MEMORIA DE ANÁLISIS DE DATOS

Estudio de los datos de compraventa de ciclomotores de un portal web

GRUPO III

Amalio Cabeza Palacios

Rafael Delgado

Álvaro Navarro Mora

Lea Ross

Curso: 2021-2022

ÍNDICE

1. INTRODUCCIÓN

En base a las técnicas y conocimientos adquiridos en la asignatura Fundamentos en Ingeniería de Datos, presentamos a continuación la propuesta para la mejora del manejo de un sistema de ventas del portal web de la India *droom.in*, a partir del desarrollo de una herramienta que recomiende el mejor precio para un nuevo artículo. El conjunto de datos utilizado para llevar a cabo este trabajo ha sido obtenido gracias a la web de recursos de datos *kaggle.com*.

Mediante el manejo de librerías de análisis de datos en R, extraemos la información relevante del conjunto de datos obtenido de la citada web, con el objetivo de construir un modelo que permita predecir una variable objetivo -precio-, en base a la venta de productos de características similares publicados en la web.

La siguiente figura ilustra brevemente la metodología seguida para organizar el proyecto, en ella podemos ver los pasos y técnicas tratadas durante la experimentación. Podemos apreciar que los pasos siguen un orden secuencial fundamentalmente, pero permitiendo siempre la mejora de un paso anterior si fuese necesario. A continuación, se detallarán todas estas acciones mostrando la relevancia y utilidad que cada una desempeña.

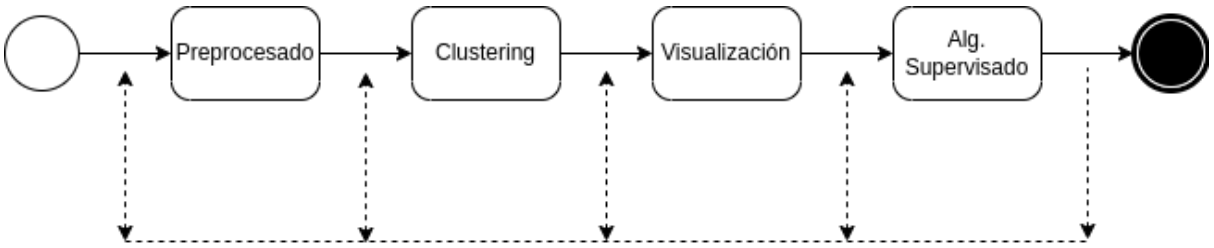

Figura 1. Diagrama de flujo de trabajo de los métodos realizados.

2. PREPROCESADO DE LOS DATOS

Según Dorian Pyle, el propósito de esta etapa consiste en *manipular y transformar datos sin procesar para que la información contenida en el conjunto de datos se pueda exponer o hacer más fácilmente accesible*. En este apartado explicaremos qué cambios se han aplicado sobre los datos para obtener un nuevo conjunto con el que trabajar en la experimentación de secciones futuras.

En las siguientes figuras se muestra una primera tabla con la estructura inicial del conjunto de datos del cuál partíamos, y a continuación un breve resumen con el contenido de los datos.

Model Name	Model Year	KMS driven	Owner	Location	Mileage	Power	Price
Nombre y marca del vehículo.	Año de construcción del vehículo	Total de kilómetros conducidos.	Representación del número de dueños del vehículo.	Localización del vendedor.	Consumo medio del vehículo (kmpl).	Caballos de potencia del vehículo.	Precio de venta del vehículo.

Figura 2. Descripción de los datos originales.

	model name	kms driven	owner	location	mileage	power
Length	7857	7857	7857	7857	7857	7857
Class	character	character	character	character	character	character
Mode	character	character	character	character	character	character

Figura 3. Detalle de las columnas de texto.

	model_year	price
Min	1950	0
1st Qu	2014	42000
Median	2016	75000
Mean	2015	106791
3rd Qu	2018	125000
Max	2021	3000000

Figura 4. Detalle de las columnas numéricas.

Al visualizar los datos manualmente, pudimos apreciar sin mucha dificultad que los datos necesitaban algunos cambios:

- Detectamos que en la columna “model_name” contiene el año de construcción del vehículo, información redundante puesto que ya contamos con una columna con esa información.

- También en la columna "model_name" observamos que habitualmente viene información adicional, como la cilindrada y/o la marca de la moto.

- Las columnas "kms_driven" y mileage muestran la información de forma aparentemente inconsistente, es decir, parece que la información es correcta, pero los formatos que utilizan varían.

- En la columna "power" ocurre algo similar, ya que con los vehículos eléctricos esta columna indica la potencia en kW en lugar de en caballos de potencia.

Tras analizar minuciosamente los datos, procedimos a realizar con R los cambios pertinentes. Los más relevantes fueron los siguientes:

- A partir de la columna model_name se crearon tres columnas, una primera que almacene el modelo, otra la marca y otra la cilindrada del vehículo. Por último, se eliminó la primera columna.
- Cambiamos el nombre de la columna "mileage" por "consumption" de forma que su contenido esté representado de forma más clara. También se modifica su tipo a numérico.
- La columna "power" es renombrada a "Bhp", representando inequívocamente los caballos de potencia de cada vehículo. Los vehículos cuya potencia se medía en kW ha sido transformada en Bhp según la fórmula extraída de (https://www.rapidtables.com/convert/power/kw-to-bhp.html) :

  $$1\,hp \;=\; 0.745699872\,kW$$

  Debemos aclarar que no hemos perdido la información de las motocicletas eléctricas, ya que se generó una nueva columna, denominada "isElectric", que almacena dicha información.

