Proyecto Final

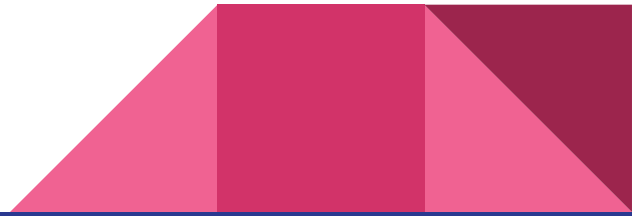
ciencia de datos

Alcon Alan

Elegir el dataset

A lo largo del curso pudimos ver diferentes métodos de aprendizajes automáticos tanto como supervisados y no supervisados, además vimos diferentes formas de analizar datos, información útil.

Lo que esperaba hacer con el trabajo era predecir el precio con los otros datos, que a mi parecer podría resultar en un buen predictor.



Elegir el dataset

Algunas preguntas que tome en cuenta

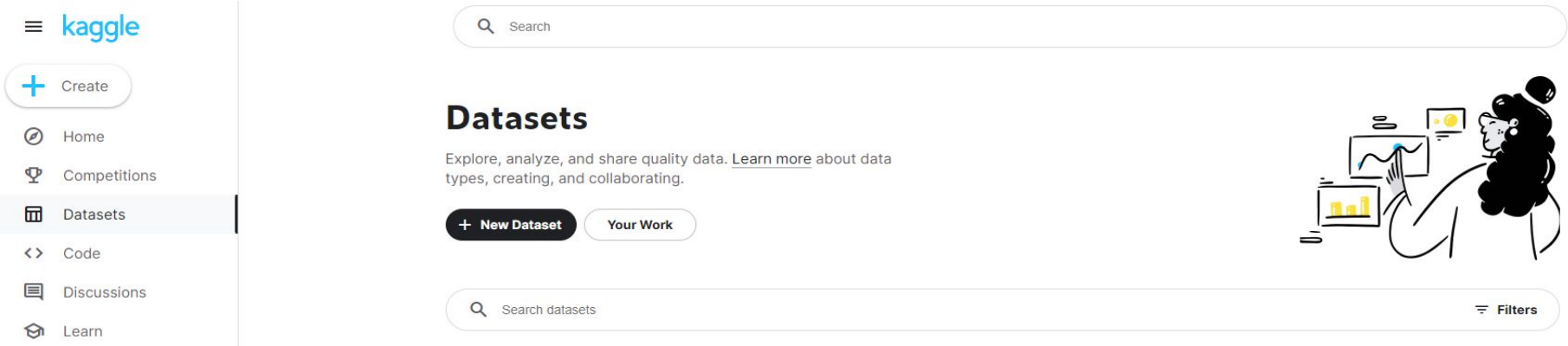
- ¿Cuál quiero que sea el tema?
- ¿Qué tipo de aprendizaje automático me gustaría hacer?
- ¿Qué puedo hacer con el dataset?

Elegir el dataset

El dataset que elegí fue de autos sacado de kaggle.

Las columnas que tiene son: año, precio de venta, kilometraje, dueños anteriores, combustible, vendedor, transmisión, consumo de combustible, cilindrada, potencia de frenado, torque y asientos.

Este dataset posee (8128, 13) filas y columnas respectivamente.



Revisión de datos

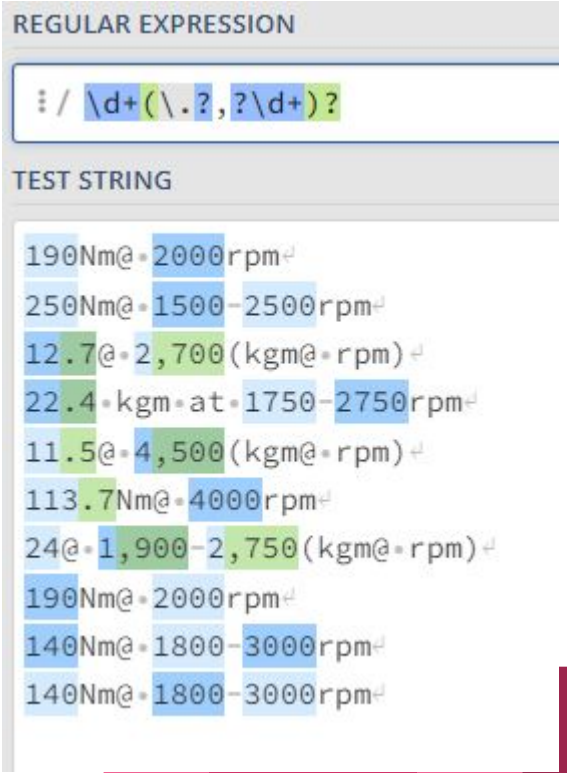
algunos problemas de limpieza que tuve son:

- Datos nulos.
- Unidades junto al valor numérico.
- Datos con valores no numéricos.

