Análisis de robos de autos en Argentina

Período 2018/19

Brum, Luciano - Carpaneto, Agustin - Spiousas, Ignacio

Introducción

En Argentina se reportan cerca de 17 mil robos de automotores por semestre. Aproximadamente un 5% de esos autos son reportados como recuperados. En el presente trabajo se realizó con una base de datos de denuncias de robo y recupero de vehículos automotores de la nación Argentina[1]. En primer lugar, realizamos un análisis exploratorio e intentamos, mediante algoritmos de machine learning, responder dos preguntas: Dado los datos de un auto y de su dueño ¿Será recuperado en caso de ser robado? Y, en caso de ser recupearado, ¿Cuál es el tiempo estimado de recupero?.

Descripción de los datasets

Para llevar adelante el análisis utilizamos el dataset que motivó las preguntas junto con tres datasets adicionales relacionados a patentamientos y flota circulante de automotres(ver anexo). El dataset original consiste en las denuncias de robo o recupero de automotores por mes desde Enero 2018 hasta Septiembre 2019. Fue obtenido de la plataforma de datos abiertos de la Nación y cuenta con las columnas (*features*) que aparecen en la tabla 1. Para poder trabajar con mayor facilidad, unificamos todos los datasets mensuales de robo y recupero.

Tabla 1 - Descripción del dataset que usamos como base para los análisis.

Columna	Descripción
tramite_tipo	Especifica si se trata de una denuncia de robo o recupero
tramite_fecha	Fecha del trámite
fecha_inscripcion_inicial	Fecha de insripicón del automotor
registro_seccional_descripcion	Departamento de inscripción del automotor
registro_seccional_provincia	Provincia de inscripción del automotor
automotor_origen	Origen del automotor (nacional, importado o Protocolo 21)
automotor_anio_modelo	Modelo del automotor
automotor_tipo_descripcion	Tipo del automotor (por ejemplo, sedan)
automotor_marca_descripcion	Marca del automotor
automotor_uso_descripcion	Uso del automotor (privado, público u oficial)
titular_tipo_persona	Indica si el titular es una persona física o jurídica
titular_domicilio_localidad	Domicilio del titular del automotor
titular_domicilio_provincia	Provincia del domicilio del titular del automotor
titular_genero	Género del titular del automotor
titular_anio_nacimiento	Año de nacimiento del titular del automotor
titular_pais_nacimiento	País de nacimiento del titular del automotor

[1] https://datos.gob.ar/dataset/justicia-robos-recuperos-autos

titular_porcentaje_titularidad	Indica el porcentaje de titularidad del denunciante

Análisis de datos exploratorio (EDA)

En primer lugar, como parte del EDA se limpiaron de datos, eliminaron las filas o columnas que tenían valores NaN (según convenga) y las columnas que consideramos que no nos iban a servir ni para visualizar los datos ni para ajustar los modelos de predicción.

Cómo determinamos cuáles autos han sido recuperados

Si bien el dataset tiene denuncias de robo y recupero por separado, y estas están anonimizadas, pudimos determinar qué autos habían sido recuperados (y el tiempo de recupero) buscando aquellas muestras en las que existieran duplicados de todos los datos menos el tipo de trámite (robo o recupero). De este modo, creamos una nueva columna (Recuperado) que tiene valor 0 si no fue recuperado y valor 1 si fue recuperado y para los recuperados se agrega una columna con el tiempo de recupero en días.

Robos y recuperos por mes

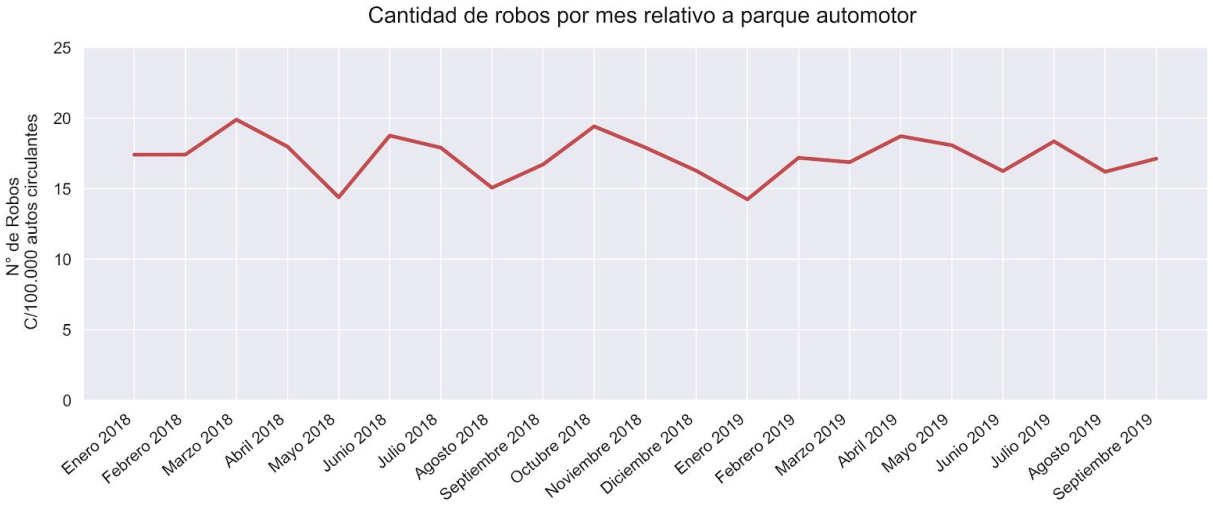


Figura 1. Número de robos por mes relativo al parque automotor circulante en Argentina. El gráfico se muestra expresado cada 100.000 autos circulantes.

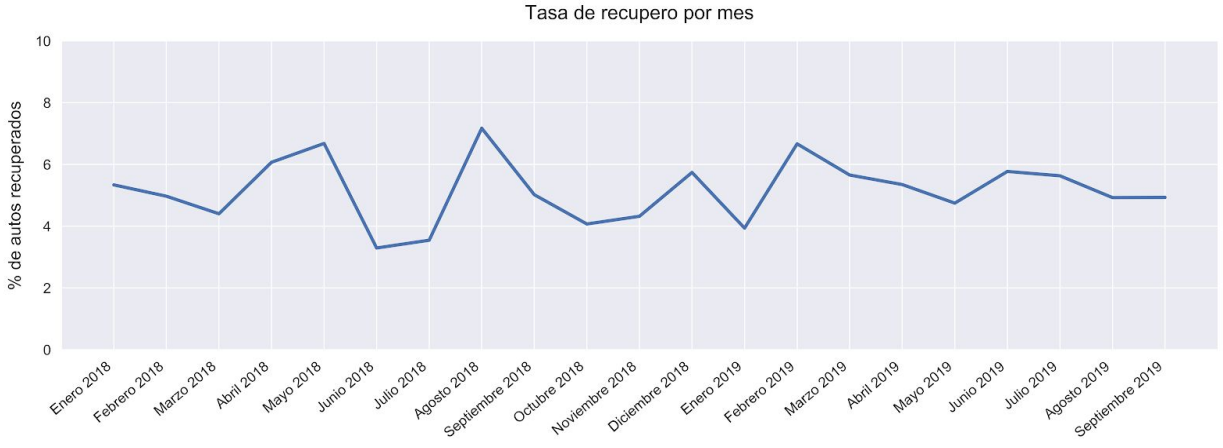


Figura 2. Tasa de recupero por mes. Número de autos recuperados en relación a la cantidad de autos robados por mes.

