



Data Science

Comisión 29825

Predicción de precios de automóviles usados

Integrantes:

- Diego Pokorski
- Edgar García Ramírez

Tutor

- Néstor Jesus Ramírez Reyes



Abstract

Actualmente el uso del automóvil se ha vuelto importante para la vida cotidiana y laboral; por tal motivo se requiere conocer en qué momento y en qué circunstancias es viable comprar un automóvil para cuidar la salud financiera de las personas.

Precisamente hacia esta directriz está encaminado el presente estudio. El estudio muestra el comportamiento de los precios a lo largo de los últimos años, considerando factores como: modelo, fabricante, año, etc. para conocer a fondo la toma de decisiones de este caso de estudio.

En este proyecto se tratará de predecir el precio del próximo año 2023 en base a los datos obtenidos en el dataset.

Es importante mencionar que los datos obtenidos del dataset son de Estados Unidos de América.



Objetivo

Diseñar un modelo de aprendizaje supervisado que permita predecir mediante métodos de regresión el precio de un automóvil en el 2023.

Contexto Comercial

Actualmente el uso del automóvil se ha vuelto importante para la vida cotidiana y laboral; por tal motivo se requiere conocer en qué momento y en qué circunstancias es viable comprar un automóvil para cuidar la salud financiera de las personas.

Precisamente hacia esta directriz está encaminado el presente estudio. El estudio muestra el comportamiento de los precios a lo largo de los últimos años, considerando factores como: modelo, fabricante, año, etc. para conocer a fondo la toma de decisiones de este caso de estudio.



Data Acquisition

Los datos adquiridos son recopilación tipo **Third-Party Data** porque provienen de un website llamada kaggle, es una fuente de datos externa.

Dataset - Used Cars

- enlace: <https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data>
- columnas: 25
- renglones: 426,881
- tamaño: 1.45Gb



Tipo de columnas - Diccionario de datos

id: [INT] auto increment

url: [STRING] URL del website

region: [STRING] Región de la venta

region_url: [STRING] Website de la venta

price: [INT] Precio del automóvil usado

year: [INT] Año de la venta

manufacturer: [STRING] Fabricante

model: [STRING] Modelo

condition: [STRING] Condición: Excellent, fair, good and new

cylinders: [STRING] Números de cilindros

fuel: [STRING] Gasolina: Diesel, electric, gas and hybrid

odometer: [INT] Odometro

title_status: [STRING] Estatus del automóvil: Clean, lien, missing, part only

transmission: [STRING] Transmisión: Automatic or manual

VIN: [STRING] VIN

drive: [STRING] 4wd, fwd, rwd

size: [STRING] Tamaño: Compact, Full-size, mid-size, sub-compact

type: [STRING] Tipo: bus, pickup, van, etc

paint_color: [STRING] Color

image_url: [STRING] URL de la imagen

description: [STRING] Descripción

country: [STRING] País

state: [STRING] Estado

lat: [STRING] Latitud

long: [STRING] Longitud

posting_date: [STRING] Fecha de registro

Exploratory Data

	id	url	region	region_url	price	year	manufacturer	model	condition	cylinders	...	size	type	paint_color	image_url	description	county	state	lat	long	posting_date	
0	7222695916	https://prescott.craigslist.org/cto/d/prescott...	prescott	https://prescott.craigslist.org	6000	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	az	NaN	NaN	NaN
1	7218891961	https://fayar.craigslist.org/ctd/d/bentonville...	fayetteville	https://fayar.craigslist.org	11900	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	ar	NaN	NaN	NaN
2	7221797935	https://keys.craigslist.org/cto/d/summerland-k...	florida keys	https://keys.craigslist.org	21000	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	fl	NaN	NaN	NaN
3	7222270760	https://worcester.craigslist.org/cto/d/west-br...	worcester / central MA	https://worcester.craigslist.org	1500	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	ma	NaN	NaN	NaN
4	7210384030	https://greensboro.craigslist.org/cto/d/trinit...	greensboro	https://greensboro.craigslist.org	4900	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	nc	NaN	NaN	NaN
5 rows × 26 columns																						

Devuelve información
(número de filas, número
de columnas, índices, tipo
de las columnas y
memoria usado) sobre el
DataFrame df.

```
>>> df.info()
```

```
id                0
url               0
region            0
region_url        0
price             0
year             1205
manufacturer      17646
model             5277
condition         174104
cylinders         177678
fuel              3013
odometer          4400
title_status      8242
transmission      2556
VIN              161042
drive            130567
size             306361
type             92858
paint_color      130203
image_url         68
description        70
county           426880
state             0
lat               6549
long              6549
posting_date      68
dtype: int64
```