- Se eliminó el texto innecesario de las columnas "kms_driven", mileage y power, luego cambiamos su tipado de forma que se considerasen variables numéricas.
- La columna "owner" original especifica cuántos propietarios ha tenido el vehículo mediante palabras (con los valores "first hand", "second hand", etc…), por lo cual, para trabajar de manera más sencilla con la columna, se decidió transformar el contenido por su equivalente numérico.
- La columna "price" contenía exactamente treinta y una instancias con ceros en su interior, consideramos que este es un valor imposible al tratarse de un portal web, por lo que, al ser pocas instancias, se decidió eliminarlas del conjunto de datos.

Cómo resultado de la etapa de preprocesamiento obtenemos los resultados expuestos en las siguientes tablas.

	model	location
Length	7826	7826
Class	character	character
Mode	character	character

	isElectric
Mode	logical
FALSE	7799
TRUE	27

Figuras 5 y 6. Detalle de las columnas no numéricas tras el preprocesado.

	model year	kms driven	owner	BHP	price	cilindrada
Min	1950	1	1.000	6.10	2000	100.0
1st Qu	2014	8901	1.000	14.00	42500	150.0
Median	2016	17000	1.000	19.00	75000	200.0
Mean	2015	23065	1.169	20.80	107214	242.6
3rd Qu	2018	30000	1.000	24.16	125000	350.0
Max	2021	1000000	4.000	197.30	3000000	959.0

Figura 7. Detalle de las columnas numéricas tras el preprocesado.

2. Tratamiento preliminar de los datos faltantes.

Una vez teníamos los datos con el formato indicado, era necesario plantearse un tratamiento de estos más avanzados y pensar cómo disminuir el número de valores omisos, o faltantes, con los que contaba el dataset.

Los datos faltantes o *missing values* (en inglés) son *valores no disponibles que serían útiles o significativos para el análisis de los resultados* (Dagnino Jorge). No es difícil suponer que la presencia de estos datos pueda suponer un problema si hay un gran número de ellos, en nuestro caso particular contábamos con un total de 2792 valores omisos.

No existe ninguna fórmula ni técnica perfecta con la que se pueda corregir estos valores, sin embargo, están establecidas varias soluciones que, si bien no aseguran ser una solución óptima, nos permiten

corregir estos datos perdidos con un coste computacional muy bajo. En primera instancia nos decidimos por una de estas, en concreto la solución fue asignar la media de la columna dónde hubiese una pérdida de información. De esta forma, ya podíamos comenzar el desarrollo de nuestro modelo objetivo. No obstante, cómo veremos en futuros apartados, se pueden realizar técnicas más sofisticadas para asignar estos valores; de forma que podamos aspirar a desarrollar un modelo más preciso.

3. APRENDIZAJE NO SUPERVISADO

El objetivo de las técnicas de aprendizaje no supervisado en este proyecto ha sido mejorar la calidad del preprocesado de los datos, en concreto el tratamiento de los valores nulos presentes en el dataset original. Se ha empleado el algoritmo K-Means para separar las distintas instancias en una serie de clusters, posteriormente se ha tomado la media de cada dato y cluster para reemplazar los valores nulos en las instancias que los presentaban.

3.1 Elección del número de clusters

Se ha hecho una experimentación para determinar el número de clusters óptimo para la separación de los datos, empleando dos métricas diferentes: la suma de los cuadrados intra-cluster o *Within Cluster Sum of Squares (WCSS)*, y el valor medio del coeficiente de la silueta, empleando el *Elbow Method.*

La suma de los cuadrados intra-cluster consiste en calcular la suma de las distancias euclídeas entre cada punto del cluster y su centroide asociado, dividirla entre el número de puntos y finalmente calcular la media entre todos los clusters. De esta forma determinamos que cuanto mayor sea esta medida, mayor similitud habrá entre los elementos de los clusters, por lo que su clasificación será mejor. Esta medida tiende a mejorar cuanto mayor sea el número de clusters escogido, por lo que debemos determinar a partir de qué punto deja de tener sentido seguir aumentando la cantidad de clusters, de lo contrario tendríamos tantos clusters como instancias y cada sería su propio centroide.

El valor medio del coeficiente de la silueta se calcula del siguiente modo:

a(x)= distancia promedio de x a todos los demás puntos en el mismo cluster.

b(x)= distancia promedio de x a todos los demás puntos en el cluster más cercano

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

$$SC = \frac{1}{N}\sum_{i=1}^{N} s(x)$$

Figuras 7 y 8. Cálculo del valor medio del coeficiente de la silueta.

Esta medida puede oscilar entre -1 y 1, representando 1 la mejor agrupación posible y -1 la peor.

Realizando la experimentación con estas dos métricas se han obtenido los siguientes resultados:

Número de clusters	WCSS %	Media del coeficiente de la silueta
2	60,6	0,87
3	74	0,59
4	84,1	0,56
5	88,1	0,55
6	90,5	0,48
7	91,3	0,49
8	92,8	0,48
9	93,8	0,45

Figuras 9. Resultados de las métricas de evaluación del aprendizaje no supervisado.

Como podemos observar la suma de los cuadrados intra-cluster aumenta con el número de clusters, mientras que la media del coeficiente de la silueta disminuye.