Se puede observar que, a pesar de tener pequeñas fluctuaciones, tanto la cantidad de autos robados como la tasa de recupero se mantienen relativamente constantes a lo largo de los meses. No se observa una tendencia de las mismas hacia la alza o hacia la baja

Robos y recuperos por día de la semana

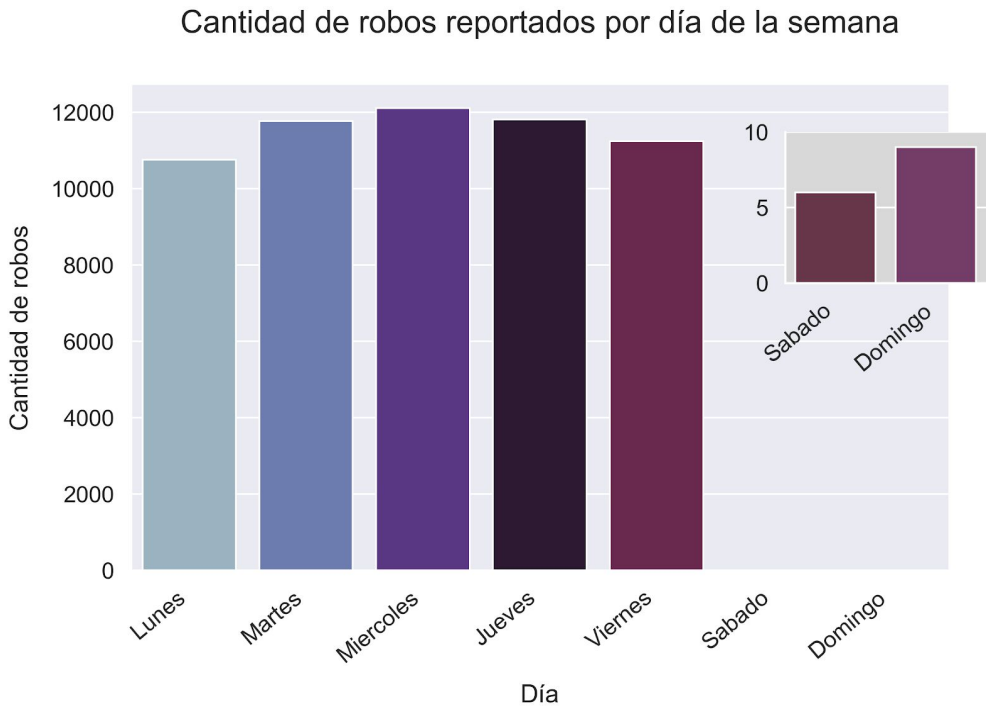


Figura 3. Cantidad de robos reportados por día de la semana

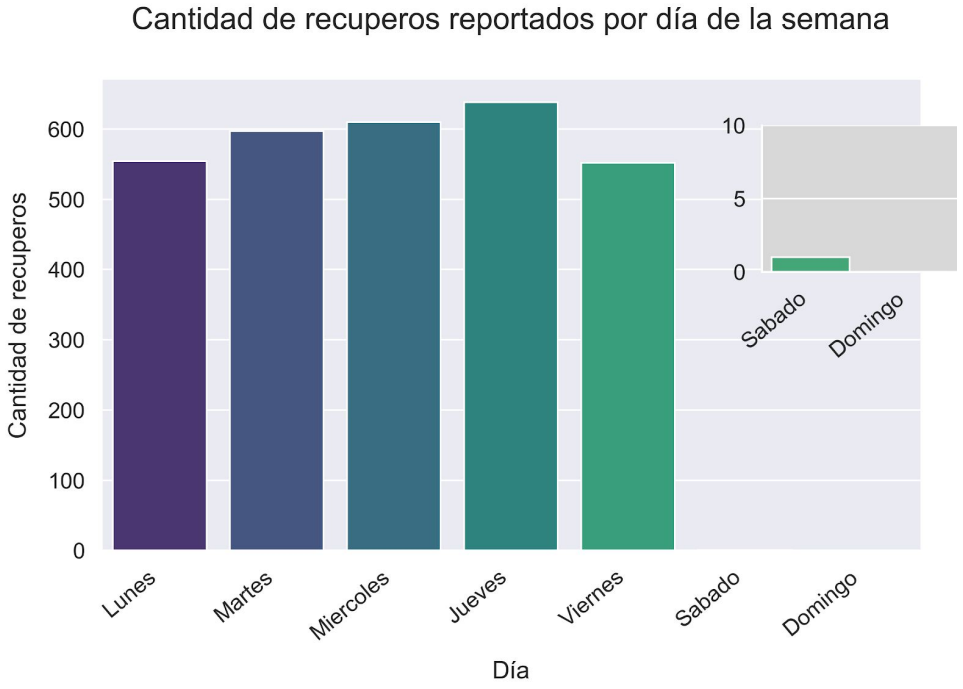


Figura 4. Cantidad de recuperos por día de la semana

El día de la semana que mayor cantidad de trámites de denuncia de robo presenta es el miércoles, mientras que el mayor cantidad de denuncias de recupero es el jueves. En ambos casos se ve que el número de trámites se reduce drásticamente los fines de semana, esto probablemente se deba simplemente a que los trámites no son ingresados al sistema y no a que los robos y recuperos se reduzcan.

Robos y recuperos por marca

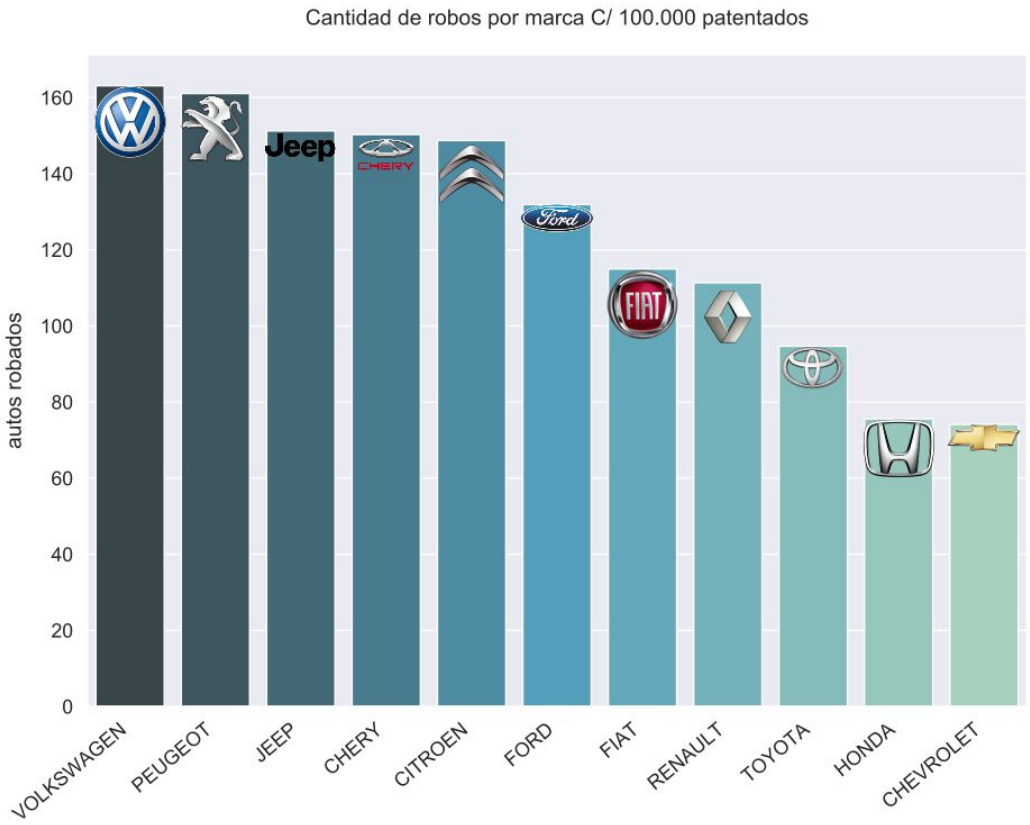


Figura 5. Cantidad de robos reportados por marca (para las diez marcas más robadas)

Para poder realizar el conteo de las marcas más robadas, relativizamos el número de robos reportados para cada una a la cantidad de automotores patentados de cada una en el período 2018/2019. Lo ideal sería hacerlo al parque automotor circulante por marca pero este dato no está disponible.