Mostrar la suma de las
columnas null or NA del
dataset.

```
>>> df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 426880 entries, 0 to 426879
Data columns (total 26 columns):
#   Column              Non-Null Count  Dtype
---  -
0   id                  426880 non-null  int64
1   url                 426880 non-null  object
2   region              426880 non-null  object
3   region_url          426880 non-null  object
4   price               426880 non-null  int64
5   year                425675 non-null  float64
6   manufacturer        409234 non-null  object
7   model               421603 non-null  object
8   condition           252776 non-null  object
9   cylinders            249202 non-null  object
10  fuel                423867 non-null  object
11  odometer            422480 non-null  float64
12  title_status        418638 non-null  object
13  transmission        424324 non-null  object
14  VIN                 265838 non-null  object
15  drive               296313 non-null  object
16  size                120519 non-null  object
17  type                334022 non-null  object
18  paint_color         296677 non-null  object
19  image_url           426812 non-null  object
20  description          426810 non-null  object
21  county              0 non-null       float64
22  state               426880 non-null  object
23  lat                 420331 non-null  float64
24  long                420331 non-null  float64
25  posting_date        426812 non-null  object
dtypes: float64(5), int64(2), object(19)
memory usage: 84.7+ MB
```

Devuelve una tupla con
el número de filas y
columnas del DataFrame
df.

```
>>> df.shape
```

```
(426880, 26)
```

Contiene 426,880 filas y
26 columnas el dataset

Data Wrangling

Validar si existen datos duplicados.

```
>>> df.duplicated().value_counts()
```

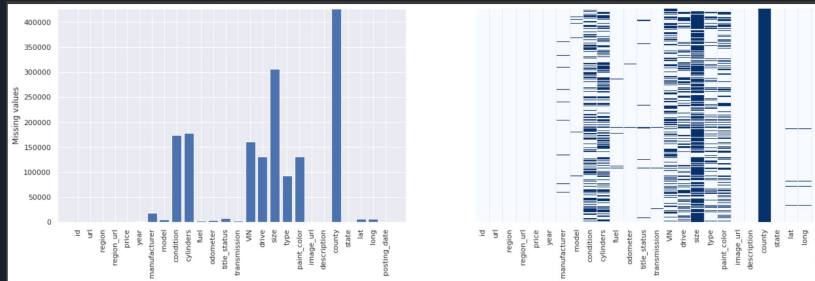
```
False      426880  
dtype: int64
```

Determinamos que no existen renglones o registros duplicados.

Mostrar datos con valores NAN (missing values).


```
>>> df.isna().sum().to_frame()
```

Mostrar en forma gráfica los missing values



Se validó cuántos valores nulos existen en el dataset con `isna()` y `isnull()`. Podemos observar que el campo 'price' no contiene valores nulos, lo que nos indica que tiene valores consistentes. Se eliminarán los registros que tengan NAN en el campo Year debido a que este campo debería de estar con información para que sea útil para el análisis. Podemos observar que las columnas con mayor missing Values son: condition cylinders, VIN, drive, size, type, paint_color y country. La única columna que se eliminará porque se encuentra vacía es: county.

id	0
url	0
region	0
region_url	0
price	0
year	1205
manufacturer	17646
model	5277
condition	174104
cylinders	177678
fuel	3013
odometer	4400
title_status	8242
transmission	2556
VIN	161042
drive	130567
size	306361
type	92858
paint_color	130203
image_url	68
description	70
county	426880
state	0
lat	6549
long	6549
posting_date	68



Dependiente

- price: Cuantitativa continua

Independientes

- year: Cuantitativa discreta
- manufacturer: Cualitativa nominal
- model: Cualitativa nominal
- condition: Cualitativa ordinal
- cylinders: Cualitativa nominal
- fuel: Cualitativa nominal
- odometer: Cuantitativa continua
- title_status: Cualitativa nominal
- transmission: Cualitativa nominal
- drive: Cualitativa nominal
- size: Cualitativa nominal
- type: Cualitativa nominal
- paint_color: Cualitativa nominal
- county: Cualitativa nominal
- state: Cualitativa nominal

Otro

- id
- url
- region
- region_url
- VIN
- image_url
- description
- lat
- long
- posting_date



