Universidad del Desarrollo
Facultad de Ingeniería

ALGORITMOS DE COMPARACIÓN PARA ELEGIR UN AUTOMÓVIL

Las personas compran autos chinos como su primera opción

POR: VERÓNICA ALEJANDRA CESTARI OSAL

ADOLFO ANDRÉS LOBOS MUÑOZ

Proyecto de grado presentado a la Facultad de Ingeniería de la Universidad del Desarrollo para optar al grado académico de Magíster en Data Science

PROFESOR GUÍA:

Cristian Candia Ph.D.

Diciembre 2022

SANTIAGO

AGRADECIMIENTO

Verónica Cestari:

Agradezco a mi marido Alejandro Montilla por ayudarme en este camino de estudio con el cuidado de nuestros hijos Luciano Montilla y Allegra Montilla, todo ese tiempo hizo posible completar este nuevo grado.

Agradecimiento a Adolfo Lobos por esperarme 1 año por el nacimiento de mi hija para continuar con nuestros ramos y sobre todo hacer este proyecto final en equipo.

Agradecimiento a mamá Julieta Osal porque también fue clave para ayudarme con el cuidado de mis hijos mientras tenía clases online y vivíamos en pandemia.

Adolfo Lobos

Siempre mi luz del norte a seguir ha sido mi familia, Jenny Bravo mi esposa y mis hijos Camila y Adolfo quienes siempre han empujado y apoyado para que pueda alcanzar este objetivo académico. A todos los docentes que aportaron en mi formación durante este proceso y mi compañera de estudios que ha permitido enfrentar los desafíos de cada asignatura para poder cumplirlos.

TABLA DE CONTENIDO

Resumen

Este documento muestra cómo podemos determinar la preferencia de los consumidores Chilenos para comprar un vehículo de procedencia China por sobre el de otro país.

Las costumbres apuntan que las personas son tradicionales en nuestro país y cuando un producto nuevo ingresa al mercado se hace difícil su penetración y esto no ha sido la excepción de los vehículos de origen Chino.

1. Introducción

En nuestro país en la década de los 80 penetraron los autos japoneses en el mercado nacional la cual fue tímida y con el pasar de los años se han transformado en marcas de altos estándares en cuanto a calidad, seguridad, lujo entre otros y claramente sus precios dejaron de ser los más económicos del mercado. Hoy con los autos de origen China esta ocurriendo algo similar su ingreso fue con un precio más competitivo que el resto y con mayor cantidad de adicionales que sus competencias en los distintos segmentos. Esto ha llevado a los usuarios a empezar a preferir estas marcas; sin embargo, no tenemos certeza si esto es una tendencia o solo una moda.

Un adicional al comportamiento de los consumidores, fue que en nuestro país hubo una inyección de efectivo en las familias producto del 10% durante el año 2020, esto permitio a las familias poder tomar la decisión de comprar su primer vehiculo, la renovación del que ya poseian o simplemnte comprar uno mas para el nucleo familiar.

Es aca frente a estos escenarios que la pregunta planteada necesita de una respuesta y nuestros análisis se centraran en construir un modelo que permita predecir la compra de vehiculos Chinos y que features pueden influir en esta desición. Para tales efectos es que se decidió el utilizar el modelo Random Forest con el objetivo de determinar si la variable precio es la principal en determinar la elección.

2. Trabajo Relacionado

Lili Zhang, J. P. (2021). Measuring Customer Similarity and Identifying Cross-Selling Products by Community Detection. *Mary Ann Liebert, Inc. DOI: 10.1089/big.2020.0044*, 133 - 142. https://www.liebertpub.com/doi/pdf/10.1089/big.2020.0044

Sari, Puspita. & Purwadinata, Adelia. (2019). Analysis Characteristics of Car Sales in e-commerce Data Using Clustering Model. *Journal of Data Science and Its Applications*, 19-28. https://commdis.telkomuniversity.ac.id/jdsa/index.php/jdsa/article/view/19/9

3. Hipótesis y Objetivos

En el contexto de la venta de los autos chinos ha venido incrementando desde el año 2017 hasta la fecha, eventualmente podrían generar quiebres de stock por que los consumidores se han atrevido a comprar marcas sin trayectoria nacional en comparación con otras marcas de vehículos reconocidos en nuestro mercado por su calidad mecánica y seguridad.

¿Es posible que el atributo económico influya directamente en los consumidores para la toma de decisión de compra de un auto chino, o predominen las condiciones de seguridad y confort que le dan las marcas tradicionales?

Objetivos generales

- Identificar si las personas prefieren comprar un vehículo de procedencia China por sobre el de otro origen.

Objetivos específicos

- Analizar las variables relacionadas con inscripción vehicular de Chile.
- Limpiar y normalizar los atributos de las fuentes de datos orígenes.
- Aplicar un modelo que permita la identificación de clientes que comprarían un automóvil Chino.
- Probar si el features precio es determinante en la elección de compra de un vehículo de origen chino.

4. Datos y Metodología

Para poder aplicar el modelo predictivo se tomaron los datos entregados; sin embargo, estos no entregaban una correlación que fortaleciera los resultados entre las distintas features con lo cual se complementaron los datos con la base de tasación de vehículos 2022 publicada en la página de SII (Servicio de impuestos internos) generando un dataset con valores cuantitativos que fortalecerán los resultados de los modelos aplicados y de esta forma apalancara nuestra hipótesis en cuanto a la variable económica como terminante para realizar la compra.

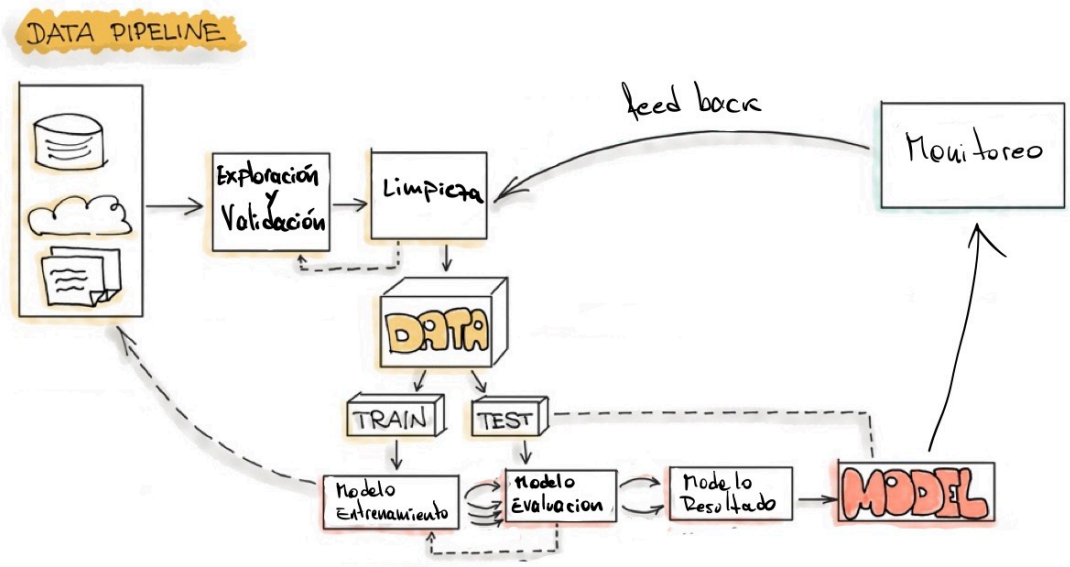
Una vez que contamos con la data nuestro proyecto de análisis se dividió en dos Pipeline, el primero: Data y el segundo: Machine Learning.

Para la ejecución esta fue desarrollada en cascada, con el Pipeline Data, comenzamos con la preparación, exploración y validación de los datos, de esta etapa generamos una tarea de limpieza de valores Nulos por cada una de las distintas columnas asociada al data frame autoVF, para este tipo de valor tomamos la decisión de ser reemplazados por valores promedios. Una vez finalizado el proceso de limpieza se dividieron los datos en dos grupos: Entrenamiento (70%) y Test (30%).

En el segundo Pipeline Machine Learning, comenzó el proceso de construcción de nuestro modelo para ello definimos el modelo aplicando Random Forest.

Este proceso se puede ver representado en la figura 1 (MLOps, s.f.). Esta imagen fue tomada y adaptada a nuestro trabajo.

Figura 1 Representación gráfica del proceso



Los datos utilizados para este trabajo fueron proporcionados por la Universidad; sin embargo el valor de las tasaciones fiscales de los automóviles se descargó de la página oficial del Servicio impuestos internos.