Robos y recuperos por modelo

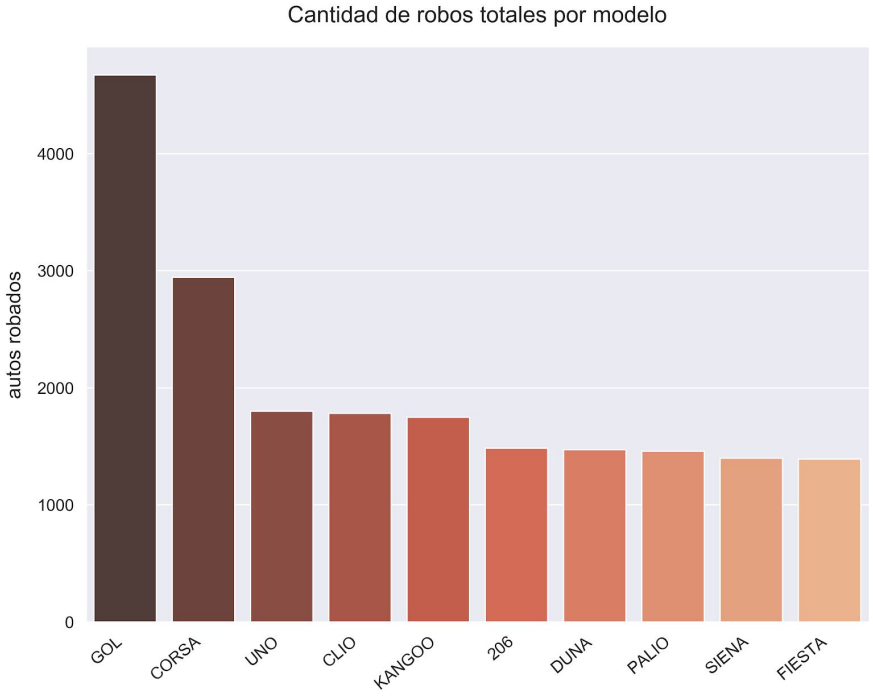


Figura 6. Cantidad de autos robados por modelo en Argentina

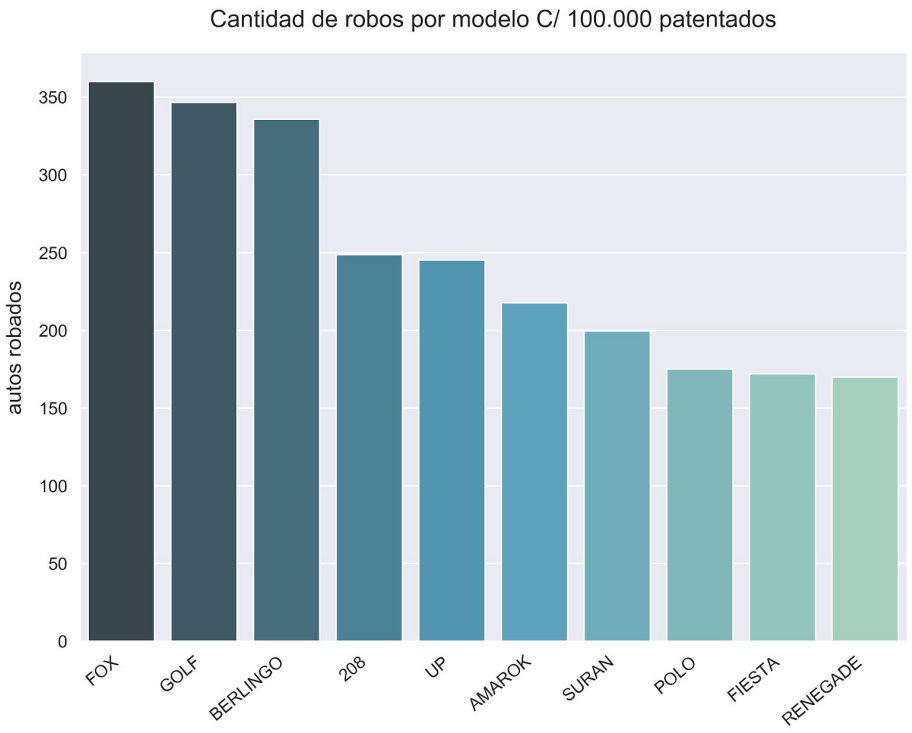


Figura 7. Cantidad de autos robados por modelo cada 100.000 autos de este modelo patentados en el período analizado en Argentina

Realizando el conteo absoluto de robos por modelo arroja un panorama incorrecto ya que un número alto puede deberse simplemente a una mayor cantidad de automotores de dicho modelo circulando. Por ello relativizamos nuevamente la cantidad de robos de cada modelo a la cantidad patentada durante el período 2018/2019. En este caso, además de observar que los “modelos más robados” cambian totalmente, existe el problema de que varios de los modelos que figuran en el conteo absoluto ya no se producen más (ej Duna) y por lo tanto no hay datos contra los que relativizarlo. Por ello, para relativizar solo utilizamos los datos de aquellos autos robados que hayan sido patentados durante el periodo 2018/2019.

Robos y recuperos por edad del titular

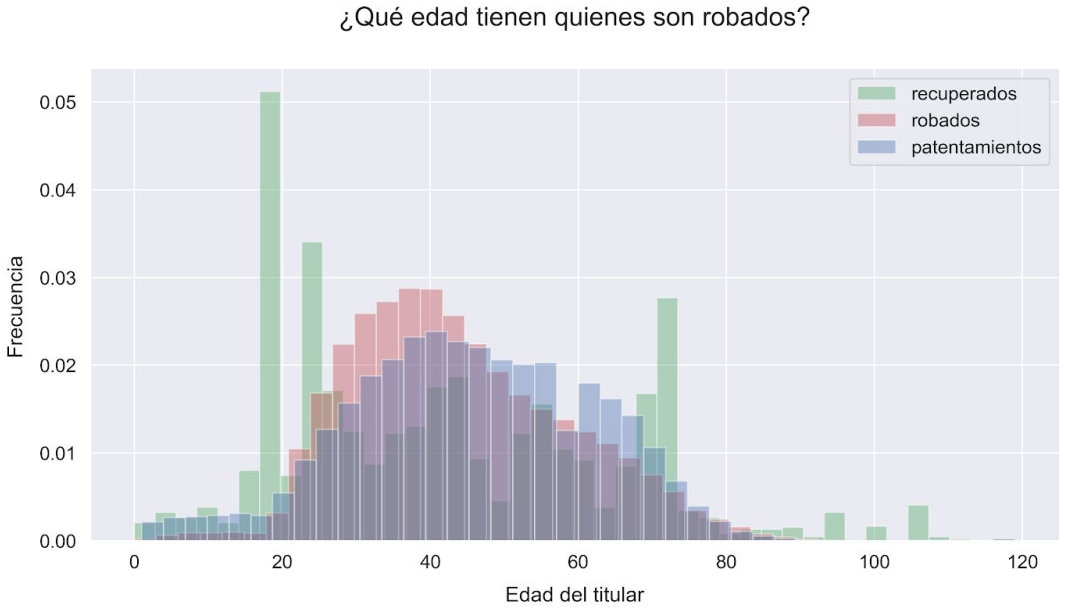


Figura 8. Distribución de edades según patentamientos, robos y recuperos en el período analizado según datos del propietario.

