BMW Price Prediction Challenge

Franco Cibils

Contents

1 Introducción

La capacidad de determinar el precio de un auto en base a sus características es un desafío constante en la industria automotriz, en particular en la reventa. Tanto compradores como vendedores buscan obtener el mejor precio posible por el auto que están comprando o vendiendo; y existen múltiples factores que pueden impactar en el precio de un vehículo, sin embargo, algunos son más importantes que otros.

Por este motivo resulta crucial tener una noción no solamente de qué factores determinan el precio de un auto, sino sobre cuál es efectivamente el precio adecuado de este a partir de sus características particulares (motor, distancia recorrida, tipo de combustible, etc). El objetivo de este trabajo es tomar una marca en particular, BMW, y estudiar las distintas variables que potencialmente explican el valor de un auto e implementar distintos algoritmos que utilicen dichas variables para predecir lo más fielmente posible el precio de los distintos modelos.

A tal fin, se partirá de un modelo de regresión lineal como *benchmark* que funcionará como punto de partida para la posterior comparación con modelos más complejos como *bagging* y *gradient boosting machine*, entre otros.

2 Revisión bibliográfica

Dada la amplia disponibilidad de datos respecto a este tema, la bibliografía es extensa respecto a esta clase de problemas. Una técnica que suele aparecer bastante en la literatura revisada es la de *random forest*, aunque también se ha visto el uso de otros algoritmos como *SVM* y redes neuronales. Algunos de los trabajos revisados (no necesariamente académicos) son los siguientes: *Car's Selling Price Prediction using Random Forest Machine Learning Algorithm (Pandey, Rastogi, Singh)*, *How much is my car worth? A methodology for predicting used car prices using Random Forest (Pal, Arora, Palakurthy, Sundararaman, Kohli)*, *Car Price Prediction using Machine Learning Techniques (Gegic, Isakovic, Keco, Masetic, Kevric)* y *Predicting the Price of Used Cars using Machine Learning Techniques (Pudaruth)*.

3 Base de datos

Se trabaja con una base de datos obtenida en *Kaggle*[*] para autos de BMW comercializados en el 2018. El objetivo, señalado como un *challenge* en Kaggle, consiste en estimar el precio de un auto, en este caso particular el precio de un modelo cualquiera de BMW, a partir de sus características y atributos.

Este dataset consiste, inicialmente, en 18 variables y 4843 observaciones. Las variables (junto con el tipo de variable) son las siguientes:

- **price** (integer): Precio del modelo de auto.
- **maker_key** (character): Fabricante del auto (dado que en este caso solo tratamos con autos de la marca BMW esta variable resulta completamente inútil).
- **model_key** (character): Modelo del auto. Existen 75 modelos de autos diferentes.
- **mileage** (integer): Cantidad de millas recorridas.
- **engine_power** (integer): Potencia del motor (medida en caballos de fuerza).
- **registration_date** (character): Fecha de fabricación del auto.
- **fuel** (character): Tipo de combustible usado por el auto. Consiste en 4 subclases: *diesel, petrol, electric, hybrid petrol.*
- **paint_color** (character): Color del modelo. Existen 10 colores diferentes.
- **car_type** (character): Tipo de auto. Existen 8 subcategorías: *convertible, coupe, estate, hatchback, sedan, subcompact, SUV, van.*
- **sold_at** (character): Fecha de venta del vehículo.
- **feature_1 - feature_8** (boolean): 8 características del equipamiento del auto. No se aclara a qué característica corresponde cada variable, pero los autores del *dataset* señalan que son importantes para determinar el precio de un auto.

En la base de datos no hay *missing values* para ninguna observación.

[*] https://www.kaggle.com/danielkyrka/bmw-pricing-challenge

4 Initial data preprocessing

Antes de llevar a cabo un análisis exploratorio de los datos para familirizarse mejor con este *dataset*, se lleva a cabo una cuidadosa inspección de los datos y se implementan algunas consideraciones:

- En primer lugar, como se mencionó en el apartado anterior, la variable **maker_key** resulta irrelevante puesto que todos los autos son BMW, en otras palabras, esta variable no tiene variabilidad alguna por lo que se procede a descartarla.

- Una observación tiene un valor negativo para la variable **mileage**. Lo más probable es que sea un error de entrada de datos, por lo que se toma el valor positivo que resulta razonable.

- Existen dos autos con precios absurdos para la clase de auto que son (hasta 10 veces más caros que el promedio para ese modelo). Nuevamente, se considera que estos valores se deben a un error de entrada de datos, por los que se los cambia manualmente. El criterio fue observar el precio promedio del modelo del auto y evaluar donde pudo estar el error de imputación. De haber ocurrido más veces, se hubiese procedido con algún método de correción de valores automático. Cabe aclarar que un análisis similar se podría haber hecho para los demás modelos de autos observando el *boxplot* del precio, sin embargo, en dicha figura (que no se muestra en este trabajo), si bien se observan outliers, estos no están los suficientemente alejados como para considerar un error de imputación como es el caso de los dos autos anteriores.

- Para llevar a cabo el EDA se transforman las variables correspondientes en variables categóricas.

- Los autores del dataset señalan la posibilidad de que el precio de un auto se vea influenciado por la época del año en la que se vende. Por este motivo se crea una nueva variable **sold_at_season** para distinguir en qué estación del año (verano, invierno, etc.) se vendió el auto.

- Finalmente, cuán viejo un auto es brinda información sumamente importante para determinar el precio del mismo. Es por que ello que se crea una nueva variable que refleje la antigüedad del auto, **car_age**.

5 Exploratory Data Analysis

En primer lugar, se evalúa la proporción de cada categoría dentro de cada variable. Comenzamos por variables categóricas por su naturaleza.

- **fuel_type**: Consiste en las 4 categorías mencionadas anterioremente (*diesel, electric, hybrid-petrol, petrol*), de las cuales *diesel* representa casi el 96% de los casos; *petrol* representa casi el 4%; y el resto está distribuido en las demás categorías.

- **car_type**: Las principales categorías son *estate* (33%), *sedan* (24%), *SUV* (21%) y *hatchback* (14%). El resto está comprendido por las categorías *convertible, coupe, subcompact* y *van*.

- **paint_color**: Los colores más usuales son el negro, blanco, azul y gris. Mientras que los menos usuales son el beige, marrón, verde, naranja, rojo y plateado.

Luego, se transforman tres variables continuas (antigüedad del auto, precio y millas recorridas) en categóricas siguiendo un criterio arbitrario pero razonable.

- **car_age**: Se clasifica en 4 categorías: menor a 2 años (2%), 2 a 5 años (65%), 5 a 10 años (27%) y mayor a 10 años (6%).

- **mileage**: Se clasifica en 5 categorías: menos de 10k millas recorridas (0.2%), de 10k a 50k (5%), de 50k a 100k (17%), de 100k a 150k (32%) y más de 150k (43%).

- **price**: Se clasifica en 4 categorías: menos de 10k (20%), 10k a 20k (60%), 20k a 30k (13%) y más de 30k (7%).

A continuación, evalúo la correlación entre las 4 variables continuas disponibles (**price, mileage, engine_power y car_age**.

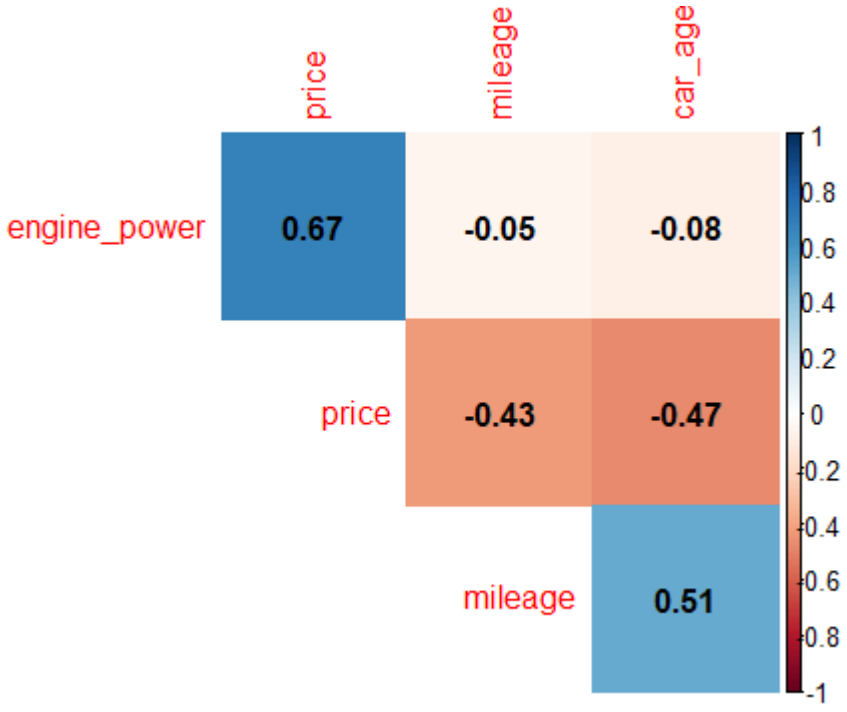

Figure 1: Correlation plot between continuous variables

En la figura anterior se puede observar, como es de esperar, que existe una correlación positiva y relativamente fuerte en el precio de un auto y la potencia del motor, como también entre la antigüedad del auto y la cantidad de millas recorridas. Por otra parte, existe una correlación negativa moderada, nuevamente uno habría de esperar este resultado, entre el precio de un auto y la cantidad de millas recorridas, al igual que con la antigüedad del mismo. Finalmente, no hay motivos para creer que la potencia del motor debiera tener alguna correlación con la cantidad de millas recorridas y la antigüedad del vehículo, lo cual se confirma en los muy leves valores negativos para la correlación entre estas variables.

Se explora a continuación la distribución de las variables continuas a través de histogramas, como también la frecuencia de las subcategorías en las variables categóricas (a excepción de **model_key** debido a la enorme cantidad de clases).

Histogramas

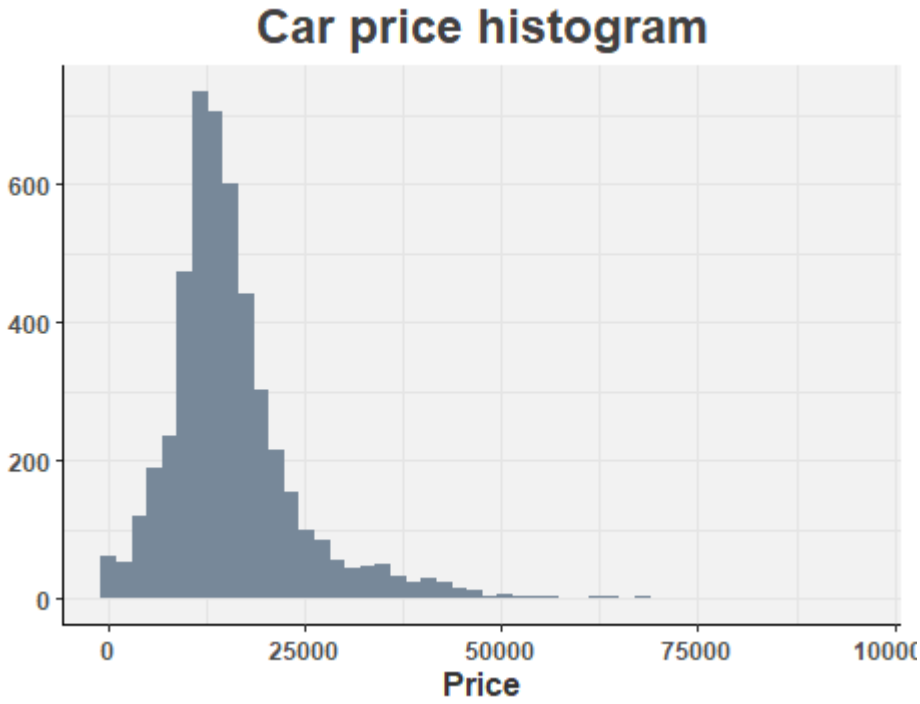

Figure 2: Car price histogram

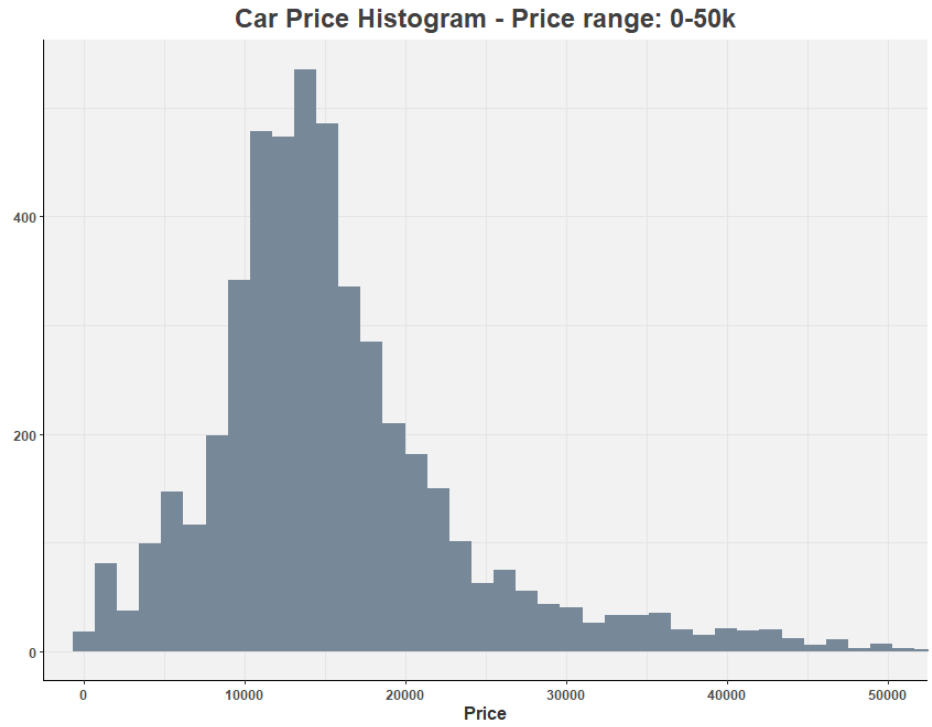

Figure 3: Car price histogram (0-50k range)