Análisis exploratorio de datos (EDA)

El resultado final se tiene un dataset con 14 variables y 263,000 registros para el entrenamiento de algoritmos.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 263000 entries, 27 to 426879
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   price           263000 non-null  int64
1   year            263000 non-null  float64
2   manufacturer     263000 non-null  object
3   model           263000 non-null  object
4   condition        263000 non-null  object
5   cylinders        263000 non-null  object
6   fuel            263000 non-null  object
7   odometer        263000 non-null  float64
8   title_status    263000 non-null  object
9   transmission     263000 non-null  object
10  drive           263000 non-null  object
11  type            263000 non-null  object
12  paint_color     263000 non-null  object
13  state           263000 non-null  object
dtypes: float64(2), int64(1), object(11)
memory usage: 38.2+ MB
```

Resultados

Se realizó la comparación de varios modelos para poder determinar los 4 mejores.

	models	MSE-test	MSE-train	MAE-test	MAE-train	MAPE-test	MAPE-train	R2-test	R2-train
3	RandomForestRegressor	1.561569e+07	2.284752e+06	2070.953664	778.687233	23.923205	10.529530	0.880907	0.982689
8	BaggingRegressor	1.733819e+07	3.297504e+06	2215.802827	898.490586	22.437685	10.011483	0.867771	0.975016
11	HistGradientBoostingRegressor	2.792163e+07	2.733416e+07	3466.596810	3443.282776	52.424496	52.678377	0.787056	0.792900
10	GradientBoostingRegressor	3.773564e+07	3.750435e+07	4219.852572	4216.961645	79.587147	78.921768	0.712210	0.715844
4	XGBoost	3.776829e+07	3.751762e+07	4223.087198	4221.315063	79.699944	79.041793	0.711961	0.715743
2	KNeighborsRegressor	4.170302e+07	2.620211e+07	3924.933749	3013.264025	34.904382	24.874665	0.681953	0.801477
9	AdaBoostRegressor	5.899451e+07	5.941448e+07	6105.023198	6121.123499	74.794589	74.667208	0.550079	0.549839
1	DecisionTreeRegressor	6.649062e+07	6.616107e+07	6223.134205	6214.139927	86.600596	86.276622	0.492910	0.498723
12	MLPRegressor	7.531137e+07	7.613092e+07	6679.530409	6712.773770	79.182122	78.241575	0.425639	0.423185
0	LinearRegression	8.051063e+07	8.098795e+07	6921.343891	6951.709956	90.844037	88.614647	0.385987	0.386385
5	Ridge	8.051063e+07	8.098795e+07	6921.345577	6951.711660	90.843923	88.614551	0.385987	0.386385
6	Lasso	8.051072e+07	8.098796e+07	6921.502170	6951.862283	90.837392	88.608314	0.385986	0.386385
7	BayesianRidge	8.051082e+07	8.098796e+07	6921.417651	6951.784457	90.839059	88.610442	0.385985	0.386385

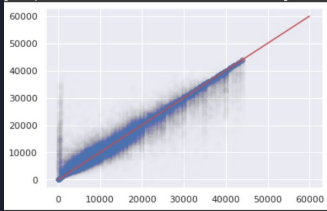
	models	MSE-test	MSE-train	MAE-test	MAE-train	MAPE-test	MAPE-train	R2-test	R2-train
0	GridSearchCV para ajustar los hiperparámetros ...	8.051063e+07	8.098795e+07	6921.343891	6951.709956	90.844037	88.614647	0.385987	0.386385
1	GridSearchCV para ajustar los hiperparámetros ...	1.431249e+07	1.182778e+05	1891.342383	13.430851	19.902864	3.102846	0.890846	0.999104

El análisis anterior nos muestra que los 4 modelos con mejores resultados, según la métrica de R2 son:

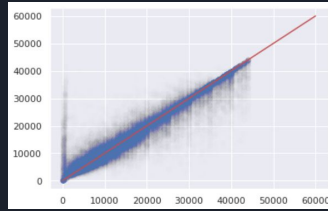
Hiperparámetros del modelo RandomForestRegressor(), RandomForestRegressor(), BaggingRegressor(), HistGradientBoostingRegressor()

Análisis de los modelos

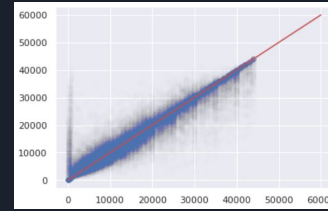
H. RandomForestRegressor()



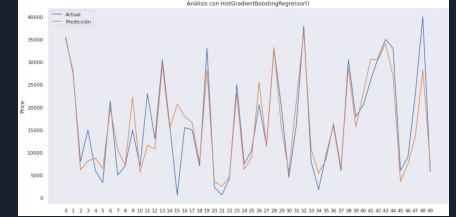
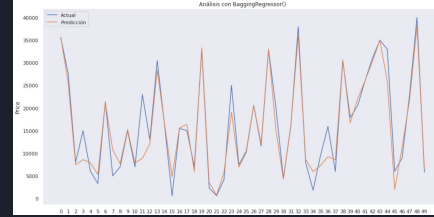
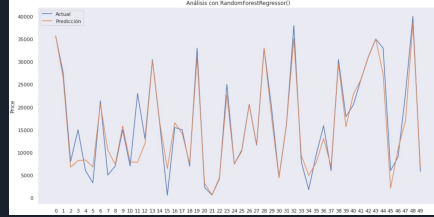
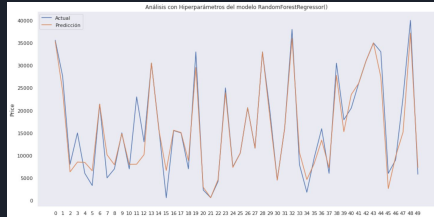
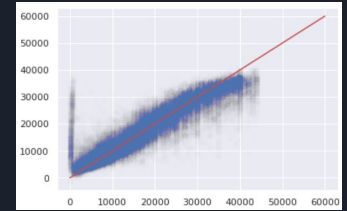
RandomForestRegressor()



BaggingRegressor()



HistGradientBoostingRegressor()



	models	score
0	Hiperparámetros del modelo RandomForestRegressor...	89.08%
1	RandomForestRegressor	88.09%
2	BaggingRegressor	86.78%
3	HistGradientBoostingRegressor	78.71%
4	LinearRegression	38.60%

El modelo de aprendizaje automático más eficaz parece ser el modelo **Random Forest Regressor**, que tiene los parámetros de evaluación más bajos.



Ejemplo

Automóvil con las siguientes características:

- Año: 2020
- Cilindros: 4,
- Kilometraje: 100000.0,'
- Marca: nissan'
- Modelo: maxima
- Estado del automóvil: good
- Tipo de combustible: gas
- Estatus: clean
- Transmisión: automatic'
- Modo de tracción trasera: rwd
- Tipo de carrocería: sedan
- Color: blue
- Estado: ca

Predicción : \$26,093.92

	models	Precio estimado
0	Hiperparámetros del modelo RandomForestRegress...	26093.92



Insights

- Los campos del dataset: 'id', 'url', 'region_url', 'VIN', 'image_url', 'description', 'lat', 'long', 'posting_date', 'size', 'region', 'posting_date', 'posting_date_format' después del análisis se determinó que no son aptos para obtener una predicción de precios
- La mayor concentración de precios se encuentra entre 2,500.00– 44,512.00 USD, 1er y 3er cuartil respectivamente
- La transmisión automática es la que contiene más outliers por arriba de los \$100,000
- En los datos del dataset no existieron rows duplicados, pero si missing values
- 'Good' es la condición más frecuente que se encuentran los automóviles usados, siendo 'New' la menos presente
- '6 Cylinders' tiene mayor presencia en este dataset por lo tanto es directamente proporcional al fuel 'Gas'
- El '4wd' y 'fwd' tienen una presencia similar en el dataset
- 'Sedan' es el tipo de automóvil que más tiene automóviles usados
- 'Silver', 'White' y 'Black' son los colores con mayor presencia en este dataset
- 'Ford', 'Chevrolet' y 'Toyota' tiene los 3 primeros lugares en automóviles usados
- El estado con mayor cantidad de automóviles usados es 'CA' (California)
- Las medidas de tendencia central son: Media = 17,389.72, Mediana= 15,499.00, Moda = \$29,990.00, por lo que obtenemos una curva de asimetría positiva
- Al finalizar las pruebas con diferentes métricas, se concluyó que RandomForestRegressor(), BaggingRegressor() y HistGradientBoostingRegressor() son los más óptimos a utilizar para el entrenamiento en este ejemplo

Conclusión

El modelo de aprendizaje automático más eficaz es el modelo **Random Forest Regressor** (optimización GridSearchCV) con una precisión estimada del 89.3% según la métrica R2