Revisión de datos

Estos problemas fueron solucionados usando:

- drop.
- dummies.
- separando las unidades.



Revisión de datos

Antes (8128, 13)

name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	torque	seats
Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl	1248 CC	74 bhp	190Nm@ 2000rpm	5.0
Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl	1498 CC	103.52 bhp	250Nm@ 1500-2500rpm	5.0
Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl	1497 CC	78 bhp	12.7@ 2,700(kgm@ rpm)	5.0
Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl	1396 CC	90 bhp	22.4 kgm at 1750-2750rpm	5.0
Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl	1298 CC	88.2 bhp	11.5@ 4,500(kgm@ rpm)	5.0

Después (7901,14)

year	selling_price	km_driven	owner	mileage	engine	max_power	seats	fuel_Diesel	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	transmission_Manual
2014	450000	145500	0.0	23.40	1248.0	74.00	5.0	1	0	0	1	0	1
2014	370000	120000	1.0	21.14	1498.0	103.52	5.0	1	0	0	1	0	1
2006	158000	140000	2.0	17.70	1497.0	78.00	5.0	0	0	1	1	0	1
2010	225000	127000	0.0	23.00	1396.0	90.00	5.0	1	0	0	1	0	1
2007	130000	120000	0.0	16.10	1298.0	88.20	5.0	0	0	1	1	0	1

Modelo de aprendizaje

Primero utilice Árbol de decisiones en su forma de regresión.

Luego use Random Forest.

Utilicé después Regresión lineal.

Modelo de aprendizaje

Árbol de decisiones dio muy bien pero había un problema de overfitting.

Random Forest fue mejor que Árbol de decisiones.

Regresión Lineal me quede con algunas columnas.

Árbol de decisiones

```
1 mean_squared_error(y_train,arbol_o.predict(X_train),squared=False)
```

```
14983.898027264946
```

```
1 mean_squared_error(y_test,arbol_o.predict(X_test),squared=False)
```

```
210038.26686957412
```

Modelo usado Árbol de decisiones

Con overfitting

```
1 mean_squared_error(y_train,arbol.predict(X_train),squared=False)
```

```
271808.6636743847
```

```
1 mean_squared_error(y_test,arbol.predict(X_test),squared=False)
```

```
299063.31451924774
```

Modelo usado Árbol de decisiones

Sin overfitting

Random Forest

```
1 modelo.best_estimator_
```

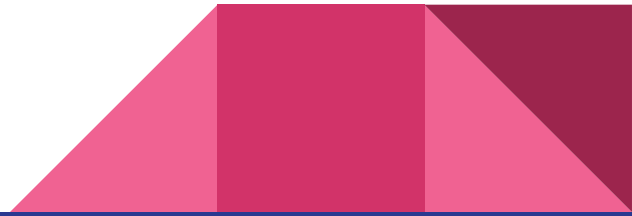
```
RandomForestRegressor(max_depth=80, n_estimators=5)
```

```
1 mean_squared_error(y_train,modelo.predict(X_train),squared=False)
```

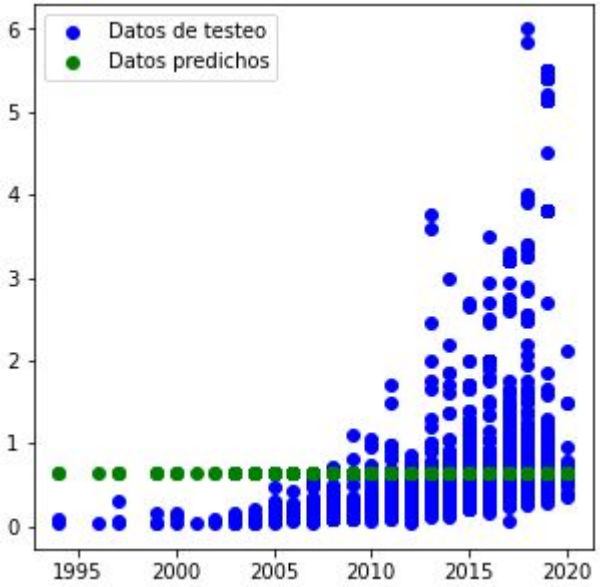
```
182716.3787898812
```

```
1 mean_squared_error(y_test,modelo.predict(X_test),squared=False)
```