La descripción del data frame *Diccionario_Modelos_Inscripciones.csv – tabla 1* es el contener el detalle de todos los modelos de autos asociado a las distintas marcas, el campo CHINA es el relevante para nosotros ya que con él se puede segmentar la procedencia de los automóviles.

Tabla 1 Diccionario_Modelos_Inscripciones.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25626 entries, 0 to 25625
Data columns (total 18 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   MARCA EN BASE MODELO EN BASE  25626 non-null  object
 1   FECHA                         22993 non-null  object
 2   VEHICULO                      25626 non-null  object
 3   TIPO                          25622 non-null  object
 4   MARCA EN BASE                 25625 non-null  object
 5   MODELO HOMOLOGADO             25593 non-null  object
 6   MODELO EN BASE                25625 non-null  object
 7   MARCA                         25626 non-null  object
 8   MODELO FAMILIA                25626 non-null  object
 9   SPORT                         25626 non-null  object
 10  CHINA                         25626 non-null  object
 11  MARCA MODELO HOMOLOGADO       25626 non-null  object
 12  MARCA MODELO FAMILIA          25626 non-null  object
 13  N SEGMENTOS                   25626 non-null  int64
 14  N SPORT NO SPORT              25626 non-null  object
 15  SEGMENTO                      25626 non-null  object
 16  CLASIFICACION                 25626 non-null  object
 17  N MH                          25626 non-null  object
dtypes: int64(1), object(17)
```

	MARCA EN BASE MODELO EN BASE	FECHA	VEHICULO	TIPO	MARCA EN BASE	MODELO HOMOLOGADO	MODELO EN BASE	MARCA	MODELO FAMILIA	SPORT	CHINA	MARCA MODELO HOMOLOGADO	MARCA MODELO FAMILIA	N SEGMENTOS
0	ABARTH 595 E6D 695 RIVALE 1.4	43617	LIVIANO Y MEDIANO	Vehículo de Pasajeros	ABARTH	595	595 E6D 695 RIVALE 1.4	ABARTH	595	SPORT	OTRO	ABARTH 595	ABARTH 595	
1	ABARTH 595E6D 1.4	43617	LIVIANO Y MEDIANO	Vehículo de Pasajeros	ABARTH	595	595E6D 1.4	ABARTH	595	SPORT	OTRO	ABARTH 595	ABARTH 595	
2	ABARTH ABARTH 595 1.4	43800	Livianos y Medianos	AUTO	ABARTH	595	ABARTH 595 1.4	ABARTH	595	SPORT	OTRO	ABARTH 595	ABARTH 595	
3	ABARTH SIN MODELO	43556	LIVIANO Y MEDIANO	Vehículo de Pasajeros	ABARTH	SIN MODELO	SIN MODELO	ABARTH	SIN MODELO	NO SPORT	OTRO	ABARTH SIN MODELO	ABARTH SIN MODELO	
4	ACURA MDX	44136	LIVIANO Y MEDIANO	STW	ACURA	MDX	MDX	ACURA	MDX	NO SPORT	OTRO	ACURA MDX	ACURA MDX	

En el caso del dataset *patentes_historico_02.parquet – tabla 2* se puede tener la relación patente y RUN del propietario por cada año que la persona ha sido su propietario de un vehículo con lo cual un vehículo puede tener en el tiempo más de un RUN ya que este cambio de dueño y el año de patente es el que determina quien al momento de sacar el permiso de circulación es el propietario. Estos datos son relevantes para visualizar el comportamiento en el tiempo en cuanto a preferencias de las nuevas adquisiciones que puedan ir desarrollando en el tiempo las personas.

Tabla 2 patentes_historico_02.parquet

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26395312 entries, 0 to 26395311
Data columns (total 7 columns):
 #   Column           Dtype
---  ------           -----
 0   marca            object
 1   patente          object
 2   rut              object
 3   ano_patente      object
 4   tipo             object
 5   modelo           object
 6   ano_fabricacion  object
dtypes: object(7)
```

	marca	patente	rut	ano_patente	tipo	modelo	ano_fabricacion
0	mazda	HTKT99	0000099ac6876dd89340ff9141f50194c78e24cd62c11c...	2017	station wagon	demio 1.5 aut	2006
1	mazda	HTKT99	0000099ac6876dd89340ff9141f50194c78e24cd62c11c...	2018	station wagon	demio 1.5 aut	2006
2	mazda	LYRJ39	0000449f1c3136b7bb209a4096e403f51cbe6845343212...	2021	automovil	all new 3 2.0 aut	2020
3	mazda	CYGD16	000054d225f0246573b76f084823d11056da0b5091dbdf...	2016	station wagon	cx 7 2.5 at	2011
4	mazda	FWXD51	000136330e202fee9b4b6a87a98bd4487b92ba0ffd711c...	2017	station wagon	cx 5 gt awd 2.5 at	2013

El dataset *one_match_marca_modelo_VSII.xlsx – tabla 3* es el resultado de la unión que se realizó entre *one_match_marca_modelo* y *liv2022.xlsx – tabla 4,* archivo de SII que nos permitió obtener un dataset con la información necesaria para relacionar el modelo y obtener su tasación fiscal. Este valor económico es crucial, de acuerdo con nuestros modelos ya que es la variable que puede determinar finalmente la elección del automóvil.

Tabla 3 one_match_marca_modelo_VSII.xlsx

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35888 entries, 0 to 35887
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   marca          35888 non-null  object
 1   modelo         35888 non-null  object
 2   automovil      35888 non-null  int64 
 3   camioneta      35888 non-null  int64 
 4   furgon         35888 non-null  int64 
 5   jeep           35888 non-null  int64 
 6   station wagon  35888 non-null  int64 
 7   total          35888 non-null  int64 
 8   match_1        35888 non-null  object
 9   match_liv      35888 non-null  object
dtypes: int64(6), object(4)
```

	marca	modelo	automovil	camioneta	furgon	jeep	station wagon	total	match_1	match_liv
0	alfa romeo	4c coupe 1.7 aut	1	0	0	0	0	1	4c	4C
1	alfa romeo	giulia 2.0	4	0	0	0	0	4	giulia	GIULIA
2	alfa romeo	giulia quadrifoglio verdi 2.9 aut	1	0	0	0	0	1	giulia	GIULIA
3	alfa romeo	giulia veloce 2.0	1	0	0	0	0	1	giulia	GIULIA
4	alfa romeo	giulia veloce 280cv 2.0 aut	19	0	0	0	0	19	giulia	GIULIA

Tabla 4 liv2022.xlsx

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70411 entries, 0 to 70410
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   año           70411 non-null  int64  
 1   marca         70411 non-null  object 
 2   modelo        70411 non-null  object 
 3   puertas       70411 non-null  int64  
 4   cilindrada    70411 non-null  int64  
 5   hp            5669 non-null   float64
 6   combustible   70411 non-null  object 
 7   transmisión   70411 non-null  object 
 8   marchas       5651 non-null   float64
 9   tracción      5669 non-null   object 
 10  pais          7060 non-null   object 
 11  equipamiento  70411 non-null  object 
 12  tasacion      70411 non-null  float64
 13  Continente    70411 non-null  object 
dtypes: float64(3), int64(3), object(8)
```

	año	marca	modelo	puertas	cilindrada	hp	combustible	transmisión	marchas	tracción	pais	equipamiento	tasacion
0	2018	ASTON MARTIN	DB11	2	4000	510.0	Bencina	Automática	8.0	4x2 (2WD)	INGLATERRA	Full	118773599.0
1	2018	MAZDA	MX5	2	2000	158.0	Bencina	Automática	6.0	4x2 (2WD)	JAPÓN	Full	15585496.0
2	2018	MAZDA	MX5	2	2000	158.0	Bencina	Automática	6.0	4x2 (2WD)	JAPÓN	Full	16429414.0
3	2018	MAZDA	MX5	2	2000	158.0	Bencina	Mecánica	6.0	4x2 (2WD)	JAPÓN	Full	12671450.0
4	2018	MCLAREN	570S SPIDER	2	3800	570.0	Bencina	Automática	8.0	4x2 (2WD)	INGLATERRA	Full	119525329.0

Finalmente el dataset en el que convergió toda la información como con la maraca, patente, propietario, características técnicas de cada vehículo entre otros *datosVF.csv – tabla 5*, con él se basó todo nuestro análisis desarrollado en este trabajo.

