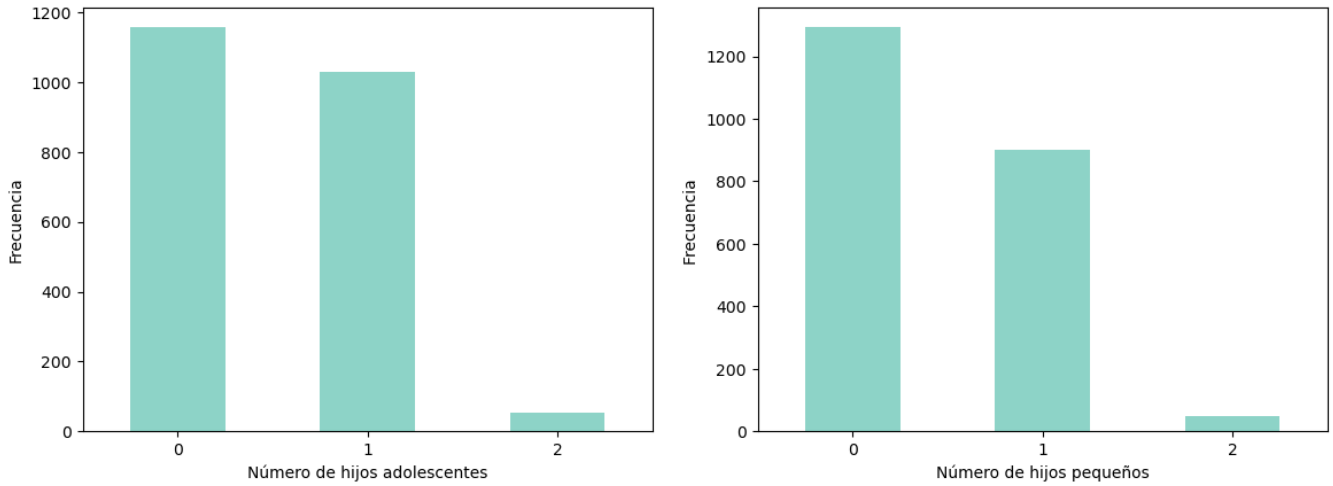


Figura 6: Gráfica representativa de la frecuencia de hijos menores y adolescentes de los consumidores. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Los hijos que tienen los consumidores se dividen según dos etapas de vida. Por un lado, se analizan los hijos pequeños que tiene cada uno y en otra se recopilan aquellos que son adolescentes. En ambos casos, los hijos van desde 0 hasta 2 y analizándolos en conjunto, se observa como no existen consumidores de la marca con más de 3 hijos.

El perfil del consumidor que más hijos tiene de media son aquellos que están casados, a pesar de haber una mayor cantidad de clientes solteros. Analizando el conjunto con los niveles de estudios, el tipo de consumidor que más hijos tiene de media son aquellos que poseen una formación de graduado. En cuanto al país de origen de los consumidores, los que más hijos tienen de media son aquellos que proceden de España, coincidiendo con ser el país de origen con más clientes. Los únicos consumidores que no tienen ningún hijo en su familia son los procedentes de México.

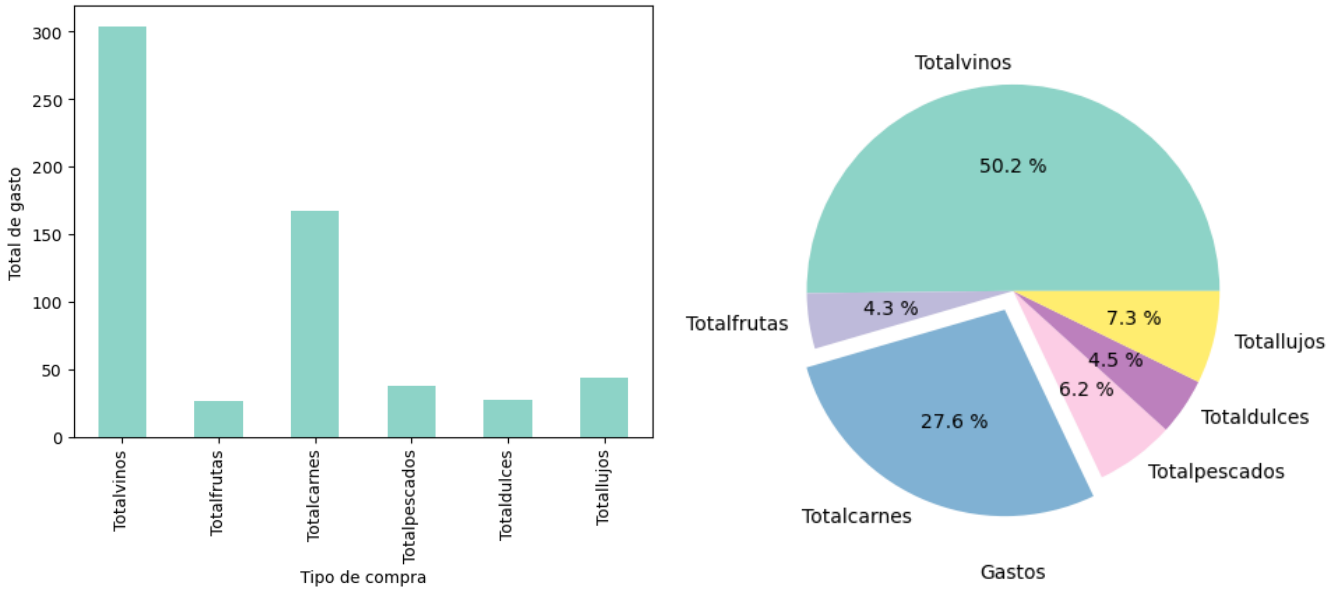


Figura 7: Gráfica representativa de las distintas variables de compras. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

De misma forma que se han analizado las demás variables también es importante conocer en qué gastan el dinero los clientes de la marca, ya que eso ofrece información importante a la hora de crear nuevos productos o de enfocar las campañas de marketing. En las dos gráficas superiores se observa la cantidad de los ingresos que destinan los consumidores para distintos tipos de productos analizados.

Los productos donde los consumidores gastan más son los catalogados en el apartado de vinos, seguido de la carne y siendo el último la fruta. Haciendo un análisis general, los que más gastan en la mayoría de los distintos productos son los consumidores provenientes de México. Los clientes de México se caracterizan por tener unos niveles de estudio básico, así como unos niveles altos de ingresos y unas edades avanzadas. Hay algunas excepciones dentro de los grupos de consumidores: un ejemplo es la asociación entre el gasto en vino y el nivel de estudios elevados, ya que los que poseen una titulación de doctorado son, de media, quienes más consumen y, en el caso del gasto en lujos, quienes más gastan son aquellos que tienen de media los salarios mas bajos. Esto puede ayudar a hacernos una idea del tipo de usuario que más gasta en los diferentes tipos de productos,

para así enfocar de una manera más estratégica las diferentes ofertas.

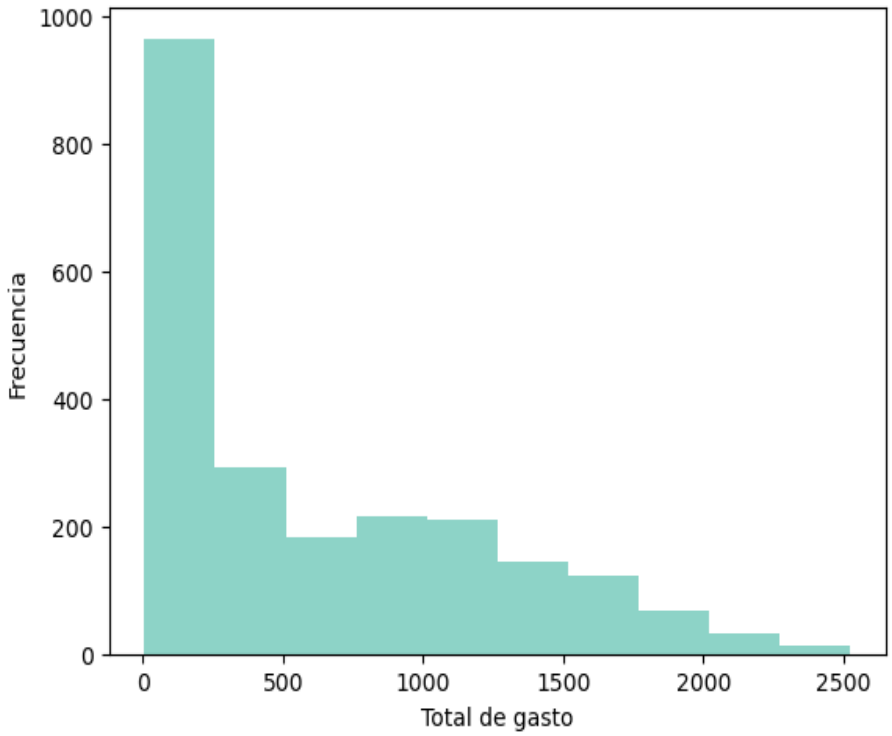


Figura 8: Gráfica representativa de la frecuencia del total de gasto de los consumidores. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Esta gráfica es una recopilación del total de gasto que realizan los consumidores de la marca. Se observa como la mayoría de los clientes gastan en torno a los 150 dólares y que, una vez superada esa cantidad, disminuye de forma casi proporcional la pendiente de gasto.

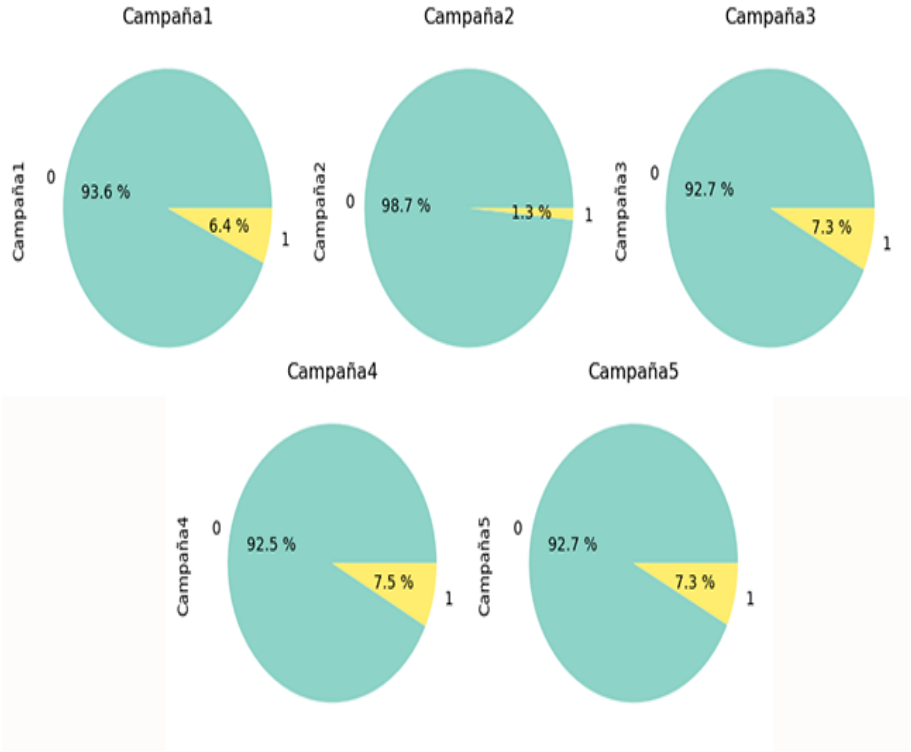


Figura 9: Gráfica representativa de la aceptación de las distintas campañas. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Las cinco gráficas superiores hacen referencia a las distintas campañas realizadas y si los consumidores las aceptaron o no. A simple vista, se observa como hay un gran desequilibrio entre las instancias de las campañas realizadas, predominando en todas el rechazo a la oferta realizada por parte de los clientes.

Analizando cada una por separado, se observa como la primera campaña solo un 6,4 % de los consumidores registrados aceptaron la estrategia de marketing realizada por la empresa. La que más se diferencia es la segunda campaña pues solo el 1,3 % de los consumidores la aceptaron. La últimas tres campañas obtuvieron resultados parecidos, siendo estos entorno al 7,3 %. Estos resultados son negativos pues son muy pocos los usuarios que aceptaron las ofertas realizadas por la empresa y pudo suponer más inversión a la hora de realizarlas que el beneficio obtenido por ellas.

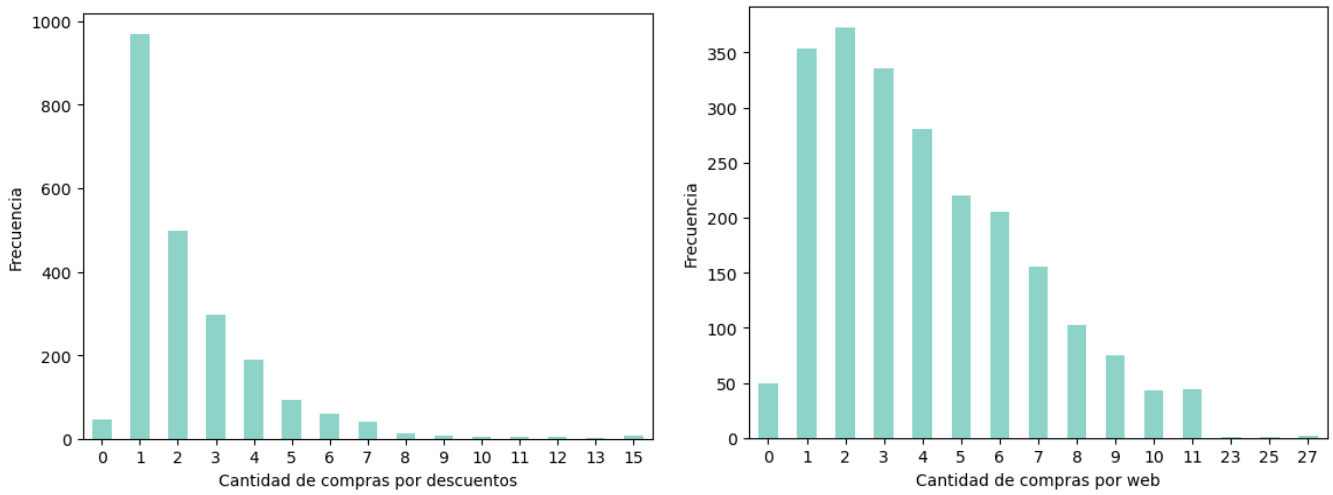


Figura 10: Gráficas representativas de la frecuencia de compras con descuentos y por web por parte de los consumidores. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

En estas gráficas se analizan el número de compras que realizan los consumidores por distintos tipos de canales de distribución. En la primera, los usuarios realizan sus compras mediantes descuentos proporcionados por la marca sobre sus productos. En ella se observa como la mayoría de los consumidores solamente realizan una compra mediante este tipo de oferta. Tras esto, la cantidad de usuarios que compran más de un producto con algún tipo de descuento empieza a tender a cero, siendo el máximo de descuentos aceptado por un consumidor de 15 compras.

En la segunda gráfica se ve representada las distintas compras realizadas por la página web de la empresa. A diferencia de los descuentos, los consumidores realizan mayor cantidad de compras por este canal de distribución. Se mantiene un poco constante a partir de la primera compra hasta la cuarta y a partir de ahí empieza a bajar la cantidad de consumidores que realizan mayores cantidades de compras. Hay algunos clientes que han llegado a realizar hasta 27 compras por página web, mientras que por descuento el máximo de repetición era de 15.

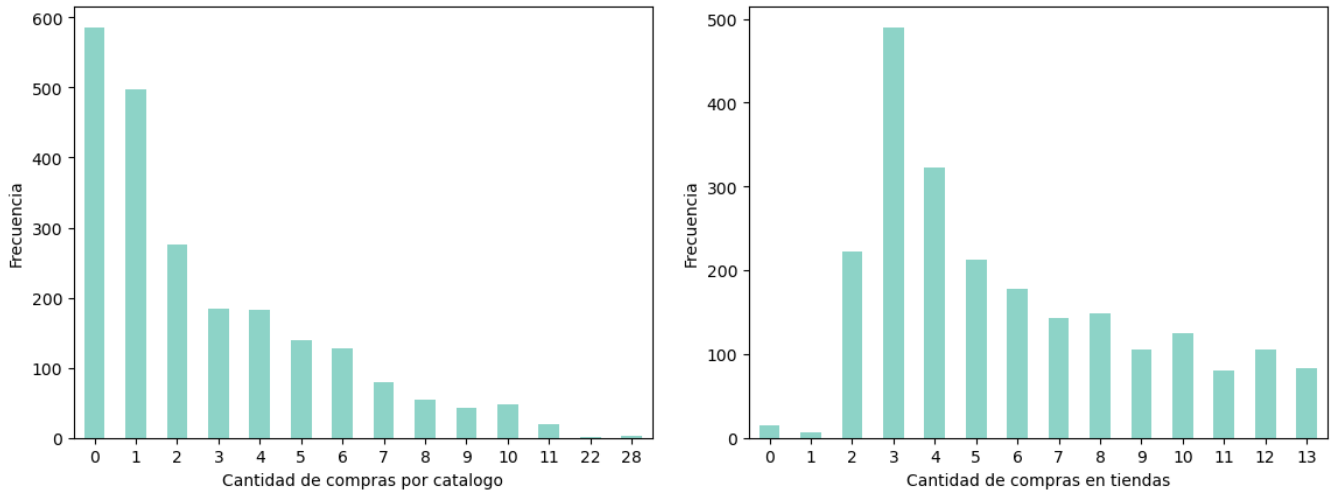


Figura 11: Gráficas representativas de la frecuencia de compras por catálogo y en tienda por parte de los consumidores. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Otro de los canales de distribución que utiliza la empresa es mediante catálogo. Este tipo de canal de distribución consiste en el acceso del consumidor a una especie de revista donde se encuentran todos los productos y tras esto pueden elegirlo mediante teléfono, correo o por plataformas en línea. En este caso, se observa una alta frecuencia de consumidores que no realizan ninguna compra. Así mismo, tras realizar una compra, disminuye la cantidad de consumidores que realizan más de una hasta tender a cero.

Por último, en cuanto a las tiendas físicas, se observa como la gran mayoría de los consumidores realizan más de dos compras de los productos de la marca. Una gran cantidad de consumidores realizan tres compras y tras esta cantidad, el número de clientes que realizan más cantidad de compras empieza a disminuir. Para las compras en físico, el máximo de repeticiones realizadas es de un total de 13 compras. Con esta información, se puede saber en qué tipo de canales realizan más compras y menos para poder adaptar las ofertas a aquellos lugares que visiten más.

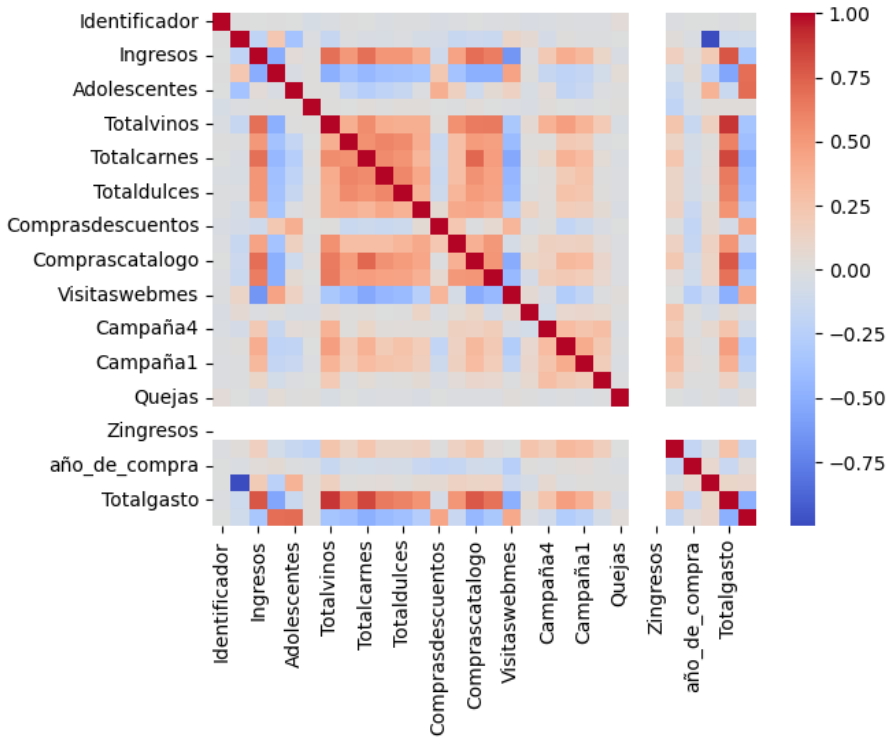


Figura 12: Gráfica representativa de la correlación de las variables. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

La gráfica que se presenta arriba corresponde a una gráfica de correlación realizada en base a las distintas variables pertenecientes al conjunto de datos. En ambos ejes, se encuentran representados las distintas variables, cada una con su tipo de medidas correspondientes.

La diagonal principal corresponde a la correlación que posee cada variable consigo misma. Por ello, se observa una diagonal con un resultado de 1, representando esto una correlación positiva. Los valores de correlación van desde -1 hasta 1, indicando el primero una correlación negativa entre dos variables y la segunda una correlación positiva.

Analizando ya los datos de la gráfica presente, se puede observar como hay una correlación positiva entre los diversos gastos que realizan los clientes en distintos productos. Esto significa que cuando el gasto en uno de los productos aumenta, el gasto en el otro producto también aumenta y viceversa. Cuanto más gasta el consumidor por ejemplo en productos cárnicos, también aumenta el gasto en

vino.

Agregando a lo anterior, se observa una correlación positiva entre los distintos gastos de productos y los distintos canales de distribución que posee la marca. Si un cliente aumenta el gasto en cualquiera de los productos que aparecen en las variables analizadas, aumentará también el número de compras en los distintos canales ya que poseen una correlación positiva.

Sin embargo, no todas las variables presentan una correlación positiva. Se puede observar como la variable de visitas que realiza el usuario a la página web al mes, presenta una correlación negativa con los ingresos familiares del consumidor. Esto quiere decir que cuantas más visitas realizan a la página web, menores ingresos poseen los consumidores. Esto puede deberse, por ejemplo, a que al tener unos ingresos menores, el usuario tienda a pasar más tiempo en la página pensando que producto comprar.

5.4. Estudio

Hasta ahora, se ha analizado el conjunto de datos y cómo interactúan las variables entre sí para sacar información de utilidad para el departamento de marketing. En este apartado del trabajo de fin de máster se llevará a cabo diversas técnicas de aprendizaje automático para poder dar solución a los objetivos planteados en un inicio.

Primero, se llevará a cabo la preparación del conjunto de datos para poder aplicarle los métodos seleccionados de agrupación y clasificación de *machine learning*. Como el conjunto de datos está formado por variables categóricas, se optó por transformar estas variables de forma binaria mediante la técnica *one-hot* y así poder utilizarlas en el algoritmo posteriormente.

Para poder llevar a cabo los distintos métodos planteados, se ha realizado una separación entre las variables explicativas y la variable explicada. La realización de esta separación es necesaria para poder crear luego el conjunto de entrenamiento y el conjunto de test para poder aplicar las técnicas de aprendizaje automático.

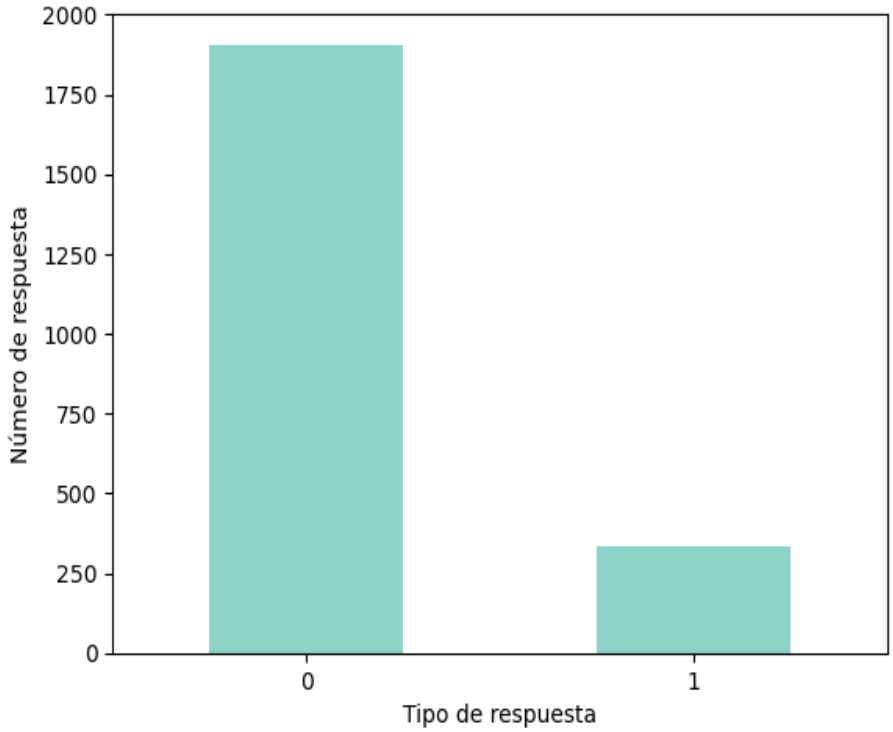


Figura 13: Gráfica representativa de variable respuesta. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Como se observa en la gráfica superior, las clases se encuentran desbalanceadas, esto quiere decir que existen clases minoritarias. Mediante la aplicación de la técnica *SMOTE*, se puede crear nuevas instancias sintéticas que permiten crear un equilibrio entre la clase minoritaria y la mayoritaria. De este modo, se evita que surjan sesgos a la hora de realizar los entrenamientos al existir clases con menor representación. La aplicación de esta técnica ayuda a mejorar la capacidad predicativa del modelo de aprendizaje automático realizado.

El último paso que se realizó antes de aplicar los algoritmos correspondientes fue la división del conjunto en dos conjuntos distintos. Por un lado, se creó un conjunto de entrenamiento al que se le asignó un 75 % de los datos de la base de datos y por otro lado, se creó un conjunto de prueba con el resto de los datos, un 25 %. Esta división supone que el número de instancias que conforman el conjunto *train* sea de 2859 mientras que el conjunto *test* está formado por 953 instancias.

5.4.1. Árbol de decisión

Para dar respuesta al cuarto objetivo marcado, se realizarán técnicas de clasificación para poder predecir si el consumidor aceptará la oferta realizada por parte de la marca o no. Si bien existen diversas técnicas que se podrían aplicar al caso, se ha optado por realizar un árbol de decisión.

Una vez se ha decidido la técnica a utilizar, se aplicará un algoritmo de árbol de decisión extraído de las librerías en Python de *machine learning*. Para ello, se crea un objeto utilizando el clasificador seleccionado y luego se entrena utilizando los conjuntos creados anteriormente para el *test*. De este modo, el algoritmo creado aprenderá de tal forma que será capaz de tomar distintas decisiones en base a los conjuntos de datos ofrecidos.

Una vez se tiene el algoritmo creado, se realiza una predicción utilizando este modelo. En este caso, a la hora de evaluar el rendimiento del modelo creado, se obtiene un porcentaje de acierto en el conjunto de entrenamiento del 100 %, mientras que para el conjunto *test* solo se obtiene un 86,43 %. Con esto se puede saber que el modelo que se está aplicando está clasificando bien las instancias del conjunto de prueba.

		Predicción	
		Positivos	*Negativos*
Observación	*Positivos*	416	58
	Negativos	71	408

Cuadro 6: Representación de la matriz de confusión. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

La realización de la matriz de confusión permite ver de forma visual la calidad de la predicción realizada y así poder evaluar su capacidad de clasificar correctamente las clases. En esta tabla se recogen los resultados obtenidos tras clasificar las instancias en las distintas combinaciones existentes de clase y se observa cómo se han clasificado correctamente un total de 416 instancias como positivas y 408 como negativas. Respecto a las clasificadas incorrectamente, vemos 58 falsos

positivos y 71 falsos negativos.

Por otra parte, para comprobar si el modelo ejecutado es bueno o no, se han realizado diversas métricas para evaluar la clasificación hecha por el árbol de decisión. La primera métrica evaluada ha sido la precisión, la cual indica el porcentaje de instancias clasificadas correctamente. En esta métrica, los resultados se dividen según los dos tipos de instancias que tiene la variable "respuesta", es decir, 0 para cuando el consumidor rechaza la oferta y 1 para el caso contrario. Tras analizar los resultados se ve como el 88 % de las instancias clasificadas como 0 se han clasificados correctamente bajo esta etiqueta. En cambio, para el caso de las instancias que representan la aceptación de la oferta, solo el 85 % se han clasificado correctamente.

Agregando a lo anterior, se ha analizado también la exhaustividad que estudia la proporción de instancias positivas en relación con el total. Para el caso de las que representan el rechazo de la oferta, el 85 % fueron clasificadas correctamente, y para aquellas que representan la aceptación de la oferta, el 88 % se clasificaron bien.

Para combinar ambas métricas se utiliza el *F1-score*, que da un resultado general del rendimiento del modelo seleccionado. En el caso de la clase 0, la puntuación que se obtiene es de 86 %, y para la clase 1, de 87 %. También se han analizado otras métricas como la exactitud o también conocido como *accuracy*. En esta métrica se mide el total de instancias clasificas correctamente con respecto al total de las mismas. Viendo los resultados obtenidos, se ve como el algoritmo creado arroja una exactitud del 86 %.

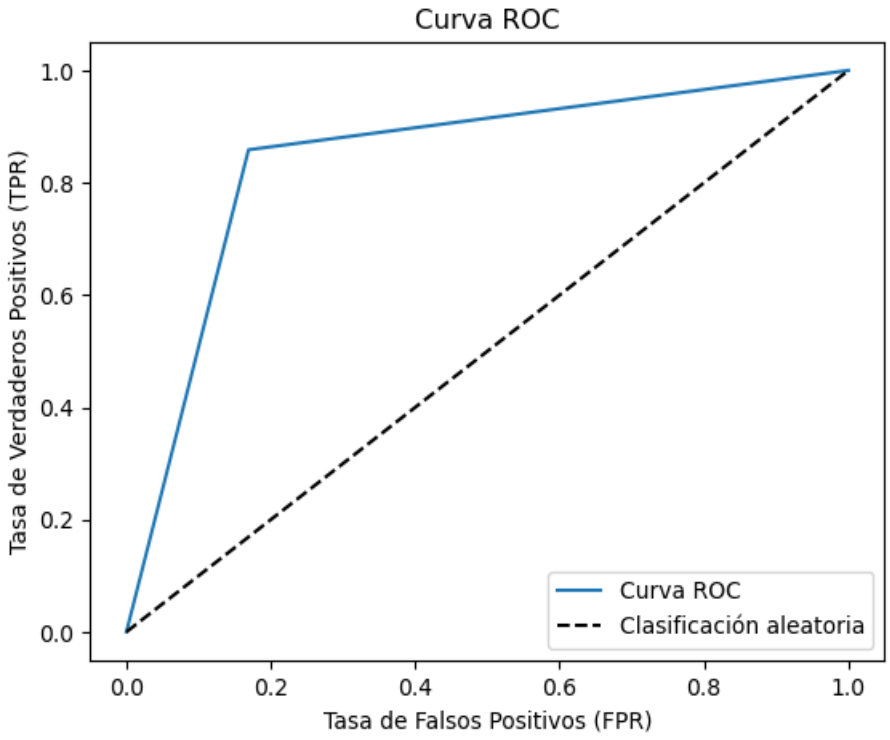


Figura 14: Gráfica representativa de la curva roc. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

La gráfica superior representa la curva ROC (*Receiver Operating Characteristic* que se utiliza, al igual que las anteriores métricas, para evaluar el algoritmo creado. En ella se representan los verdaderos positivos y los falsos positivos. Cuanto más se acerque la curva ROC, es decir, la línea continua azul, a la esquina superior izquierda que representa el 1, mejor será el rendimiento del algoritmo generado. La línea discontinua gris que se encuentra por debajo de la curva ROC determina si el modelo posee capacidad para clasificar las instancias analizadas y diferenciarlas entre ellas. En este caso, el resultado del área bajo la curva ROC es de 0.8647, queriendo decir que el modelo tiene buena capacidad para realizar una clasificación precisa.

5.4.2. Árbol de clasificación con *cross-validation*

Como todo modelo de aprendizaje automático, es recomendable compararlo con otro algoritmo para comprobar si hay algún otro que arroje unos resultados mejores. Por ello, se ha optado por realizar el mismo proceso pero aplicándole un método de validación cruzada. A diferencia del algoritmo aplicado en el apartado anterior, en este caso se realiza una subdivisión de los conjuntos y se van realizando distintas iteraciones donde cada una se combina con un subconjunto distinto.

Las subdivisiones que se suelen hacer para este tipo de proceso suele ser entre 5 y 10. Para esta ocasión, se ha optado por realizar 5 subdivisiones ya que los resultados al aumentar las divisiones apenas variaban. Una vez se han realizado todas las subdivisiones y se han iterado, se realiza un promedio de los resultados de todas para obtener una media general de los resultados del algoritmo del árbol de clasificación. Las precisiones obtenidas son de 84,37 % 17,63 %, 84,15 %, 85,71 % y 82,58 %.

Tras las divisiones y el cálculo de la precisión de cada una de las divisiones, al igual que en el apartado anterior, se realiza el cálculo de la predicción del modelo. Utilizando este proceso, se obtiene un acierto del conjunto de entrenamiento del 100 %. En cambio, en la evaluación del rendimiento del conjunto de prueba se ha obtenido un 86,14 %. Con esto, podemos saber que el algoritmo creado con el proceso de validación cruzada está clasificando bien las instancias.

		Predicción	
		Positivos	*Negativos*
Observación	*Positivos*	422	52
	Negativos	80	399

Cuadro 7: Representación de la matriz de confusión. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

La matriz de confusión se realiza para poder evaluar la precisión de la predicción realizada, ya

que permite ver de una forma visual qué tan bien se clasifican las clases de manera correcta. Esta tabla muestra los resultados de la clasificación de las instancias donde se han utilizado diversas combinaciones de clases. Como puede verse en la matriz, 422 casos se han identificado correctamente como positivos y 399 como negativos. Los valores incorrectos dan un total de 52 falsos positivos y 80 falsos negativos.

Por otro lado, se ha utilizado una serie de métricas para evaluar la clasificación que hizo el árbol de decisiones con el fin de determinar si el modelo se implementó con éxito o no. La precisión, que representa el porcentaje de instancias correctamente clasificadas, ha sido la primera métrica evaluada. Los resultados se dividen en esta métrica según los dos tipos de instancias que tiene la variable respuesta, 0 para cuando el cliente rechaza la oferta y 1 para el caso contrario. Tras el análisis de resultados, quedó claro que el 88 % de las instancias etiquetadas como 0 se habían asignado de forma correcta a esta categoría. Por el contrario, solo el 84 % de las instancias que representan la aceptación de la oferta, han sido correctamente clasificadas.

Además de lo mencionado anteriormente, también se ha examinado la exhaustividad, que analiza el porcentaje de instancias positivas con respecto al total. Las instancias que representan el rechazo de la oferta se clasificaron correctamente en el 83 % de los casos, y las instancias que representan la aceptación de la oferta se clasificaron correctamente en el 89 % de los casos.