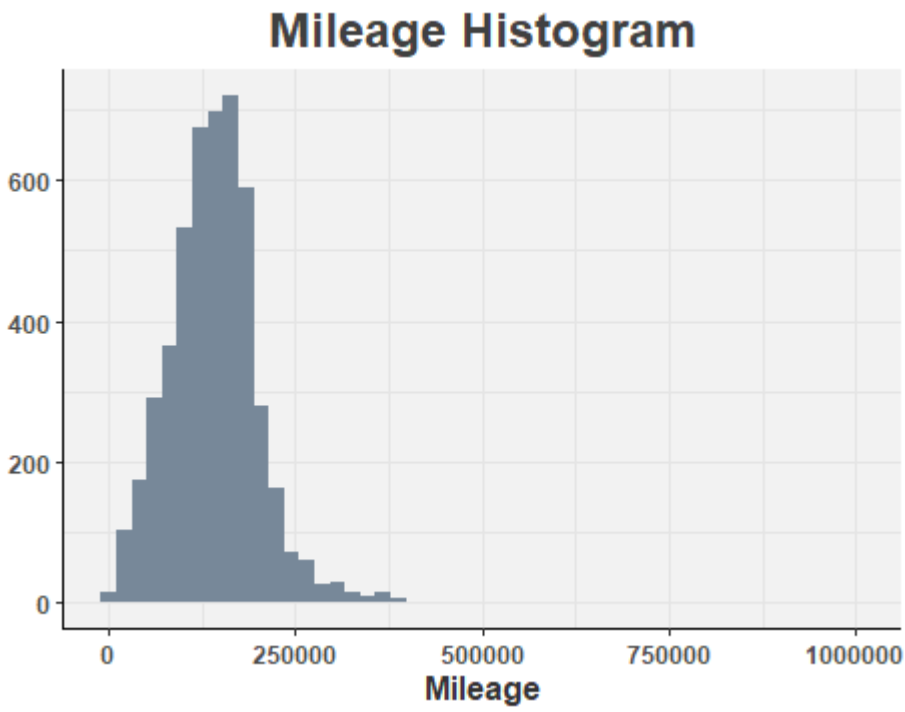

Figure 4: Mileage histogram

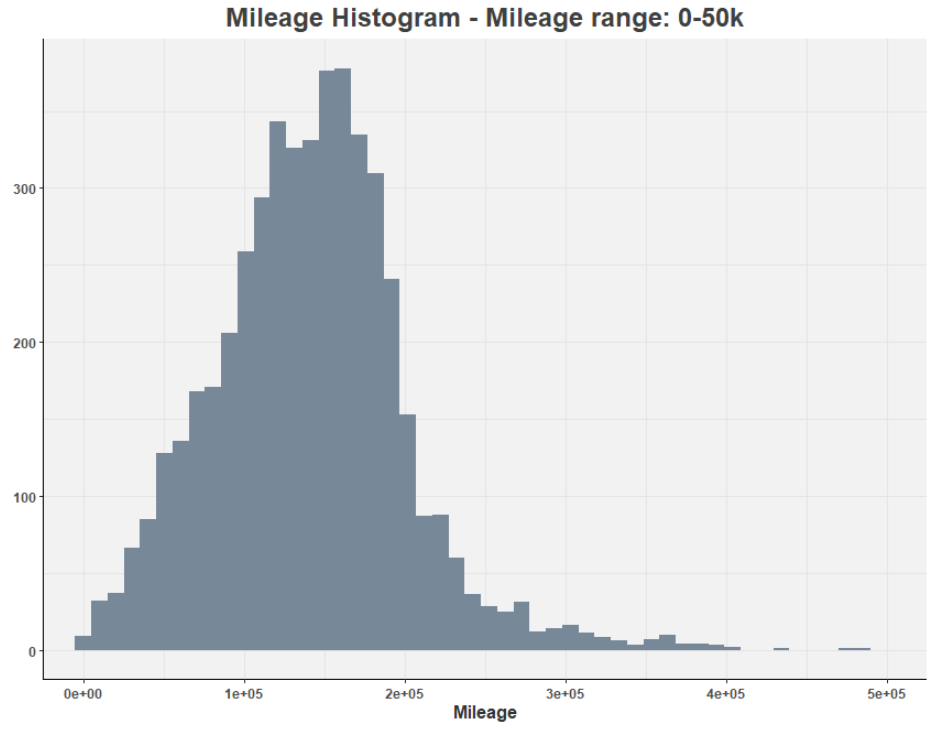

Figure 5: Mileage histogram (0-500k range)

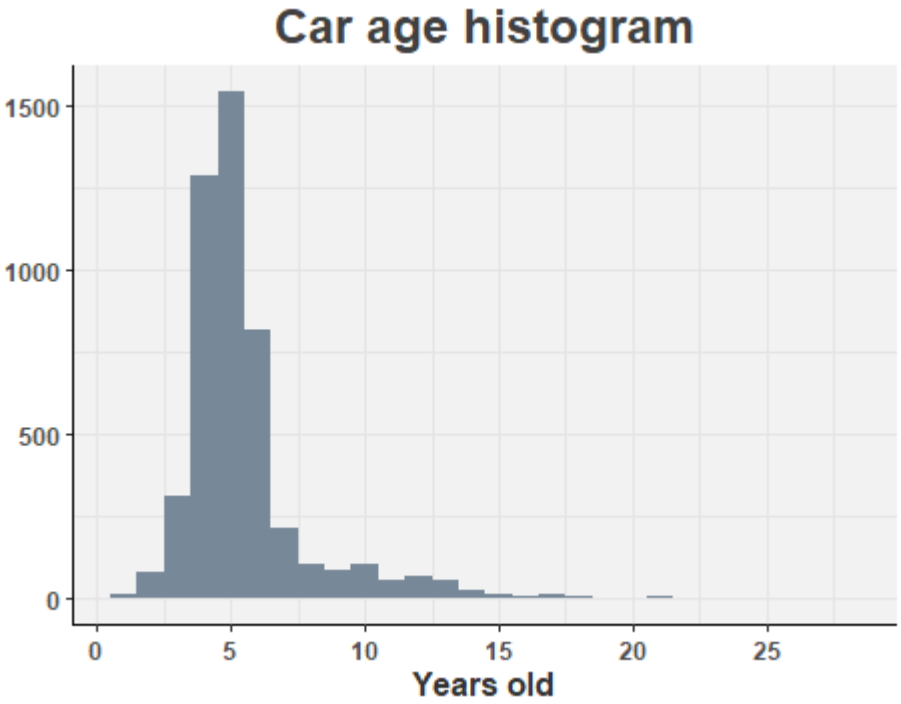

Figure 6: Car age histogram

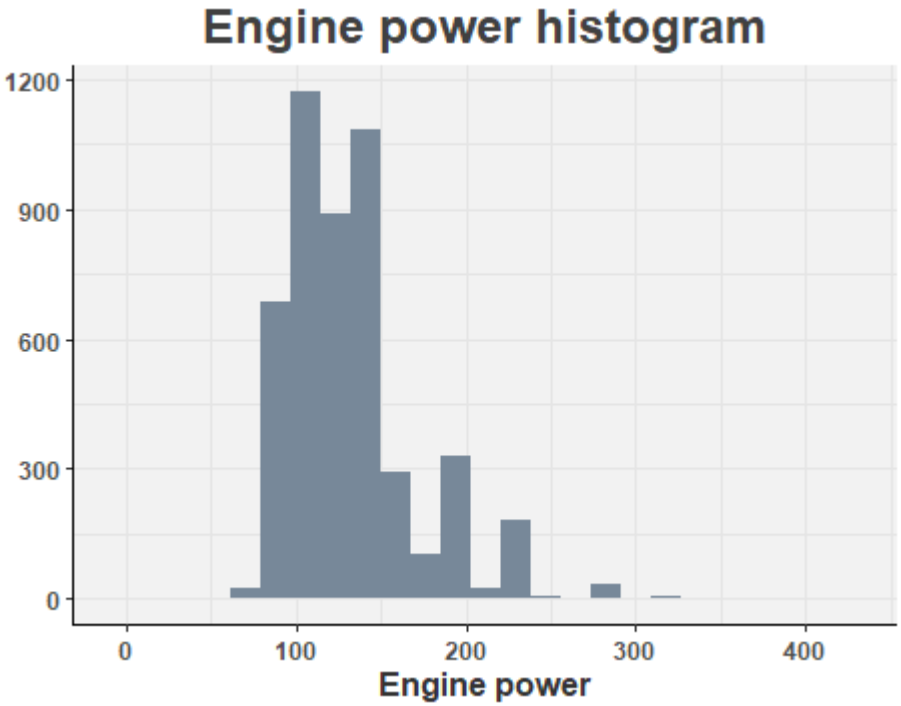

Figure 7: Engine power histogram

Figura 2 y 3: Se observa que a grandes rasgos el histograma del precio presenta una distribución

normal. En la figura 3 se muestra el mismo histograma para valores menores a 50000, que representa más del 90% de las observaciones.

Figura 4 y 5: Se observa que la mayor cantidad de observaciones se ubica por debajo de las 300000 millas recorridas. En la figura 5 se observa el histograma para valores menores a 500000 millas recorridas. Pareciera que la distribución se asemeja a una normal.

Figura 6: La mayor cantidad de valores están concentrados en tornos a los 5 años de antigüedad. Los vehículos menores a los 10 años de antigüedad concentran cerca del 94% de las observaciones.

Figura 7: La mayor parte de los vehículos tienen una potencia del motor menor a los 200 caballos de fuerza, concentradas la mayoría de las observaciones en el intervalo de 100 a 200 caballos de fuerza.