Tabla 5 datosVF.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7267875 entries, 0 to 7267874
Data columns (total 26 columns):
 #   Column           Dtype
---  ------           -----
 0   Unnamed: 0       int64
 1   marca_x          object
 2   patente          object
 3   rut              object
 4   ano_patente      int64
 5   tipo             object
 6   modelo_x         object
 7   ano_fabricacion  int64
 8   origen           object
 9   match_1          object
 10  match_liv        object
 11  key_name         object
 12  año              float64
 13  marca            object
 14  modelo_y         object
 15  puertas          float64
 16  cilindrada       float64
 17  hp               float64
 18  combustible      object
 19  transmisión      object
 20  marchas          float64
 21  tracción         object
 22  pais             object
 23  equipamiento     object
 24  tasacion         float64
 25  Continente       object
dtypes: float64(6), int64(3), object(17)
```

	MARCA EN BASE MODELO EN BASE	FECHA	VEHICULO	TIPO	MARCA EN BASE	MODELO HOMOLOGADO	MODELO EN BASE	MARCA	MODELO FAMILIA	SPORT	CHINA	MARCA MODELO HOMOLOGADO	MARCA MODELO FAMILIA	S
0	ABARTH 595 E6D 695 RIVALE 1.4	43617	LIVIANO Y MEDIANO	Vehículo de Pasajeros	ABARTH	595	595 E6D 695 RIVALE 1.4	ABARTH	595	SPORT	OTRO	ABARTH 595	ABARTH 595	
1	ABARTH 595E6D 1.4	43617	LIVIANO Y MEDIANO	Vehículo de Pasajeros	ABARTH	595	595E6D 1.4	ABARTH	595	SPORT	OTRO	ABARTH 595	ABARTH 595	
2	ABARTH ABARTH 595 1.4	43800	Livianos y Medianos	AUTO	ABARTH	595	ABARTH 595 1.4	ABARTH	595	SPORT	OTRO	ABARTH 595	ABARTH 595	
3	ABARTH SIN MODELO	43556	LIVIANO Y MEDIANO	Vehículo de Pasajeros	ABARTH	SIN MODELO	SIN MODELO	ABARTH	SIN MODELO	NO SPORT	OTRO	ABARTH SIN MODELO	ABARTH SIN MODELO	
4	ACURA MDX	44136	LIVIANO Y MEDIANO	STW	ACURA	MDX	MDX	ACURA	MDX	NO SPORT	OTRO	ACURA MDX	ACURA MDX	

4.1. Metodología

(Rodrigo, Random Forest con Python, 2020) El modelo Random Forest aleatorio, es el que aplicamos para nuestro trabajo y consiste en un conjunto de árboles de decisión individuales entrenados en muestras aleatorias tomadas de los datos de entrenamiento originales usando bootstrap. Esto significa que cada árbol se entrena con datos ligeramente diferentes. En cada árbol, las observaciones se distribuyen por las ramas (nodos) que forman la estructura del árbol hasta llegar al nodo final. Las predicciones para nuevas observaciones se obtienen sumando las predicciones de todos los árboles individuales que componen el modelo.

Para comprender cómo funciona el modelo Random Forest, primero debemos comprender los conceptos de ensemble y bagging.

Método de ensemble

Todos los modelos estadísticos y de aprendizaje automático tienen el problema de equilibrar el sesgo y la varianza. El término sesgo se refiere a la distancia promedio entre las predicciones del modelo y los valores reales. Esto refleja la capacidad del modelo para comprender las relaciones del mundo real que existen entre las variables predictoras y de respuesta. Por ejemplo, si una relación sigue un modelo no lineal, el sesgo es alto porque el modelo de regresión lineal no puede modelar la relación correctamente, independientemente de la cantidad de datos disponibles. El término varianza se refiere a la medida en que un modelo varía según los datos utilizados

en el entrenamiento. Idealmente, las pequeñas variaciones en los datos de entrenamiento no deberían cambiar el modelo de manera significativa. Esto se debe a que el modelo memoriza los datos en lugar de aprender la verdadera relación entre el predictor y las variables de respuesta. Por ejemplo, los modelos de árbol con muchos nodos suelen cambiar su estructura con pequeños cambios en los datos de entrenamiento, lo que genera una gran variación.

A medida que aumenta la complejidad del modelo, se vuelve más flexible para adaptarse a las observaciones, lo que reduce el sesgo y mejora la previsibilidad. Sin embargo, una vez que alcanzamos una cierta cantidad de flexibilidad, nos encontramos con el problema del sobreajuste. El modelo se ajusta demasiado bien a los datos de entrenamiento para predecir con precisión nuevas observaciones. El mejor modelo es el que logra el mejor equilibrio de sesgo y varianza. ¿Cómo se controlan las desviaciones y varianzas en los modelos de árbol? En general, los árboles más pequeños (menos ramificados) tienen varianzas más bajas, pero no representan bien las relaciones entre las variables. En otras palabras, el sesgo será mayor. Por el contrario, los árboles grandes se ajustan tan bien a los datos de entrenamiento que tienen poco pero más sesgo. Una forma de evitar esto es establecer un método.

Los métodos de ensemble tienen como objetivo combinar múltiples modelos en un nuevo modelo para lograr un equilibrio de sesgo y varianza, lo que resulta en mejores predicciones que los modelos individuales originales.

Los dos tipos de agregación más utilizados son:
Bagging ajuste varios modelos, cada uno con un subconjunto diferente de los datos de

entrenamiento. En la previsión, todos los modelos que componen la agregación participan en la previsión. El valor final es la media de todas las predicciones (variables continuas) o clases más frecuentes (variables categóricas).

Los modelos Random Forest entran en esta categoría.

Boosting: algunos modelos simples, conocidos como aprendices débiles, se ajustan secuencialmente de modo que cada modelo aprende de los errores del modelo anterior. Al igual que con el empaquetado, tomamos como valor final la media de todas las predicciones (variables continuas) o el tipo más común (variables cualitativas). Los tres métodos de mejora más utilizados son AdaBoost, Slope Boost y Random Gradient Boost. El objetivo final es el mismo, pero existen dos diferencias clave para lograr el mejor equilibrio entre el sesgo y la varianza: ¿Cómo podría reducirse el número total de errores? El error total del modelo se puede descomponer en $bias + varianza + \epsilon$.

La encapsulación utiliza un modelo con muy poco sesgo, pero con mucho sesgo. Agregarlos puede reducir el sesgo sin hacerlo demasiado grande. El impulso utiliza un modelo altamente sesgado con poca variación, y el ajuste secuencial del modelo reduce el sesgo. Por lo tanto, cada estrategia reduce parte del error total. Todo se reduce a cómo incorpora variaciones en su modelo. En encapsulación, cada modelo es diferente de los demás porque cada modelo se entrena en diferentes muestras tomadas usando bootstrap). Con mejoras, el modelo se ajustó secuencialmente, con diferentes ajustes a medida que la importancia (peso) de las observaciones cambiaba de una iteración a otra.

La clave para que los métodos de agregación funcionen mejor que los modelos individuales es que los modelos que los generan sean lo más diversos posible (sus errores no están correlacionados entre sí). Una analogía que refleja este concepto es: Considere un juego simple donde los equipos tienen que responder preguntas sobre varios temas. Un equipo de muchos jugadores, cada uno experto en un tema diferente, tiene más posibilidades de ganar que un equipo de jugadores expertos en un tema o que saben un poco sobre el tema.

5. Resultados

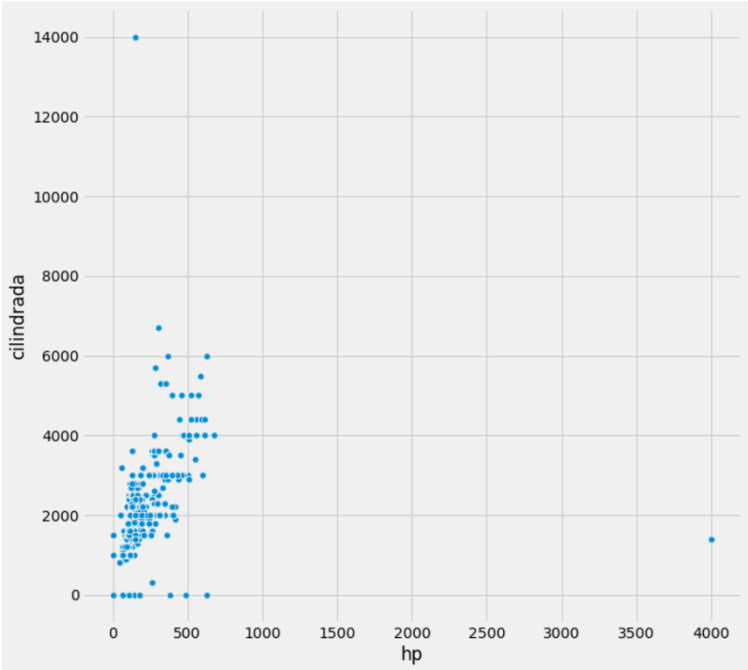
Después de hacer la limpieza de los datos, se procedió a chequear la correlación numérica de la variable ‘hp’ con los otros atributos numéricos arrojando los resultado que se muestran en la tabla 7:

Tabla 7: Correlación numérica de ‘hp’ y otros atributos