El *F1-score*, que proporciona una evaluación amplia del rendimiento del modelo elegido, se utiliza para combinar ambas métricas examinadas. Ambas clases recibieron una puntuación general del 86 %. También se ha realizado un análisis de otra métrica como el *accuracy*, que mide el número de instancias que clasifica correctamente con respecto al número total de las mismas. Al examinar los resultados, está claro que el algoritmo producido obtiene unos resultados con una precisión del 86 %.

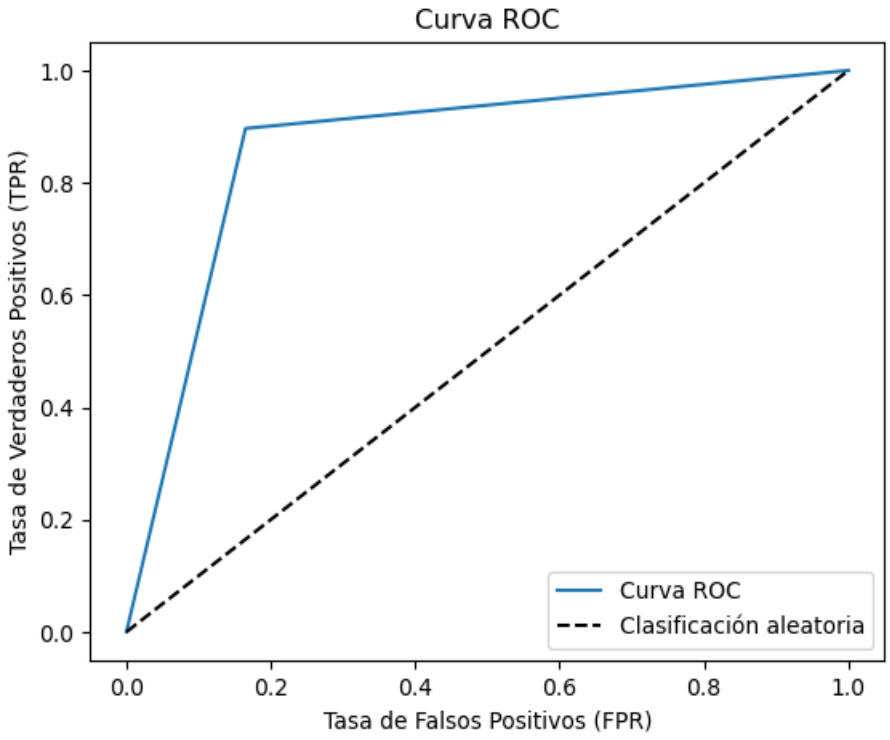


Figura 15: Gráfica representativa de la curva roc. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

En el gráfico superior, al igual que se ha contemplado antes, se muestra la curva ROC. En ella se ve de forma gráfica tanto los positivos reales como los falsos. El rendimiento del árbol generado será mejor cuanto más cerca esté la curva ROC, representada por la línea azul, de la esquina superior izquierda, que representa el valor 1. La línea gris discontinua de la curva ROC indica si el modelo es capaz de clasificar las instancias analizadas y diferenciarlas. El modelo puede realizar una clasificación precisa en este caso porque el área bajo la curva ROC es de 0,8616.

5.4.3. *Clustering*

El *clustering* en el ámbito del marketing se utiliza para poder encontrar patrones entre los diversos perfiles de los consumidores y poder agruparlos según comportamientos parecidos. Esto se realiza para poder segmentar de una mejor forma el mercado y poder personalizar las estrategias de marketing según los perfiles. Por ello, para dar explicación y solución a algunos de los objetivos que se han planteado en este trabajo de investigación, se ha realizado la aplicación de los algoritmos de *clustering*.

Lo primero que se ha realizado en esta parte ha sido determinar el número óptimo de *clusters* que se van a utilizar a la hora de aplicar el algoritmo de *K-means*. Para ello, al igual que en el caso de los algoritmos de clasificación, se han tratado las variables categóricas mediante la técnica de *one-hot*. De este modo, las variables cualitativas pueden ser procesadas por el algoritmo de agrupación. Una vez que las variables se han convertido en variables binarias, se crea una lista vacía para poder almacenar la suma de los cuadrados de las distancias para los distintos números de *cluster* (k). Para obtener dicha suma, se ha creado un bucle que va iterando los distintos valores.

Gracias a este tipo de método, se puede ver de forma visual el número de valores de *clusters* que se debe seleccionar para tener un número óptimo de agrupaciones. En la gráfica que podemos ver a continuación, se muestra la suma de los cuadrados de las distancias que supone la variabilidad de los distintos *cluster* y el número de *cluster*. En este caso, como se creó un bucle de entre 1 y 15, el máximo de *cluster* analizado será 14. Para seleccionar el número óptimo se debe observar el punto de inflexión donde la gráfica tiene un cambio y empieza a tender a cero. En el caso de este conjunto de datos estudiados, se ve a simple vista como la disminución de los datos empieza a partir del valor 3. Si bien es cierto que el estudio se podría realizar con el valor 2 o con el 4, esto podría suponer la pérdida de información al ser tan pocos agrupamientos o la obtención de unos resultados bajos.

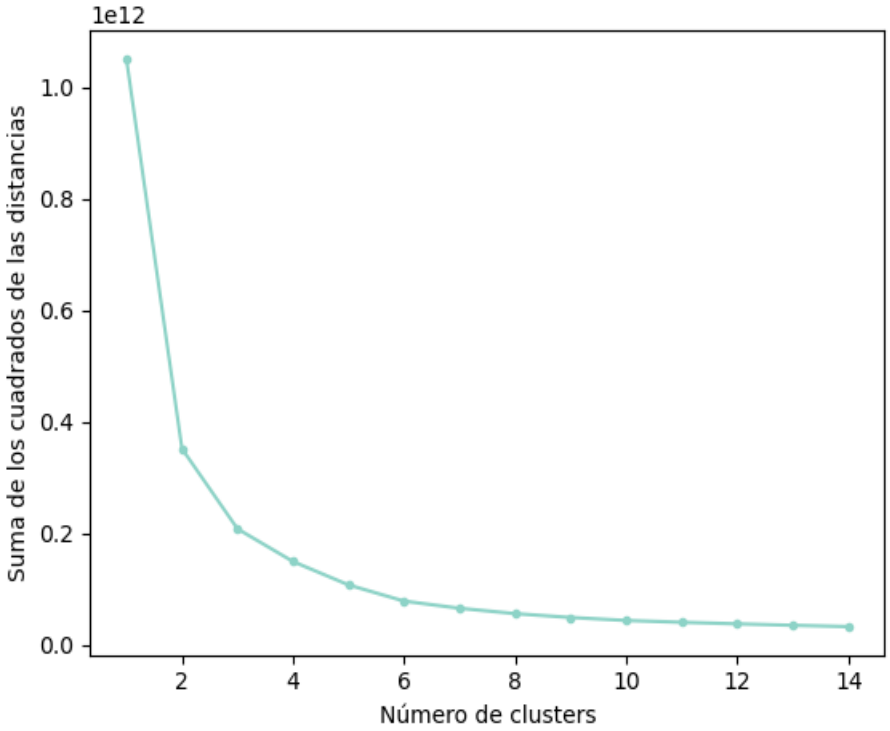


Figura 16: Gráfica del método codo. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Una vez se ha determinado el número de *clusters* a analizar, se crea un algoritmo utilizando la librería de *machine learning* que se ha estado utilizando a lo largo de toda la investigación. Tras esto, se entrena el modelo utilizando el algoritmo de *K-means*. Conforme se va entrenando, se le va asignando a cada instancia uno de los tres *cluster* realizados y se ajustan los centroides, que representan los puntos centrales de los *cluster* y, además, se busca minimizar la suma de los cuadrados de las distancias.

Tras obtener los distintos *clusters* con las instancias clasificadas en el que le corresponde, se debe realizar una evaluación del algoritmo para comprobar si realmente se ha ejecutado correctamente y si los resultados son adecuados. Para ello, se ha usado la métrica de silueta, la cual consiste en evaluar la homogeneidad en los *clusters* y la separación entre ellos. Cuanto más cercano a 1 sea el resultado más parecidas serán las instancias dentro de cada uno y más separados se encontrarán los grupos. En este caso, el resultado que se obtiene es de 0.4914, indicando que existe cierto grado de superposición o cercanía entre distintas instancias de diversos *clusters*.

Clusters	0	1	2
Identificador	5377.431034	5770.388646	5662.222672
Cumpleaños	1966.140394	1967.531295	1973.198381
Ingresos	52385.061926	76967.652111	28348.147099
Niños	0.416256	0.084425	0.808367
Adolescentes	0.815271	0.350801	0.311741
Ultimacompra	49.624384	49.091703	48.561404
Totalvinos	288.646552	616.861718	30.568151
Totalfrutas	18.821429	57.052402	5.990553
Totalcarnes	100.912562	397.494905	25.570850
Totalpescados	25.158867	82.835517	9.068826
Totaldulces	18.320197	60.053857	6.056680
Totallujos	45.883005	70.196507	17.715250
Comprasdescuentos	3.100985	1.605531	2.141700
Comprasweb	4.732759	5.398836	2.156545
Comprascatalogo	2.243842	5.457060	0.529015
Comprastiendas	6.051724	8.401747	3.082321
Visitaswebmes	5.692118	3.155750	6.908232
Campaña3	0.065271	0.068413	0.085020
Campaña4	0.087438	0.135371	0.004049
Campaña5	0.006158	0.229985	0.000000
Campaña1	0.020936	0.183406	0.001350
Campaña2	0.014778	0.026201	0.000000
Quejas	0.004926	0.007278	0.016194
Zcostecontacto	3.000000	3.000000	3.000000
Zingresos	11.000000	11.000000	11.000000
Respuesta	0.108374	0.234352	0.114710

Cuadro 8: Cuadro con los valores cuantitativos medios de cada variable en los distintos *clusters*. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Tras realizar un algoritmo de *clustering* e identificar los tres tipos de agrupamiento distintos, se han obtenido los diversos valores medios de cada variable clasificada. Esto se observa en la tabla superior, donde se muestran los valores de cada variable por cada *cluster* realizado.

A modo de ejemplo, se puede contemplar cómo la variable "ingresos" se ha divido en tres agrupamientos distintos. Los ingresos familiares del primer *cluster* rondan la media de 52385 dólares anuales, mientras que en el *cluster* segundo el salario es mucho más alto, siendo este de 76967 dólares. En cambio, en el último *cluster* se observa como los ingresos familiares son mucho más bajos que en los casos anteriores, con un promedio de 28348 dólares. Si se siguen analizando las demás variables, se puede apreciar una especie de patrón donde se le asigna al primer *cluster* los valores medios, al segundo *cluster* los valores más altos y al tercer *cluster* los valores más bajos.

Clusters	0	1	2
País	US	US	US
Educación	Graduado	PhD	PhD
Estado civil	Soltero	Casado	Pareja de hecho

Cuadro 9: Cuadro con los valores cualitativos de cada variable en los distintos *clusters*. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

Al igual que con las variables numéricas, se han obtenido la moda de cada variable categórica para los tres tipos de *clusters* creados con el algoritmo. En estas variables se puede apreciar cómo existen algunas instancias que se pisan entre distintos *clusters*. Por ejemplo, el país que más se repite en los tres es Estados Unidos, lo cual podría suponer que alguno se superponga ya que el resultado de silueta obtenido era de 0.5 aproximadamente. En cambio, en la variable estado civil sí se aprecia una mejor separación entre las instancias, asignándole a cada agrupamiento un tipo de estado distinto.

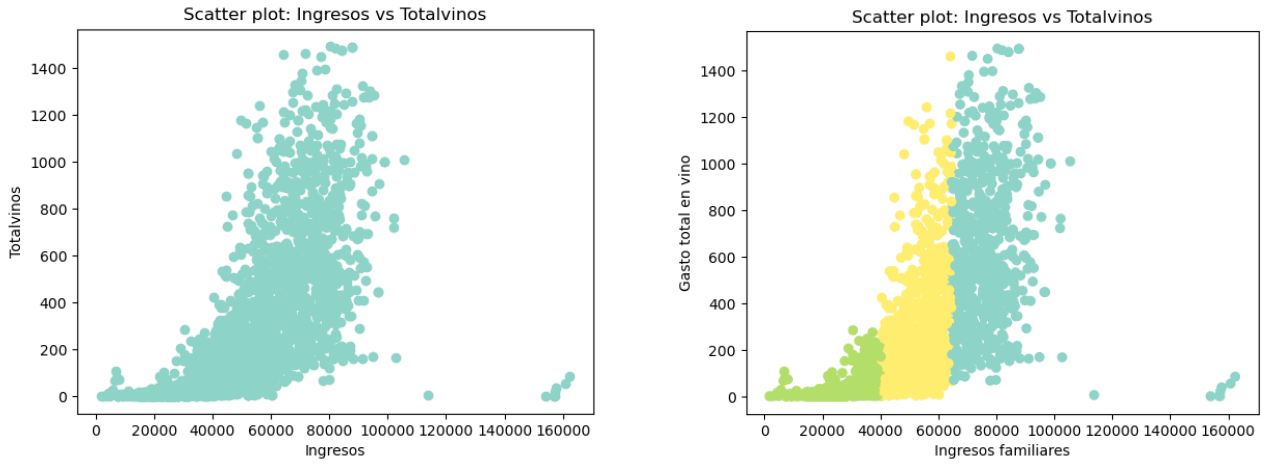


Figura 17: Gráficas representativas de la diferencia entre las instancias sin agrupar y agrupadas. Elaboración propia con datos extraídos de [Parr-Rud, 2014]

En las gráficas superiores se puede ver dos gráficos de dispersión. Tomando como ejemplo las variables ingresos y gastos totales en vino, se observa cómo interactúan entre sí ambas. Analizando estas dos variables se nota como los puntos se mueven hacia la derecha conforme se aumenta el valor del eje de la x, suponiendo esto una relación positiva entre ellas. Además, se observa una cohesión entre las variables debido a la proximidad que existe entre los puntos. También se pueden observar algunos valores extremos en cuanto a los ingresos. En la parte de limpieza del conjunto de datos se decidió dejar al interpretarse como unos sueldos más altos que la media y no errores por parte del consumidor.

Comparando las dos gráficas, más allá del análisis anterior que se puede ir realizando con las diversas combinaciones de variable, la principal diferencia entre ambas es la existencia de la división en colores. Esta división por color corresponde a los tres tipos de *clusters* que se han ejecutado. De este modo, resulta más fácil agrupar clientes que tienen características en común y enfocar las mismas estrategias de marketing al mismo segmento. Por ejemplo, un grupo de clientes serían aquellos que poseen bajos ingresos y que gastan poco en vinos mientras que otro serían aquellos que poseen unos ingresos familiares altos y que gastan una mayor cantidad en este tipo de productos.

6. RESULTADOS

En esta sección, se resumen los resultados obtenidos a partir del análisis del conjunto de datos *Marketing Campaign* de forma objetiva y precisa. A lo largo del estudio se han examinado datos cuantitativos y cualitativos mediante la aplicación de técnicas de aprendizaje automático que permiten examinar el comportamiento de los consumidores.

Durante la realización del código de *machine learning* se han realizado distintos métodos para obtener aquel que arrojase resultados de mayor precisión a la hora de clasificar las instancias. Uno de los comprobados, que se puede observar el resultado en el anexo adjunto al estudio, fue un *KNN* donde se obtuvo un *accuracy* de 0.7848. De entre los realizados, se optó por ejecutar un método de árbol de clasificación, ya que se obtenían mejores resultados a la hora de clasificar. Como se busca un modelo que prediga si el cliente va a aceptar la oferta o no realizada por parte de la marca, es necesario obtener un alto porcentaje de *accuracy*.

Una vez que se optó por un método de *machine learning*, se procedió a realizar dos tipos del mismo código. En uno, se ejecutó tal cual el árbol de decisión, mientras que en el otro se realizó un proceso de validación cruzada donde se divide el conjunto de datos en subconjuntos para obtener un rendimiento más confiable ya que utiliza al máximo los datos. Pero el resultado obtenido tras la realización de los mismos nos muestra como la diferencia entre ambos a la hora de examinar la exactitud era de centésimas. Como se ha mencionado en la parte del estudio, y se puede comprobar en el anexo adjunto, el primer árbol de decisión arroja un *accuracy* de 0.8646 mientras que el árbol de decisión con validación cruzada es de 0.8614. Esta diferencia mínima no aporta ningún beneficio a la hora de realizar este segundo modelo ya que al realizar múltiples iteraciones, el algoritmo tarda más en procesarse.

En conclusión, aparte de buscar un modelo que clasifique correctamente las instancias analizadas, también hay que buscar un modelo que se ejecute de manera rápida para poder dar solución cuanto antes a los problemas planteados. Por ello, la realización de un algoritmo de árbol de decisión es un buen método para poder predecir si un cliente acabará aceptando la oferta lanzada por la empresa

o no.

En esta misma línea, se ha realizado un algoritmo de agrupación para poder dar respuesta a otros objetivos marcados en este trabajo de investigación. Para ello, se ha elegido la realización de un algoritmo de agrupación *K-means*. Se ha realizado un estudio para conocer cuál es el número de *clusters* óptimo con el cual se obtiene un mayor resultado tras aplicar la métrica de silueta. Al aplicarle un número de cuatro *clusters*, el resultado obtenido empeoraba con respecto a realizarlo con tres. Esta solución que se obtiene es de 0.4495, queriendo decir que las instancias separadas en distintos *cluster* se superponen sin tener una separación clara definida.

Tras este resultado negativo, se optó por realizar un estudio de *cluster* con un número de dos agrupaciones. Al realizar este cambio de agrupamiento al algoritmo se obtuvo un resultado de silueta de 0.5666. Esto supone una mejora considerable, pero el problema de realizar solamente dos agrupaciones a las instancias es que se estaría perdiendo mucha información. En este caso, se ha optado por tener más instancias que se superpongan a cambio de tener más divisiones entre los clientes.

A la hora de realizar este tipo de técnica de aprendizaje automático es obtener la máxima información posible sobre los distintos perfiles de los consumidores para poder adaptar de forma correcta las diversas estrategias de marketing realizadas por parte de la empresa. Por ello, se optó por tener un resultado un poco peor pero ganar en información respecto a ellos. Así, al tener más segmentado los consumidores, se puede aplicar una estrategia más personalizada y enfocada a ciertos nichos de clientes. Con el fin de ser más eficaces y obtener unos resultados mejores por parte de las campañas realizadas.

7. CONCLUSIONES Y PROPUESTAS DE FUTURO

Después de analizar el conjunto de datos, sus variables cuantitativas y cualitativas, así como la realización de dos técnicas de aprendizaje automático, de agrupación y de clasificación, se han llegado a las siguientes conclusiones. Por ello, se realizará una síntesis del trabajo de investigación dando respuesta a los objetivos marcados al inicio de este. Posteriormente, se presentarán posibles propuestas de futuro asentando una base sólida para continuar ampliando esta investigación.

Como primer objetivo se planteó un estudio de las variables para saber cómo influían a la hora de aceptar o no las ofertas realizadas por la marca. Tras el análisis, se puede concluir que si el consumidor realiza con más frecuencia compras por catálogo, por descuento o en web, tiende a aceptar menos las ofertas realizadas. En cambio, si el consumidor compra más en tiendas físicas, tiende a aceptarlas. Analizando los ingresos anuales familiares, se observa el mismo patrón que a la hora de comprar, cuanto más ingresos tienen, menos aceptan las ofertas. Este patrón se extiende también al gasto en ciertos productos como la carne, los dulces o el lujo, cuanto más gastan menos aceptan las ofertas. En cambio, para las frutas, el pescado y el vino, el consumidor acepta y rechaza las ofertas por igual independientemente del gasto que realice. Si se analizan las variables categóricas, no se observa una distinción entre las instancias ya que hay consumidores que rechazan y aceptan las ofertas por igual.

El segundo objetivo propuesto fue un estudio de las posibles segmentaciones de los consumidores. Para dar respuesta a este objetivo se realizó un algoritmo de *clustering* que agrupó las distintas variables del conjunto de datos en tres grupos distintos. Esto sirve para agrupar instancias parecidas y encontrar patrones entre las distintas características de los consumidores. Por un lado, se ha creado un *cluster* que representa a aquellos clientes que tienen unos ingresos y gastos medios, además se caracterizan por tener unos niveles de estudio de graduados y un estado civil de solteros.

El agrupamiento dos se caracteriza por reunir a los usuarios que tienen unos ingresos familiares anuales altos y que por tanto, también tienen unos gastos altos. Se caracterizan por ser clientes que están doctorados, es decir, poseen la mayor formación y que además se encuentran casados. Este

cluster cuenta con la distinción de los demás que es el grupo que más ofertas acepta. Y respecto al último grupo, se agrupan los clientes que tienen menores salarios y que menos gastos hacen. Se caracterizan por ser doctorados y por tener un estado civil de pareja de hecho. A diferencia de los otros grupos, estos usuarios son los que más tiempo pasan visitando la página *online* de la marca.

El tercer objetivo era analizar los canales de distribución para llevar a cabo campañas de marketing más eficaces. Junto con los agrupamientos anteriores, se puede saber qué tipo de canal usan más ciertos tipos de consumidores. Por ejemplo, aquellos que se han catalogado en el primer *cluster* son los que más utilizan la página *online* de la marca, mientras que el *cluster* dos el canal que más usa son los catálogos y el resto son utilizados por el último agrupamiento. Sabiendo esto, cuando se realice las estrategias de marketing correspondientes, se puede conocer a qué tipo de canal enfocarlo y dirigir las campañas directamente a esos.

Como último objetivo se planteó la realización de un algoritmo que fuese capaz de predecir si un cliente aceptaría o no la oferta realizada. Por ello, se ha realizado un modelo de clasificación de aprendizaje automático, un árbol de decisión. Tras crear las particiones correspondientes y el entrenamiento del modelo, se han obtenido unos resultados de predicciones del 0.86 aproximadamente. De este modo, las instancias del modelo que evaluemos serán clasificadas correctamente en un 86 %.

Todo este análisis sirve para poder conocer a los consumidores que adquieren los productos de la marca y sobre todo, que aceptan las campañas de marketing creadas por la empresa. Así, con los conocimientos obtenidos tras el análisis, se puede crear campañas nuevas de una manera más eficaz, centrando todo el desarrollo en un segmento u otro dependiendo de los objetivos marcados. Más aún, la posibilidad de predecir si el consumidor se verá afectado de manera positiva o negativa por estas campañas.

Como línea de estudio futuro, se puede continuar analizando estos datos aplicando otras técnicas de aprendizaje automático tanto de agrupamiento como de clasificación para buscar unos mejores resultados. También, se puede realizar todo el desarrollo de una estrategia de marketing con la

cual obtener resultados mejores que los obtenidos en la analizada, pues un alto número de clientes han rechazado la oferta propuesta. Tanto a modo de aprendizaje como de modo profesional, dando solución a nuevos retos y siendo capaces de aplicar nuevas técnicas para formarse en ellas.

8. BIBLIOGRAFÍA

Referencias

[com, 2021] (2021). Comportamiento del consumidor y el proceso de decisión de compra. *Ciencia latina*, 5(6):14216–14241.

[Acosta, 2017] Acosta, A. L. (2017). Canales de distribución.

[Ariza-López et al., 2018] Ariza-López, F. J., Rodríguez-Avi, J., and Alba-Fernández, V. (2018). Control estricto de matrices de confusión por medio de distribuciones multinomiale. *GeoFocus Revista Internacional de Ciencia y Tecnología de la Información Geográfica*, pages 215–226.

[Bekkar and Alitouche, 2013] Bekkar, M. and Alitouche, T. (2013). Imbalanced data learning approaches review. *International Journal of Data Mining Knowledge Management Process*, 3.

[Bekkar et al., 2013] Bekkar, M., Djema, H., and Alitouche, T. (2013). Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3:27–38.

[Berger et al., 2010] Berger, J., Sorensen, A. T., and Rasmussen, S. J. (2010). Positive effects of negative publicity: When negative reviews increase sales. *Mark. Sci.*, 29(5):815–827.

[Blythe, 2004] Blythe, J. (2004). *Essentials of Marketing*. Financial Times Prentice Hall.

[Borja-Robalino et al., 2020] Borja-Robalino, R., Monleon-Getino, A., and Benedé, J. (2020). Estandarización de métricas de rendimiento para clasificadores machine y deep learning.

[Bouza and Santiago, 2014] Bouza, C. and Santiago, A. (2014). *LA MINERÍA DE DATOS: ARBOLES DE DECISIÓN Y SU APLICACIÓN EN ESTUDIOS MÉDICOS*, pages 64–78.

[Breiman et al., 2017] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification And Regression Trees*.

[Brennan and Prediger, 1981] Brennan, R. L. and Prediger, D. J. (1981). Coefficient kappa: Some uses, misuses, and alternatives. *Educational and Psychological Measurement*, 41(3):687–699.

[Chawla et al., 2002] Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357.

[De Benito, 2014] De Benito, D. R.-R. (2014). Proceso de decisión del consumidor: factores explicativos del visionado de películas en sala de cine de los jóvenes universitarios españoles.

[De Matos and Veiga, 2004] De Matos, C. A. and Veiga, R. T. (2004). The Effects of Negative Publicity and Company Reaction on Consumer Attitudes. *Social Science Research Network*.

[Düntsch and Gediga, 2019] Düntsch, I. and Gediga, G. (2019). Confusion matrices and rough set data analysis. *Journal of Physics: Conference Series*, 1229(1):012055.

[Estarelles et al., 1992] Estarelles, R., De la Fuente, E. I., and Olmedo, P. (1992). Aplicación y valoración de diferentes algoritmos no-jerárquicos en el análisis cluster y su representación gráfica. *Anuario de Psicología*, 55:63–90.

[Ezugwu et al., 2022] Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., and Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743.

[Gaido, 2023] Gaido, M. (2023). Distributed silhouette algorithm: Evaluating clustering on big data.

[Hoo et al., 2017] Hoo, Z. H., Candlish, J., and Teare, D. (2017). What is an ROC curve? *Emergency Medicine Journal*, 34(6):357–359.

[Hossin and Sulaiman, 2015] Hossin, M. and Sulaiman, M. R. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining Knowledge Management Process*, 5(2):01–11.

[Humbría, 2010] Humbría, M. A. (2010). Proceso de decision de compra del cliente marabino ante la publicidad de las tarjetas de la banca universal. *COEPTUM*, 2(1):1–17.

[Kotler and Armstrong, 2012] Kotler, P. and Armstrong, G. M. (2012). *MARKETING 14ED*.

[Kotler and Keller, 2006] Kotler, P. and Keller, K. L. (2006). *Dirección de Marketing*. Pearson Educación.

[Lozano-Torres et al., 2021] Lozano-Torres, B. V., Toro-Espinoza, M. F., and Calderón-Argoti, D. J. (2021). El marketing digital: herramientas y tendencias actuales. *Dominio de las Ciencias*, 7(6):907–921.

[Maldonado et al., 2022] Maldonado, S., Vairetti, C., Fernández, A., and Herrera, F. (2022). FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification. *Pattern Recognition*, 124:108511.

[McKinney, 2012] McKinney, W. (2012). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*.

[Mulaomerović-Šeta et al., 2023] Mulaomerović-Šeta, A., Blagojevic, B., Mihailovic, V., and Petroselli, A. (2023). A silhouette-width-induced hierarchical clustering for defining flood estimation regions. *Hydrology*, 10:126.

[Nanjundan et al., 2019] Nanjundan, S., Sankaran, S., Arjun, C., and Anand, G. (2019). Identifying the number of clusters for k-means: A hypersphere density based approach.

[Nebreda, 1992] Nebreda, L. V. (1992). Analisis del proceso de decision del consumidor, para la estrategia comercial de la empresa. `https://dialnet.unirioja.es/descarga/articulo/786117.pdf`. Accessed: 2023-2-24.

[Park et al., 2004] Park, S. H., Goo, J. M., and Jo, C. H. (2004). Receiver Operating Characteristic (ROC) Curve: Practical Review for Radiologists. *Korean Journal of Radiology*, 5(1):11.

[Parr-Rud, 2014] Parr-Rud, O. (2014). *Business Analytics Using SAS Enterprise Guide and SAS Enterprise Miner: A Beginner's Guide*. SAS Institute.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *HAL (Le Centre pour la Communication Scientifique Directe)*.

[Prada Conde, 2022] Prada Conde, L. (2022). Aplicación de técnicas de clustering como paso previo a la detección de anomalías en redes defnidas por sofware.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[Ramón and Morán, 2014] Ramón, C. A. and Morán, J. E. P. (2014). *Procesos de venta, ciclo formativo de grado medio*.

[Refaeilzadeh et al., 2009] Refaeilzadeh, P., Tang, L., and Liu, H. (2009). *Cross-Validation*, pages 532–538. Springer US, Boston, MA.

[Rodríguez et al., 2016] Rodríguez, P. V., Duque, N., and Ovalle, D. A. (2016). Método Híbrido de Recomendación Adaptativa de Objetos de Aprendizaje basado en Perfiles de Usuario. *Formación universitaria*, 9(4):83–94.

[Thinsungnoen et al., 2015] Thinsungnoen, T., Kaoungku, N., Durongdumronchai, P., Kerdprasop, K., and Kerdprasop, N. (2015). The clustering validity with silhouette and sum of squared errors. pages 44–51.

[Ul Haq et al., 2019] Ul Haq, I., Gondal, I., Vamplew, P., and Brown, S. (2019). *Categorical Features Transformation with Compact One-Hot Encoder for Fraud Detection in Distributed Environment: 16th Australasian Conference, AusDM 2018, Bahrurst, NSW, Australia, November 28–30, 2018, Revised Selected Papers*, pages 69–80.

[Umargono et al., 2020] Umargono, E., Suseno, J., and Gunawan, S. (2020). K-means clustering optimization using the elbow method and early centroid determination based on mean and median formula.

[Vergara, 2022] Vergara, M. (2022). El comportamiento del consumidor post covid-19: Oportunidad o desafÍo para los emprendedores. pages 102–112.

[Zhang et al., 2010] Zhang, Y., Yan, Z.-J., and Soong, F. (2010). Cross-validation based decision tree clustering for hmm-based tts. pages 4602–4605.

A. ANEXO: CÓDIGO DEL TRATAMIENTO DE LA BASE DE DATOS

A continuación se incluye todo el código realizado para el análisis del conjunto de datos. Se puede visualizar el tratamiento de los datos, el estudio estadístico, las representaciones gráficas y las distintas técnicas de *machine learning* utilizadas.

CÓDIGO TFM

June 29, 2023

0.1 INSTALACIÓN DE PAQUETES

Instalación del paquete para poder realizar la técnica de smote ya que no viene instalada.

[1]:
```
pip install imbalanced-learn
```

```
Requirement already satisfied: imbalanced-learn in
c:\users\lucia\anaconda3\lib\site-packages (0.10.1)
Requirement already satisfied: scikit-learn>=1.0.2 in
c:\users\lucia\anaconda3\lib\site-packages (from imbalanced-learn) (1.0.2)
Requirement already satisfied: joblib>=1.1.1 in
c:\users\lucia\anaconda3\lib\site-packages (from imbalanced-learn) (1.2.0)
Requirement already satisfied: scipy>=1.3.2 in
c:\users\lucia\anaconda3\lib\site-packages (from imbalanced-learn) (1.9.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\users\lucia\anaconda3\lib\site-packages (from imbalanced-learn) (2.2.0)
Requirement already satisfied: numpy>=1.17.3 in
c:\users\lucia\anaconda3\lib\site-packages (from imbalanced-learn) (1.21.5)
Note: you may need to restart the kernel to use updated packages.
```

Instalación del paquete para poder realizar la gráfica del árbol de decisión ya que no viene instalada.