Se puede observar que la distribución de edades de los titulares de autos que fueron denunciados como robados presenta un corrimiento hacia la derecha con respecto a aquella de los titulares de todos los autos patentados, por lo tanto podemos decir que los titulares más jóvenes son aquellos que sufren más robos.

Robos y recuperos por provincia

La distribución de cantidad total de robos por provincia nos hace creer que el sistema de reporte de las denuncias centralizado no está igualmente implementado en todo el país. Esta observación surge de la relación entre robos reportados y habitantes, habiendo provincias con muchos habitantes en las que la cantidad de denuncias es inusualmente baja (por ejemplo Mendoza).
Para reportar la cantidad de robos , relativizamos la cantidad de robos por provincia a la cantidad de automotores circulando en cada una de ellas. De todas formas, como mencionamos previamente hay provincias que tienen muy pocas denuncias de robo, por lo cual los valores de estas provincias se encuentran subestimados.

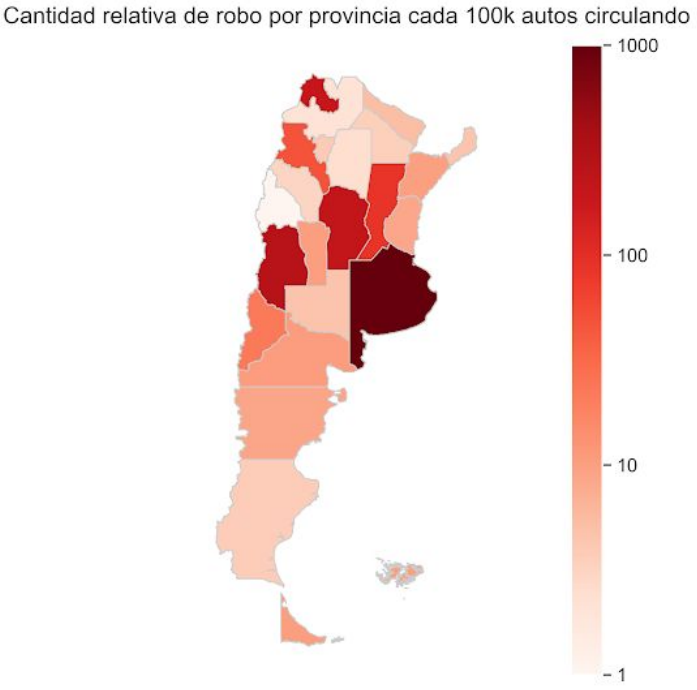


Figura 9. Cantidad de autos robados cada 100mil circulando por provincia.

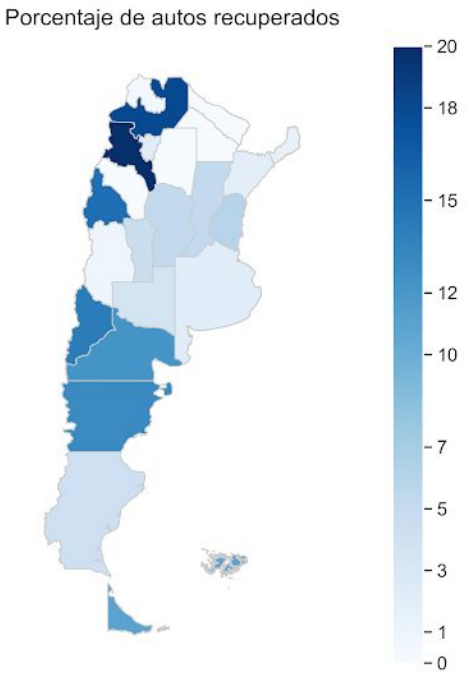


Figura 10. Cantidad de autos recuperados cada 100 robados por provincia.

En el caso de los recuperos, reportamos la tasa por provincia (% de autos robados que fueron recuperados). En este caso debido a la disparidad en el número de denuncias de robo, hay algunas provincias en las cuales la tasa está sobre estimada.

Feature engineering

Para la construcción del dataset final que utilizamos para entrenar a los modelos, eliminamos algunas features categóricas (ej. registro_seccional_descripcion) que poseían demasiadas variables e iban a dar lugar a una disminución del *sample to feature ratio* y a complejizar demasiado la matriz de features de ser utilizadas como dummies.
Generamos nuevas features a partir de aquellas features originales que contenían los datos en formato fecha (“tramite_fecha”, “fecha_inscripcion_inicial”) como día de la semana del robo, mes del año, año de patentamiento, etc.
Además a partir de ciertas features categóricas (“automotor_uso_descripcion”, “titular_genero”, “automotor_origen”) generamos features numéricas (dummies de una sola variable)
Las features finales, que luego fueron seleccionadas o no, para cada modelo fueron:

Tabla 2 - Features finales utilizadas en los modelos:

Columna	Descripción
tramite_tipo	Especifica si se trata de una denuncia de robo o recupero
dia_robo	Día de la semana (Lun: 1 - Dom: 7) del trámite
mes_robo	Mes del año (Ene: 1 - Dice: 12) del trámite
dia_del_anio	Día del año (1-365) del trámite
anio_pat	Año de patentamiento del automotor
automotor_anio_modelo	Año de fabricación del automotor
tit_radicado	Indica si el auto está patentado en la provincia de origen del titular (1-0)

automotor_origen	Origen del automotor (nacional, importado o Protocolo 21)
titular_masculino	Indica si el titular es masculino (1-0)
uso_privado	Indica si el uso del automotor es privado (1-0)
importado	Indica si el automotor es importado (1-0)
persona_fisica	Indica si el titular es una persona física (1-0)
unico_duenio	Indica si el automotor tiene un único dueño (1-0)

La entrada de los datos probablemente se haya realizado a mano, por lo tanto dentro de las features que explicitan el tipo, marca y modelo del automotor, existían entradas que referían a lo mismo pero escritas de maneras dispares (Ej: en marca Volkswagen, Wolskwagen, Folkswagen), para corregir estas diferencias y unificar las variables utilizamos expresiones regulares (regex) y reemplazamos aquellas que estaban ingresadas de manera incorrecta o diferente.