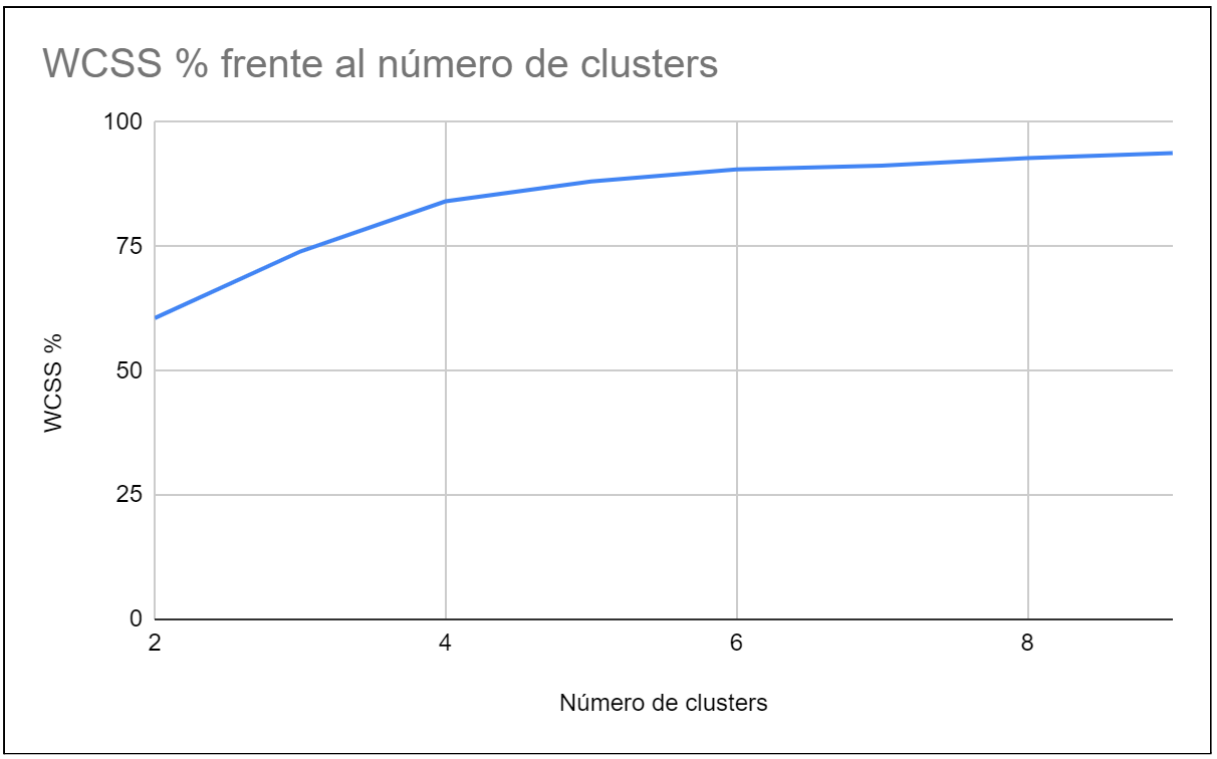

Figura 10. Diagrama del incremento de la suma de los cuadrados intra-cluster.

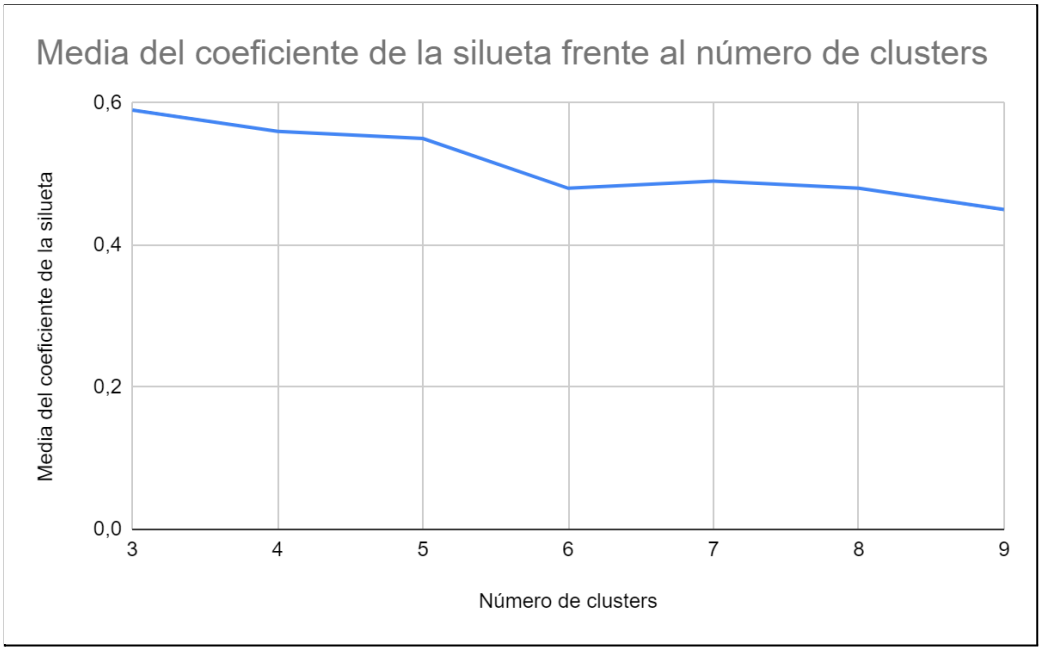

Figura 11. Diagrama de la media del coeficiente de la silueta.