[2]:
```
pip install graphviz
```

```
Requirement already satisfied: graphviz in c:\users\lucia\anaconda3\lib\site-
packages (0.20.1)
Note: you may need to restart the kernel to use updated packages.
```

Importar las librerías con las que voy a trabajar

[3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from sklearn.model_selection import train_test_split
from imblearn import under_sampling, over_sampling
from imblearn.over_sampling import SMOTE
from sklearn import tree
from sklearn import metrics
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, roc_auc_score,␣
 ↪roc_curve
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn import datasets
from sklearn.tree import DecisionTreeClassifier
import graphviz
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import silhouette_score
```

0.2 LECTURA DE FICHEROS

[4]:
```
#Leer el fichero y ver los datos (Marketing Analytics) para saber con que datos␣
 ↪voy a trabajar. A comparación con la otra
#base de datos (todas parten de los mismos datos pero una selecciona unas␣
 ↪variables y otra otras) esta tiene menos variables
#pero tiene la de country.
datos = pd.read_csv("E:\\Master\\TFM\\marketing_data.csv", sep=",")
datos
```

[4]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	\
0	1826	1970	Graduation	Divorced	$84,835.00	0	
1	1	1961	Graduation	Single	$57,091.00	0	
2	10476	1958	Graduation	Married	$67,267.00	0	
3	1386	1967	Graduation	Together	$32,474.00	1	
4	5371	1989	Graduation	Single	$21,474.00	1	
…	…	…	…	…	…	…	
2235	10142	1976	PhD	Divorced	$66,476.00	0	
2236	5263	1977	2n Cycle	Married	$31,056.00	1	
2237	22	1976	Graduation	Divorced	$46,310.00	1	
2238	528	1978	Graduation	Married	$65,819.00	0	
2239	4070	1969	PhD	Married	$94,871.00	0	

	Teenhome	Dt_Customer	Recency	MntWines	…	NumStorePurchases	\
0	0	6/16/14	0	189	…	6	
1	0	6/15/14	0	464	…	7	
2	1	5/13/14	0	134	…	5	
3	1	5/11/14	0	10	…	2	
4	0	4/8/14	0	6	…	2	
…	…	…	…	…	…	…	
2235	1	3/7/13	99	372	…	11	
2236	0	1/22/13	99	5	…	3	
2237	0	12/3/12	99	185	…	5	
2238	0	11/29/12	99	267	…	10	
2239	2	9/1/12	99	169	…	4	

```
      NumWebVisitsMonth  AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  \
0                     1             0             0             0
1                     5             0             0             0
2                     2             0             0             0
3                     7             0             0             0
4                     7             1             0             0
…                   …             …             …             …
2235                  4             0             0             0
2236                  8             0             0             0
2237                  8             0             0             0
2238                  3             0             0             0
2239                  7             0             1             1

      AcceptedCmp1  AcceptedCmp2  Response  Complain  Country
0                0             0         1         0       SP
1                0             1         1         0       CA
2                0             0         0         0       US
3                0             0         0         0      AUS
4                0             0         1         0       SP
…              …             …       …       …       …
2235             0             0         0         0       US
2236             0             0         0         0       SP
2237             0             0         0         0       SP
2238             0             0         0         0      IND
2239             0             0         1         0       CA

[2240 rows x 28 columns]
```

[5]:
```
#Para conocer las columnas que hay en total en el conjunto de datos
nombres_columnas = datos.columns
print(nombres_columnas)
```

```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', ' Income ',
       'Kidhome', 'Teenhome', 'Dt_Customer', 'Recency', 'MntWines',
       'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Response', 'Complain', 'Country'],
      dtype='object')
```

[6]:
```
#Para saber el tipo de dato con el que se va a trabajar
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 28 columns):
 #   Column               Non-Null Count  Dtype
```

```
---  ------                --------------  -----
 0   ID                    2240 non-null   int64
 1   Year_Birth            2240 non-null   int64
 2   Education             2240 non-null   object
 3   Marital_Status        2240 non-null   object
 4    Income               2216 non-null   object
 5   Kidhome               2240 non-null   int64
 6   Teenhome              2240 non-null   int64
 7   Dt_Customer           2240 non-null   object
 8   Recency               2240 non-null   int64
 9   MntWines              2240 non-null   int64
 10  MntFruits             2240 non-null   int64
 11  MntMeatProducts       2240 non-null   int64
 12  MntFishProducts       2240 non-null   int64
 13  MntSweetProducts      2240 non-null   int64
 14  MntGoldProds          2240 non-null   int64
 15  NumDealsPurchases     2240 non-null   int64
 16  NumWebPurchases       2240 non-null   int64
 17  NumCatalogPurchases   2240 non-null   int64
 18  NumStorePurchases     2240 non-null   int64
 19  NumWebVisitsMonth     2240 non-null   int64
 20  AcceptedCmp3          2240 non-null   int64
 21  AcceptedCmp4          2240 non-null   int64
 22  AcceptedCmp5          2240 non-null   int64
 23  AcceptedCmp1          2240 non-null   int64
 24  AcceptedCmp2          2240 non-null   int64
 25  Response              2240 non-null   int64
 26  Complain              2240 non-null   int64
 27  Country               2240 non-null   object
dtypes: int64(23), object(5)
memory usage: 490.1+ KB
```

[7]:
```
#estos datos(marketing campaign) tienen dos columnas que la otra base de datos␣
 ↪no tiene, le falta la columna country.
datos1= pd.read_csv("E:\\Master\\TFM\\marketing_campaign.csv", sep="\t")
datos1
```

[7]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	\
0	5524	1957	Graduation	Single	58138.0	0	
1	2174	1954	Graduation	Single	46344.0	1	
2	4141	1965	Graduation	Together	71613.0	0	
3	6182	1984	Graduation	Together	26646.0	1	
4	5324	1981	PhD	Married	58293.0	1	
…	…	…	…	…	…	…	
2235	10870	1967	Graduation	Married	61223.0	0	
2236	4001	1946	PhD	Together	64014.0	2	
2237	7270	1981	Graduation	Divorced	56981.0	0	

```
2238   8235        1956      Master       Together  69245.0        0
2239   9405        1954         PhD        Married  52869.0        1

      Teenhome Dt_Customer  Recency  MntWines  …  NumWebVisitsMonth  \
0            0  04-09-2012       58       635  …                  7
1            1  08-03-2014       38        11  …                  5
2            0  21-08-2013       26       426  …                  4
3            0  10-02-2014       26        11  …                  6
4            0  19-01-2014       94       173  …                  5
…          …         …      …        …  …                 …
2235         1  13-06-2013       46       709  …                  5
2236         1  10-06-2014       56       406  …                  7
2237         0  25-01-2014       91       908  …                  6
2238         1  24-01-2014        8       428  …                  3
2239         1  15-10-2012       40        84  …                  7

      AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  \
0                0             0             0             0             0
1                0             0             0             0             0
2                0             0             0             0             0
3                0             0             0             0             0
4                0             0             0             0             0
…              …           …           …           …           …
2235             0             0             0             0             0
2236             0             0             0             1             0
2237             0             1             0             0             0
2238             0             0             0             0             0
2239             0             0             0             0             0

      Complain  Z_CostContact  Z_Revenue  Response
0            0              3         11         1
1            0              3         11         0
2            0              3         11         0
3            0              3         11         0
4            0              3         11         0
…          …            …        …       …
2235         0              3         11         0
2236         0              3         11         0
2237         0              3         11         0
2238         0              3         11         0
2239         0              3         11         1

[2240 rows x 29 columns]
```

[8]:
```
#Para comprobar el tipo de dato
datos1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

0.3 TRATAMIENTO DE LA BASE DE DATOS

UNIFICACIÓN DE AMBAS BASES

[9]:
```
# Antes de tratar los datos, se ha realizado una unificación de bases para que la que se va a utilzar tenga todas las columnas
# Combinación de las bases de datos en función de la columna ID, que es la clave primaria qu los une y se ha añadido al final
combinacion= pd.merge(datos1, datos[['ID', 'Country']], on='ID', how='left')

# Se copia la columna Country de la base de datos original a la nueva base de datos
```

```
datos1['Country'] = combinacion['Country']

# Guardado de la nueva base de datos en un archivo CSV
datos1.to_csv('nueva.csv', index=False)
```

[10]:
```
# Para comprobar que todo se ha guardado correctamente y que están bien los
↪datos que se van a utilizar
combinacion
```

[10]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	\
0	5524	1957	Graduation	Single	58138.0	0	
1	2174	1954	Graduation	Single	46344.0	1	
2	4141	1965	Graduation	Together	71613.0	0	
3	6182	1984	Graduation	Together	26646.0	1	
4	5324	1981	PhD	Married	58293.0	1	
…	…	…	…	…	…	…	
2235	10870	1967	Graduation	Married	61223.0	0	
2236	4001	1946	PhD	Together	64014.0	2	
2237	7270	1981	Graduation	Divorced	56981.0	0	
2238	8235	1956	Master	Together	69245.0	0	
2239	9405	1954	PhD	Married	52869.0	1	

	Teenhome	Dt_Customer	Recency	MntWines	…	AcceptedCmp3	\
0	0	04-09-2012	58	635	…	0	
1	1	08-03-2014	38	11	…	0	
2	0	21-08-2013	26	426	…	0	
3	0	10-02-2014	26	11	…	0	
4	0	19-01-2014	94	173	…	0	
…	…	…	…	…	…	…	
2235	1	13-06-2013	46	709	…	0	
2236	1	10-06-2014	56	406	…	0	
2237	0	25-01-2014	91	908	…	0	
2238	1	24-01-2014	8	428	…	0	
2239	1	15-10-2012	40	84	…	0	

	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	Complain	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	
…	…	…	…	…	…	
2235	0	0	0	0	0	
2236	0	0	1	0	0	
2237	1	0	0	0	0	
2238	0	0	0	0	0	
2239	0	0	0	0	0	

```
      Z_CostContact  Z_Revenue  Response  Country
0                 3         11         1       US
1                 3         11         0       US
2                 3         11         0       US
3                 3         11         0       US
4                 3         11         0       US
…               …         …        …       …
2235              3         11         0       SA
2236              3         11         0       SA
2237              3         11         0       SA
2238              3         11         0       SA
2239              3         11         1       SA

[2240 rows x 30 columns]
```

[11]:
```
#Visualizar la base de datos combinada con la cual vamos a trabajar
datos2= pd.read_csv("E:\\Master\\TFM\\nueva.csv", sep=",")
datos2
```

[11]:
```
          ID  Year_Birth   Education Marital_Status   Income  Kidhome  \
0      5524        1957  Graduation         Single  58138.0        0
1      2174        1954  Graduation         Single  46344.0        1
2      4141        1965  Graduation       Together  71613.0        0
3      6182        1984  Graduation       Together  26646.0        1
4      5324        1981         PhD        Married  58293.0        1
…     …         …          …             …      …      …
2235  10870        1967  Graduation        Married  61223.0        0
2236   4001        1946         PhD       Together  64014.0        2
2237   7270        1981  Graduation       Divorced  56981.0        0
2238   8235        1956      Master       Together  69245.0        0
2239   9405        1954         PhD        Married  52869.0        1

      Teenhome Dt_Customer  Recency  MntWines  …  AcceptedCmp3  \
0            0  04-09-2012       58       635  …             0
1            1  08-03-2014       38        11  …             0
2            0  21-08-2013       26       426  …             0
3            0  10-02-2014       26        11  …             0
4            0  19-01-2014       94       173  …             0
…        …          …      …        …  …            …
2235         1  13-06-2013       46       709  …             0
2236         1  10-06-2014       56       406  …             0
2237         0  25-01-2014       91       908  …             0
2238         1  24-01-2014        8       428  …             0
2239         1  15-10-2012       40        84  …             0

      AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
```

```
0                0              0              0              0          0
1                0              0              0              0          0
2                0              0              0              0          0
3                0              0              0              0          0
4                0              0              0              0          0
…              …              …              …              …          …
2235             0              0              0              0          0
2236             0              0              1              0          0
2237             1              0              0              0          0
2238             0              0              0              0          0
2239             0              0              0              0          0

      Z_CostContact  Z_Revenue  Response  Country
0                 3         11         1       US
1                 3         11         0       US
2                 3         11         0       US
3                 3         11         0       US
4                 3         11         0       US
…               …         …          …       …
2235              3         11         0       SA
2236              3         11         0       SA
2237              3         11         0       SA
2238              3         11         0       SA
2239              3         11         1       SA

[2240 rows x 30 columns]
```

[12]:
```
#Para saber la cantidad de datos y variables que tiene la base de datos
datos2.shape
```

[12]:
```
(2240, 30)
```

[13]:
```
# Comprobación de nuevo de las columnas para saber que se ha unificado bien
nombres_columnas = datos2.columns
print(nombres_columnas)
```

```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response',
       'Country'],
      dtype='object')
```

[14]:
```
datos2.info()
#Son todos números enteros menos los ingresos que son numeros decimales
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 30 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Dt_Customer          2240 non-null   object
 8   Recency              2240 non-null   int64
 9   MntWines             2240 non-null   int64
 10  MntFruits            2240 non-null   int64
 11  MntMeatProducts      2240 non-null   int64
 12  MntFishProducts      2240 non-null   int64
 13  MntSweetProducts     2240 non-null   int64
 14  MntGoldProds         2240 non-null   int64
 15  NumDealsPurchases    2240 non-null   int64
 16  NumWebPurchases      2240 non-null   int64
 17  NumCatalogPurchases  2240 non-null   int64
 18  NumStorePurchases    2240 non-null   int64
 19  NumWebVisitsMonth    2240 non-null   int64
 20  AcceptedCmp3         2240 non-null   int64
 21  AcceptedCmp4         2240 non-null   int64
 22  AcceptedCmp5         2240 non-null   int64
 23  AcceptedCmp1         2240 non-null   int64
 24  AcceptedCmp2         2240 non-null   int64
 25  Complain             2240 non-null   int64
 26  Z_CostContact        2240 non-null   int64
 27  Z_Revenue            2240 non-null   int64
 28  Response             2240 non-null   int64
 29  Country              2240 non-null   object
dtypes: float64(1), int64(25), object(4)
memory usage: 525.1+ KB
```

LIMPIEZA ESTADO CIVIL

[15]:
```
#Cambio de los nombres de las variables del inglés al español, ya que toda la␣
 ↪investigación va a ser en español
varespañol={'ID': 'Identificador', 'Year_Birth': 'Cumpleaños', 'Education':␣
 ↪'Educacion', 'Marital_Status':'Estadocivil', 'Income': 'Ingresos', 'Kidhome':
 ↪'Niños',
       'Teenhome': 'Adolescentes', 'Dt_Customer': 'Inscripcion', 'Recency':
 ↪'Ultimacompra', 'MntWines': 'Totalvinos', 'MntFruits': 'Totalfrutas',
```

```
      'MntMeatProducts': 'Totalcarnes', 'MntFishProducts':'Totalpescados',␣
 ↪'MntSweetProducts':'Totaldulces',
      'MntGoldProds':'Totallujos', 'NumDealsPurchases':'Comprasdescuentos',␣
 ↪'NumWebPurchases':'Comprasweb',
      'NumCatalogPurchases':'Comprascatalogo', 'NumStorePurchases':
 ↪'Comprastiendas', 'NumWebVisitsMonth':'Visitaswebmes',
      'AcceptedCmp3':'Campaña3', 'AcceptedCmp4':'Campaña4', 'AcceptedCmp5':
 ↪'Campaña5', 'AcceptedCmp1':'Campaña1',
      'AcceptedCmp2':'Campaña2', 'Complain':'Quejas', 'Z_CostContact':
 ↪'Zcostecontacto', 'Z_Revenue':'Zingresos', 'Response':'Respuesta',
      'Country':'Pais'}
datos2= datos2.rename(columns=varespañol)
```

[16]:
```
#Para ver si se han cambiado bien las variables
datos2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 30 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Identificador      2240 non-null   int64
 1   Cumpleaños         2240 non-null   int64
 2   Educacion          2240 non-null   object
 3   Estadocivil        2240 non-null   object
 4   Ingresos           2216 non-null   float64
 5   Niños              2240 non-null   int64
 6   Adolescentes       2240 non-null   int64
 7   Inscripcion        2240 non-null   object
 8   Ultimacompra       2240 non-null   int64
 9   Totalvinos         2240 non-null   int64
 10  Totalfrutas        2240 non-null   int64
 11  Totalcarnes        2240 non-null   int64
 12  Totalpescados      2240 non-null   int64
 13  Totaldulces        2240 non-null   int64
 14  Totallujos         2240 non-null   int64
 15  Comprasdescuentos  2240 non-null   int64
 16  Comprasweb         2240 non-null   int64
 17  Comprascatalogo    2240 non-null   int64
 18  Comprastiendas     2240 non-null   int64
 19  Visitaswebmes      2240 non-null   int64
 20  Campaña3           2240 non-null   int64
 21  Campaña4           2240 non-null   int64
 22  Campaña5           2240 non-null   int64
 23  Campaña1           2240 non-null   int64
 24  Campaña2           2240 non-null   int64
 25  Quejas             2240 non-null   int64
```

```
 26  Zcostecontacto     2240 non-null   int64
 27  Zingresos          2240 non-null   int64
 28  Respuesta          2240 non-null   int64
 29  Pais               2240 non-null   object
dtypes: float64(1), int64(25), object(4)
memory usage: 525.1+ KB
```

[17]:
```
#Comprobación de las variables categóricas (Estado civil) para saber si las␣
 ↪respuestas son correctas o hay algún error
#tengo que cambiar en la columna de estado civil el alone por single.
datos2.groupby("Estadocivil").describe().index
```

[17]:
```
Index(['Absurd', 'Alone', 'Divorced', 'Married', 'Single', 'Together', 'Widow',
       'YOLO'],
      dtype='object', name='Estadocivil')
```

[18]:
```
#Cambio de respuestas que significan lo mismo para unificarlas en una
datos3= datos2.replace(['Absurd','Alone','YOLO'], 'Single')
```

[19]:
```
#Para cambiar el nombre de las respuestas categóricas de inglés al español

datos3['Estadocivil'] = datos3['Estadocivil'].replace('Divorced', 'Divorciado')
datos3['Estadocivil'] = datos3['Estadocivil'].replace('Single', 'Soltero')
datos3['Estadocivil'] = datos3['Estadocivil'].replace('Married', 'Casado')
datos3['Estadocivil'] = datos3['Estadocivil'].replace('Widow', 'Viudo')
datos3['Estadocivil'] = datos3['Estadocivil'].replace('Together',␣
 ↪'Parejadehecho')
```

[20]:
```
#Comprobación de que no queda ninguna instancia mal o en inglés
datos3.groupby("Estadocivil").describe().index
```

[20]:
```
Index(['Casado', 'Divorciado', 'Parejadehecho', 'Soltero', 'Viudo'],
dtype='object', name='Estadocivil')
```

[21]:
```
# Comprobraciñon general de los datos
datos3
```

[21]:

	Identificador	Cumpleaños	Educacion	Estadocivil	Ingresos	Niños	\
0	5524	1957	Graduation	Soltero	58138.0	0	
1	2174	1954	Graduation	Soltero	46344.0	1	
2	4141	1965	Graduation	Parejadehecho	71613.0	0	
3	6182	1984	Graduation	Parejadehecho	26646.0	1	
4	5324	1981	PhD	Casado	58293.0	1	
…	…	…	…	…	…	…	
2235	10870	1967	Graduation	Casado	61223.0	0	
2236	4001	1946	PhD	Parejadehecho	64014.0	2	
2237	7270	1981	Graduation	Divorciado	56981.0	0	
2238	8235	1956	Master	Parejadehecho	69245.0	0	

```
2239           9405        1954          PhD          Casado   52869.0       1

      Adolescentes Inscripcion  Ultimacompra  Totalvinos  …  Campaña3  \
0                0  04-09-2012            58         635  …         0
1                1  08-03-2014            38          11  …         0
2                0  21-08-2013            26         426  …         0
3                0  10-02-2014            26          11  …         0
4                0  19-01-2014            94         173  …         0
…              …         …             …          …  …       …
2235             1  13-06-2013            46         709  …         0
2236             1  10-06-2014            56         406  …         0
2237             0  25-01-2014            91         908  …         0
2238             1  24-01-2014             8         428  …         0
2239             1  15-10-2012            40          84  …         0

      Campaña4  Campaña5  Campaña1  Campaña2  Quejas  Zcostecontacto  \
0            0         0         0         0       0               3
1            0         0         0         0       0               3
2            0         0         0         0       0               3
3            0         0         0         0       0               3
4            0         0         0         0       0               3
…          …       …       …         …     …               …
2235         0         0         0         0       0               3
2236         0         0         1         0       0               3
2237         1         0         0         0       0               3
2238         0         0         0         0       0               3
2239         0         0         0         0       0               3

      Zingresos  Respuesta  Pais
0            11          1    US
1            11          0    US
2            11          0    US
3            11          0    US
4            11          0    US
…          …         …    …
2235         11          0    SA
2236         11          0    SA
2237         11          0    SA
2238         11          0    SA
2239         11          1    SA

[2240 rows x 30 columns]
```

LIMPIEZA EDUCACIÓN

[22]:
```
# Comprobación de que los datos categóricos de la variable eduación son␣
↪correctos
datos3.groupby("Educacion").describe().index
```

[22]:
```
Index(['2n Cycle', 'Basic', 'Graduation', 'Master', 'PhD'], dtype='object',
name='Educacion')
```

[23]:
```
# Se reemplaza el 2 ciclo por master ya que significan lo mismo
datos4= datos3.replace(['2n Cycle'], 'Master')
```

[24]:
```
# Descripción de la variable educación para saber que se ha cambiado␣
 ↪correctamente la instancia
datos4.groupby("Educacion").describe().index
```

[24]:
```
Index(['Basic', 'Graduation', 'Master', 'PhD'], dtype='object',
name='Educacion')
```

[25]:
```
#Para cambiar el nombre de las respuestas categóricas de inglés al español

datos4['Educacion'] = datos4['Educacion'].replace('Basic', 'Basico')
datos4['Educacion'] = datos4['Educacion'].replace('Graduation', 'Graduado')
```

[26]:
```
# Comprobación general de los datos para saber que están bien
datos4
```

[26]:

	Identificador	Cumpleaños	Educacion	Estadocivil	Ingresos	Niños	\
0	5524	1957	Graduado	Soltero	58138.0	0	
1	2174	1954	Graduado	Soltero	46344.0	1	
2	4141	1965	Graduado	Parejadehecho	71613.0	0	
3	6182	1984	Graduado	Parejadehecho	26646.0	1	
4	5324	1981	PhD	Casado	58293.0	1	
…	…	…	…	…	…	…	
2235	10870	1967	Graduado	Casado	61223.0	0	
2236	4001	1946	PhD	Parejadehecho	64014.0	2	
2237	7270	1981	Graduado	Divorciado	56981.0	0	
2238	8235	1956	Master	Parejadehecho	69245.0	0	
2239	9405	1954	PhD	Casado	52869.0	1	

	Adolescentes	Inscripcion	Ultimacompra	Totalvinos	…	Campaña3	\
0	0	04-09-2012	58	635	…	0	
1	1	08-03-2014	38	11	…	0	
2	0	21-08-2013	26	426	…	0	
3	0	10-02-2014	26	11	…	0	
4	0	19-01-2014	94	173	…	0	
…	…	…	…	…	…	…	
2235	1	13-06-2013	46	709	…	0	
2236	1	10-06-2014	56	406	…	0	
2237	0	25-01-2014	91	908	…	0	
2238	1	24-01-2014	8	428	…	0	
2239	1	15-10-2012	40	84	…	0	

```
      Campaña4  Campaña5  Campaña1  Campaña2  Quejas  Zcostecontacto  \
0            0         0         0         0       0               3
1            0         0         0         0       0               3
2            0         0         0         0       0               3
3            0         0         0         0       0               3
4            0         0         0         0       0               3
…          …         …         …         …     …               …
2235         0         0         0         0       0               3
2236         0         0         1         0       0               3
2237         1         0         0         0       0               3
2238         0         0         0         0       0               3
2239         0         0         0         0       0               3

      Zingresos  Respuesta  Pais
0            11          1    US
1            11          0    US
2            11          0    US
3            11          0    US
4            11          0    US
…          …         …    …
2235         11          0    SA
2236         11          0    SA
2237         11          0    SA
2238         11          0    SA
2239         11          1    SA

[2240 rows x 30 columns]
```

LIMPIEZA DE NULOS

[27]:
```
#para saber en qué columna hay valores nulos
datos4.isnull().any()
```

[27]:
```
Identificador        False
Cumpleaños           False
Educacion            False
Estadocivil          False
Ingresos              True
Niños                False
Adolescentes         False
Inscripcion          False
Ultimacompra         False
Totalvinos           False
Totalfrutas          False
Totalcarnes          False
Totalpescados        False
Totaldulces          False
Totallujos           False
```

```
Comprasdescuentos    False
Comprasweb           False
Comprascatalogo      False
Comprastiendas       False
Visitaswebmes        False
Campaña3             False
Campaña4             False
Campaña5             False
Campaña1             False
Campaña2             False
Quejas               False
Zcostecontacto       False
Zingresos            False
Respuesta            False
Pais                 False
dtype: bool
```

[28]:
```
#Para saber el número total de datos nulos que hay por columna
datos4.isnull().sum()
```

[28]:
```
Identificador         0
Cumpleaños            0
Educacion             0
Estadocivil           0
Ingresos             24
Niños                 0
Adolescentes          0
Inscripcion           0
Ultimacompra          0
Totalvinos            0
Totalfrutas           0
Totalcarnes           0
Totalpescados         0
Totaldulces           0
Totallujos            0
Comprasdescuentos     0
Comprasweb            0
Comprascatalogo       0
Comprastiendas        0
Visitaswebmes         0
Campaña3              0
Campaña4              0
Campaña5              0
Campaña1              0
Campaña2              0
Quejas                0
Zcostecontacto        0
```

```
Zingresos             0
Respuesta             0
Pais                  0
dtype: int64
```

[29]:
```
#Para saber la media de los valores de cada columna
datos4.mean(axis=0)
```

```
C:\Users\Lucia\AppData\Local\Temp\ipykernel_12016\2816139399.py:2:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  datos4.mean(axis=0)
```

[29]:
```
Identificador         5592.159821
Cumpleaños            1968.805804
Ingresos             52247.251354
Niños                    0.444196
Adolescentes             0.506250
Ultimacompra            49.109375
Totalvinos             303.935714
Totalfrutas             26.302232
Totalcarnes            166.950000
Totalpescados           37.525446
Totaldulces             27.062946
Totallujos              44.021875
Comprasdescuentos        2.325000
Comprasweb               4.084821
Comprascatalogo          2.662054
Comprastiendas           5.790179
Visitaswebmes            5.316518
Campaña3                 0.072768
Campaña4                 0.074554
Campaña5                 0.072768
Campaña1                 0.064286
Campaña2                 0.013393
Quejas                   0.009375
Zcostecontacto           3.000000
Zingresos               11.000000
Respuesta                0.149107
dtype: float64
```

[30]:
```
#Para rellenar los datos nulos con la media de la columna ingresos.
media= datos4['Ingresos'].mean()
datos4['Ingresos'] = datos4['Ingresos'].fillna(media)
```

[31]:
```
#Volvemos a comprobar que no hay ningún valor nulo.
datos4.isnull().sum()
```

[31]:
```
Identificador        0
Cumpleaños           0
Educacion            0
Estadocivil          0
Ingresos             0
Niños                0
Adolescentes         0
Inscripcion          0
Ultimacompra         0
Totalvinos           0
Totalfrutas          0
Totalcarnes          0
Totalpescados        0
Totaldulces          0
Totallujos           0
Comprasdescuentos    0
Comprasweb           0
Comprascatalogo      0
Comprastiendas       0
Visitaswebmes        0
Campaña3             0
Campaña4             0
Campaña5             0
Campaña1             0
Campaña2             0
Quejas               0
Zcostecontacto       0
Zingresos            0
Respuesta            0
Pais                 0
dtype: int64
```

0.3.1 COMPROBACIÓN DE ERRORES

[32]:
```
#Para sacar los estadísticos
datos4.describe()
```

[32]:

	Identificador	Cumpleaños	Ingresos	Niños	Adolescentes	\
count	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	
mean	5592.159821	1968.805804	52247.251354	0.444196	0.506250	
std	3246.662198	11.984069	25037.797168	0.538398	0.544538	
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	
25%	2828.250000	1959.000000	35538.750000	0.000000	0.000000	
50%	5458.500000	1970.000000	51741.500000	0.000000	0.000000	
75%	8427.750000	1977.000000	68289.750000	1.000000	1.000000	
max	11191.000000	1996.000000	666666.000000	2.000000	2.000000	

```
       Ultimacompra   Totalvinos  Totalfrutas  Totalcarnes  Totalpescados  \
count   2240.000000  2240.000000  2240.000000  2240.000000    2240.000000
mean      49.109375   303.935714    26.302232   166.950000      37.525446
std       28.962453   336.597393    39.773434   225.715373      54.628979
min        0.000000     0.000000     0.000000     0.000000       0.000000
25%       24.000000    23.750000     1.000000    16.000000       3.000000
50%       49.000000   173.500000     8.000000    67.000000      12.000000
75%       74.000000   504.250000    33.000000   232.000000      50.000000
max       99.000000  1493.000000   199.000000  1725.000000     259.000000

       …  Visitaswebmes     Campaña3     Campaña4     Campaña5     Campaña1  \
count  …    2240.000000  2240.000000  2240.000000  2240.000000  2240.000000
mean   …       5.316518     0.072768     0.074554     0.072768     0.064286
std    …       2.426645     0.259813     0.262728     0.259813     0.245316
min    …       0.000000     0.000000     0.000000     0.000000     0.000000
25%    …       3.000000     0.000000     0.000000     0.000000     0.000000
50%    …       6.000000     0.000000     0.000000     0.000000     0.000000
75%    …       7.000000     0.000000     0.000000     0.000000     0.000000
max    …      20.000000     1.000000     1.000000     1.000000     1.000000

          Campaña2       Quejas  Zcostecontacto  Zingresos    Respuesta
count  2240.000000  2240.000000          2240.0     2240.0  2240.000000
mean      0.013393     0.009375             3.0       11.0     0.149107
std       0.114976     0.096391             0.0        0.0     0.356274
min       0.000000     0.000000             3.0       11.0     0.000000
25%       0.000000     0.000000             3.0       11.0     0.000000
50%       0.000000     0.000000             3.0       11.0     0.000000
75%       0.000000     0.000000             3.0       11.0     0.000000
max       1.000000     1.000000             3.0       11.0     1.000000

[8 rows x 26 columns]
```

LIMPIEZA DE LA VARIABLE CUMPLEAÑOS

[33]:
```
#Comprobar los valores unicos ya que se ha comprobado con el describe que hay␣
↪un valor extraño
datos4['Cumpleaños'].unique()
```

[33]:
```
array([1957, 1954, 1965, 1984, 1981, 1967, 1971, 1985, 1974, 1950, 1983,
       1976, 1959, 1952, 1987, 1946, 1980, 1949, 1982, 1979, 1951, 1969,
       1986, 1989, 1963, 1970, 1973, 1943, 1975, 1996, 1968, 1964, 1977,
       1978, 1955, 1966, 1988, 1948, 1958, 1972, 1960, 1945, 1991, 1962,
       1953, 1961, 1956, 1992, 1900, 1893, 1990, 1947, 1899, 1993, 1994,
       1941, 1944, 1995, 1940], dtype=int64)
```

[34]:
```
#Como hay 3 valores en años que están mal ya que no sigue la media del resto,␣
↪se intuye que o son anormales o se han introducido
```

```
#incorrectamente a la hora de rellenar los datos. Por ello, se van a reemplazar␣
 ↪por la media de los datos en vez de eliminarlos.
datos4['Cumpleaños'] = datos4['Cumpleaños'].replace(1893, 1969)
datos4['Cumpleaños'] = datos4['Cumpleaños'].replace(1900, 1969)
datos4['Cumpleaños'] = datos4['Cumpleaños'].replace(1899, 1969)
```

[35]:
```
#Comprobar que los datos se han limpiado correctamente
datos4['Cumpleaños'].unique()
```

[35]:
```
array([1957, 1954, 1965, 1984, 1981, 1967, 1971, 1985, 1974, 1950, 1983,
       1976, 1959, 1952, 1987, 1946, 1980, 1949, 1982, 1979, 1951, 1969,
       1986, 1989, 1963, 1970, 1973, 1943, 1975, 1996, 1968, 1964, 1977,
       1978, 1955, 1966, 1988, 1948, 1958, 1972, 1960, 1945, 1991, 1962,
       1953, 1961, 1956, 1992, 1990, 1947, 1993, 1994, 1941, 1944, 1995,
       1940], dtype=int64)
```

LIMPIEZA DE LA VARIABLE INGRESOS

[36]:
```
#Para ver cual es el mayor dato que se habia comprobado previamente con el␣
 ↪describe
datos4['Ingresos'].max()
```

[36]:
```
666666.0
```

[37]:
```
#Como hay un valor en los ingresos que no sigue la media de los demas: un␣
 ↪ingreso de 666.666€, se intuye que es anormal o que
#se ha introducido incorrectamente a la hora de rellenar los datos. Por ello,␣
 ↪se va a reemplazar por la media de los datos
#en vez de eliminarlo
datos4['Ingresos']= datos4['Ingresos'].replace(666666.0, 52247.251354)
```

[38]:
```
#comprobar que se ha limpiado bien observando el numero máximo
datos4['Ingresos'].max()
```

[38]:
```
162397.0
```

[39]:
```
#Comprobar que no hay más datos erróneos
datos4['Ingresos'].unique()
```

[39]:
```
array([58138., 46344., 71613., …, 56981., 69245., 52869.])
```

0.4 GUARDADO DE ARCHIVOS

[40]:
```
#Guardado del csv de la base de datos tratada y con la cual se va a trabajar
datos4.to_csv('ARCHIVOTRATADO.csv', sep=';')
```

[41]:
```
#Guardado de la base de datos sin tratar por si más adelante se necesita
datos.to_csv('ARCHIVO2.csv', sep=';')
```

0.5 ANÁLISIS ESTADÍSTICO VARIABLES CUANTITATIVAS

[42]:
```
#Descripción estadistica traspuesta para pasar al documento
datos4.describe().T
```

[42]:

	count	mean	std	min	25%	\
Identificador	2240.0	5592.159821	3246.662198	0.0	2828.25	
Cumpleaños	2240.0	1968.901786	11.694076	1940.0	1959.00	
Ingresos	2240.0	51972.957270	21405.824379	1730.0	35538.75	
Niños	2240.0	0.444196	0.538398	0.0	0.00	
Adolescentes	2240.0	0.506250	0.544538	0.0	0.00	
Ultimacompra	2240.0	49.109375	28.962453	0.0	24.00	
Totalvinos	2240.0	303.935714	336.597393	0.0	23.75	
Totalfrutas	2240.0	26.302232	39.773434	0.0	1.00	
Totalcarnes	2240.0	166.950000	225.715373	0.0	16.00	
Totalpescados	2240.0	37.525446	54.628979	0.0	3.00	
Totaldulces	2240.0	27.062946	41.280498	0.0	1.00	
Totallujos	2240.0	44.021875	52.167439	0.0	9.00	
Comprasdescuentos	2240.0	2.325000	1.932238	0.0	1.00	
Comprasweb	2240.0	4.084821	2.778714	0.0	2.00	
Comprascatalogo	2240.0	2.662054	2.923101	0.0	0.00	
Comprastiendas	2240.0	5.790179	3.250958	0.0	3.00	
Visitaswebmes	2240.0	5.316518	2.426645	0.0	3.00	
Campaña3	2240.0	0.072768	0.259813	0.0	0.00	
Campaña4	2240.0	0.074554	0.262728	0.0	0.00	
Campaña5	2240.0	0.072768	0.259813	0.0	0.00	
Campaña1	2240.0	0.064286	0.245316	0.0	0.00	
Campaña2	2240.0	0.013393	0.114976	0.0	0.00	
Quejas	2240.0	0.009375	0.096391	0.0	0.00	
Zcostecontacto	2240.0	3.000000	0.000000	3.0	3.00	
Zingresos	2240.0	11.000000	0.000000	11.0	11.00	
Respuesta	2240.0	0.149107	0.356274	0.0	0.00	

	50%	75%	max
Identificador	5458.5	8427.75	11191.0
Cumpleaños	1970.0	1977.00	1996.0
Ingresos	51741.5	68275.75	162397.0
Niños	0.0	1.00	2.0
Adolescentes	0.0	1.00	2.0
Ultimacompra	49.0	74.00	99.0
Totalvinos	173.5	504.25	1493.0
Totalfrutas	8.0	33.00	199.0
Totalcarnes	67.0	232.00	1725.0
Totalpescados	12.0	50.00	259.0
Totaldulces	8.0	33.00	263.0
Totallujos	24.0	56.00	362.0
Comprasdescuentos	2.0	3.00	15.0
Comprasweb	4.0	6.00	27.0

```
Comprascatalogo        2.0      4.00      28.0
Comprastiendas         5.0      8.00      13.0
Visitaswebmes          6.0      7.00      20.0
Campaña3               0.0      0.00       1.0
Campaña4               0.0      0.00       1.0
Campaña5               0.0      0.00       1.0
Campaña1               0.0      0.00       1.0
Campaña2               0.0      0.00       1.0
Quejas                 0.0      0.00       1.0
Zcostecontacto         3.0      3.00       3.0
Zingresos             11.0     11.00      11.0
Respuesta              0.0      0.00       1.0
```

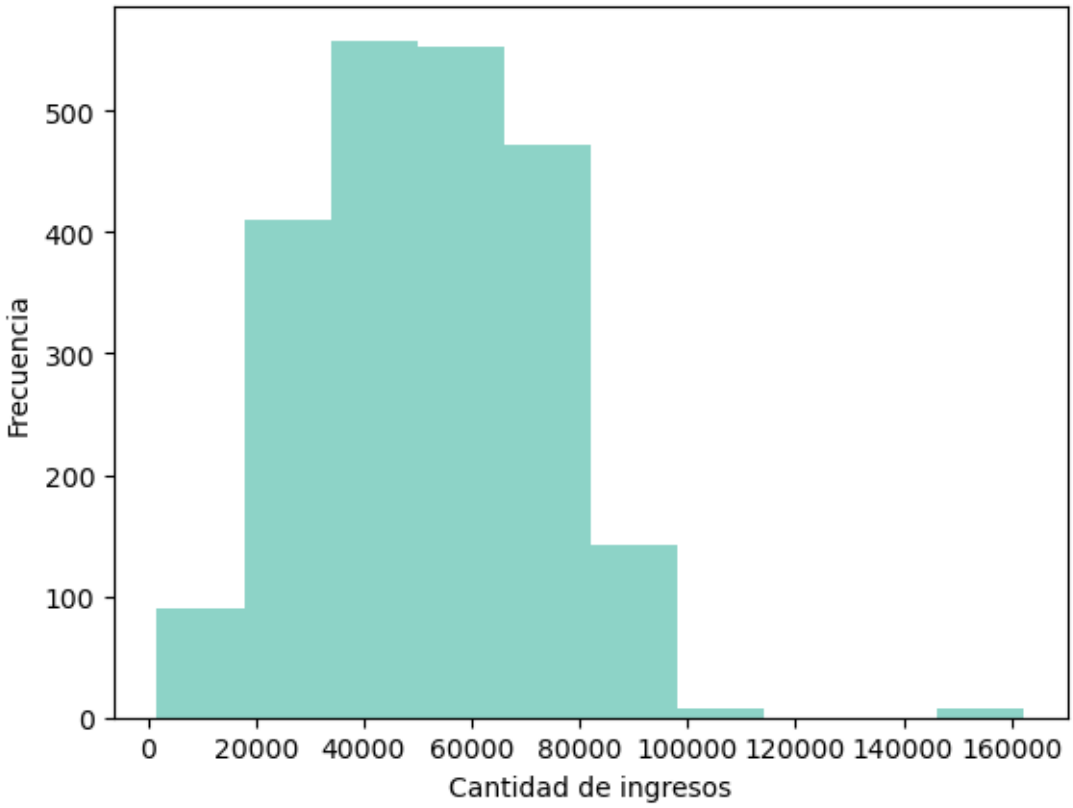
COMPROBACIÓN DE INGRESOS

[43]:
```
#Como se ha limpiado antes esta variable, se ordena de mayor a menor para␣
 ↪acabar de comprobar
datos4.sort_values(by=['Ingresos'], ascending=False)
```