Para las variables “automotor_tipo_descripcion”, “automotor_marca_descripcion”, “automotor_uso_descripcion”, “titular_pais_nacimiento” del dataset original, generamos matrices de dummies a las cuales les quitamos la última columna antes de agregarlas al dataset ya que de dejarlas generaríamos colinearidad entre variables (la ausencia del resto de las variables en la matriz de dummies indica la presencia de la variable que removimos)

Predicción de recuperación de un automotor

Uno de los objetivos del trabajo es crear un modelo que pueda predecir si un auto robado será o no recuperado basado en los features que tenemos. Para esto utilizamos la columna “recuperado” (creada en el EDA) como label, es decir, si un auto fue recuperado o no.

Uno de los principales problemas que tiene nuestro dataset a la hora de predecir si un auto robado será recuperado o no es que las clases (recuperados y no recuperados) están desbalanceadas, es decir, sólo una pequeña proporción de los autos robados fue recuperada (~5%). Creemos que este desbalanceo no es un defecto del dataset sino, al igual que en el ejemplo del fraude bancario, un resultado de la “rareza” de los eventos de recupero (mucho menor que en, por ejemplo, Reino Unido, donde la tasa de recupero es cercana al 45%).

Aumento sintético de las muestras minoritarias

Para lidiar con el desbalanceo de la muestra surgen básicamente dos opciones complementarias: disminuir la cantidad de muestras de la clase mayoritaria (undersampling) y aumentar la cantidad de muestras de la clase minoritaria (oversampling). El undersampling se trata de “tirar” muestras de, en nuestro caso, autos no recuperados, si bien existen varias estrategias para decidir cuáles de esas muestras tirar. Por otro lado, el oversampling es más complejo, ya que se deben crear muestras sintéticas de la clase minoritaria (autos recuperados en nuestro caso).

En este trabajo, utilizamos una combinación de ambas técnicas: disminuimos la cantidad de muestras mayoritarias y aumentamos la cantidad de muestras minoritarias. Para lo primero simplemente quitamos muestras de autos no recuperados al azar mientras que para el segundo utilizamos una técnica denominada SMOTe[2](Chawla et. al), basada en generar muestras sintéticas en el hipersegmento entre dos muestras de la clase minoritaria (utiliza un enfoque similar al KNN).

[2] Del inglés *Synthetic Minority Over-sampling Technique*.

Una vez elegidos los algoritmos tenemos que decidir qué parámetros vamos a utilizar en los mismos. Esto es, cuántas muestras vamos a eliminar en el *undersampleo* y cuántas muestras vamos a crear con el SMOTe. Para esto ajustamos varias versiones de un KNN variando paramétricamente ambas magnitudes: **1-** La cantidad de muestras no recuperadas que conservamos (25% o 10486 muestras, 50% o 20972 muestras y 75% o 31457 muestras); y **2-** La razón entre la categoría mayoritaria y minoritaria (0.5 y 1, es decir incrementar la categoría minoritaria hasta que represente el 50% de la mayoritaria o hasta igualarla en número).

El accuracy en problemas de clasificación es la cantidad de positivos verdaderos sobre todas las clasificaciones posibles. Este indicador es bueno en los casos en los cuales la cantidad de muestras de ambas etiquetas están balanceadas. Esto se debe a que aún clasificando mal aquellos que están en menor proporción el valor de accuracy puede ser alto por el simple hecho de clasificar bien las muestras pertenecientes a la categoría de mayor proporción. Si queremos evitar este inconveniente es mejor utilizar el recall como indicador de buen funcionamiento de nuestro algoritmo predictivo. El recall nos indica cuantos positivos de todo el universo de muestras positivas realmente capturamos. Este indicador no le va a dar tanto peso a los negativos clasificados como positivos cómo sí a los positivos clasificados como negativos.

K-nearest neighbours

Lo primero que hicimos fue ajustar un modelo de KNN con los datos desbalanceados. Lo ajustamos para valores de K de 1 a 10. El modelo con mayor recall fue para K=1. A continuación podemos ver la matriz de confusión del mismo

El recall obtenido con el mejor KNN y el dataset original es de 0.26 mientras que la accuracy es de 0.93. Como es de esperarse en muestras desbalanceadas, el accuracy es muy alto, debido a que la mayoría de las muestras se califican como “no recuperadas”, es decir, cometiendo muchos errores por falsos negativos (645 en este caso). Esperamos que con el balanceo del dataset mejore sustancialmente el recall aunque es esperable también una disminución de la accuracy.

Tabla 3 - matriz de confusión del KNN ajustados con los datos originales (antes de balancear)

	No recuperados predicción	**Recuperados predicción**
No recuperados verdadero	17385	591
Recuperados verdadero	645	232

KNN con los datos balanceados

Luego de ajustar un KNN para cada combinación de parámetro (3 proporciones de subsampleo, 2 ratios de SMOTe y 10 Ks, 60 modelos en total) nos quedamos con la combinación de parámetros que nos dio la mayor recall. Como puede verse en la figura 11, esto ocurre para K=9, un undersampling del 25% (10486 samples no recuperadas) y una proporción de SMOTe de 1 (es decir, con la misma cantidad de muestras recuperadas que no recuperadas). A continuación vemos la matriz de confusión para este modelo.

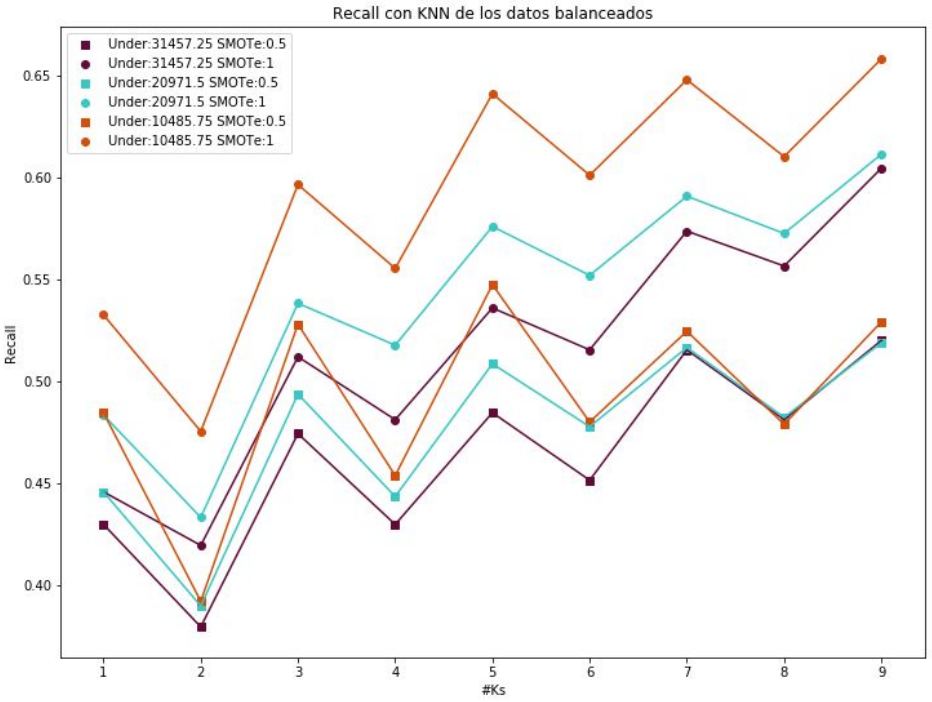


Figura 11 - Recall para distintos valores de under y oversampling en función del número de K en el ajuste de un KNN

Tal como anticipamos, el recall aumentó de 0.26 a 0.66, disminuyendo los falsos negativos de 645 a 300. Por otro lado, la accuracy bajó de 0.93 a 0.71 debido al crecimiento de los falsos positivos (de 591 a 5231).