Observando estos resultados hemos determinado que el número óptimo de clusters a emplear es 5, ya que combina un buen resultado con ambas métricas. Empleando este número de clusters obtenemos el siguiente resultado al calcular sus siluetas:

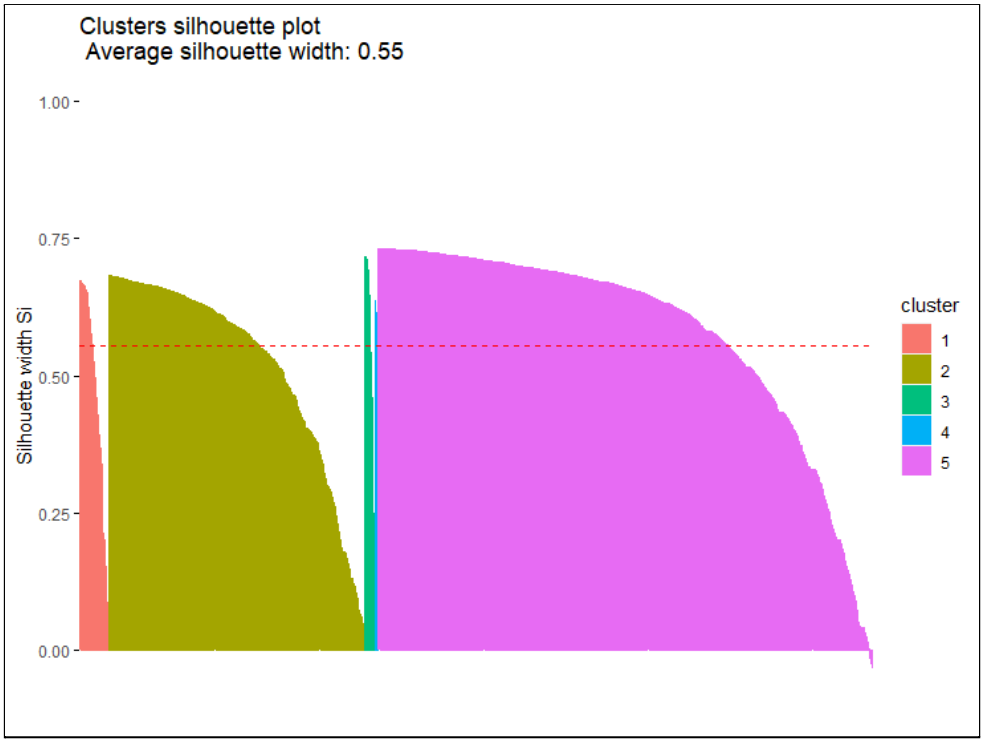

Figura 12. Gráfico de las siluetas de los clústers.

3.2 Tratamiento de los valores nulos empleando los resultados del clustering

El objetivo de este clustering es mejorar la calidad del tratamiento de datos, concretamente el reemplazo de los valores nulos. Originalmente los valores nulos se han tratado en la etapa de preprocesado, reemplazando estos valores por la media de todas las instancias del atributo en cuestión. Como mejora en este proceso, hemos aplicado la misma técnica, pero calculando la media únicamente de las instancias pertenecientes al mismo cluster de forma que el valor insertado será más similar al dato que la instancia debería haber tenido originalmente, y por tanto mejorando la calidad de los datos.

4. VISUALIZACIÓN

4.1 PowerBI

Para la visualización hemos seleccionado la herramienta de PowerBI. PowerBI permite una conexión directa con los datos y modelos, para visualizarlos con facilidad y ayudarnos en la creación de informes útiles. Hemos seleccionado esta herramienta dado que funciona bien con el formato csv, que es el formato de los datos después del preprocesado, y por la experiencia que tenemos con ella.

4.2 Procedimiento

Hemos realizado una conexión de PowerBI con GitHub posteriormente a realizar la etapa de preprocesado y añadir los datos al repositorio común. Posteriormente, los datos se analizaron en PowerBi para saber si era necesario hacer algún cambio adicional. Se ha encontrado que el formato de algunas columnas incluye números decimales, que se han convertido en números enteros para simplificar la presentación.

Para mejorar el valor informativo de las representaciones y aumentar la pertinencia, se crearon grupos de datos. Esto se hizo para las columnas de precio, kilómetros recorridos y BHP. Para el precio, se han realizado grupos de 5000, por los kilómetros recorridos grupos de 20.000 y por el BHP grupos de 20. En el siguiente paso, se creó una nueva columna para el precio en euros. La moneda original en los datos era la rupia india y no es muy común para los usuarios europeos. Para calcular el precio de las motos en euros, se utilizó el tipo de cambio actual que está sujeto a unas pocas fluctuaciones. La siguiente figura muestra el modelo de datos terminado en PowerBi con las columnas agregadas.

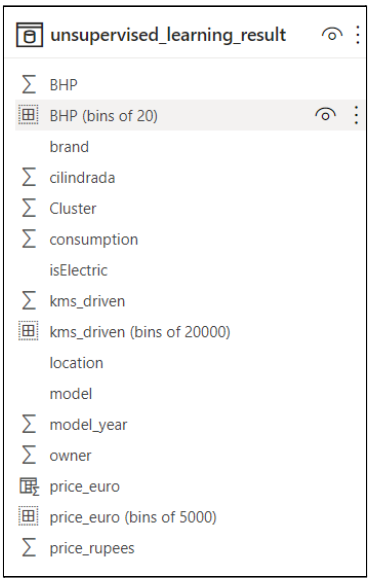

Figura 13. Modelo de datos en PowerBI

4.3 Los gráficos

El tablero creado en PowerBi es interactivo. Esto significa que todos los gráficos se adaptan automáticamente cuando se realiza una selección. Si, por ejemplo, se selecciona una determinada marca, solo se muestran los datos de las motos que son de esta marca.

La primera visualización (figura 14) muestra el precio medio de las motos usadas según la región o ciudad en la que se ofrecen las motocicletas. Cuanto mayor sea el círculo por región, más motocicletas se ofrecen allí.Se hace evidente que hay regiones donde se venden más motocicletas (Bangalore, Chennai) y que la mitad de las motocicletas que se ofrecen en Bangalore están en el rango de precios superior.