[43]:
```
      Identificador  Cumpleaños Educacion    Estadocivil  Ingresos  Niños  \
617            1503        1976       PhD  Parejadehecho  162397.0      1
687            1501        1982       PhD         Casado  160803.0      0
1300           5336        1971    Master  Parejadehecho  157733.0      1
164            8475        1973       PhD         Casado  157243.0      0
1653           4931        1977  Graduado  Parejadehecho  157146.0      0
…               …           …         …             …         …      …
1975          10311        1969  Graduado         Casado    4428.0      0
1846           9931        1963       PhD         Casado    4023.0      1
1524          11110        1973  Graduado        Soltero    3502.0      1
21             5376        1979  Graduado         Casado    2447.0      1
1245           6862        1971  Graduado     Divorciado    1730.0      0

      Adolescentes Inscripcion  Ultimacompra  Totalvinos  …  Campaña3  \
617              1  03-06-2013            31          85  …         0
687              0  04-08-2012            21          55  …         0
1300             0  04-06-2013            37          39  …         0
164              1  01-03-2014            98          20  …         0
1653             0  29-04-2013            13           1  …         0
…              …           …             …           …  …         …
1975             1  05-10-2013             0          16  …         0
1846             1  23-06-2014            29           5  …         0
1524             0  13-04-2013            56           2  …         0
21               0  06-01-2013            42           1  …         0
1245             0  18-05-2014            65           1  …         0

      Campaña4  Campaña5  Campaña1  Campaña2  Quejas  Zcostecontacto  \
617          0         0         0         0       0               3
687          0         0         0         0       0               3
```

```
1300          0          0          0          0       0                3
164           0          0          0          0       0                3
1653          0          0          0          0       0                3
…           …          …          …          …       …              …
1975          0          0          0          0       0                3
1846          0          0          0          0       0                3
1524          0          0          0          0       0                3
21            0          0          0          0       0                3
1245          0          0          0          0       0                3

      Zingresos  Respuesta  Pais
617          11          0    SP
687          11          0    US
1300         11          0    SP
164          11          0   IND
1653         11          0    SA
…           …          …    …
1975         11          0    SP
1846         11          0    SP
1524         11          0   IND
21           11          0    US
1245         11          0    SP

[2240 rows x 30 columns]
```

[44]:
```
#Histograma de la variable ingresos
datos4['Ingresos'].plot(kind='hist', cmap='Set3')
plt.xlabel('Cantidad de ingresos')
plt.ylabel('Frecuencia')
plt.show()
```



CONVERSION DE DT CUSTOMER(INSCRIPCION) PARA SACAR EL AÑO DE LOS CLIENTES

[45]:
```
import warnings

# Filtrar las advertencias para que el código no quede con 30 páginas de mensaje
warnings.filterwarnings("ignore", category=UserWarning, message="Parsing .* in␣
 ↪DD/MM/YYYY format.*")

#Convertir la variable inscripción que es tipo objeto a fecha para poder␣
 ↪manipularla luego
datos4['Inscripcion'] = pd.to_datetime(datos4['Inscripcion'])

# Para que vuelvan a saltar
warnings.resetwarnings()
```

[46]:
```
#Comprobración de que se ha transformado bien la variable a datetime
datos4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 30 columns):
```

```
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Identificador      2240 non-null   int64
 1   Cumpleaños         2240 non-null   int64
 2   Educacion          2240 non-null   object
 3   Estadocivil        2240 non-null   object
 4   Ingresos           2240 non-null   float64
 5   Niños              2240 non-null   int64
 6   Adolescentes       2240 non-null   int64
 7   Inscripcion        2240 non-null   datetime64[ns]
 8   Ultimacompra       2240 non-null   int64
 9   Totalvinos         2240 non-null   int64
 10  Totalfrutas        2240 non-null   int64
 11  Totalcarnes        2240 non-null   int64
 12  Totalpescados      2240 non-null   int64
 13  Totaldulces        2240 non-null   int64
 14  Totallujos         2240 non-null   int64
 15  Comprasdescuentos  2240 non-null   int64
 16  Comprasweb         2240 non-null   int64
 17  Comprascatalogo    2240 non-null   int64
 18  Comprastiendas     2240 non-null   int64
 19  Visitaswebmes      2240 non-null   int64
 20  Campaña3           2240 non-null   int64
 21  Campaña4           2240 non-null   int64
 22  Campaña5           2240 non-null   int64
 23  Campaña1           2240 non-null   int64
 24  Campaña2           2240 non-null   int64
 25  Quejas             2240 non-null   int64
 26  Zcostecontacto     2240 non-null   int64
 27  Zingresos          2240 non-null   int64
 28  Respuesta          2240 non-null   int64
 29  Pais               2240 non-null   object
dtypes: datetime64[ns](1), float64(1), int64(25), object(3)
memory usage: 525.1+ KB
```

[47]:
```
# Comprobación de todos los datos en formato tabla
datos4
```

[47]:

	Identificador	Cumpleaños	Educacion	Estadocivil	Ingresos	Niños	\
0	5524	1957	Graduado	Soltero	58138.0	0	
1	2174	1954	Graduado	Soltero	46344.0	1	
2	4141	1965	Graduado	Parejadehecho	71613.0	0	
3	6182	1984	Graduado	Parejadehecho	26646.0	1	
4	5324	1981	PhD	Casado	58293.0	1	
…	…	…	…	…	…	…	
2235	10870	1967	Graduado	Casado	61223.0	0	
2236	4001	1946	PhD	Parejadehecho	64014.0	2	

```
2237           7270        1981  Graduado     Divorciado   56981.0      0
2238           8235        1956    Master  Parejadehecho   69245.0      0
2239           9405        1954       PhD         Casado   52869.0      1

      Adolescentes Inscripcion  Ultimacompra  Totalvinos  …  Campaña3  \
0                0  2012-04-09            58         635  …         0
1                1  2014-08-03            38          11  …         0
2                0  2013-08-21            26         426  …         0
3                0  2014-10-02            26          11  …         0
4                0  2014-01-19            94         173  …         0
…              …         …             …           …  …        …
2235             1  2013-06-13            46         709  …         0
2236             1  2014-10-06            56         406  …         0
2237             0  2014-01-25            91         908  …         0
2238             1  2014-01-24             8         428  …         0
2239             1  2012-10-15            40          84  …         0

      Campaña4  Campaña5  Campaña1  Campaña2  Quejas  Zcostecontacto  \
0            0         0         0         0       0               3
1            0         0         0         0       0               3
2            0         0         0         0       0               3
3            0         0         0         0       0               3
4            0         0         0         0       0               3
…          …         …         …         …     …               …
2235         0         0         0         0       0               3
2236         0         0         1         0       0               3
2237         1         0         0         0       0               3
2238         0         0         0         0       0               3
2239         0         0         0         0       0               3

      Zingresos  Respuesta  Pais
0            11          1    US
1            11          0    US
2            11          0    US
3            11          0    US
4            11          0    US
…          …         …    …
2235         11          0    SA
2236         11          0    SA
2237         11          0    SA
2238         11          0    SA
2239         11          1    SA

[2240 rows x 30 columns]
```

[48]:
```
#Como no interesa tratar los datos con el año de nacimiento se crea una columna␣
↪que sean los años
```

```
datos4["añocompra"] = datos4["Inscripcion"].dt.year
datos4["añocompra"]
datos4["años"]= datos4["añocompra"] - datos4['Cumpleaños']
datos4["años"]
```

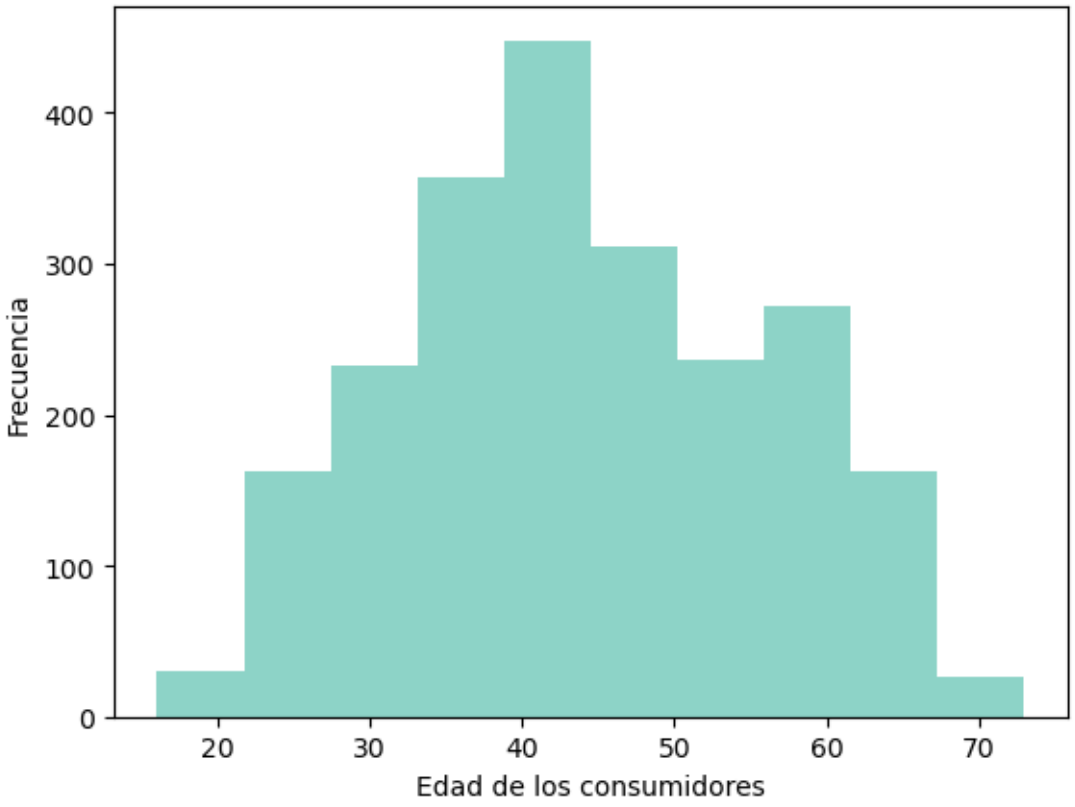
[48]:
```
0       55
1       60
2       48
3       30
4       33
        ..
2235    46
2236    68
2237    33
2238    58
2239    58
Name: años, Length: 2240, dtype: int64
```

[49]:
```
#Comprobación de que se ha creado la columna correspondiente
datos4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 32 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Identificador       2240 non-null   int64
 1   Cumpleaños          2240 non-null   int64
 2   Educacion           2240 non-null   object
 3   Estadocivil         2240 non-null   object
 4   Ingresos            2240 non-null   float64
 5   Niños               2240 non-null   int64
 6   Adolescentes        2240 non-null   int64
 7   Inscripcion         2240 non-null   datetime64[ns]
 8   Ultimacompra        2240 non-null   int64
 9   Totalvinos          2240 non-null   int64
 10  Totalfrutas         2240 non-null   int64
 11  Totalcarnes         2240 non-null   int64
 12  Totalpescados       2240 non-null   int64
 13  Totaldulces         2240 non-null   int64
 14  Totallujos          2240 non-null   int64
 15  Comprasdescuentos   2240 non-null   int64
 16  Comprasweb          2240 non-null   int64
 17  Comprascatalogo     2240 non-null   int64
 18  Comprastiendas      2240 non-null   int64
 19  Visitaswebmes       2240 non-null   int64
 20  Campaña3            2240 non-null   int64
 21  Campaña4            2240 non-null   int64
```

```
 22  Campaña5            2240 non-null   int64
 23  Campaña1            2240 non-null   int64
 24  Campaña2            2240 non-null   int64
 25  Quejas              2240 non-null   int64
 26  Zcostecontacto      2240 non-null   int64
 27  Zingresos           2240 non-null   int64
 28  Respuesta           2240 non-null   int64
 29  Pais                2240 non-null   object
 30  añocompra           2240 non-null   int64
 31  años                2240 non-null   int64
dtypes: datetime64[ns](1), float64(1), int64(27), object(3)
memory usage: 560.1+ KB
```

[50]:
```
#Histograma para ver la edad de los consumidores
datos4['años'].plot(kind='hist', cmap='Set3')
plt.xlabel('Edad de los consumidores')
plt.ylabel('Frecuencia')
plt.show()
```



[51]:
```
#Datos estadísticos con la edad de los consumidores y no el año de nacimiento
datos4['años'].describe()
```

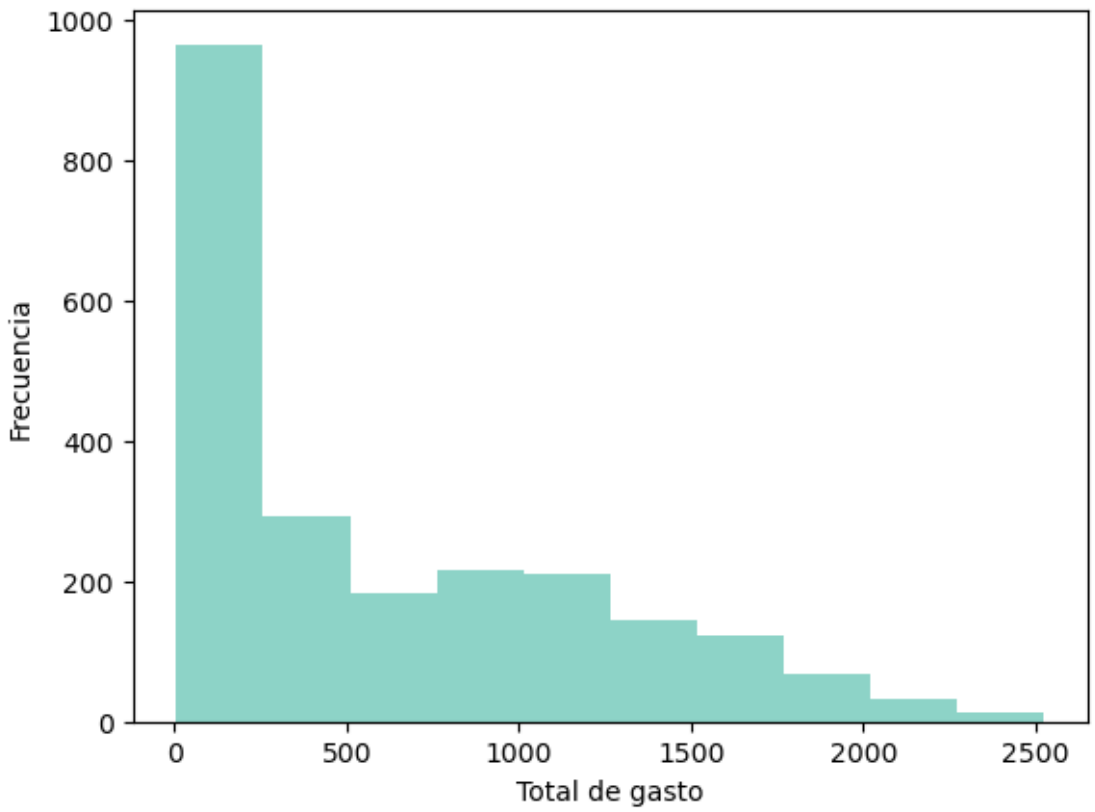
[51]:
```
count    2240.000000
mean       44.126339
std        11.731156
min        16.000000
25%        36.000000
50%        43.000000
75%        54.000000
max        73.000000
Name: años, dtype: float64
```

ANALIZAR EL TOTAL DE GASTO DE LOS CLIENTES

[52]:
```
#Creación de una variable con el total del gasto de los distintos productos
datos4['Totalgasto']= datos4.loc[:, ['Totalvinos', 'Totalfrutas',
      'Totalcarnes', 'Totalpescados', 'Totaldulces',
       'Totallujos']].sum(axis=1)
datos4['Totalgasto']
```

[52]:
```
0       1617
1         27
2        776
3         53
4        422
        ...
2235    1341
2236     444
2237    1241
2238     843
2239     172
Name: Totalgasto, Length: 2240, dtype: int64
```

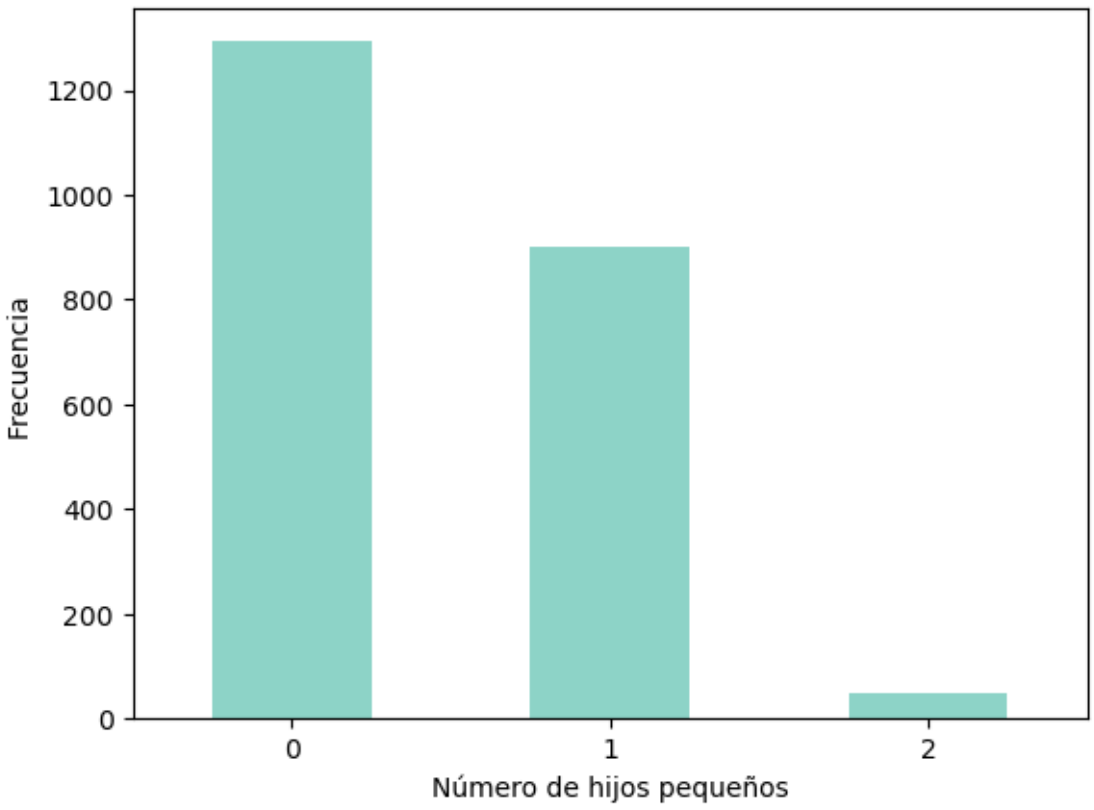
[53]:
```
#Histograma del total de gasto
datos4['Totalgasto'].plot(kind='hist', cmap='Set3')
plt.xlabel('Total de gasto')
plt.ylabel('Frecuencia')
plt.show()
```



ANALIZAR EL NUMERO TOTAL DE HIJOS

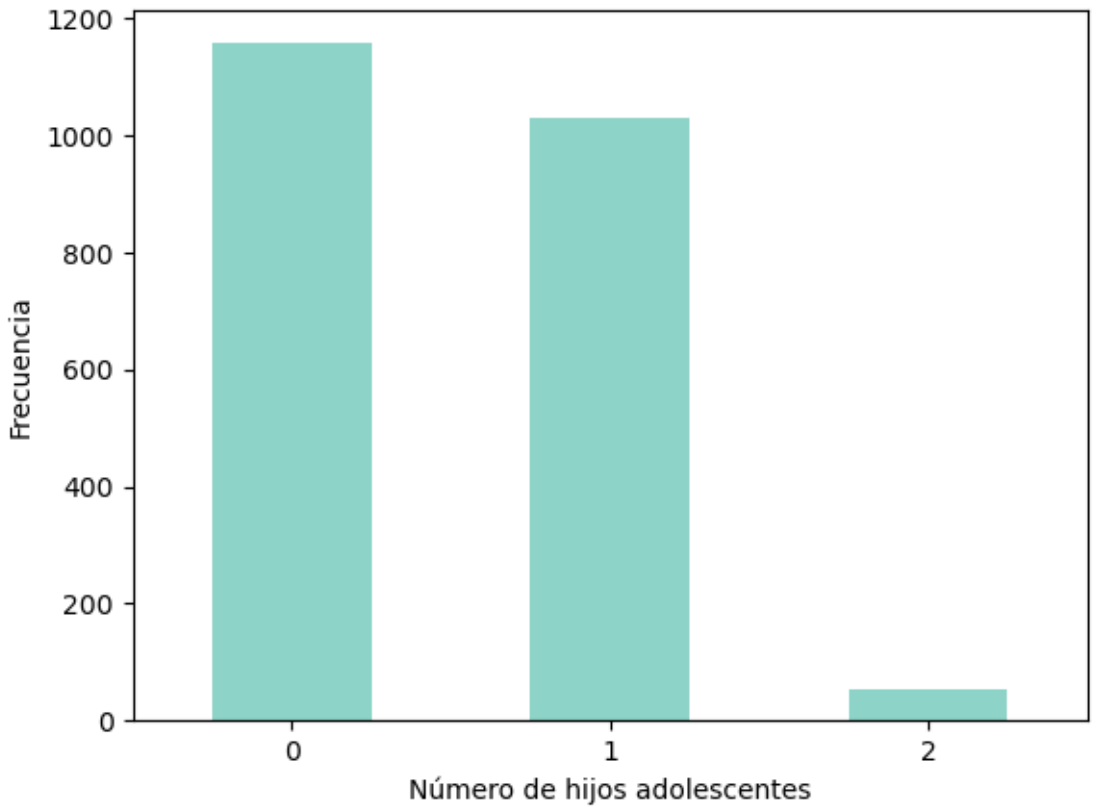
[54]:
```
# Gráfico de barras de los niños que hay en cada familia de los consumidores
datos4['Niños'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Número de hijos pequeños')
plt.ylabel('Frecuencia ')
```

[54]:
```
Text(0, 0.5, 'Frecuencia ')
```



[55]:
```
# Gráfica de barra de los adolescentes que hay en cada familia de los␣
 ↪consumidores
datos4['Adolescentes'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Número de hijos adolescentes')
plt.ylabel('Frecuencia')
```

[55]:
```
Text(0, 0.5, 'Frecuencia')
```



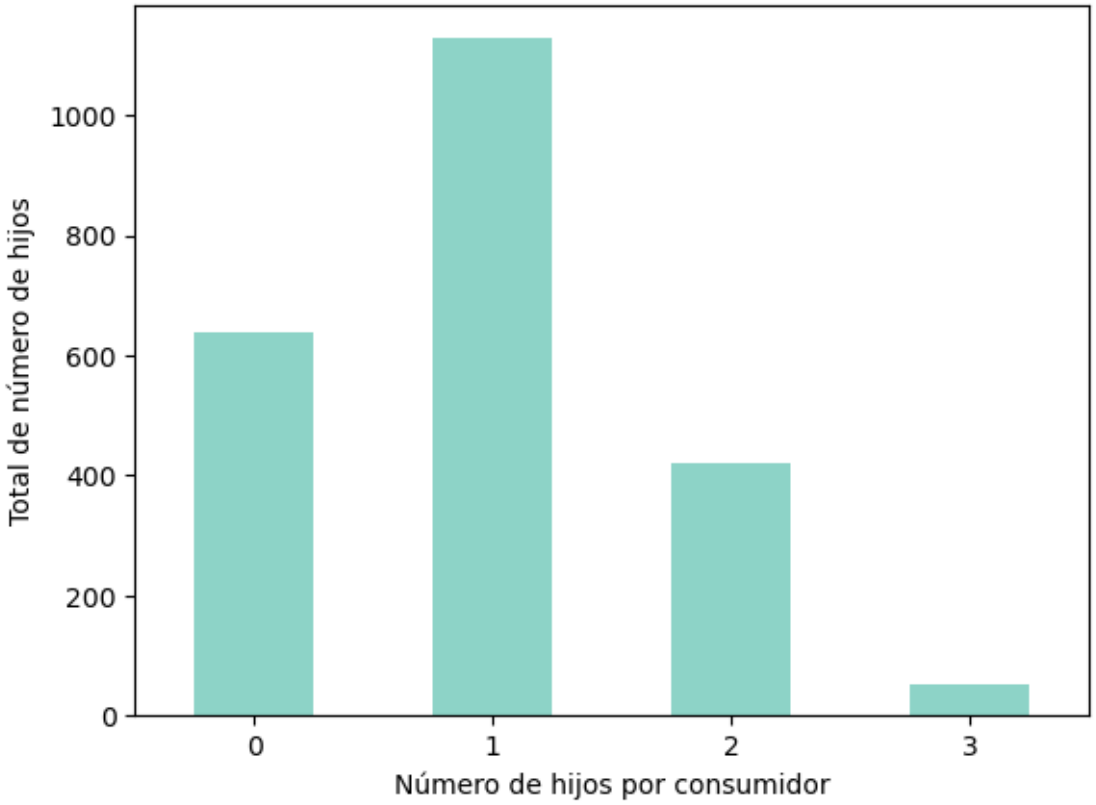
[56]:
```
# Creación de una nueva variable del total de hijos que tienen los consumidores
datos4['Totalhijos']= datos4.loc[:, ['Niños', 'Adolescentes']].sum(axis=1)
datos4['Totalhijos']
```

[56]:
```
0       0
1       2
2       0
3       1
4       1
       ..
2235    1
2236    3
2237    0
2238    1
2239    2
Name: Totalhijos, Length: 2240, dtype: int64
```

[57]:
```
# Representación gráfica de los hijos totales que hay en la familia de los␣
 ↪consumidores
datos4['Totalhijos'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
```

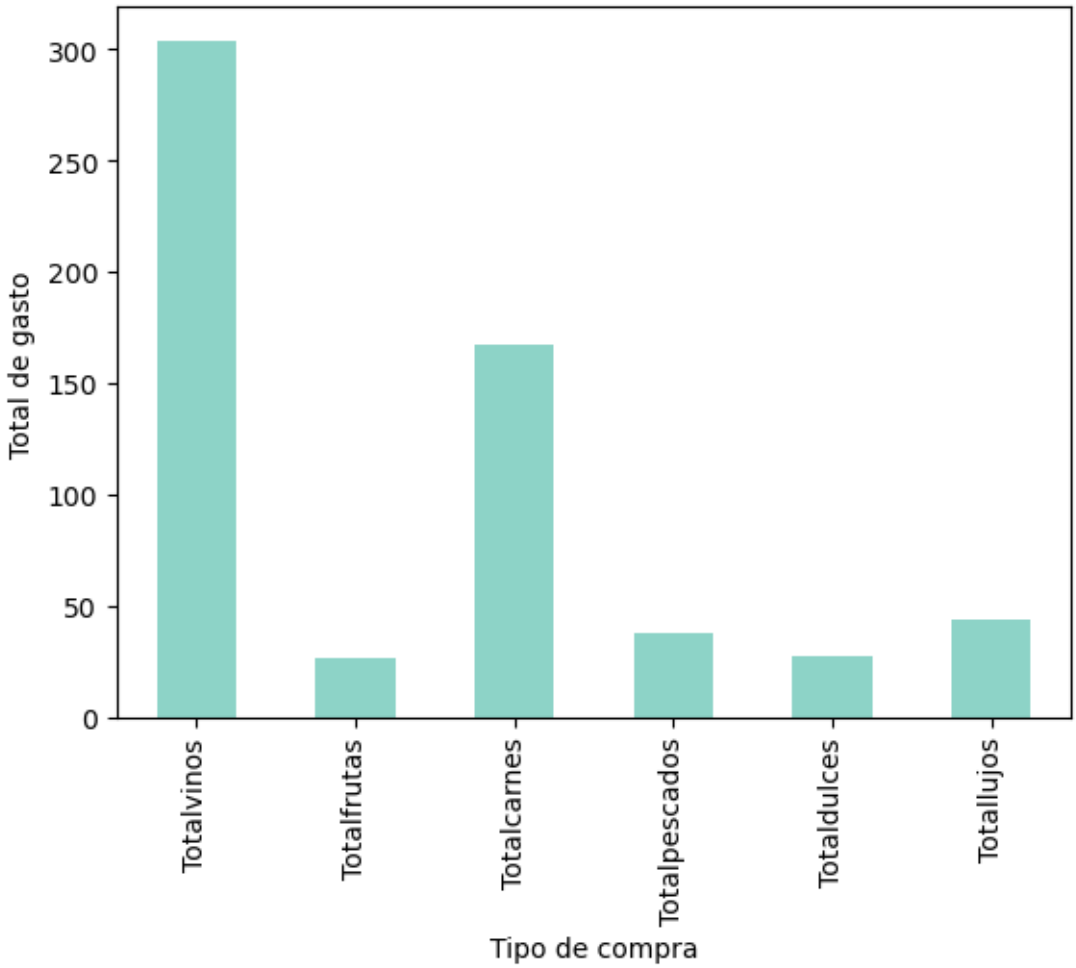
```
plt.xlabel('Número de hijos por consumidor')
plt.ylabel('Total de número de hijos')
```

[57]: Text(0, 0.5, 'Total de número de hijos')

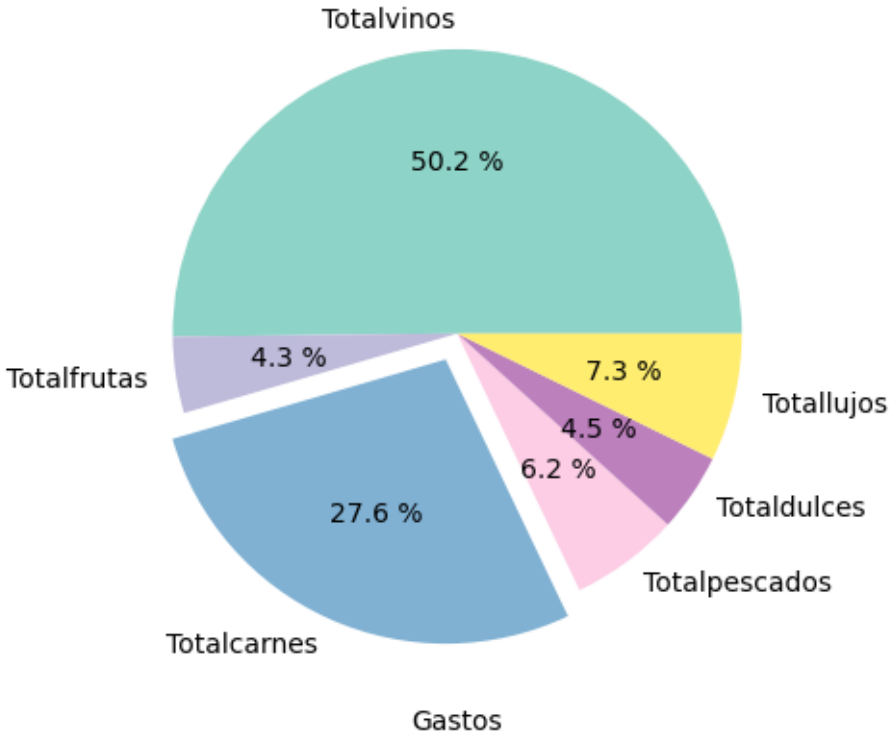


ANÁLISIS INDIVIDUAL DE COMPRA

[58]:
```
# Representación del gasto medio de cada tipo de compra
columnas_loc = datos4.loc[:, ['Totalvinos', 'Totalfrutas',
       'Totalcarnes', 'Totalpescados', 'Totaldulces',
       'Totallujos']]
medias = columnas_loc.mean()
medias.plot.bar( cmap='Set3')
plt.xlabel('Tipo de compra')
plt.ylabel('Total de gasto')
plt.show()
```



[59]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
columnas_loc = datos4.loc[:, ['Totalvinos', 'Totalfrutas',
       'Totalcarnes', 'Totalpescados', 'Totaldulces',
       'Totallujos']]
medias = columnas_loc.mean()
desfase=(0,0,0.1,0,0,0)
medias.plot.pie( cmap='Set3',  autopct="%0.1f %%", explode=desfase)
plt.xlabel('Gastos')
plt.ylabel('    ')
plt.show()
```



[60]:
```
# ordenado descendente
educacion=datos4.groupby(['años'])['Totallujos'].mean()
educacion.sort_values(ascending=False)
```

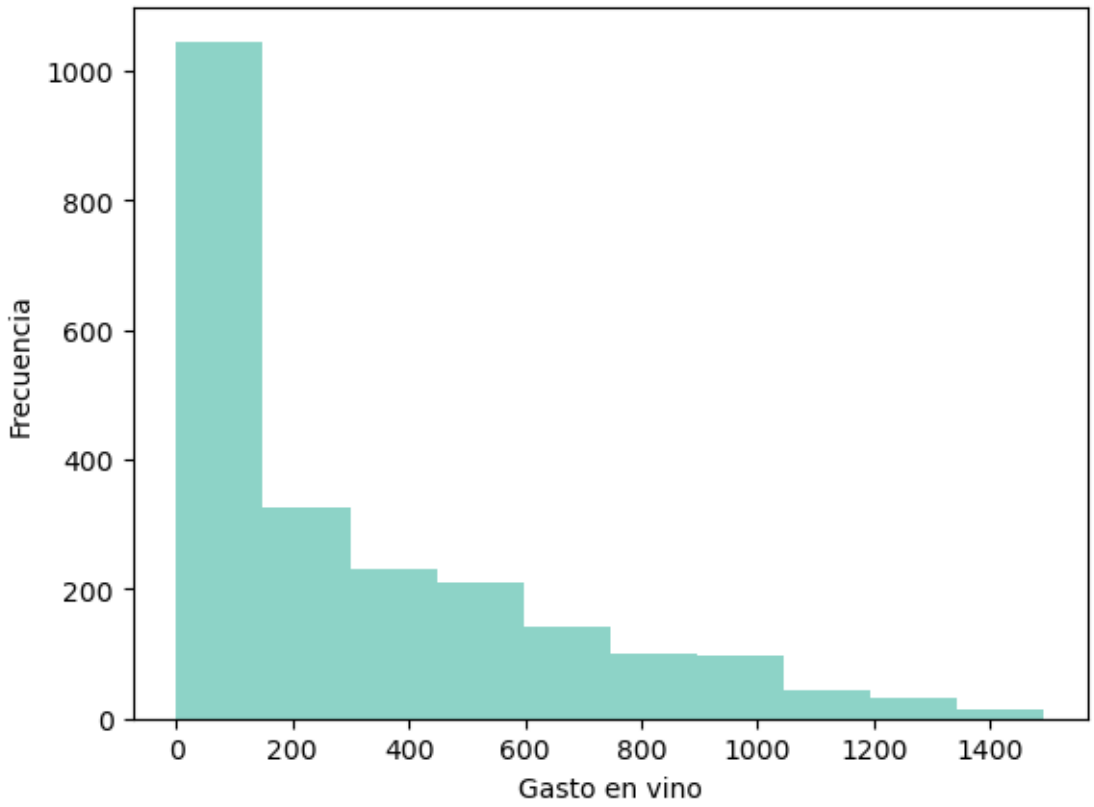
[60]:
```
años
20    86.000000
22    80.923077
19    74.250000
68    67.333333
62    65.440000
61    60.083333
29    58.095238
51    56.977778
52    56.422222
66    56.380952
59    53.829268
50    52.586957
67    51.666667
57    51.000000
16    50.000000
21    49.846154
56    49.681818
```