Como mencionamos anteriormente, en nuestro caso el recall resulta más importante ya que los falsos negativos son más costosos que los falsos positivos. Por esto, consideramos a este modelo como un modelo de performance superior que utilizando los datos sin balancear.

Tabla 4 - matriz de confusión del KNN ajustados con los datos balanceados

	No recuperados predicción	**Recuperados predicción**
No recuperados verdadero	12745	5231
Recuperados verdadero	300	577

Random Forest

A continuación ajustamos un clasificador de Random Forest. Haciendo un GridSearch obtuvimos que los mejores parámetros son:

Tabla 5 - parámetros resultantes del GridSearch para Random Forest

Parámetro	**Valor**
criterion	gini
max_depth	40
max_features	sqrt
n_estimators	50

Tabla 6 - matriz de confusión del Random Forest ajustados con los datos originales (antes de balancear)

	No recuperados predicción	Recuperados predicción
No recuperados verdadero	17907	69
Recuperados verdadero	508	369

Como era de esperarse el Random Forest tiene una mejor performance con el dataset original que el KNN. Este se debe a que los algoritmos de Random Forest, al basarse en comparaciones uno a uno, son menos sensibles a los datos con categorías desbalanceadas.

Random Forest con el dataset balanceado

A continuación ajustamos el mismo modelo que en la sección anterior pero utilizando under y oversampling con los parámetros que mejor ajustaban el KNN (25% de undersampling y proporción de SMOTe 1). Vale la pena mencionar que, utilizar los mejores parámetros (para under- y oversampling) obtenidos iterando sobre el KNN tiene sus limitaciones. Para ajustar un Random Forest óptimo deberíamos repetir la metodología ajustando este tipo de modelos.

Tabla 7 - matriz de confusión del Random Forest ajustados con los datos balanceados

	No recuperados predicción	Recuperados predicción
No recuperados verdadero	17639	337
Recuperados verdadero	414	463

Al igual que en el KNN, al balancear el dataset el Random Forest aumenta su recall (de 0.42 a 0.53) pero, comparado con el KNN, la disminución de la acuracy es mucho menor (de 0.97 a 0.96).

Random Forest con PCA

Como última opción, ajustamos un Random Forest utilizando sólo las primeras cinco componentes del PCA (que explican el 99.98% de la varianza).

Tabla 8 - matriz de confusión del Random Forest ajustados con los datos balanceados y las 5 primeras componentes del PCA.

	No recuperados predicción	Recuperados predicción
No recuperados verdadero	8302	9674
Recuperados verdadero	200	677

Con esta combinación obtenemos el mejor recall (0.77) pero un valor de accuracy bastante por debajo de los obtenidos con el KNN ajustado a los datos balanceados. Es decir, el Random Forest ajustado a las primeras cinco componentes de PCA disminuye los falsos positivos (414 a 200) pero aumenta notablemente los falsos negativos (337 a 9674).

Predicción del tiempo de recupero

Como mencionamos anteriormente, tuvimos la posibilidad de emparejar aquellas denuncias de robo que ocurrieron durante el periodo 2018/2019 con su correspondiente denuncia de recupero gracias a la duplicación de varios datos clave como por ejemplo el año del modelo del automotor, el año del nacimiento del titular, la provincia donde fue radicado el automotor, etc. Cabe destacar que en el dataset existen denuncias de recupero de autos robados antes de este periodo y por lo tanto carecíamos de la información necesaria (denuncia de robo) para utilizarlos. Gracias a este emparejamiento, pudimos determinar el tiempo que transcurrió desde que el automotor fue denunciado como robado hasta que se informó el recupero, estableciendo una nueva variable “días” que tratamos de predecir utilizando varios modelos de regresión.

Antes de comenzar a preparar el dataset para entrenar a los diferentes modelos analizamos cómo es la distribución del número de días realizando un histograma de frecuencias, se puede observar que existen varias muestras para las cuales el valor de días es 0 (fig 12), esto puede estar dado por automotores que fueron realmente robados y recuperados el mismo día o, posiblemente, por titulares que denunciaron el robo y luego se dieron cuenta que el automotor no había sido robado. Teniendo en cuenta los errores de ingreso de datos con los que nos encontramos previamente también puede existir el incorrecto ingreso de la fecha del trámite al sistema. Estas muestras con valor 0 perjudicaban el rendimiento de todos los métodos que utilizamos, por lo que decidimos removerlas para continuar con la predicción.

La distribución de los días posee una media de 125,72 días y una mediana de 99,5 días

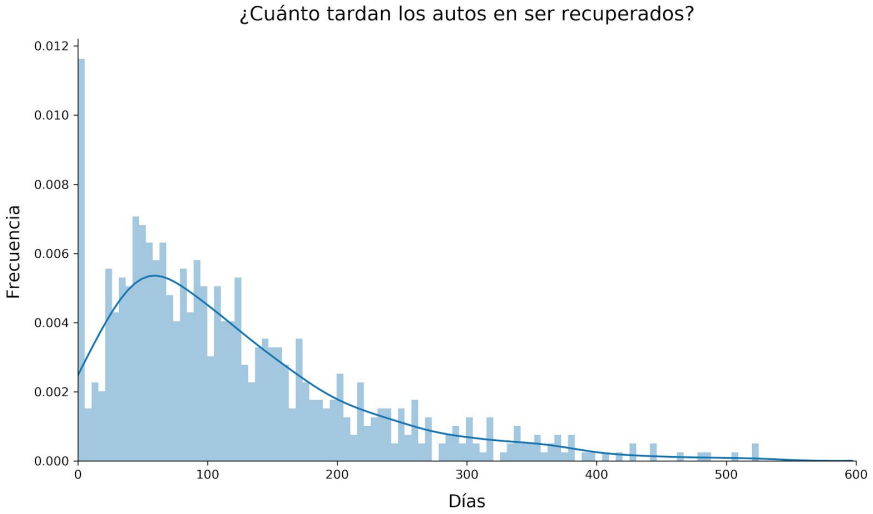


Figura 12 - Distribución de los días que tarda un automotor en ser recuperado

Para decidir qué features utilizar para entrenar a los modelos, primero generamos un pairplot para evaluar de manera rápida si alguna de las features que poseíamos correlacionaba con nuestra variable a predecir, también evaluamos la correlación numérica de todas las variables con la variable a predecir y entre sí (para evaluar la existencia de colinearidad), Luego de generar varias predicciones con los diferentes modelos decidimos quedarnos con la combinación de features que mejores resultados dió. Las features seleccionadas fueron las siguientes:
Automotor_anio_modelo, unico_duenio, mes_robo, titular_radicado, titular_masculino, uso_privado, importado, dummies de modelo y dummies de provincia de registro

Para poder realizar las predicciones de manera correcta, antes de realizar la separación entre set de entrenamiento y set de test y ya que la distribución de la variable a predecir no es normal, generamos bines a partir de esta para poder estratificarla y tener representación de todo el rango de valores en ambos sets. Una vez separados los sets, estandarizamos tanto los datos (matriz X) de entrenamiento como de testeo utilizando un *StandardScaler* ajustado a los datos de entrenamiento, esto es necesario ya que algunas variables tienen valores que representan años y tienen valores en el orden de los miles, mientras que otras representan, por ejemplo, meses y van en una escala del 1 al 12.

