Técnicas de selección de instancias en aprendizaje automático. Estudio y análisis empírico

por

Marco Antonio Peña Cubillos

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

Universidad de Huelva & Universidad Internacional de Andalucía

Septiembre de 2022

Técnicas de selección de instancias en aprendizaje automático. Estudio y análisis empírico

Marco Antonio Peña Cubillos
Máster en Economía, Finanzas y Computación

Supervisado por:
Antonio Javier Tallón Ballesteros
Universidad de Huelva

Abstract

The rise of technology has brought an immense way to generate large amounts of data, which over time has increased in popularity along with machine learning and the possibilities for improving upon previously known and used methods, allowing it to serve as a basis for future research. A new perspective of analysis and use of machine learning techniques to predict results and reduce instances is proposed. This research studies and empirically analyzes some instance selection techniques that allow to considerably reduce the original dataset, in order to be able to comfortably manipulate these large datasets, generating subsets that assimilate to the original one, considering that large datasets require a large computational resource. Eleven traditional classification datasets with large data volumes will be used, where the experimental phase will show the performance of these techniques and the considerable reduction of the original dataset to a subset of data. This will permit anyone to build a classification model and see if its performance is in accordance with the original set, a part of this research was addressed to build the classification model and see its performance. The data presented refer to machine learning models with three significant metrics such as ROC Curves, Accuracy and Kappa Coefficient. Furthermore, the research collected results from the large datasets which were analyzed to determine which of the techniques used best reduced the original dataset.

Keywords— Machine learning, Supervised learning, Instance selection, Algorithms

Resumen

El auge de la tecnología ha traído una inmensa forma generar grandes cantidades de datos, que con el tiempo ha aumentado su popularidad junto con el Aprendizaje Automático y las posibilidades de mejorar los métodos conocidos y utilizados, esto puede servir para investigaciones futuras. Se propone una nueva perspectiva de análisis y uso de técnicas de aprendizaje automático para predecir resultados y reducir instancias. Este Trabajo Fin de Máster estudia y analiza empíricamente algunas técnicas de selección de instancias que permitan reducir considerablemente el conjunto de datos original, con el fin de poder manipular cómodamente estos grandes conjuntos de datos, generando subconjuntos que se asimilen al original, considerando que los grandes conjuntos de datos requieren un gran recurso computacional. Se utilizarán once conjuntos de datos de clasificación tradicionales con grandes volúmenes de datos, donde en la fase experimental se mostrará el rendimiento de estas técnicas y la reducción considerable del conjunto de datos original a un subconjunto de datos. Esto permitirá luego a quién desee, construir un modelo de clasificación y ver si su desempeño es acorde con el conjunto original, se abordó una parte de este Trabajo Fin de Máster a construir el modelo de clasificación y ver su desempeño. Los datos presentados se refieren a los modelos de aprendizaje automático con tres métricas significativas como lo son Curvas ROC, Precisión y Coeficiente de Kappa. Por otra parte, la investigación recolectó los resultados de los grandes conjuntos de datos que fueron analizados para determinar cuál de las técnicas utilizadas logra reducir mejor el conjunto de datos original.

Palabras clave—Aprendizaje automático, Aprendizaje supervisado, Selección de instancias, Algoritmos

Agradecimientos

Dedicado a
mi familia

Índice General

Índice de Tablas

Índice de Figuras

1. Introducción

El Aprendizaje Automático (Machine Learning) es un subcampo de la inteligencia artificial que se define en términos generales como la capacidad de una máquina para para emular la inteligencia humana mediante el aprendizaje del entorno. Los sistemas de inteligencia artificial se utilizan para realizar tareas complejas de forma similar a como los humanos resuelven los problemas. Se establece a partir de los datos con el uso de técnicas provenientes de diferentes campos de la informática y las matemáticas con distintos enfoques o estilos, combina la potencia de cálculo que ofrecen los ordenadores modernos con algoritmos estadísticos, que son capaces de aprender del conjunto de datos la forma que pueda ayudar a los humanos a comprender conceptos más complejos y mejorar la toma de decisiones y, finalmente dar solución al problema. Uno de los aspectos más importantes de los métodos a utilizar es la complejidad computacional. Es muy común que se necesite utilizar estos métodos en grandes conjuntos de datos, por lo tanto, los algoritmos empleados con una complejidad de alto grado podrían ser problemáticos. La complejidad del entrenamiento del modelo y su aplicación son usualmente los dos problemas que se suelen presentar. Cuando se manejan grandes cantidades de datos y se aplica el modelo entrenado en el mundo real se tiene un tiempo estimado de entrega por lo que hace crucial el bajo coste computacional. Dependiendo del problema que se intente resolver y el conjunto de datos que manipulemos o tengamos acceso, necesitamos tener también claridad de seleccionar qué tipo de algoritmo es el acorde. El Aprendizaje Automático tiene diferentes tipos de aprendizaje que se pueden utilizar en distintos contextos para brindarnos respuestas según la naturaleza de nuestras necesidades; estos son los tipos:

- Aprendizaje supervisado: Se realiza cuando se entrena con un conjunto de datos que incluye ejemplos etiquetados. Es decir, nos permite realizar futuras predicciones mediante el ingreso de un vector de características.
- Aprendizaje no supervisado: Las máquinas no reconocen los patrones en bases de datos con etiquetas, en este caso buscan similitudes. En este aprendizaje, los algoritmos no están preparados para detectar un dato específico.
- Aprendizaje por refuerzo: Este aprendizaje se da cuando la máquina aprende por medio de prueba y error hasta obtener la óptima solución para completar una tarea dada.
- Aprendizaje semisupervisado: Se utiliza para las mismas aplicaciones que el aprendizaje supervisado, pero en este caso con datos etiquetados y no etiquetados con el fin de crear un modelo computacional.

El Machine Learning es también denominado maestro del reconocimiento de patrones (Duda et al., 2001), una de la situaciones más presentes en el mundo actual con la llegada de Internet, y sin duda, muy utilizado en la inteligencia artificial.

Machine Learning consiste en dar a los computadores la habilidad de aprender sin estar explícitamente programado (Samuel, 1959), el algoritmo de aprendizaje le permite a la máquina identificar la categoría perteneciente según los patrones en los datos observados; en pocas palabras su objetivo es asignar objetos a clases previamente definidas de un grupo de objetos almacenados ya clasificados (Witten and Frank, 2002) . Los patrones son regularmente descritos mediante un conjunto de atributos de naturaleza numérica y/o nominal. Técnicamente un problema de reconocimiento de patrones (Brereton and Lloyd, 2010) consiste en construir un mapeo $f : U \rightarrow D$ que asigna a cada instancia $x \in U$ descrito por el conjunto de atributos $\vartheta = \{\vartheta 1, ..., \vartheta M\}$ una clase de decisión D de las N posibles en $D = D1, ..., DN$

Cuando se realiza un modelo cabe aclarar que no existe un modelo único para todos los problemas de reconocimiento de patrones y un algoritmo de aprendizaje no es aplicable para todos los problemas (Settouti et al., 2016), por lo que es admirable a destacar en la literatura sobre el reconocimiento de patrones, la existencia de investigaciones donde diversos algoritmos son evaluados sobre múltiples conjuntos de datos (Settouti et al., 2016) (Nápoles et al., 2017) (Felix et al., 2018), demostrándose la variedad existente de estos algoritmos y lo difícil que se vuelve la selección del mas adecuado para un problema previamente dado. Con los estudios anteriormente mencionados, no existen algunos similares de métodos de selección de instancias y problemas de clasificación, que sean de utilidad para investigaciones a futuro.

El interés para llevar a cabo este estudio sobre esta temática es el papel importante que juega la selección de instancias en el procesamiento de datos (García et al., 2015) para machine Learning. El crecimiento exponencial de la cantidad de datos numéricos que se generan actualmente en las empresas y la necesidad de desarrollar y tomar mejores decisiones de negocio, lo que supone un punto de partida para futuras investigaciones en este campo. La selección de instancias es el proceso de reducir la cantidad de datos utilizados para el descubrimiento de conocimiento en bases de datos. Idealmente, esto conducirá a un modelo con al menos el mismo rendimiento que un modelo entrenado con todos los datos. El objetivo principal en la selección de instancias es encontrar un modelo de minería de datos entrenado en un subconjunto de datos DM_s, que funcione igual de bien que uno entrenado en la totalidad de los datos DM_t, evaluado en una determinada medida de rendimiento P. $P\left(DM_s\right) = P\left(DM_t\right)$, en esta tesis no vamos abarcar en la construcción del modelo, sino en el método de selección de instancias con mayor desempeño a la hora de reducir el conjunto de datos original el cual nos permita manejar conjuntos de datos de gran tamaño teniendo en cuenta las limitaciones computacionales.

Uno de los principales retos de la comunidad de la minería de datos es conseguir enfoques rápidos, escalables y precisos (Chawla et al., 2004) para la gestión de los datos. Entre las posibles soluciones para hacer frente a grandes volúmenes de datos es la reducción de los conjuntos de datos. Una técnica de reducción para este problema es la selección de instancias. La selección de instancias es el tema en particular de esta tesis.

1.1. Aprendizaje automático supervisado

El Aprendizaje Automático supervisado es la construcción de algoritmos capaces aprender a producir patrones y modelos computacionales utilizando datos externos previamente, para predecir el valor o el destino de instancias futuras (Fernandes de Mello and Antonelli Ponti, 2018)(García et al., 2010) (Benavoli et al., 2016)

El objetivo del aprendizaje supervisado es descubrir qué relación hay entre los atributos de entrada ya sean variables o características con un atributo objetivo. La variable o atributo objetivo puede ser numérica o categórica: Si la variable es numérica, la tarea de a predecir se denomina regresión, diferente de la de clasificación donde la variable objetivo es discreta o categórica. Los algoritmos de clasificación del Aprendizaje Automático supervisado tienen como principal objetivo categorizar el conjunto de datos a partir de una información previamente dada.

La clasificación está presente en muchos problemas de la ciencia de los datos. Numerosas aplicaciones del Aprendizaje Automático se han propuesto durante los últimos años para resolver estos problemas, así como una gran variedad de métodos y algoritmos. Técnicas basadas en reglas, técnicas basadas en la lógica, técnicas basadas en instancias, y técnicas estocásticas.

El término modelo que usaremos en esta tesis se denota como la estructura que generan algunos algoritmos de aprendizaje tras superar la fase de aprendizaje. Esta fase de aprendizaje consiste en entrenar el algoritmo con un conjunto de datos etiquetados, de tal manera que el algoritmo pueda descubrir la relación entre los atributos de entrada y el atributo objetivo. Sin embargo, no todos los algoritmos crean un modelo de predicción, En consecuencia, los algoritmos pueden agruparse en dos grupos: aprendizaje ansioso y aprendizaje perezoso de los cuales no hablaremos en esa tesis.

Los conjuntos de datos utilizados para el entrenamiento normalmente se describen como un conjunto de instancias. Cada instancia es un vector de valores de atributos, uno de los cuáles es la variable objetivo. Por lo general, los valores de los atributos son nominales o numéricos, pero existen otros como booleano, de fecha, binarios, entre otros.

El aprendizaje supervisado tiene una multitud de campos de aplicación, como la bioinformática, las finanzas, medicina, ingeniería, telecomunicaciones, cual es el mejor momento para llamar a un cliente, predicciones económicas, fluctuaciones en el mercado bursátil, actualizaciones de Twitter y otras redes sociales, optimizar campañas publicitarias, entre otros.

1.1.1. Clasificación

El objetivo de los problemas de clasificación en aprendizaje automático supervisado es predecir valores categóricos o nominales y a partir de datos previamente etiquetados de tipo continuo, binario, continuo o categóricos e indistintamente. Como medida de desempeño de los algoritmos existen varias para evaluarlos, entre las que comúnmente se encuentra son:

Curvas ROC: Es una métrica de las más utilizadas en modelos binarios, permite mostrar el desempeño (Cerezo, 2004) de un modelo de clasificación a través de parámetros de la matriz de confusión, donde podemos ver el desempeño completo de un modelo de clasificación mediante:

- Tasa de verdaderos positivos (TVP): mide la sensibilidad (Hajian-Tilaki, 2013) del modelo donde VP (verdaderos positivos) y FN (falsos negativos) de tal manera que:

$$TVP = \frac{VP}{(VP+FN)}$$

- Tasa de falsos positivos (TFP), donde FP (falsos positivos) y VN (verdaderos negativos), tal que:

$$TPR = \frac{FP}{(FP+VN)} = 1 - especificidad$$

$$especifidad = \frac{VN}{(VN+FP)}$$

donde FP y VP tienen valores en el rango [0,1]. Valores entre 0 y 1: el rendimiento es mayor cuando $AUC \to 1$

Una curva ROC (Park et al., 2004) es representación gráfica de TVP en función de TPR según varie su umbral de discriminación. La curva ROC gráfica se produce trazando la sensibilidad (tasa de verdaderos positivos) en el eje Y contra 1- especificidad (tasa de falsos positivos) en el eje X para los distintos valores tabulados.

Una curva ROC sigue la línea diagonal $y = x$ que produce los resultados de FP (Hoo et al., 2017) en la misma proporción que los VP. Por lo tanto, si el resultado de un test con una precisión razonable su curva ROC estará el triángulo superior izquierdo por encima de la línea $y = x$

El área bajo la curva ROC (Area Under Curve AUC) es una medida global (Gonçalves et al., 2014) que tiene la capacidad de discriminar si una condición específica está presente o no. Un AUC de 0,5 representa una prueba sin capacidad de discriminación (es decir, no mejor que el azar), mientras que un AUC de 1,0 representa una prueba con una discriminación perfecta. Es conveniente porque es invariable respecto a la escala, permite medir adecuadamente que tan bien se clasifican las predicciones, en lugar de valores absolutos y, es invariable con respecto al umbral de clasificación. Esta medida nos da a conocer la calidad de las predicciones del modelo, sin tener que umbral de clasificación se seleccione.