```
17    49.500000
55    48.372881
44    47.323944
27    47.302326
45    47.245283
23    47.000000
58    46.843137
25    44.962963
60    44.854167
47    44.480000
48    44.163934
64    44.031250
34    42.280702
38    42.000000
49    41.692308
35    41.447761
46    41.285714
31    41.177778
54    40.882353
42    40.730337
41    40.197802
37    39.780488
63    39.533333
32    39.340909
30    39.260000
43    39.253521
24    39.166667
36    38.984127
70    38.333333
28    38.114286
65    34.111111
33    33.736842
40    33.275362
53    30.222222
39    29.140351
18    28.666667
26    26.117647
72    21.000000
69    17.833333
71    12.000000
73     6.000000
Name: Totallujos, dtype: float64
```

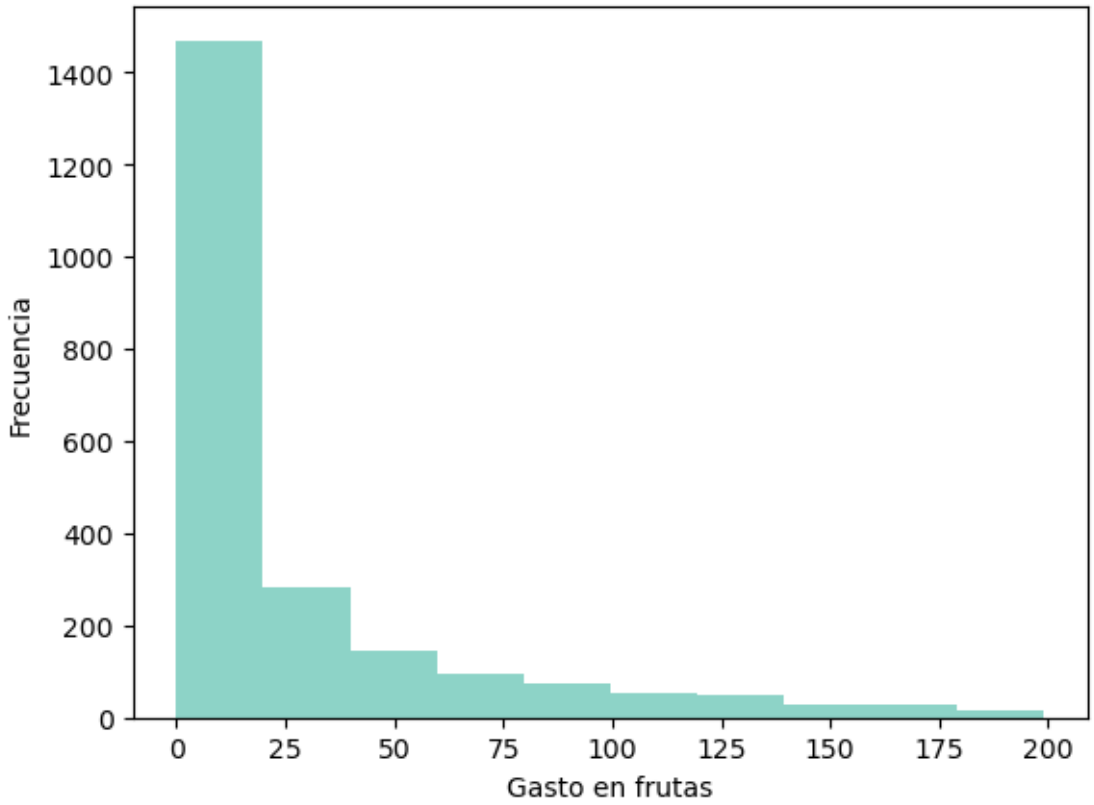
[61]:
```
vinos=datos4.groupby(['Educacion'])['Totallujos'].apply(lambda x: x.mode())
vinos.sort_values(ascending=False)
```

[61]:
```
Educacion
Basico      0    15
Master      0     3
Graduado    0     1
PhD         0     0
Name: Totallujos, dtype: int64
```

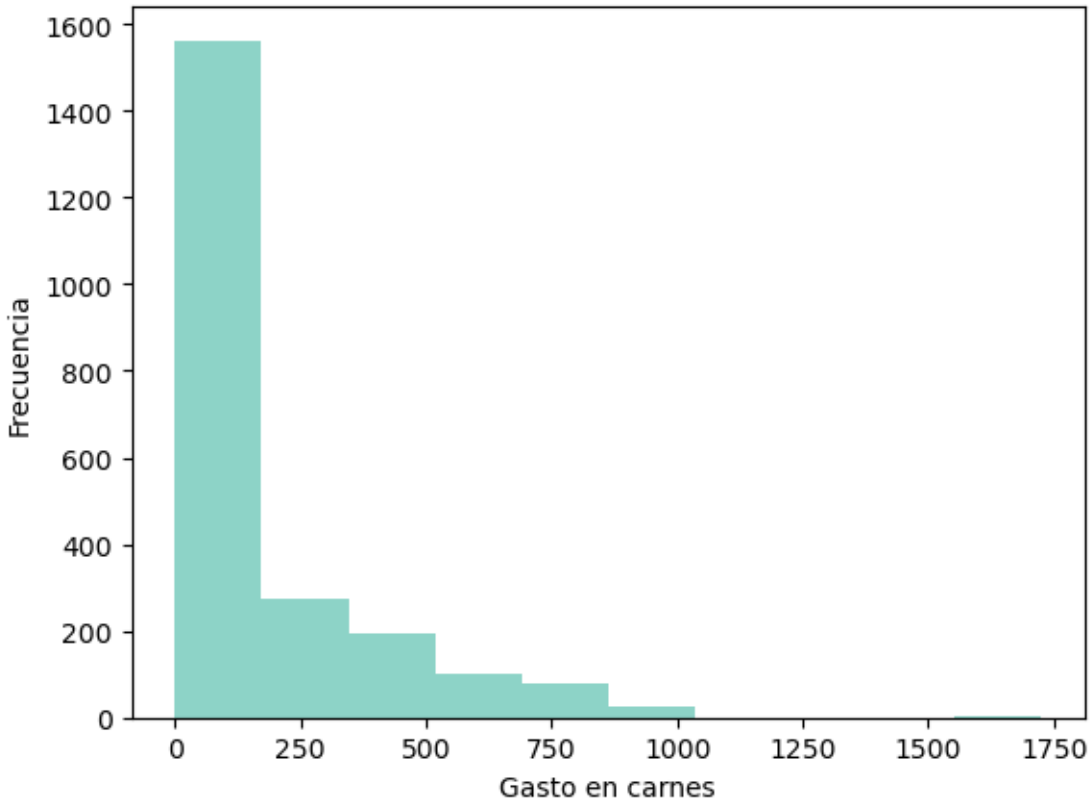
[62]:
```
# Representación del gasto en vino de los consumidores
datos4['Totalvinos'].plot(kind='hist', cmap='Set3')
plt.xlabel('Gasto en vino')
plt.ylabel('Frecuencia')
plt.show()
```



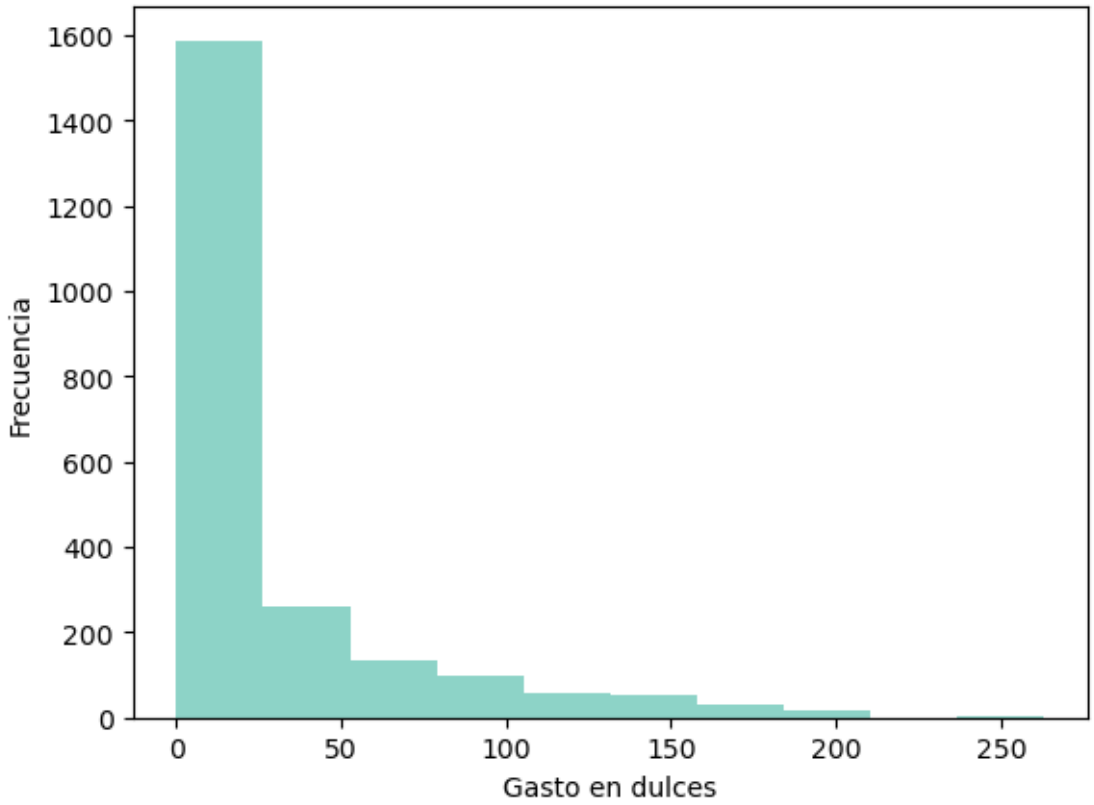
[63]:
```
# Representación del gasto en fruta de los consumidores
datos4['Totalfrutas'].plot(kind='hist', cmap='Set3')
plt.xlabel('Gasto en frutas')
plt.ylabel('Frecuencia')
plt.show()
```



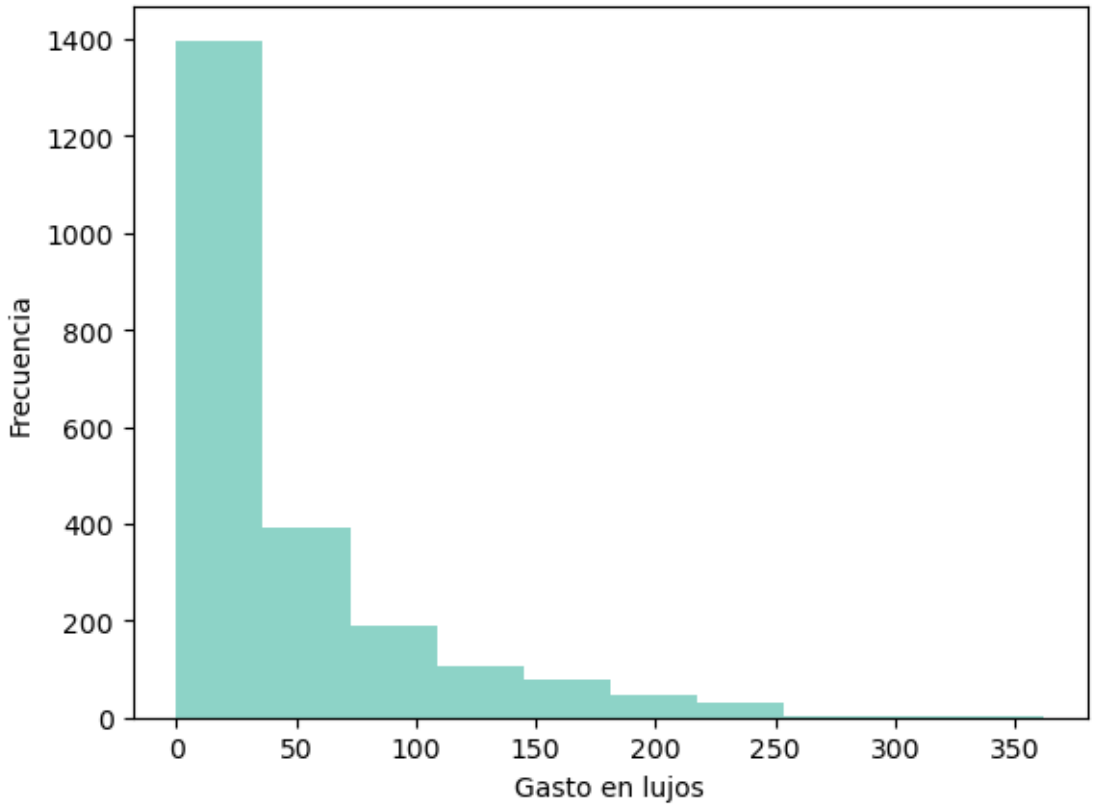
[64]:
```
# Representación del gasto en carne de los consumidores
datos4['Totalcarnes'].plot(kind='hist', cmap='Set3')
plt.xlabel('Gasto en carnes')
plt.ylabel('Frecuencia')
plt.show()
```



[65]:
```
# Representación del gasto en dulces de los consumidores
datos4['Totaldulces'].plot(kind='hist', cmap='Set3')
plt.xlabel('Gasto en dulces')
plt.ylabel('Frecuencia')
plt.show()
```



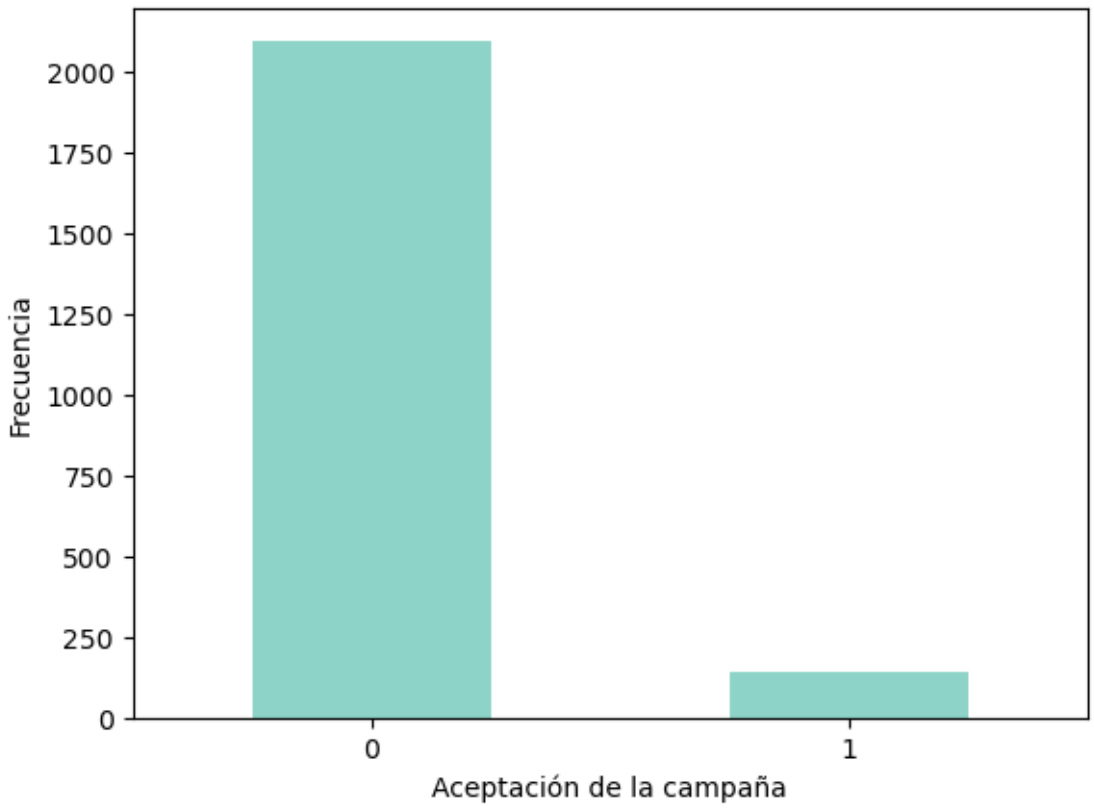
[66]:
```
# Representación del gasto en lujos de los consumidores
datos4['Totallujos'].plot(kind='hist', cmap='Set3')
plt.xlabel('Gasto en lujos')
plt.ylabel('Frecuencia')
plt.show()
```



Análisis de la aceptación de las campañas

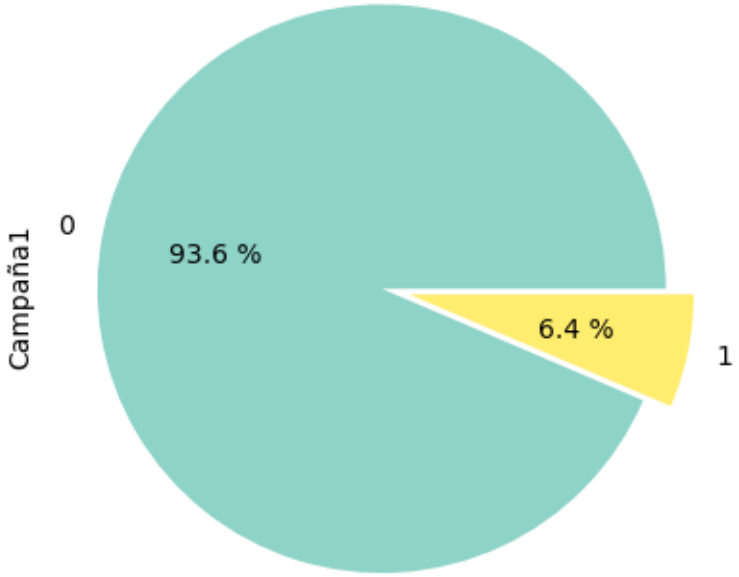
[67]:
```
# Representación de aceptación de la campaña 1
datos4['Campaña1'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Aceptación de la campaña')
plt.ylabel('Frecuencia')
```

[67]:
```
Text(0, 0.5, 'Frecuencia')
```



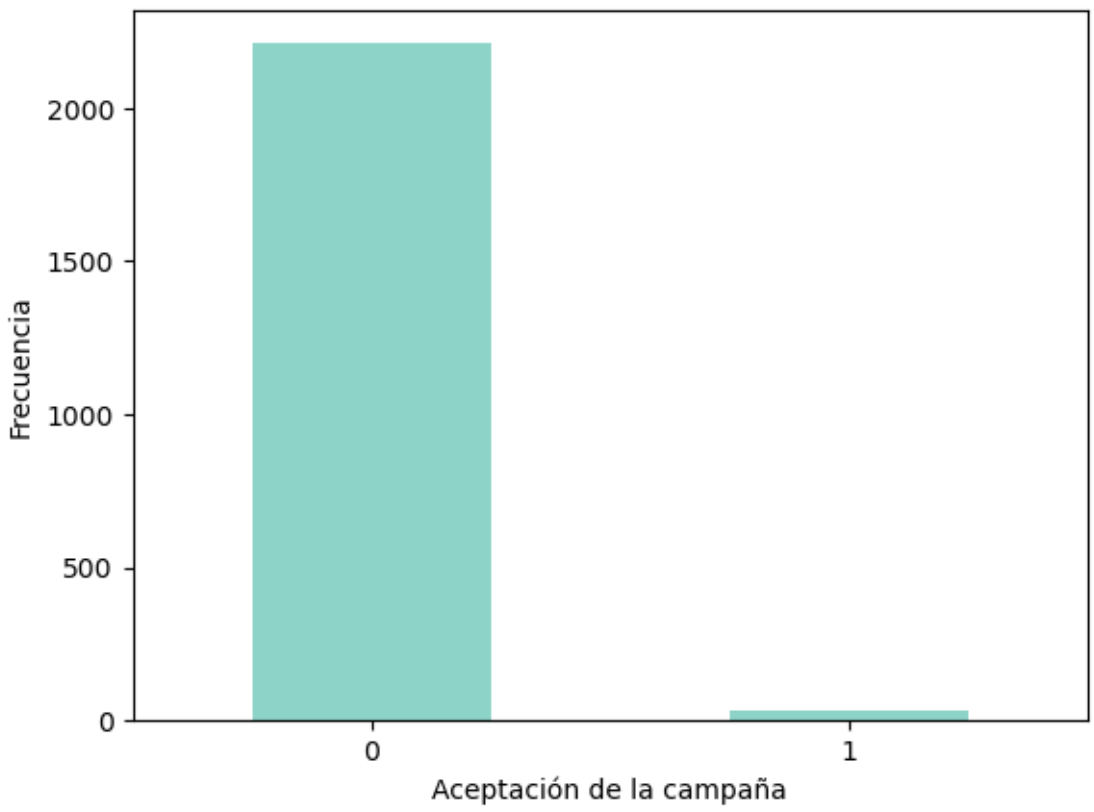
[68]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0.1, 0)
datos4['Campaña1'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[68]: <AxesSubplot:ylabel='Campaña1'>



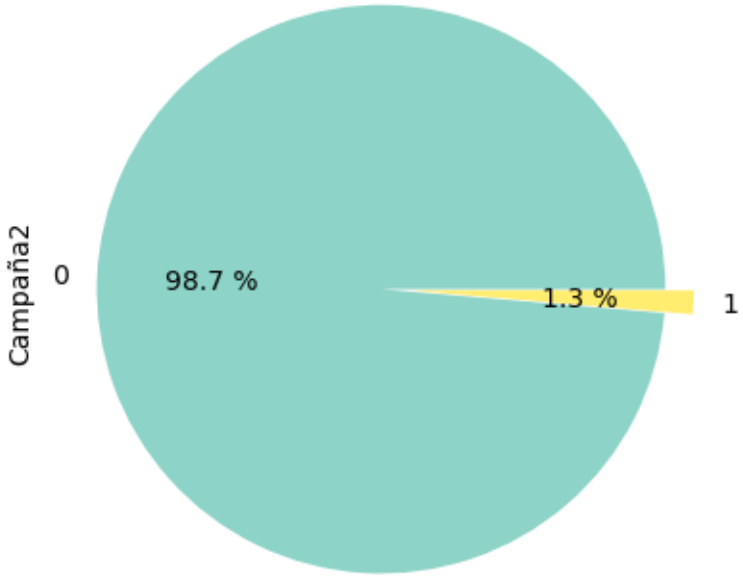
[69]:
```
# Representación de aceptación de la campaña 2
datos4['Campaña2'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
↪cmap='Set3')
plt.xlabel('Aceptación de la campaña')
plt.ylabel('Frecuencia')
```

[69]:
```
Text(0, 0.5, 'Frecuencia')
```



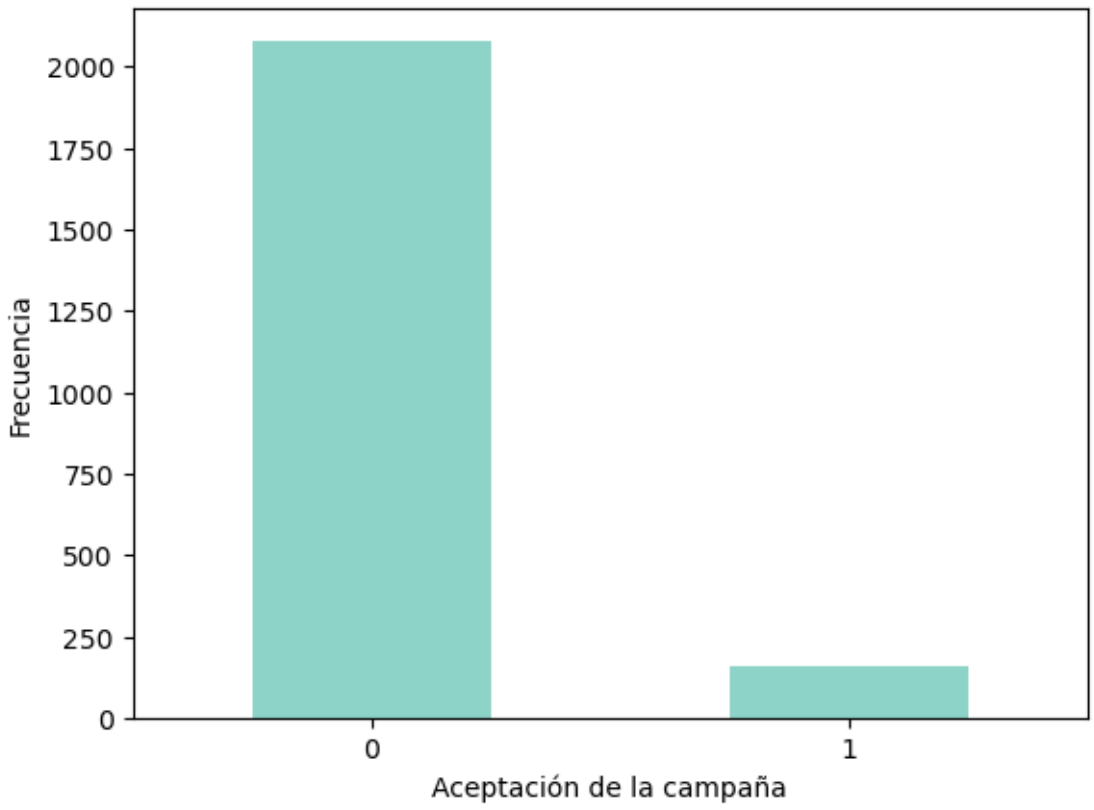
[70]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0.1, 0)
datos4['Campaña2'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[70]: <AxesSubplot:ylabel='Campaña2'>



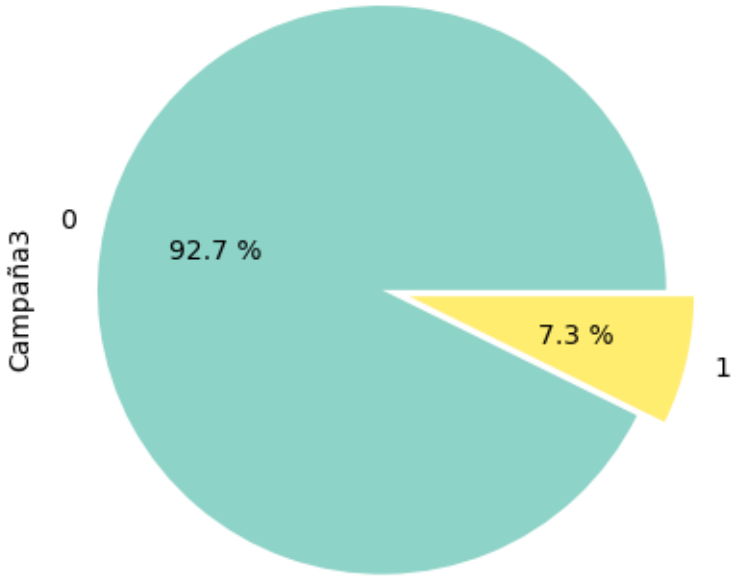
[71]:
```
# Representación de aceptación de la campaña 3
datos4['Campaña3'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
↪cmap='Set3')
plt.xlabel('Aceptación de la campaña')
plt.ylabel('Frecuencia')
```

[71]:
```
Text(0, 0.5, 'Frecuencia')
```



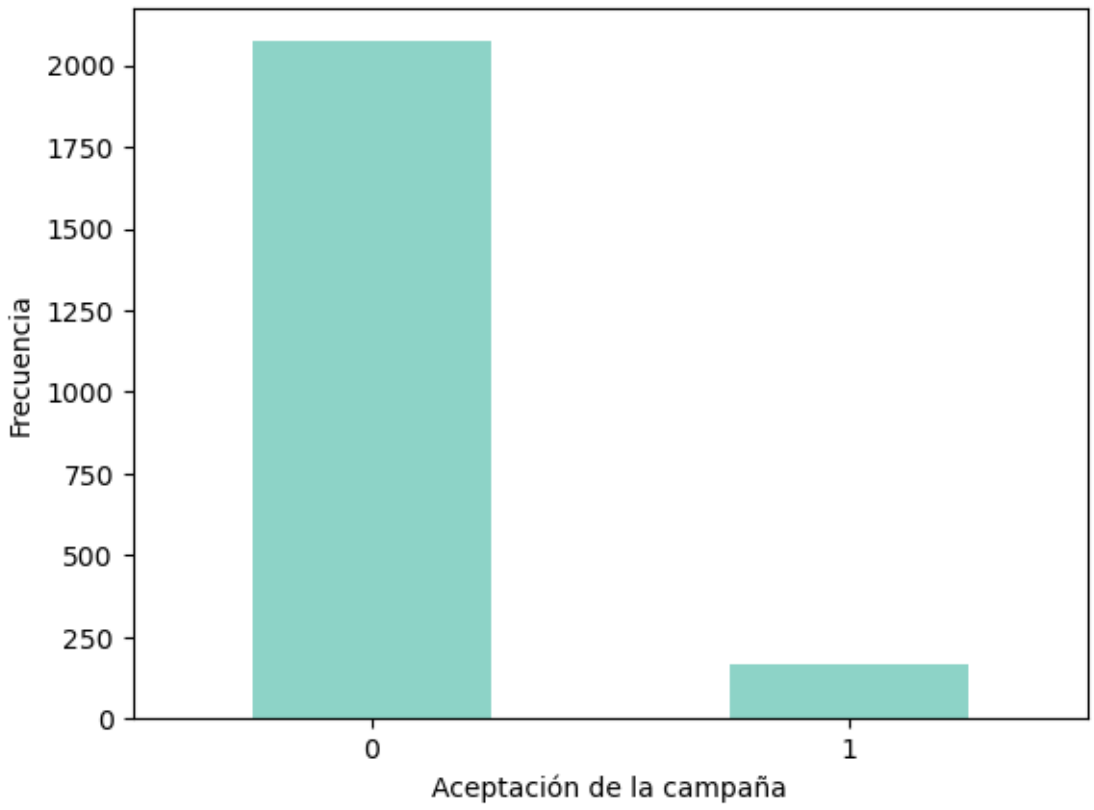
[72]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0.1, 0)
datos4['Campaña3'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[72]: <AxesSubplot:ylabel='Campaña3'>



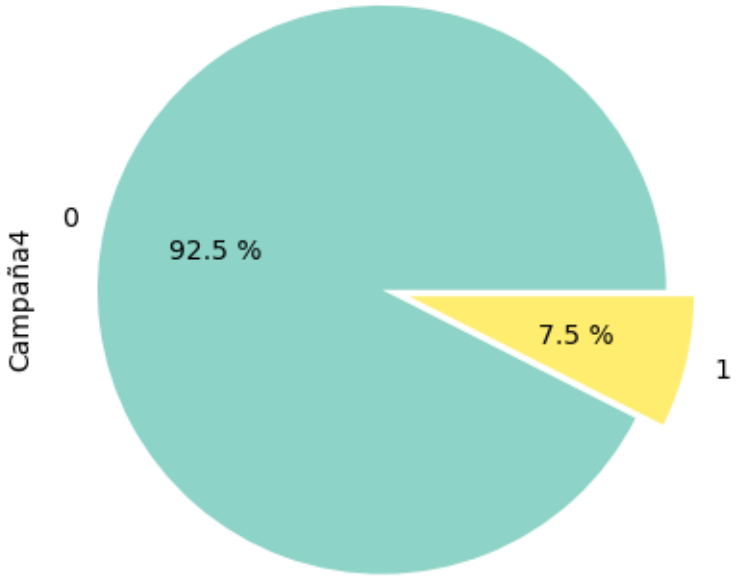
[73]:
```
# Representación de aceptación de la campaña 4
datos4['Campaña4'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Aceptación de la campaña')
plt.ylabel('Frecuencia')
```

[73]:
```
Text(0, 0.5, 'Frecuencia')
```



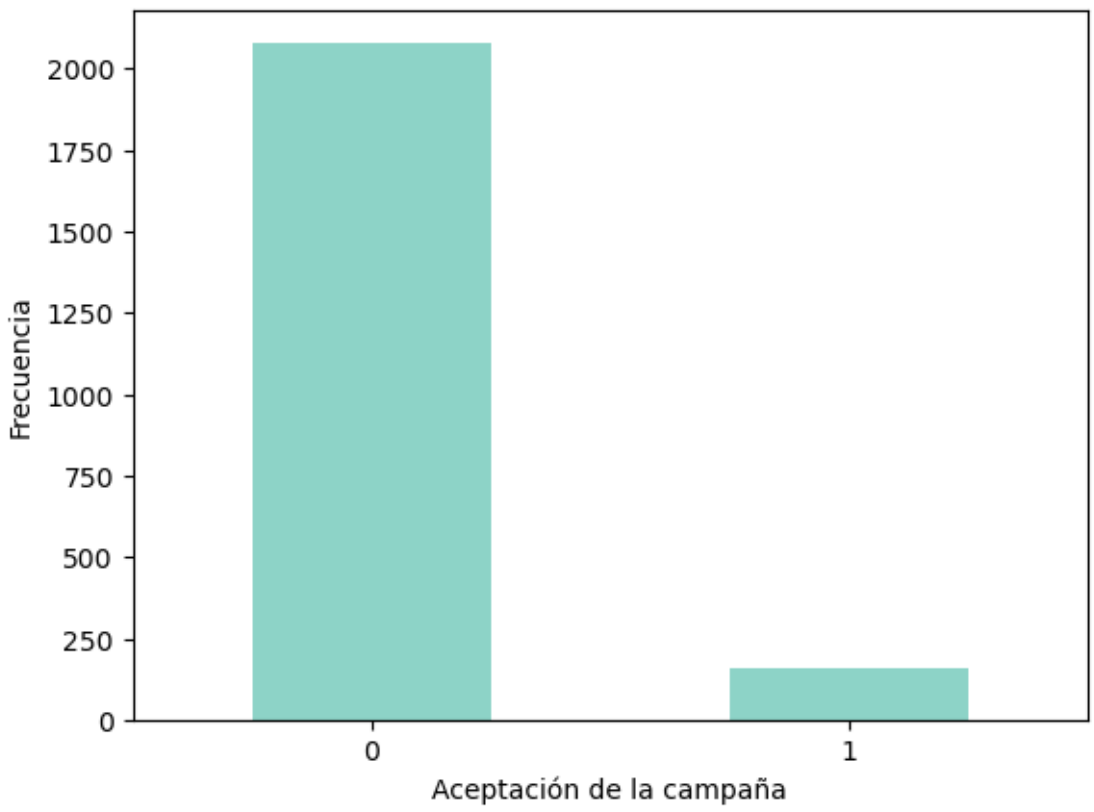
[74]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0.1, 0)
datos4['Campaña4'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[74]: <AxesSubplot:ylabel='Campaña4'>



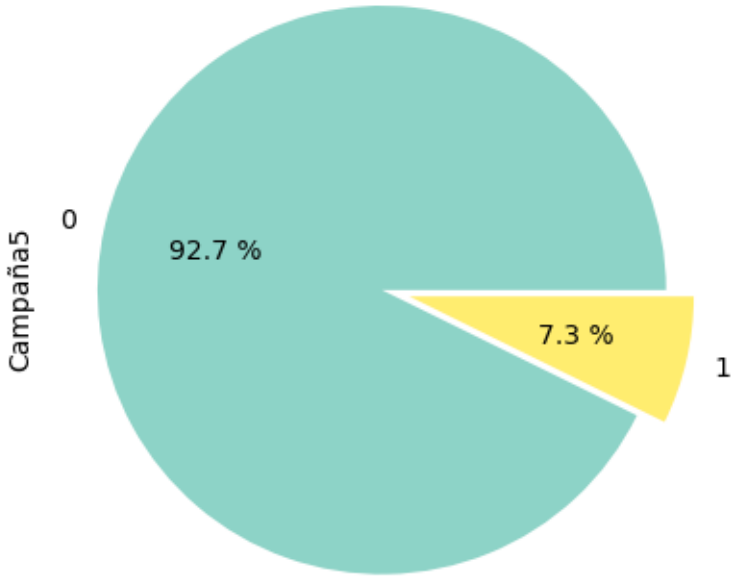
[75]:
```
# Representación de aceptación de la campaña 5
datos4['Campaña5'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Aceptación de la campaña')
plt.ylabel('Frecuencia')
```

[75]:
```
Text(0, 0.5, 'Frecuencia')
```

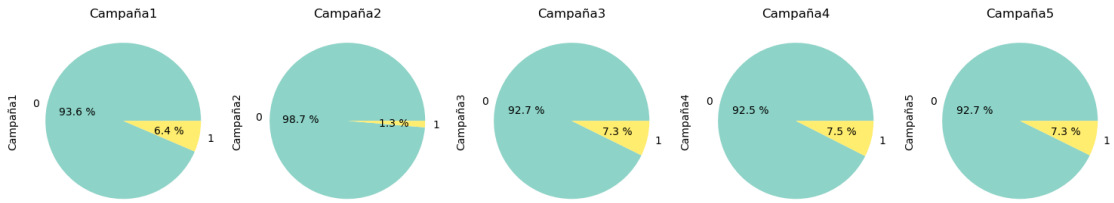


[76]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0.1, 0)
datos4['Campaña5'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[76]: <AxesSubplot:ylabel='Campaña5'>



[77]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Escogemos las columnas de las campañas
columnas = ['Campaña1', 'Campaña2', 'Campaña3', 'Campaña4', 'Campaña5']

# Crear los subplots
fig, axs = plt.subplots(1, 5, figsize=(15, 3))

# Generar las gráficas en bucle
for i, columna in enumerate(columnas):
    datos4[columna].value_counts().plot.pie(
        rot=0, cmap='Set3', autopct="%0.1f %%", ax=axs[i])
    axs[i].set_title(columna)

# Ajustar los espacios
plt.tight_layout()
plt.show()
```