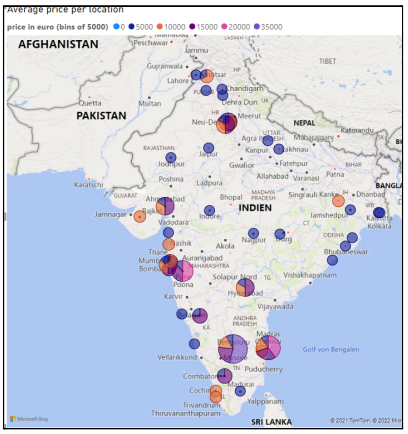

Figura 14. Media del precio por localidad

La figura 15 muestra el número de motocicletas por marca. La mayoría de las motos ofrecidas proceden de las marcas “Bajaj”, “Royal” y “Hero”.

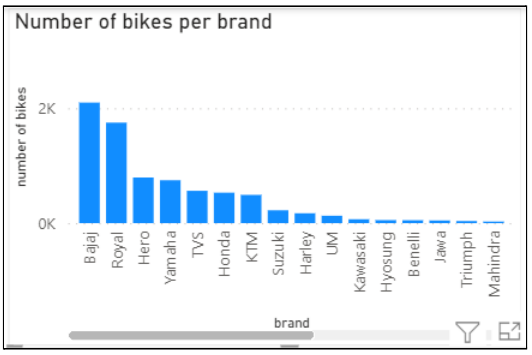

Figura 15. Número de motos por marca

La Figura 16 muestra la relación entre precio y BHP. Para ello, se utilizaron los grupos de datos previamente creados para ambas variables con el fin de crear una imagen más clara. Esta figura

muestra claramente que existe una relación entre el precio y BHP de una motocicleta. Este conocimiento se puede utilizar para examinar la conexión más de cerca (aprendizaje supervisado).

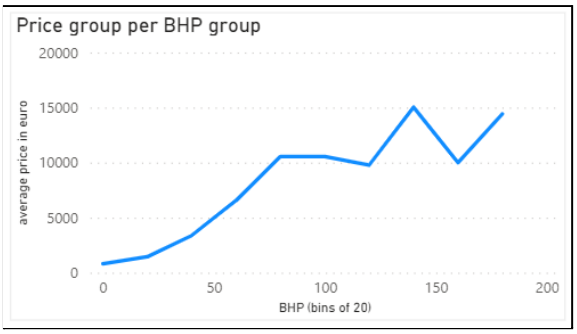

Figura 16. Precio medio por grupo de BHP

La pregunta de si existe una conexión entre el número de propietarios y el precio se responde en la Figura 17. Es evidente que cuanto mayor es el número de propietarios, menor es el precio medio por motocicleta.

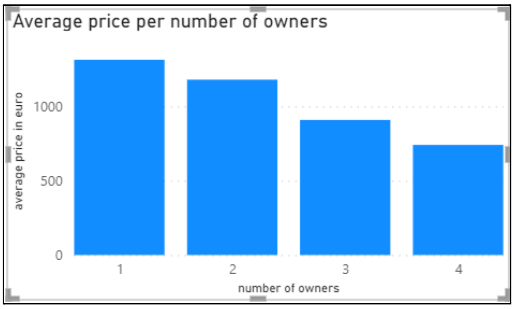

Figura 17. Precio medio por número de propietario

En la Figura 18, los kilómetros recorridos se relacionan con el precio promedio en grupos de datos de 20,000. Se esperaba que hubiera una clara relación entre las dos variables. Cuanto mayor sea el número de kilómetros recorridos, más celoso será el precio. Esta suposición fue refutada por la visualización creada.

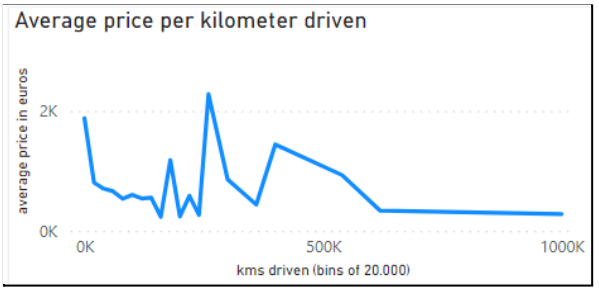

Figura 18. Precio medio por grupos de kilómetros recorridos

5. APRENDIZAJE SUPERVISADO

Gran parte del trabajo realizado en lo que conocemos hoy en día como la ciencia de datos se basa en algoritmos y técnicas de aprendizaje supervisado, el cual es una parte sustancialmente interesante de este proyecto. Para ello necesitaremos una entrada, conformada por el propio dataset sobre el que hemos realizado el preprocesamiento, la visualización y clustering, y tendremos como salida unos modelos predictivos los cuales nos ayudarán a obtener el principal objetivo del proyecto. Mediremos estos resultados con la meta de poder compararlos y obtener el mejor resultado final posible, acorde a lo que podemos esperar de este aprendizaje. Se detalla el objetivo, la implementación y los resultados finales a continuación.

5.1 Objetivo

Como objetivo principal se ha planteado la obtención de un precio estimado para una motocicleta que se quiera poner en venta, teniendo como referencia los datos de ventas con los que se cuenta. Esto podría servir a vendedores y compradores por igual, que teniendo en cuenta el estado de una motocicleta, puedan presuponer un coste aproximado aún sin haber mediado palabra. Se intentará llevar a cabo esta tarea intentando maximizar la exactitud de la predicción. Para esto nos servirá el trabajo realizado en fases anteriores, a lo que sumaremos ciertas técnicas de preparación convenientes para los modelos a implementar y que serán expuestas en el siguiente apartado.

