Data preprocessing techniques in supervised machine learning: handling of noise and selection of characteristics.

by

Gerardo Felix Benjamín

A thesis submitted in conformity with the requirements
for the MSc in Economics, Finance and Computer Science

University of Huelva & International University of Andalusia

September 2020

Técnicas de preprocesamiento de datos en aprendizaje automático supervisado: tratamiento de ruido y selección de características.

Gerardo Felix Benjamín gerardfelix87@gmail.com

Antonio Javier Tallón Ballesteros antonio.tallon@diesia.uhu.es

Máster en Economía, Finanzas y Computación

Universidad de Huelva y Universidad Internacional de Andalucía

2020

Abstract

Machine Learning (ML) techniques have been used for many years to infer knowledge from data. Several supervised and unsupervised learning algorithms have been proposed to learn from data, especially for supervised learning where the prior data pre-processing plays an essential role. This work focuses mainly on a comparative analysis of 37 well-known regression algorithms, which were compared on 141 traditional regression datasets with large data volumes where the experimental evidence showed that *Matern 5/2* (M5) is significantly the best algorithm, and 10 very popular classifiers in a great deal of datasets. Besides, a study on a good number of problems including different levels of noise is conducted as well as an analysis of feature selection as a data preparation stage.

Keywords: Machine learning, Supervised learning, Nonparametric Test, Feature selection, Noise.

Resumen

Las técnicas de aprendizaje automático (ML) se han utilizado durante muchos años para inferir el conocimiento de los datos. Se han propuesto varios algoritmos de aprendizaje supervisados y no supervisados para aprender de los datos, especialmente para el aprendizaje supervisado en el que el procesamiento previo de datos desempeña un papel esencial. Este trabajo se centra principalmente en un análisis comparativo de 37 algoritmos de regresión conocidos, que se compararon en 141 conjuntos de datos de regresión tradicionales con grandes volúmenes de datos donde la evidencia experimental mostró que el *Matern 5/2* (M5) es significativamente el mejor algoritmo, y 10 clasificadores muy populares en una gran cantidad de conjuntos de datos. Además, se realiza un estudio sobre un buen número de problemas que incluyen diferentes niveles de ruido, así como un análisis de la selección de características como una etapa de preparación de datos.

Palabras clave: Aprendizaje automático, Aprendizaje supervisado, Test no paramétrico, Selección de características, Ruido.

Agradecimientos

A la Universidad Internacional de Andalucía por la oportunidad de cursar este máster.

A mi tutor, Antonio Tallón por el apoyo durante la realización de este trabajo.

A todas aquellas personas que han contribuido a llegar hasta el final de máster.

A mis padres por ser mi inspiración.

Tabla de Contenidos

Lista de Tablas

Lista de Figuras

1 Introducción

Machine Learning (ML), que se traduce como Aprendizaje Automático es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente a partir de datos, para identificar patrones y tomar decisiones con la mínima intervención humana[1][2]. Infiere conocimientos a partir de datos con el uso de técnicas que provienen de diferentes campos dentro de la Informática y las Matemáticas mediante diferentes enfoques o estilos, como tablas de decisión [3], árboles de decisión[4], modelos basados en reglas[5], enfoques conexionistas (redes neuronales artificiales [6]), máquinas de soporte vectorial -*support vector machines* (SVM)[7]- entre otros, cada uno con sus propias ventajas y limitaciones. Estas familias de algoritmos dan solución a problemas de regresión lineal y clasificación, según el tipo de situación que se presente, agrupadas en tres categorías principales, siendo las dos primeras las más tradicionales en la literatura:

Aprendizaje supervisado: permite realizar predicciones futuras basadas en comportamientos o características de datos históricos etiquetados.

Aprendizaje no supervisado: permite realizar predicciones futuras basadas en comportamientos o características de datos históricos no etiquetados.

Aprendizaje por refuerzo: su objetivo es que un algoritmo aprenda a partir de la propia experiencia. Esto es, que sea capaz de tomar la mejor decisión ante diferentes situaciones de acuerdo a un proceso de prueba y error en el que se recompensan las decisiones correctas.

Menos común pero no menos importante es la hibridación entre el aprendizaje supervisado y no supervisado, *el aprendizaje semi-supervisado*, el cual se utiliza para las mismas aplicaciones que el aprendizaje supervisado, pero con datos etiquetados y no etiquetados para crear un modelo computacional.

A su vez, el ML también es denominado, a veces, reconocimiento de patrones [8], uno de los problemas más omnipresentes del mundo real y, sin duda, muy utilizado en Inteligencia Artificial [9]. Consiste en identificar la categoría correcta (entre las de un conjunto predefinido) a la que pertenece un patrón observado mediante el uso de algoritmos informáticos, es decir, tiene como objetivo asignar objetos a clases predefinidas de un grupo de objetos previamente

almacenados que ya están clasificados [10]. Estos patrones a menudo son descritos mediante un conjunto de atributos predictivos de naturaleza numérica y/o nominal. Formalmente hablando, un problema de reconocimiento de patrones [8] consiste en construir un mapeo $f: U \rightarrow D$ que asigna a cada instancia $x \in U$ descrito por el conjunto de atributos $\vartheta = \{\vartheta_1, \ldots, \vartheta_M\}$ una clase de decisión *D* de las *N* posibles en $D = \{D_1, \ldots, D_N\}$.

No existe un único modelo para todos los problemas de reconocimiento de patrones y un algoritmo de aprendizaje no es aplicable a todos los problemas[11], por lo que otro aspecto a destacar en la literatura sobre reconocimiento de patrones, es la existencia de trabajos donde se evalúa el desempeño de diversos algoritmos sobre múltiples conjuntos de datos [11][12][13], demostrándose la variedad existente de estos y lo difícil que resulta seleccionar el más adecuado para un problema dado. Hasta donde sabemos, no existen estudios similares como los antes mencionados que aborden un análisis similar para problemas de regresión lineal mediante un gran número de base de datos, lo que permitiría seleccionar un top 10 de algoritmos de regresión lineal, de utilidad para futuras investigaciones.

De ahí a que surja el interés por llevar a cabo un estudio sobre esta temática a pesar de que los mismos se han aplicado a muchos problemas en econometría [15], motivado por el crecimiento exponencial de la cantidad de datos numéricos que se generan actualmente en las empresas y la necesidad de desarrollar mejores acciones de negocio, lo que será un punto de partida para futuras investigaciones. Un modelo de regresión lineal es fácilmente comprendido ya que muchos procesos siguen de forma natural una relación que puede aproximarse mediante una relación lineal. Analizar algoritmos de regresión se convierte en uno de los objetivos de este trabajo, pero sin preprocesamiento de los datos ya que mejorar los resultados de la regresión lineal implica que esta debe realizarse específicamente para cada conjunto de datos aumentando en gran medida la carga de trabajo. No obstante, la necesidad de estas transformaciones adicionales es explotada, aunque en menor medida (con un menor número de bases de datos) en este mismo trabajo.

Por otra parte, no hay certeza de que un algoritmo funcione mejor para un conjunto de datos dado, que otro que parece mucho más competitivo, según el teorema *No-Free-Lunch*, el cual plantea que el mejor algoritmo no será el mismo para todos los conjuntos de datos[16][17]. Por ende, en aprendizaje supervisado, la elección del algoritmo en esta larga lista de métodos es

primordial para una mejor predicción y crear modelos de computacionales que den como resultado: predicciones de alto valor para tomar mejores decisiones. Sobre este tema, varios autores como [16][17] presentan una guía para analizar y diseñar experimentos exploratorios e inferenciales entre conjuntos de datos y algoritmos de aprendizaje. De la misma manera que existen algoritmos de aprendizaje existen técnicas estadísticas para evaluar el desempeño de estos, resumidos por Demšar[18] y extendidos por [19][20][21]. Hablamos de pruebas no paramétricas, donde la calidad de los resultados no depende, a priori, de la distribución de los datos subyacentes [11].

La limpieza de datos, la detección y eliminación de valores atípicos son ejemplos de técnicas que deben abordarse dentro de un proceso de minería de datos [22][23]. Otro aspecto de especial interés y que está estrechamente vinculado con el ML es la selección de características (atributos) para el preprocesado de los conjuntos de datos antes de ser utilizados. La existencia de características irrelevantes, redundantes y el gran número de estas presentes en muchos problemas, puede producir problemas de sobreaprendizaje, hacer más confusos los modelos resultantes y causar un impacto negativo en las predicciones en ciertos algoritmos de aprendizaje [19][20]. Indistintamente, se tiene valores atípicos (*outliers*), eliminarlos del todo o sustituirlos no es la opción más adecuada porque se modifican las inferencias que se realicen a partir de esta información, introduciendo un sesgo. La mejor opción es detectarlos, quitarles peso a esas observaciones atípicas o tratar con la variabilidad producida por estos datos en un problema de ML. Los *outliers* no son objeto de estudio en el presente Trabajo Fin de Master sino el ruido en los valores de los atributos, que ha sido agregado artificialmente.

La selección de características es el proceso de identificar y eliminar tantas características irrelevantes y redundantes como sea posible, o sea, reducir la dimensionalidad del problema con el fin de mejorar la capacidad de generalización. Además, acelera el proceso de aprendizaje (aunque se añada este coste al proceso). Mejora la interpretabilidad de un modelo reduciendo la complejidad del mismo permitiendo que los algoritmos de ML funcionen más rápido y eficientemente. Por consiguiente, resulta de gran importancia y por tal motivo, será tratado en este trabajo. El hecho de que muchas características dependan unas de otras, influencia indudablemente la precisión de los algoritmos de aprendizaje supervisados en ML. En relación al ruido, es importante destacar que ha sido agregado artificialmente a un problema, practica muy

común para evaluar la robustez de un algoritmo en ML y no al ruido incluido en el conjunto de datos original.

En la literatura de minería de datos se han presentado varias definiciones sobre ruido, pero nos quedamos con aquel valor atípico (ruido) que es muy diferente del resto de los datos basados en alguna medida de distribución [26]–[31], es decir, aquel que no sigue el mismo modelo que el resto de los datos y parece que procede de una distribución de probabilidad diferente[32]. La detección de ruido es un procedimiento que selecciona k muestras que son considerablemente diferentes o excepcionales con respecto a los datos restantes [33][23]. Este trabajo no se centra en el manejo del ruido de manera independiente para lo cual existen diferentes técnicas estadísticas sino en el tratamiento y detección de los mismos mediante la selección de características y cómo influye esto en el proceso de aprendizaje automático de modelos computacionales.

El objetivo de este trabajo de fin de máster es analizar empíricamente el efecto del ruido y la conveniencia de aplicar o no selección de atributos en tareas de aprendizaje automático.