Precisión (Accuracy): La precisión (Kull and Flach, 2014) es una de las métricas para evaluar los modelos de clasificación. Informalmente, la precisión es la fracción de predicciones que nuestro modelo ha clasificado acertadamente. Esta métrica nos da la calidad de los clasificadores y se encuentra su valor entre [0,1]. Formalmente, la precisión tiene la siguiente definición:

$$Accuracy = \frac{Nmero\ de\ predicciones\ correctas}{Nmero\ total\ de\ predicciones\ realizadas} = \frac{VP+VN}{Total\ de\ la\ muestra}$$

Coeficiente de Kappa: El coeficiente de Kappa de Cohen, se usa para evaluar la concordancia o reproducibilidad (Tallón-Ballesteros and Riquelme, 2014) observada en el conjunto de datos, permite ver la proporción de los resultados observados más allá del azar respecto del máximo de resultados esperados más allá del azar.

El coeficiente toma valores entre [-1,1], dependiendo del carácter observado y su prevalencia así mismo es su grado de acuerdo, mientras este más cercano a 1 mayor es el grado de acuerdo. Un valor 0 significa que no hay acuerdo, lo que la observación ocurriría al azar, diferente de que si el resultado es 1 su grado de acuerdo seria perfecto. Esta medida (Eugenio and Glass, 2004) se considera bastante acertada y solidad más que una precisión estándar, ya que este coeficiente nos permite ver la coincidencia que se produce por la casualidad.

1.2. Introducción al preprocesamiento de datos

Como se ha mencionado en la literatura del Aprendizaje Automático, los datos de entrada son la piedra angular este. Los algoritmos necesitan que los conjuntos de datos sean precisos y que estén bien estructurados para poder realizar su entrenamiento. En la realidad y por desgracia, estos conjuntos de datos no suelen estar bien estructurados, diversos factores externos suelen afectar la consistencia de los datos en el mundo real, por ejemplo, la presencia de ruido, datos superfluos, la inmensa cantidad de atributos e instancias en los conjuntos de datos (Triguero et al., 2014). Por esta razón las tareas de preprocesamiento de datos ocupan un rol importante en el tiempo en los flujos de trabajo de aprendizaje automático. Usualmente se suelen agrupar en dos grupos de técnicas usualmente utilizadas como la preparación de datos y la reducción de datos.

1.2.1. Preparación de datos

Formalmente la preparación de los datos es el proceso de recopilación, combinación, estructuración y organización de los datos (Singh et al., 2016), pero la preparación de los datos es más que organizarlos. La preparación de los datos comprende el proceso de limpieza y transformación de los datos brutos antes de su procesamiento y respectivo análisis. Es un paso fundamental antes del procesamiento que de manera usual implica reformatear los datos, hacer correcciones en ellos y combinar conjuntos de datos para así enriquecerlos. ¿Por qué es tan importante? Porque se pueden hacer análisis más precisos y significativos. A partir de un análisis de datos más significativo se obtienen mejores resultados. Imaginemos un problema de interés: los conjuntos de datos tienen que ser recogidos en primer lugar de la fuente o fuentes y luego preparados para su uso. Los pasos para la preparación de los datos se agrupan y explican brevemente a continuación:

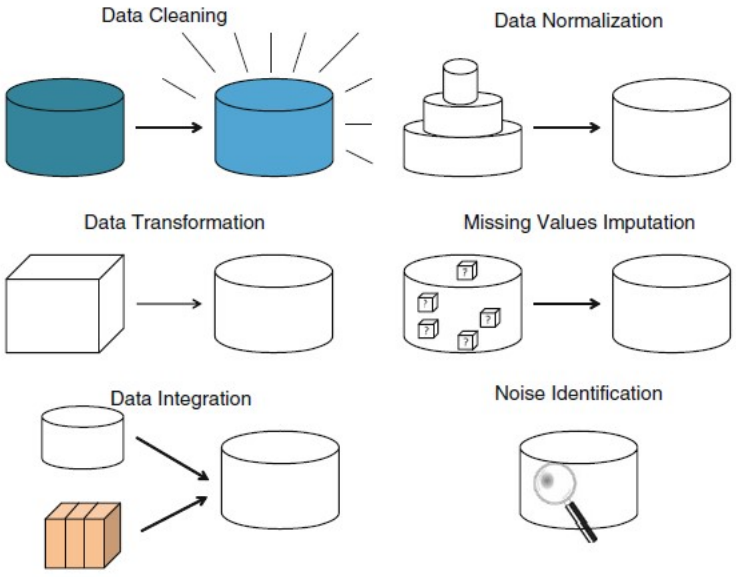

Figura 1: Preparación de datos. Reproducción del libro Data Mining Preprocessing in Data Mining (García, Luengo, y Herrera, 2015)

- Limpieza de datos: El paso más importante (Zhang et al., 2003) de la tarea de preparación de los datos, ya que se ocupa de corregir los datos incoherentes rellenando valores faltantes y suavizando

los datos ruidosos. En este paso se puede encontrar muchas filas en el conjunto de datos que no tengan valor para los atributos de interés, puede haber registros que se encuentren duplicados y quizá encontrarnos con algún otro error no esperado. Todos estos problemas que se presentan en los conjuntos de datos se tratan en el primer paso de la preparación de datos. En pocas palabras, el objetivo de la limpieza de datos es obtener un conjunto de datos bien estructurado y adecuado para alimentar nuestro algoritmo.

- Transformación de datos: Este paso requiere la eliminación de cualquier ruido de los datos, la normalización, la agregación y la generalización.
- Integración de datos: Esta etapa implica la integración de esquemas, la resolución de datos procedentes de múltiples fuentes de datos. Hay que tener especial cuidado para evitar instancias duplicadas y dominios de datos diferentes.
- Normalización de datos: Los distintos atributos pueden tener rangos diferentes y algunos métodos, como los métodos basados en la distancia, son muy sensibles a la escala de las características. La normalización proporciona el mismo rango y escala para todos los atributos.

1.2.2. Reducción de datos

Hasta ahora los conjuntos de datos disponibles aumentan progresivamente de tamaño. Como consecuencia muchos sistemas tienen problemas a la hora de procesar los conjuntos de datos para obtener información o conocimiento (García-Pedrajas and de Haro-García, 2014) que se pueda explotar de tal manera con las herramientas y técnicas convencionales de procesamiento de datos se dificulta obtener este conocimiento.

Los sistemas de información actuales producen enormes cantidades de datos que son difíciles de medir (Kim and Oommen, 2003), esto significa que la cuestión de cómo resolver los problemas del Aprendizaje Automático a gran escala esta abierta y requiere de muchos esfuerzos de investigación y recursos computacionales.

El objetivo de la reducción de datos es disminuir la dificultad y mejorar la calidad de los conjuntos de datos resultantes mediante la reducción de su tamaño. Así la reducción de datos identifica y a su vez lleva a descartar la información que es poco relevante o que se repite. El tamaño de los conjuntos de datos puede reducirse tanto en atributos como en instancias. Una vez realizada la reducción de datos, podemos encontrar conjuntos de datos con unas dimensiones más reducidas que representen al conjunto de datos original. Esto supone también que el conjunto de datos contenga una cantidad de información idéntica o similar a la del conjunto de datos original. Como se puede observar en la figura 2 las instancias se consideran representadas en filas de manera que, si no es así, estaríamos analizando atributos.

El objetivo de reducir los datos es no perder la información extraída, sino aumentar la eficacia del Aprendizaje Automático cuando los conjuntos de datos son bastante grandes (Yıldırım et al., 2016). Se considera que este el punto más crítico al momento de realizar la reducción de datos, a continuación, se agrupan algunas técnicas relevantes:

- Discretización de datos: Los conjuntos de datos suelen tener tres tipos de atributos: Nominales, Continuos y Ordinales. En algunos casos los algoritmos solo aceptan atributos categóricos. La

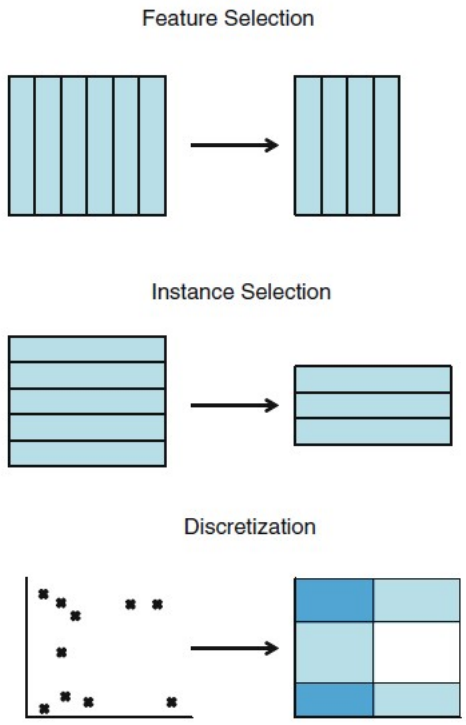

Figura 2: Tipos de reducción de datos. Reproducción del libro Data Mining Preprocessing in Data Mining (García, Luengo, y Herrera, 2015)

etapa de la discretización de los datos ayuda a dividir los atributos continuos y a reducir el tamaño del conjunto de datos, preparándolo así para su respectivo análisis.

El proceso de discretización de los datos también puede considerarse parte de la etapa de preparación de los datos.

La decisión de un incluirlo como una tarea de reducción de datos se explica en (García et al., 2015): La etapa de la discretización mapea los datos de un rango de valores numéricos en un subconjunto reducido de valores categóricos.

- Selección de atributos: La selección de atributos es una técnica de reducción de la dimensionalidad del conjunto de datos eliminando los atributos poco relevantes o redundantes (Li et al., 2017). El objetivo de la selección de atributos (García et al., 2015) es obtener un subconjunto representativo de atributos que tenga un menor numero de estos, tal que la distribución de la probabilidad resultante de los atributos de salida de los datos sea lo más parecida posible a la distribución original obtenida usando todos los atributos. Esto facilita la comprensión de los patrones extraídos y aumenta la velocidad del aprendizaje.

- Extracción de atributos: Es un proceso mediante el cual un conjunto inicial de datos se reduce a un subconjunto para mejorar su procesamiento. Estos métodos seleccionan o fusionan subconjuntos, crean nuevos atributos artificiales, incluyen modificaciones en los atributos o eliminan atributos.

- Generación de instancias: Se crean o ajustan ejemplos artificiales o ejemplos sustitutos (Garcia

et al., 2012b) con el objetivo de mejorar la representación de los límites de decisión en el aprendizaje supervisado, dado que ayuda al conjunto de datos a reducir su tamaño.

- Selección de instancias: Consiste en encontrar el subconjunto más representativo del conjunto de datos inicial, sin disminuir la capacidad de predicción original. Estos métodos están orientados a la forma inteligente de elección del mejor subconjunto posible de los datos originales mediante el uso de reglas o heurística. En otras palabras, si entrenamos un algoritmo sobre el conjunto original de datos y el otro algoritmo con el mejor subconjunto posible, ambos algoritmos deben tener un rendimiento similar (Nanni and Lumini, 2011). La selección de instancias se puede tomar como un caso de generación de instancias en que las instancias a generar se limitan de las instancias originales. Estos métodos toman un papel esencial en los procesos de reducción de datos. Los procesos de selección de atributos o discretización reducen la complejidad, los métodos de selección de instancias reducen el tamaño del conjunto de datos (García et al., 2016a).

2. Selección de instancias

Como se ha explicado anteriormente en la literatura, el propósito a menos lo que se pretende de los algoritmos de selección de instancias, es reducir la complejidad de los algoritmos de aprendizaje al reducir el número de instancias de los conjuntos de datos (Leyva et al., 2015). La intención de estos algoritmos es poder extraer el subconjunto con mayor significatividad de instancias dejando a un lado aquellas instancias que no aportan información relevante o valiosa. La figura 3 figura ilustra el proceso de selección de instancias. Reducir el conjunto de datos deja ver dos ventajas principales:

- Reducir el espacio que ocupa este en el sistema (Olvera-López et al., 2010)
- Disminuir el tiempo de procesamiento de las tareas de aprendizaje

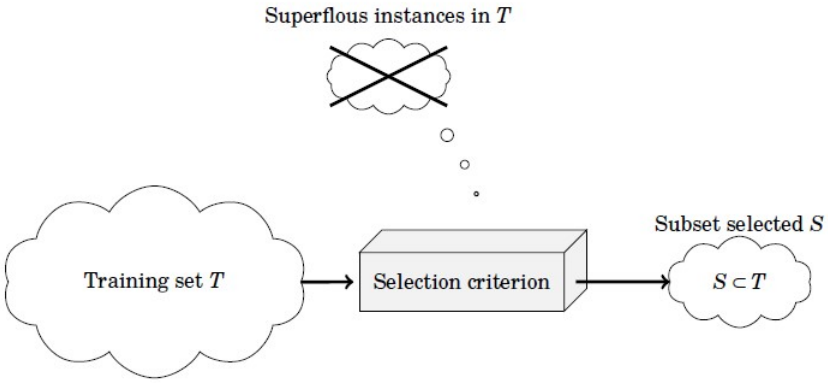

Figura 3: Proceso de selección de instancias. Reproducción del artículo A review of instance selection methods (Olvera-López, Carrasco-Ochoa, Martínez-Trinidad, y Kittler, 2010)

El conjunto de instancias seleccionadas lo podemos utilizar para entrenar cualquier tipo de clasificador pero, en el pasado algunos algoritmos de selección de instancias han sido creados para el clasificador de k vecinos más cercanos (Cover and Hart, 1967), o kNN en su abreviación. De tal manera para el proceso de selección de instancias es también utilizado el término selección de prototipos (García et al., 2015). En esta tesis se utiliza el termino de selección de instancias para referirse al proceso que supone

la selección de un subconjunto de instancias del conjunto original de datos, sin considerar el algoritmo que posteriormente sea utilizado.

En el momento de analizar conjuntos de datos o grandes volúmenes de datos en el mundo real empresarial, la necesidad de utilizar un algoritmo de selección de instancias se hace más notoria. Hoy en día el tamaño de los conjuntos de datos es mayor y, a medida que avanza la tecnología se hace cada vez más grande estos conjuntos de datos. Por otra parte, los conjuntos de datos en la realidad suelen contener instancias ruidosas, valores perdidos, valores atípicos y anomalías.