Barcharts

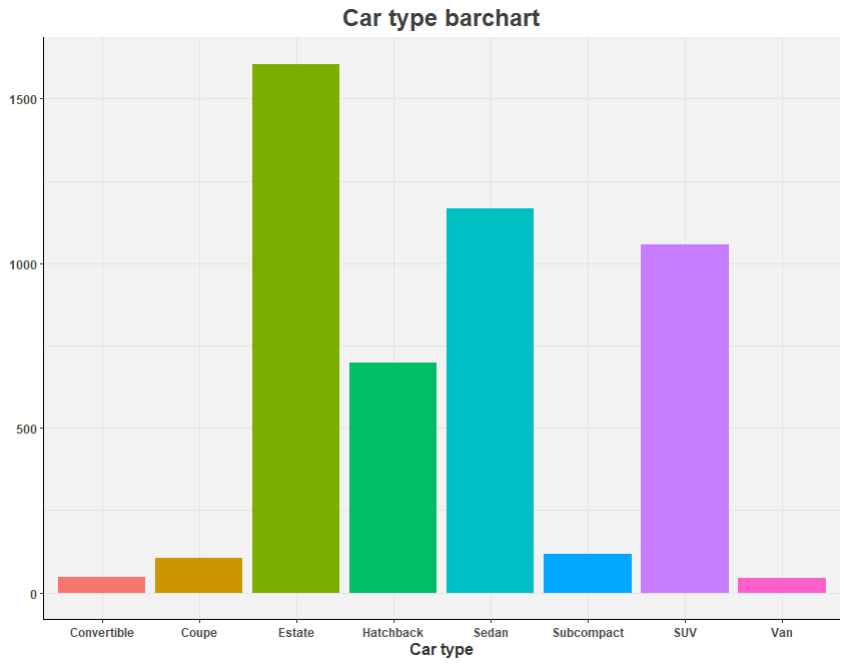

Figure 8: Car type barchart

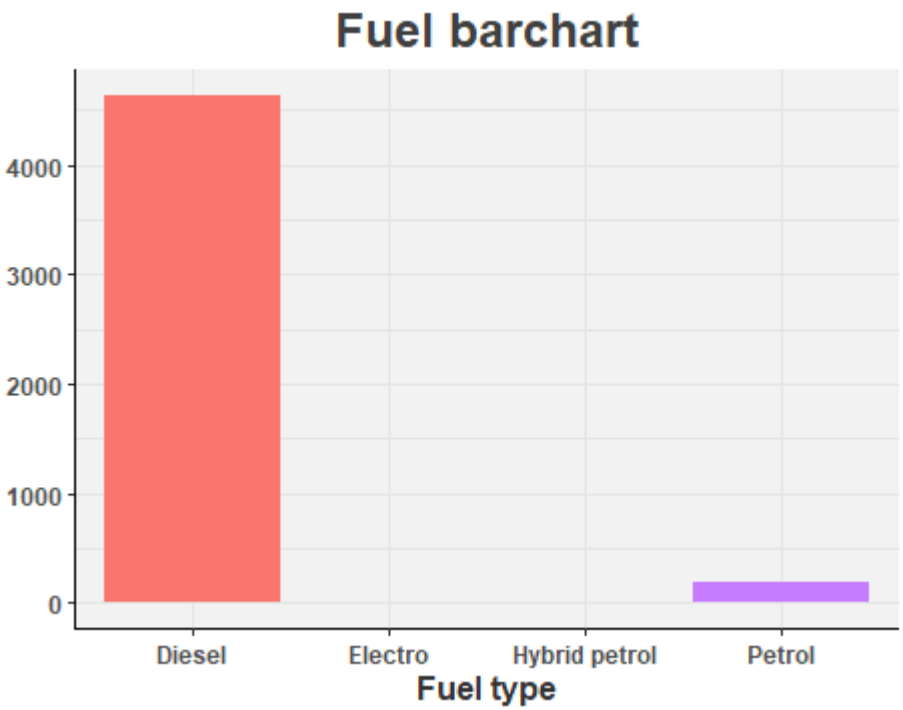

Figure 9: Fuel type barchart

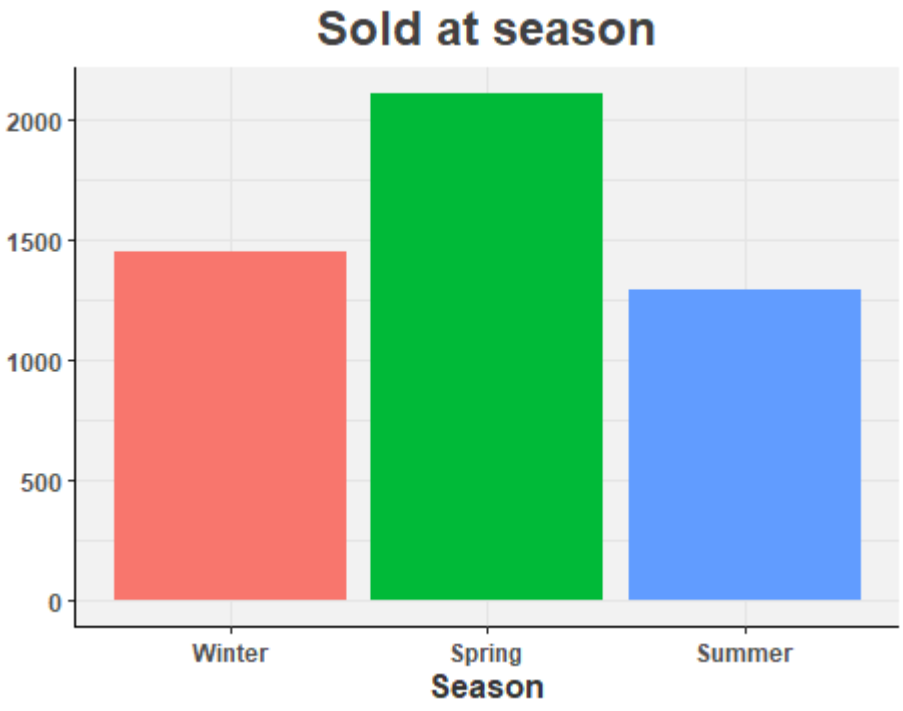

Figure 10: Sold-at season barchart

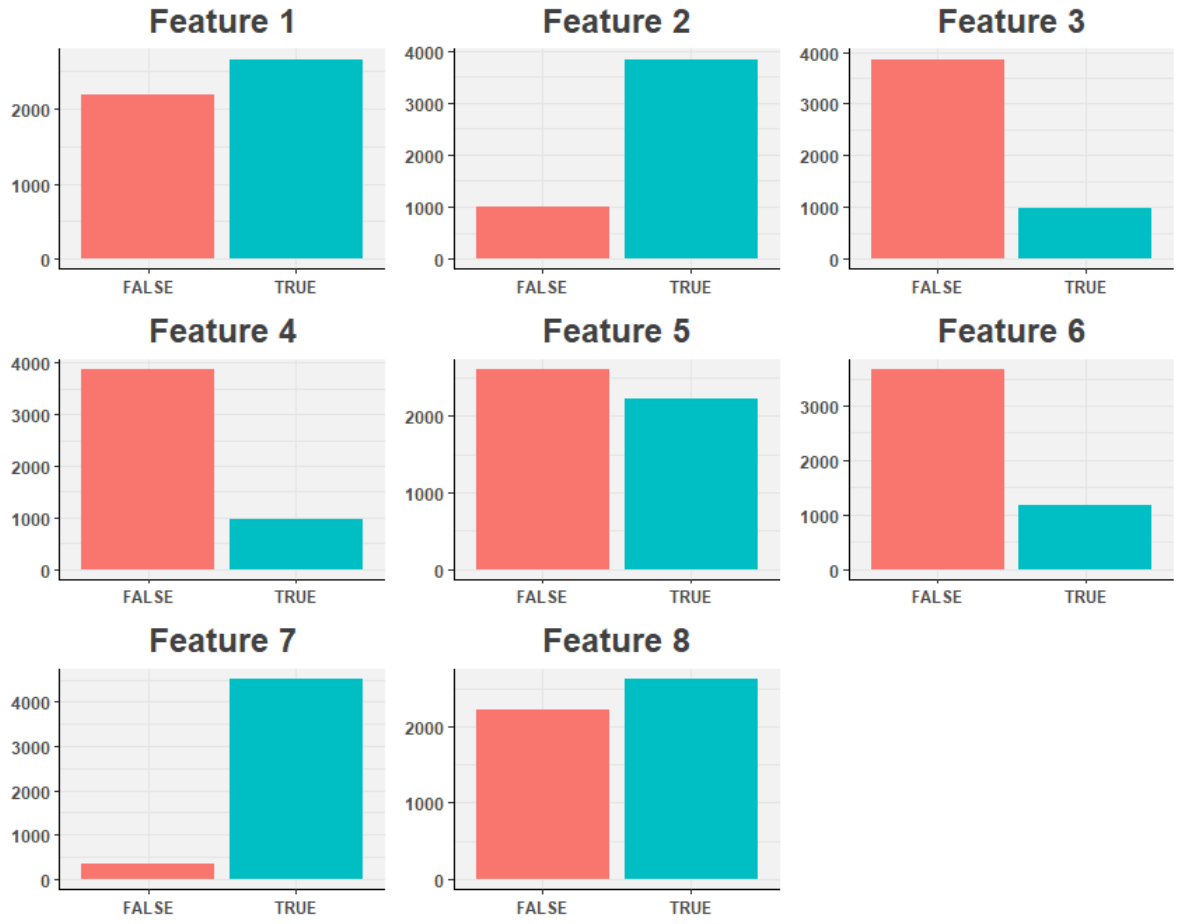

Figure 11: Features barchart

Figura 8: Se observa que la mayor proporción de autos corresponden a las categorías de *estate, hatchback, sedan* y *SUV*. El resto de las observaciones están divididas entre *convertible, coupe, subcompact* y *van*.

Figura 9: Como se había mencionado anteriormente casi todos los vehículos utilizan *diesel* como fuente de combustible, seguido por *petrol* y, en menor medida (tan solo unas pocas observaciones), *electro* y *hybrid-petrol*.

Figura 10: Se observa que primavera fue la época en la que más autos se vendieron, seguido por el invierno, y luego verano. No hay autos que se hayan vendido en otoño (probablemente esto se deba a la fecha de creación de la base de datos).

Figura 11: Las *features 1, 5 y 8* parecen bastante equilibradas, mientras que las *features 2, 3, 4, 6 y 7* no. Sin más información sobre estas variables no se puede comentar mucho más.

A continuación se exploran la relación que puedan existir entre dos o tres variables a través de

distintas figuras.