2 Aprendizaje automático supervisado

El aprendizaje supervisado en ML estudia la construcción de sistemas capaces de aprender modelos computacionales para predecir el valor de ciertas variables dependientes a partir de cierto número de variables independientes [1][2][20][21][36]. Las variables independientes X pueden ser continuas, categóricas o binarias mientras que las variables dependientes Y son continuas cuando se trata de regresión lineal o categóricas para problemas de clasificación.

Numerosas aplicaciones de ML involucran a estas tareas han sido utilizadas durante más de 50 años, así como una gran variedad de métodos y algoritmos han sido propuestos para entrenar modelos para aplicaciones de reconocimiento de patrones. La tabla 1 detalla cómo de diversificada se encuentra la literatura sobre ML demostrándose cómo de disperso está el ML en la actualidad y la importancia del mismo.

Por solo mencionar algunos ejemplos, ML incluye trabajos de detección de fraude en transacciones, predicción de fallos en equipos tecnológicos, selección de clientes potenciales basándose en comportamientos en las redes sociales e interacciones en la web, predicción del

tráfico urbano, conocer cuál es el mejor momento para publicar tuits y actualizaciones de Facebook. Hacer prediagnósticos médicos basados en síntomas del paciente, cambiar el comportamiento de una aplicación (*app*) móvil para adaptarse a las costumbres y necesidades de cada usuario, detectar intrusiones en una red de comunicaciones de datos, decidir cuál es la mejor hora para llamar a un cliente, predicciones económicas y fluctuaciones en el mercado bursátil, mapeos y modelados 3D, sistemas de reconocimiento de voz, optimización e implementación de campañas digitales publicitarias, entre otras.

Tabla 1. Publicaciones relacionadas con ML.[a]

Disciplinas	Cantidad	Subdisciplinas	Cantidad	Lenguajes	Cantidad	Tipos de contenido	Cantidad
Ciencia de la computación	209912	**Inteligencia artificial**	158530	**Inglés**	498087	**Capítulos de libro**	339526
Ingeniería	84139	**Sistemas de información (incluido internet)**	56455	**Alemán**	4542	**Artículos de conferencias** **Artículos**	168037 151835
Administración y negocios	24095	**Redes de comunicaciones informáticas**	47509	**Francés**	211	**Entrada en trabajos de referencia**	9265
Medicina y salud pública	24078	**Procesamiento de imágenes y visión por computador**	45420	**Dutch**	100	**Protocolos** **Book**	1630 669
Educación	15268	**Minería de datos y Descubrimiento del conocimiento**	45416	**Italiano**	44	**Proceedings en conferencias**	308

Nota: [a] https://link.springer.com/search?query=machine+learning.

Tomado 18/09/2020

2.1 Clasificación

Los problemas de clasificación como forma de aprendizaje supervisado tienen como objetivo predecir valores categóricos (nominales) Y a partir de datos históricos etiquetados X de tipo binario, continuo o categóricos e indistintamente. Para evaluar el desempeño de los algoritmos existen varias medidas de evaluación, entre las más comunes se encuentran:

-*Curvas ROC* [37]: es una de las métricas más utilizadas. Muestra el desempeño de un modelo de clasificación a través de los parámetros de la matriz de confusión (describe el desempeño completo de un modelo de clasificación) mediante:

- Tasa de verdaderos positivos (TVP): mide la *sensibilidad* del modelo donde VP (verdaderos positivos) y FN (falsos negativos) de tal forma que:

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

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

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

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

donde FP y VP tienen valores en el rango $[0,1]$.

Una curva ROC representa TVP frente a TPR en diferentes umbrales de clasificación. Una forma de interpretar el ROC es como la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio. Oscila entre $[0, 1]$, donde un modelo cuyas predicciones son un 100% incorrectas tiene un ROC de 0.0 mientras que otro cuyas predicciones sean de un 100% correctas tiene un ROC de 1.0. El área bajo la curva ROC (*Area Under Curve*, AUC) es conveniente por dos razones, es *invariable con respecto a la escala*, mide qué tan bien se clasifican las predicciones, en lugar de sus valores absolutos y es *invariable con respecto al umbral de clasificación*. Mide la calidad de las predicciones del modelo, sin tener en cuenta qué

umbral de clasificación se elige. Sin embargo, la invariabilidad de escala no siempre es conveniente, así como tampoco la invariabilidad del umbral de clasificación para muchos problemas.

-*Precisión* (Accuracy) [38]: es una forma de medir la frecuencia con la que el algoritmo clasifica instancias procedentes de un conjunto de datos correctamente, o sea, es la relación entre el número de predicciones correctas y el número total de muestras de entrada.

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

-*Coeficiente de Kappa*: El coeficiente kappa de Cohen [39]–[41] se basa en comparar la concordancia observada en conjunto de datos, respecto a la que podría ocurrir al azar, mide la fiabilidad. Toma valores $[-1\ ,1]$ y mientras más cercano a 1 mayor es el grado de concordancia. Un valor de 0 si la concordancia observada coincide con la que ocurriría por puro azar y exactamente 1 concordancia perfecta. Por lo general, se considera una medida más sólida que la precisión estándar, ya que este coeficiente tiene en cuenta la coincidencia que se produce por casualidad.

2.2 Regresión

Existen diferentes tipos de regresión, en los que cabe mencionar la regresión lineal y la no paramétrica, cuya diferencia radica en que de antemano conocemos o no la forma que tendrá la función, que buscamos como aproximación al modelado de los datos del conjunto de entrenamiento.

La regresión lineal como forma de aprendizaje supervisado tiene como objetivo predecir valores continuos a partir de datos históricos etiquetados**.** Para aprender este tipo de modelos es necesario establecer la relación entre un cierto número de características X continuas y una variable objetivo Y continua, donde el comportamiento de una variable dependiente Y, se puede explicar a través de al menos una variable independiente X, lo que representamos mediante una recta $Y = f(X)$[42][43], o sea, una o varias variable(s) dependientes pueden ser escritas en términos de una combinación lineal de las variables independientes[42].

Según sea el número de variables independientes x_i estamos en presencia de una regresión lineal simple o múltiple como se detalla a continuación, donde a su vez se precisan las distintas métricas de evaluación del error existentes para evaluar el desempeño de estos modelos [15] [42][43] [44].

Regresión lineal simple

Para la estimación de una relación lineal entre dos variables, el algoritmo de regresión establecerá un modelo para ajustar la relación de dependencia entre una característica específica independiente (un único valor explicativo) *x* y el valor "resultado" correspondiente (un valor de la variable dependiente *y*), es decir, ajustar los puntos (x_i, y_i) a una recta de la forma:

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

donde β_0 = es el coeficiente de intersección, (corte con el eje y), β_1 = pendiente, coeficiente de regresión de la variable independiente, x_i = la variable independiente y e_i = es el error observado (residual) o la desviación de y_i de la línea $\beta_0 + \beta_1 x_i$ para la i-ésima observación en la muestra. En general β_0 y β_1 son los coeficientes de regresión y denotan la pendiente de la recta y el corte con el eje Y. La regresión lineal simple se ilustra en la Figura 1.

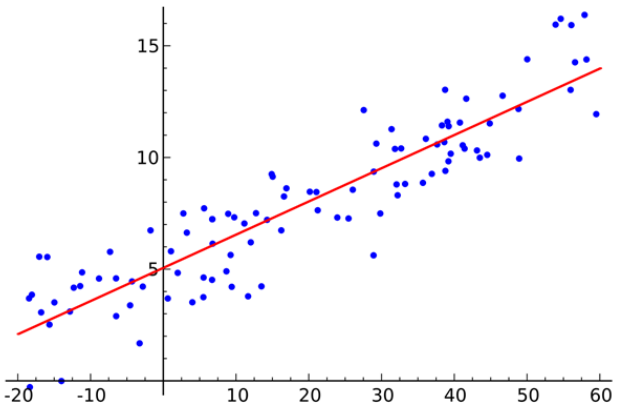

Figura 1. Regresión lineal simple**.**

Regresión lineal múltiple

Hasta ahora, solo hemos considerado un regresor X además de la constante en la ecuación de regresión lineal pero los problemas en economía suelen incluir más de un regresor [15]. La regresión lineal múltiple introduce más de una variable independiente para predecir el comportamiento de la variable dependiente Y. Se fija la variable que se quiere predecir Y y se determina la relación con el resto de variables predictoras (independientes), por lo que ahora tenemos un hiperplano de la siguiente manera:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + e_i$$

donde x_n = es la enésima variable independiente, y_i = la variable dependiente, e_i = es el error observado de y_i de la línea $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$ para la i-ésima observación en la muestra, β_0 = el coeficiente de intersección (corte con el eje y), β_n = coeficiente de regresión de la enésima variable independiente X. A modo de ejemplo, la Figura. 2 muestra una regresión lineal múltiple con dos características, x_1 y x_2.

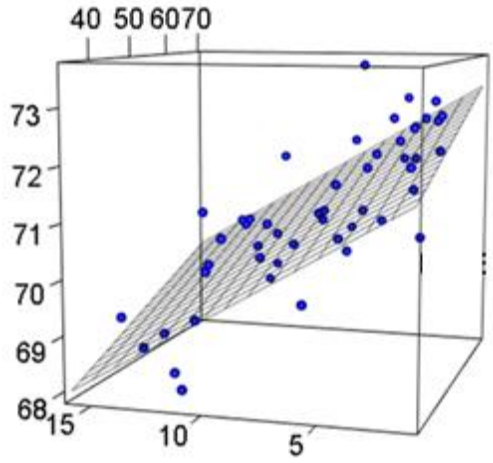

Figura 2. Regresión lineal múltiple.

Mínimos cuadrados ordinarios

El primer paso para abordar un problema de regresión es estimar el modelo. El criterio matemático estándar utilizado es el de los mínimos cuadrados ordinarios, cuyo objetivo es la obtención de un hiperplano de forma tal que se minimice la suma de los errores cuadrados observados (SSE): $e_1^2 + e_2^2 + \cdots + e_n^2$ en una muestra de tamaño *n* [15][42][43] para cada una de las observaciones de las variables independientes x_i y dicho hiperplano (residuos) y_i.

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Dicho de otro modo, $e_i = y_i$-$\hat{y}_i$ (es la diferencia entre el valor observado y_i y el valor predicho (o ajuste predicho) de y_i para la i-ésima observación en la muestra $\hat{y}_i$. En regresión lineal simple $\hat{y}_i = \beta_0 + \beta_1 x_i$ y $\hat{y}_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$ para regresión lineal múltiple. Por tanto, la mejor recta de regresión lineal para una muestra que utiliza el método de mínimos cuadrados ordinarios garantiza que $\sum_{i=1}^{n} e_i = 0$ [42].

La Figura 3 muestra este procedimiento a modo de ejemplo en el cual se establece una línea arbitraria y luego se calcula la distancia de la recta a los puntos de datos correspondientes a los valores (x_i, y_i). Esta distancia, las líneas verticales, son los "residuos" o los "errores de predicción". El algoritmo de regresión lineal recalculará (y moverá) la recta con cada interacción, buscando aquella que mejor se ajuste a los puntos de datos (x_i, y_i), en otras palabras, la línea con el menor error (la más cercana al máximo número de puntos).