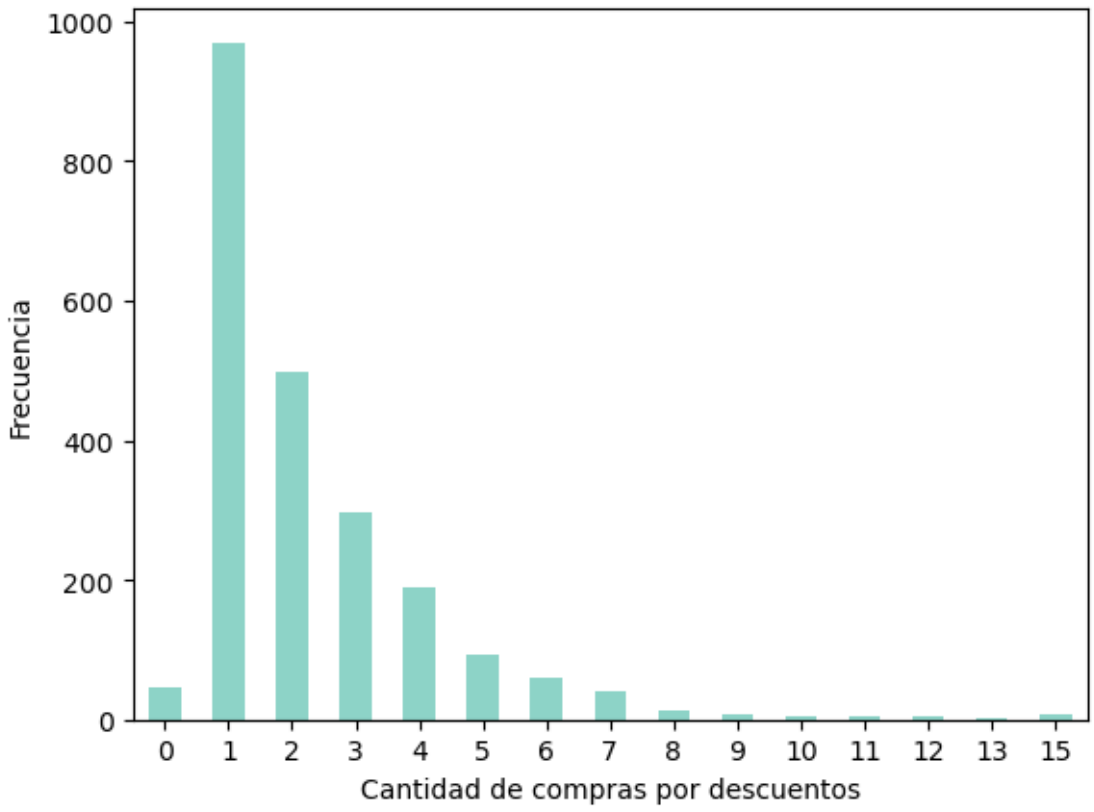
Análisis tipos de compras

[78]:
```
# Comprobación de los valores únicos que hay de las cantidades de compras con␣
↪descuentos
datos4['Comprasdescuentos'].unique()
```

[78]:
```
array([ 3,  2,  1,  5,  4, 15,  7,  0,  6,  9, 12,  8, 10, 13, 11],
      dtype=int64)
```

[79]:
```
# Representación gráfica de las cantidades de compras con descuentos que se␣
↪realizan  y su frecuencia
datos4['Comprasdescuentos'].value_counts(sort=False).sort_index().plot.
↪bar(rot=0, cmap='Set3')
plt.xlabel('Cantidad de compras por descuentos')
plt.ylabel('Frecuencia')
```

[79]:
```
Text(0, 0.5, 'Frecuencia')
```

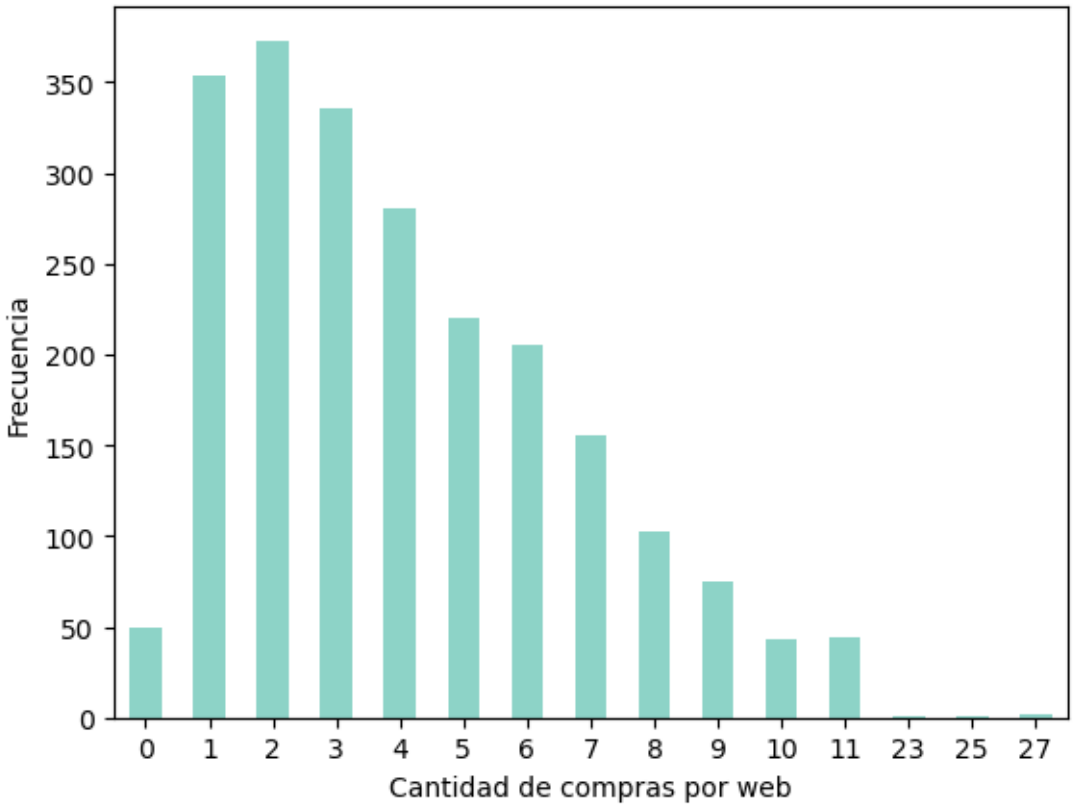


[80]:
```
# Comprobación de los valores únicos que hay de las cantidades de compras en web
datos4['Comprasweb'].unique()
```

[80]:
```
array([ 8,  1,  2,  5,  6,  7,  4,  3, 11,  0, 27, 10,  9, 23, 25],
      dtype=int64)
```

[81]:
```
# Representación gráfica de las cantidades de compras por web que se realizan y␣
 ↪su frecuencia
datos4['Comprasweb'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Cantidad de compras por web')
plt.ylabel('Frecuencia')
```

[81]:
```
Text(0, 0.5, 'Frecuencia')
```

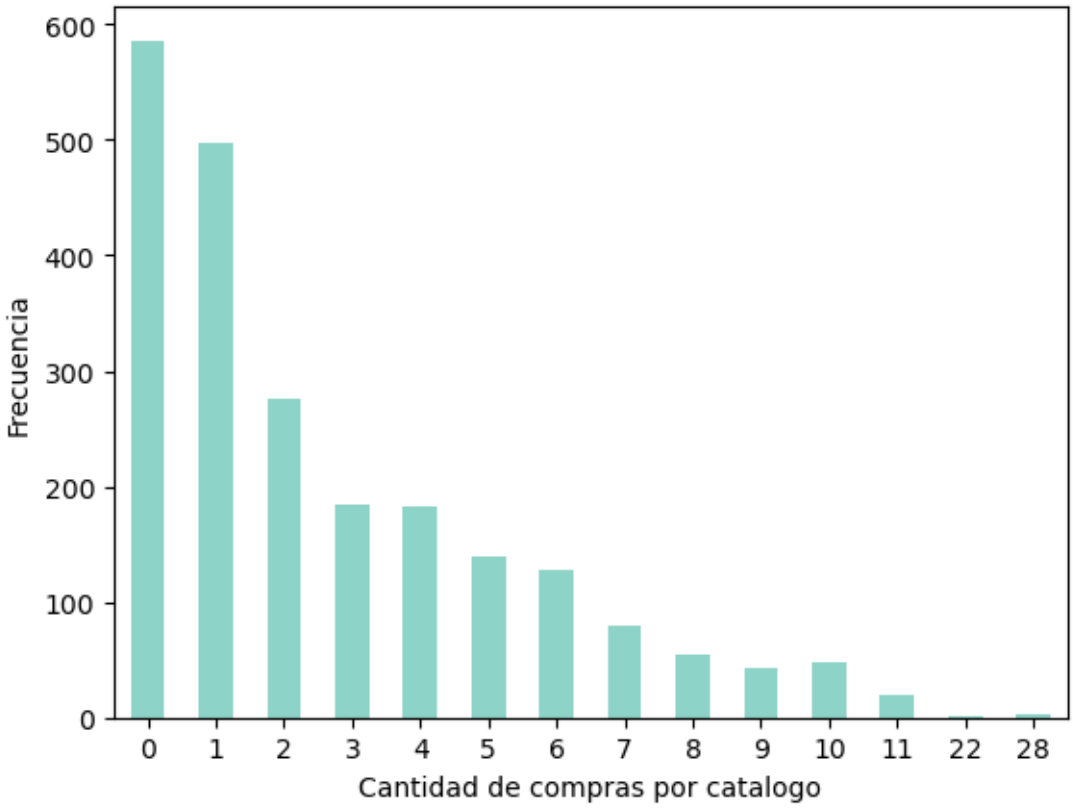


[82]:
```
# Comprobación de los valores únicos que hay de las cantidades de compras en␣
 ↪catálogo
datos4['Comprascatalogo'].unique()
```

[82]:
```
array([10,  1,  2,  0,  3,  4,  6, 28,  9,  5,  8,  7, 11, 22],
      dtype=int64)
```

[83]:
```
# Representación gráfica de las cantidades de compras por catálogo que se␣
 ↪realizan y su frecuencia
datos4['Comprascatalogo'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Cantidad de compras por catalogo')
plt.ylabel('Frecuencia')
```

[83]:
```
Text(0, 0.5, 'Frecuencia')
```

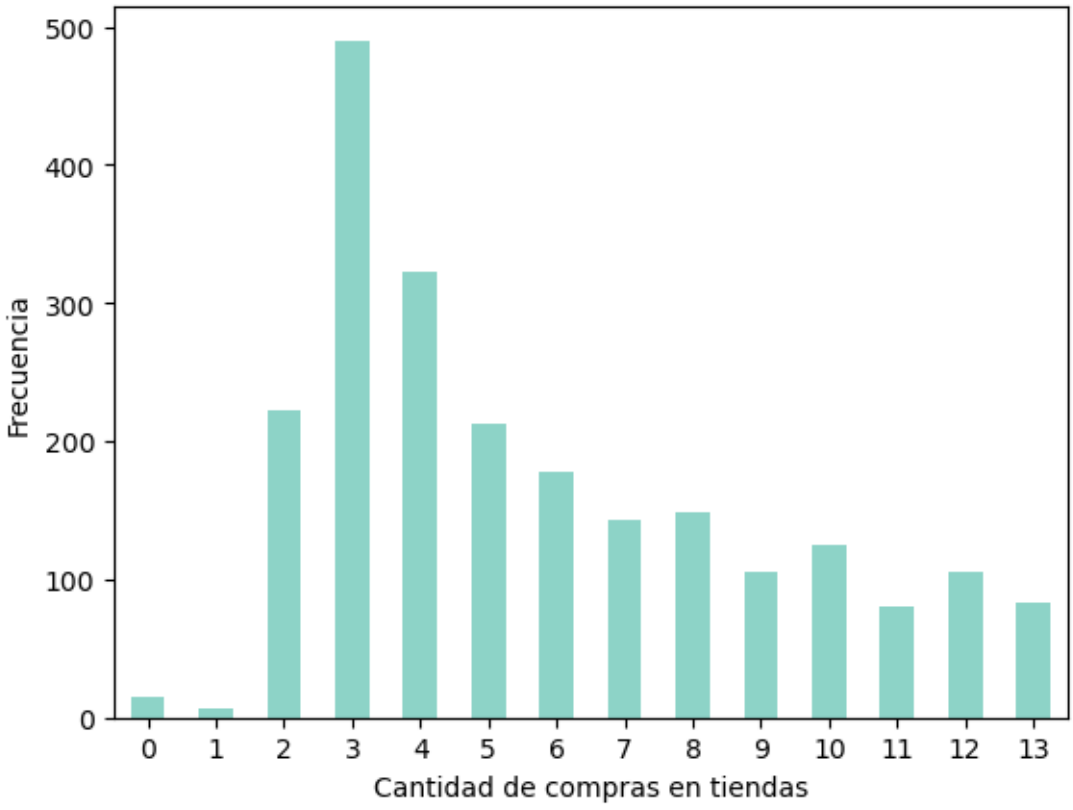


[84]:
```
# Comprobación de los valores únicos que hay de las cantidades de compras en␣
 ↪tienda
datos4['Comprastiendas'].unique()
```

[84]:
```
array([ 4,  2, 10,  6,  7,  0,  3,  8,  5, 12,  9, 13, 11,  1],
      dtype=int64)
```

[85]:
```
# Representación gráfica de las cantidades de compras en tienda que se realizan␣
 ↪y su frecuencia
datos4['Comprastiendas'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Cantidad de compras en tiendas')
plt.ylabel('Frecuencia')
```

[85]:
```
Text(0, 0.5, 'Frecuencia')
```

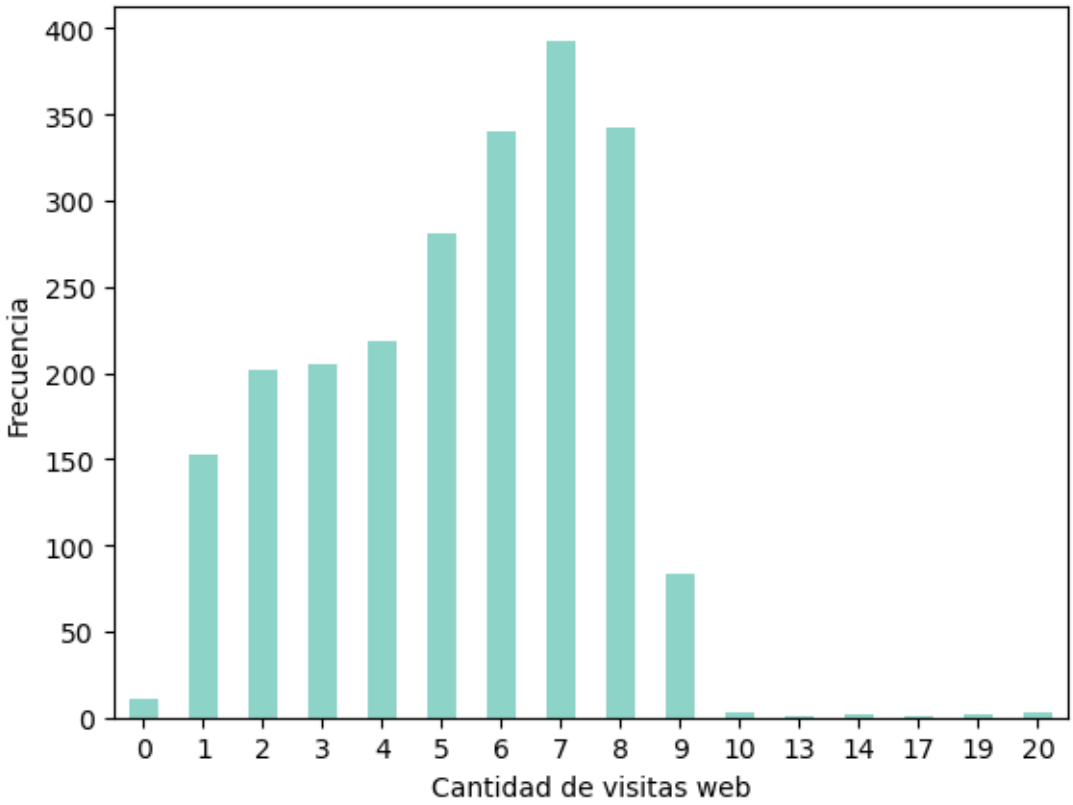


[86]:
```
# Comprobación de los valores únicos que hay devisitas a la web
datos4['Visitaswebmes'].unique()
```

[86]:
```
array([ 7,  5,  4,  6,  8,  9, 20,  2,  3,  1, 10,  0, 14, 19, 17, 13],
      dtype=int64)
```

[87]:
```
# Representación gráfica de las cantidades de visitas a la web y su frecuencia
datos4['Visitaswebmes'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Cantidad de visitas web')
plt.ylabel('Frecuencia')
```

[87]:
```
Text(0, 0.5, 'Frecuencia')
```



0.6 ANÁLISIS ESTADÍSTICO DE LAS VARIABLES CATEGÓRICAS

[88]:
```
# Descripción estadística de las variables categóricas
datos4[['Educacion','Estadocivil','Pais']].describe()
```

[88]:

	Educacion	Estadocivil	Pais
count	2240	2240	2240
unique	4	5	8
top	Graduado	Casado	SP
freq	1127	864	1095

Variable educación con distintas variables

[89]:
```
# Agrupar por variables cualitativas y calcular la media de ingresos
datos4.groupby(['Educacion'])['Ingresos'].mean()
```

[89]:
```
Educacion
Basico      20306.259259
Graduado    52170.574992
Master      51063.727768
PhD         56105.210405
Name: Ingresos, dtype: float64
```

[90]:
```
# ordenado descendente
educacion=datos4.groupby(['Educacion'])['Respuesta'].mean()
educacion.sort_values(ascending=False)
```

[90]:
```
Educacion
PhD          0.207819
Master       0.137871
Graduado     0.134871
Basico       0.037037
Name: Respuesta, dtype: float64
```

[91]:
```
datos4.groupby(['Niños'])['Educacion'].apply(lambda x: x.mode())
```

[91]:
```
Niños
0      0    Graduado
1      0    Graduado
2      0    Graduado
Name: Educacion, dtype: object
```

[92]:
```
# ordenado descendente
educacion=datos4.groupby(['Educacion'])['Niños'].mean()
educacion.sort_values(ascending=False)
```

[92]:
```
Educacion
Basico       0.629630
Master       0.462478
Graduado     0.444543
PhD          0.401235
Name: Niños, dtype: float64
```

[93]:
```
datos4.groupby(['Adolescentes'])['Educacion'].apply(lambda x: x.mode())
```

[93]:
```
Adolescentes
0              0    Graduado
1              0    Graduado
2              0    Graduado
Name: Educacion, dtype: object
```

[94]:
```
# ordenado descendente
educacion=datos4.groupby(['Educacion'])['Adolescentes'].mean()
educacion.sort_values(ascending=False)
```

[94]:
```
Educacion
PhD          0.598765
Graduado     0.494232
Master       0.490401
Basico       0.092593
Name: Adolescentes, dtype: float64
```

Estado civil con distintas variables

[95]:
```
# Agrupar por variables cualitativas y calcular la media de ingresos
#ordenado descendente
civil2= datos4.groupby(['Estadocivil'])['Ingresos'].mean()
civil2.sort_values(ascending=False)
```

[95]:
```
Estadocivil
Viudo            56426.561706
Divorciado       52834.228448
Parejadehecho    52174.143122
Casado           51729.210370
Soltero          51051.329080
Name: Ingresos, dtype: float64
```

[96]:
```
# Agrupar por variables cualitativas y calcular la media de respuesta
civil=datos4.groupby(['Estadocivil'])['Respuesta'].mean()
civil.sort_values(ascending=False)
```

[96]:
```
Estadocivil
Viudo            0.246753
Soltero          0.223819
Divorciado       0.206897
Casado           0.113426
Parejadehecho    0.103448
Name: Respuesta, dtype: float64
```

[97]:
```
# ordenado descendente
educacion=datos4.groupby(['Estadocivil'])['Niños'].mean()
educacion.sort_values(ascending=False)
```

[97]:
```
Estadocivil
Soltero          0.464066
Casado           0.456019
Parejadehecho    0.450000
Divorciado       0.413793
Viudo            0.233766
Name: Niños, dtype: float64
```

[98]:
```
datos4.groupby(['Niños'])['Estadocivil'].apply(lambda x: x.mode())
```

[98]:
```
Niños
0      0    Casado
1      0    Casado
2      0    Casado
Name: Estadocivil, dtype: object
```

[99]:
```
# ordenado descendente
educacion=datos4.groupby(['Estadocivil'])['Adolescentes'].mean()
```

```
educacion.sort_values(ascending=False)
```

[99]:
```
Estadocivil
Viudo            0.636364
Divorciado       0.590517
Parejadehecho    0.529310
Casado           0.511574
Soltero          0.408624
Name: Adolescentes, dtype: float64
```

[100]:
```
datos4.groupby(['Adolescentes'])['Estadocivil'].apply(lambda x: x.mode())
```

[100]:
```
Adolescentes
0             0    Casado
1             0    Casado
2             0    Casado
Name: Estadocivil, dtype: object
```

País con distintas variables

[101]:
```
# Agrupar por variables cualitativas y calcular la moda de respuesta
datos4.groupby(['Pais'])['Respuesta'].apply(lambda x: x.mode())
```

[101]:
```
Pais
AUS   0    0
CA    0    0
GER   0    0
IND   0    0
ME    0    1
SA    0    0
SP    0    0
US    0    0
Name: Respuesta, dtype: int64
```

[102]:
```
# Agrupar por variables cualitativas y calcular la media de respuesta
pais= datos4.groupby(['Pais'])['Respuesta'].mean()
pais.sort_values(ascending=False)
```

[102]:
```
Pais
ME     0.666667
SP     0.160731
SA     0.154303
AUS    0.143750
CA     0.141791
GER    0.141667
US     0.119266
IND    0.087838
Name: Respuesta, dtype: float64
```

[103]:
```
# Agrupar por variables cualitativas y calcular la media de ingresos
pais= datos4.groupby(['Pais'])['Ingresos'].mean()
pais.sort_values(ascending=False)
```

[103]:
```
Pais
ME     57680.333333
US     53200.555071
CA     53044.621279
SA     53007.614989
GER    52927.633378
AUS    51840.276672
SP     51565.824203
IND    49038.244942
Name: Ingresos, dtype: float64
```

[104]:
```
datos4.groupby(['Adolescentes'])['Pais'].apply(lambda x: x.mode())
```

[104]:
```
Adolescentes
0              0    SP
1              0    SP
2              0    SP
Name: Pais, dtype: object
```

[105]:
```
# ordenado descendente
educacion=datos4.groupby(['Pais'])['Adolescentes'].mean()
educacion.sort_values(ascending=False)
```

[105]:
```
Pais
ME     0.666667
CA     0.555970
AUS    0.518750
US     0.513761
SP     0.507763
IND    0.493243
SA     0.477745
GER    0.450000
Name: Adolescentes, dtype: float64
```

[106]:
```
datos4.groupby(['Niños'])['Pais'].apply(lambda x: x.mode())
```

[106]:
```
Niños
0      0    SP
1      0    SP
2      0    SP
Name: Pais, dtype: object
```

[107]:
```
educacion=datos4.groupby(['Pais'])['Niños'].mean()
educacion.sort_values(ascending=False)
```

[107]:
```
Pais
IND    0.520270
AUS    0.518750
SA     0.486647
SP     0.426484
CA     0.417910
US     0.403670
GER    0.400000
ME     0.000000
Name: Niños, dtype: float64
```

Años con otras variables

[108]:
```
# Agrupar por variables cualitativas y calcular la media de respuesta
años= datos4.groupby(['años'])['Respuesta'].mean()
años.sort_values(ascending=False)
```

[108]:
```
años
16    1.000000
17    0.500000
70    0.444444
28    0.342857
18    0.333333
30    0.320000
66    0.285714
64    0.250000
19    0.250000
22    0.230769
23    0.227273
68    0.222222
57    0.211538
26    0.205882
32    0.204545
42    0.202247
29    0.190476
60    0.187500
40    0.173913
43    0.169014
63    0.166667
69    0.166667
27    0.162791
47    0.160000
36    0.158730
39    0.157895
33    0.157895
52    0.155556
55    0.152542
20    0.142857
```

```
46    0.142857
54    0.137255
35    0.134328
51    0.133333
41    0.131868
50    0.130435
44    0.126761
38    0.125000
37    0.121951
58    0.117647
56    0.113636
31    0.111111
25    0.111111
48    0.098361
59    0.097561
49    0.096154
45    0.094340
61    0.083333
62    0.080000
21    0.076923
65    0.055556
53    0.055556
34    0.035088
67    0.000000
24    0.000000
71    0.000000
72    0.000000
73    0.000000
Name: Respuesta, dtype: float64
```

[109]:
```
datos4['años'].sort_values(ascending=False)
```

[109]:
```
1950    73
424     72
415     71
358     70
2084    70
        ..
1170    18
914     18
2213    17
1850    17
46      16
Name: años, Length: 2240, dtype: int64
```

[110]:
```
#Agrupar por variables cualitativas y calcular la media de ingresos
años= datos4.groupby(['años'])['Ingresos'].mean()
```

```
años.sort_values(ascending=False)
```

[110]:
```
años
72    93027.000000
17    81937.000000
71    75865.000000
66    68562.095238
69    67986.833333
68    66899.222222
20    65593.000000
18    63545.000000
70    63006.250150
67    62051.583333
52    61926.627808
56    61897.886364
60    61267.500000
64    59496.843750
65    59302.555556
22    58952.692308
51    58890.555556
58    58584.181399
62    56562.480000
55    56475.546633
54    56311.656916
59    55141.981740
61    55086.000000
19    55013.750000
48    54019.836066
49    53628.807692
45    53296.301887
63    52779.541712
50    52544.358755
57    52007.177911
46    51766.918367
41    51666.934066
47    51343.260000
73    51141.000000
42    50902.792165
44    50822.778188
30    50686.585027
37    50578.207317
36    50559.591291
40    49490.887701
43    48621.387362
39    48487.899147
33    48423.236842
53    48110.444444
```

```
38    48015.579545
27    47937.284915
35    47862.791045
31    47582.894475
25    46783.962963
34    46289.004410
21    45378.538462
23    44107.454545
29    43123.571429
32    42161.619349
24    42132.718806
28    41972.342857
26    39088.823529
16     7500.000000
Name: Ingresos, dtype: float64
```

[111]:
```
ordenar= datos4.groupby(['años'])['Ingresos'].apply(lambda x: x.mode())
ordenar.sort_values(ascending=False)
```

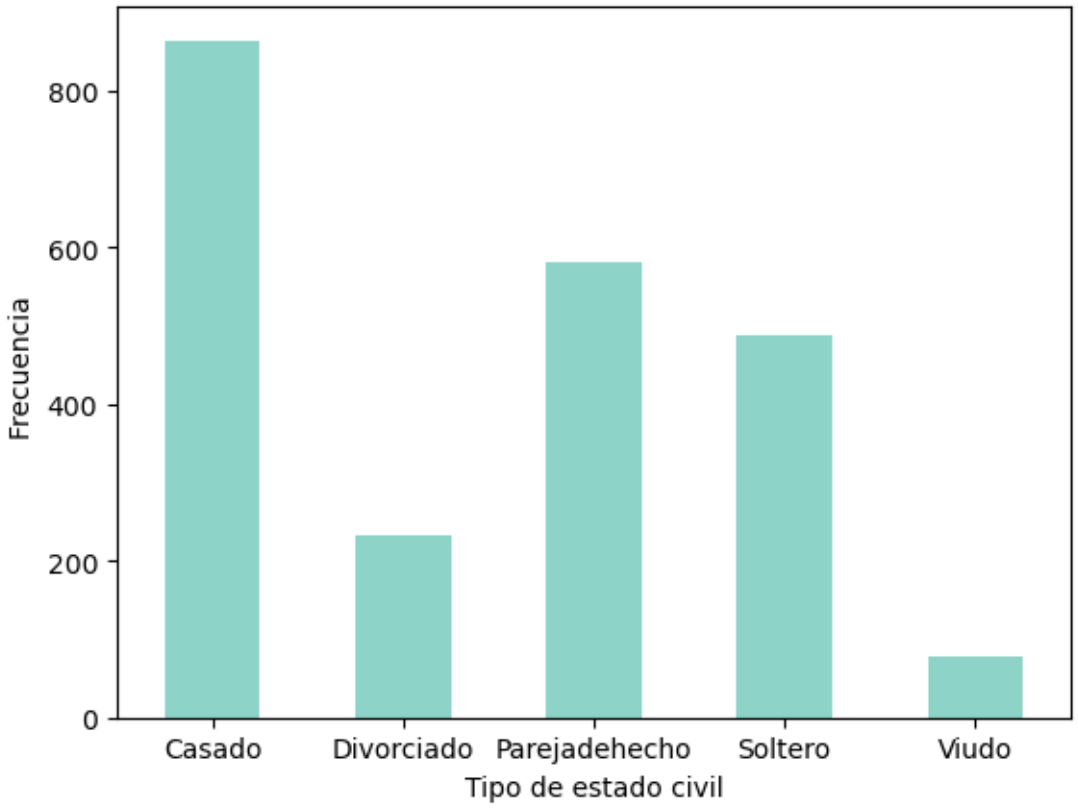
[111]:
```
años
69    5    113734.0
54    5     98777.0
18    2     95529.0
60    2     94384.0
72    0     93027.0
             …
36    0      7500.0
34    0      7500.0
27    0      7500.0
20    0      7500.0
16    0      7500.0
Name: Ingresos, Length: 250, dtype: float64
```

[ ]:
```

```

ANÁLISIS DE LAS VARIABLES

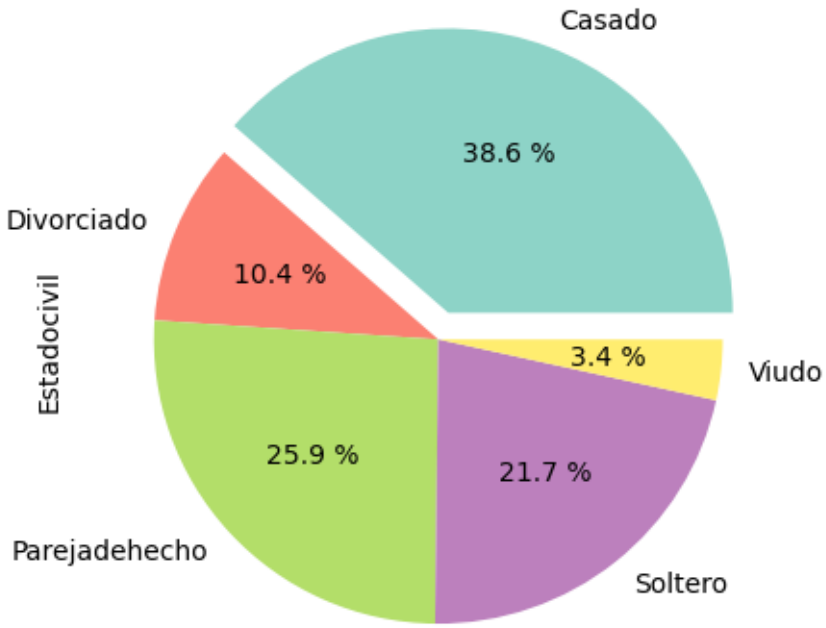
[112]:
```
# Representación en barras de la frecuencia de las instancias de la variable␣
 ↪estado civil
datos4['Estadocivil'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Tipo de estado civil')
plt.ylabel('Frecuencia ')
```