Boxplot

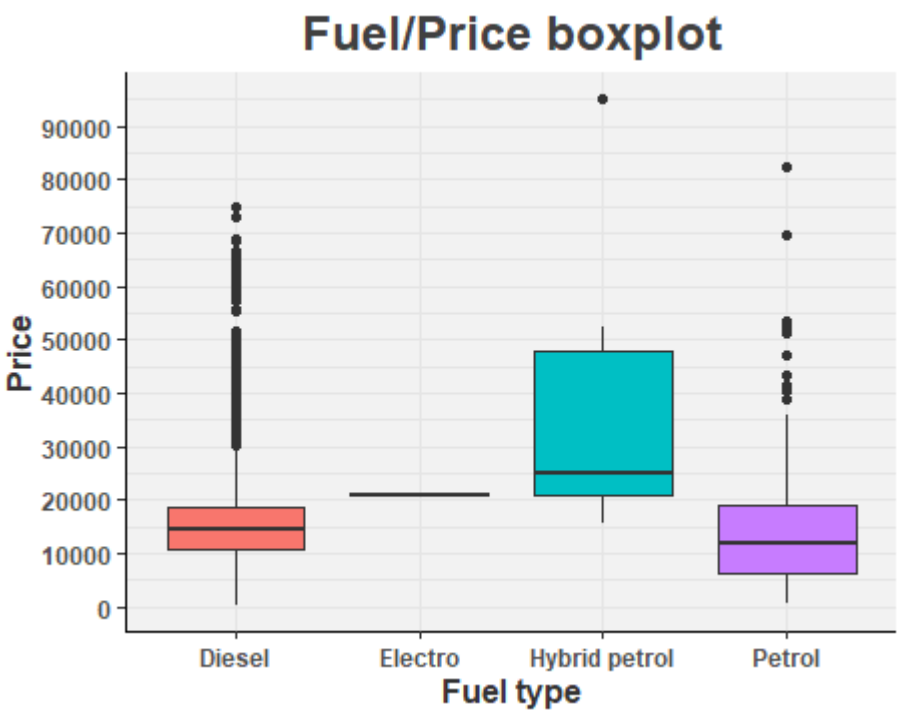

Figure 12: Fuel/Price boxplot

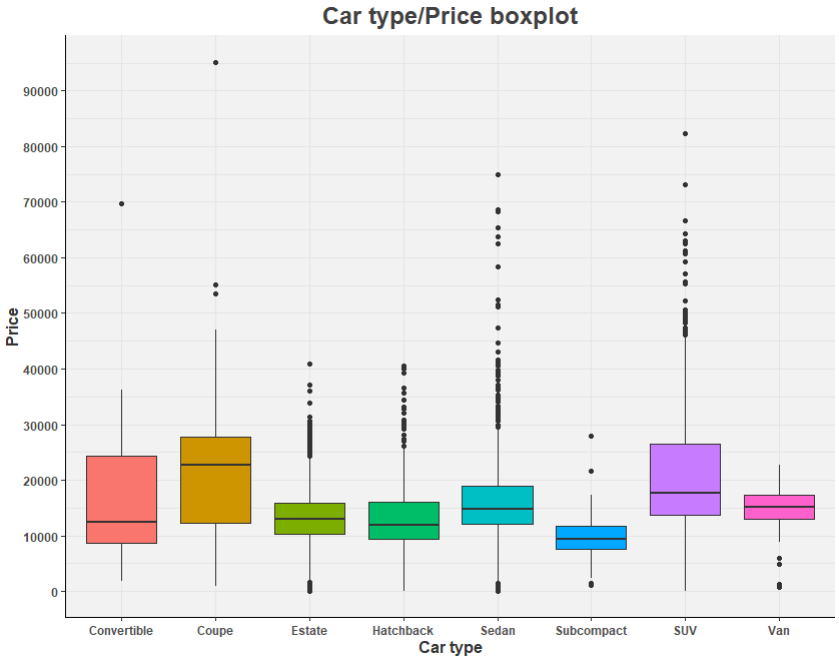

Figure 13: Car Type/Price boxplot

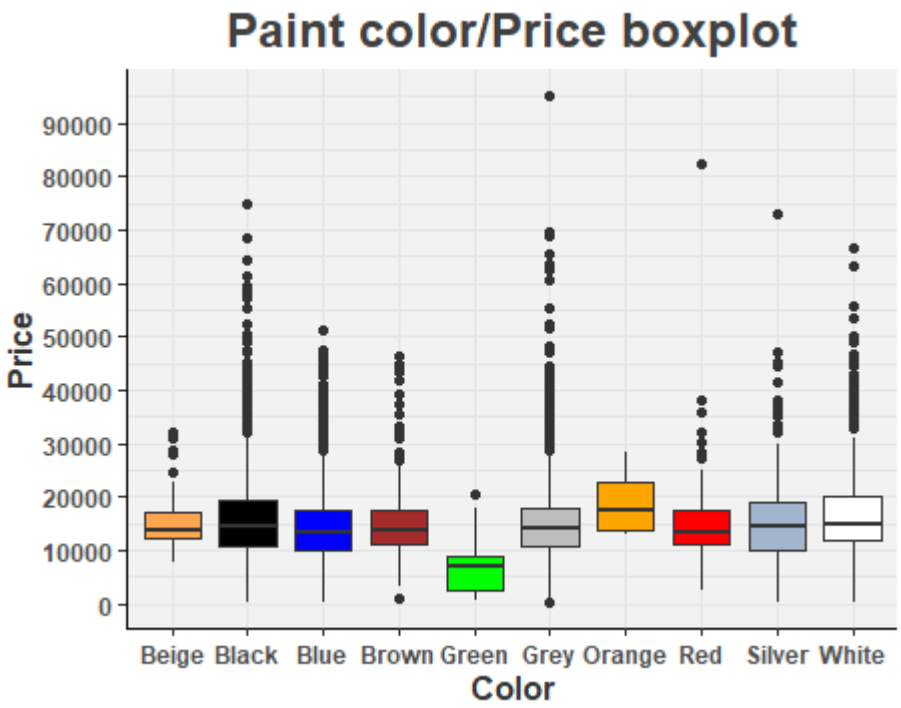

Figure 14: Color/Price boxplot

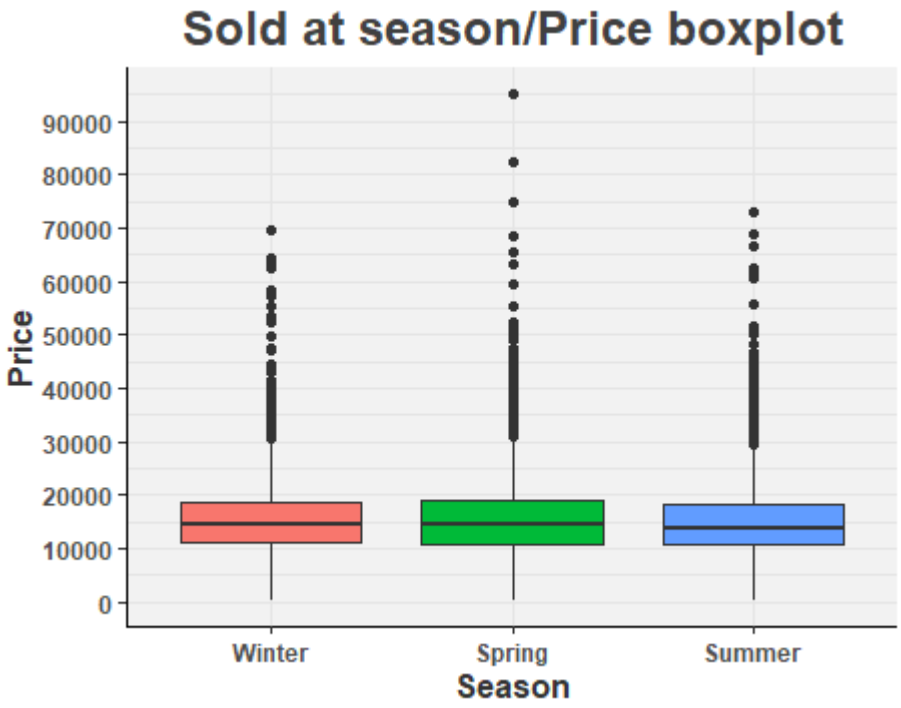

Figure 15: Sold-at Season/Price boxplot

Figura 12: Se puede notar rápidamente que la categoría *diesel* tiene una media de precio

relativamente baja. También se observa que para esta categoría existen muchos *outliers*. Las categorías *electro y hybrid petrol* tienen una media más alta relativo a los otros dos tipos de combustible. Y respecto a la categoría *petrol*, esta tiene la media más baja, con una cantidad moderada de *outliers*.

Figura 13: Las categorías *convertible, estate, hatchback, subcompact y sedan* exhiben medias similares con gran cantidad de *outliers*. Por otra parte, *coupe, SUV y van* exhiben medias más altas, también con bastantes *outliers*, en particular *SUV*.

Figura 14: Se puede observar a grandes rasgos que todos los colores tienen una gran cantidad de *outliers*. Se puede notar que el verde y el naranja destacan por tener la media más baja y más alta, respectivamente. Por otra parte, los demás colores tienen medias similares.

Figura 15: La media del precio es indistinta según la estación en la que se venda el auto. No parece confirmarse lo que los autores del *dataset* especulaban, es decir, que un auto podía venderse más caro simplemente porque era verano. Se observa una gran cantidad de *outliers* para todas las categorías.

Scatterplot

Figura 16: Como se había anticipado en la figura de correlación entre las variables continuas, existe una relación negativa entre el precio y la cantidad de millas recorridas. A su vez, se incorpora la antigüedad del auto como tercera variable en la figura donde un color más oscuro corresponde a mayor antigüedad y uno más claro a menor antigüedad. Nuevamente, se confirma que a mayor antigüedad, se observa una mayor cantidad de millas recorridas como también un menor precio. Esto se observa fácilmente por el hecho de que los colores se van oscureciendo a medida que nos desplazamos hacia abajo a la derecha.

Figura 17: En esta figura se toma como tercer variable el tipo de combustible utilizado por el vehículo. Se confirma la amplia mayoría de autos que utilizan *diesel*. No pareciera haber ninguna otra relación, más alla de que a mayor cantidad de millas recorridas, menor es el precio.

Figura 18: Al igual que en las dos figuras anteriores, se trabaja con el precio y la cantidad de millas recorridas en los ejes, considerando como tercer variable el tipo de vehículo. Nuevamente, no parece surgir ninguna otra relación más allá de la ya mencionada entre precio y millas recorridas.

Figura 19: Se confirma nuevamente la relación negativa entre el precio y la antigüedad del

vehículo que pareciera ser exponencial en lugar de lineal. Como tercer variable se considera el tipo de vehículo pero no parece surgir ninguna relación relevante.

Figura 20: En esta figura se confirma que a mayor potencia del motor mayor es el precio, aunque parece haber bastante dispersión en el precio para cierta potencia determinada. No parece surgir ninguna relación cuando se considera como tercer variable el tipo de vehículo.