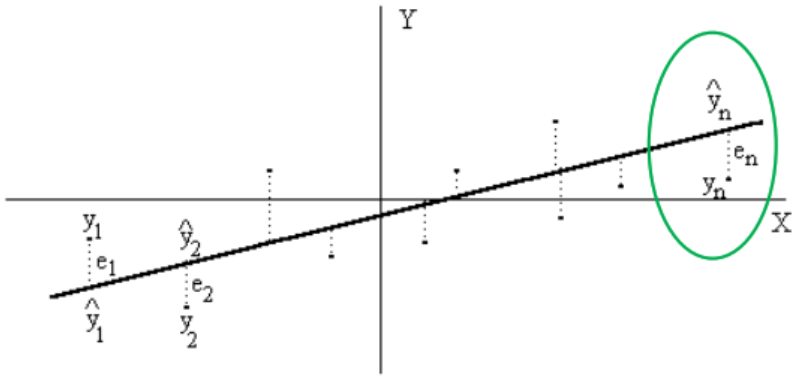

Figura 3. Ajuste de la recta por mínimos cuadrados ordinarios.

Este método de estimación es el más habitual cuando se realiza el ajuste de un modelo de regresión lineal, por lo tanto, el objetivo de un modelo de regresión lineal es determinar una función matemática sencilla que explique el comportamiento que existe entre una variable dependiente *Y* dados los valores de una u otras variables independientes (variables explicativas) $X = x_i = x_1, \ldots, x_n$. A continuación, se describen otras métricas de evaluación del error, entre las más utilizadas están:

Error Absoluto Medio (MAE**):** es la media de la suma de los valores absolutos de la diferencia entre los valores reales y_i y los valores predichos $\hat{y}_i$. Nos da la medida de qué tan lejos están las predicciones del resultado real. Sin embargo, no nos dan ninguna idea de la dirección del error, es decir, si estamos prediciendo los datos por debajo de lo esperado o por encima de ellos.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Error cuadrático medio (MSE): es la media de la diferencia entre el valor real y_i y su estimación $\hat{y}_i$ al cuadrado. Este método penaliza más las diferencias mayores (cuanto mayor sea este valor, peor es el modelo). Nunca es negativo, pero es cero para un modelo perfecto. Es la medida estándar para los problemas de regresión lineal y viene dado por:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Raíz del error cuadrático medio (RMSE): es la raíz cuadrada del MSE. Este valor se puede interpretar (aproximadamente) como el tamaño absoluto promedio de las desviaciones de los individuos de la línea de regresión muestral [42]. Éste es el medidor de evaluación más popular para determinar el desempeño de los modelos de regresión lineal ya que RMSE amplifica y penaliza con mayor fuerza aquellos errores de mayor magnitud.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

El coeficiente de determinación (R^2**):** se entiende como una versión estandarizada del MSE, que proporciona una mejor interpretación del rendimiento del modelo. Si definimos SSE como $SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ y TSOS como $\sum_{i=1}^{n}(y_i - \tilde{y})^2$ (la suma total de los cuadrados), donde $\tilde{y}$ es la media de todos los valores de y observados; técnicamente, el R^2 representa la varianza de la respuesta capturada por el modelo:

$$R^2 = \frac{TSOS - SSE}{TSOS}$$

El mejor resultado posible es 1.0, y ocurre cuando la predicción coincide con los valores de la variable objetivo. R^2 puede tomar valores negativos pues la predicción puede ser arbitrariamente mala, pero cuando la predicción coincide con la esperanza de los valores de la variable objetivo, el resultado de R^2 es 0. De acuerdo a [42] R^2 es una medida de evaluación muy popular pero no debe utilizarse únicamente para evaluar la idoneidad de un modelo de regresión lineal sin una justificación adicional, por ejemplo:

1. El valor de R^2 es muy sensible al tamaño de la muestra y aumenta al agregar más predictores al modelo. También puede provocar un aumento del MSE para tamaños de muestra pequeños.
2. R^2 está influenciado por el rango de los predictores, en el sentido de que, si el rango de X aumenta o disminuye, R^2 aumenta o disminuye, respectivamente.
3. La magnitud de las pendientes no se mide por R^2.
4. Solo mide la fuerza del componente lineal de un modelo.
5. Un nivel alto o bajo de R^2 no necesariamente indica la previsibilidad del modelo.

2.3 Selección de características

El objetivo de la selección de características (*feature selection*) es encontrar el subconjunto de atributos más apropiados para un problema de ML mientras que la generación consiste en generar (y seleccionar entre ellos) nuevos atributos a partir de los ya existentes. De manera general, es encontrar el subconjunto mínimo de atributos que hace óptima la predicción. Para definir un método de selección de características es necesario definir un espacio de búsqueda y un método de evaluación de la calidad de los subconjuntos como se describe a continuación[24] [39][40][45]–[47].

GENERACIÓN DE ATRIBUTOS: Construcción de nuevos atributos.

Análisis de componentes principales (en inglés, *Principal Component Analyis*, PCA) [50][51][52]. Este método construye nuevos atributos como combinación lineal de los anteriores identificando un primer componente que explique la mayor cantidad posible de varianza, un segundo componente que explique la siguiente mayor cantidad de varianza y así sucesivamente.

PCA es una técnica estadística multivariante que consiste en intentar determinar si un conjunto de datos se puede expresar mediante una dimensionalidad inferior al número real de atributos de

un problema, dado que este método ordena esos nuevos atributos por importancia (varianza explicada), también se puede utilizar como método de selección. Como es una transformación lineal de los atributos originales, o sea, una transformación no supervisada, no hay garantía de que se generen características que discriminen mejor las clases. Si el número de características es muy grande, el método suele ser lento pero elimina redundancia (correlación) de los datos por lo que ha sido aplicado en campos como el reconocimiento facial, compresión de imágenes y como una técnica común para encontrar patrones en datos de gran dimensión [50].

Proyecciones aleatorias (Random Projections)[22][23]: este método proyecta los datos a dimensiones inferiores mediante matrices aleatorias. Similares resultados a PCA y más rápido, siempre que el número de dimensiones de la proyección no sea demasiado pequeño. Para un número grande de atributos obtiene resultados similares a PCA, con un menor esfuerzo computacional[53] a pesar de tiene un rendimiento predictivo inferior al PCA[54].

SELECCIÓN DE CARACTERÍSTICAS : Para definir un método de selección de características es necesario definir un espacio de búsqueda y un método de evaluación de la calidad de los subconjuntos, de ahí a que se clasifiquen en dos tipos principales: ranking y selección de subconjuntos[25], [55]–[58][59].

i. ***Ranking*** (evaluación y ordenación de las características de manera individual eliminándose las menos valoradas)[60][61][62]: Dado unas características $\gamma_1, \gamma_2, \ldots, \gamma_n$, se evalúa cada γ_i de manera independiente calculando medidas de correlación de la característica con la clase. Una característica γ_1 está correlacionada con la clase, si conocer su valor implica que podemos predecir la clase con cierta probabilidad. Existen varios criterios para evaluar estas características como son *la entropía* (information gain), *chi-square*, *información mutua* (mutual information*)*, etc. Una vez evaluadas y ordenadas, se eliminan las k peores. La principal ventaja es la rapidez y sus principales inconvenientes son que no elimina características redundantes y no detecta características que funcionen bien de manera conjunta.

ii. ***Selección de subconjuntos*** (subset selection)[63][64] no es más que la búsqueda del subconjunto de características más relevante): Estos métodos recorren un espacio de búsqueda de subconjuntos de características, evaluando subconjuntos completos de características. No se recorre el espacio entero (búsqueda exhaustiva, la cual sería más

preciso, pero poco práctico), sino sólo aquellos subconjuntos más prometedores, además, evalúan el subconjunto de manera conjunta. Como representación tenemos a:

- *Correlation-based feature selection* (CFS) estos métodos evalúan un subconjunto de características calculando la media de las correlaciones de cada atributo con la clase, así como las correlaciones por redundancias entre atributos:

$$\text{Evaluación } (A_i) = \frac{\sum_j \cup (A_j, C)}{\sqrt{\sum_i \sum_j \cup (A_i, A_j)}} = \frac{correlación\ con\ la\ clase}{correlación\ entre\ atributos}$$

Tiene como ventaja el hecho de ser un método rápido y que elimina características redundantes, pero como *Ranker*, puede eliminar características que de por sí solas no están correlacionados con la clase, pero con otra característica sí.

- *Wrapper*[65][66]: evalúan un subconjunto de características ejecutando un algoritmo de minería de datos concreto sobre un conjunto de entrenamiento, donde el valor del subconjunto es el percentage de aciertos obtenido con esas características. Tienen como ventajas la obtención de subconjuntos de características adecuadas para un algoritmo de minería de datos concreto y que evalúan a las características de los subconjuntos de manera realmente conjunta. Entre los principales inconvenientes es que son muy lentos y pueden llevar a un sobreaprendizaje.

iii. ***Relief*** [67][68]–[70]: es un algoritmo de *Ranking* pero tiene las ventajas de *Wrapper*. Capaz de detectar interacciones entre características, las relevantes e incluso aquellas que funcionen bien en grupos, es rápido, pero no detecta características redundantes y es necesario normalizar las características.

Para la selección de características existen en la literatura diferentes métodos de búsqueda tales como: *Exhaustive_Search*: búsqueda exhaustiva (muy lento), *Greedy_stepwise:* escalada (muy rápido) en sus 2 variantes (selección hacia adelante y selección hacia detrás), *best first* (lento), *Genetic_Search*: búsqueda genética (rápido), *Rank_Search*: ordena los atributos y después construye el subconjunto de manera incremental, en dirección del mejor al peor, hasta que no merece la pena añadir nuevos atributos (rápido), entre otros.

3 Caracterización de los conjuntos de datos y algoritmos

En esta sección se describen los materiales (conjuntos de datos) y métodos (algoritmos) utilizados para realizar las simulaciones asistidos por las herramientas $WEKA_{v3.8.4}$ [71][72] (software libre) y $Matlab_{vR2020a}$ [19][20]. Nos apoyamos en los repositorios de base de datos de la Universidad de California en Irvine, UCI ML (repositorio habitual en comparaciones experimentales [75]), Keel [76] y en varios conjuntos de datos utilizados en los artículos [16][13].

3.1 Conjuntos de datos

La tabla 2 describe los conjuntos de datos seleccionados para el análisis de los diferentes algoritmos de ML de regresión: el número de patrones (#pat.), cantidad de características (#inp.), Estos conjuntos de datos contienen grandes volúmenes de datos (de 321 a 13750), con características que varían de 2 a 60.

Tabla 2. Caracterización de los 141 conjuntos de datos adoptados para el análisis de los modelos de regresión.