Han sido numerosos los intentos de entrenar un clasificador sobre conjuntos de datos de millones de instancias, ya que esto suele ser una tarea compleja o incluso inabordable para el clasificador. Por tanto la selección adecuada de un subconjunto de instancias es de tal manera una buena opción para reducir el tamaño de la muestra (García et al., 2015), y mejorar el posterior tratamiento de la muestra.

2.1. Taxonomía

La selección de instancias suele tener como meta encontrar un subconjunto de datos que al reducirlo sea similar al conjunto de datos original, tal que su desempeño al ser entrenado sea igual en el subconjunto de datos como entrenarlo con todos los datos. En general tres objetivos (Garcia et al., 2012a) son buscados y son los siguientes:

- Activación (Enabling): Anteriormente los conjuntos de datos que se utilizaban en el Aprendizaje Automático eran de menor tamaño que los actuales. Hoy en día, en la vida real nos enfrentamos a conjuntos de datos con millones de instancias y una enorme cantidad de atributos. Por tanto, para poder realizar modelos y ajustarlos de una manera significativa, se vuelve necesario disminuir el numero de instancias considerablemente. La selección de instancias por tanto permite que los algoritmos utilizados trabajen de una forma óptima y eficiente al disminuir el número de instancias con las que se va a entrenar el modelo.
- Enfoque (Focusing): En algunos casos los conjuntos de datos contienen información específica del dominio, lo que hace necesario reducir los puntos en los datos de partes especificas y enforcarse en las partes importantes de los conjuntos de datos.
- Limpieza (Cleaning): Los conjuntos de datos comúnmente contienen datos redundantes, ruido e incluso anomalías. Aplicando los métodos de selección de instancias previamente es posible mejorar la calidad del conjunto de datos limpiándolos y eliminando datos que sean redundantes.

Los métodos de selección de instancias normalmente suelen clasificarse en lo siguiente: La dirección de la búsqueda, el tipo de selección y la evaluación de la búsqueda (Garcia et al., 2012a). En la figura 4 podemos observar los métodos de selección de instancias desde el inicio o desde sus orígenes hasta 2012. La tabla 1 recopila algunas características importantes de los algoritmos de selección de instancias más utilizados.

Algunos términos en general han sido utilizados para la selección de instancias mas relevantes del conjunto de entrenamiento. La selección de instancias en un principio fue pensada para trabajar con otros métodos de aprendizaje, como arboles de decisión, ANNs (Artificial neural networks), SVMs (Support vector machines, SVM).

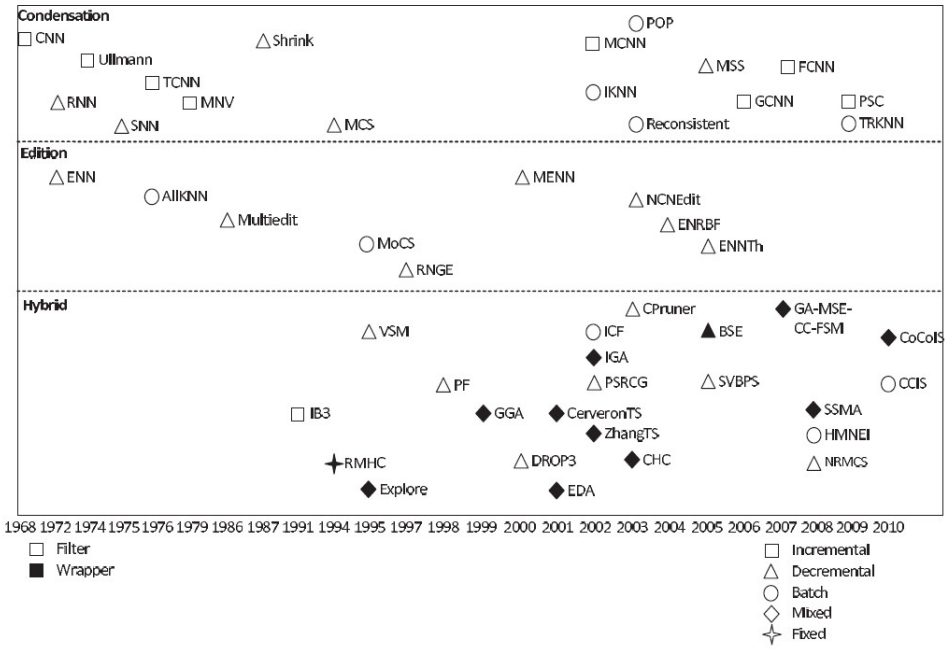

Figura 4: Métodos de selección de instancias de acuerdo a su taxonomía. Reproducción del artículo Prototype selection for nearest neighbor classification: Taxonomy and empirical study (Garcia, Derrac, Cano, y Herrera, 2012)

Tabla 1: Métodos de selección de instancias por taxonomía. Reproducción del artículo y sus autores (Jankowski and Grochowski, 2004)

Tipo de selección	Dirección de búsqueda	Algoritmo	Año	Autor
Edición	Incremento	LSSm	2015	(Leyva et al., 2015)
	Disminución	ENN	1972	(Wilson, 1972)
	Lote	All kNN	1976	(Tomek, 1976)
Condensación	Incremento	CNN	1968	(Hart, 1968)
	Incremento	PSC	2010	(Olvera-López et al., 2010)
	Disminución	RNN	1972	(Gates, 1972)
	Disminución	MSS	2002	(Barandela et al., 2005)
Híbrido	Disminución	DROP1-5	2000	(Wilson and Martinez, 2000)
	Lote	ICF	2002	(Brighton and Mellish, 2002)
	Lote	HMN-EI	2008	(Marchiori, 2008)
	Lote	LSBo	2015	(Leyva et al., 2015)

2.1.1. Dirección de búsqueda

La selección de instancias se puede considerar como la búsqueda del problema, en pocas palabras, a partir de una medida la meta es encontrar el conjunto de datos con mayor representatividad de instancias para esa medida (Cano et al., 2005). Entre sus características se pueden definir en cinco grupos en función de la dirección de búsqueda:

- Incremento (Incremental): Inician con un conjunto de datos vacío y se van agregando instancias en función de criterios predefinidos. El problema aquí es que posiblemente no se tengan todos los datos desde el principio y su sensibilidad al orden de aparición de las instancias. Las principales ventajas son que los conjuntos de datos pueden ser procesados como un flujo, son considerablemente rápidos y, no necesitan demasiados requisitos para su almacenamiento.

- Disminución (Decremental): Funciona de forma análoga y en la dirección opuesta a la Incremental, empiezan con todo el conjunto de datos y las va eliminando instancias una a una siguiendo criterios predefinidos. El orden es importante pero no tanto como en el grupo anterior. En este caso es importante que todo el conjunto de datos este disponible desde el principio y, su principal inconveniente es que todo el conjunto de datos tiene que guardar en la memoria.

- Lote (Batch): Comienzan con un conjunto de datos completo, donde se comprueba que todas las instancias del conjunto cumplan algún criterio. Las instancias se analizan por grupos, es decir, se procesan en grupos sucesivamente y son etiquetadas para eliminarlas al final del algoritmo. La ventaja que se tiene es que prevalece una visión total del conjunto de datos en todo momento.

- Mixtos (Mixed): Se inicia con un subconjunto no vacío al cual se le pueden añadir y eliminar instancias según un criterio predefinido. Estos tienen lugar entre los tres grupos anteriores.

- Fijos (Fixed): Estos métodos son similares a los mixtos, son una subfamilia, pero, en este caso, el número de adiciones que se realicen, deben ser igual al total de eliminaciones, tal que la cantidad de instancias para el subconjunto está predefinida desde el principio como parámetro de entrada del algoritmo.

2.1.2. Tipo de selección

Lo fundamental en todo proceso de clasificación son los limites de clasificación. Los límites de decisión están conformados por instancias con dos o mas clases diferentes que estén cercanas entre sí. Por lo tanto, estas instancias pueden ser puntos fronterizos o estar cerca de los límites, o puntos centrales (Liu and Motoda, 2002) que estén lejos de los límites (Wilson and Martinez, 1997).
Un factor que distingue a las técnicas de reducción de instancias es si buscan retener los puntos fronterizos, los puntos centrales o algún otro conjunto de puntos. Algunos algoritmos tratan de eliminar los puntos fronterizos, estos eliminan puntos que son ruidosos o que no coinciden con sus vecinos. Por tal razón se eliminan los puntos fronterizos cercanos, dejando atrás los límites de decisión más suaves. En este caso los algoritmos (Wilson and Martinez, 2000) no eliminan puntos internos que no afecten a la frontera de decisión. Puede resultar necesario considerar un gran número de puntos fronterizos para definir completamente una frontera, por lo que algunos algoritmos conservan los puntos centrales para utilizar las instancias más típicas de una clase particular para clasificar las instancias cercanas a ellas.

- Algoritmos de condensación: Estos algoritmos intentan eliminar los puntos internos bajo la premisa de que no afecten la clasificación, por lo cual intentan retener las instancias cercanas a

los puntos de decisión. En pocas palabras intentan eliminar las instancias alejadas de la superficie de decisión. Usualmente logran alcanzar altos índices de reducción, pero el principal problema de estos métodos es que las instancias ruidosas (Jankowski and Grochowski, 2004) los afectan considerablemente.

- Algoritmos de edición: Estos algoritmos funcionan de manera contraria, la manera que trabajan es eliminando los puntos fronterizos o las instancias que no coincidan con sus vecinos. Los algoritmos de decisión se enfocan en la eliminación del ruido, en consecuencia, sus índices de reducción son inferiores en comparación a los de los algoritmos de condensación.

- Algoritmos híbridos: Estos algoritmos se sitúan entre las técnicas anteriormente mencionadas. Se enfoca en combinar los dos enfoques anteriores para crear un modelo con capacidad de generalización. Su objetivo es hallar el subconjunto de instancias de menor tamaño y con mayor precisión. Por tal razón se eliminan las instancias tanto centrales como fronterizas.

  En los últimos años un nuevo enfoque de selección ha surgido, se podría decir que lo mas cercano de clasificarse sería de enfoque hibrido, pero no encaja en ninguna de las categorías anteriores. Los métodos de rango (Rico-Juan and Iñesta, 2012) Estos métodos clasifican o asocian una puntuación a cada instancia, es decir, su utilidad para el proceso de clasificación, tras esto selecciona un subconjunto con las mejores instancias (Valero-Mas et al., 2017)

2.1.3. Evaluación de la búsqueda

Mediante la selección de instancias, podemos reducir el conjunto de entrenamiento, lo que puede ser de gran ayuda para reducir el tiempo en el proceso de entrenamiento, en especial cuando estamos en el proceso de clasificación de los clasificadores basados en instancias, ya que, para clasificar una instancia, estos clasificadores recorren por todo el conjunto de entramiento. Según la estrategia utilizada para la selección de instancias, podemos dividir los métodos (Olvera-López et al., 2010) de selección de instancias en dos grupos que son los siguientes:

- Envolvente (Wrapper): El criterio de selección utilizado se basa en la precisión que se puede obtener por un clasificador. La decisión de eliminar o seleccionar una instancia se obtiene usualmente mediante un clasificador que suele ser el kNN

- Filtro (Filter): El criterio de decisión se basa en utilizar alguna heurística o regla y no en un clasificador, en pocas palabras una función de selección no basada en un clasificador.

2.1.4. El problema computacional

Un problema considerable se presenta cuando hay grandes conjuntos de datos y tienen muchos atributos predictores es la maldición de la dimensionalidad (García et al., 2016a). Esto puede convertirse en un serio impedimento para el funcionamiento de la mayoría de los métodos debido a los costes de complejidad. Esta definición por (Bellman and Kalaba, 1959) consistía en caracterizar un problema que aumenta a medida que se añaden más atributos o características a un modo.

2.1.5. Métodos seleccionados

Como estudio empírico de esta tesis en técnicas de selección de instancias, se seleccionaron diversos métodos con distintos criterios de decisión, para poder determinar cuál de estas técnicas tenía mejor rendimiento. Dicho esto inicialmente nos centramos en El Data sampling (Muestreo de datos) que es una

técnica de análisis estadístico (Thompson, 2012) que se utiliza para seleccionar, manipular y analizar un subconjunto de datos, el cual identifica patrones y tendencias en el conjunto de datos más amplio analizado. Esto permite trabajar con datos de una menor cantidad para construir y ejecutar modelos más rápidamente sin dejar de producir resultados precisos. Data sampling (Mahmud et al., 2020) ha sido una técnica importante para manejar grandes conjuntos de datos en la etapa de preprocesamiento, nos ayuda a representar mejor un conjunto antes de cargarlo en un modelo. En esta tesis, probamos empíricamente métodos que nos proporcionen una selección de instancias o manipulación de instancias que nos permitan reducir conjuntos de datos de gran cantidad en subconjuntos sin perder precisión, ya que se busca representar en el subconjunto las instancias más relevantes. Consideramos el sampling como método de reducción de instancias, para este caso, tomamos dos métodos de sampling que son los siguientes:

- Kennard-Stone: Originalmente se diseñó para conjuntos de datos de calibración y prueba a partir de un conjunto de datos original (Li et al., 2020). En este caso utilizamos este método para reducir el tamaño de los conjuntos de datos para mejorar el rendimiento de un modelo de clasificación. El método selecciona un subconjunto de muestras que proporciona una cobertura uniforme sobre el conjunto de datos e incluye muestras en el límite del conjunto de datos. De tal manera que realiza la división del conjunto de datos original en el subconjunto de calibración y validación, tal que cada uno de ellos contenga las muestras que puedan lograr captar la máxima variabilidad del conjunto original.

- Método de muestreo aleatorio (Random sampling method RS) (Rendon et al., 2020): Es un método que elige los elementos que constituirán los subconjuntos de la muestra de manera aleatoria. Es también conocido como muestreo probabilístico, es un método que permite aleatorizar la selección de la muestra. En esta tesis se utilizó el método de muestreo aleatorio simple (Simple random sampling) (Etikan and Bala, 2017) que es un subconjunto de una población estadística en el que cada miembro del subconjunto tiene la misma probabilidad de ser seleccionado. Una muestra aleatoria simple es una de las técnicas de selección de muestras que pretende ser una representación no sesgada de un grupo.