[112]:
```
Text(0, 0.5, 'Frecuencia ')
```



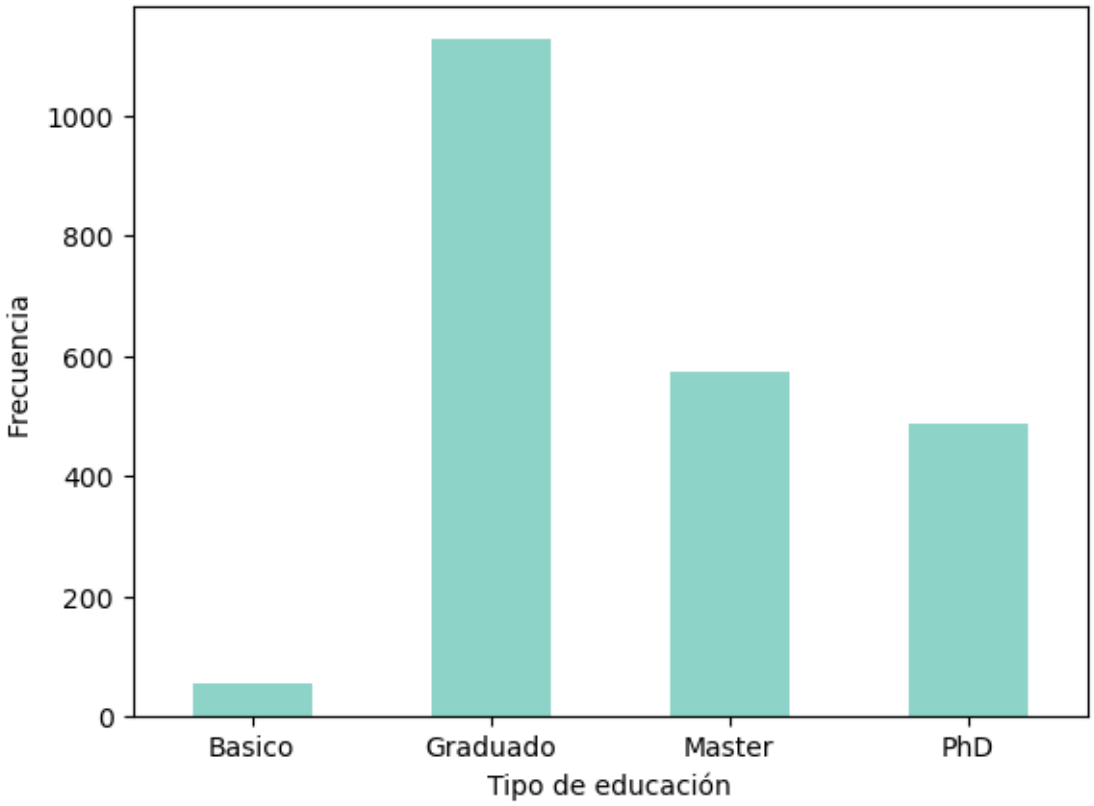
[113]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0.1, 0, 0, 0, 0)
datos4['Estadocivil'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[113]:
```
<AxesSubplot:ylabel='Estadocivil'>
```



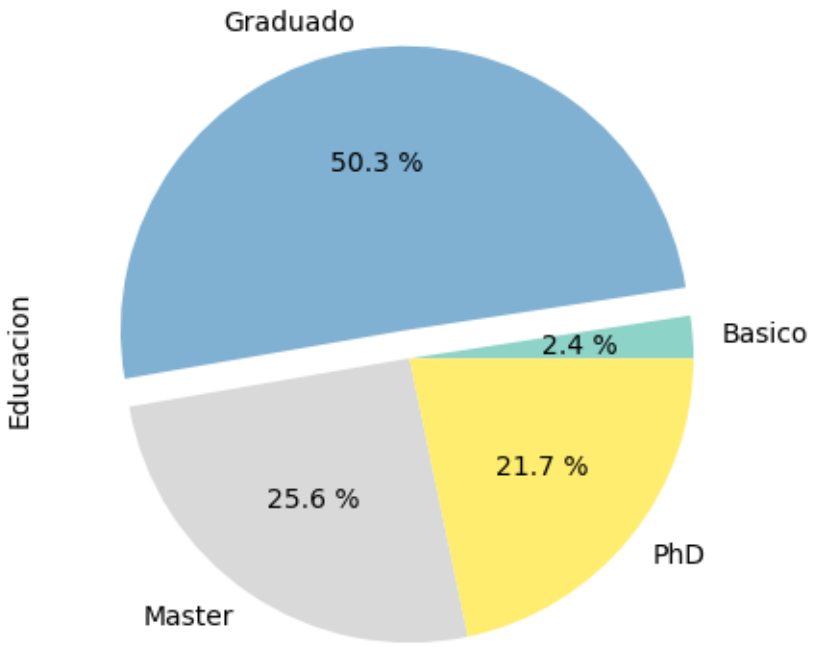
[114]:
```
# Representación en barras de la frecuencia de las instancias de la variable
 ↪educación
datos4['Educacion'].value_counts(sort=False).sort_index().plot.bar(rot=0,
 ↪cmap='Set3')
plt.xlabel('Tipo de educación')
plt.ylabel('Frecuencia ')
```

[114]:
```
Text(0, 0.5, 'Frecuencia ')
```



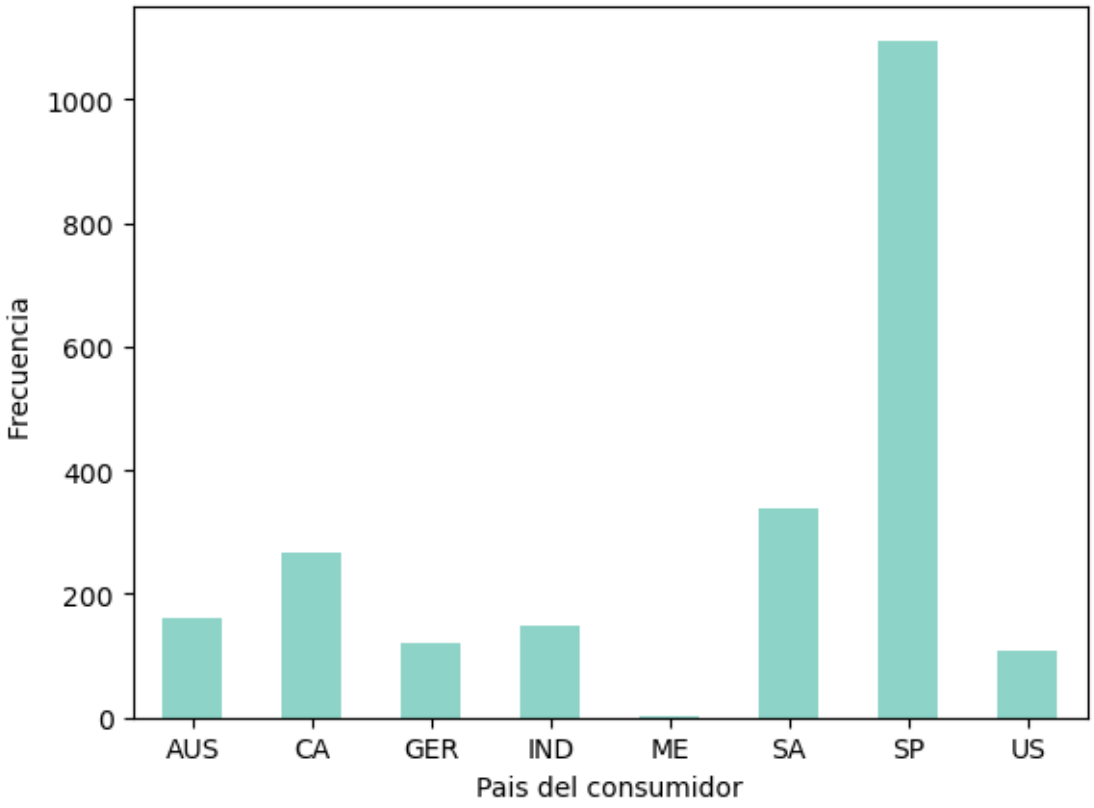
[115]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0, 0.1, 0, 0)
datos4['Educacion'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[115]: <AxesSubplot:ylabel='Educacion'>



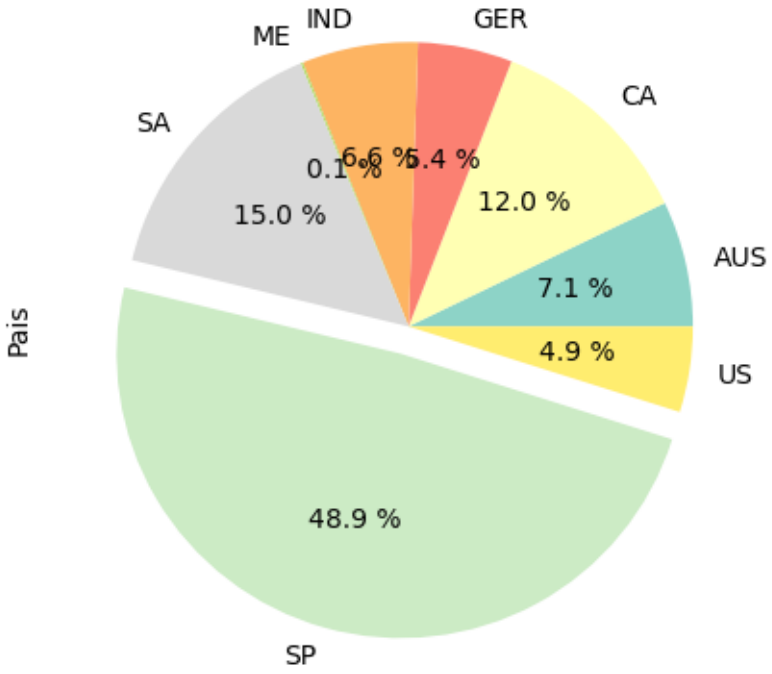
[116]:
```
# Representación en barras de la frecuencia de las instancias de la variable␣
 ↪pais
datos4['Pais'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Pais del consumidor')
plt.ylabel('Frecuencia ')
```

[116]:
```
Text(0, 0.5, 'Frecuencia ')
```



[117]:
```
# Igual que el anterior pero en porcentaje y en gráfico circular
desfase = (0, 0, 0, 0,0,0,0.1,0)
datos4['Pais'].value_counts(sort=False).sort_index().plot.pie(rot=0,␣
 ↪cmap='Set3', autopct="%0.1f %%", explode=desfase)
```

[117]:
```
<AxesSubplot:ylabel='Pais'>
```



ANÁLISIS CORRELACIÓN

[118]:
```
# Creación de la matriz de correlación de todas las variables
correlacion=datos4.corr()
datos4.corr()
```

[118]:

	Identificador	Cumpleaños	Ingresos	Niños	\
Identificador	1.000000	0.003024	0.000086	0.002406	
Cumpleaños	0.003024	1.000000	-0.198733	0.234004	
Ingresos	0.000086	-0.198733	1.000000	-0.510551	
Niños	0.002406	0.234004	-0.510551	1.000000	
Adolescentes	-0.002580	-0.363147	0.034163	-0.036133	
Ultimacompra	-0.046524	-0.019651	0.006941	0.008827	
Totalvinos	-0.022878	-0.162916	0.685887	-0.496297	
Totalfrutas	0.004600	-0.013715	0.505459	-0.372581	
Totalcarnes	-0.004437	-0.030897	0.684299	-0.437129	
Totalpescados	-0.024475	-0.042499	0.518656	-0.387644	
Totaldulces	-0.007642	-0.019565	0.518260	-0.370673	
Totallujos	-0.013438	-0.057427	0.384477	-0.349595	
Comprasdescuentos	-0.037166	-0.067985	-0.107365	0.221798	
Comprasweb	-0.018924	-0.153918	0.450124	-0.361647	
Comprascatalogo	-0.003440	-0.125371	0.693565	-0.502237	
Comprastiendas	-0.014927	-0.139382	0.626823	-0.499683	
Visitaswebmes	-0.007446	0.117469	-0.646725	0.447846	

```
Campaña3              -0.036040    0.061007 -0.015320  0.014674
Campaña4              -0.025387   -0.064340  0.217443 -0.161600
Campaña5              -0.007517    0.015289  0.395256 -0.205634
Campaña1              -0.021614   -0.008229  0.324927 -0.172339
Campaña2              -0.015061   -0.007658  0.103884 -0.081716
Quejas                 0.033883   -0.004334 -0.030595  0.040207
Zcostecontacto              NaN         NaN       NaN       NaN
Zingresos                   NaN         NaN       NaN       NaN
Respuesta             -0.021968    0.018417  0.160647 -0.080008
añocompra             -0.006414   -0.024984  0.026687  0.053339
años                  -0.003389   -0.998297  0.199663 -0.230151
Totalgasto            -0.018054   -0.113710  0.789064 -0.556669
Totalhijos            -0.000146   -0.095451 -0.340882  0.689971

                    Adolescentes  Ultimacompra  Totalvinos  Totalfrutas  \
Identificador         -0.002580     -0.046524   -0.022878     0.004600
Cumpleaños            -0.363147     -0.019651   -0.162916    -0.013715
Ingresos               0.034163      0.006941    0.685887     0.505459
Niños                 -0.036133      0.008827   -0.496297    -0.372581
Adolescentes           1.000000      0.016198    0.004846    -0.176764
Ultimacompra           0.016198      1.000000    0.016064    -0.004306
Totalvinos             0.004846      0.016064    1.000000     0.389637
Totalfrutas           -0.176764     -0.004306    0.389637     1.000000
Totalcarnes           -0.261160      0.023056    0.562667     0.543105
Totalpescados         -0.204187      0.001079    0.399753     0.594804
Totaldulces           -0.162475      0.022670    0.386581     0.567164
Totallujos            -0.021725      0.016693    0.387516     0.392995
Comprasdescuentos      0.387741     -0.001098    0.010940    -0.132114
Comprasweb             0.155500     -0.010726    0.542265     0.296735
Comprascatalogo       -0.110769      0.025110    0.635226     0.487917
Comprastiendas         0.050695      0.000799    0.642100     0.461758
Visitaswebmes          0.134884     -0.021445   -0.320653    -0.418383
Campaña3              -0.042677     -0.032991    0.062202     0.014727
Campaña4               0.038886      0.018826    0.373286     0.010152
Campaña5              -0.191050      0.000129    0.472613     0.215833
Campaña1              -0.140090     -0.019283    0.354133     0.194748
Campaña2              -0.015605     -0.001781    0.205907    -0.009773
Quejas                 0.003138      0.013231   -0.039007    -0.005166
Zcostecontacto              NaN           NaN         NaN          NaN
Zingresos                   NaN           NaN         NaN          NaN
Respuesta             -0.154446     -0.198437    0.247254     0.125289
añocompra             -0.008260     -0.026084   -0.154188    -0.055150
años                   0.361517      0.018067    0.153404     0.010453
Totalgasto            -0.138384      0.020433    0.891839     0.614229
Totalhijos             0.698433      0.018053   -0.351909    -0.394853

                    Totalcarnes  Totalpescados  …  Campaña1  Campaña2  \
```

Identificador	-0.004437	-0.024475	…	-0.021614	-0.015061
Cumpleaños	-0.030897	-0.042499	…	-0.008229	-0.007658
Ingresos	0.684299	0.518656	…	0.324927	0.103884
Niños	-0.437129	-0.387644	…	-0.172339	-0.081716
Adolescentes	-0.261160	-0.204187	…	-0.140090	-0.015605
Ultimacompra	0.023056	0.001079	…	-0.019283	-0.001781
Totalvinos	0.562667	0.399753	…	0.354133	0.205907
Totalfrutas	0.543105	0.594804	…	0.194748	-0.009773
Totalcarnes	1.000000	0.568402	…	0.309761	0.043033
Totalpescados	0.568402	1.000000	…	0.260762	0.002577
Totaldulces	0.523846	0.579870	…	0.241818	0.009985
Totallujos	0.350609	0.422875	…	0.166396	0.049990
Comprasdescuentos	-0.122415	-0.139361	…	-0.123244	-0.037695
Comprasweb	0.293761	0.293681	…	0.155143	0.034188
Comprascatalogo	0.723827	0.534478	…	0.308097	0.099852
Comprastiendas	0.479659	0.459855	…	0.183249	0.085189
Visitaswebmes	-0.539470	-0.446003	…	-0.192502	-0.007196
Campaña3	0.018272	0.000357	…	0.094751	0.072020
Campaña4	0.102912	0.016843	…	0.251300	0.292210
Campaña5	0.373769	0.199578	…	0.403078	0.221533
Campaña1	0.309761	0.260762	…	1.000000	0.175315
Campaña2	0.043033	0.002577	…	0.175315	1.000000
Quejas	-0.023483	-0.020953	…	-0.025499	-0.011334
Zcostecontacto	NaN	NaN	…	NaN	NaN
Zingresos	NaN	NaN	…	NaN	NaN
Respuesta	0.236335	0.111331	…	0.293982	0.169293
añocompra	-0.082472	-0.067611	…	0.037101	0.000887
años	0.025987	0.038419	…	0.010368	0.007685
Totalgasto	0.842965	0.642818	…	0.381523	0.135813
Totalhijos	-0.502208	-0.425503	…	-0.224887	-0.069823

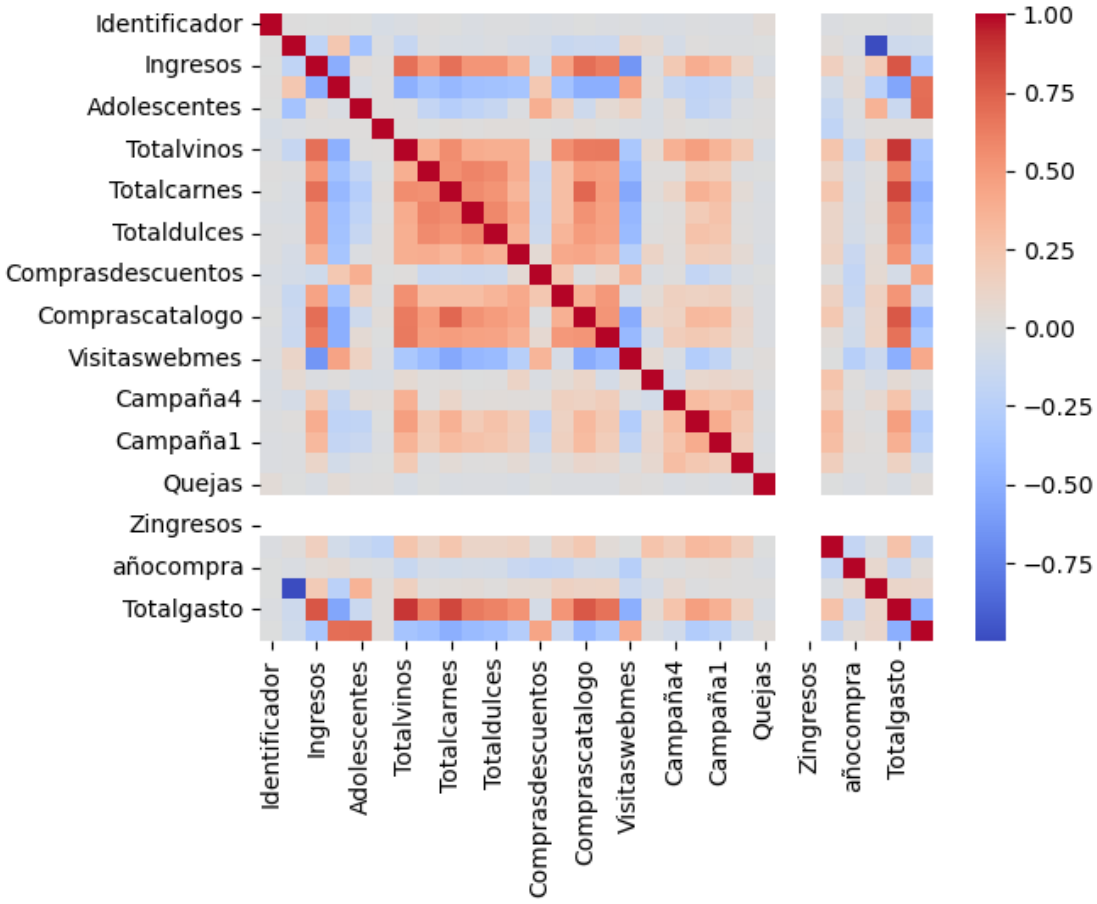
	Quejas	Zcostecontacto	Zingresos	Respuesta	añocompra	\
Identificador	0.033883	NaN	NaN	-0.021968	-0.006414	
Cumpleaños	-0.004334	NaN	NaN	0.018417	-0.024984	
Ingresos	-0.030595	NaN	NaN	0.160647	0.026687	
Niños	0.040207	NaN	NaN	-0.080008	0.053339	
Adolescentes	0.003138	NaN	NaN	-0.154446	-0.008260	
Ultimacompra	0.013231	NaN	NaN	-0.198437	-0.026084	
Totalvinos	-0.039007	NaN	NaN	0.247254	-0.154188	
Totalfrutas	-0.005166	NaN	NaN	0.125289	-0.055150	
Totalcarnes	-0.023483	NaN	NaN	0.236335	-0.082472	
Totalpescados	-0.020953	NaN	NaN	0.111331	-0.067611	
Totaldulces	-0.022485	NaN	NaN	0.117372	-0.073366	
Totallujos	-0.030861	NaN	NaN	0.139850	-0.142530	
Comprasdescuentos	0.000420	NaN	NaN	0.002238	-0.186210	
Comprasweb	-0.016310	NaN	NaN	0.148730	-0.167257	
Comprascatalogo	-0.020453	NaN	NaN	0.220810	-0.085421	

Comprastiendas	-0.016524	NaN	NaN	0.039363	-0.097291
Visitaswebmes	0.019769	NaN	NaN	-0.003987	-0.249489
Campaña3	0.008415	NaN	NaN	0.254258	0.011088
Campaña4	-0.027611	NaN	NaN	0.177019	-0.011664
Campaña5	-0.009419	NaN	NaN	0.326634	0.021133
Campaña1	-0.025499	NaN	NaN	0.293982	0.037101
Campaña2	-0.011334	NaN	NaN	0.169293	0.000887
Quejas	1.000000	NaN	NaN	-0.001707	-0.024304
Zcostecontacto	NaN	NaN	NaN	NaN	NaN
Zingresos	NaN	NaN	NaN	NaN	NaN
Respuesta	-0.001707	NaN	NaN	1.000000	-0.171030
añocompra	-0.024304	NaN	NaN	-0.171030	1.000000
años	0.002902	NaN	NaN	-0.028339	0.083259
Totalgasto	-0.037058	NaN	NaN	0.265298	-0.144235
Totalhijos	0.031066	NaN	NaN	-0.169163	0.032215

	años	Totalgasto	Totalhijos
Identificador	-0.003389	-0.018054	-0.000146
Cumpleaños	-0.998297	-0.113710	-0.095451
Ingresos	0.199663	0.789064	-0.340882
Niños	-0.230151	-0.556669	0.689971
Adolescentes	0.361517	-0.138384	0.698433
Ultimacompra	0.018067	0.020433	0.018053
Totalvinos	0.153404	0.891839	-0.351909
Totalfrutas	0.010453	0.614229	-0.394853
Totalcarnes	0.025987	0.842965	-0.502208
Totalpescados	0.038419	0.642818	-0.425503
Totaldulces	0.015222	0.603016	-0.383137
Totallujos	0.048928	0.524262	-0.266095
Comprasdescuentos	0.056904	-0.065112	0.439684
Comprasweb	0.143672	0.519837	-0.146361
Comprascatalogo	0.119990	0.778577	-0.439904
Comprastiendas	0.133264	0.674669	-0.321125
Visitaswebmes	-0.131656	-0.500218	0.418419
Campaña3	-0.060167	0.053385	-0.020402
Campaña4	0.063456	0.253290	-0.087563
Campaña5	-0.014008	0.470058	-0.285642
Campaña1	0.010368	0.381523	-0.224887
Campaña2	0.007685	0.135813	-0.069823
Quejas	0.002902	-0.037058	0.031066
Zcostecontacto	NaN	NaN	NaN
Zingresos	NaN	NaN	NaN
Respuesta	-0.028339	0.265298	-0.169163
añocompra	0.083259	-0.144235	0.032215
años	1.000000	0.104934	0.097029
Totalgasto	0.104934	1.000000	-0.498888
Totalhijos	0.097029	-0.498888	1.000000

[30 rows x 30 columns]

[119]:
```
# Representación gráfica de la matriz de correlación
sns.heatmap(correlacion, annot=False, cmap='coolwarm')
plt.show()
```



[120]:
```
#hay dos datos: z_revenue(Zingresos) y z_costcontact(Zcostecontacto), que solo␣
 ↪tiene el mismo resultado.
#No se sabe a que hace referencia.
datos4['Zingresos']
datos4['Zcostecontacto']
```

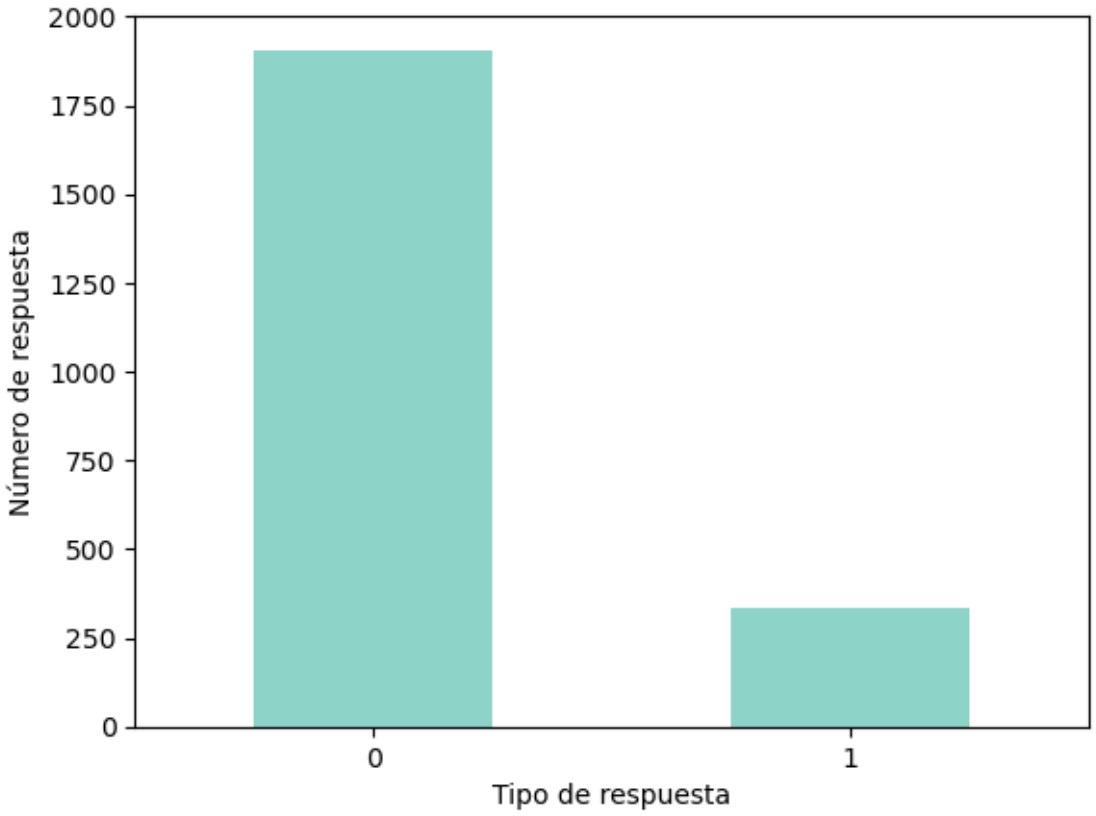
[120]:
```
0       3
1       3
2       3
3       3
4       3
```

```
       ..
2235    3
2236    3
2237    3
2238    3
2239    3
Name: Zcostecontacto, Length: 2240, dtype: int64
```

0.7 REALIZACIÓN DE MACHINE LEARNING

[121]:
```
# Análisis de la variable salida para ver si está balanceado o no
datos4['Respuesta'].value_counts(sort=False).sort_index().plot.bar(rot=0,␣
 ↪cmap='Set3')
plt.xlabel('Tipo de respuesta')
plt.ylabel('Número de respuesta')
```

[121]:
```
Text(0, 0.5, 'Número de respuesta')
```



[122]:
```
# Apertura del archivo del conjunto de datos ya limpiado
datos5= pd.read_csv("ARCHIVOTRATADO.csv", sep=";")
```

[123]:
```
# Codificar las variables categóricas con la técnica one-hot para volverlas␣
 ↪dummies
data = pd.get_dummies(datos5, columns=['Educacion','Estadocivil','Pais',␣
 ↪'Inscripcion'])

# Separar las variables predictoras y la variable objetivo
X = data.drop('Respuesta', axis=1)
y = data['Respuesta']

# Aplicar la técnica SMOTE para balancear las clases
smote = SMOTE()
X_resampled, y_resampled = smote.fit_resample(X, y)

# Dividir en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled,␣
 ↪test_size=0.25, random_state=42)

#Para saber como está estratificada
print("Tamaño del conjunto x_train :", X_train.shape)
print("Tamaño del conjunto y_train :", y_train.shape)
print("Tamaño del conjunto x_test :", X_test.shape)
print("Tamaño del conjunto y_test :", y_test.shape)
```

```
Tamaño del conjunto x_train : (2859, 706)
Tamaño del conjunto y_train : (2859,)
Tamaño del conjunto x_test : (953, 706)
Tamaño del conjunto y_test : (953,)
```

0.8 ARBOL DE DECISIÓN

(Clasificación: si el objetivo es predecir si un cliente realizará una compra o no, se puede aplicar un algoritmo de clasificación, como árboles de decisión, regresión logística o SVM. Agrupamiento: si se desea identificar patrones en los datos sin tener una variable objetivo definida, se puede aplicar un algoritmo de agrupamiento, como k-means o clustering jerárquico.)

[124]:
```
#Creamos primero la estructura del árbol
#y la entrenamos con los conjuntos de entrenamiento de las variables␣
 ↪predictoras y la respuesta.

arb = tree.DecisionTreeClassifier()
arbol = arb.fit(X_train, y_train)
```

Matriz de Confusión

[125]:
```
#Realizamos la predicción
y_pred = arbol.predict(X_test)

# Evaluamos el modelo
```

```
print("% de acierto en el conjunto de entrenamiento: ", arbol.score(X_train,␣
 ↪y_train))
print("% de acierto en el conjunto de test: ", arbol.score(X_test, y_test))

#Calcular el accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Precisión sin poda:", accuracy)
```

```
% de acierto en el conjunto de entrenamiento:  1.0
% de acierto en el conjunto de test:  0.8646379853095488
Precisión sin poda: 0.8646379853095488
```

[126]:
```
# creación de la matriz de confusión
metrics.confusion_matrix(y_test, y_pred)
```

[126]:
```
array([[408,  71],
       [ 58, 416]], dtype=int64)
```

[127]:
```
# Cálculo de todas las métricas
classReport = metrics.classification_report(y_test, y_pred)
print(classReport)
```

```
              precision    recall  f1-score   support

           0       0.88      0.85      0.86       479
           1       0.85      0.88      0.87       474

    accuracy                           0.86       953
   macro avg       0.86      0.86      0.86       953
weighted avg       0.86      0.86      0.86       953
```

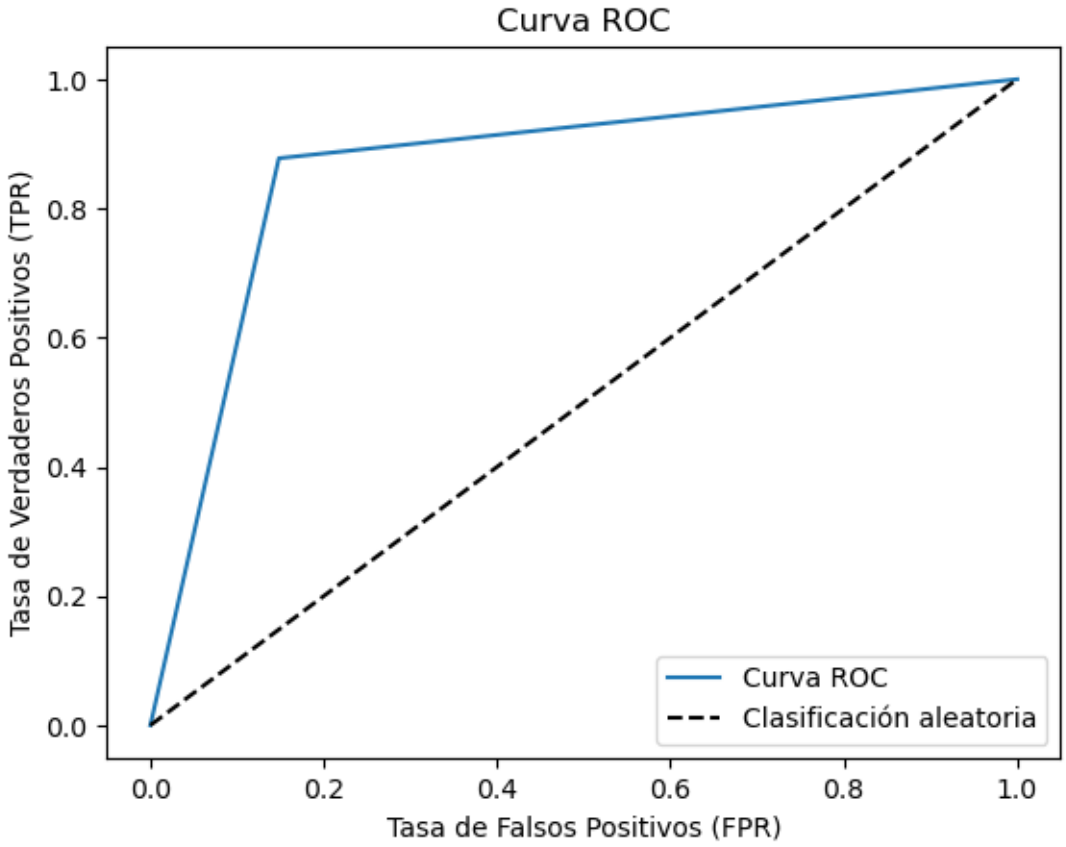
AUC

[128]:
```
print(f'AUC del modelo Árbol: {roc_auc_score(y_test, y_pred)}')
#En la práctica, un valor de AUC de alrededor de 0.7 a 0.8 se considera␣
 ↪aceptable en muchos problemas,
#mientras que un valor superior a 0.9 se considera excelente. Sin embargo, el␣
 ↪valor óptimo de AUC puede variar dependiendo
#del contexto del problema y de las implicaciones prácticas de la clasificación␣
 ↪incorrecta de las instancias.
```

```
AUC del modelo Árbol: 0.8647058305365432
```

[129]:
```
# Para la creación de la curva roc de forma gráfica
fpr, tpr, thresholds = roc_curve(y_test, y_pred)

# Graficar la curva ROC
plt.plot(fpr, tpr, label='Curva ROC')
```

```
plt.plot([0, 1], [0, 1], 'k--', label='Clasificación aleatoria')
plt.xlabel('Tasa de Falsos Positivos (FPR)')
plt.ylabel('Tasa de Verdaderos Positivos (TPR)')
plt.title('Curva ROC')
plt.legend()
plt.show()
```



[130]:
```
# Inicio del código para la creación del árbol de decisión de forma gráfica
import os

# Obtener el valor actual de la variable de entorno PATH
current_path = os.environ.get('PATH', '')

# Agregar la ruta de Graphviz al PATH
graphviz_path = r'C:\Program Files\Graphviz\bin'
new_path = f'{current_path};{graphviz_path}'

# Actualización de la variable de entorno PATH
os.environ['PATH'] = new_path
```

[131]:
```
# Código en sí de la creación del árbol de decisión

# Obtener los nombres de las características y las clases
feature_names = data.columns[:-1].tolist()  # Para obtener los nombres de las␣
 ↪características (excepto la última)
target_names = data.iloc[:, -1].astype(str).unique().tolist()

# Crear la representación del árbol de decisión en formato DOT
dot_data = tree.export_graphviz(arbol, out_file=None,
                                feature_names=feature_names,
                                class_names=target_names,
                                filled=True, rounded=True,
                                special_characters=True)

# Generar el gráfico del árbol de decisión usando la libreria Graphviz
graph = graphviz.Source(dot_data)
graph.render("mkt_decision_tree")  #para guardarlo

# Visualizar el gráfico
graph.view()
```