Datos	#pat.	#inp.	Datos	#pat.	#inp.	Datos	#pat.	#inp.
abalone-3	4177	8	ilpd-indian-liver	583	9	tic-tac-toe	958	9
abalone-7	3295	8	kin8nm	8192	8	titanic	2201	3
abalone-11	3842	8	laser	993	4	treasury	1049	15
abalone-28	4177	8	led-display	1000	7	twonorm	7400	20
ailerons	13750	40	led7digit	500	7	vehicle-silh	846	18
airSelfNoise	1503	6	mammographic	961	5	vehicle0	846	18
airQuality	9357	12	meta	528	21	vehicle1	846	18
artificial-ch	10218	7	mfeat-morph	2000	6	vehicle2	846	18
anacalt	4052	7	mortgage	1049	15	vehicle3	846	18
auto-au1-1000	1000	20	monks1	556	6	visualizing-soil	8641	4
auto-au7-300	700	12	monks2	601	6	volcanoes-a1	3252	3
auto-au6-1000	1000	40	monks3	554	6	volcanoes-a2	1623	3
balance-scale	625	4	nursery	12960	8	volcanoes-a3	1521	3

banana	5300	2	oocytes2f	912	25	volcanoes-a4	1515	3
bank-marketing	4521	16	page-blocks	5473	10	volcanoes-b1	10176	3
bank-auth	1372	4	parkinson_msr	1040	25	volcanoes-b2	10668	3
bank32nh	8192	32	parkinsons_updrs	5875	21	volcanoes-b3	10386	3
bank8FM	8192	8	pendigits	10992	16	volcanoes-b4	10190	3
bias_correction	7752	23	phoneme	5404	5	volcanoes-b5	9989	3
blood_transfusion	748	4	pima-diabetes	768	8	volcanoes-b6	10130	3
breast-c-wisc	699	9	pima-10an-nn	768	8	volcanoes-d1	8753	3
breast-c-wisc-diag	569	31	pima-20an-nn	768	8	volcanoes-d2	9172	3
car-evaluation	1728	6	pima-5an-nn	768	8	volcanoes-d3	9285	3
cardiographt-3	2126	35	plastic	1650	2	volcanoes-d4	8654	3
cardiography-10	2126	35	puma32h	8192	32	volcanoes-e1	1183	3
ccpp	9568	4	puma8NH	8192	8	volcanoes-e2	1080	3
climate-simulation	540	20	qsar-bio	1055	41	volcanoes-e3	1277	3
connect-vowel	990	13	quake	2178	3	volcanoes-e4	1252	3
flare	1389	12	ring	7400	20	volcanoes-e5	1112	3
compactiv	8192	21	ringnorm	7400	20	wall-following	5456	24
concrete	1030	8	robot-sensor-2	5456	2	wankara	321	9
conditionBased	11934	17	robot-sensor4	5456	4	waveformv1	5000	21
congress-voting	435	16	robot- sensor24	5456	24	waveformv2	5000	40
conn-bench-vowel	528	11	oocytes_ 4d	1022	41	wdbc	6574	14
contraceptive	1473	9	satimage	6435	36	wilt	4839	5
cpu.act	8192	21	segment	2310	20	wind	569	30
cpu.small	8192	12	sensory	576	11	winequality5	6462	11
credit-approval	690	15	seismic-bumps	2584	18	winequality_r	1599	11
cylinder-bands	512	35	skillCraft1	3395	18	winequality_w	4898	11
delta_ailerons	7129	5	sml2010	4137	21	winequality_w5	4873	11
delta_elevators	9517	6	australian-credit	690	14	wizmir	1461	9
ele-2	1056	4	statlog-image	2310	18	yeast	1484	8

electricalGrid	10000	13	statlog-landsat	4435	36	yeast-4c	1299	8
energy-y1	768	8	steel-plates	1941	27	yeast1	1484	8
energy-y2	768	8	stock	950	9	yeast3	1484	8
friedman	1200	5	synthetic-control	600	60	german	999	23
			thyroid-hypo	3163	25	housing	506	13

Un gran número de datos porque algoritmos con muy buen desempeño promedio sobre un grupo de datos pequeño logran resultados significativamente peores cuando se amplía esta cantidad, o sea, algoritmos con desempeño sub-óptimos mejoran cuando se introduce una mayor cantidad de datos[16].

Las tablas 3 y 4 muestran los conjuntos de datos utilizados para realizar selección de características para luego evaluar las diferencias (con y sin selección de características). La tabla 3 posee los conjuntos de datos sobre regresión lineal con instancias que varían desde las 270 hasta 1484 y con características de 7 hasta 60. La tabla 4 asociada a problemas de clasificación contiene conjuntos de datos que varían de 214 a 1473 instancias, características de 7 hasta 60 y número de clases de 2 hasta 8. En ambas tablas los datos poseen ruido artificial de 5, 10, 15 y 20 por ciento según sea el caso, de acuerdo a [77].

Tabla 3. Conjuntos de datos de regresión para selección de características.

Datos	#pat.	#inp.	Datos	#pat.	#inp.
contraceptive20	1473	9	ionosphere20	208	60
ecoli15	336	7	pima20	208	60
ecoli20	336	7	sonar10	208	60
glass15	214	9	sonar15	569	30
glass20	214	9	sonar20	569	30
heart15	270	13	wdbc5	569	30
heart20	270	13	wdbc10	569	30
ionosphere15	351	33	wdbc15	178	13
contraceptive20	351	33	wdbc20	1484	8
ecoli15	768	8	wine20	1484	8

Tabla 4. Conjuntos de datos sobre clasificación para selección de características.

Datos	#pat.	#inp.	#Clases	Datos	#pat.	#inp.	#Clases
contraceptive5	1473	9	3	ionosphere10	351	34	2
contraceptive10	1473	9	3	pima5	768	8	2
contraceptive15	1473	9	3	pima10	768	8	2
ecoli5	336	7	8	pima15	768	8	2
ecoli10	336	7	8	sonar5	208	60	2
glass5	214	9	7	wdbc5	569	30	2
glass10	214	9	7	wdbc10	569	30	2
heart5	270	13	2	wine5	178	13	3
heart10	270	13	2	wine10	178	13	3
ionosphere5	351	34	2	wine15	178	13	3

De manera general, se asume que varios de los conjuntos de datos presentados en las tablas 2,3 y 4 contengan ruido de forma natural. Además, cada uno de los experimentos sigue un procedimiento de validación cruzada (*k-fold cross-validation*) para $k = 10$ [78] siguiendo la misma descripción que se indica en la Figura 4.

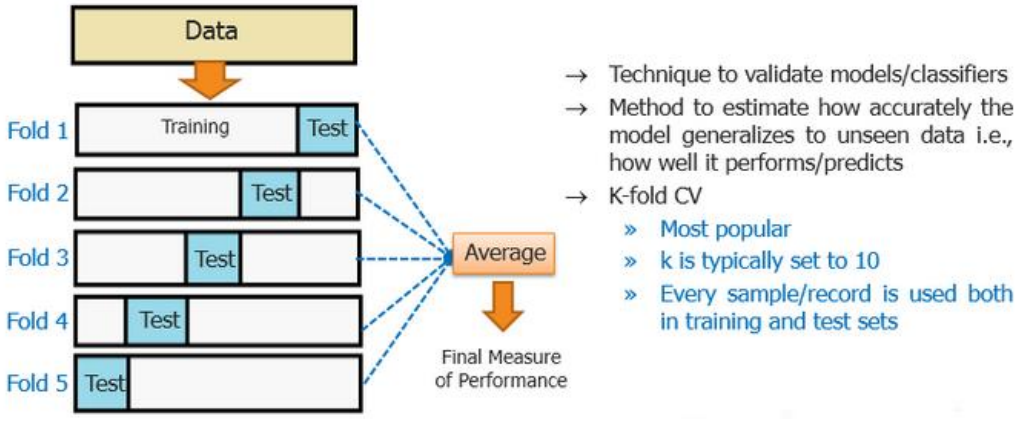

Figura 4. Validación cruzada: www.edureka.in/data-science

3.2 Algoritmos para clasificación y regresión.

En esta subsección describimos los algoritmos empleados con fines de evaluación comparativa implementados para WEKA$_{v3.8.4}$ y Matlab$_{vR2020a,}$ sin el ajuste de hiperparámetros para las tareas

de clasificación y regresión lineal. Excepto, *Multilayer perceptron* (MLP) para las pruebas que se realizan para selección de características, modificándose el hiperparámetro *semilla* (seed= 10) y *Linear Regression* (LR) para la selección de un top 10 de algoritmos de regresión lineal (utiliza selección de características en su implementación por defecto), ambos implementados para $WEKA_{v3.8.4}$. A pesar de que una buena elección de parámetros en un algoritmo de aprendizaje aumenta el desempeño de estos en diferentes conjuntos de datos, un robusto algoritmo de aprendizaje permite la obtención de resultados suficientemente buenos incluso cuando sus parámetros no están optimizados para un conjunto de datos específico[12][79].

Para la elección de un top 10 de algoritmos de clasificación se tiene como referencia el trabajo de [16], algoritmos que al igual que los obtenidos en el estudio en regresión lineal (obtención de un top 10 de algoritmos de regresión lineal) son los que se utilizarán en el proceso de selección de características para la experimentación en secciones posteriores. La tabla 5 describe estos algoritmos de aprendizaje supervisado.

Tabla 5. Descripción de los algoritmos de aprendizaje supervisado.