5.2 Implementación

Para la implementación se han usado mayoritariamente el paquete caret para el entrenamiento de los modelos de regresión y tidyverse para la manipulación de datos previa a la creación de los mencionados modelos.

En este proceso se usarán las variables numéricas, que serán suficientes para estudiar el estado de una motocicleta en el dominio del problema que nos encontramos. Es necesario comentar también que las variables que no se incluyen en el entrenamiento se eliminan del dataset para hacerlo menos pesado.

En la implementación se pueden encontrar unos pasos determinados que han sido claves para llegar hasta la meta planteada:

- Primero, como es normal, la lectura de los datos preparados en fases anteriores (formato csv)
- Seguidamente, el proceso de elección de variables predictoras que posteriormente se pasarán a los modelos, mediante un estudio de la correlación e influencia en el precio de la motocicleta.

Para ello se crean gráficos enfrentando cada variable numérica y que nos sirven para ver si hay patrones de comportamiento en el precio según la variación de las diferentes columnas, que son el kilometraje de la motocicleta, la cilindrada de esta, la potencia, el número de propietarios que ha tenido, el consumo de combustible y el año del modelo que se ha vendido.

A continuación se muestra uno de los gráficos que, a modo de resumen, permiten ver estos patrones que se mencionan. Para esta tarea se ha usado el paquete GGally, y en concreto su función ggpairs la cual hace un plot teniendo en cuenta todas las columnas del dataset.

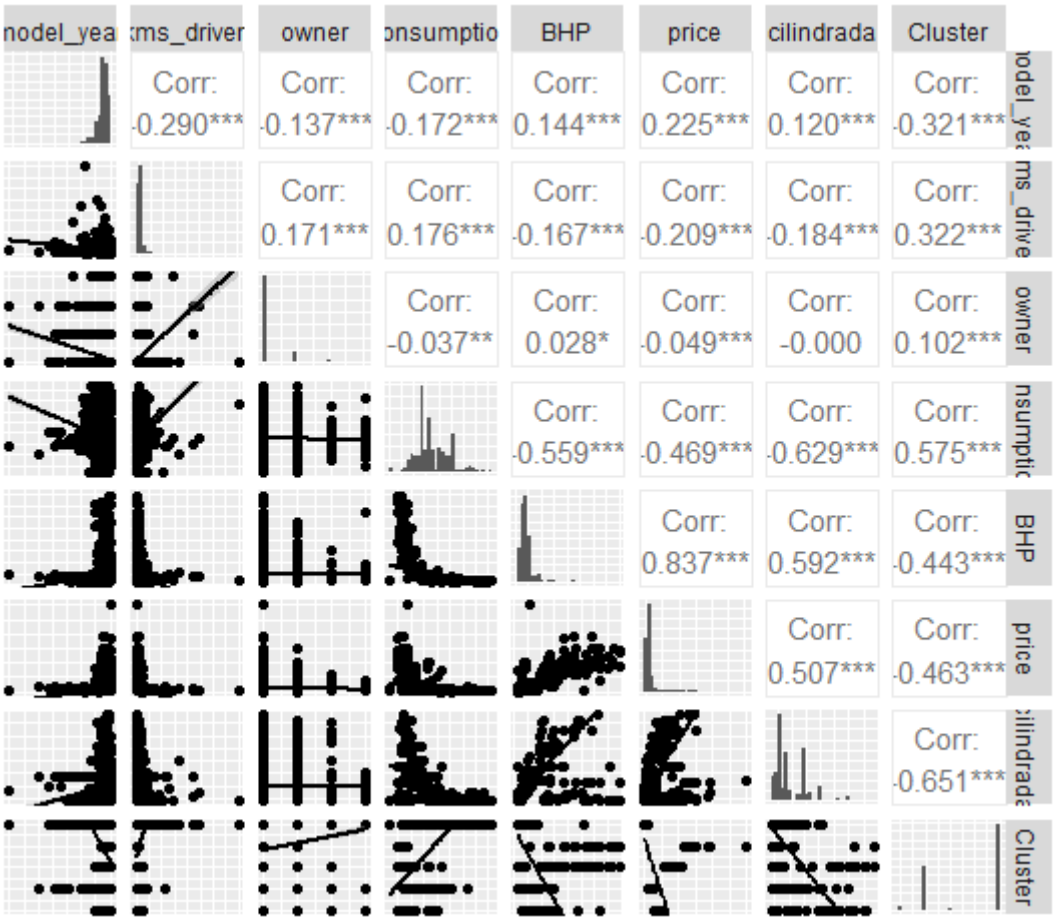

Figura 19. Resumen de correlación entre variables del dataset.

Adicionalmente, se estandariza el conjunto de datos usando la normalización mediante escala del paquete 'dplyr'. Esto es especialmente necesario para el modelo KNN, ya que el cálculo de distancias con columnas que tienen rangos de valores diferentes arrojaría un pobre resultado.

Esto ha provocado que se usen dos particiones diferentes para entrenamiento, una para regresión lineal múltiple y randomForest y otra para knn, siendo normalizado para este último solamente. Los otros dos modelos se ha probado que tienen mejor rendimiento con un dataset sin normalizar.

- Elección de modelos predictores que puedan ser útiles y se adecuen al objetivo que se persigue. Entrenamiento de dichos modelos. Se ejecuta la función train y se obtiene el modelo en sí.