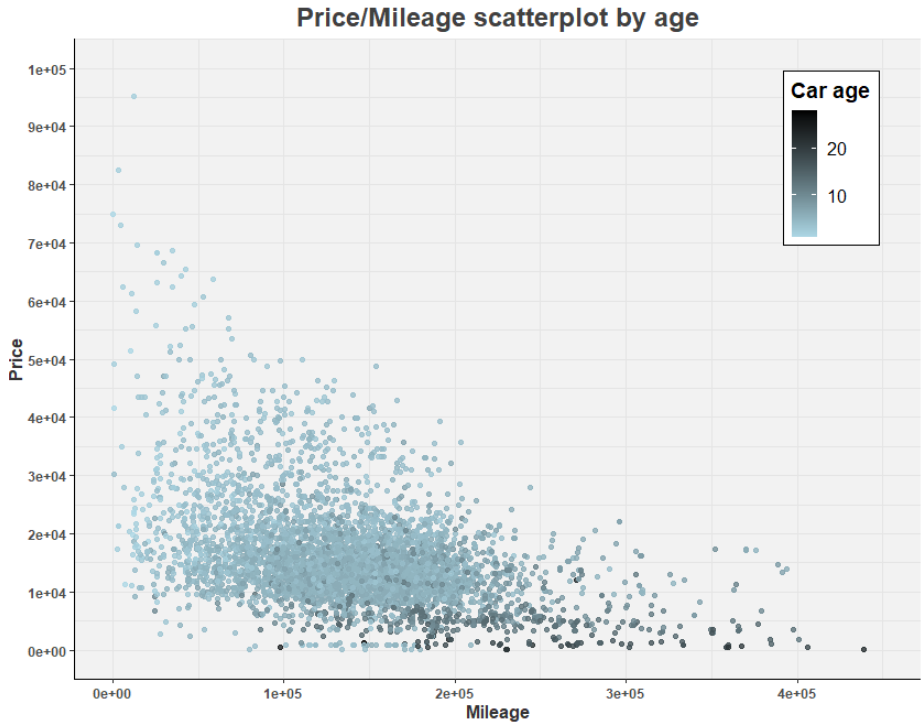

Figure 16: Mileage/Price by Car age scatterplot

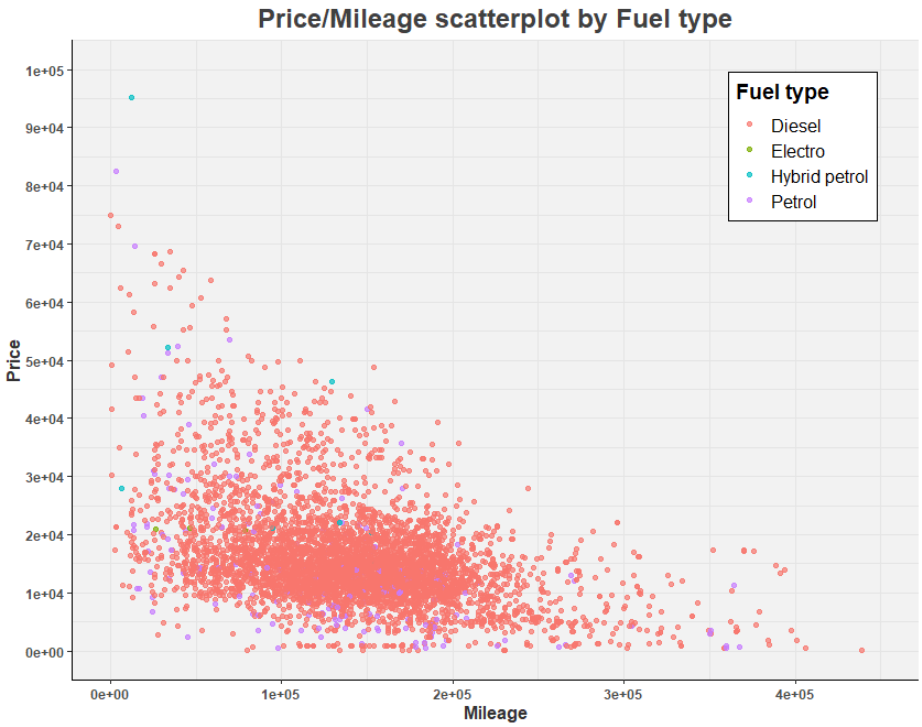

Figure 17: Mileage/Price by Fuel type scatterplot

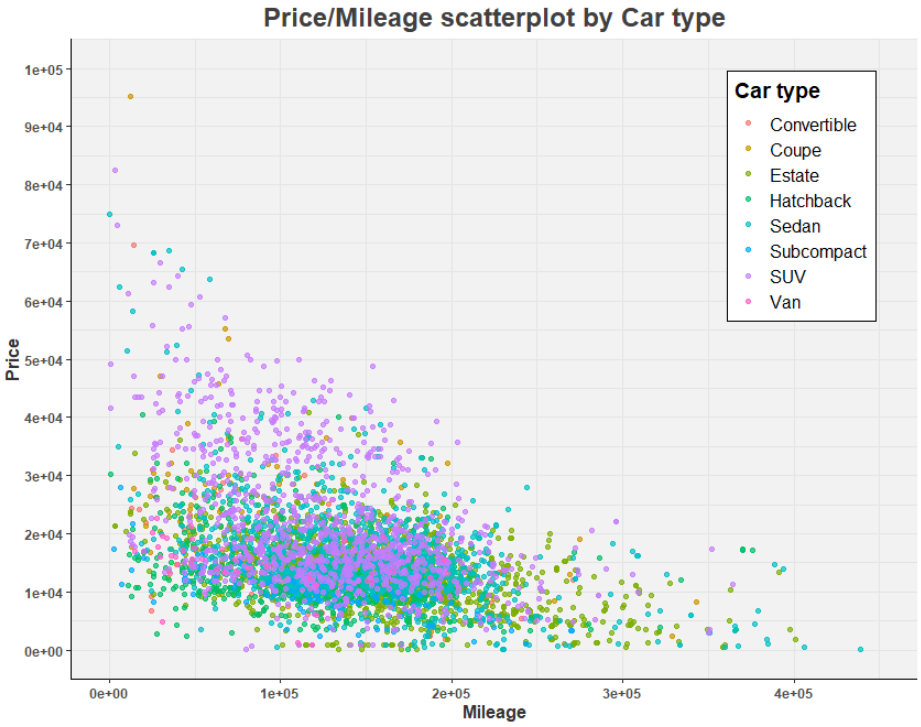

Figure 18: Mileage/Price by Car type scatterplot

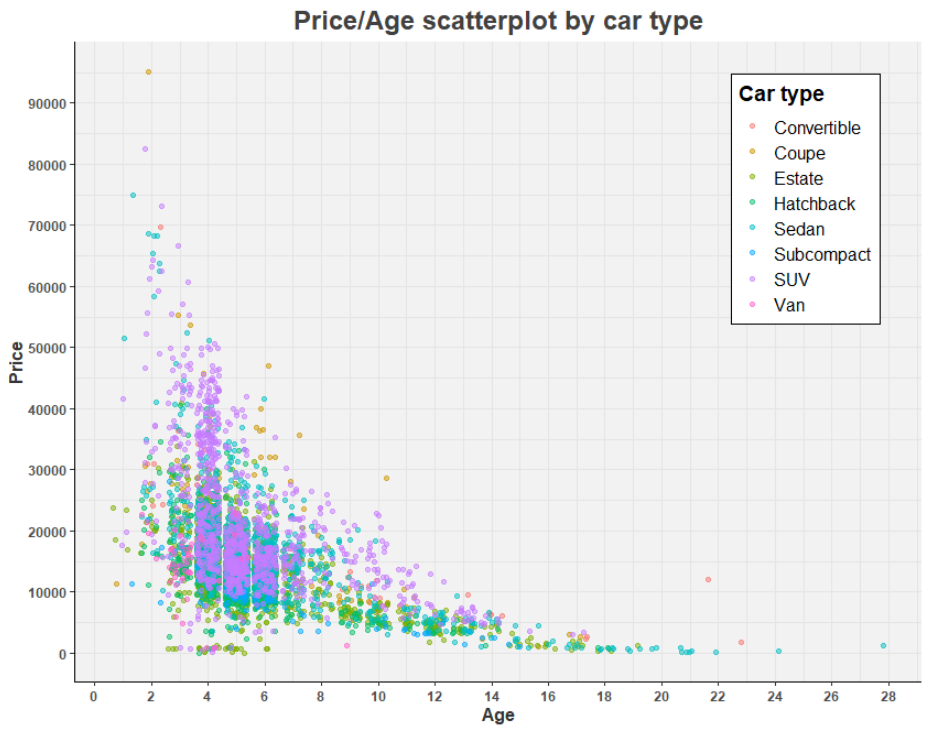

Figure 19: Age/Price by Car type scatterplor

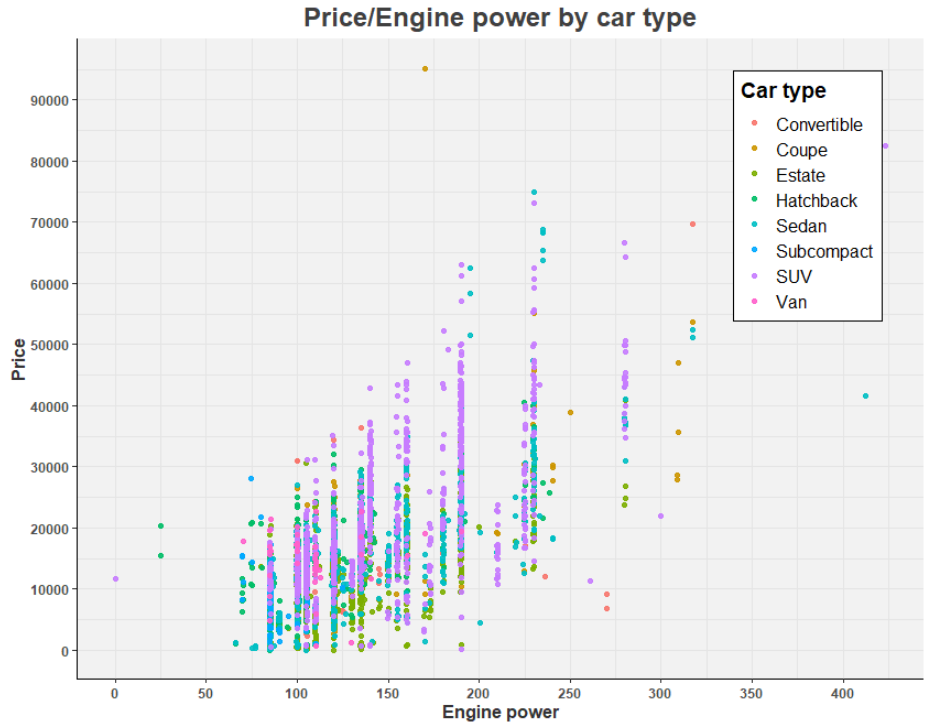

Figure 20: Engine/Price by Car type scatterplot

6 Further data pre-processing

En base a lo observado en las figuras anteriores, se lleva a cabo la siguiente ingeniería (sencilla) de atributos:

- El precio de venta no parece variar significativamente dependiendo en la época en que fue vendido, por lo que la variable **sold_at_season** se descarta.

- La variable **fuel_type** se reclasifica en dos categorías en una nueva variable denominada **fuel_cat**. Una de estas categorías corresponde a los autos que utilizan combustibles que son más caros (*electro y hybrid-petrol*), y la otra categoría corresponde a los autos más baratos (*diesel y petrol*).

- La variable **car_type** se reclasifica siguiendo un criterio similar al expuesto para la variable anterior en una nueva variable llamada **car_cat**. En este caso también se determinan dos nuevas categorías: categorías de precios altos (*coupe, SUV y van*) y categorías de precios bajos (*convertible, estate, hatchback, sedan, subcompact*).

- Finalmente, para la variable **paint_color** se determinan 3 categorías en una nueva variable llamada **color_cat**: color caro (*orange*), color barato (*green*) y color normal (*beige, black, brue, brown, grey, red, silver, white*).

- Cabe señalar que el criterio para reclasificar las 3 variables anteriores utiliza una comparación de la media de cada categoría con las demás categorías.

De esta forma, las variables a utilizar en los modelos para predecir la variable **price** son las siguientes: **model_key**, **mileage**, **engine_power**, **car_age**, **fuel_cat**, **car_cat**, **color_cat**, **feature_1**, **feature_2**, **feature_3**, **feature_4**, **feature_5**, **feature_6**, **feature_7** y **feature_8**

7 Predicción

7.1 Modelos

Para predecir el precio de un auto se parte de un modelo de regresión lineal como *benchmark* que servirá como punto de comparación para los demás modelos. En particular, se consideran los siguientes modelos: *ridge*, *lasso*, *elastic net*, *decision tree*, *bagging*, *random forest* y *gradient boosting machine*.

Debido a la naturaleza de los modelos y las restricciones computaciones de *R*, dos *datasets* ligeramente diferentes fueron utilizados. Uno de ellos contiene todas las variables a excepción de **model_key** (llamo a este dataset "simple"), y el otro contiene absolutamente todas las variables mencionadas anteriormente (llamo a este dataset "complejo"). En particular, los modelos que utilizan el *dataset* "simple" son el de regresión lineal, *ridge*, *lasso*, *elastic net* y *random forest*. Mientras que los demás modelos utilizan el *dataset* complejo.

El motivo se debe a que existen modelos de autos que tienen muy pocas observaciones (varios tienen tan solo una observación), lo que hace muy probable[†] que al hacer el *train-test split* ciertos modelos de autos solamente aparezcan en el *test set*. En este caso, el modelo no aprende estas categorías y al momento de evaluar el modelo fuera de la muestra, y por cómo funciona el algoritmo de fondo, no se puede llevar a cabo una predicción para estos modelos de autos que durante el aprendizaje no aparecieron. Es por ello que para estos algoritmos se decidió simplemente no utilizar esta variable, a pesar de lo sumamente informativo que puede llegar a ser saber el modelo de un auto para determinar su precio.

Lo anterior es válido para los modelos de regresión lineal, *ridge*, *lasso* y *elastic net*. Para el caso de *random forest* el problema en utilizar la variable **model_key** radica en que tiene más categorías del máximo permitido por el algoritmo en *R*. En particular, el algoritmo permite un máximo de 53 categorías, mientras que esta variable tiene 75 categorías. Una posibilidad para resolver este problema fue implementar *One-Hot Encoding*, sin embargo, al hacerlo ocurría un problema similar al mencionado para los otros modelos[‡].

[†]Y de hecho ocurrió cuando corrí el código para diferentes seeds.

[‡]Cabe destacar que este problema no ocurre en *Python* que tiene otro algoritmo de fondo que permite manejar muchas categorías, pero no sé hacerlo en *Python*, por lo que no lo hice.

Para los demás modelos se utiliza el *dataset* "complejo". Para estos modelos también se utilizo el *dataset* "simple", pero al observar, como uno habría de esperar, que los resultados mejoraban significativamente al utilizar **model_key**, se decidió trabajar con el *dataset* "complejo" para estos modelos.

En todos los modelos en los que se deben elegir hiperparámetros, el método elegido de validación cruzada es *10-Fold CV*. Cuando este no sea el caso se aclarará correspondientemente. Respecto al método de búsqueda de hiperparámetros adecuados, se utiliza el método de *Grid Search*. Para este fin, en cualquier de los *datasets* considerados se llevó a cabo un *train-test split* de *70/30* para implementar validación cruzada y la posterior evaluación y comparación de los modelos.

Finalmente, la métrica utilizada para evaluar la performance de los diferentes modelos para observaciones fuera de la muestra es el error cuadrático medio. Para facilitar la comprensión se trabaja específicamente con la raíz del error cuadrático medio, lo cual procura las mismas unidades que la variable objetivo.

A continuación, se presentan individualmente los resultados de los diferentes modelos, como también algunas observaciones que valen la pena destacar respecto a estos. En la próxima subsección se comparan los resultados de los diferentes modelos.

Regresión Lineal

Como se mencionó anteriormente, para este modelo se utilizan todas las variables señaladas en el apartado del preprocesamiento de datos, a excepción de **model_key**. Como este modelo simple no admite ajuste de hiperparámetros, se entrena con la muestra *train* y se evalúan los resultados con la muestra *test*. El RMSE fuera de la muestra es de 4502.31.

Recordemos que este modelo sirve como *benchmark* para los próximos modelos.

Ridge

Este modelo utiliza el *dataset* "simple". Se utiliza una grilla para los valores de $\lambda \in [0, 22500]$. En la figura 21, a modo de ilustración se pueden observar los valores de los coeficientes para diferentes valores de λ utilizando el *train set*.

El valor de λ óptimo es 151.897 y el RMSE fuera de muestra es 4500.83. No se observa ninguna mejora significativa respecto al *benchmark*. Nuevamente, a modo de ilustración, en la figura 22 se muestra el error cuadrático medio para distintos valores de λ y el óptimo señalado por validación

cruzada.