En la literatura ya mencionada, se ha señalado que existen un número de algoritmos de selección de instancias existentes y cada año aparecen muchos otros. Los algoritmos de selección de instancias más influyentes según (García et al., 2016b) se presentan a continuación:

- Condensed Nearest Neighbours (CNN): El algoritimo CNN de (Hart, 1968) se considera el primer intento de reducir el tamaño del conjunto de entrenamiento con su regla del vecino más cercano condensando (CNN). El concepto de consistencia que tiene con respecto al conjunto de entrenamiento es primordial en el uso de este algoritmo. Su algoritmo encuentra un subconjunto S del conjunto de entrenamiento T tal que cada miembro de T este mas cerca de un miembro de S que posea la misma la clase que a un miembro de S con una clase diferente. De tal manera que el subconjunto S pueda utilizarse para clasificar de manera correcta todas las instancias pertenecientes de T, suponiendo previamente que T no tenga dos instancias en T con entradas idénticas, pero con clases diferentes. Comienza seleccionando aleatoriamente una instancia perteneciente a cada clase de salida de T y colocándola en S. Tal que cada instancia de T se clasifica usando únicamente las instancias de S. En este proceso, cuando una instancia esta mal clasificada, se añade a S, asegurando que así se clasificará correctamente. Este proceso se repite hasta que no queden más instancias en T que estén mal clasificadas (García et al., 2015). Al final, el algoritmo terminará devolviendo S como conjunto de datos seleccionado.

Algorithm 1: Condensed Nearest Neighbour (CNN)

Input: A training set $X = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
Output: The set of selected instances $S \subseteq X$

```
S = {x_1}
foreach x ∈ X do
    if x is misclassified using S then
        Add x to S
        Restart
return S
```

Figura 5: Algoritmo CNN. Reproducción de Tesis Doctoral (Arnaiz González et al., 2018)

- Edited Nearest Neighbour (ENN): La primera propuesta de esta técnica para editar los conjuntos de datos fue presentada por (Wilson, 1972) y como se mencionó anteriormente en la tabla 1 su clasificación. Su funcionamiento inicia seleccionando todo el conjunto de datos X en el cuál se elimina cada instancia que no este bien clasificada por sus K vecinos más cercanos. El número de vecinos más cercanos K es un parámetro del algoritmo. En la literatura original se fijó K = 3 tal como se puede observar en la figura 6 su seudocódigo. Esta técnica elimina tanto instancias ruidosas como las fronterizas, obteniendo límites de decisión más nítidos. A su vez las instancias centrales no se ven afectadas por este proceso. El objetivo de este algoritmo a diferencia del anterior no es la reducción del conjunto de datos, sino mejorar la precisión del subconjunto seleccionado. Dicho anteriormente a diferencia de CNN este algoritmo mejora la precisión del subconjunto que se reduce y debido a la capacidad que tiene de limpieza, ha sido utilizado por otros algoritmos para el filtrado de ruido (García et al., 2015), como por ejemplo, DROP3, ICF entre otros, los cuáles no hacen parte de esta tesis.

Algorithm 2: Edited Nearest Neighbours (ENN)

Input: A training set $X = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, the number of nearest neighbours k
Output: The set of selected instances $S \subseteq X$

```
S = X
foreach x ∈ S do
    if x is misclassified using its k nearest neighbours then
        Remove x to S
return S
```

Figura 6: Algoritmo ENN. Reproducción de Tesis Doctoral (Arnaiz González et al., 2018)

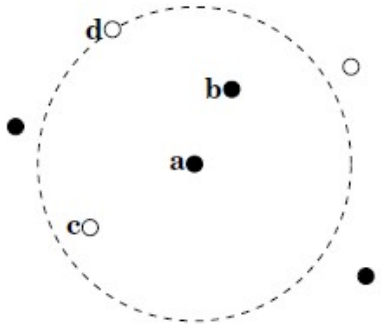

Figura 7: Representación de los vecinos más cercanos y su relación de asociación en un espacio bidimensional, donde cada punto representa una instancia, dos clases diferentes y un parámetro de k = 3. Reproducción de Tesis Doctoral (Arnaiz González et al., 2018)

2.2. Caracterización de los conjuntos de datos y algoritmos de clasificación

En esta sección se describen los conjuntos de datos y métodos (algoritmos) utilizados para realizar los experimentos asistidos por las herramientas WEKA (software libre) , RapidMiner, Keel y Python. Para la obtención de los conjuntos de datos, nos apoyamos en los repositorios de bases de datos de la Universidad de California en Irvine, UCI ML (repositorio habitual para experimentos en machine learning), en el repositorio de Keel.

2.2.1. Conjuntos de datos

La tabla 2 describe los conjuntos de datos seleccionados para el análisis de diferentes algoritmos de selección de instancias, posteriormente el conjunto de datos que dé como resultado el algoritmo de selección de instancias, será previamente analizado con un algoritmo de clasificación. Estos conjuntos de datos tienen grandes volúmenes de datos, fueron previamente seleccionados para esta tesis con el fin de mostrar el desempeño de los algoritmos de selección de instancias.

Tabla 2: Caracterización de los conjuntos de datos adoptados para el análisis de modelos de selección de instancias y clasificación

Datos	Instancias	Atributos	Clases
Pendigits	10992	16	10
Shuttle	58000	9	7
Volcanoes-b6	10130	3	5
Connect-4	67557	42	3
Volcanoes-d3	9285	3	5
Volcanoes-b4	10190	3	5
PishingWebsites	11055	30	2
Electricity	45312	8	2
Elevators	16599	18	2
Artificial-Characters	10218	7	10
Amazon-Employee	32769	9	2

Los conjuntos de datos previamente seleccionados poseen un gran número de datos ya que la temática

central de este estudio es probar los algoritmos de selección de instancias en conjuntos de datos de gran tamaño. Los resultados de cada algoritmo varían cuando se seleccionan conjuntos de datos de menor tamaño, siendo significativamente peores que cuando se utilizan para grandes conjuntos de datos.
La tabla 3 muestra los conjuntos de datos utilizados para realizar selección de instancias con los algoritmos Kennard-Stone y Random sampling method (RS) para luego evaluar su desempeño.

Tabla 3: Conjuntos de datos para selección de instancias con Kennard-Stone y Random sampling method (RS)

Datos	Instancias	Atributos	Clases
Shuttle	58000	9	7
Volcanoes-b6	10130	3	5
Connect-4	67557	42	3
Volcanoes-d3	9285	3	5
Volcanoes-b4	10190	3	5
PishingWebsites	11055	30	2

La tabla 4 muestra los conjuntos de datos utilizados para realizar la selección de instancias con los algoritmos CNN y ENN para luego evaluar su desempeño.

Tabla 4: Conjuntos de datos para selección de instancias con los algoritmos CNN y ENN

Datos	Instancias	Atributos	Clases
Pendigits	10992	16	10
Electricity	45312	8	2
Elevators	16599	18	2
Artificial-Characters	10218	7	10
Amazon-Employee	32769	9	2

2.2.2. Algoritmos de clasificación

En esta subsección describimos en la tabla 5 los algoritmos de clasificación empleados con el fin de realizar una comparativa, estos serán empleados al final del proceso de selección de instancias siguiendo la literatura según (García et al., 2015). Estos algoritmos fueron implementados en WEKA, sin el ajuste de hiperparámetros para las tareas de clasificación y regresión lineal.

Tabla 5: Algoritmos de clasificación

Algoritmos	Descripción	Plataforma
Simple Logistic (SL)	Es un clasificador que construye modelos de regresión logística lineal.	Weka
SMO	Este clasificador implementa una máquina vector soporte con un algoritmo de optimización mínima.	Weka
Random Tree (RT)	Arbol de decisión sin poda que considera k atributos elegidos de manera aleatoria en cada nodo.	Weka
Random Forest (RF)	Es un conjunto de arboles de decisión combinados con bagging.	Weka

3. Metodología y experimentación

Los algoritmos de selección de instancias tienen como función reducir el tamaño de los conjuntos de datos y seleccionar el subconjunto que sea más representativo posible. Cuando se realiza selección de instancias hay que considerar que son problemas de múltiple objetivo. Por un lado, se debe tener en cuenta la reducción, y por el otro, la precisión. Sin embargo, según la literatura y (Leyva et al., 2015) estas dos suelen ir en direcciones opuestas. Esta sección presenta la metodología más utilizada comúnmente según la literatura de selección de instancias.

3.1. Validación del modelo

Cuando se estima la precisión del modelo predictivo con un clasificador o regresor esto es necesario para evaluar los algoritmos de selección de instancias. La precisión del modelo debe estimarse de manera correcta, es necesario seleccionar un método de estimación con baja varianza y sesgo (Kohavi et al., 1995). Según la literatura la razón por la cual hay varios métodos es porque no todos los métodos son adecuados según las condiciones en las que se presente el estudio (Schaffer, 1994). Teniendo en cuenta la literatura con (García et al., 2015), antes de aplicar nuestra técnica de selección de instancias, debemos entrenar nuestro conjunto de datos, el cual podemos observar en la figura 8, que a su vez fue el método aplicado en los conjuntos de datos.

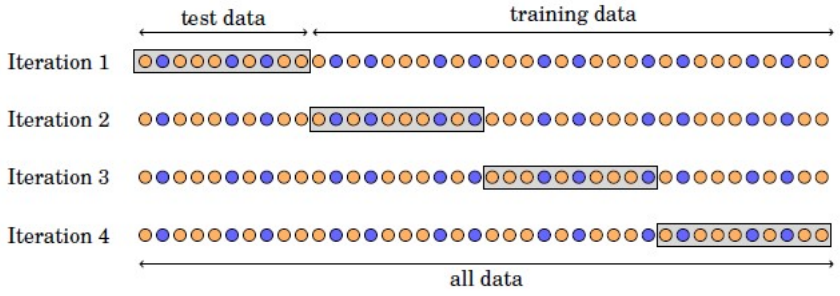

Figura 8: Diagrama de número de k-fold cross-validation (Validación cruzada) con k=4. Donde cada punto representa una instancia y el color de cada punto define la clase a la que pertenece

3.2. Holdout

La estimación Holdout, o test de estimación de la muestra, se realiza de forma sencilla, dividiendo el conjunto de datos original en dos subconjuntos distintos. Se utiliza un conjunto de entrenamiento y un conjunto de prueba (Holdout). La configuración común para este método es utilizar dos de las tres partes del conjunto de datos original para el entrenamiento y, la parte restante para pruebas. Su principal problema es que los datos no pueden estar adecuados para el entrenamiento, ya sea por falta de la tercera parte del conjunto de datos original.

3.3. Cross-validation (Validación cruzada)

Comúnmente en la literatura se refiere a k-fold cross-validation o validación cruzada, este método divide el conjunto de datos original en k conjuntos o fold de tamaño aproximadamente igual. Se entrena k número de veces, utilizando k – 1 fold para entrenamiento, y el fold restante es utilizado para prueba (test). Los conjuntos de entrenamiento y prueba son intercambiados a través de la ejecución como se

puede observar en la figura 8. Una completa validación cruzada (Cross-validation) requiere el máximo numero posible de combinaciones para ser probadas, lo cual es demasiado costoso para aplicar en la práctica, un número que es comúnmente utilizado en la práctica es 10 folds (McLachlan et al., 2005).

3.4. Experimentación y resultados

3.4.1. Kennard-stone y random sampling (RS)

El objetivo de este Trabajo Fin de Máster es comparar el desempeño de diferentes técnicas de selección de instancias en grandes conjuntos de datos. Se inicia la parte experimental con los conjuntos de datos originales mencionados en la tabla 2. Teniendo en cuenta la literatura citada anteriormente, los experimentos usaron una validación cruzada de k=4. La primera etapa dará el resultado de la aplicación de las técnicas de selección de instancias Kennard-stone y Random sampling, las cuales las vamos a describir en dos fases experimentales. La primera fase experimental consiste en reducir el conjunto de datos entrenados mediante la aplicación de las técnicas mencionadas en 5000 samples (muestras) para posteriormente construir el modelo de clasificación los cuales se probaron con los clasificadores mencionados en la tabla 5, de tal manera que el desempeño de estas técnicas y los resultados presentados se evaluarán con tres métricas significativas como Curvas ROC, Precisión y Coeficiente de Kappa.
La segunda fase de experimentación de la primera etapa dará el resultado de la aplicación de las técnicas de selección de instancias Kennard-stone y Random sampling utilizando un valor de 2500 samples (muestras) para posteriormente construir el modelo de clasificación y evaluar su desempeño mediante los resultados recolectados teniendo en cuenta las métricas y parámetros anteriormente mencionados en la primera fase.
Como anteriormente se mencionó en la literatura, la complejidad que tienen los algoritmos de selección de instancias al momento de utilizarse en conjuntos de datos de enorme tamaño, se hizo presente en este Trabajo Fin de Máster. Anteriormente en la subsección 2.1.4 se acotó sobre la limitación computacional y en este caso no es diferente, por tal razón se decidió dividir en dos fases los experimentos de la primera etapa. En este caso se comparó el desempeño entre Kennard-stone y Random sampling con el fin de analizar y descubrir que algoritmo presentaba un mejor desempeño.
En la primera fase, como se planteó en la tabla 6, se muestran los conjuntos de datos entrenados utilizados con un número de samples (muestras) para poder observar su desempeño con esta cantidad. En la segunda fase experimental como se muestra en la tabla 7 se redujo aún más la cantidad de samples (muestras) con el propósito de reducir lo mayor posible el conjunto de datos entrenados buscando un mejor desempeño y mayor facilidad al momento de manipular grandes conjuntos de datos, con el fin de encontrar el subconjunto que se asimile en su desempeño al conjunto de datos original.

Tabla 6: Conjuntos de datos fase experimental 1.