```
ano_patente        0.005932
ano_fabricacion    0.017066
puertas            0.040954
cilindrada         0.675703
hp                 1.000000
marchas            0.530667
Name: hp, dtype: float64
```

De acuerdo con la imagen se generó una correlación con la feature ‘hp’ entregando un valor 0.675703 con la cilindrada.

Figura 2: Gráfica de correlación entre ‘hp’ y ‘cilindrada’



Con la ayuda del gráfico – figura 2 (Lili Zhang, 2021) define a segmentación de clientes, “Divide a los clientes en grupos que maximizan las similitudes de los clientes dentro de un grupo y sus diferencias entre grupos según los atributos de los clientes”.

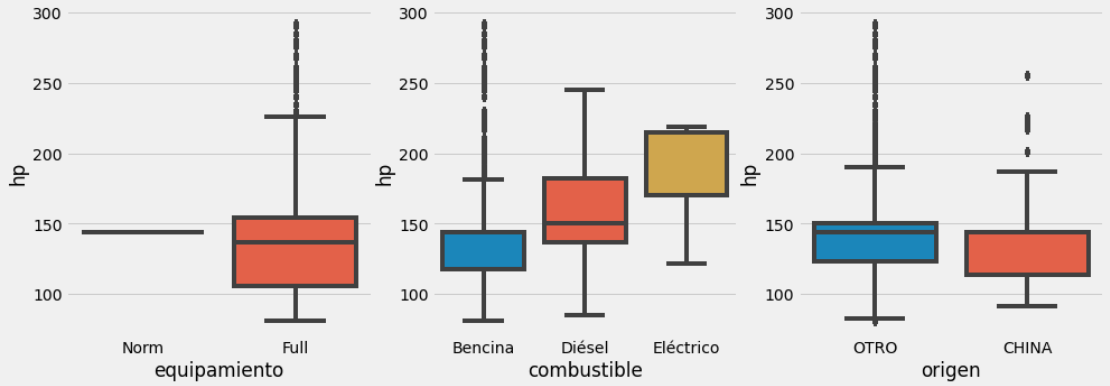
Por lo anterior en nuestro estudio segmentamos los clientes que tuvieran vehículos que cumplieran con los siguientes criterios; una cilindrada mayor a 1.000 cc y menor a 3.000 cc y adicionalmente considerar su HP en un rango mayor a 80 hp y menor a 300 hp. Bajo estas consideraciones estriamos cubriendo el mayor segmento de automóviles que son comprados en nuestro país.

Este resultado nos permitió identificar las primeras seis variables que usaremos para entrenar el modelo y maximizar las similitudes entre los consumidores en estudio.

Obteniendo como resultado una base con 6.955.911 registros.

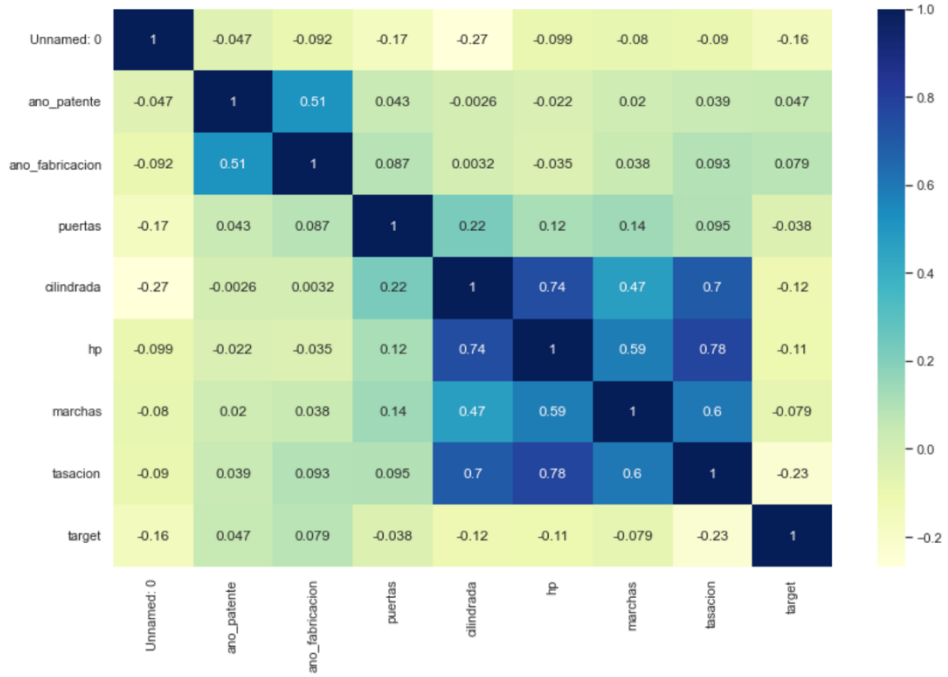
Las gráficas en la figura 3 se acompañan evidencian de los features que consideramos relevantes y pueden influir en el consumidor para la elección de su futuro vehículo; como es el equipamiento, combustible y finalmente el origen.

Figura 3: Diagrama de Caja de relación de ‘hp’ y ‘equipamiento’, ‘combustible’ y ‘origen’



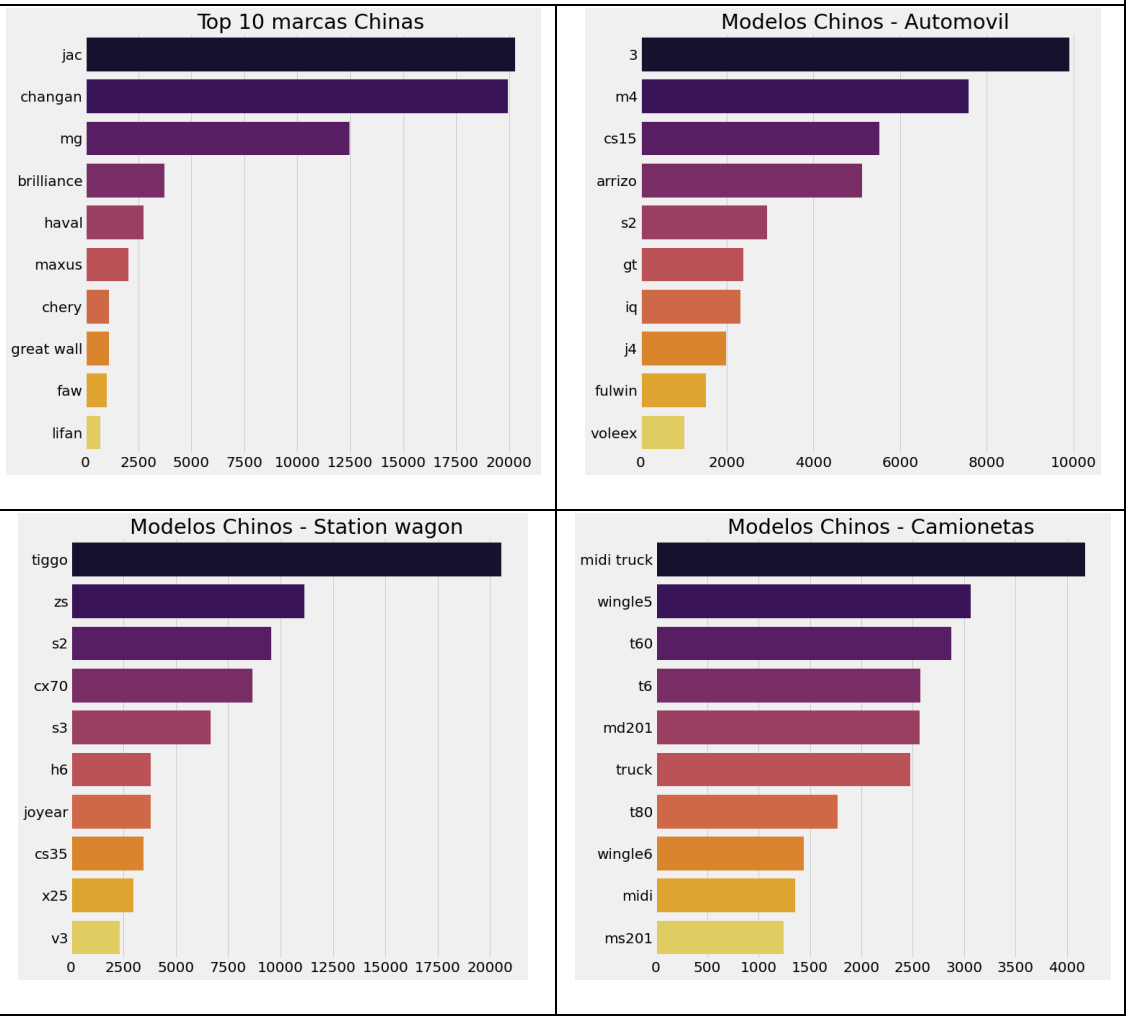
A continuación, en la figura 4, se evidencia el diagrama de correlación el que se utilizó para el primer entrenamiento del modelo.

Figura 4: Diagrama de correlación de variables.



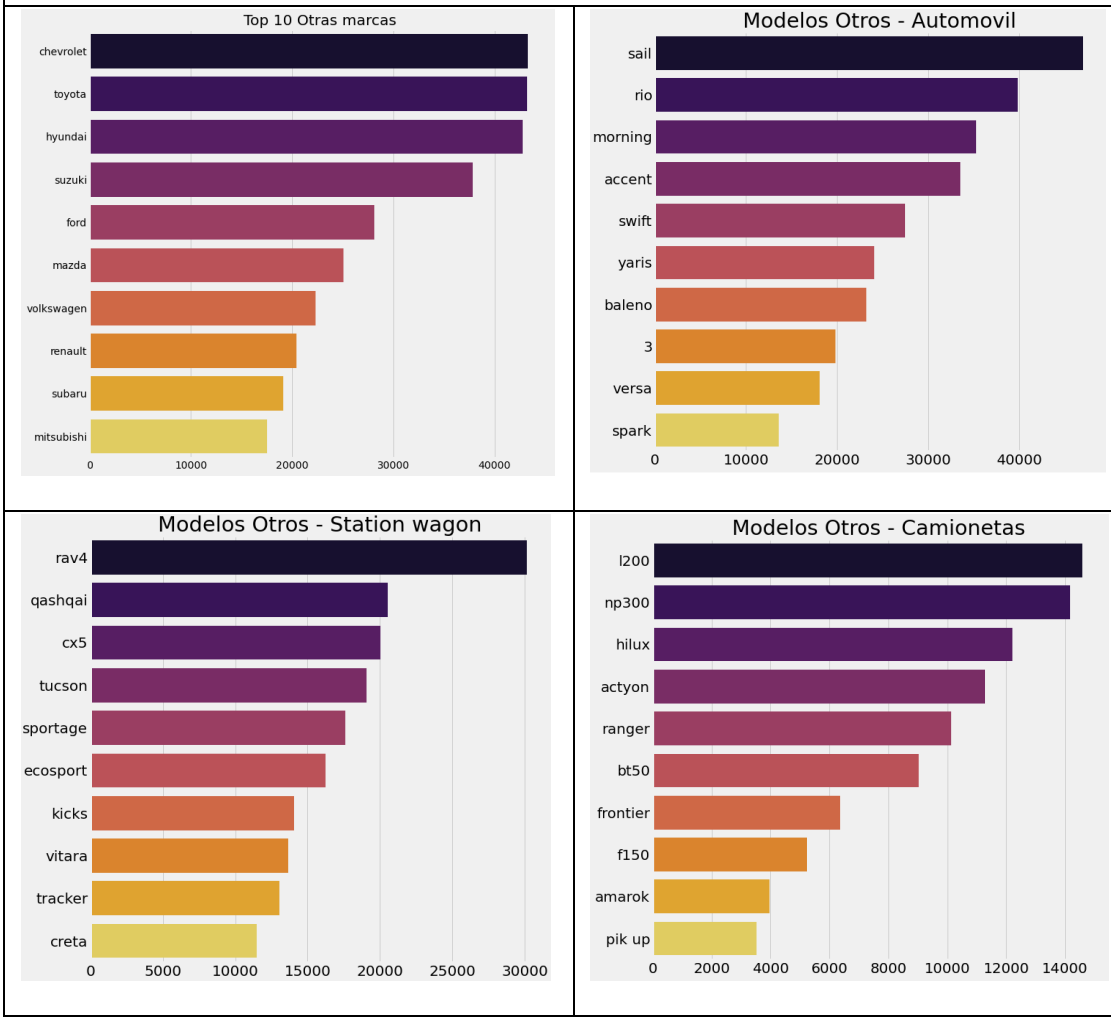
En la figura 5 y figura 6 se muestran graficas para representar los análisis exploratorios de las marcas con mayor presencia, modelos por origen Chino y el de otras marcas.

Figura 5: Análisis exploratorio marcas Chinas



A continuación se muestran las 10 marcas top para los autos de origen Otros, además de los 10 modelos top según el tipo de vehículo, con esto comparamos que los clientes tradicionales compiten con clientes nuevos en la misma línea de auto según su equipamiento, tipo de automóvil.

Figura 6: Análisis exploratorio Otras marcas



A continuación presentamos variables de tipo categoría aplicamos el método *LabelEncoder* a fin de poder categorizar los atributos de la fuente de datos previamente limpiada.

Tabla 8: Extracto de variables categorizadas

	Unnamed: 0	marca_x	patente	rut	ano_patente	tipo	modelo_x	ano_fabricacion	origen	match_1	...	hp	combustible	transmisión	marchas	tracción	
count	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	...	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	6.955911e+06	
mean	1.252576e+07	3.674951e+01	6.145927e+05	6.336423e+05	2.019898e+03	2.062717e+00	6.025419e+03	2.018063e+03	8.757000e-01	3.053241e+02	...	1.399707e+02	1.089173e-01	1.390298e+00	5.760471e+00	3.171937e-01	
std	8.000888e+06	1.672539e+01	3.247431e+05	3.657679e+05	1.208693e+00	1.923206e+00	2.989005e+03	1.293366e+00	3.299236e-01	1.566774e+02	...	3.120315e+01	3.427416e-01	9.203864e-01	7.780779e-01	7.157882e-01	6
min	2.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.016000e+03	0.000000e+00	0.000000e+00	2.016000e+03	0.000000e+00	0.000000e+00	...	8.100000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0
25%	4.616004e+06	2.500000e+01	3.675605e+05	3.170220e+05	2.019000e+03	0.000000e+00	3.083000e+03	2.017000e+03	1.000000e+00	1.550000e+02	...	1.210000e+02	0.000000e+00	0.000000e+00	5.771504e+00	0.000000e+00	8
50%	1.336752e+07	3.800000e+01	6.334790e+05	6.334820e+05	2.020000e+03	2.000000e+00	7.038000e+03	2.018000e+03	1.000000e+00	3.680000e+02	...	1.435323e+02	0.000000e+00	2.000000e+00	5.771504e+00	0.000000e+00	
75%	2.006489e+07	5.300000e+01	8.338280e+05	9.500530e+05	2.021000e+03	4.000000e+00	8.440000e+03	2.019000e+03	1.000000e+00	4.310000e+02	...	1.435323e+02	0.000000e+00	2.000000e+00	6.000000e+00	0.000000e+00	
max	2.583336e+07	5.900000e+01	1.244466e+06	1.267559e+06	2.021000e+03	4.000000e+00	1.036400e+04	2.021000e+03	1.000000e+00	5.260000e+02	...	2.920000e+02	2.000000e+00	2.000000e+00	1.000000e+01	2.000000e+00	2

8 rows × 25 columns

Tabla 9: Nombre de Columnas