[131]:
```
'mkt_decision_tree.pdf'
```

Poda con cross validation

[132]:
```
#Comprobación del accuracy del modelo por si al realizar una técnica de poda␣
 ↪arroja un mejor modelo que sin ella

#primero creamos la estructura del arbol
arb2 = tree.DecisionTreeClassifier()

# Realizar la poda de Cross Validation
scores = cross_val_score(arb2, X, y, cv=5)  # cv indica el número de divisiones␣
 ↪en Cross Validation

# Imprimir los puntajes de precisión de cada división
print("Precisión por división:", scores)

# Ajustar el modelo utilizando los datos de entrenamiento
arbol2 = arb2.fit(X_train, y_train)
#Realizamos la predicción
y_pred = arbol2.predict(X_test)

# Evaluamos el modelo
print("% de acierto en el conjunto de entrenamiento: ", arbol2.score(X_train,␣
 ↪y_train))
print("% de acierto en el conjunto de test: ", arbol2.score(X_test, y_test))
```

```
#Calcular el accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Precisión con poda:", accuracy)
```

```
Precisión por división: [0.84375    0.84151786 0.82589286 0.85714286 0.17633929]
% de acierto en el conjunto de entrenamiento:  1.0
% de acierto en el conjunto de test:  0.8614900314795383
Precisión con poda: 0.8614900314795383
```

[133]:
```
# creación de la matriz de confusión
metrics.confusion_matrix(y_test, y_pred)
```

[133]:
```
array([[399,  80],
       [ 52, 422]], dtype=int64)
```

[134]:
```
# Cálculo de todas las métricas
classReport = metrics.classification_report(y_test, y_pred)
print(classReport)
```

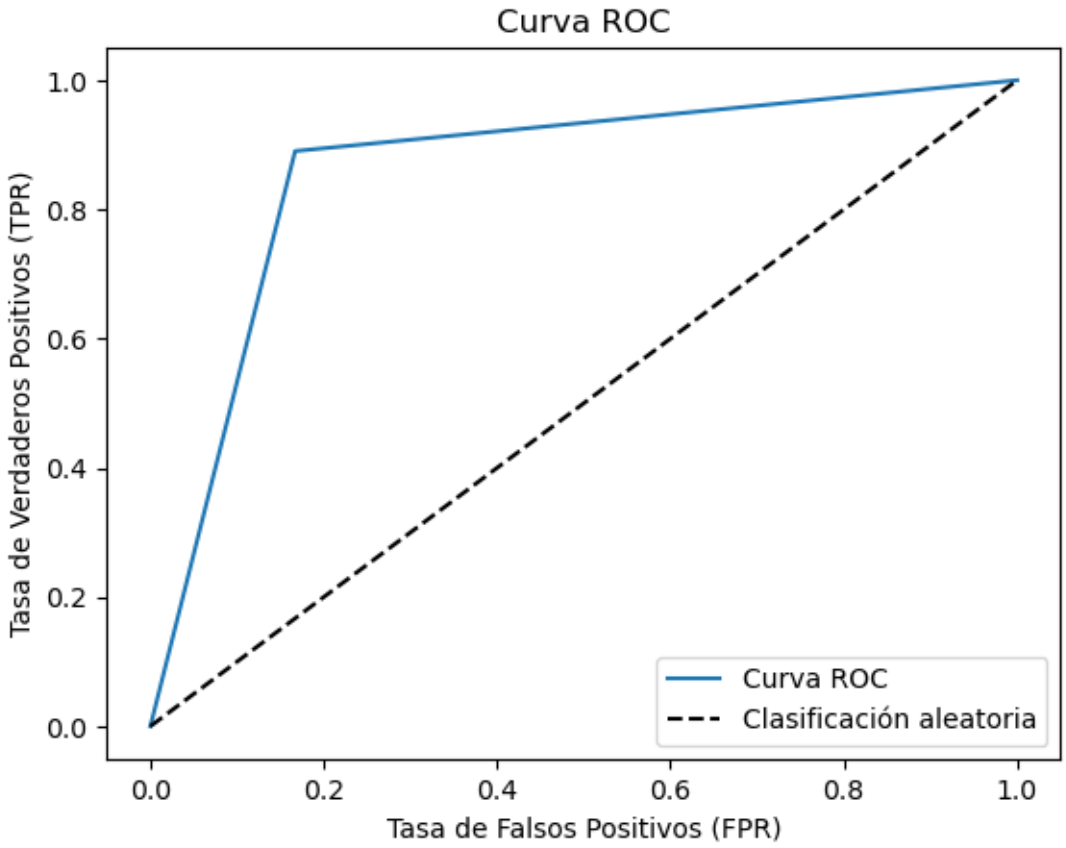
```
              precision    recall  f1-score   support

           0       0.88      0.83      0.86       479
           1       0.84      0.89      0.86       474

    accuracy                           0.86       953
   macro avg       0.86      0.86      0.86       953
weighted avg       0.86      0.86      0.86       953
```

[135]:
```
print(f'AUC del modelo Árbol: {roc_auc_score(y_test, y_pred)}')
#En la práctica, un valor de AUC de alrededor de 0.7 a 0.8 se considera␣
 ↪aceptable en muchos problemas,
#mientras que un valor superior a 0.9 se considera excelente. Sin embargo, el␣
 ↪valor óptimo de AUC puede variar dependiendo
#del contexto del problema y de las implicaciones prácticas de la clasificación␣
 ↪incorrecta de las instancias.
```

```
AUC del modelo Árbol: 0.8616403724355417
```

[136]:
```
# Creación de la gráfica curva roc para un arbol podado
fpr, tpr, thresholds = roc_curve(y_test, y_pred)

# Graficar la curva ROC
plt.plot(fpr, tpr, label='Curva ROC')
plt.plot([0, 1], [0, 1], 'k--', label='Clasificación aleatoria')
plt.xlabel('Tasa de Falsos Positivos (FPR)')
plt.ylabel('Tasa de Verdaderos Positivos (TPR)')
plt.title('Curva ROC')
plt.legend()
```

```
plt.show()
```



[137]:
```
# Creación del arbol de decisión y sus pasos para guardarlo

# Obtemer los nombres de las características y las clases
feature_names = data.columns[:-1].tolist()  # Obtener los nombres de las␣
 ↪características (excepto la última)
target_names = data.iloc[:, -1].astype(str).unique().tolist()

# Crear la representación del árbol de decisión en formato DOT
dot_data = tree.export_graphviz(arbol2, out_file=None,
                                feature_names=feature_names,
                                class_names=target_names,
                                filled=True, rounded=True,
                                special_characters=True)

# Generar el gráfico del árbol de decisión usando Graphviz
graph = graphviz.Source(dot_data)
graph.render("mkt_decision_tree_crossvalidation")  # Guardado del árbol
```

```
# Visualizar el gráfico
graph.view()
```

[137]: 'mkt_decision_tree_crossvalidation.pdf'

KNN

[138]:
```
#Creación de un knn por el simple hecho de comprobar que modelo daba un mejor
 ↪ajuste
k = 3
knn = KNeighborsClassifier(n_neighbors=k)

# Entrenar el clasificador
knn.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba
y_pred = knn.predict(X_test)

# Calcular la precisión del modelo
accuracy = accuracy_score(y_test, y_pred)
print("Precisión del modelo:", accuracy)
```

```
Precisión del modelo: 0.7848898216159497
```

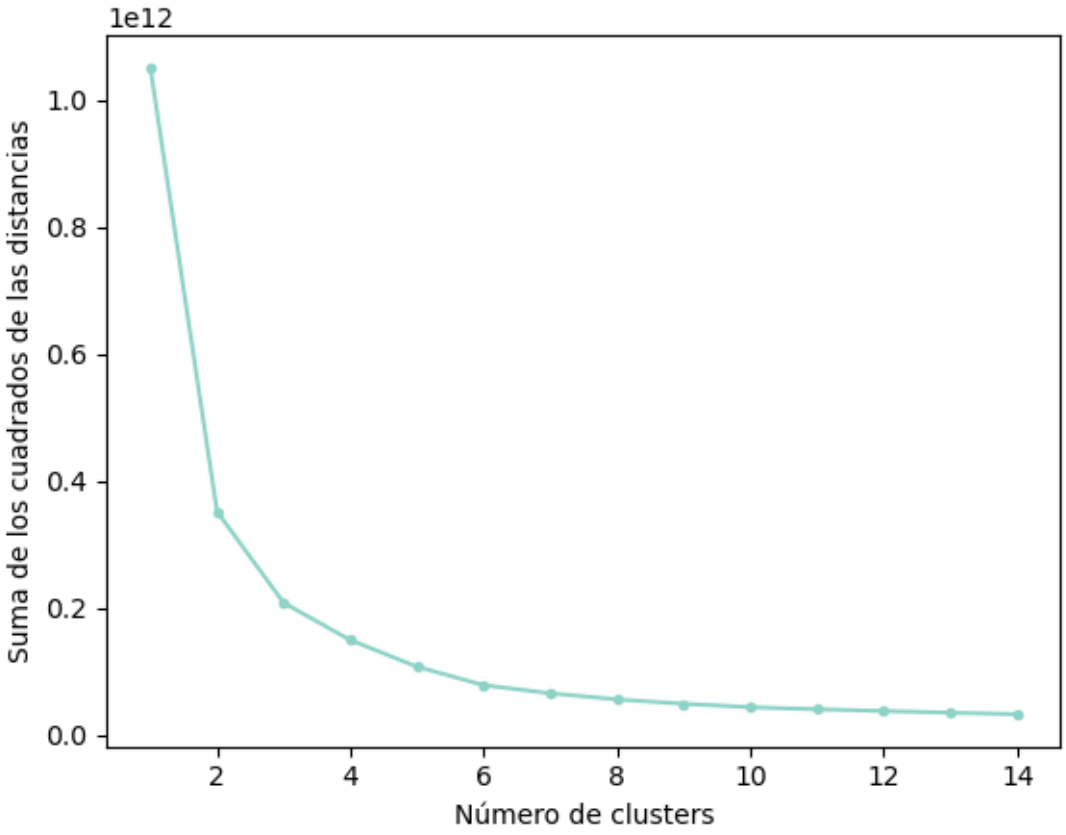
```
C:\Users\Lucia\anaconda3\lib\site-
packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

[139]:
```
# Comprobación de todas las métricas
classReport2 = metrics.classification_report(y_test, y_pred)
print(classReport2)
```

```
              precision    recall  f1-score   support

           0       0.88      0.66      0.76       479
           1       0.73      0.91      0.81       474

    accuracy                           0.78       953
   macro avg       0.80      0.79      0.78       953
weighted avg       0.80      0.78      0.78       953
```

0.9 CLUSTER

[140]:
```
# Método de codo para saber cual es el nñumero óptimo de kmeans a utilizar

# Seleccioanr los datos
datos7 = pd.read_csv("ARCHIVOTRATADO.csv", sep=";")

# Codificar las variables categóricas con la técnica one-hot
data = pd.get_dummies(datos7, columns=['Educacion','Estadocivil','Pais',␣
 ↪'Inscripcion'])

# Seleccionar columnas numéricas relevantes para el análisis
X= data.select_dtypes(include=['int', 'float'])


# Calcular la suma de los cuadrados de las distancias para diferentes valores␣
 ↪de k
sse = []
for k in range(1, 15):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    sse.append(kmeans.inertia_)

# Graficar la suma de los cuadrados de las distancias en función del número de␣
 ↪clusters
hex='#8dd3c7'
plt.plot(range(1, 15), sse, marker='o', color=hex , markersize=3)
plt.xlabel('Número de clusters')
plt.ylabel('Suma de los cuadrados de las distancias')
plt.show()
```



[141]:
```
#estudio de centroides de los cluster para crear los cluster

# Datos de entrada
datos8 = pd.read_csv("ARCHIVOTRATADO.csv", sep=";")
# Codificar las variables categóricas con la técnica one-hot
data = pd.get_dummies(datos8, columns=['Educacion','Estadocivil','Pais',␣
 ↪'Inscripcion'])

# Seleccionar columnas numéricas relevantes para el análisis
X= data.select_dtypes(include=['int', 'float'])

# Crear un objeto KMeans
kmeans = KMeans(n_clusters=3, random_state=42)

# Entrenar el modelo con los datos de entrada
kmeans.fit(X)

# Imprimir las etiquetas de los clusters para cada ejemplo
print(kmeans.labels_)
```

```
# Imprimir las coordenadas de los centroides de los clusters
print(kmeans.cluster_centers_)
```

```
[2 2 0 … 2 0 2]
[[ 1.11298692e+03  5.76322674e+03  1.96753634e+03  7.69498532e+04
   8.57558140e-02  3.50290698e-01  4.90886628e+01  6.16405523e+02
   5.69694767e+01  3.97324128e+02  8.28023256e+01  6.01802326e+01
   7.01162791e+01  1.60901163e+00  5.40261628e+00  5.45203488e+00
   8.40552326e+00  3.15988372e+00  6.83139535e-02  1.35174419e-01
   2.29651163e-01  1.83139535e-01  2.61627907e-02  7.26744186e-03
   3.00000000e+00  1.10000000e+01  2.34011628e-01]
 [ 1.12302838e+03  5.66187162e+03  1.97319730e+03  2.83319365e+04
   8.09459459e-01  3.10810811e-01  4.85621622e+01  3.03256757e+01
   5.99864865e+00  2.55972973e+01  9.08108108e+00  6.06486486e+00
   1.77364865e+01  2.14189189e+00  2.15405405e+00  5.28378378e-01
   3.07972973e+00  6.90810811e+00  8.51351351e-02  4.05405405e-03
   1.66533454e-16  1.35135135e-03 -3.12250226e-17  1.62162162e-02
   3.00000000e+00  1.10000000e+01  1.14864865e-01]
 [ 1.12180296e+03  5.38368596e+03  1.96614409e+03  5.23550398e+04
   4.15024631e-01  8.16502463e-01  4.96256158e+01  2.88532020e+02
   1.88214286e+01  1.00575123e+02  2.50849754e+01  1.81391626e+01
   4.58669951e+01  3.09852217e+00  4.72783251e+00  2.24261084e+00
   6.04433498e+00  5.69334975e+00  6.52709360e-02  8.74384236e-02
   6.15763547e-03  2.09359606e-02  1.47783251e-02  4.92610837e-03
   3.00000000e+00  1.10000000e+01  1.08374384e-01]]
```

[142]:
```
#Saber que tipo de datos son los que resultan del modelo
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Columns: 707 entries, Unnamed: 0 to Inscripcion_31-12-2013
dtypes: float64(1), int64(26), uint8(680)
memory usage: 1.9 MB
```

[143]:
```
# comprobar las columnas que hay con las dummies
data.columns
```

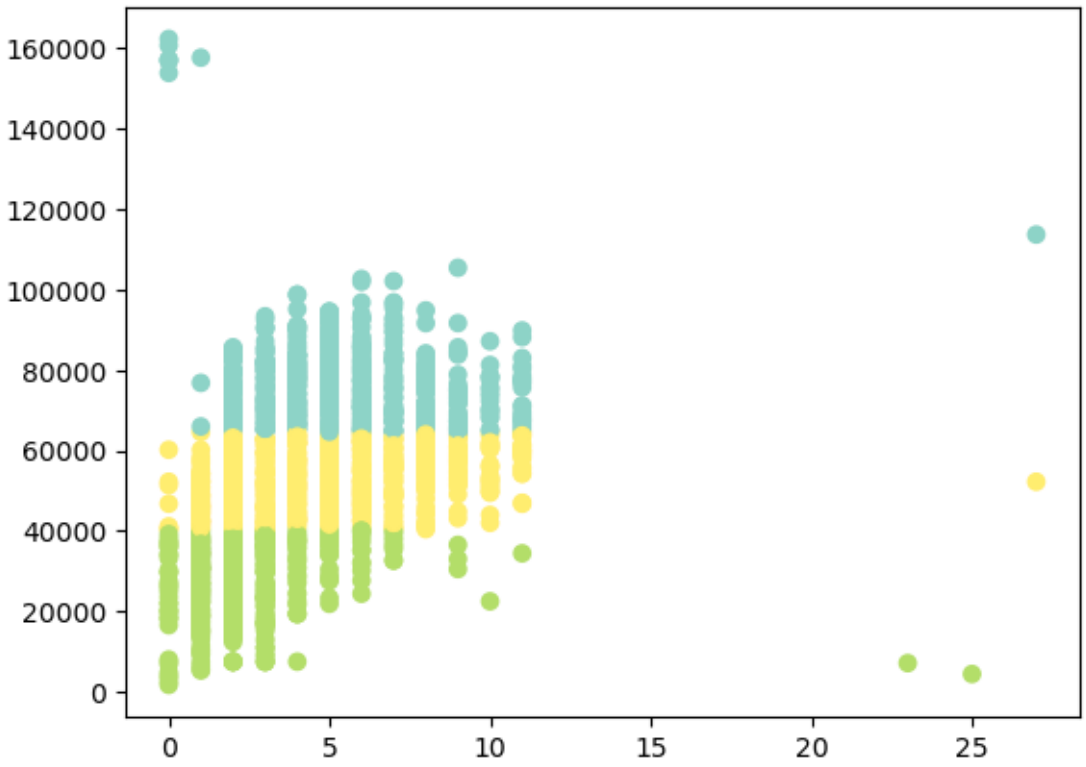
[143]:
```
Index(['Unnamed: 0', 'Identificador', 'Cumpleaños', 'Ingresos', 'Niños',
       'Adolescentes', 'Ultimacompra', 'Totalvinos', 'Totalfrutas',
       'Totalcarnes',
       …
       'Inscripcion_31-03-2014', 'Inscripcion_31-05-2013',
       'Inscripcion_31-05-2014', 'Inscripcion_31-07-2012',
       'Inscripcion_31-07-2013', 'Inscripcion_31-08-2012',
       'Inscripcion_31-08-2013', 'Inscripcion_31-10-2012',
       'Inscripcion_31-12-2012', 'Inscripcion_31-12-2013'],
      dtype='object', length=707)
```

[144]:
```
#comprobación de las columnas sn las dummies
X.columns
```

[144]:
```
Index(['Unnamed: 0', 'Identificador', 'Cumpleaños', 'Ingresos', 'Niños',
       'Adolescentes', 'Ultimacompra', 'Totalvinos', 'Totalfrutas',
       'Totalcarnes', 'Totalpescados', 'Totaldulces', 'Totallujos',
       'Comprasdescuentos', 'Comprasweb', 'Comprascatalogo', 'Comprastiendas',
       'Visitaswebmes', 'Campaña3', 'Campaña4', 'Campaña5', 'Campaña1',
       'Campaña2', 'Quejas', 'Zcostecontacto', 'Zingresos', 'Respuesta'],
      dtype='object')
```

[145]:
```
# Gráfico de dispersión de los cluster
# Asignar cada observación al cluster correspondiente
labels = kmeans.predict(X)

# Visualizar clusters en una gráfica de dispersión
plt.scatter(X['Comprasweb'], X['Ingresos'], c=labels, cmap='Set3')
```

[145]:
```
<matplotlib.collections.PathCollection at 0x1f8b5792ee0>
```



[146]:
```
# Métrica de silueta para comprobar si el cluster es bueno o no. Se evalua de 0␣
↪a 1.
```

```
# X es la matriz de características
# labels es una lista o array que contiene las etiquetas de clúster asignadas a␣
 ↪cada muestra

silhouette_avg = silhouette_score(X, labels)
print("El índice de silueta promedio es:", silhouette_avg)
```

```
El índice de silueta promedio es: 0.4914924038943047
```

[147]:
```
# Indice intracluster para evaluar la distancia de los datos
# X es la matriz de características
# num_clusters es el número de clústeres a evaluar

kmeans = KMeans(n_clusters=3)
kmeans.fit(X)

wcss = kmeans.inertia_
print("La suma de las distancias cuadradas intraclúster es:", wcss)
```

```
La suma de las distancias cuadradas intraclúster es: 207707317697.16895
```

[148]:
```
# Indice itnercluster para evaluar la distancia de los datos
# Obtener los centroides de los clústeres
centroids = kmeans.cluster_centers_

# Calcula la distancia euclidiana entre los centroides
distance = np.linalg.norm(centroids[0] - centroids[1])

print("La distancia interclúster es:", distance)
```

```
La distancia interclúster es: 24601.86390716588
```

[149]:
```
# Código para sacar los datos de cada variable agrupado por cluster

# Obtener las etiquetas de cluster asignadas a cada muestra
labels = kmeans.labels_

# Agregar las etiquetas de cluster al DataFrame
X['cluster_label'] = labels

# Crear un diccionario para almacenar los datos de cada cluster
cluster_data = {}
for label in set(labels):
    cluster_data[label] = X[X['cluster_label'] == label].drop('cluster_label',␣
 ↪axis=1)

# Imprimir los datos de cada cluster
```

```
for label, cluster in cluster_data.items():
    print("Cluster", label)
    print(cluster)
    print()
```

```
Cluster 0
      Unnamed: 0  Identificador  Cumpleaños      Ingresos  Niños  \
0              0           5524        1957  58138.000000      0
1              1           2174        1954  46344.000000      1
4              4           5324        1981  58293.000000      1
5              5           7446        1967  62513.000000      0
6              6            965        1971  55635.000000      0
…            …             …          …            …    …
2233        2233           9432        1977  52247.251354      1
2235        2235          10870        1967  61223.000000      0
2236        2236           4001        1946  64014.000000      2
2237        2237           7270        1981  56981.000000      0
2239        2239           9405        1954  52869.000000      1

      Adolescentes  Ultimacompra  Totalvinos  Totalfrutas  Totalcarnes  …  \
0                0            58         635           88          546  …
1                1            38          11            1            6  …
4                0            94         173           43          118  …
5                1            16         520           42           98  …
6                1            34         235           65          164  …
…              …             …           …            …          …  …
2233             0            23           9           14           18  …
2235             1            46         709           43          182  …
2236             1            56         406            0           30  …
2237             0            91         908           48          217  …
2239             1            40          84            3           61  …

      Visitaswebmes  Campaña3  Campaña4  Campaña5  Campaña1  Campaña2  Quejas  \
0                 7         0         0         0         0         0       0
1                 5         0         0         0         0         0       0
4                 5         0         0         0         0         0       0
5                 6         0         0         0         0         0       0
6                 6         0         0         0         0         0       0
…               …         …         …         …         …         …     …
2233              6         0         0         0         0         0       0
2235              5         0         0         0         0         0       0
2236              7         0         0         0         1         0       0
2237              6         0         1         0         0         0       0
2239              7         0         0         0         0         0       0

      Zcostecontacto  Zingresos  Respuesta
0                  3         11          1
1                  3         11          0
```

```
4                  3          11           0
5                  3          11           0
6                  3          11           0
…                  …          …            …
2233               3          11           0
2235               3          11           0
2236               3          11           0
2237               3          11           0
2239               3          11           1

[812 rows x 27 columns]

Cluster 1
      Unnamed: 0  Identificador  Cumpleaños  Ingresos  Niños  Adolescentes  \
2              2           4141        1965   71613.0      0             0
15            15           2114        1946   82800.0      0             0
18            18           6565        1949   76995.0      0             1
23            23           4047        1954   65324.0      0             1
29            29           1966        1965   84618.0      0             0
…            …             …           …       …           …             …
2211        2211          10469        1981   88325.0      0             0
2213        2213           3661        1995   80617.0      0             0
2217        2217           9589        1948   82032.0      0             0
2221        2221           7366        1982   75777.0      0             0
2238        2238           8235        1956   69245.0      0             1

      Ultimacompra  Totalvinos  Totalfrutas  Totalcarnes  …  Visitaswebmes  \
2               26         426           49          127  …              4
15              23        1006           22          115  …              3
18              91        1012           80          498  …              5
23               0         384            0          102  …              4
29              96         684          100          801  …              2
…              …           …            …           …   …              …
2211            42         519           71          860  …              2
2213            42         594           51          631  …              2
2217            54         332          194          377  …              1
2221            12         712           26          538  …              1
2238             8         428           30          214  …              3

      Campaña3  Campaña4  Campaña5  Campaña1  Campaña2  Quejas  \
2            0         0         0         0         0       0
15           0         0         1         1         0       0
18           0         0         0         1         0       0
23           0         0         0         0         0       0
29           0         0         1         0         0       0
…            …         …         …         …         …       …
2211         0         0         0         0         0       0
2213         0         0         0         0         0       0
```

```
2217         0         0         0         0         0       0
2221         0         1         1         0         0       0
2238         0         0         0         0         0       0

      Zcostecontacto  Zingresos  Respuesta
2                  3         11          0
15                 3         11          1
18                 3         11          0
23                 3         11          0
29                 3         11          0
…                  …         …           …
2211               3         11          0
2213               3         11          0
2217               3         11          0
2221               3         11          1
2238               3         11          0

[687 rows x 27 columns]

Cluster 2
      Unnamed: 0  Identificador  Cumpleaños  Ingresos  Niños  Adolescentes  \
3              3           6182        1984   26646.0      1             0
7              7           6177        1985   33454.0      1             0
8              8           4855        1974   30351.0      1             0
9              9           5899        1950    5648.0      1             1
11            11            387        1976    7500.0      0             0
…             …              …           …       …         …             …
2223        2223           1448        1963   33562.0      1             2
2229        2229          10084        1972   24434.0      2             0
2230        2230           7004        1984   11012.0      1             0
2232        2232           8080        1986   26816.0      0             0
2234        2234           8372        1974   34421.0      1             0

      Ultimacompra  Totalvinos  Totalfrutas  Totalcarnes  …  Visitaswebmes  \
3               26          11            4           20  …              6
7               32          76           10           56  …              8
8               19          14            0           24  …              9
9               68          28            0            6  …             20
11              59           6           16           11  …              8
…              …            …            …            …  …              …
2223            33          21           12           12  …              4
2229             9           3            2            8  …              7
2230            82          24            3           26  …              9
2232            50           5            1            6  …              4
2234            81           3            3            7  …              7

      Campaña3  Campaña4  Campaña5  Campaña1  Campaña2  Quejas  \
3            0         0         0         0         0       0
```

```
7              0          0          0          0          0          0
8              0          0          0          0          0          0
9              1          0          0          0          0          0
11             0          0          0          0          0          0
…            …          …          …          …          …          …
2223           0          0          0          0          0          0
2229           0          0          0          0          0          0
2230           1          0          0          0          0          0
2232           0          0          0          0          0          0
2234           0          0          0          0          0          0

      Zcostecontacto  Zingresos  Respuesta
3                  3         11          0
7                  3         11          0
8                  3         11          1
9                  3         11          0
11                 3         11          0
…                …         …          …
2223               3         11          0
2229               3         11          0
2230               3         11          0
2232               3         11          0
2234               3         11          0

[741 rows x 27 columns]
```

[157]:
```
# Aqui ya se agrupan por cluster y se pueden evaluar los distintos grupos
clustermedia= X.groupby('cluster_label').mean().T
clustermedia
```

[157]:
```
cluster_label                0             1             2
Unnamed: 0          1121.890394   1112.809316   1123.083671
Identificador       5377.431034   5770.388646   5662.222672
Cumpleaños          1966.140394   1967.531295   1973.198381
Ingresos           52385.061926  76967.652111  28348.147099
Niños                 0.416256      0.084425      0.808367
Adolescentes          0.815271      0.350801      0.311741
Ultimacompra         49.624384     49.091703     48.561404
Totalvinos          288.646552    616.861718     30.568151
Totalfrutas          18.821429     57.052402      5.990553
Totalcarnes         100.912562    397.494905     25.570850
Totalpescados        25.158867     82.835517      9.068826
Totaldulces          18.320197     60.053857      6.056680
Totallujos           45.883005     70.196507     17.715250
Comprasdescuentos     3.100985      1.605531      2.141700
Comprasweb            4.732759      5.398836      2.156545
```

Comprascatalogo	2.243842	5.457060	0.529015
Comprastiendas	6.051724	8.401747	3.082321
Visitaswebmes	5.692118	3.155750	6.908232
Campaña3	0.065271	0.068413	0.085020
Campaña4	0.087438	0.135371	0.004049
Campaña5	0.006158	0.229985	0.000000
Campaña1	0.020936	0.183406	0.001350
Campaña2	0.014778	0.026201	0.000000
Quejas	0.004926	0.007278	0.016194
Zcostecontacto	3.000000	3.000000	3.000000
Zingresos	11.000000	11.000000	11.000000
Respuesta	0.108374	0.234352	0.114710

[158]:
```
# Calcular los valores más frecuentes de las variables categóricas en cada␣
 ↪cluster
clustermoda = X.groupby('cluster_label').agg(lambda x: x.value_counts().
 ↪index[0]).T
clustermoda
```

[158]:

cluster_label	0	1	2
Unnamed: 0	0.000000	2.0	3.0
Identificador	5524.000000	4141.0	6182.0
Cumpleaños	1975.000000	1970.0	1976.0
Ingresos	52247.251354	83844.0	7500.0
Niños	0.000000	0.0	1.0
Adolescentes	1.000000	0.0	0.0
Ultimacompra	25.000000	54.0	49.0
Totalvinos	14.000000	483.0	2.0
Totalfrutas	0.000000	0.0	0.0
Totalcarnes	11.000000	124.0	5.0
Totalpescados	0.000000	0.0	0.0
Totaldulces	0.000000	0.0	0.0
Totallujos	4.000000	0.0	1.0
Comprasdescuentos	1.000000	1.0	1.0
Comprasweb	2.000000	4.0	1.0
Comprascatalogo	1.000000	4.0	0.0
Comprastiendas	4.000000	10.0	3.0
Visitaswebmes	7.000000	2.0	7.0
Campaña3	0.000000	0.0	0.0
Campaña4	0.000000	0.0	0.0
Campaña5	0.000000	0.0	0.0
Campaña1	0.000000	0.0	0.0
Campaña2	0.000000	0.0	0.0
Quejas	0.000000	0.0	0.0
Zcostecontacto	3.000000	3.0	3.0
Zingresos	11.000000	11.0	11.0
Respuesta	0.000000	0.0	0.0

[152]:
```
# saber las variables que hay en los cluster
X.columns
```

[152]:
```
Index(['Unnamed: 0', 'Identificador', 'Cumpleaños', 'Ingresos', 'Niños',
       'Adolescentes', 'Ultimacompra', 'Totalvinos', 'Totalfrutas',
       'Totalcarnes', 'Totalpescados', 'Totaldulces', 'Totallujos',
       'Comprasdescuentos', 'Comprasweb', 'Comprascatalogo', 'Comprastiendas',
       'Visitaswebmes', 'Campaña3', 'Campaña4', 'Campaña5', 'Campaña1',
       'Campaña2', 'Quejas', 'Zcostecontacto', 'Zingresos', 'Respuesta',
       'cluster_label'],
      dtype='object')
```

[153]:
```
#Comprobación de las variables categóricas dentro de los cluster

# Agregar las etiquetas de cluster al conjunto de datos
data['Cluster'] = labels

# Iterar sobre cada cluster y obtener las variables dummy para ese cluster
for cluster in range(3):
    cluster_data = data[data['Cluster'] == cluster]
    cluster_dummies = cluster_data.filter(like='Pais')

    print(f"Cluster {cluster} - Variables dummy:")
    print(cluster_dummies.head().sum())
    print()
```

```
Cluster 0 - Variables dummy:
Pais_AUS    0
Pais_CA     0
Pais_GER    0
Pais_IND    0
Pais_ME     0
Pais_SA     0
Pais_SP     0
Pais_US     5
dtype: int64

Cluster 1 - Variables dummy:
Pais_AUS    1
Pais_CA     0
Pais_GER    0
Pais_IND    0
Pais_ME     0
Pais_SA     0
Pais_SP     0
Pais_US     4
dtype: int64
```

```
Cluster 2 - Variables dummy:
Pais_AUS    0
Pais_CA     0
Pais_GER    0
Pais_IND    0
Pais_ME     0
Pais_SA     0
Pais_SP     0
Pais_US     5
dtype: int64
```

[154]:
```python
#Comprobación de las variables categóricas dentro de los cluster

# Agregar las etiquetas de cluster al conjunto de datos
data['Cluster'] = labels

# Iterar sobre cada cluster y obtener las variables dummy para ese cluster
for cluster in range(3):
    cluster_data = data[data['Cluster'] == cluster]
    cluster_dummies = cluster_data.filter(like='Educacion')

    print(f"Cluster {cluster} - Variables dummy:")
    print(cluster_dummies.head().sum())
    print()
```

```
Cluster 0 - Variables dummy:
Educacion_Basico      0
Educacion_Graduado    3
Educacion_Master      1
Educacion_PhD         1
dtype: int64

Cluster 1 - Variables dummy:
Educacion_Basico      0
Educacion_Graduado    1
Educacion_Master      1
Educacion_PhD         3
dtype: int64

Cluster 2 - Variables dummy:
Educacion_Basico      1
Educacion_Graduado    1
Educacion_Master      0
Educacion_PhD         3
dtype: int64
```

[155]:
```
#Comprobación de las variables categóricas dentro de los cluster

# Agregar las etiquetas de cluster al conjunto de datos
data['Cluster'] = labels

# Iterar sobre cada cluster y obtener las variables dummy para ese cluster
for cluster in range(3):
    cluster_data = data[data['Cluster'] == cluster]
    cluster_dummies = cluster_data.filter(like='Estadocivil')

    print(f"Cluster {cluster} - Variables dummy:")
    print(cluster_dummies.head().sum())
    print()
```

```
Cluster 0 - Variables dummy:
Estadocivil_Casado           1
Estadocivil_Divorciado       1
Estadocivil_Parejadehecho    1
Estadocivil_Soltero          2
Estadocivil_Viudo            0
dtype: int64

Cluster 1 - Variables dummy:
Estadocivil_Casado           3
Estadocivil_Divorciado       0
Estadocivil_Parejadehecho    1
Estadocivil_Soltero          1
Estadocivil_Viudo            0
dtype: int64

Cluster 2 - Variables dummy:
Estadocivil_Casado           2
Estadocivil_Divorciado       0
Estadocivil_Parejadehecho    3
Estadocivil_Soltero          0
Estadocivil_Viudo            0
dtype: int64
```

Gráficas de dispersión sin cluster

[156]:
```
# Bucle para que python itere cada variable con otra para crear los gráficos de␣
↪dispersión y no ir haciendolo poco a poco

import os
# Obtener la lista de nombres de columnas del DataFrame
columnas = datos7.columns
```

```
# Crear un directorio para guardar las imágenes
directorio = 'C:\\Users\\Lucia'
os.makedirs(directorio, exist_ok=True)

# Generar scatter plots para cada variable y guardar las imágenes
for columna_x in columnas:
       for columna_y in columnas:
        # Excluir la variable si es la misma en el eje x e y
        if columna_x != columna_y:
            hex='#8dd3c7'
            plt.scatter(datos7[columna_x], datos7[columna_y], color=hex)
            plt.title('Scatter plot: {} vs {}'.format(columna_x, columna_y))
            plt.xlabel(columna_x)
            plt.ylabel(columna_y)

            # Guardar la imagen en un archivo en el directorio especificado
            ruta_imagen = os.path.join(directorio, 'scatter_{}_vs_{}.png'.
 ↪format(columna_x, columna_y))
            plt.savefig(ruta_imagen)

            # Limpiar el gráfico para la siguiente iteración
            plt.clf()

# Notificar la finalización del proceso
print('Imágenes guardadas')
```

```
Imágenes guardadas
```

```
<Figure size 640x480 with 0 Axes>
```