Datos	Instancias	Atributos	Clases
Shuttle	5000	9	7
Volcanoes-b6	5000	3	5
Connect-4	5000	42	3
Volcanoes-d3	5000	3	5
Volcanoes-b4	5000	3	5
PishingWebsites	5000	30	2

Tabla 7: Conjuntos de datos fase experimental 2.

Datos	Instancias	Atributos	Clases
Shuttle	2500	9	7
Volcanoes-b6	2500	3	5
Connect-4	2500	42	3
Volcanoes-d3	2500	3	5
Volcanoes-b4	2500	3	5
PishingWebsites	2500	30	2

Tabla 8: Resultados de las métricas en los datos originales

Datos	Métricas	Clasificadores			
		SL	SMO	RT	RF
Shuttle	Curva ROC	0,991	0,953	0,999	1,000
	Precisión	96,103	96,841	99,951	99,972
	Coeff. Kappa	0,886	0,910	0,998	0,999
Volcanoes-b6	Curva ROC	0,917	0,500	0,684	0,890
	Precisión	96,565	96,170	94,394	96,446
	Coeff. Kappa	0,294	0,000	0,252	0,350
Connect-4	Curva ROC	0,857	0,723	0,727	0,934
	Precisión	75,861	76,074	70,242	81,924
	Coeff. Kappa	0,445	0,451	0,402	0,583
Volcanoes-d3	Curva ROC	0,500	0,503	0,585	0,789
	Precisión	94,401	94,401	89,750	94,272
	Coeff. Kappa	0,000	0,000	0,114	0,126
Volcanoes-b4	Curva ROC	0,846	0,500	0,678	0,783
	Precisión	96,477	96,193	94,152	96,153
	Coeff. Kappa	0,297	0,000	0,259	0,292
PishingWebsites	Curva ROC	0,986	0,935	0,967	0,995
	Precisión	93,777	93,704	95,513	97,141
	Coeff. Kappa	0,873	0,872	0,909	0,941

Tabla 9: Resultados de las métricas en los datos con Kennard-stone fase 1.

Datos	Métricas	Clasificadores			
		SL	SMO	RT	RF
Shuttle	Curva ROC	0,990	0,825	1,000	1,000
	Precisión	93,144	92,193	99,965	99,965
	Coeff. Kappa	0,787	0,747	0,999	0,999
Volcanoes-b6	Curva ROC	0,915	0,500	0,672	0,869
	Precisión	96,683	96,170	93,880	96,446
	Coeff. Kappa	0,334	0,000	0,222	0,350
Connect-4	Curva ROC	0,779	0,666	0,621	0,820
	Precisión	71,610	73,120	59,698	71,953
	Coeff. Kappa	0,271	0,329	0,201	0,291
Volcanoes-d3	Curva ROC	0,500	0,503	0,607	0,787
	Precisión	94,401	94,401	90,482	94,272
	Coeff. Kappa	0,000	0,000	0,157	0,141
Volcanoes-b4	Curva ROC	0,840	0,500	0,696	0,809
	Precisión	96,350	96,193	93,759	96,271
	Coeff. Kappa	0,279	0,000	0,249	0,354
PishingWebsites	Curva ROC	0,986	0,940	0,967	0,996
	Precisión	93,668	94,030	95,694	96,924
	Coeff. Kappa	0,871	0,879	0,913	0,937

Tabla 10: Resultados de las métricas en los datos con Random sampling fase 1.

Datos	Métricas	Clasificadores			
		SL	SMO	RT	RF
Shuttle	Curva ROC	0,991	0,922	0,999	1,000
	Precisión	96,172	95,600	99,862	99,841
	Coeff. Kappa	0,890	0,8713	0,996	0,995
Volcanoes-b6	Curva ROC	0,915	0,501	0,663	0,862
	Precisión	96,683	96,170	94,196	96,328
	Coeff. Kappa	0,348	0,000	0,229	0,319
Connect-4	Curva ROC	0,846	0,712	0,654	0,864
	Precisión	75,151	75,127	63,516	75,447
	Coeff. Kappa	0,426	0,427	0,266	0,398
Volcanoes-d3	Curva ROC	0,500	0,503	0,568	0,773
	Precisión	94,401	94,401	90,654	94,099
	Coeff. Kappa	0,000	0,000	0,106	0,115
Volcanoes-b4	Curva ROC	0,846	0,500	0,659	0,759
	Precisión	96,350	96,193	94,270	96,036
	Coeff. Kappa	0,257	0,000	0,227	0,300
PishingWebsites	Curva ROC	0,986	0,934	0,954	0,993
	Precisión	93,596	93,632	94,066	96,707
	Coeff. Kappa	0,870	0,870	0,879	0,933

Tabla 11: Resultados de las métricas en los datos con Kennard-stone fase 2.

Datos	Métricas	Clasificadores			
		SL	SMO	RT	RF
Shuttle	Curva ROC	0,986	0,824	0,998	1,000
	Precisión	92,951	92,206	99,855	99,958
	Coeff. Kappa	0,776	0,747	0,9959	0,998
Volcanoes-b6	Curva ROC	0,914	0,500	0,675	0,847
	Precisión	96,525	96,170	94,551	96,210
	Coeff. Kappa	0,252	0,000	0,262	0,355
Connect-4	Curva ROC	0,768	0,580	0,598	0,771
	Precisión	68,709	67,584	59,005	67,383
	Coeff. Kappa	0,168	0,085	0,164	0,106
Volcanoes-d3	Curva ROC	0,500	0,503	0,611	0,784
	Precisión	94,401	94,401	88,501	94,099
	Coeff. Kappa	0,000	0,000	0,119	0,065
Volcanoes-b4	Curva ROC	0,840	0,500	0,688	0,778
	Precisión	96,350	96,193	94,270	96,193
	Coeff. Kappa	0,279	0,000	0,274	0,318
PishingWebsites	Curva ROC	0,985	0,937	0,922	0,991
	Precisión	93,668	93,596	90,810	95,152
	Coeff. Kappa	0,872	0,870	0,815	0,902

Tabla 12: Resultados de las métricas en los datos con Random sampling fase 2.

Datos	Métricas	Clasificadores			
		SL	SMO	RT	RF
Shuttle	Curva ROC	0,992	0,900	0,995	1,000
	Precisión	95,910	94,744	99,675	99,751
	Coeff. Kappa	0,881	0,843	0,999	0,993
Volcanoes-b6	Curva ROC	0,914	0,500	0,674	0,861
	Precisión	96,604	96,170	94,236	96,249
	Coeff. Kappa	0,313	0,000	0,227	0,304
Connect-4	Curva ROC	0,835	0,703	0,626	0,837
	Precisión	74,529	74,706	62,184	73,380
	Coeff. Kappa	0,407	0,415	0,223	0,327
Volcanoes-d3	Curva ROC	0,500	0,503	0,593	0,777
	Precisión	94,401	94,401	91,214	94,272
	Coeff. Kappa	0,000	0,000	0,140	0,063
Volcanoes-b4	Curva ROC	0,849	0,500	0,661	0,793
	Precisión	96,350	96,193	93,720	95,957
	Coeff. Kappa	0,257	0,000	0,212	0,285
PishingWebsites	Curva ROC	0,985	0,936	0,913	0,990
	Precisión	93,379	93,813	90,846	95,984
	Coeff. Kappa	0,865	0,874	0,814	0,918

3.4.2. Condensed Nearest Neighbor (CNN) y Edited Nearest Neighbor (ENN)

Como etapa dos de la sección de experimentación y resultados, seleccionamos los conjuntos de datos mencionados en la tabla 4, a los que se les aplicará las técnicas de selección de instancias propuestas en esta segunda etapa.

Esta segunda etapa de la investigación se ejecutó CNN y ENN para poder observar el desempeño de cada uno en estos. En esta segunda etapa se busca determinar cuál de las dos técnicas reduce significativamente el número de instancias, dando como resultado un subconjunto de datos de menor tamaño. Este subconjunto que de como resultado, se puede utilizar en futuras investigaciones para un modelo de clasificación, el cual en esta etapa no se construirá, ya que el objetivo de la etapa dos es determinar que técnica tiene mejor desempeño.

La segunda etapa de experimentación generó resultados en gráficos de dispersión, con la finalidad de analizar visualmente las dos técnicas de selección de instancias.