```
Index(['Unnamed: 0', 'marca_x', 'patente', 'rut', 'ano_patente', 'tipo',
       'modelo_x', 'ano_fabricacion', 'origen', 'match_1', 'match_liv',
       'key_name', 'marca', 'puertas', 'cilindrada', 'hp', 'combustible',
       'transmisión', 'marchas', 'tracción', 'pais', 'equipamiento',
       'tasacion', 'Continente', 'target'],
      dtype='object')
```

Resultado total **6.955.911** registros de los que por target se distribuyen en (1) – 864.620 y (0) – 6.091.291, en 25 columnas.

Como se muestra en la tabla 10 los valores detallados, es notorio un desbalance en la data por lo que se procedió a implementar el método de sample a fin de equilibrar la data con los clientes que compraron un auto de origen Chino o de otro, la muestra fue de 864.620 registros.

Tabla 10: Dimensión de dataframe después del Sample de los datos

```
Dimensión por tipo de clases con Sample:
Dim class_0: (864620, 25)
Dim class_1: (864620, 25)
```

Como en todo estudio predictivo, no solo es importante ajustar el modelo, sino también cuantificar su capacidad para predecir nuevas observaciones. Para poder hacer esta evaluación, se dividen los datos en dos grupos, un 70% de entrenamiento y 30% de test – tabla 11.

Tabla 11: Variables Features y Labels