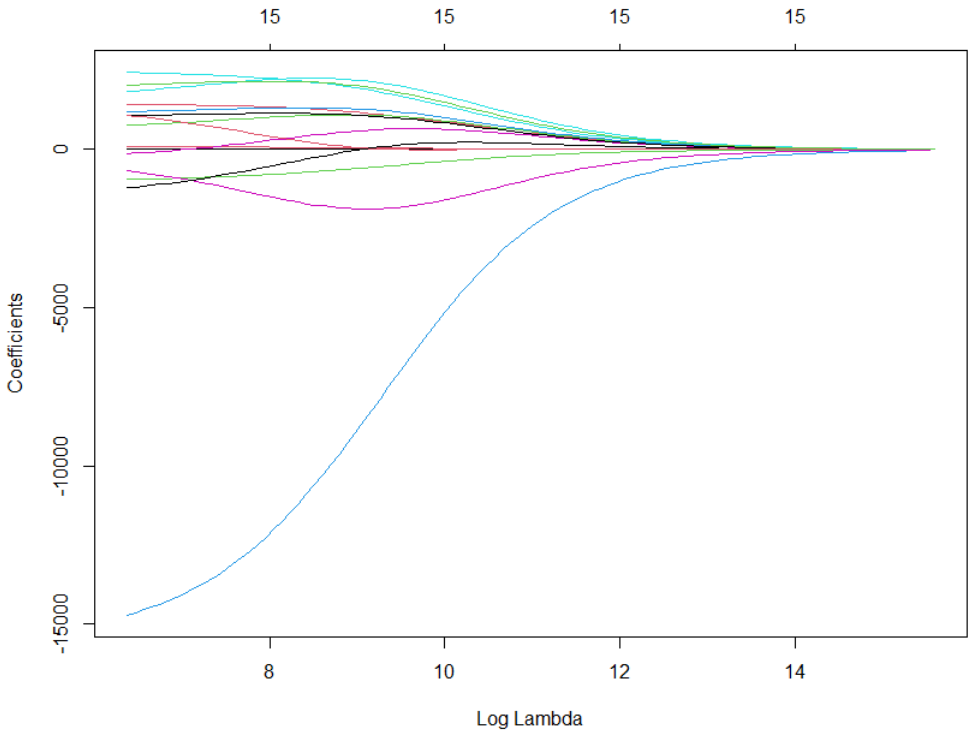

Figure 21: Ridge - Log Lambda vs Coefficients

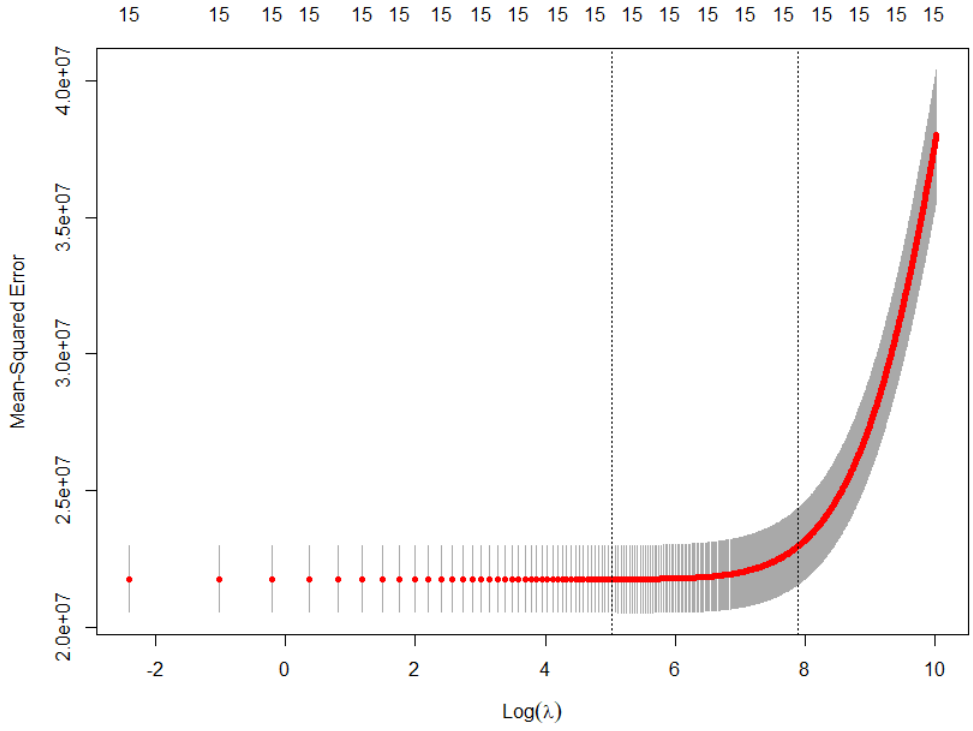

Figure 22: Ridge - Log Lambda vs MSE

Lasso

Este modelo utiliza el *dataset* "simple". Se utiliza una grilla para los valores de $\lambda \in [0, 22500]$. En la figura 23, a modo de ilustración se pueden observar los valores de los coeficientes para diferentes valores de λ utilizando el *train set*.

El valor de λ óptimo es 20.331 y señala el uso de 14 variables en lugar de 15. El RMSE fuera de muestra es 4494.48. No se observa ninguna mejora significativa respecto al *benchmark*. Nuevamente, a modo de ilustración, en la figura 24 se muestra el error cuadrático medio para distintos valores de λ y el óptimo señalado por validación cruzada.

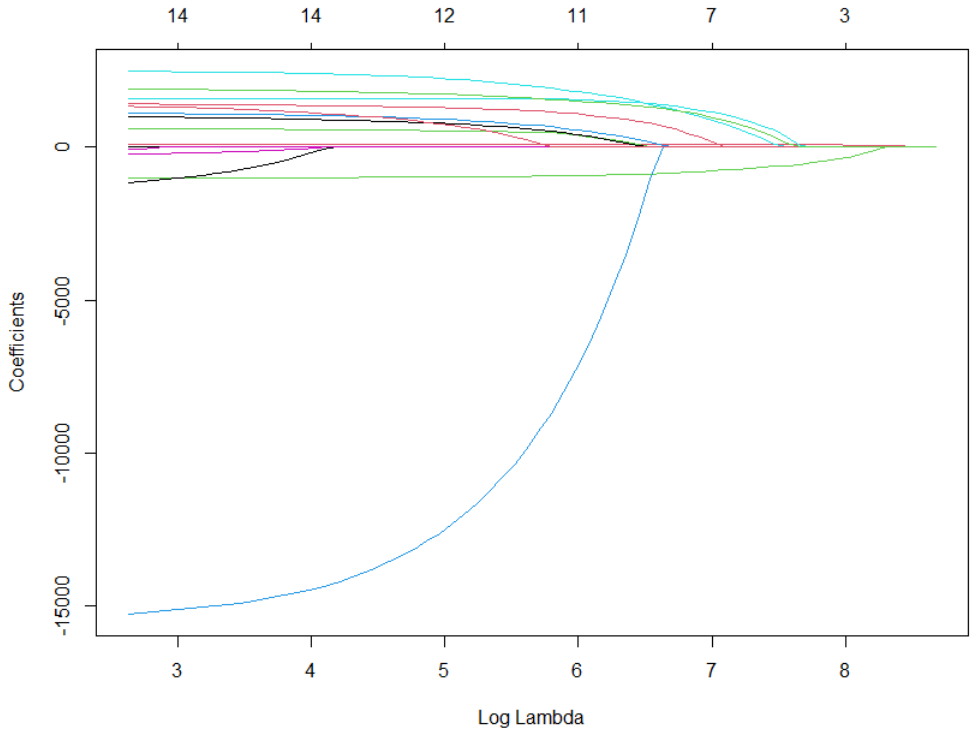

Figure 23: Lasso - Log Lambda vs Coefficients

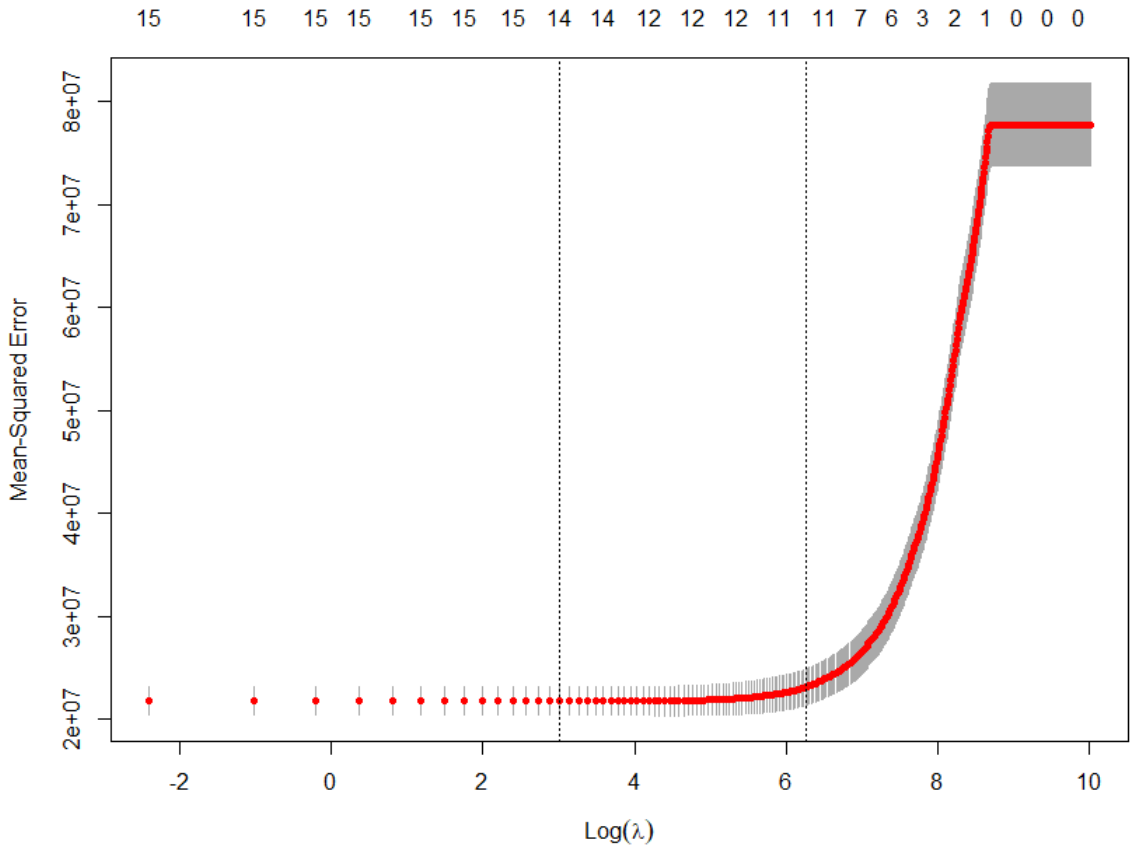

Figure 24: Lasso - Log Lambda vs MSE

Elastic Net

Este modelo utiliza el *dataset* "simple". Se utiliza una grilla para valores de λ y α. La secuencia de valores para λ es la misma que en los dos modelos anteriores. Mientras que para α se utiliza una secuencia de 0.05 a 0.95 con saltos de 0.05 (19 valores en total).

Por validación cruzada se obtiene que los valores óptimos para λ y α son 9.036 y 0.95, respectivamente. Este último valor indica un modelo muy cercano al *lasso*. A modo de ilustración se muestra en la figura 25 el error cuadrático medio para distintos valores de λ, junto con el óptimo señalado, para el mejor α señalado por validación cruzada. Notar que, como es de esperar, la figura es muy similar a la anterior que correspondía al modelo *lasso*.

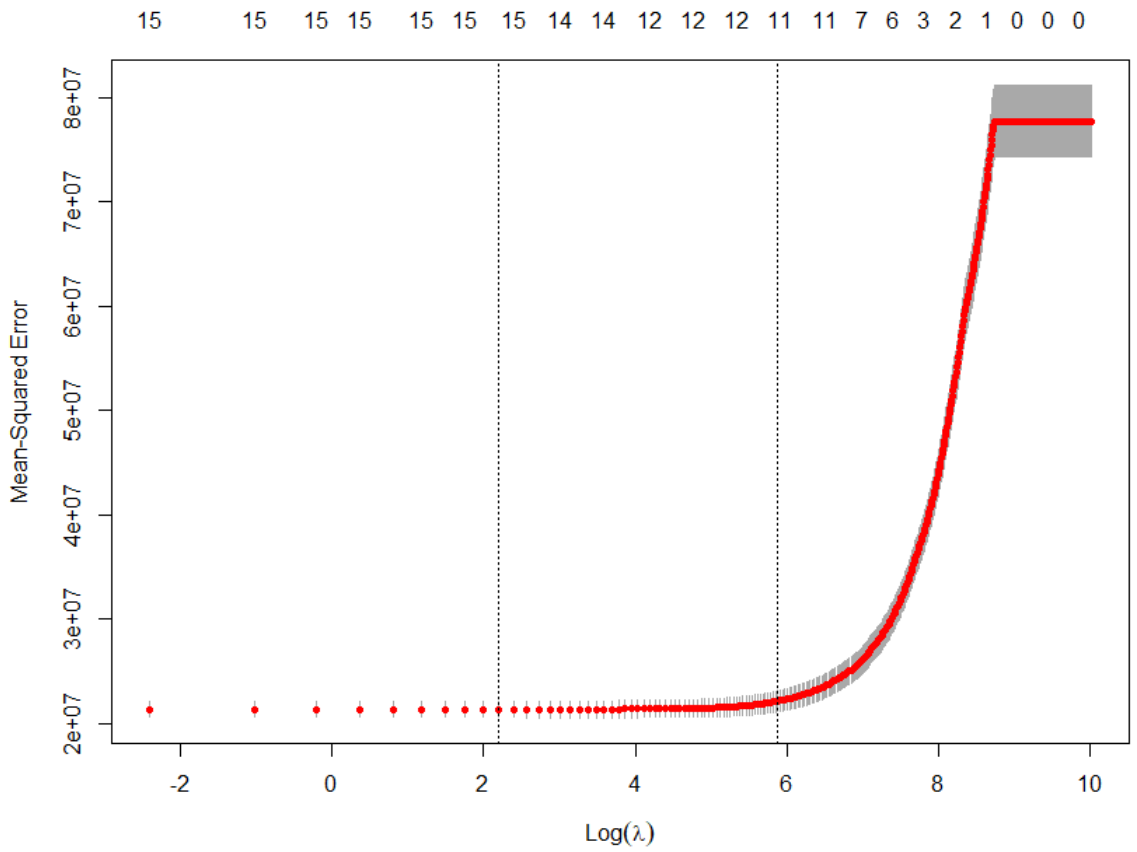

Figure 25: Elastic Net - Log Lambda vs MSE

El RMSE para el *test set* es de 4498.87. No se observa ninguna mejora significativa respecto al *benchmark* y empeora el resultado frente al modelo *lasso*.