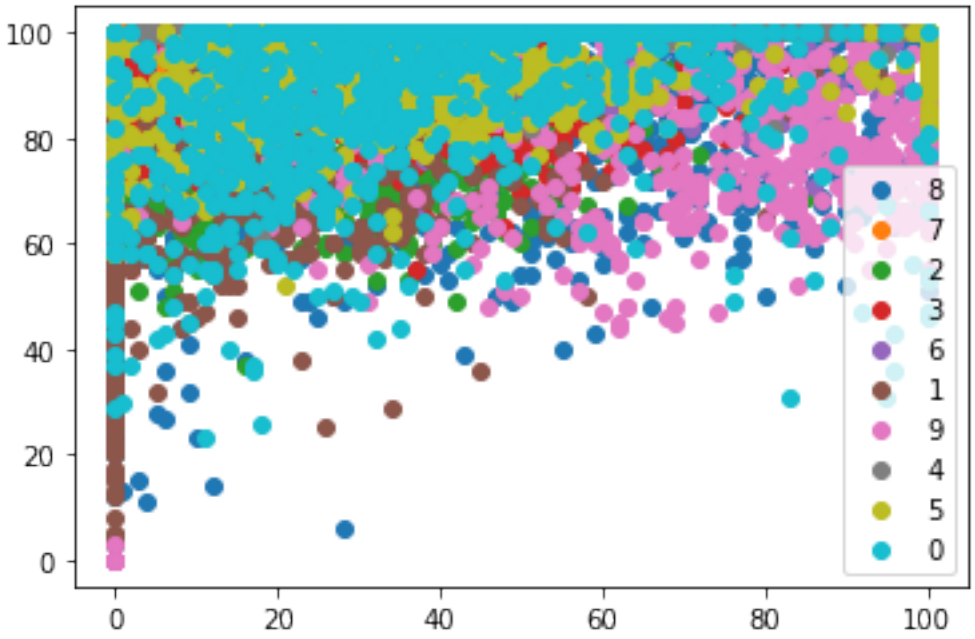

Figura 9: Instancias de Pendigits

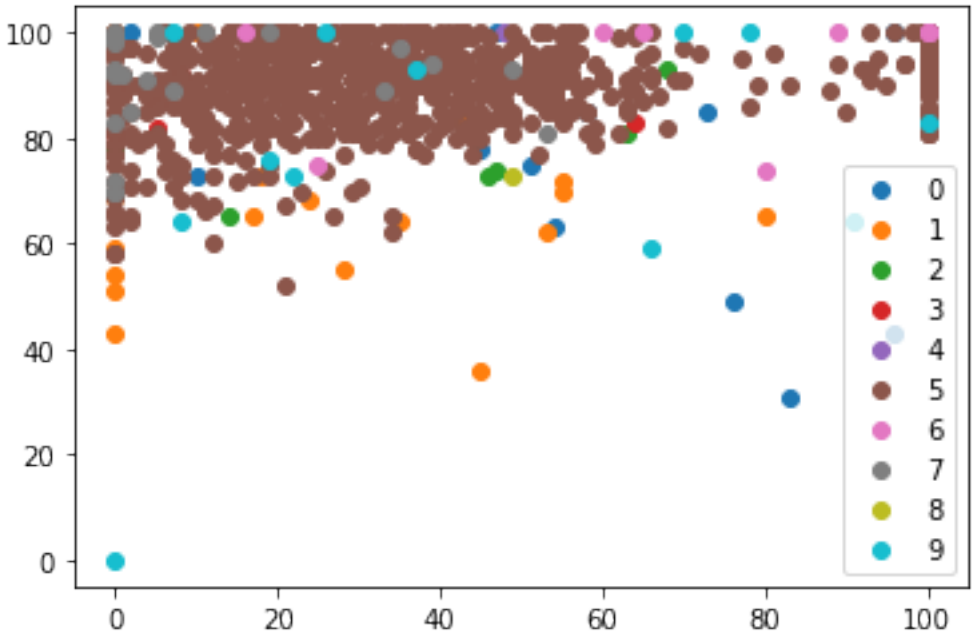

Figura 10: Instancias de Pendigits con CNN

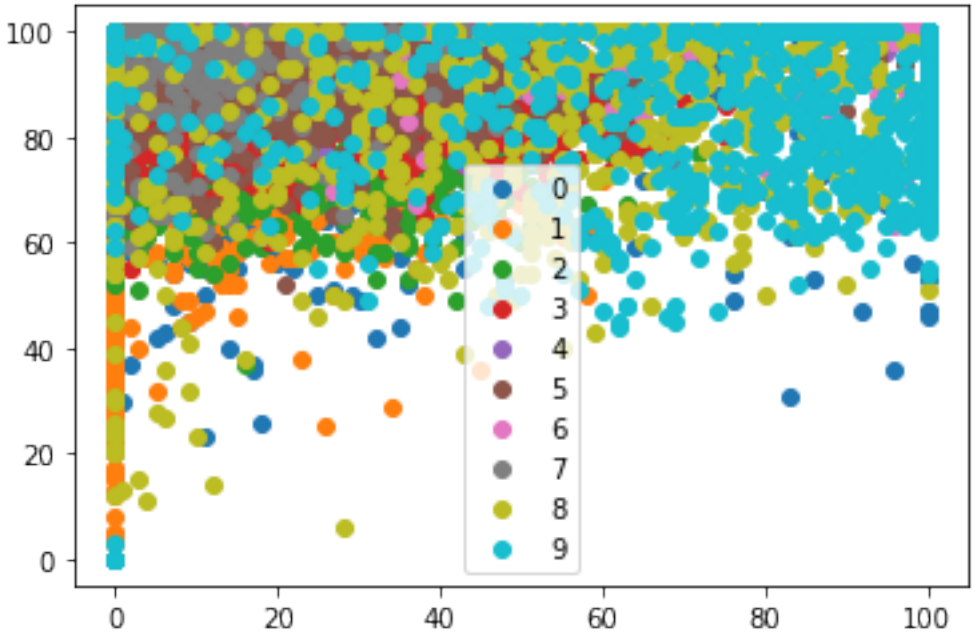

Figura 11: Instancias de Pendigits con ENN

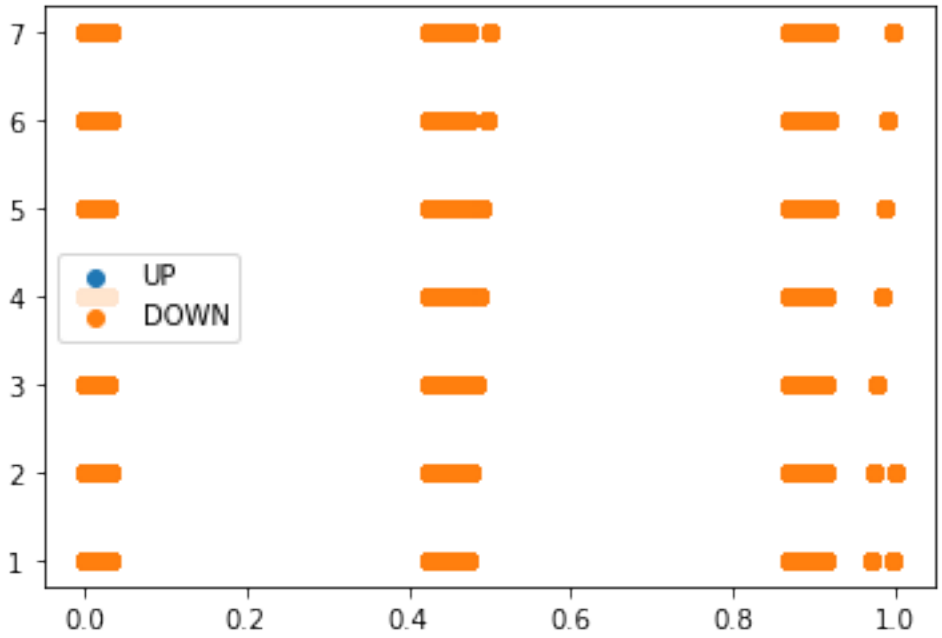

Figura 12: Instancias de Eletricity

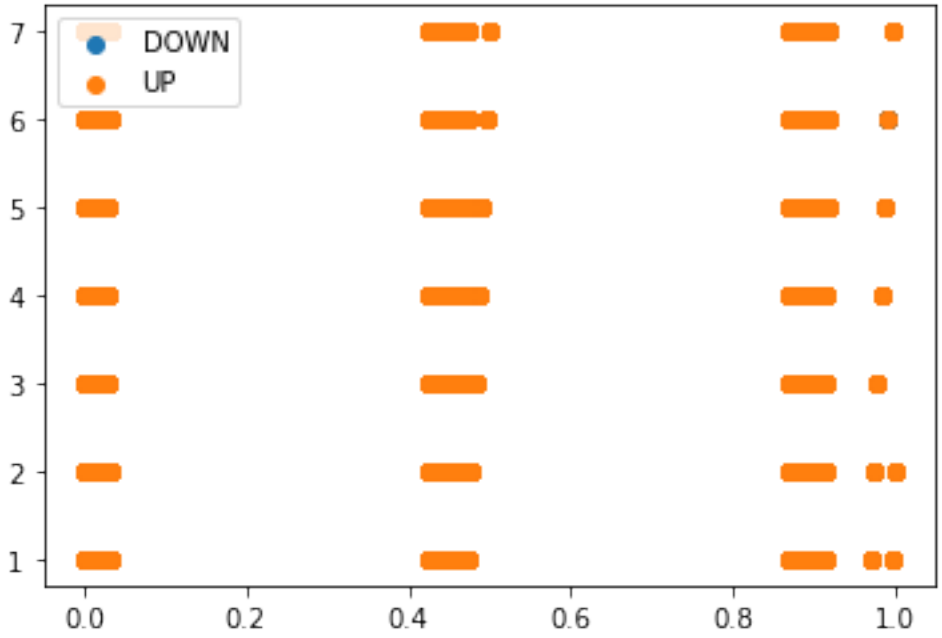

Figura 13: Instancias de Eletricity con CNN

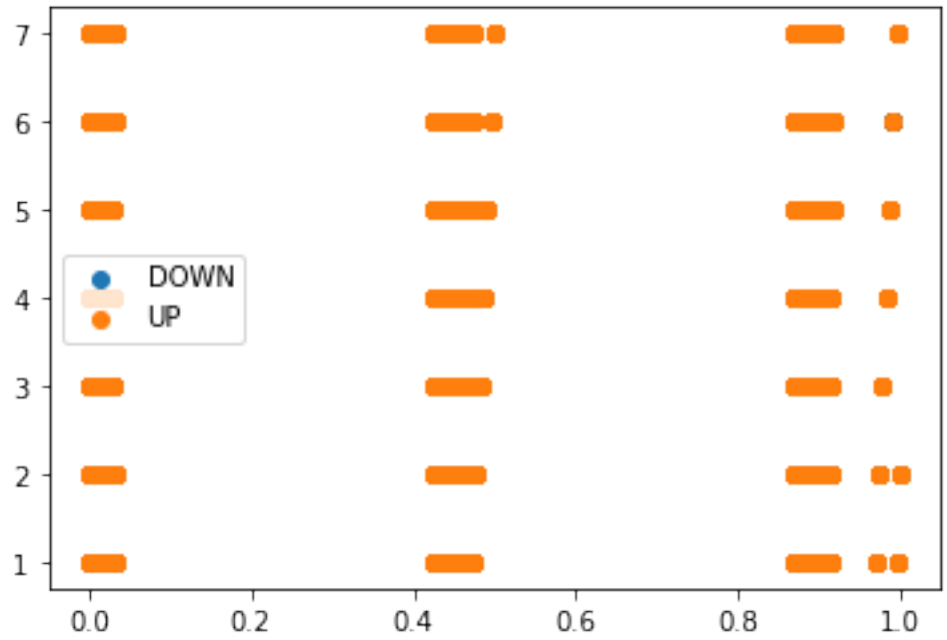

Figura 14: Instancias de Eletricity con ENN

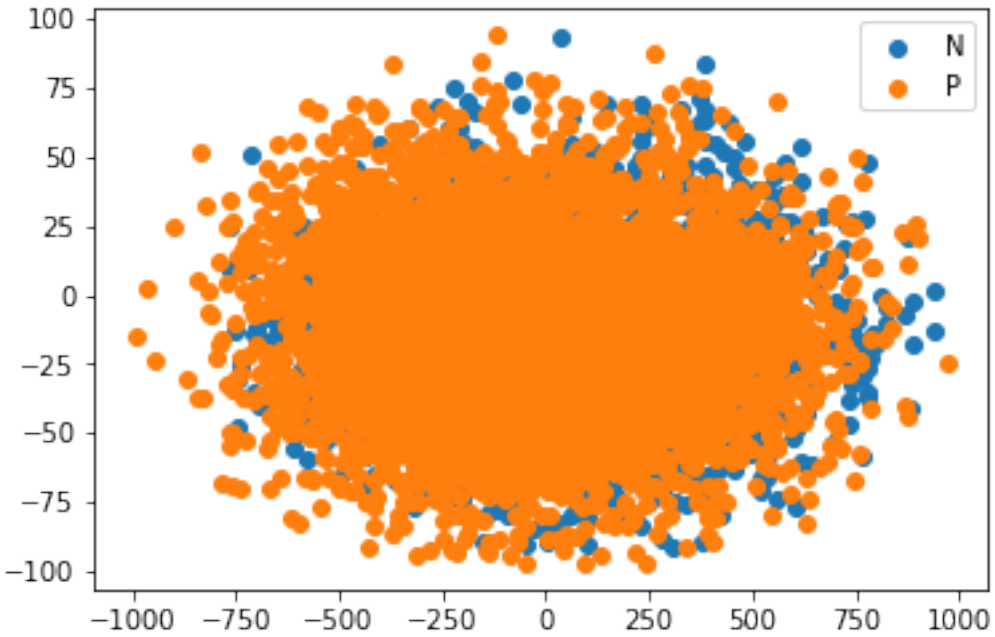

Figura 15: Instancias de Elevators

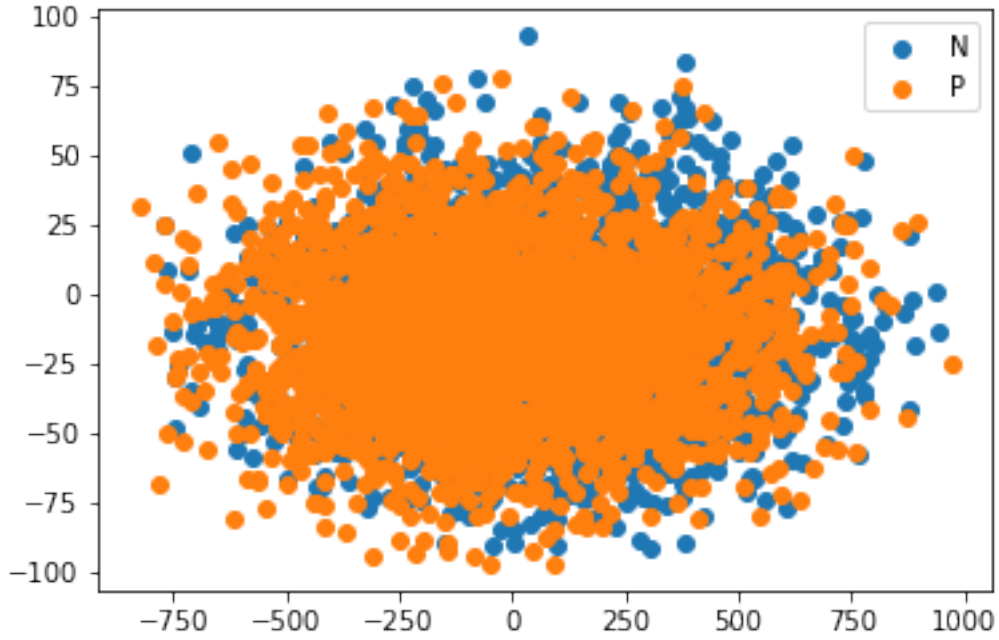

Figura 16: Instancias de Elevators con CNN

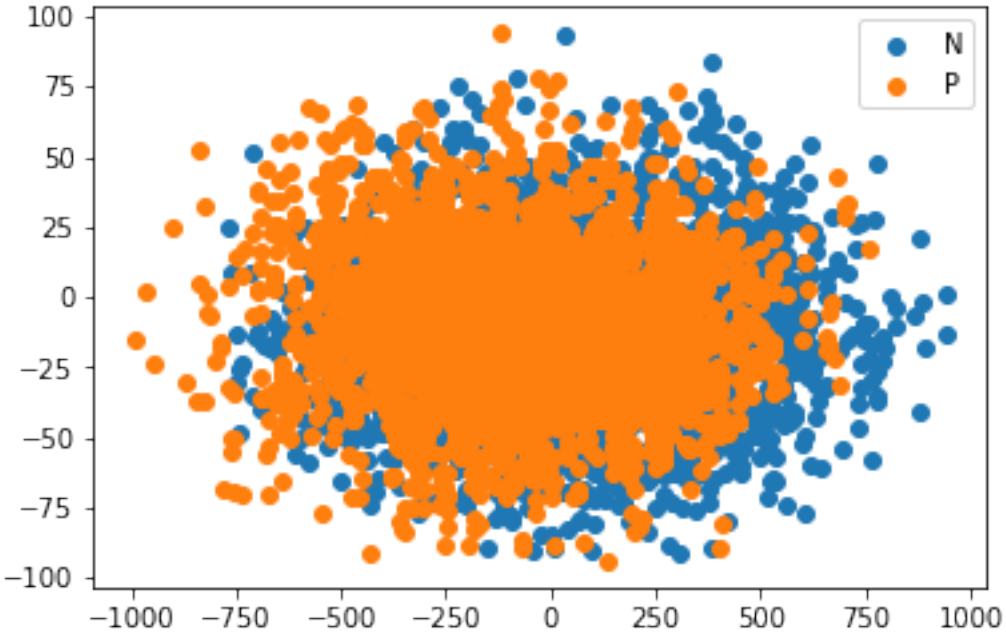

Figura 17: Instancias de Elevators con ENN

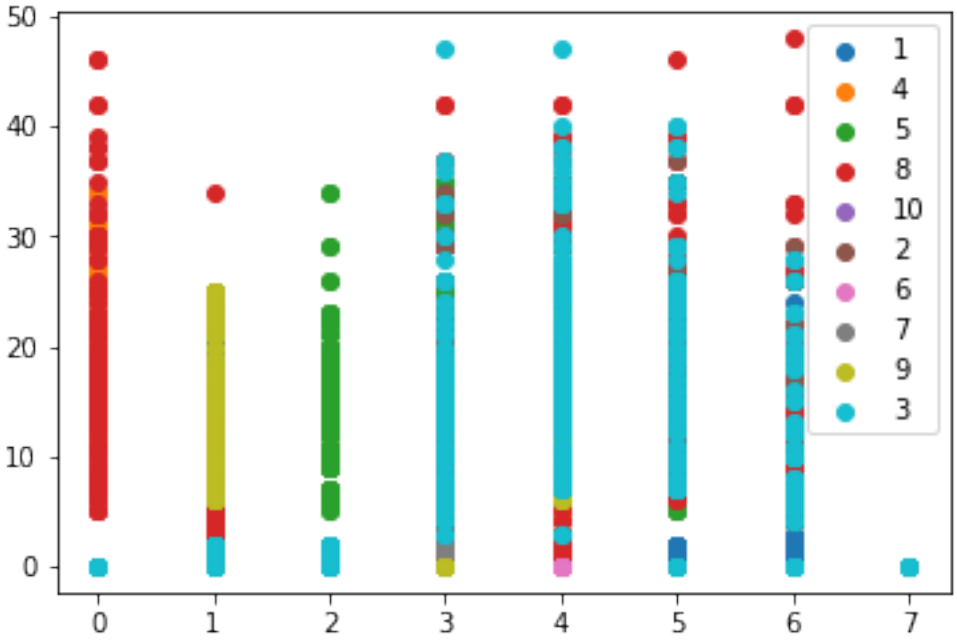

Figura 18: Instancias de Artificial-Characters

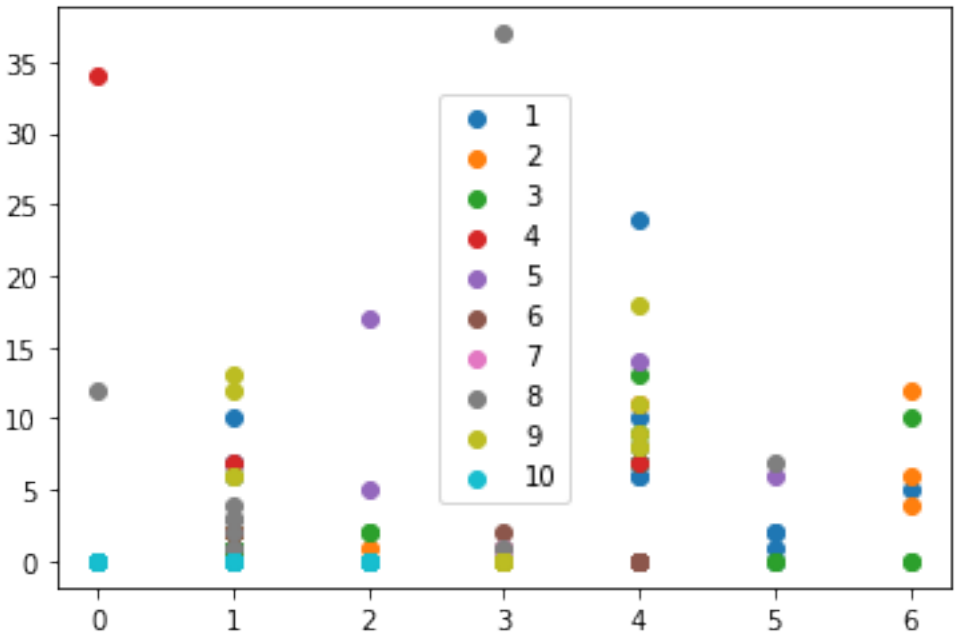

Figura 19: Instancias de Artificial-Characters con CNN

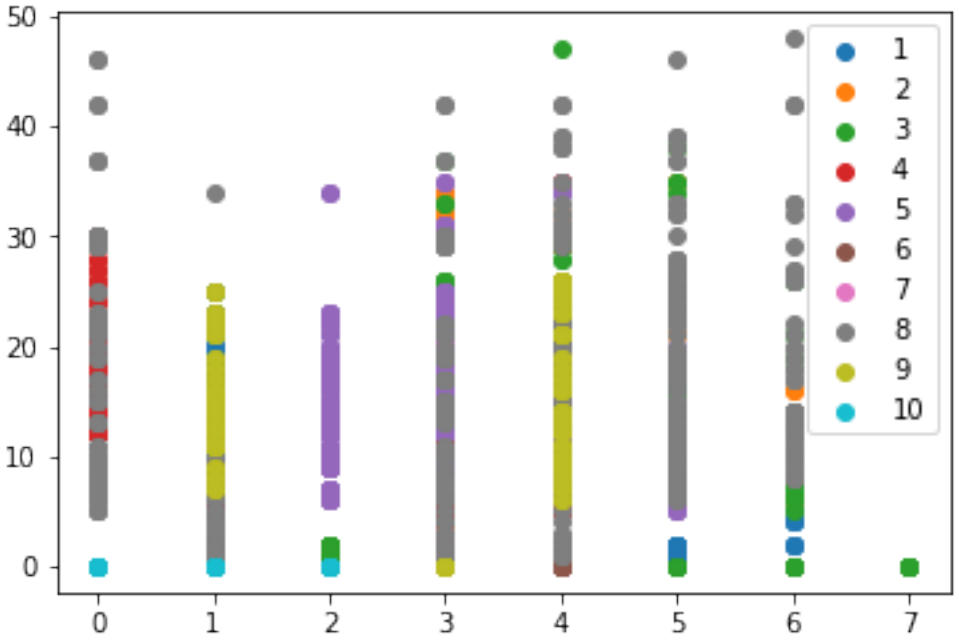

Figura 20: Instancias de Artificial-Characters con ENN

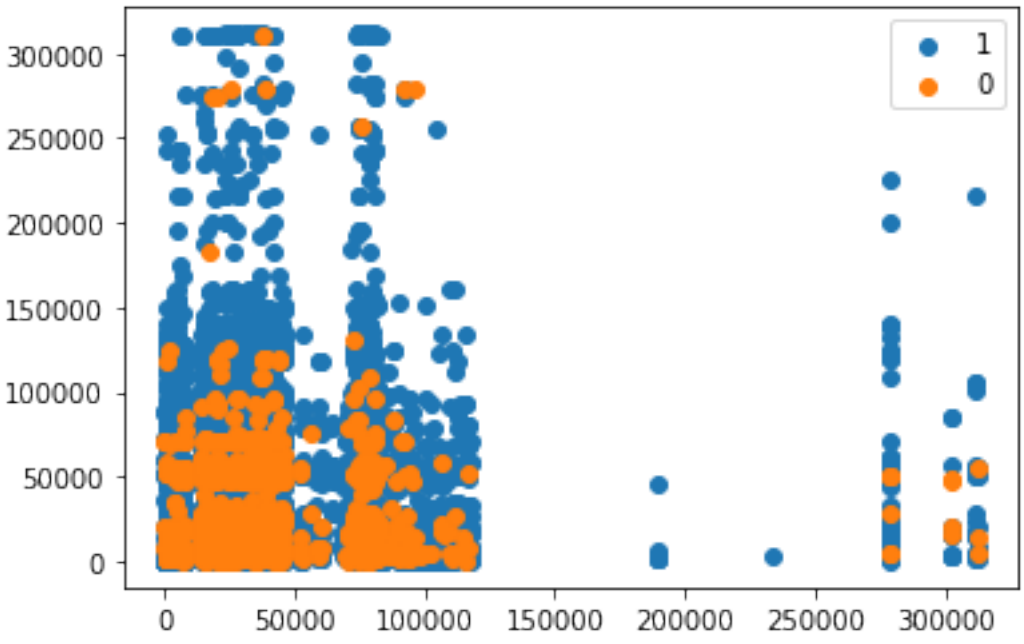

Figura 21: Instancias de Amazon-Employee

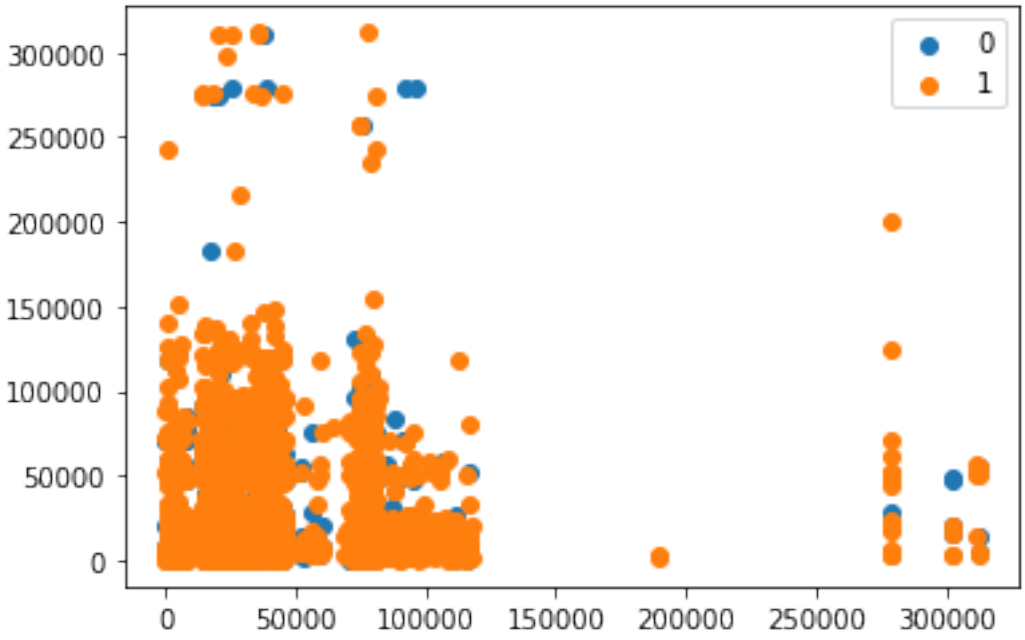

Figura 22: Instancias de Amazon-Employee con CNN

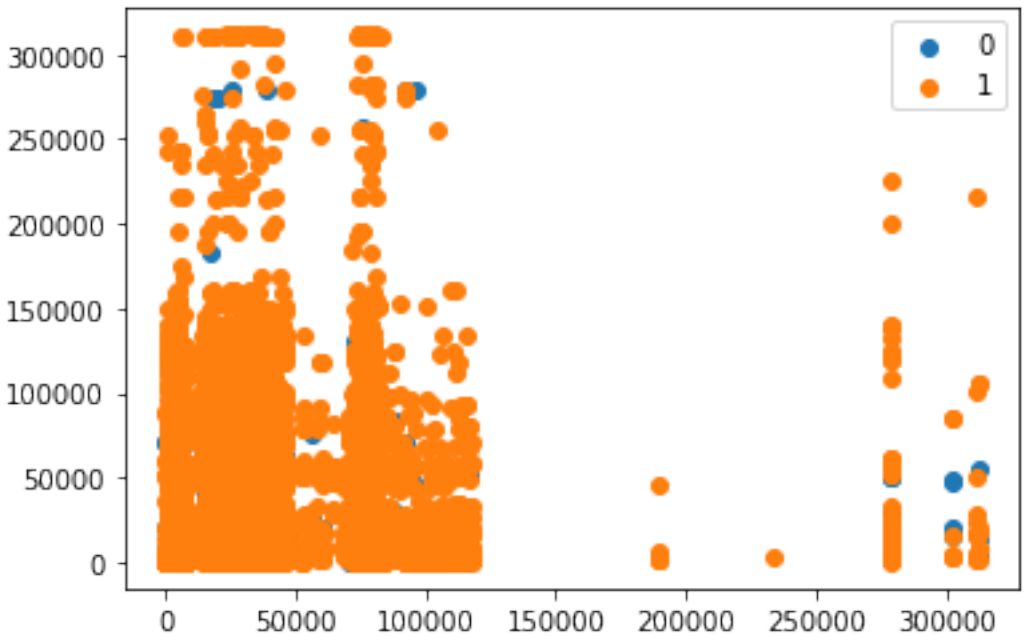

Figura 23: Instancias de Amazon-Employee con ENN

La tabla 13 y 14 muestra los conjuntos de datos utilizados luego de ser sometidos a las técnicas de selección de instancias CNN y ENN.

Tabla 13: Conjuntos de datos aplicados con CNN

Datos	Instancias	Atributos	Clases
Pendigits	888	16	10
Electricity	21332	8	2
Elevators	7217	18	2
Artificial-Characters	1432	7	10
Amazon-Employee	5381	9	2

Tabla 14: Conjuntos de datos aplicados con ENN

Datos	Instancias	Atributos	Clases
Pendigits	8101	16	10
Electricity	25972	8	2
Elevators	7716	18	2
Artificial-Characters	3687	7	10
Amazon-Employee	21737	9	2

3.5. Análisis de resultados

3.5.1. Kennard-stone y random sampling (RS)

Como primera etapa de nuestra fase de experimentación nos enfocamos en la comparación de dos técnicas de muestreo. Los samples(muestras) de los conjuntos de datos pueden ser divididos en tres subconjuntos de datos: calibración, validación y predicción. La calibración y validación son utilizados para la construcción del modelo, mientras que el subconjunto de predicción (Gareth et al., 2013) es utilizado en pruebas de test para medir su habilidad y capacidad que tiene para la predicción (Galvao et al., 2005) Basándonos en la literatura previamente (Lee et al., 2018), hay diferentes maneras de obtener los subconjuntos de calibración y validación, en nuestro caso como hicimos mención previamente en la figura 8 se utilizó una validación cruzada k = 4.