```
X=df_under[['ano_fabricacion', 'ano_patente', 'marca', 'puertas', 'cilindrada', 'hp',
               'combustible', 'tasacion','tipo', 'marchas']]  # Features
y=df_under['target']   # Labels
```

Se realiza un modelo inicial utilizando 10 árboles (n_estimators=10) y manteniendo el resto de hiperparámetros con su valor por defecto.

Una vez entrenado el modelo, se evalúa la capacidad predictiva empleando el conjunto de test, entregando como resultado el error de test es 0.3092410018732746.

Ahora es necesario como estamos usando Random Forest tenemos la ventaja de disponer del Out-of-Bag error – figura 7, lo que permite obtener una estimación del error de test sin recurrir a la validación cruzada, que es computacionalmente costosa. En la implementación de RandomForestRegressor, la métrica devuelta como oob_score es el $R2$.

Figura 7: Validación empleando el out-of-bag-error

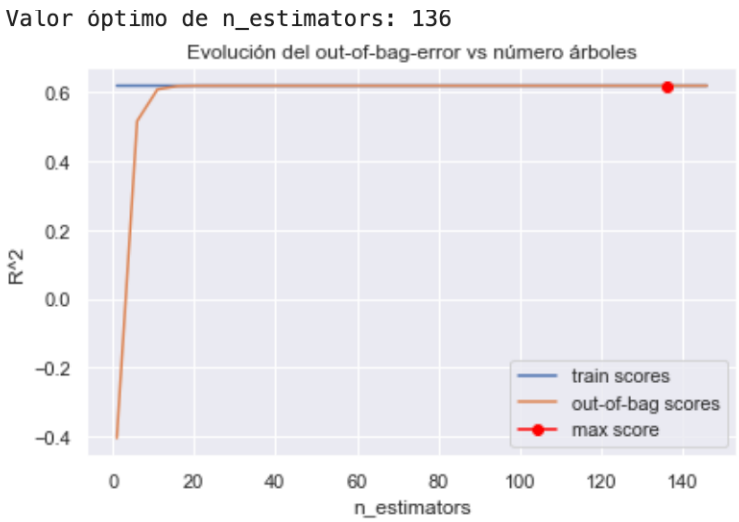
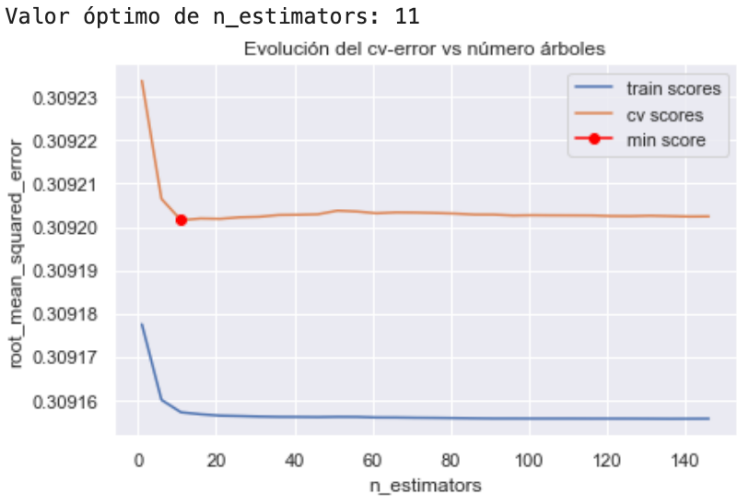


Figura 8: Validación empleando k-cross-validation y neg_root_mean_squared_error



Ambas métricas indican en la figura 8 que, a partir de entre 11 y 136 árboles, el error de validación del modelo se estabiliza.

Ahora aplicamos una validación usando el Max Feature es uno de los hiperparámetros más importantes de random forest, ya que es el que permite controlar cuánto se decorrelacionan los árboles entre sí, para nuestro modelo el valor de max_feature = 10, resultado representado en la figura 9.

Figura 9: Valor óptimo de max_features R^2

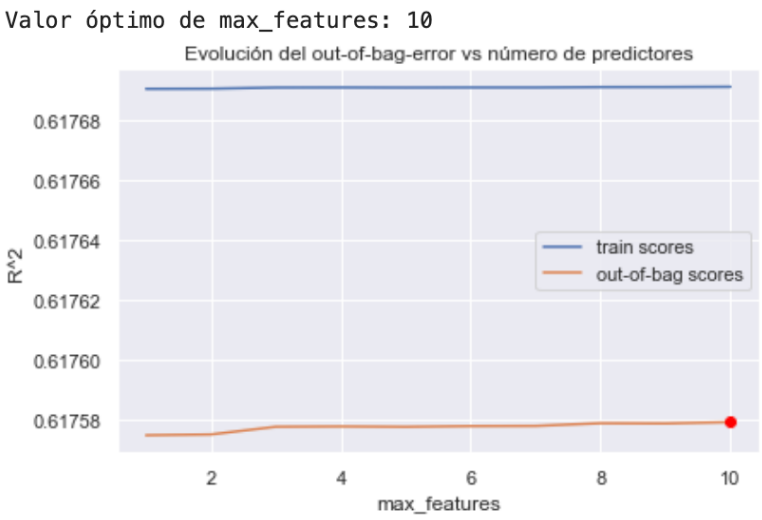
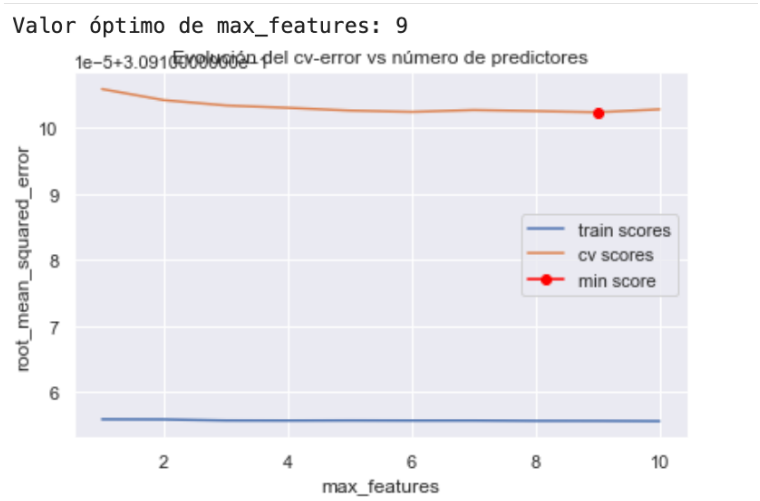


Figura 10: Valor óptimo de max_features root_mean_squared_error



Acorde a las dos métricas utilizadas, el valor óptimo de max_features está entre 9 y 10 – Figura 10.

Se aplicó el Grid Search para entender su impacto en el modelo e identificar rangos de interés, la búsqueda final no debe hacerse de forma secuencial, ya que cada hiperparámetro interacciona con los demás. El resultado obtenido se muestra en la tabla 12.

Tabla 12: Grid search basado en out-of-bag error:

```
Modelo: {'max_depth': None, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': None, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': None, 'max_features': 9, 'n_estimators': 150} ✓
Modelo: {'max_depth': 3, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': 3, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': 3, 'max_features': 9, 'n_estimators': 150} ✓
Modelo: {'max_depth': 10, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': 10, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': 10, 'max_features': 9, 'n_estimators': 150} ✓
Modelo: {'max_depth': 20, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': 20, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': 20, 'max_features': 9, 'n_estimators': 150} ✓
```

	oob_r2	max_depth	max_features	n_estimators
2	0.617580	NaN	9.0	150.0
11	0.617580	20.0	9.0	150.0
10	0.617579	20.0	7.0	150.0
1	0.617579	NaN	7.0	150.0

Loop para ajustar un modelo con cada combinación de hiperparámetros.

No quedamos conformes con los resultados obtenidos por lo que decidimos hacer algunos ajustes en los datos agregando otras variables como la antigüedad del vehículo y el modelo del auto, pero a su vez eliminamos la segmentación previamente realizada, esto nos arrojó un total de 7.267.875 de registros.

Se volvió a calcular la correlación de los atributos con respecto al feature ‘hp’. Los valores obtenidos se ven en la Tabla 13.

Tabla 13: *Correlación numérica de 'hp' y otros atributos*