Algoritmo	Descripción	Herramienta	Tarea
***Multilayer perceptron* (MLP)**	Red neuronal que utiliza el algoritmo de backpropagation con neuronas ocultas sigmoides para entrenar el modelo [80].	**Weka**	Regresión Clasificación
***Linear Regression* (LR)**	Aprende un modelo de regresión lineal utilizando el criterio de *Akaike* para la selección del modelo.		Regresión
***Simple Linear Regression* (SLR)**	Aprende un modelo de regresión lineal simple. Elige el atributo que da como resultado el error al cuadrado más bajo.		Regresión
***Gaussian Processes* (GP)**	Implementa procesos gaussianos para regresión sin el ajuste de hiperparámetros. Utiliza un kernel lineal.		Regresión
***SMOreg* (SMO)**	Implementa un *support vector machine* [81][82].		Regresión
SMO (SMO_1)	Implementa un *support vector machine* con un algoritmo secuencial de optimización mínima [83].		Clasificación
Simple Logistic (SL)	Clasificador para construir modelos de regresión logística lineal [84][85].		Clasificación
***IBK*: *k-Nearest Neighbors* (*k*NN)**	Algoritmo de aprendizaje perezozo basado en una función de similitud [86][87].		Regresión
***KStar* (K*)**	Es un clasificador basado en instancias, similar a kNN, que utiliza una función de distancia basada en la entropía [88].		Regresión
LWL	Algoritmo perezozo basado en instancias con aprendizaje ponderado local[89].		Regresión Clasificación
***Decision Stump* (DS)**	Es un árbol de decisión de un solo nodo que desarrolla una regresión basada en una sola entrada usando entropía. Usualmente se utiliza en conjunción con *boosting* y se basa		Regresión Clasificación

	en el error cuadrático medio.		
M5P	Implementa el algoritmo del modelo de árbol M5[90][91].	**Weka**	Regresión
***Random Forest* (RF)**	*Bagging* de árboles aleatorios [32][33].		Regresión Clasificación
***Random Tree* (RT)**	Árbol de decisión sin poda que considera k atributos elegidos aleatoriamente en cada nodo [94].		Regresión Clasificación
***REPTree* (REPT)**	Construye un árbol de decisión / regresión usando la ganancia / varianza de la información y lo poda utilizando la poda del error reducido (con backfitting), (i.e. as in *C4.5*).		Regresión Clasificación
***Bagging* (BAG)**	Enfoque de aprendizaje de clasificación múltiple para reducir la varianza, por defecto utiliza un algoritmo *REPTree* como algoritmo base [95].		Regresión Clasificación
***Stacking* (STA)**	Enfoque de aprendizaje de clasificación múltiple que combina varios clasificadores utilizando un método de apilamiento [96].		Regresión
***Adaboost* (ADA)**	Basado en *Boosting*[97].		Clasificación
***Additive Regression* (AR)**	Enfoque de aprendizaje de clasificación múltiple que mejora el desempeño de un clasificador base de regresión, reduce la tasa de aprendizaje y previene el sobreajusteque (overfitting)[98].		Regresión
***Regression by discretization* (RBD)**	Enfoque de aprendizaje de clasificación múltiple que admite la estimación de densidad condicional mediante la construcción de un estimador de densidad univariante a partir de los valores objetivo en los datos de entrenamiento. Utiliza un algoritmo J48 como base[99].		Regresión
***M5Rules* (M5R)**	Algoritmo basado en reglas que genera una lista de decisiones para problemas de regresión utilizando divide y vencerás [100][90][91].		Regresión
***Linear* (LRM)**	Un modelo de regresión lineal con solo intercepto y términos lineales.	*Matlab*	Regresión
***Interactions linear* (IL)**	Un modelo de regresión lineal con términos lineales, de interacción e intercepto.		Regresión
***Robust linear* (RL)**	Un modelo de regresión lineal robusto con solo intercepto y términos lineales.		Regresión
Stepwise linear regression (SLR)	Un modelo de regresión lineal con términos determinados por un algoritmo *stepwise* (paso a paso) que agrega o recorta automáticamente el modelo.		Regresión
Árboles de regresión [45][46]			
***Fine tree* (FT)**	Un árbol de regresión fino (el tamaño mínimo de la hoja es 4).		Regresión
***Medium tree* (MT)**	Un árbol de regresión mediano (el tamaño mínimo de la hoja es 12).		Regresión
***Coarse tree* (CT)**	Un árbol de regresión grueso (el tamaño mínimo de la hoja es 36).		Regresión
Regresión SVM se considera una técnica no paramétrica porque se basa en funciones kernel[103][104]–[107]			
***Linear SVM* (LSVM)**	SVM que sigue una estructura lineal simple en los datos,		Regresión

	utilizando un kernel lineal. Es el SVM más fácil de interpretar.		
***Quadratic SVM* (QSVM)**	SVM que utiliza un kernel cuadrático.		Regresión
Cubic SVM (CSVM)	SVM que utiliza un kernel cúbico.		Regresión
***Fine Gaussian SVM* (FGSVM)**	SVM que sigue una estructura finamente detallada en los datos. Utiliza el kernel gaussiano con la escala de kernel sqrt (P) / 4, siendo P el número de predictores.		Regresión
***Medium Gaussian SVM* (MGSVM)**	SVM que utiliza una estructura menos fina en los datos. Utiliza el kernel gaussiano con la escala del kernel sqrt (P) / 4, siendo P el número de predictores.		Regresión
***Coarse Gaussian SVM* (CGSVM)**	SVM que sigue una estructura aproximada en los datos. Utiliza el kernel gaussiano con la escala del kernel sqrt (P) / 4, siendo P el número de predictores.		Regresión
Procesos gaussianos para regresión (GPR): son modelos no paramétricos basados en modelos probabilísticos [108]**.**			
***Rational Quadratic* (RQ)**	Un modelo GPR que utiliza un kernel cuadrático racional.		Regresión
Squared exponential (SE)	Un modelo GPR que utiliza un kernel exponencial al cuadrado.		Regresión
***Matern 5/2* (M5)**	Un modelo GPR que utiliza un kernel *matern 5/2*.		Regresión
***Exponential* (EM)**	Un modelo GPR que utiliza un kernel exponencial.		Regresión
***Rational Quadratic* (RQ)**	Un modelo GPR que utiliza un kernel cuadrático racional.		Regresión
***Boosted trees* (BOT)**	*Boosting* de árboles de regresión que utiliza el algoritmo *LSBoost.*		Regresión
***Bagged trees* (BAT)**	*Bagging* de árboles de regresión.		Regresión

4 Análisis y discusión de los resultados.

En este epígrafe se realizan varios experimentos para evaluar las capacidades de los algoritmos de regresión y clasificación a través de la metodología propuesta por Demšar [18] y ampliada en los trabajos de [19][20][21] con la ayuda del paquete para pruebas estadísticas *SCMAMP* de R [85].

Para estas evaluaciones, en primer lugar, se aplica un método estadístico no paramétrico de comparación múltiple como es el test de Friedman para análisis de varianzas por rangos[109][110][111] para k muestras relacionadas ($k > 2$), el cual detecta si al menos dos grupos de entre los comparados son significativamente diferentes. Otra alternativa, es el test de rangos con signo de Wilcoxon[112] para cuando $k = 2$, etc. [18]. Luego, en una segunda etapa, una vez obtenido el ranking de los algoritmos se determina la superioridad o no de cada uno con respecto a los demás y si esta superioridad es estadísticamente significativa. Para ello recurrimos a los procedimientos *post hoc* del test de Wilcoxon [112] para ajustar los *p-value* obtenidos en lugar de utilizar Friedman, como sugiere Benavoli et *al.* [21], mediante comparaciones $1 \times N$ (1

against all) o (N × N) (*all against all*) [19][20]. Los procedimientos *post-hoc* son necesarios ya que en el análisis por pares, si intentamos sacar una conclusión que implique más de una comparación por pares, acumulamos un error proveniente de su combinación [12].

4.1 Top 10 de algoritmos de regresión.

En la primera simulación nos centramos en determinar el comportamiento de los algoritmos de regresión cuantificando su desempeño de acuerdo al RMSE en regresión mediante el análisis de 141 conjuntos de datos y 37 algoritmos de ML. No se realiza preprocesamiento de los datos ya que si el preprocesamiento favorece algún algoritmo con respecto a otros, este impacto debe ser aleatorio, y por tanto no estadísticamente significativo para la comparación[16].

Para comenzar, la figura 5 muestra el promedio del RMSE alcanzado por cada algoritmo de regresión para cada conjunto de datos seleccionados (véase tabla 2). Los resultados muestran que *RQ* es el mejor algoritmo en promedio, aunque esto no es concluyente, se necesitan test estadísticos que lo corroboren, pero nos da una idea sobre el comportamiento promedio en base al RMSE de los algoritmos frente a los conjuntos de datos. Le sigue EM también disponible en MATLAB.

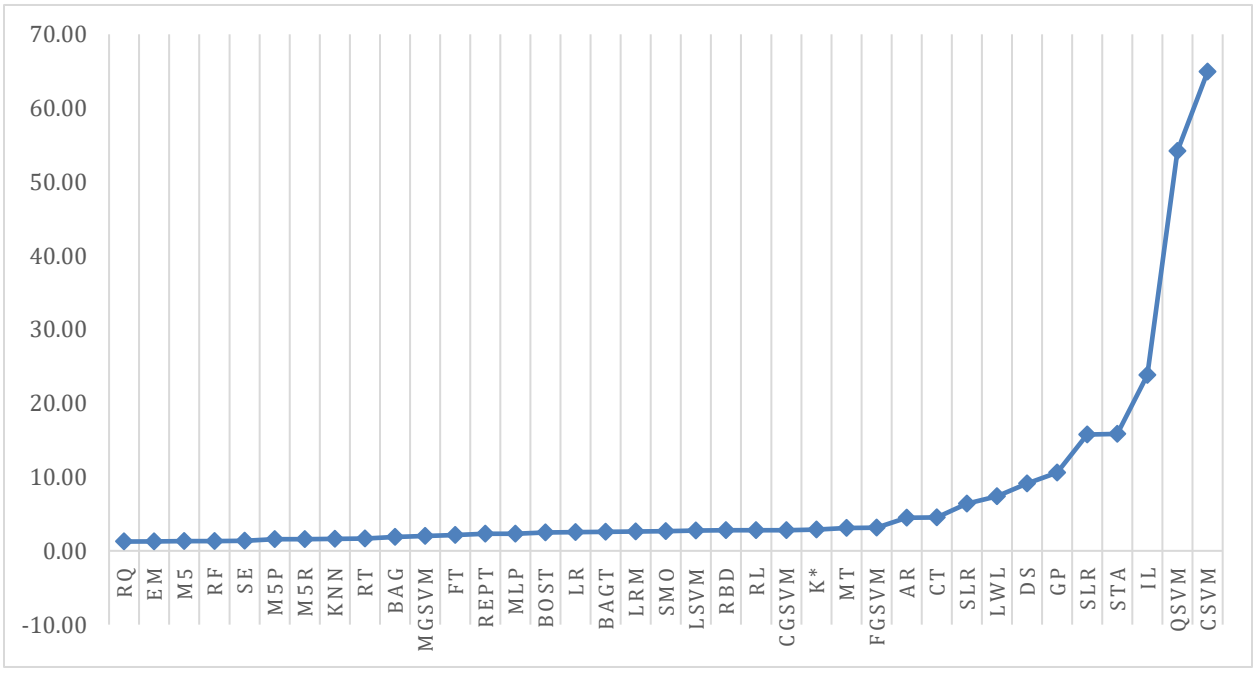

Figura 5. *RMSE* promedio de los algoritmos de regresión lineal.

Luego, se aplica la corrección de Iman y Davemport [113] al test de Friedman obteniendo como resultado el rechazo de la hipótesis nula ($p\text{-}value < 2.2E-16 < 0{,}05$) para un chi-cuadrado = 95.907, diferencias significativas muy altas entre al menos 2 algoritmos. La figura 6 muestra el ranking calculado por esta prueba para cada algoritmo donde *M5* es el mejor ranquin con 30.38 (mayor valor) a pesar de no alcanzar el menor promedio del *RMSE* mientras que *STA* con 5.93 (menor valor) es el de peor desempeño de todos.