Esta tesis evaluó el desempeño de estas técnicas en distintos conjuntos de datos con el fin de demostrar cual de las dos tiene mejor rendimiento para uso y aplicación en reducción de instancias y posteriormente en la construcción de modelo de clasificación (Galvao et al., 2005).

Como parte inicial de nuestra experimentación se probaron todos los conjuntos de datos en la construcción de los modelos de clasificación, a partir de esta premisa se empezó a simular experimentos buscando reducir las instancias de cada conjunto de datos con el fin de encontrar un subconjunto de datos con mayor representatividad al conjunto de datos original, en pocas palabras es seleccionar las muestras uniformemente distribuidas y capaces de obtener una variabilidad existente. En este Trabajo Fin de Máster se muestra y compara los resultados divididos en dos fases.

En las tablas 9 y 10 de la fase 1, se demuestra que los resultados con un número de muestras definido para ambas técnicas son similares, se hace una mención que Kennard-stone es muy utilizado en el área de la quimiometría (Martins et al., 2010) y Random sampling una técnica estadística aplicada en diversas áreas (Akdemir et al., 2015).

En las tablas 11 y 12 de la fase 2, se observa también que los resultados con un número de muestras definido previamente para ambas técnicas son similares, variando en muy poco la diferencia de uno sobre otro.

Basándonos en los resultados obtenidos y en la literatura (Ferreira et al., 2021), la técnica Kennard-stone supera en desempeño de selección de instancias y construcción del modelo de clasificación a la técnica Random sampling, como anotación importante se puede decir que a pesar de que el desempeño no se evidencie ampliamente en los 4 clasificadores utilizados, el desempeño es mucho mejor utilizando Kennard-stone en vez de Random sampling es evidente.

3.5.2. Condensed Nearest Neighbor (CNN) y Edited Nearest Neighbor (ENN)

Cuando se habla de reducción de instancias no se puede dejar a un lado a las técnicas Condensed Nearest Neighbor (CNN) y Edited Nearest Neighbor (ENN), las cuales han sido pilares para investigaciones y experimentos a lo largo de los años, en esta tesis la finalidad fue demostrar cual de las dos técnicas

tenia mejor desempeño al momento de encontrar un subconjunto con mayor similitud al conjunto de datos original.