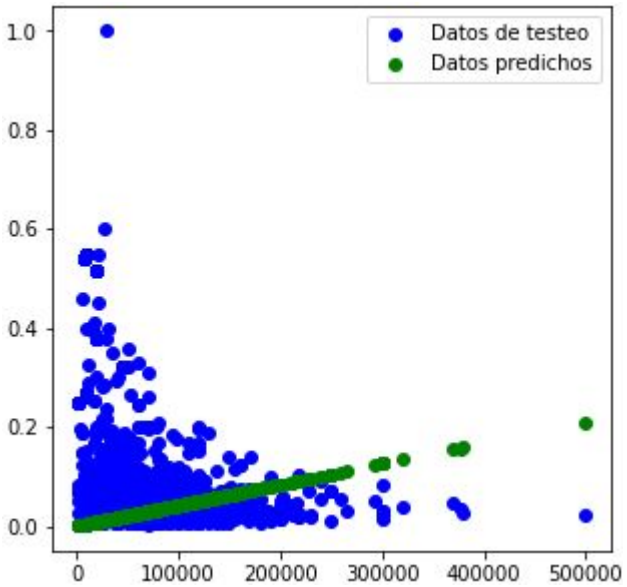
```
234018.26338230298
```



Regresión lineal

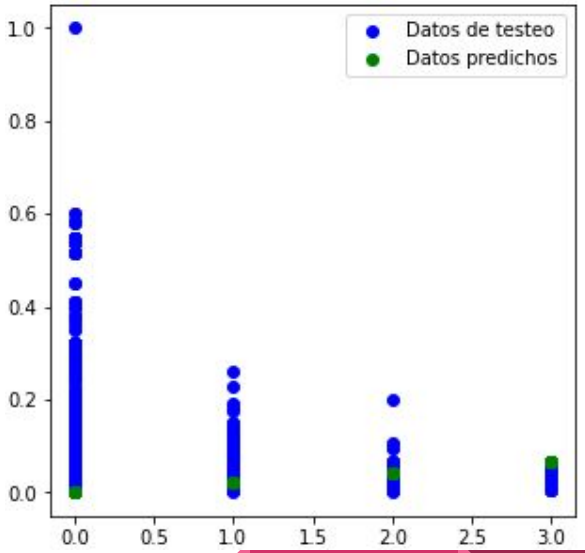


año



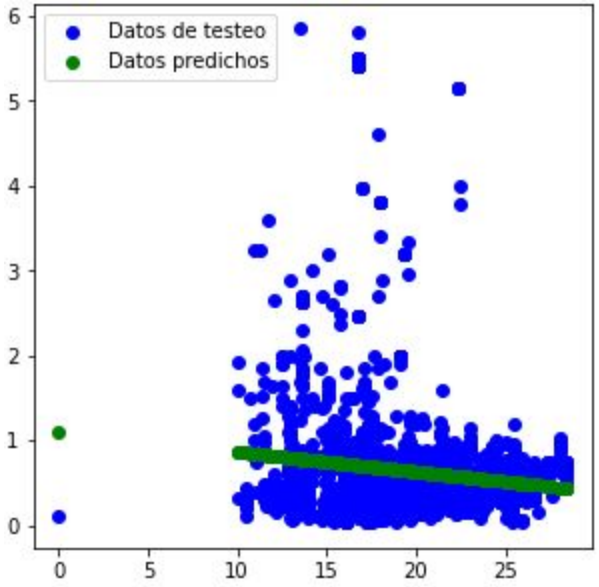
kilometraje

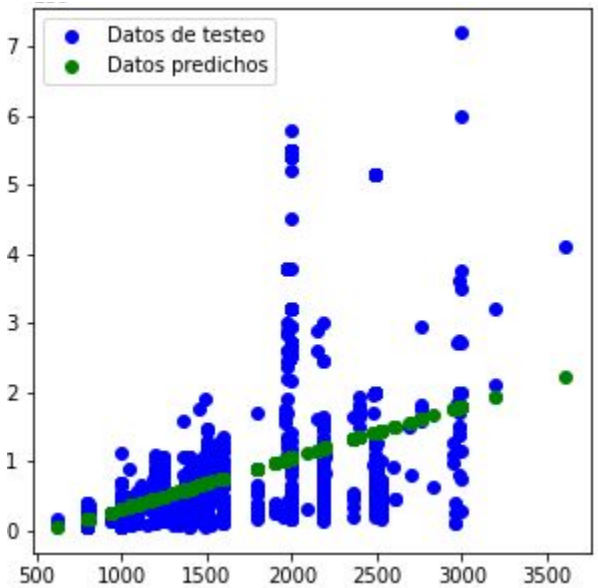
dueños anteriores

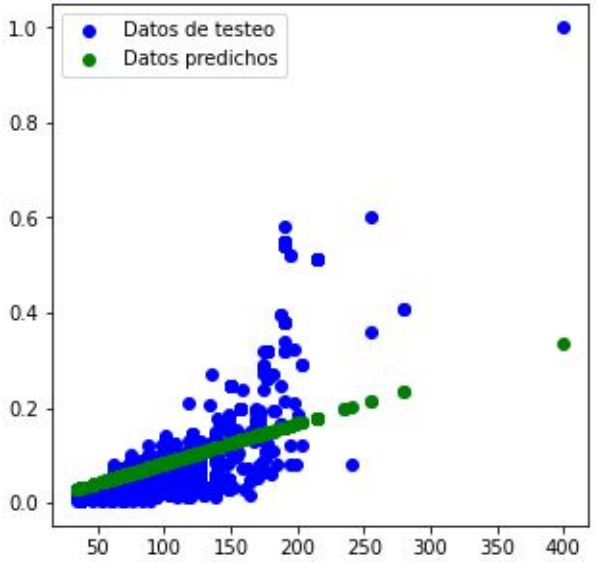


Regresión lineal

potencia de frenado





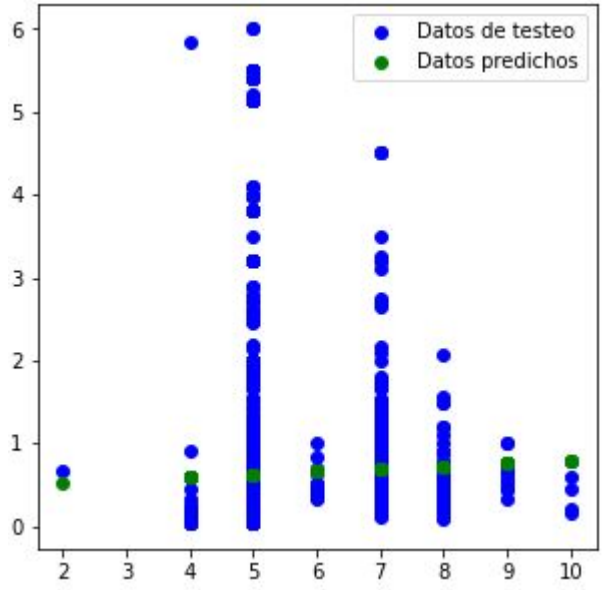