En relación a esta elección, se ha aprovechado el conocimiento adquirido en clase sobre modelos de regresión lineal múltiple, knn y árboles de decisión, los cuales se ajustan a las necesidades del proyecto. Todos ellos se han implementado usando la función train del método caret.

Uno de los parámetros de la función train es el train control. En este caso se ha usado en los tres modelos la técnica 'repeated cross validation'. Para la regresión lineal múltiple se ha configurado finalmente que se hagan 10 cruces diferentes y se repita el proceso 5 veces, lo que hemos comprobado que mejora el rendimiento respecto a otros métodos.

De igual manera se hace para el modelo de los k-vecinos más cercanos, con 10 cruces y 5 repeticiones. Para el KNN también se configura el valor tuneGrid, mediante el cual se configura para que pruebe con los números k del 1 al 20. El mejor resulta ser el k=2.

Como detalle, se debe comentar que la proporción conjunto de training/testing que se ha hecho ha sido de un 75% / 25%.

- Validación con conjunto de testing y comparación de medidas de error entre los diferentes modelos y sus resultados.

5.3 Resultados y conclusiones

Primero de todo, la medida usada para ver qué tan buenos son los modelos es Rsquared, conocida como 'r cuadrado', que es un valor entre 0 y 1 que denota el porcentaje de acierto obtiene tras la validación con el conjunto de testing. Se ha usado esta medida porque es globalmente conocida y representa suficientemente bien si logramos el objetivo o no.

Según esta métrica mencionada, el resultado arroja que el mejor modelo resulta ser el de randomForest, siendo igualmente el modelo que más tarda en entrenarse, con un tiempo considerablemente mayor al de RLM o KNN. El r2 de RandomForest ha sido de aproximadamente 0.93. Con un menor rendimiento tenemos KNN, el cual aproximadamente da un valor de rsquared de 0.83. Por último, regresión lineal resulta ser el menos bueno de los modelos pero el de mayor velocidad de entrenamiento, siendo su métrica un 0.79. A continuación se exponen las medidas en ejecución de los modelos.

```
> MLR_MODEL <- multiple_regresion_prediction(training)
> MLR_PRED <- predict(MLR_MODEL, newdata = testing)
> #Mostrar medidas obtenidas
> evaluacion_lm <- postResample(pred = MLR_PRED, obs = testing$price)
> evaluacion_lm
        RMSE     Rsquared          MAE
64074.913279     0.794634 34644.371077
>
```

Figura 20. Rsquared de RLM.

La correspondiente a KNN, el segundo modelo más rápido en entrenarse y el segundo también en rendimiento.

```
> #MODELO KNN
> KNN_MODEL <- knn_prediction(training_knn)
> KNN_PRED <- predict(KNN_MODEL, newdata = testing_knn)
> evaluacion_knn <- postResample(pred = KNN_PRED, obs = testing_knn$price)
> evaluacion_knn
     RMSE  Rsquared       MAE
0.3857742 0.8353857 0.1314820
```

Figura 21. Rsquared de KNN.

Por último, el mejor modelo que sería el random forest, y su métrica.

```
> RF_MODEL <- random_forest_prediction(training)
> RF_PRED <- predict(RF_MODEL, newdata = testing)
> evaluacion_rf <- postResample(pred = RF_PRED, obs = testing$price)
> evaluacion_rf
        RMSE     Rsquared          MAE
3.508699e+04 9.292948e-01 1.488645e+04
```

Figura 22. Rsquared de RF.

6. DETECCIÓN DE ANOMALÍAS

6.1 Definiciones y aplicación de la detección de anomalías

Una anomalía es un valor atípico que está numéricamente distante del resto de los datos. Las anomalías pueden distorsionar las evaluaciones de datos, por lo que es importante identificarlas y tratarlas en consecuencia. La detección de anomalías es una forma de detectar instancias inusuales en un conjunto de datos.

La detección de anomalías en conjuntos de datos tiene muchos usos prácticos diferentes. En primer lugar, se puede utilizar el procedimiento para detectar comportamientos maliciosos. Esto es importante en la banca, por ejemplo, donde se puede contrarrestar el fraude con tarjetas de crédito. Además, detectando y analizando anomalías se pueden dar alertas a las técnicas de servicio. Esto puede ayudar a habilitar el mantenimiento predictivo de, por ejemplo, máquinas. Dado que las anomalías pueden distorsionar las evaluaciones de datos, la detección y el tratamiento de anomalías también se pueden utilizar para un aprendizaje supervisado “más limpio”. Otro caso de uso sería la evaluación de la competencia de un modelo.

6.2 Detección de anomalías en BigML

BigML ofrece una amplia variedad de recursos básicos de aprendizaje automático que se pueden combinar para resolver tareas complejas de aprendizaje automático. La interfaz BigML permite ver y interactuar fácilmente con las anomalías detectadas en un conjunto de datos. Un análisis de anomalías en BigML consta de dos pasos o dos opciones.

Después de cargar el conjunto de datos deseado en la aplicación, se puede crear un detector de anomalías con solo unos pocos clics. La visualización resultante muestra las 10 anomalías principales en el conjunto de datos. Se calcula una "anomaly score" en porcentaje para cada anomalía mostrada. Esto indica cuán anómala es esa instancia en relación con otras instancias (figura 19). El usuario puede seleccionar las 10 anomalías o un número específico y luego crear un nuevo conjunto de datos. Este conjunto de datos puede contener todos los datos excluyendo las anomalías seleccionadas o todas las anomalías seleccionadas. El detector de anomalías entrenado no hace predicciones en el sentido supervisado. No puede predecir un valor para un campo en el conjunto de datos.