En la sección 3.4.2 se muestran los experimentos realizados cuyos resultados muestran algo interesante a la hora de reducir instancias de grandes conjuntos de datos. CNN es una técnica de submuestreo que busca un subconjunto de una colección de muestras que no suponga ninguna pérdida de rendimiento denominado conjunto mínimo consistente (Hart, 1968).

Revisando los resultados obtenidos en las gráficas de dispersión de CNN podemos ver que la técnica se centra en las instancias de clase minoritaria a lo largo de la frontera de decisión entre las dos clases, específicamente en las instancias mayoritarias alrededor de las instancias de la clase minoritaria
Esto nos manifiesta que, aunque el argumento trata de equilibrar la distribución de clases, la técnica seguirá agregando instancias mal clasificadas al conjunto de datos transformado.

Por otro lado, tenemos a la técnica ENN, técnica que consiste en una regla para encontrar ejemplos ambiguos y ruidosos en el conjunto de datos. Esta regla utiliza k = 3 vecino más cercanos para localizar aquellas instancias en el conjunto de datos que estén mal clasificados y que se eliminen antes de aplicar una regla de clasificación k = 1.

El funcionamiento de ENN es muy diferente a CNN, teniendo en cuenta que cuando se utiliza ENN para realizar un submuestreo, se aplica la regla a cada instancia de la clase mayoritaria, lo que permite eliminar las instancias clasificadas erróneamente que pertenezcan a la clase minoritaria y mantener las instancias correctamente clasificadas.

Por tal razón, al momento de analizar lo anteriormente expuesto y las gráficas de dispersión podemos observar que CNN reduce considerablemente el número de instancias del conjunto de datos original, permitiendo reducir la capacidad de memoria al momento de utilizarse, por otro lado ENN tiene un funcionamiento totalmente diferente a CNN, ya que CNN funciona de tal manera de selección de muestreo aleatorio y va agregando instancias mal clasificadas al subconjunto de datos, repitiendo esto hasta que no quede ninguna mal clasificada. En cambio ENN permite mantener las instancias correctamente clasificadas y eliminando las mal clasificadas, por tal razón, cuando se observa las gráficas de dispersión, CNN tiene un mejor desempeño a la hora de reducir instancias y encontrar un subconjunto de datos que se pueda utilizar para construir el modelo de clasificación. Si nos referimos a ENN se considera técnica de selección de instancias pero su objetivo no es totalmente reducir las instancias y obtener un subconjunto de datos mínimo, sino prevalecer las instancias correctamente clasificadas obteniendo así una mejor precisión. Como resultado del desempeño de estas dos técnicas podemos deducir empíricamente que la técnica CNN tiene un mejor desempeño ampliamente al momento de reducir instancias en comparación a ENN. Como centro de nuestra investigación, el objetivo fue evidenciar cuál de estas dos tenía mejor desempeño reduciendo instancias en grandes conjuntos de datos. A pesar de que ENN nos puede dar mejores métricas cuando se intente obtener nuestro subconjunto y posteriormente construir un modelo de clasificación con instancias mejor clasificadas, no era parte de nuestra sección de investigación. A lo largo de los años podemos observar que van surgiendo nuevas técnicas según los avances de la tecnología y la información, y la capacidad de

generarse en la actualidad grandes volúmenes de datos hacen que sea un reto manejar estos grandes conjuntos de datos. Por tal razón se consideraría CNN como solución para reducir estos conjuntos de datos y construir un subconjunto con similitud al original que permita un mejor manejo, rendimiento y reducir el impacto del problema computacional hablado anteriormente en la sección 2.1.4, donde se expone previamente estos problemas y según (Schaffer, 1994), la razón de que existan varias técnicas es porque todas ellas fallan bajo ciertas condiciones.

4. Conclusiones

Como resultado de este estudio empírico mediante la aplicación de técnicas de selección de instancias en conjuntos de datos de distintos tipos y diversos tamaños, hemos podido observar que el avance exponencial de la generación de datos hace necesario encontrar y aplicar técnicas que permitan reducir a un número considerable los datos que permitan su manipulación sin perder la capacidad de una adecuada clasificación al momento de la construcción del modelo.

Se ha aplicado cuatro tipos diferentes de técnicas de selección de instancias, de las cuales Kennard-stone y Random sampling han construido modelos de clasificación con la finalidad de encontrar empíricamente resultados que permitan a futuras investigaciones centrarse en una técnica e iniciar su reducción y manipulación de datos. Las técnicas que a futuro se elijan en distintas investigaciones han de tener una guía y objetivo claro del investigador que quiera seguir aportando a este campo de investigación, ya que se debe de tener en cuenta que estas técnicas varían según las condiciones a que se sometan. Por tal motivo se seleccionaron estas técnicas, con el fin de demostrar y aportar una nueva investigación en campos no tan similares entre sí y con distintos usos anteriormente no aplicados. Al realizar un estudio empírico se permite a futuros investigadores iniciar y centrar sus investigaciones en técnicas de reducción de instancias de una manera más fácil y con distintos campos utilizados para ampliar el espectro de uso y logrando que siga creciendo la línea de investigación.

Referencias

Akdemir, D., Sanchez, J. I., and Jannink, J.-L. (2015). Optimization of genomic selection training populations with a genetic algorithm. *Genetics Selection Evolution*, 47(1):1–10.

Arnaiz González, Á. et al. (2018). Estudio de métodos de selección de instancias.

Barandela, R., Ferri, F. J., and Sánchez, J. S. (2005). Decision boundary preserving prototype selection for nearest neighbor classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(06):787–806.

Bellman, R. and Kalaba, R. (1959). On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9.

Benavoli, A., Corani, G., and Mangili, F. (2016). Should we really use post-hoc tests based on mean-ranks? *The Journal of Machine Learning Research*, 17(1):152–161.

Brereton, R. G. and Lloyd, G. R. (2010). Support vector machines for classification and regression. *Analyst*, 135(2):230–267.

Brighton, H. and Mellish, C. (2002). Advances in instance selection for instance-based learning algorithms. *Data mining and knowledge discovery*, 6(2):153–172.

Cano, J. R., Herrera, F., and Lozano, M. (2005). A study on the combination of evolutionary algorithms and stratified strategies for training set selection in data mining. In *Soft Computing: Methodologies and Applications*, pages 271–284. Springer.

Cerezo, P. C. (2004). *Comparación de modelos de curvas ROC para la evaluación de procedimientos estadísticos de predicción en investigación de mercados.* PhD thesis.

Chawla, N. V., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. P. (2004). Learning ensembles from bites: A scalable and accurate approach. *The Journal of Machine Learning Research*, 5:421–451.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Duda, R., Hart, P., and Stork, D. (2001). John wiley & sons. *New York, NY, USA*, 2.

Etikan, I. and Bala, K. (2017). Sampling and sampling methods. *Biometrics & Biostatistics International Journal*, 5(6):00149.

Eugenio, B. D. and Glass, M. (2004). The kappa statistic: A second look. *Computational linguistics*, 30(1):95–101.

Felix, G., Nápoles, G., Falcon, R., Bello, R., and Vanhoof, K. (2018). Performance analysis of granular versus traditional neural network classifiers: Preliminary results. In *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 1–6. IEEE.

Fernandes de Mello, R. and Antonelli Ponti, M. (2018). A brief review on machine learning. *Machine Learning*, pages 1–74.

Ferreira, R. d. A., Teixeira, G., and Peternelli, L. A. (2021). Kennard-stone method outperforms the random sampling in the selection of calibration samples in snps and nir data. *Ciência Rural*, 52.

Galvao, R. K. H., Araujo, M. C. U., José, G. E., Pontes, M. J. C., Silva, E. C., and Saldanha, T. C. B. (2005). A method for calibration and validation subset partitioning. *Talanta*, 67(4):736–740.

Garcia, S., Derrac, J., Cano, J., and Herrera, F. (2012a). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):417–435.

García, S., Fernández, A., Luengo, J., and Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information sciences*, 180(10):2044–2064.

García, S., Luengo, J., and Herrera, F. (2015). *Data preprocessing in data mining*, volume 72. Springer.

García, S., Luengo, J., and Herrera, F. (2016a). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems*, 98:1–29.

Garcia, S., Luengo, J., Sáez, J. A., Lopez, V., and Herrera, F. (2012b). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE transactions on Knowledge and Data Engineering*, 25(4):734–750.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., and Herrera, F. (2016b). Big data preprocessing: methods and prospects. *Big Data Analytics*, 1(1):1–22.

García-Pedrajas, N. and de Haro-García, A. (2014). Boosting instance selection algorithms. *Knowledge-Based Systems*, 67:342–360.

Gareth, J., Daniela, W., Trevor, H., and Robert, T. (2013). *An introduction to statistical learning: with applications in R*. Spinger.

Gates, G. (1972). The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 18(3):431–433.

Gonçalves, L., Subtil, A., Oliveira, M. R., and de Zea Bermudez, P. (2014). Roc curve estimation: An overview. *REVSTAT-Statistical journal*, 12(1):1–20.

Hajian-Tilaki, K. (2013). Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation. *Caspian journal of internal medicine*, 4(2):627.

Hart, P. (1968). The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3):515–516.

Hoo, Z. H., Candlish, J., and Teare, D. (2017). What is an roc curve?

Jankowski, N. and Grochowski, M. (2004). Comparison of instances seletion algorithms i. algorithms survey. In *International conference on artificial intelligence and soft computing*, pages 598–603. Springer.

Kim, S.-W. and Oommen, B. J. (2003). A brief taxonomy and ranking of creative prototype reduction schemes. *Pattern Analysis & Applications*, 6(3):232–244.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Kull, M. and Flach, P. A. (2014). Reliability maps: a tool to enhance probability estimates and improve classification accuracy. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 18–33. Springer.

Lee, L. C., Liong, C.-Y., and Jemain, A. A. (2018). Iterative random vs. kennard-stone sampling for ir spectrum-based classification task using pls2-da. In *AIP Conference Proceedings*, volume 1940, page 020116. AIP Publishing LLC.

Leyva, E., González, A., and Pérez, R. (2015). Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective. *Pattern Recognition*, 48(4):1523–1537.

Li, T., Fong, S., Wu, Y., and Tallón-Ballesteros, A. J. (2020). Kennard-stone balance algorithm for time-series big data stream mining. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 851–858. IEEE.

Li, Y., Li, T., and Liu, H. (2017). Recent advances in feature selection and its applications. *Knowledge and Information Systems*, 53(3):551–577.

Liu, H. and Motoda, H. (2002). On issues of instance selection. *Data Mining and Knowledge Discovery*, 6(2):115.

Mahmud, M. S., Huang, J. Z., Salloum, S., Emara, T. Z., and Sadatdiynov, K. (2020). A survey of data partitioning and sampling methods to support big data analysis. *Big Data Mining and Analytics*, 3(2):85–101.

Marchiori, E. (2008). Hit miss networks with applications to instance selection.

Martins, M. N., Galvão, R. K., and Pimentel, M. F. (2010). Multivariate calibration transfer employing variable selection and subagging. *Journal of the Brazilian Chemical Society*, 21:127–134.

McLachlan, G. J., Do, K.-A., and Ambroise, C. (2005). Analyzing microarray gene expression data.

Nanni, L. and Lumini, A. (2011). Prototype reduction techniques: A comparison among different approaches. *Expert Systems with Applications*, 38(9):11820–11828.

Nápoles, G., Falcon, R., Papageorgiou, E., Bello, R., and Vanhoof, K. (2017). Rough cognitive ensembles. *International Journal of Approximate Reasoning*, 85:79–96.

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J., and Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review*, 34(2):133–143.

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):11–18.

Rendon, E., Alejo, R., Castorena, C., Isidro-Ortega, F. J., and Granda-Gutierrez, E. E. (2020). Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, 10(4):1276.

Rico-Juan, J. R. and Iñesta, J. M. (2012). New rank methods for reducing the size of the training set using the nearest neighbor rule. *Pattern Recognition Letters*, 33(5):654–660.

Samuel, A. L. (1959). Machine learning. *The Technology Review*, 62(1):42–45.

Schaffer, C. (1994). A conservation law for generalization performance. In *Machine Learning Proceedings 1994*, pages 259–265. Elsevier.

Settouti, N., El Amine Bechar, M., and Amine Chikh, M. (2016). Statistical comparisons of the top 10 algorithms in data mining for classification task.

Singh, A., Thakur, N., and Sharma, A. (2016). A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1310–1315. Ieee.

Tallón-Ballesteros, A. J. and Riquelme, J. C. (2014). Data mining methods applied to a digital forensics task for supervised machine learning. In *Computational Intelligence in Digital Forensics: Forensic Investigation and Applications*, pages 413–428. Springer.

Thompson, S. K. (2012). *Sampling*, volume 755. John Wiley & Sons.

Tomek, I. (1976). An experiment with the edited nearest-nieghbor rule.

Triguero, I., Sáez, J. A., Luengo, J., García, S., and Herrera, F. (2014). On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, 132:30–41.

Valero-Mas, J. J., Calvo-Zaragoza, J., Rico-Juan, J. R., and Iñesta, J. M. (2017). An experimental study on rank methods for prototype selection. *Soft Computing*, 21(19):5703–5715.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421.

Wilson, D. R. and Martinez, T. R. (1997). Instance pruning techniques. In *ICML*, volume 97, pages 400–411.

Wilson, D. R. and Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine learning*, 38(3):257–286.

Witten, I. H. and Frank, E. (2002). Data mining: practical machine learning tools and techniques with java implementations. *Acm Sigmod Record*, 31(1):76–77.

Yıldırım, A. A., Özdoğan, C., and Watson, D. (2016). Parallel data reduction techniques for big datasets. In *Big Data: Concepts, Methodologies, Tools, and Applications*, pages 734–756. IGI Global.

Zhang, S., Zhang, C., and Yang, Q. (2003). Data preparation for data mining. *Applied artificial intelligence*, 17(5-6):375–381.