La próxima etapa es determinar si la superioridad de los algoritmos es estadísticamente significativa o no. Recurrimos al test de Wilcoxon y a varios procedimientos *post-hoc* para ajustar los *p-value* obtenidos. Se utilizan los test de *Holm, Hommel* y *Hochberg* asumiendo en este trabajo la validación de la hipótesis nula H_0 si al menos uno de los procedimientos *post-hoc* la rechaza. Para presentar los resultados de forma sencilla, las figuras 7, 8 y 9 similares a las gráficas de diferencias críticas, se agrupan (conectan) aquellos algoritmos que no poseen diferencias significativas entre ellos para un valor de significación α=0,05, o sea, todos aquellos algoritmos que no tienen diferencias significativas entre si se dibujan con una línea gruesa horizontal.

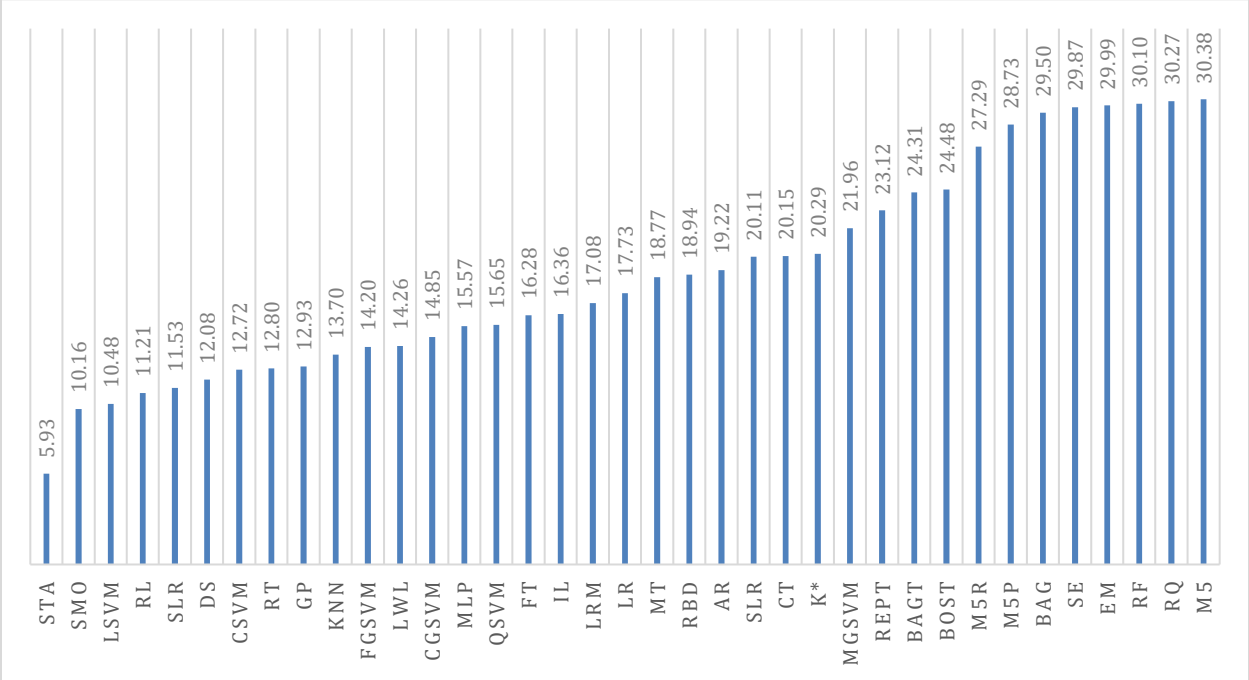

Figura 6. Rangos de Friedman para los algoritmos de regresión.

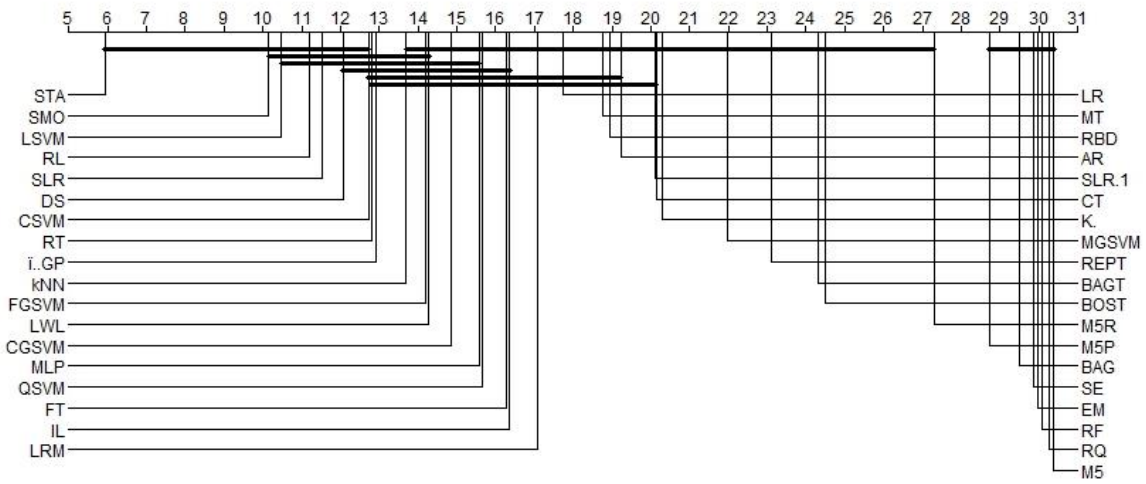

Figura 7. Diferencias significativas según el test *post-hoc* de Holm.

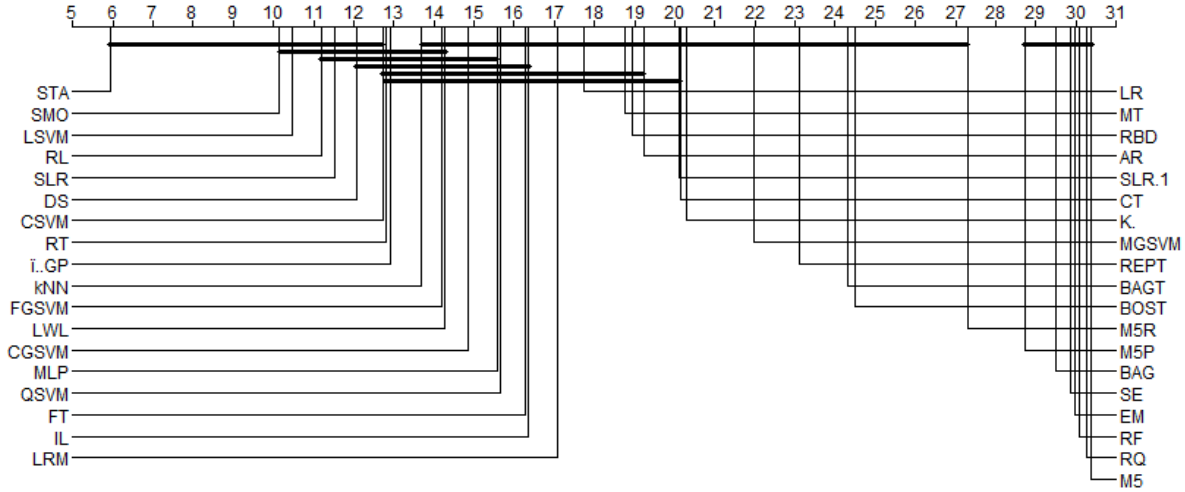

Figura 8. Diferencias significativas según el test *post-hoc* de Hommel.

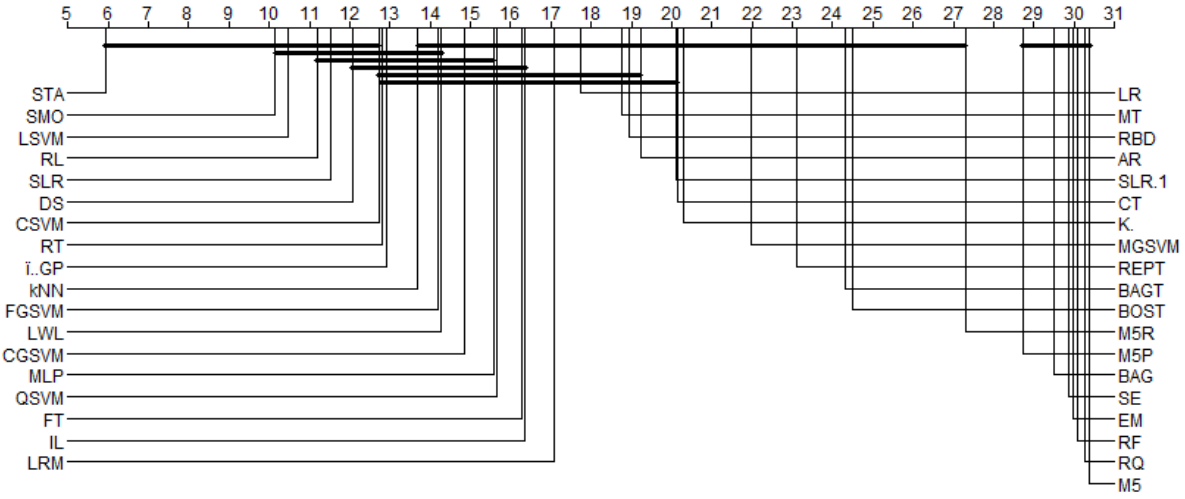

Figura 9. Diferencias significativas según el test *post-hoc* de Hochberg.

Los resultados apuntan al hecho de que *M5* se erige como el algoritmo de mejor desempeño en nuestro estudio, sin diferencias significativas con RQ, RF, EM, SE, BAG y M5P ya que ninguno de los 3 test *post-hoc* rechazaron la hipótesis nula H_0. Observar que *M5* y *RF* tienen un promedio del RMSE superior a *RQ* y son capaces de funcionar de la misma manera que *RQ.* Seguidamente se observa como *M5R, BOST, BAGT, MGSVM,* etc. no tienen diferencias significas entre sí en su comportamiento. Igualmente es pertinente indicar que *RL* tiene diferencias significativas con LSVM ya que *Holm* rechazó H_0, por solo mencionar un ejemplo.

También como M5, RQ, RF, EM, SE, BAG y M5P los mejores algoritmos de este estudio, así como BOST, BADT, REPT que le suceden para completar un top 10 de algoritmos de regresión lineal están basados en arboles de decisión, procesos gaussianos y multiclasificadores que utilizan como base un árbol de decisión.

Este análisis confirma la confiabilidad de un top 10 de algoritmos de regresión lineal para resolver problemas de ML con una amplia gama de características, recordemos el teorema de *No-Free-Lunch,* ningún algoritmo puede ser siempre el mejor, advirtiendo contra la tendencia a buscar el mejor algoritmo de aprendizaje. Además, enfatizar que muchas de estas implementaciones tienen un alto costo computacional a medida que aumentan las características.