Decision Tree

Para este modelo se utiliza el *dataset* "complejo". Se utiliza un árbol de decisión con los siguientes hiperpárametros: *cp: 0.005*, *minsplit: 10*, *xval: 10* y *max_depth: 50*. *cp* corresponde al hiperparámetro de complejidad, *minsplit* es el mínimo número de observaciones que deben existir en un nodo para intentar una nueva división, *xval* es el número de validaciones cruzadas, y *max_depth* es la profundidad máxima de cualquier nodo en el árbol final.

En la figura 26 se meustra la relación entre el error relativo obtenido por validación cruzada contra el parámetro de complejidad, *cp*, y el tamaño del árbol. Dicha figura muestra que el valor óptimo de *cp* es 0.005 y que no es necesario "podar" el árbol.

Respecto a la importancia relativa de las variables, destacan las siguientes: **model_key** (0.47), **engine_power** (0.18), **car_age** (0.15), **mileage** (0.06), **feature_8** (0.04), y el resto de la

variables comprenden la proporción restante. En otras palabras, la incorporación de la variable **model_key** parece haber sido una buena decisión.

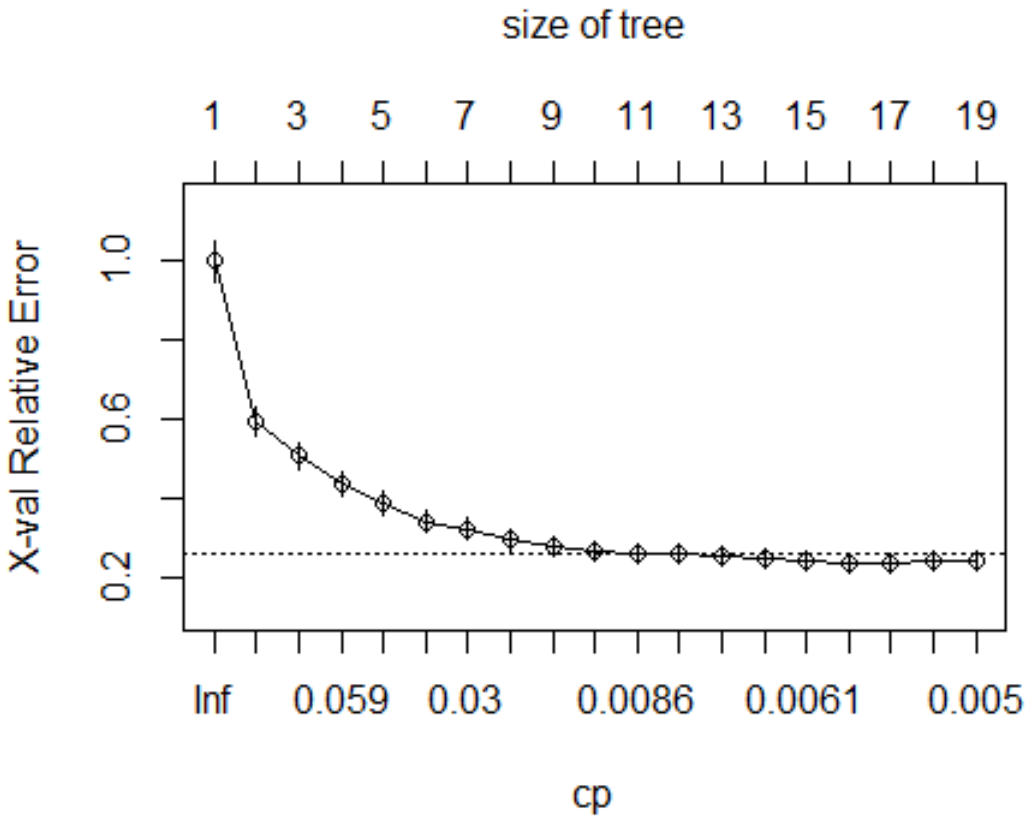

Figure 26: Decision Tree - Xval relative error vs complexity parameter

El RMSE fuera de muestra para este modelo es de 4037.96, lo cual representa una mejor significativa del 10% respecto al *benchmark* y los modelos previos.

Bagging

Para este modelo se utiliza el *dataset* "complejo". Se utiliza el método *treebag* para la estimación y el método *OOB* para la validación cruzada. Se crea una grilla de valores para la cantidad de árboles considerados ($nbagg \in \{100, 200, 300, 400, 500, 1000, 3000\}$) y para la cantidad mínima de observaciones en un node para considerar una partición ($minsplit \in \{5, 10, 15, 20, 50, 100, 200, 500\}$). Se considera un parámetro de complejidad *cp* de 0.001.

Por validación cruzada se determina que los valores óptimos para *nbagg* y *minsplit* son 100 y 5, respectivamente. El RMSE fuera de muestra con estos parámetros es de 3195.42, lo cual representa

una mejora de casi el 30%. En la figura 27 se muestra la importancia relativa de cada variable.

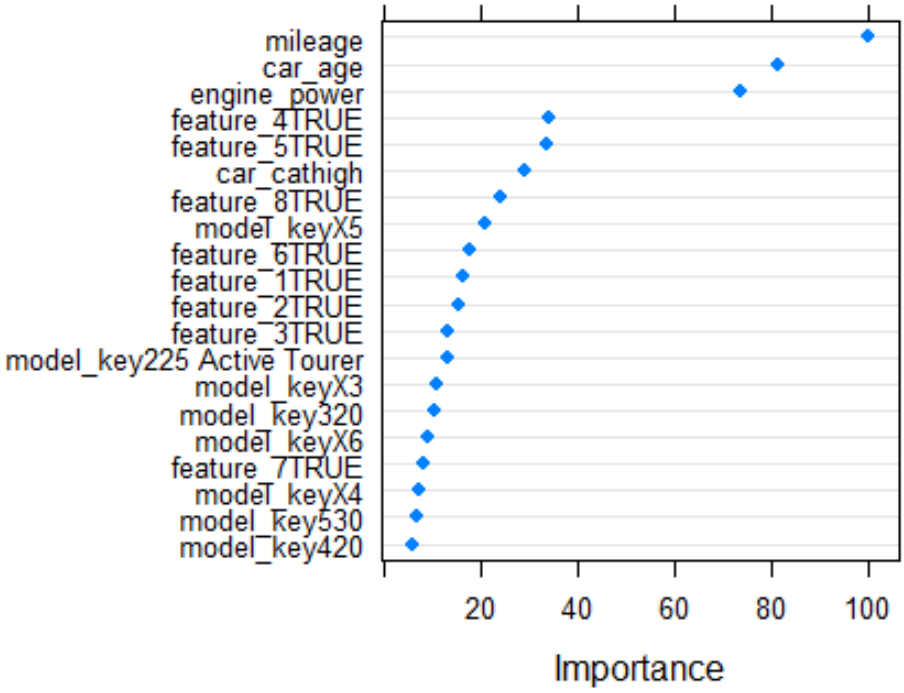

Figure 27: Bagging - Feature Importance

Random Forest

Para este modelo se utiliza el *dataset* "simple". Se crea una grilla de valores para la cantidad de variables consideradas en cada árbol ($mtry \in \{3, 4, 5, 6, 7, 8, 9\}$), la cantidad máxima de nodos terminales ($maxnode \in \{5, 10, 20, 50, 100, 200\}$, la cantidad de árboles considerados ($ntree \in \{100, 200, 500, 1000, 3000\}$) y la cantidad mínima de observaciones en un nodo en el árbol terminal ($nodesize \in \{5, 10, 20, 50, 100, 500\}$).

Por validación cruzada se obtiene que los valores óptimos para *mtry, maxnode, ntree* y *nodesize* son 9, 200, 500 y 20, respectivamente. Al entrenar el modelo con el *train set* y estos hiperparámetros y evaluar la performance en el *test set* se obtiene un RMSE de 3280.91, lo cual representa una mejora sustancial respecto al *benchmark*, aunque empeora en relación al modelo de *Bagging*.

Gradient Boosting Machine

Para este modelo se utiliza el *dataset* "complejo". Se crea una grilla de valores para el

learning rate (*shrinkage* $\in \{0.3, 0.1, 0.05, 0.01, 0.005, 0.001\}$), la profundidad máxima de cada árbol (*interaction.depth* $\in \{1, 3, 5, 7, 10, 15, 20, 30\}$) y la mínima cantidad de observaciones en cada nodo (*minobsinnode* $\in \{5, 10, 2050, 100\}$). Se utilizan 5000 árboles para cada combinación de hiperparámetros (*n.trees*), una fracción de entrenamiento δ de 0.8 y una fracción de *bag* η de 0.7.

Los valores óptimos obtenidos a través de validación cruzada para *shrinkage, interaction.depth* y *minobsinnode* son 0.05, 15 y 5, respectivamente. Con estos valores se obtiene una performance *out-of-sample* medido por el RMSE de 3424.26, lo cual representa una mejora importante respecto al modelo *benchmark*, aunque no así respecto al modelo de *Bagging*.

7.2 Comparación de modelos

A continuación, se muestra una comparación de la performance *out-of-sample* de los distintos modelos. Como se vio anteriormente, la medida utilizada es la raíz del error cuadrático medio (RMSE). También se presenta el R^2 para cada modelo (cabe aclarar que en ningún momento se utilizó el R^2 como medida de performance al momento de elegir hiperparámetros).

Se debe recordar que para los modelos de regresión lineal, *ridge* , *lasso*, *elastic net* y *random forest* se utilizo la base de datos "simple", es decir, aquella que no incluye la variable **model_key**. Mientras que para los modelos de *decision tree*, *bagging* y *gradient boosting machine* se utilizo la base de datos "compleja", es decir, la que incluye la variable **model_key**. Cabe recordar que para estos últimos tres modelos también se obtuvieron las dos medidas que se muestran a continuación, sin embargo, para simplificar la exposición, se decidió quedarse con los resultados (relativamente mejores) que surgen de usar la mencionada variable.