```
ano_patente        0.005932
ano_fabricacion    0.017066
puertas            0.040954
cilindrada         0.675703
hp                 1.000000
marchas            0.530667
tasacion           0.696490
Name: hp, dtype: float64
```

De acuerdo con la imagen se generó una correlación con el feature 'hp' de los automóviles entregando esta vez un mejor valor 0.696490 con la 'tasacion'.

En la figura 10, se muestra el nuevo diagrama de correlación, el cual mejoro notoriamente los scores de los features.

(Sari Puspita, 2019) La agrupación en clústeres es un método utilizado para crear una serie de datos para formar varios grupos en función de similitudes predeterminadas. La agrupación en clústeres son datos en un clúster que tienen un alto nivel de similitud y datos en diferentes clústeres tienen un bajo nivel de similitud. A continuación, se muestra la Figura 10 y 11 con el clustering del feature 'tasación' versus otros atributos usados como input en el modelo.

Figura 10: Histogramas de Clustering tasacion vs otros atributos

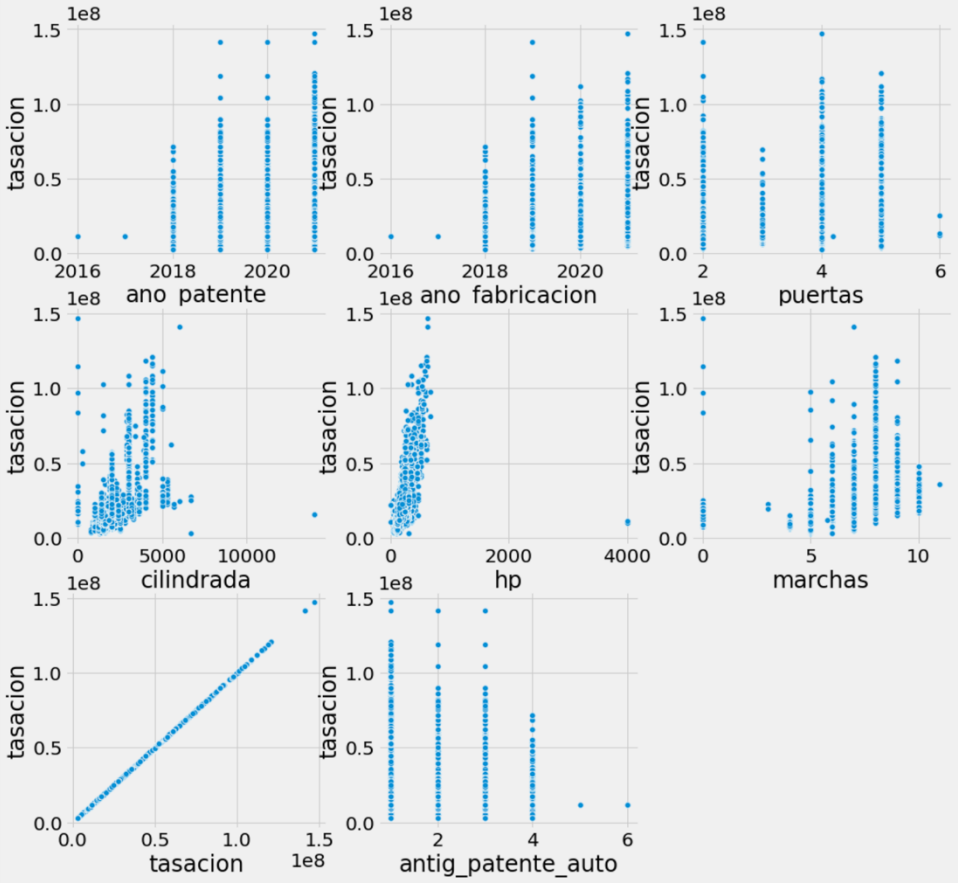
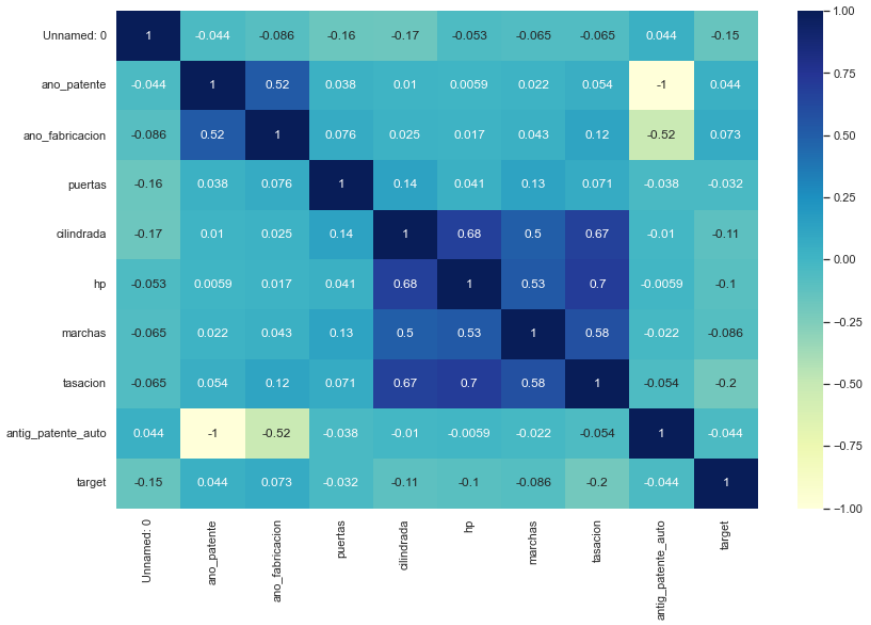


Figura 11: Nuevo Diagrama de Correlación



A continuación, se muestra la distribución por la variable 'target'.

Qué representa el target 0 y 1 = Origen

Resultado total **7.267.875** registros de los que por target se distribuyen en (1) – 871.540 y (0) – 6.396.335, en 26 columnas.

Se aplicó un balance de la data, quedando con las siguientes dimensiones 871.540 en dos clases

Se aplicó nuevamente el Modelo Ramdom Forest, en la tabla 14 se muestran los features utilizados:

Tabla 14: Variables Features y Labels