4.2 Análisis de problemas utilizando selección de características.

Los métodos de selección de características utilizados en esta sección para el análisis de los conjuntos de datos con ruido artificial son del tipo selección de subconjuntos. *CFS* (Correlation-based Feature Selection) para regresión con el método de búsqueda *BestFirst* y con el *GreedyStepwise* para clasificación. La figura 10 muestra el promedio del índice *kappa* y la Figura 11 el promedio según el *RMSE* para regresión lineal (color verde algoritmo aplicado con selección de características[1] y azul para el caso contrario) para todos los conjuntos de datos. Como se ha mencionado antes, estas figuras son solo descriptivas, solo da una visión general del comportamiento promedio de las evaluaciones realizadas.

Como segundo experimento, nos apoyamos en el test de Wilcoxon para 2 muestras relacionadas con el objetivo de identificar diferencias significativas entre pares de algoritmos, asumiendo un grado de significación de 0.05, indicándose rechazo de H_0, si el *p-value* < 0,05. La tabla 6 muestra los *p-value* corregidos según el test de, donde por ejemplo *SMO* vs *SMO*[1] no tienen diferencias significativas entre si ya que se acepta la hipótesis nula y así sucesivamente para los demás casos. Vemos que en muchos casos en donde se rechaza H_0 existe un margen muy estrecho en cuanto a promedio en las evaluaciones dadas por las medidas *kappa* y *RMSE*.

Figura 10. Índice *kappa* promedio de los algoritmos de clasificación.

Figura 11. *RMSE* promedio de los algoritmos de regresión.

Tabla 6. Diferencias significativas de acuerdo al test de Wilcoxon.

	Clasificación			**Regresión**	
Algoritmos	***p-value***	***Hipótesis***	***Algoritmos***	***p-value***	***Hipótesis***
SMO vs SMO[1]	0.1479	Aceptada	RQ vs RQ[1]	0.2057	Aceptada
MLP vs MLP[1]	0.03368	Rechazada	SE vs SE[1]	0.02846	Rechazada
SLOG vs SLOG[1]	0.4114	Aceptada	M5 vs M5[1]	0.2057	Aceptada
ADA vs ADA[1]	0.02846	Rechazada	EM vs EM[1]	0.1314	Aceptada
BAG vs BAG[1]	0.2628	Aceptada	RF vs RF[1]	0.0002967	Rechazada
J48 vs J48[1]	0.005868	Aceptada	BAG vs BAG[1]	0.006871	Rechazada
RF vs RF[1]	0.02611	Rechazada	M5P vs M5P[1]	0.1022	Aceptada

LWL vs LWL[1]	0.06768	Aceptada	M5R vs M5R[1]	0.3826	Aceptada
REPT vs REPT[1]	0.02002	Rechazada	BOST vs BOST[1]	0.001248	Rechazada
RT vs RT[1]	0.1706	Aceptada	BAGT vs BAGT[1]	0.1395	Aceptada
Nota: [1] *Algoritmo modificado con selección de características.*					

Luego, se utiliza el test de Wilcoxon para poder obtener conclusiones más fiables que el promedio de las evaluaciones sobre el conjunto de datos de estas medidas (*kappa* y *RMSE*). En la gráfica 11 y 12, se muestra una descripción de los rangos de Wilcoxon para cada par de algoritmos. Destacar que para los algoritmos de regresión lineal se utiliza un enfoque de minimización ya que utilizamos *RMSE*, esto significa que el mejor algoritmo es el de mayor ranking de Wilcoxon. Lo contrario sucede para *kappa*, donde se utiliza un enfoque de maximización y el mejor raqueado es el algoritmo con menor valor de rangos de Wilcoxon.

Figura 12. Rangos del test de Wilcoxon para las comparaciones 1 a 1 de los algoritmos de clasificación.

Figura 13. Rangos del test de Wilcoxon para las comparaciones 1 a 1 de los algoritmos de regresión lineal.

Resulta útil la selección de características ya que con menos características se mejora la capacidad predictiva de los modelos, proporcionando modelos predictivos más rápidos y eficientes sin afectar a la calidad de la solución.

5 Conclusiones

A través de experimentos apropiados, se analizó el desempeño de 37 algoritmos de regresión mediante 141 conjuntos de datos de ML asociados con problemas de regresión estándar con diferente dimensionalidad, con presencia de ruido y valores perdidos. Esta heterogeneidad en la información permitió tener una visión global de los algoritmos frente a los diferentes conjuntos de datos, utilizando el RMSE como métrica y siguiendo la metodología propuesta por Demšar para comparar múltiples algoritmos sobre múltiples conjuntos de datos.
El algoritmo más eficaz fue *M5* basado en un proceso gaussiano según el test *post-hoc* de Friedman con un comportamiento estadísticamente similar a *RQ*, *RF*, *EM*, *SE*, *BAG* y *M5P* de acuerdo a los *post-hoc* de Wilcoxon, es decir, sin diferencias significativas estadísticamente.

De igual manera mediante una experimentación con 20 conjuntos de datos de regresión y clasificación se demostró de manera sencilla como la selección de características mejora la creación de modelos de ML a partir de datos que contienen altos niveles de ruido, siendo esta

diferencia no estadísticamente significativa para la mayoría de los conjuntos de datos seleccionados mediante en la experimentación.

Referencias

[1] R. Fernandes de Mello and M. Antonelli Ponti, *Machine Learning*. Cham: Springer International Publishing, 2018.

[2] R. Fernandes de Mello, M. Antonelli Ponti, R. Fernandes de Mello, and M. Antonelli Ponti, “A Brief Review on Machine Learning,” in *Machine Learning*, Springer International Publishing, 2018, pp. 1–74.

[3] R. Kohavi, “The power of decision tables,” 1995, doi: 10.1007/3-540-59286-5_57.

[4] S. B. Kotsiantis, “Decision trees: A recent overview,” *Artificial Intelligence Review*. 2013, doi: 10.1007/s10462-011-9272-4.

[5] H. Ishibuchi, T. Nakashima, and T. Morisawa, “Voting in fuzzy rule-based systems for pattern classification problems,” *Fuzzy Sets Syst.*, 1999, doi: 10.1016/S0165-0114(98)00223-1.

[6] G. P. Zhang, “Neural networks for classification: A survey,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, 2000, doi: 10.1109/5326.897072.

[7] R. G. Brereton and G. R. Lloyd, “Support Vector Machines for classification and regression,” *Analyst*. 2010, doi: 10.1039/b918972f.

[8] R. Duda, P. Hart, and D. Stork, “Patterns Classification,” *John Wiley Sons*, 2012.

[9] G. Nápoles, I. Grau, E. Papageorgiou, R. Bello, and K. Vanhoof, “Rough Cognitive Networks,” *Knowledge-Based Syst.*, 2016, doi: 10.1016/j.knosys.2015.10.015.

[10] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. 2016.

[11] N. Settouti, M. E. A. Bechar, and M. A. Chikh, “Statistical Comparisons of the Top 10 Algorithms in Data Mining for Classication Task,” *Int. J. Interact. Multimed. Artif. Intell.*, 2016, doi: 10.9781/ijimai.2016.419.

[12] G. Nápoles, R. Falcon, E. Papageorgiou, R. Bello, and K. Vanhoof, “Rough cognitive

ensembles," *Int. J. Approx. Reason.*, 2017, doi: 10.1016/j.ijar.2017.03.011.

[13] G. Felix, G. Nápoles, R. Falcón, R. Bello, and K. Vanhoof, "Performance analysis of granular versus traditional neural network classifiers: Preliminary results," 2018, doi: 10.1109/CIVEMSA.2018.8439971.

[14] G. Nápoles, C. Mosquera, R. Falcon, I. Grau, R. Bello, and K. Vanhoof, "Fuzzy-Rough Cognitive Networks," *Neural Networks*, vol. 97, pp. 19–27, Jan. 2018, doi: 10.1016/j.neunet.2017.08.007.

[15] B. H. Baltagi, *Econometrics*, 5th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[16] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *J. Mach. Learn. Res.*, 2014, doi: 10.1117/1.JRS.11.015020.

[17] D. H. Wolpert, "The Lack of a Priori Distinctions between Learning Algorithms," *Neural Comput.*, 1996, doi: 10.1162/neco.1996.8.7.1341.

[18] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, 2006.

[19] S. García and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, 2008.

[20] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci. (Ny).*, 2010, doi: 10.1016/j.ins.2009.12.010.

[21] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?," *J. Mach. Learn. Res.*, 2016.

[22] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012.

[23] A. J. Tallón-Ballesteros and J. C. Riquelme, "Deleting or keeping outliers for classifier

training?," 2014, doi: 10.1109/NaBIC.2014.6921892.

[24] J. Li *et al.*, "Feature selection: A data perspective," *ACM Computing Surveys*. 2017, doi: 10.1145/3136625.

[25] L. Ladha and T. Deepa, "Feature Selection Methods And Algorithms," *International Journal on Computer Science and Engineering*. 2011.

[26] K. Ro, C. Zou, Z. Wang, and G. Yin, "Outlier detection for high-dimensional data," *Biometrika*, 2015, doi: 10.1093/biomet/asv021.

[27] P. Filzmoser, R. Maronna, and M. Werner, "Outlier identification in high dimensions," *Comput. Stat. Data Anal.*, 2008, doi: 10.1016/j.csda.2007.05.018.

[28] A. Zimek, E. Schubert, and H. P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining*. 2012, doi: 10.1002/sam.11161.

[29] H. Liu, X. Li, J. Li, and S. Zhang, "Efficient Outlier Detection for High-Dimensional Data," *IEEE Trans. Syst. Man, Cybern. Syst.*, 2018, doi: 10.1109/TSMC.2017.2718220.

[30] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," 2001, doi: 10.1145/375663.375668.

[31] J. Han, M. Kamber, and J. Pei, "Outlier Detection," in *Data Mining*, 2012.

[32] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study of their impacts," *Artif. Intell. Rev.*, 2004.

[33] S. Chen, W. Wang, and H. van Zuylen, "A comparison of outlier detection algorithms for ITS data," *Expert Syst. Appl.*, 2010, doi: 10.1016/j.eswa.2009.06.008.

[34] Y. Tu, "Machine learning," in *EEG Signal Processing and Feature Extraction*, 2019.

[35] A. K. Tiwari, "Introduction to machine learning," *Ubiquitous Machine Learning and Its Applications*. 2017, doi: 10.4018/978-1-5225-2545-5.ch001.

[36] S. B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” *Informatica (Ljubljana)*. 2007, doi: 10.31449/inf.v31i3.148.

[37] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, 2006, doi: 10.1016/j.patrec.2005.10.010.

[38] L. Chaitow, “Accuracy,” *Journal of Bodywork and Movement Therapies*. 2004, doi: 10.1016/j.jbmt.2004.03.001.