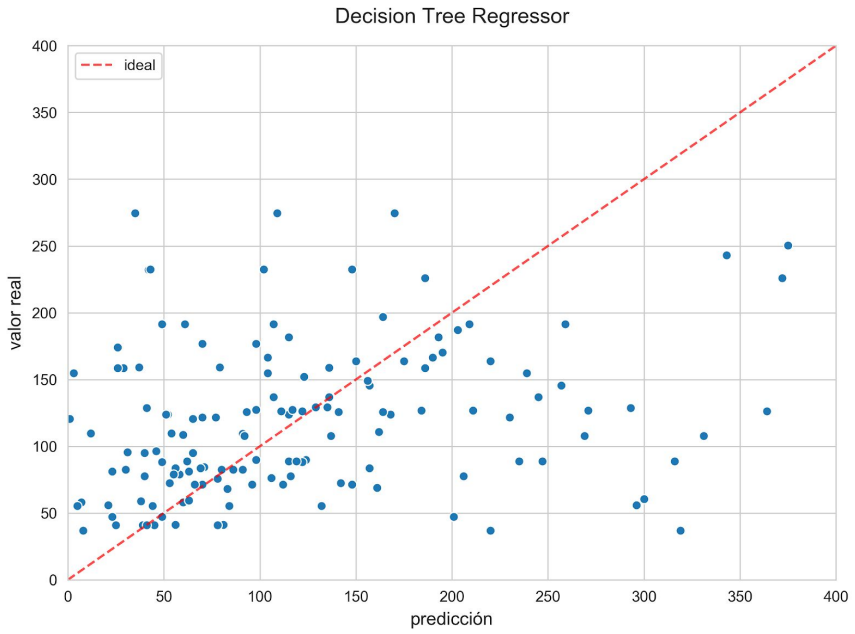
Una vez separados y estandarizados los datos, procedimos a la aplicación de varios modelos, elegimos como métrica para evaluar la performance de los modelos la raíz del error cuadrático medio (RMSE, por sus siglas en inglés) y así poder comparar entre ellos.
Dada la métrica que elegimos y las predicciones que observamos de algunos de los modelos, también generamos scatter plots con los datos del set de testeo vs. las predicciones del modelo para poder evaluar a ojo la calidad de las predicciones (una predicción perfecta arrojaría una línea recta a 45°) y analizamos la distribución de los residuales (diferencia entre la predicción y el valor real), una buena predicción debería tener una distribución normal de los residuales centrada alrededor de 0, si esta está desplazada hacia alguno de los extremos, esto implica que el modelo genera muchos errores a valores bajos (prediciendo valores más altos) o viceversa. Para poder cuantificar numéricamente y comparar entre los modelos, elegimos la mediana de los residuales como parámetro (una buena predicción debería arrojar una mediana más cercana a 0)

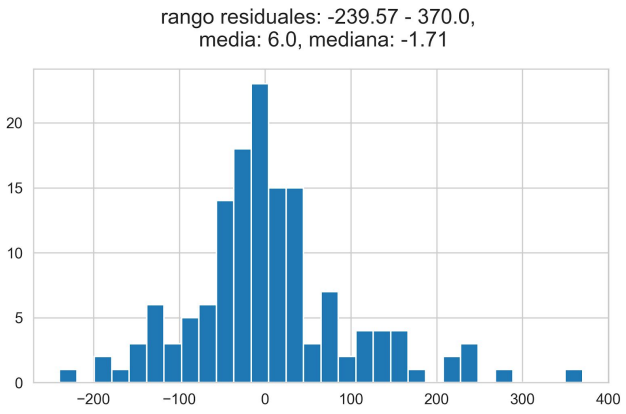
Utilizamos una regresión lineal (LR), K-Nearest Neighbors regression (KNNr), Suport Vector Regression (SVR) y Decision Trees Regression (DTr). Las métricas obtenidas fueron las siguientes:

Tabla 9: Métricas para los diferentes modelos de regresión

Algoritmo	RMSE	Mediana de residuales
LR	$1.3\ e^{16}$	-19.94
KNNr	89.49	-25.4
SVR	90.63	-10.1
DTr	**95.22**	**-1.71**

Para la regresión lineal, observando los scatter plots (no mostrado), vimos que la predicción no estaba tan errada pero predice algunos puntos con valores muy altos lo que genera métricas muy desfavorables.
Tanto en el caso de KNNr como SVR, realizamos un GridSearch con Cross Validation para poder obtener los mejores hiperparámetros. Cuando analizamos las métricas, ambos modelos tuvieron RSME mejores que el resto pero al ver el scatterplot y la mediana de los residuales, en ambos casos se ve que los modelos predijeron valores muy cercanos a la media de la distribución original. Esto se puede deber a que las features son demasiado complejas, no aportan información y, por lo tanto, no explican lo suficiente a la variable independiente, por lo tanto el modelo tiende a minimizar el error ajustando a la media en la mayoría de las predicciones.
Lo mismo sucede utilizando DTr con GridSearch y Cross Validation, pero si ajustamos los hiperparámetros a mano y modificamos solo el valor “min_samples_leaf” (el mínimo número de muestras necesarias para que un nodo se convierta en una hoja), dejando el resto de los hiperparámetros en los valores por default, obtenemos un RMSE aceptable y una buena distribución de residuales. Por lo tanto consideramos que este es el mejor modelo posible.





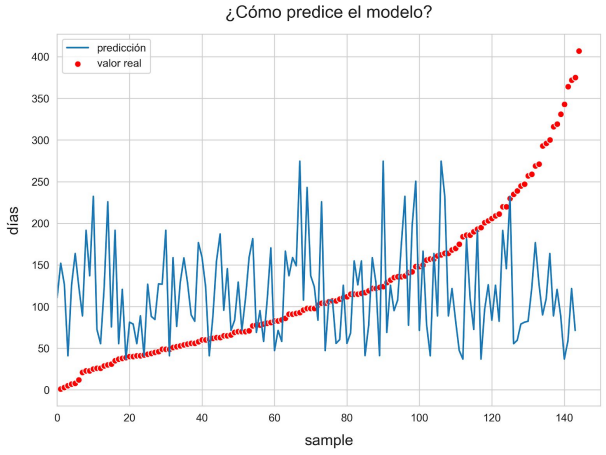


Figura 13 - a) scatter plot de las predicciones del mejor modelo (DTr) vs. los valores de test. b) distribución de los residuales. c) predicción de los valores ordenados.

Cabe destacar que, de todas maneras, aún con el mejor modelo posible, las predicciones de los valores pequeños y grandes está bastante errada, esto se puede ver si ordenamos los datos a predecir de menor a mayor y visualizamos la función de predicción para estos datos. (fig 13 c). Nuevamente esto indica que, a pesar de haber sido seleccionadas, las features no son las ideales ,son demasiado complejas o no son lo suficientemente informativas para poder predecir los días hasta que el automotor sea recuperado.