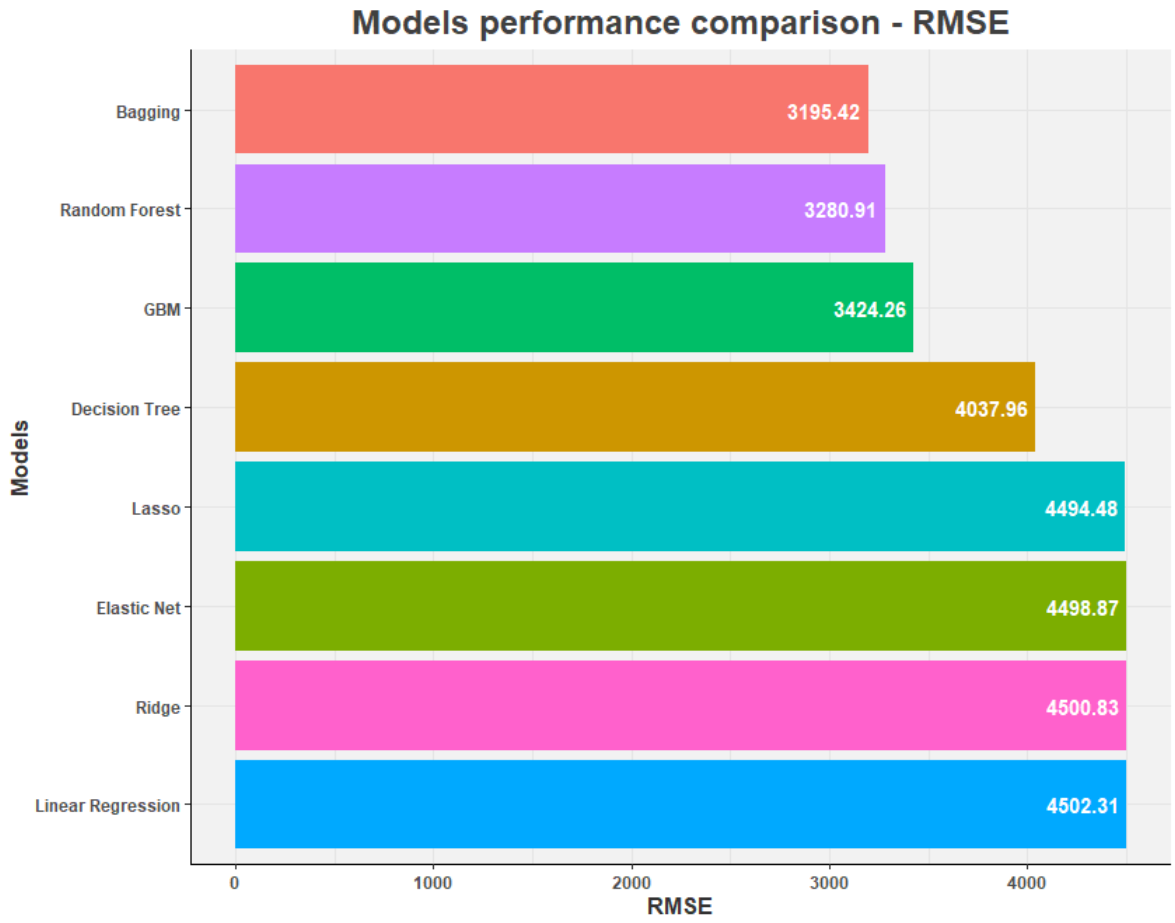

Figure 28: Comparación de modelos - RMSE

Se observa con facilidad que los modelos de *bagging*, *random forest* y *gradient boosting machine* son los que mejor performean en el *test set*, llegando a mostrar una mejora del 29%, 27% y 24% respecto al modelo *benchmark* de regresión lineal, respectivamente. También se observa una mejora significativa contra el modelo de regresión lineal cuando se utiliza un árbol de decisión, aproximadamente un 10%. Finalmente, si bien la performance mejora al utilizar *ridge, lasso* y *elastic net*, estas mejoras son ínfimas respecto al modelo de regresión lineal.

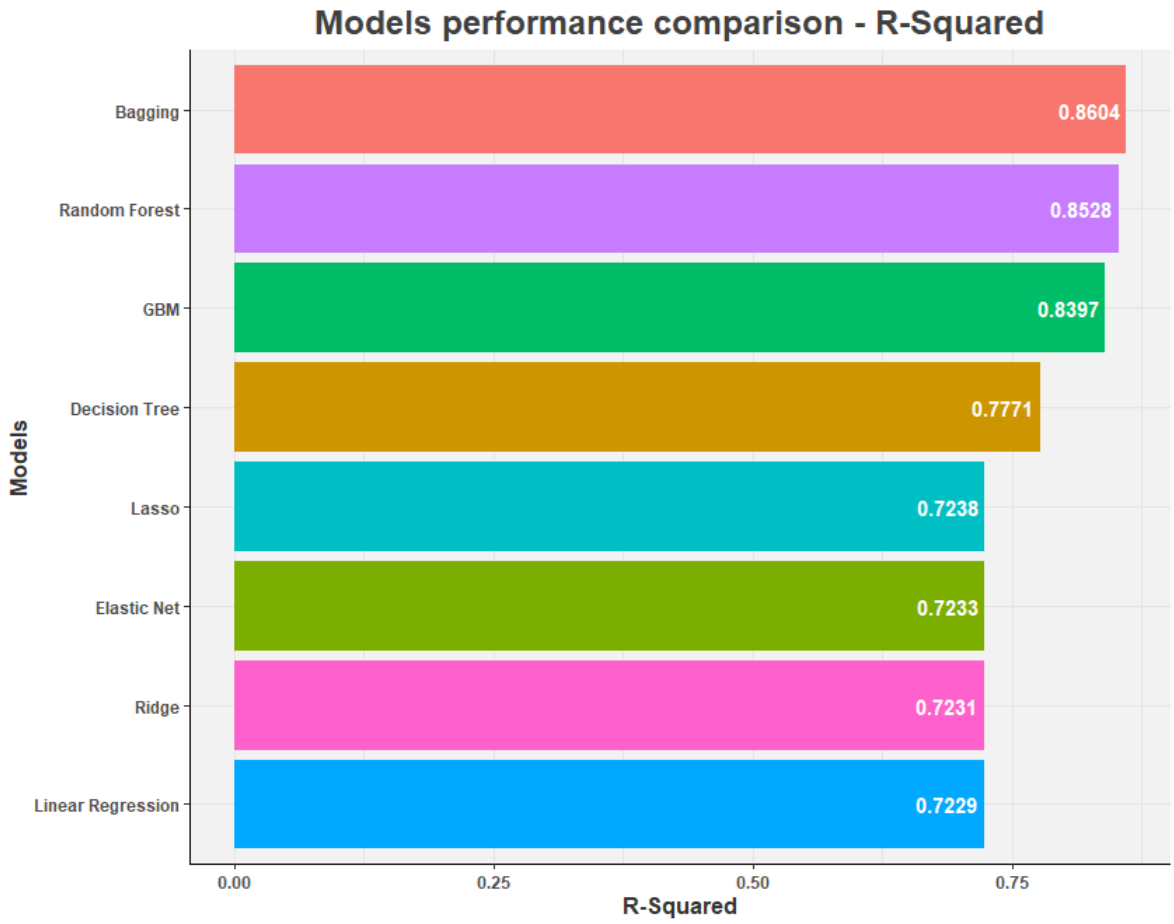

Figure 29: Comparación de modelos - R^2

Resultados análogos a los anteriores se observan al considerar como medida de performance el R^2. El modelo de *bagging* llega a explicar el 86% de la variabilidad de los datos *out-of-sample*, seguido por el modelo de *random forest* con el 85%; y *gradient boosting machine* con el 84%. Luego, se observa que la proporción explicada por el modelo de *decision tree* disminuye respecto a los mejores modelos pero persiste la mejor performance al compararlo con el *benchmark*. Nuevamente, los modelos de *lasso, ridge* y *elastic net* no presentan mejoras relevantes respecto al modelo de regresión lineal.

7.3 Actual vs Predicted

Finalmente, y a modo de ilustración, se muestran la predicciones de los modelos contra los precios verdaderos para las observaciones de la muestra de testeo.

En los modelos de regresión lineal, *ridge*, *lasso* y *elastic net* parece haber un problema sesgo, es

decir, hay estructura subyacente en los datos que estos modelos no logran capturar correctamente. En particular, para precios verdaderos muy bajos parece haber mucha varianza; por otra parte, estos modelos no predicen bien valores altos.

Al considerar el modelo de *decision tree*, no parece haber problema de sesgo pero sí de varianza. Finalmente, al considerar los tres modelos restantes, en ninguno de ellos pareciera haber un problema significativo de sesgo, aunque hay lugar para mejoras en lo que varianza respecta.

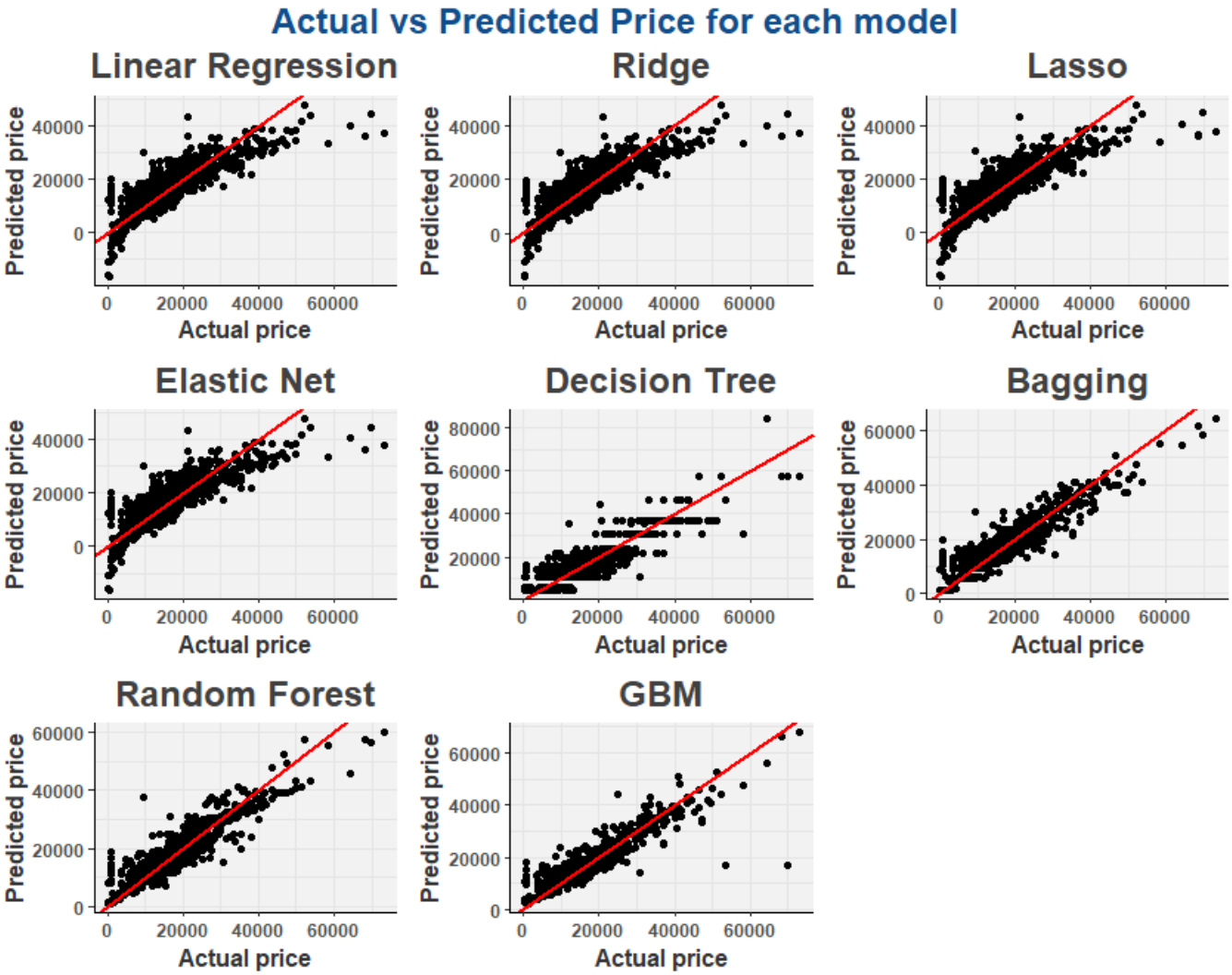

Figure 30: Actual vs Predicted

Se puede especular que el motivo por el que todos los modelos fallan para predecir el verdadero precio de autos con valores muy bajos (de 100 a 1000 dólares) se debe a que estos valores no tienen sentido alguno. De todas formas, sin mayor conocimiento sobre el rubro y el porqué de estos precios

absurdamente bajos, no se puede simplemente eliminarlos o asumir algún error de entrada de datos.

8 Conclusión

Se propuso el objetivo de predecir el precio de un auto según ciertas características lo más fielmente posible. Para ello se exploraron diferentes modelos y fueron comparados en una muestra aparte con respecto a un modelo simple de regresión lineal como *benchmark*. Se observó que modelos complejos son capaces de predecir mejor el valor de un auto respecto a modelos más simples. En particular, dentro de los modelos complejos, el de *bagging* es el que mejor performance muestra en el *test set*. En otras palabras, hay ganancias al utilizar modelos más complejos para predecir el precio de un auto.

A modo de conclusión, propongo algunas mejoras que potencialmente se podrían incorporar a este trabajo. En primer lugar, se podría implementar algún tipo de tratamiento de datos para las variables **feature_X**. También se podría llevar a cabo un análisis de *outliers* para la variable **price** para determinar qué tan factible es la existencia de precios absurdamente pequeños para un auto. Por otra parte, la inmensa cantidad de clases en la variable **model_key** exige un tratamiento más sútil para determinar como incoporar esta variable, y que la alta dimensionalidad que surge de esta no perturbe significativamente los resultados de los algoritmos.

En segundo lugar, siempre existe la posibilidad de llevar a cabo una búsqueda más exhaustiva de valores óptimos para los hiperparámetros, siendo la contracara un mayor costo computacional. A tal fin se podrían implementar métodos de *Random Search* o, de forma más sofisticada, *Informed Search* para ahorrar parte del costo computacional.

Finalmente permanece como posibilidad la implementación de otros modelos igual o más complejos que los utilizados en este trabajo, como *SVM*, *XGBoost* y redes neuronales en cualquiera de sus variantes. Potencialmente, estos dos últimos algoritmos podrían implicar mejores significativas sobre el modelo *benchmark* y también sobre el modelo de *bagging*.