Figura 23. Top 5 anomalías con el "anomaly score"(naranja) en BigML

Sin embargo, sí hay un nuevo conjunto de datos con las mismas columnas, puede obtener las puntuaciones de anomalías de todos los puntos en el conjunto de datos utilizando el "batch anomaly score function". Después es posible crear un nuevo conjunto de datos con una nueva columna que tiene una puntuación de anomalía para cada instancia (figura 20).

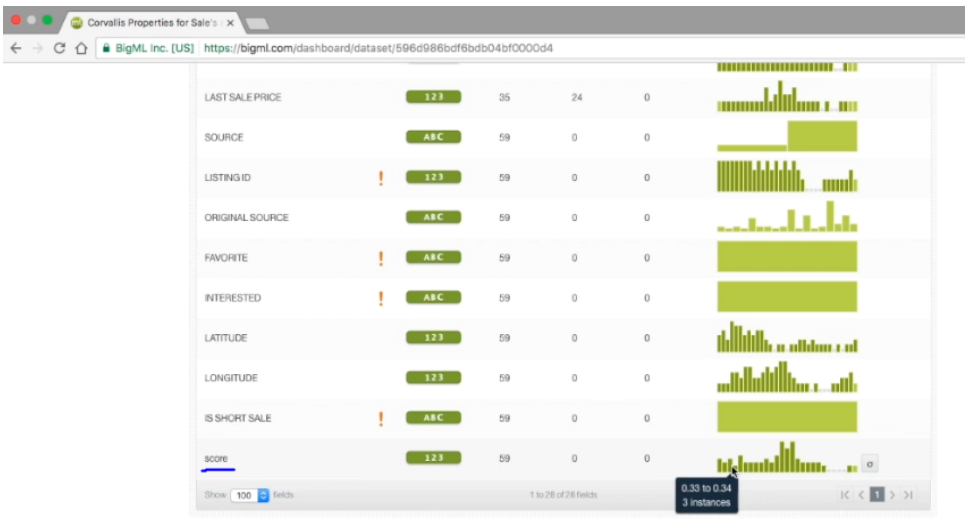

Figura 24. Nuevo conjunto de datos con la columna "score"

7. CONCLUSIONES

Teniendo en cuenta el objetivo planteado inicialmente de construir un modelo de predicción para la puesta en valor de nuevos vehículos basándonos en características y valores similares ya publicados en la web *droom.in*, podemos concluir con la finalización exitosa del trabajo.

Durante el desarrollo del presente proyecto, hemos observado la gran relevancia de la etapa de preprocesamiento de datos de cara a los resultados futuros, pues el tiempo dedicado a esta primera fase asegura, sin lugar a dudas, la calidad de esos resultados, así como también facilita y optimiza, por otra parte, la ejecución de las etapas posteriores. Por otro lado, también destacamos la utilidad que obtuvimos del clustering al usarlo como técnica para mejorar la calidad de los datos. Ya que, el uso habitual de estos métodos está más relacionado con la búsqueda de un resultado concreto, y nosotros hemos podido aprovecharlo para afinar nuestro preprocesamiento y así, llegar a un resultado satisfactorio.

En cuanto a propuestas de mejora o posibles proyectos de futuro en esta misma línea de trabajo, proponemos la comparación del lenguaje de programación R con otros lenguajes de programación con el objetivo de valorar el método más eficiente para el desarrollo de este modelo. Un apartado que sin duda nos gustaría tratar en el futuro es el de *detección de anomalías*, ya que como se detalla en el documento, existen técnicas muy útiles con las que se podrían llegar a mejorar la calidad de los datos.

8. BIBLIOGRAFÍA

AMAL NAIR (2019): Beginner's Guide To K-Means Clustering.
https://analyticsindiamag.com/beginners-guide-to-k-means-clustering/

APRENDIZAJE SUPERVISADO:

https://bookdown.org/dparedesi/data-science-con-r/aprendizaje-supervisado.html

BIGML (2017): Anomaly Detection.

https://www.youtube.com/watch?v=a5Q7b4e7lqg&list=PL1bKyu9GtNYHAk0PUojkLYZzASoYVcsTQ

BIGML (2021): General Information

https://bigml.com/education/videos

DORIAN PYLE (1999): Data Preparation for Data Mining Morgan Kaufmann Publishers.

DAGNINO, J. (2014). Datos faltantes (missing values). Rev Chil Anest, 43, 332-334.

JONATHAN RAMIREZ (2018): K-means: Elbow Method and Silhouette.
https://medium.com/@jonathanrmzg/k-means-elbow-method-and-silhouette-e565d7ab87aa

JOAQUIM SCHORK:

https://statisticsglobe.com/standardize-data-frame-columns-in-r-scale-function

KAGGLE (2021): Used Bike Prices In India. Used motorcycle pricing in Indian market dataset.
https://www.kaggle.com/ropali/used-bike-price-in-india/metadata

LUIZ FONSECA (2019): Clustering Analysis in R using K-means.
https://towardsdatascience.com/clustering-analysis-in-r-using-k-means-73eca4fb7967

OPEN DATA SCIENCE (2018): Unsupervised Learning: Evaluating Clusters.
https://odsc.medium.com/unsupervised-learning-evaluating-clusters-bd47eed175ce

POWERBI (2021): Qué es PowerBI https://powerbi.microsoft.com/es-es/what-is-power-bi/

RANDOM FOREST:

https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting

REGRESION LINEAL MÚLTIPLE:

https://www.cienciadedatos.net/documentos/25_regresion_lineal_multiple.html

RPUBS:

https://www.rpubs.com

THE CARET PACKAGE:

https://topepo.github.io/caret/model-training-and-tuning.html#model-training-and-parameter-tuning