```
X=df_under[['ano_fabricacion', 'ano_patente', 'marca', 'puertas', 'cilindrada', 'hp',
                'combustible', 'tasacion','tipo', 'marchas','antig_patente_auto', 'modelo_x']]  # Features
y=df_under['target']   # Labels
```

También se evalúa la capacidad predictiva empleando el conjunto de test, entregando como resultado el error de test es 0.007173758116434361.

Los resultados del modelo sin duda empezaron a mejorar a partir de las nuevas variables. Esto es evidente a través de la figura 11.

Figura 12: Validación empleando el out-of-bag-error

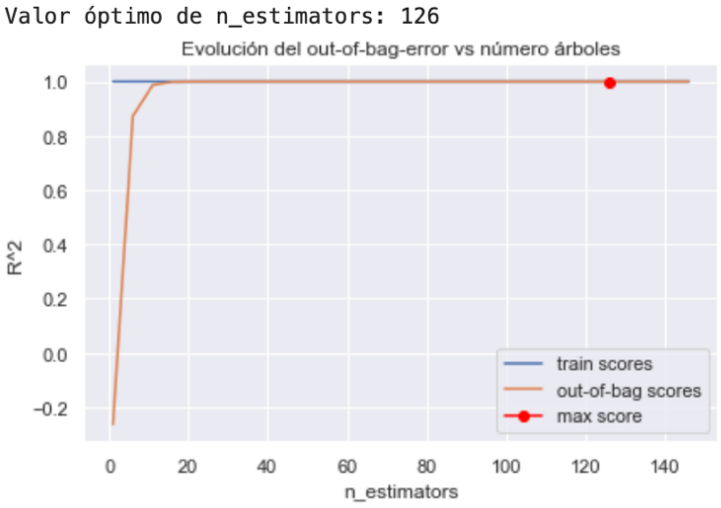
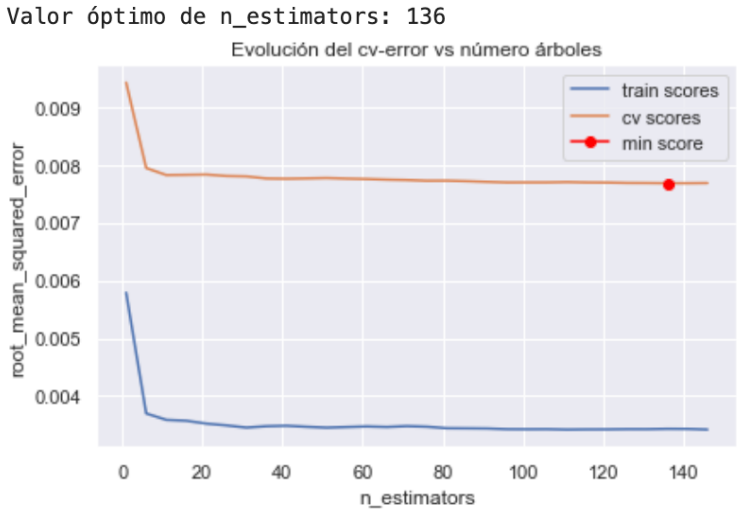


Figura 12: Validación empleando k-cross-validation y neg_root_mean_squared_error



Ambas métricas indican que, a partir de entre 126 y 136 árboles, el error de validación del modelo se estabiliza, a diferencia del entrenamiento inicial el mínimo de score es mucho más cercano en ambas validaciones, referencia de los resultados en la figura 12.

Figura 13: Valor óptimo de max_features R^2

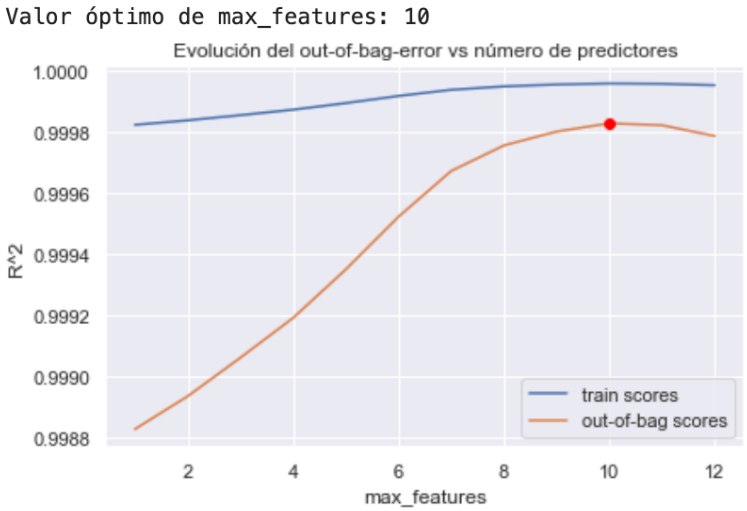
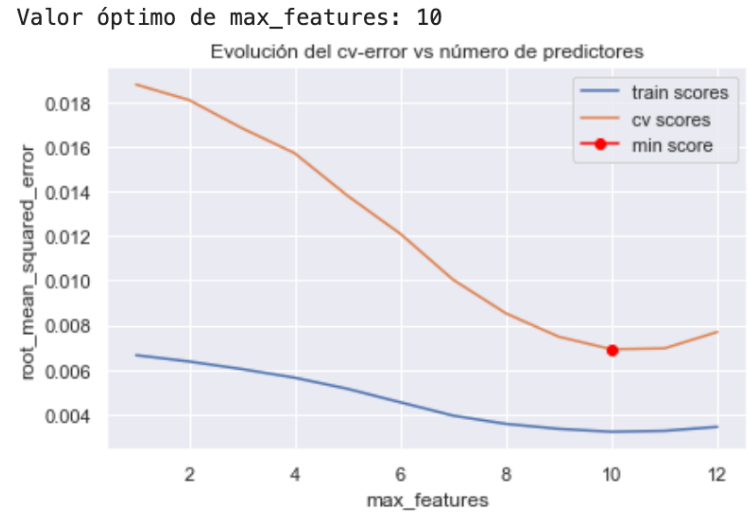


Figura 14: Valor óptimo de max_features root_mean_squared_error



Acorde a las dos métricas utilizadas, el valor óptimo de max_features es 10 para ambas – figura 13 y figura 14.

También se aplicó el Grid Search para entender su impacto en el modelo e identificar rangos de interés, obteniendo los resultados representados en la tabla 15:

Tabla 15: Grid search basado en out-of-bag error:

```
Modelo: {'max_depth': None, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': None, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': None, 'max_features': 9, 'n_estimators': 150} ✓
Modelo: {'max_depth': 3, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': 3, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': 3, 'max_features': 9, 'n_estimators': 150} ✓
Modelo: {'max_depth': 10, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': 10, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': 10, 'max_features': 9, 'n_estimators': 150} ✓
Modelo: {'max_depth': 20, 'max_features': 5, 'n_estimators': 150} ✓
Modelo: {'max_depth': 20, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': 20, 'max_features': 9, 'n_estimators': 150} ✓
```

	oob_r2	max_depth	max_features	n_estimators
2	0.999803	NaN	9.0	150.0
1	0.999668	NaN	7.0	150.0
0	0.999376	NaN	5.0	150.0
11	0.999003	20.0	9.0	150.0

Loop para ajustar un modelo con cada combinación de hiperparámetros

Los mejores hiperparámetros encontrados por las validaciones out-of-bag-error 0,9998029578728778 y 0,9998019578728778 R^2.

6. Conclusiones

Iniciamos este estudio con una hipótesis de conocer cuál era perfil de los clientes que se decidían por comprar autos chinos y no por autos de otras marcas, qué los hacia tomar esta decisión solo un factor económico, innovación, comodidad, confianza, seguridad, quiebres de inventarios de otras marcas o efectos pandémicos, el caso es que pudimos recopilar las distintas fuentes de datos que nos permitieron hacer un análisis exploratorio focalizados en los modelos de autos, accesorios, tipo, tasación entre otros variables derivadas que fueron surgiendo mientras conocíamos los cambios de clientes de auto de otras marcas por autos de marca China.

Encontramos algunas limitaciones como la tasación de los vehículos según modelo, año de fabricación, o también homologar los nombres de las marcas y modelos para poder identificar la base de autos inscritos que teníamos accesos. Este punto hubo que dedicar un tiempo adicional para alcanzar la relación modelo y tasación fiscal.

Avanzando en el *insight* de los clientes y fuimos conociendo los gustos de los clientes y nos permitió identificar la población de clientes que preferían autos de origen chino incluso causando quiebres de inventarios en Derco eran aproximadamente 870.000 clientes.

Con el propósito de predecir la compra de un auto de marca China aplicamos un modelo bajo la metodología de Ramdom Forest donde se entrenó el 70% de la data y se usó el resto para probar la efectividad del modelo, en un principio ejecutamos un modelo con un base segmentado de acuerdo a la cilindrada y hp de los autos comprados desde 2016 al 2020, además de balancear la carga de datos según nuestra variable target y obtuvimos un

score de 0,61 con un error de 0,30. Si bien no fue un mal resultado no era lo esperado, así que se continuo evaluando que otros factores influían en la compra de nuestra cartera de clientes compartidas por Derco; así que decidimos agregar datos de la base de vehículo automotor y crear variables derivadas relacionadas con el año de compra de vehículo para poder probar si efectivamente había un efecto de la pandemia a causa de mayor liquides en el mercado producto de los retiros de AFP y además un precio accesible a otros bolsillos que por primera vez estaban adquiriendo un auto.

Aunque era una posibilidad estas variables teníamos que incluirlas en el modelo y ver los resultados. Corrimos nuevamente el modelo sin segmentación, pero si balanceando la carga de datos según el target y obtuvimos un score de 0,98 con un error de 0,02. Resultado que nos permitió concluir que la segunda selección de variables incluidas en el entrenamiento del modelo condujo a una elección aleatoria atributos predictores que sin duda influyeron en la decorrelación de los árboles y así obteniendo una mayor reducción en la varianza y por consecuencia un modelo con un mejor score.

De acuerdo con el análisis realizado se evidencio que el atributo económico si tiene una influencia directa en la toma de decisiones de los clientes al momento de adquirir un nuevo vehículo. En caso de los vehículos de procedencia China cumplen con este perfil de un menor costo económico y si a eso le sumamos que este tipo de vehículos poseen los adicionales como son la seguridad y el confort (variable que encarece el precio de los modelos). Esto permite que el consumidor no deba elegir entre ambas variables frente a marcas y modelos ya que ambos atributos los posee en un único tipo de automóvil, el Chino.

Se recomienda para futuros ajustes del modelo incluir datos de comuna de residencia del cliente y comuna de compra del auto, estas características podrían influir en el comportamiento de compra de otros consumidores que residen en la misma zona de compra o residencia y como las redes se articulan en el tiempo de los intereses de las personas; sin embargo, es una hipótesis que tendría que evaluarse, si efectivamente tienen relación con el resto de atributos influyentes en el modelo de predicción.

7. Bibliografía

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learnn & TensorFlow.* O'reilly.

McKinney, W. (s.f.). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython.* O'REILLY.

Chollet, F. (s.f.). *Deep Learning with Python.* Manning.

MLOps. (s.f.). *An Overview of the End-to-End Machine Learning Workflow*. Obtenido de MLOps: https://ml-ops.org/content/end-to-end-ml-workflow

Rodrigo, J. A. (01 de Octubre de 2020). *Random Forest con Python*. Obtenido de ciencia de datos: https://www.cienciadedatos.net/documentos/py08_random_forest_python.html