Con la intención de mejorar las predicciones,utilizamos dos estrategias: por un lado generamos un RandomForest utilizando como base el árbol que mejor predicciones nos dió y no obtuvimos mejores resultados. Por otro lado, utilizamos un método de boosting (AdaBoost). Brevemente, este método de ensamble consiste en la generación de múltiples árboles de decisión de un solo nodo que tendrán mayor o menor peso según cuán bien o mal predigan los valores (en nuestro caso elegimos 5000 estimadores), una vez ajustados estos estimadores se “ensamblará” un estimador global teniendo en cuenta el peso de cada árbol. Además las muestras irán variando el peso asignado (partiendo todas del mismo peso) para que en cada sucesivo árbol tengan mayor peso aquellas muestras que fueron incorrectamente predichas en el árbol anterior (Schapire and Freund). Nuevamente, se vuelve a presentar la misma situación donde el RMSE es aceptable pero el centro la distribución de residuales está muy alejada del 0 y todas las predicciones son similares a la media de la distribución de días.

Finalmente, en pos de obtener mejores resultados, realizamos una reducción de la dimensionalidad de los datos mediante PCA. Calculamos los primeros 10 componentes principales y nos quedamos con los primeros dos ya que en con estos se explica 0.96 de la variabilidad.

Tabla 10: métricas para modelos de regresión

Algoritmo	RMSE	Mediana de residuales
RandomForest	95.22	-1.71
AdaBoost	93.54	-49.8
LR (PCA)	89.36	-23.72
DTr (PCA)	-23.72	-2.33

Nuevamente los modelos volvieron a presentar los mismos problemas, a saber: un RMSE aceptable pero con la mayoría de las predicciones muy cercanas a la media de la distribución original de los días de recupero.

Conclusiones

Como puede observarse durante el presente trabajo, el análisis de datos exploratorio es una herramienta fundamental a la hora de evaluar la implicancia de diferentes variables en un problema dado. Sumado a esto, este análisis previo permite luego ir más allá para intentar aprender de los datos disponibles y poder contestar preguntas a futuro, siempre moviéndonos dentro de los límites de la probabilidad estadística.
El dataset original de robos y recuperos que utilizamos en el trabajo demostró, luego de un análisis exhaustivo, ser rico en relación a la información que se puede obtener de él aunque no sin antes utilizar también datasets complementarios. En primer lugar, el dataset original presenta problemas importantes de estandarización en la toma de datos. Esto puede verse claramente en las líneas de código regex que tuvimos que utilizar para normalizar los nombres de las marcas y modelos como así también los tipos de vehículo. Otro problema a la hora de adquirir información a partir del dataset es que sólo tiene registros a partir del 2018 reduciendo la cantidad de muestras considerablemente ya que existen automotores recuperados en este período que no fueron robados en el mismo, por lo cual esta información se pierde.

Comenzando con el EDA, pudimos observar cierta periodicidad en los robos de automotores (período de 3 meses) aunque no podemos atribuirlo a ningún factor en particular. A nivel semanal, no se ven diferencias a lo largo de la semana entre los robos y los recuperos, ni entre los días de cada uno, salvo por el fin de semana. Aquí ambos caen considerablemente aunque lo más importante es destacar la diferencia entre recuperos casi nulos con número considerable de robos. Un análisis más profundo en relación a los autos robados arrojó ciertas marcas y modelos más robados que otros con una diferencia notable, tanto viéndolo en relación al parque automotor existente previo y en conjunto con el período analizado como también en relación a los patentados en este período. Sumado a esto, si bien el patentamiento aumenta entre los 30 y 40 años, también lo hace el robo con lo cual podríamos estar ante una correlación interesante. Por último, la combinación del mapa de robos y el de recuperos nos da un panorama más que interesante. Las provincias donde más se roban autos es dónde menos se recupera, o por lo menos se sostiene esta tendencia. Ejemplos claros, son Río Negro, Salta y Provincia de Buenos Aires.

Posteriormente, planteamos la generación de dos algoritmos predictivos utilizando los datos limpios. En el primero se buscó ser capaces de predecir si un automotor podría ser recuperado o no. Luego de recurrir a diferentes estrategias de balance de datos y explorar diferentes modelos, pudimos llegar a un modelo basado en KNN con un recall de 0,66 y un acurracy de 0,71. Si bien nos índices son bajos, la creación de

del pipeline de este modelo hace que en los próximos años, teniendo un dataset más amplio, podamos mejorar estas métricas. También teniendo hardware disponible podríamos implementar redes neuronales para tratar de obtener un mejor recall aún con el desbalance inicial. El segundo modelo se planteó para poder predecir, en caso de estimar la recuperación del automotor, en cuantos días ocurriría esto. Luego de implementar diversos modelos y estrategias de mejora no se llegó a solucionar el problema de que aún el mejor modelo predijera valores cercanos a la mediana de días de recupero del dataset. Esto es claramente un problema de complejidad de datos sumado a un bajo número de muestras totales. Creemos que ambos algoritmos y modelos son la base para generar pares mejores utilizando más hardware y combinaciones algorítmicas más complejas.

En resumen, el presente trabajo sienta las bases para generar un algoritmo predictivo capaz de identificar la probabilidad de recupero de un automotor robado utilizando sus datos y los del dueño, y un predictor de tiempo para el mismo. Esto posibilitará la tasación diferencial de seguros de automotores como así también la ejecución más rápida del proceso administrativo una vez dado el robo. Esto podría modificar sustancialmente la industria de aseguradoras de riesgo en nuestro país.

En términos de perspectivas a futuro, se encuentra disponible también el dataset de automotores patentados con detalle individual en el período Ene 18 - Sept 19 y este posee las mismas features que el dataset de robos que utilizamos en el presente trabajo. Teniendo en cuenta esto, se puede utilizar el mismo pipeline para unificar ambos datasets, poder parear aquellos autos que fueron robados y así poder entrenar un modelo que prediga si un automotor será robado o no.

Bibliografía:

- Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Int. Res. 16, 1 (June 2002), 321-357.

- Jaree Thongkam, Guandong Xu, Yanchun Zhang, and Fuchun Huang. 2008. Breast cancer survivability via AdaBoost algorithms. In Proceedings of the second Australasian workshop on Health data and knowledge management - Volume 80 (HDKM '08), James R. Warren, Ping Yu, John Yearwood, and Jon D. Patrick (Eds.), Vol. 80. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 55-64.

- Robert E. Schapire and Yoav Freund. 2012. Boosting: Foundations and Algorithms. The MIT Press.

- Performance Metrics for Classification problems in Machine Learning. Mohammed Sunasra (https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b)
- Dealing with Imbalanced Data - A guide to effectively handling imbalanced datasets in Python. Tara Boyle (https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18)

ANEXO: datasets utilizados

PRINCIPAL:

- Robos y recuperos de autos - Ministerio de Justicia y Derechos Humanos. Subsecretaría de Asuntos Registrales. Dirección Nacional de Registros Nacionales de la Propiedad Automotor y Créditos Prendarios.

  https://datos.gob.ar/dataset/justicia-robos-recuperos-autos

ACCESORIOS:

- Inscripciones iniciales de autos - Dirección Nacional de Registros Nacionales de la Propiedad Automotor y Créditos Prendarios.

  https://datos.gob.ar/dataset/justicia-inscripciones-iniciales-autos

- Estadística Anual de Parque Activo (en condiciones registrales para circular) - Registros de la propiedad automotor

  https://www.dnrpa.gov.ar/portal_dnrpa/estadisticas/rrss_tramites/tram_parque.php?anio=2019&origen=portal_dnrpa

- Estadística Anual de Inscripciones Iniciales Nacionales e Importadas por Provincia - Registros de la propiedad automotor

  https://www.dnrpa.gov.ar/portal_dnrpa/estadisticas/rrss_tramites/tram_prov.php?origen=portal_dnrpa&tipo_consulta=inscripciones