[39] C. C. Berry, “The kappa statistic,” *JAMA J. Am. Med. Assoc.*, 1992, doi: 10.1001/jama.268.18.2513.

[40] M. L. McHugh, “Interrater reliability: The kappa statistic,” *Biochem. Medica*, 2012, doi: 10.11613/bm.2012.031.

[41] B. Di Eugenio and M. Glass, “The Kappa Statistic: A Second Look,” *Comput. Linguist.*, 2004, doi: 10.1162/089120104773633402.

[42] D. S. Young, *Handbook of Regression Methods*. 2018.

[43] D. J. Olive, *Linear regression*. 2017.

[44] T. A. Stapleford, “Econometrics,” in *Modernism and the Social Sciences: Anglo-American Exchanges, c.1918-1980*, 2017.

[45] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Comput. Electr. Eng.*, 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[46] N. K. Verma and A. Salour, “Feature selection,” in *Studies in Systems, Decision and Control*, 2020.

[47] J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review,” in *Data Classification: Algorithms and Applications*, 2014.

[48] J. Miao and L. Niu, “A Survey on Feature Selection,” 2016, doi: 10.1016/j.procs.2016.07.111.

[49] V. Kumar, “Feature Selection: A literature Review,” *Smart Comput. Rev.*, 2014, doi: 10.6029/smartcr.2014.03.007.

[50] L. I. Smith, “A tutorial on Principal Components Analysis Introduction,” *Statistics (Ber).*, 2002.

[51] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*. 2010, doi: 10.1002/wics.101.

[52] A. Maćkiewicz and W. Ratajczak, “Principal components analysis (PCA),” *Comput. Geosci.*, 1993, doi: 10.1016/0098-3004(93)90090-R.

[53] E. Bingham and H. Mannila, “Random projection in dimensionality reduction: Applications to image and text data,” 2001.

[54] D. Fradkin and D. Madigan, “Experiments with random projections for machine learning,” 2003, doi: 10.1145/956750.956812.

[55] C. M. Lewandowski, N. Co-investigator, and C. M. Lewandowski, “Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning,” *Eff. Br. mindfulness Interv. acute pain Exp. An Exam. Individ. Differ.*, 2015, doi: 10.1017/CBO9781107415324.004.

[56] M. Hall and L. a Smith, “Feature Selection for Machine Learning : Comparing a Correlation-based Filter Approach to the Wrapper CFS : Correlation-based Feature,” *Int. FLAIRS Conf.*, 1999, doi: 10.1.1.50.2192.

[57] E. Tuv, A. Borisov, G. Runger, and K. Torkkola, “Feature selection with ensembles, artificial variables, and redundancy elimination,” *J. Mach. Learn. Res.*, 2009.

[58] D. Mladenić, “Feature selection for dimensionality reduction,” 2006, doi: 10.1007/11752790_5.

[59] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,” 2003.

[60] G. Hua, M. Zhang, Y. Liu, S. Ma, and L. Ru, “Hierarchical feature selection for ranking,”

in *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, 2010, pp. 1113–1114, doi: 10.1145/1772690.1772830.

[61] Z. Shi, J. Keung, K. E. Bennin, and X. Zhang, "Comparing learning to rank techniques in hybrid bug localization," *Appl. Soft Comput. J.*, vol. 62, pp. 636–648, Jan. 2018, doi: 10.1016/j.asoc.2017.10.048.

[62] G. Brown, "A new perspective for information theoretic feature selection," 2009.

[63] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," in *Machine Learning Proceedings 1994*, 1994.

[64] J. Yang and V. Honavar, "Feature subset selection using genetic algorithm," *IEEE Intell. Syst. Their Appl.*, 1998, doi: 10.1109/5254.671091.

[65] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997, doi: 10.1016/s0004-3702(97)00043-x.

[66] R. Kohavi and H. John, "Artificial Intelligence Wrappers for feature subset selection," *Artif. Intell.*, 1997, doi: 10.1016/S0004-3702(97)00043-X.

[67] K. Kira and L. A. Rendell, "A Practical Approach to Feature Selection," in *Machine Learning Proceedings 1992*, 1992.

[68] S. Yijun and L. Jian, "Iterative RELIEF for feature weighting," 2006, doi: 10.1145/1143844.1143959.

[69] R. P. L. DURGABAI and R. B. Y, "Feature Selection using ReliefF Algorithm," *IJARCCE*, 2014, doi: 10.17148/ijarcce.2014.31031.

[70] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1994, vol. 784 LNCS, pp. 171–182, doi: 10.1007/3-540-57868-4_57.

[71] A. K. Pandey and D. S. Rajpoot, "A comparative study of classification techniques by utilizing WEKA," 2016, doi: 10.1109/ICSPCom.2016.7980579.

[72] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: An update,” *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[73] E. A. Sobie, “An introduction to MATLAB,” in *Science Signaling*, Sep. 2011, vol. 4, no. 191, doi: 10.1126/scisignal.2001984.

[74] Ö. Özgün and Ö. Özgün, “MATLAB Primer,” in *MATLAB®-based Finite Element Programming in Electromagnetic Modeling*, 2018.

[75] N. Macià and E. Bernadó-Mansilla, “Towards UCI+: A mindful repository design,” *Inf. Sci. (Ny).*, 2014, doi: 10.1016/j.ins.2013.08.059.

[76] J. Alcalá-Fdez *et al.*, “KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework,” *J. Mult. Log. Soft Comput.*, 2011.

[77] X. Zhu, X. Wu, and Y. Yang, “Error detection and impact-sensitive instance ranking in noisy datasets,” 2004.

[78] J. G. Moreno-Torres, J. A. Saez, and F. Herrera, “Study on the impact of partition-induced dataset shift on k-fold cross-validation,” *IEEE Trans. Neural Networks Learn. Syst.*, 2012, doi: 10.1109/TNNLS.2012.2199516.

[79] I. Triguero, S. García, and F. Herrera, “Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study,” *Knowl. Inf. Syst.*, 2015, doi: 10.1007/s10115-013-0706-y.

[80] R. Hecht-Nielsen, “Theory of the backpropagation neural network,” 1989, doi: 10.1109/ijcnn.1989.118638.

[81] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to the SMO algorithm for SVM regression,” *IEEE Trans. Neural Networks*, 2000, doi: 10.1109/72.870050.

[82] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Min. Knowl. Discov.*, 1998, doi: 10.1023/A:1009715923555.

[83] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to Platt’s SMO algorithm for SVM classifier design,” *Neural Comput.*, 2001, doi: 10.1162/089976601300014493.

[84] M. Sumner, E. Frank, and M. Hall, “Speeding up Logistic Model Tree induction,” 2005, doi: 10.1007/11564126_72.

[85] N. Landwehr, M. Hall, and E. Frank, “Logistic model trees,” *Mach. Learn.*, 2005, doi: 10.1007/s10994-005-0466-3.

[86] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-Based Learning Algorithms,” *Mach. Learn.*, 1991, doi: 10.1023/A:1022689900470.

[87] M. L. Zhang and Z. H. Zhou, “ML-KNN: A lazy learning approach to multi-label learning,” *Pattern Recognit.*, 2007, doi: 10.1016/j.patcog.2006.12.019.

[88] J. G. Cleary and L. E. Trigg, “K*: An Instance-based Learner Using an Entropic Distance Measure,” in *Machine Learning Proceedings 1995*, 1995.

[89] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally Weighted Learning,” *Artif. Intell. Rev.*, 1997, doi: 10.1007/978-94-017-2053-3_2.

[90] J. R. Quinlan, “Learning with continuous classes,” 1992.

[91] Y. Wang and I. H. Witten, “Induction of model trees for predicting continuous classes,” *Proceedings of the 9th European Conference on Machine Learning Poster Papers*. 1997.

[92] L. Breiman, “Random forests,” *Mach. Learn.*, 2001, doi: 10.1023/A:1010933404324.

[93] Y. L. Pavlov, *Random forests*. 2019.

[94] Y. Amit and D. Geman, “Shape Quantization and Recognition with Randomized Trees,” *Neural Comput.*, 1997, doi: 10.1162/neco.1997.9.7.1545.

[95] L. Breiman, “Bagging predictors,” *Mach. Learn.*, 1996, doi: 10.1007/bf00058655.

[96] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, 1992, doi: 10.1016/S0893-

6080(05)80023-1.

[97] Y. Freund and R. E. Schapire, “Experiments with a New Boosting Algorithm DRAFT-PLEASE DO NOT DISTRIBUTE,” 1996. Accessed: Sep. 20, 2020. [Online]. Available: http://www.research.att.com/orgs/ssr/people/fyoav,schapireg/.

[98] J. H. Friedman, “Stochastic gradient boosting,” *Comput. Stat. Data Anal.*, 2002, doi: 10.1016/S0167-9473(01)00065-2.

[99] E. Frank and R. R. Bouckaert, “Conditional density estimation with class probability estimators,” 2009, doi: 10.1007/978-3-642-05224-8_7.

[100] G. Holmes, M. Hall, and E. Prank, “Generating rule sets from model trees,” 1999, doi: 10.1007/3-540-46695-9_1.

[101] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. 2017.

[102] A. D. Gordon, L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, “Classification and Regression Trees.,” *Biometrics*, 1984, doi: 10.2307/2530946.

[103] P. H. Chen, R. E. Fan, and C. J. Lin, “A study on SMO-type decomposition methods for support vector machines,” *IEEE Trans. Neural Networks*, 2006, doi: 10.1109/TNN.2006.875973.

[104] R. E. Fan, P. H. Chen, and C. J. Lin, “Working set selection using second order information for training support vector machines,” *J. Mach. Learn. Res.*, 2005.

[105] T.-M. Huang, V. Kecman, and I. Kopriva, *Kernel based algorithms for mining huge data sets*, vol. 1. Springer, 2006.

[106] J. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” 1998.

[107] V. N. Vapnik, *The Nature of Statistical Learning Theory*. 1995.

[108] C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Regression,” *Gaussian*

Process. Mach. Learn., 2006, doi: 10.1093/bioinformatics/btq657.

[109] M. Friedman, “The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance,” *J. Am. Stat. Assoc.*, 1937, doi: 10.2307/2279372.

[110] M. Friedman, “A Comparison of Alternative Tests of Significance for the Problem of ‘m’ Rankings,” *Ann. Math. Stat.*, vol. 11, no. 1, pp. 86–92, 1940, doi: 10.1214/aoms/1177731944.

[111] T. Hoskin, “Parametric and nonparametric: demystifying the terms,” *Ctsa.Mayo.Edu*, 2010.

[112] F. Wilcoxon, “Individual Comparisons by Ranking Methods,” 1992.

[113] J. M. Davenport, “Approximations of the critical region of the friedman statistic,” *Commun. Stat. - Theory Methods*, 1980, doi: 10.1080/03610928008827904.