autonomía

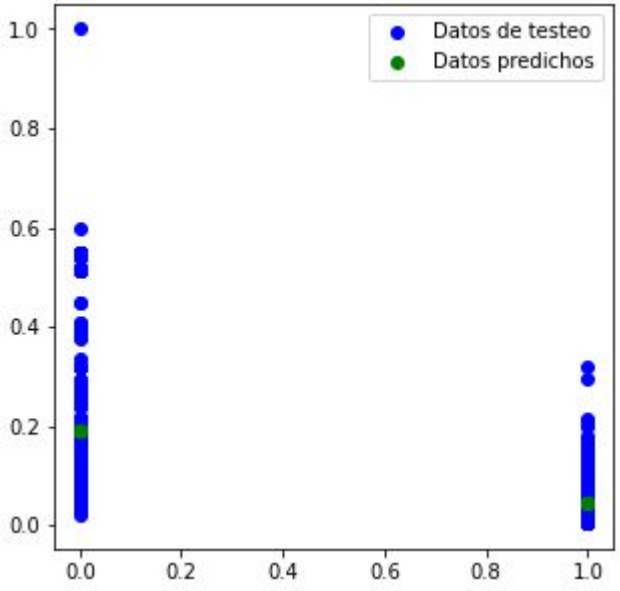
cilindrada

Regresión lineal

cantidad de asientos



tipo de transmisión



Árbol de decisiones

con menos columnas

```
1 mean_squared_error(y_train,arbol.predict(X_train),squared=False)
```

```
268633.6361783635
```

```
1 mean_squared_error(y_test,arbol.predict(X_test),squared=False)
```

```
330574.46735213924
```

Random Forest

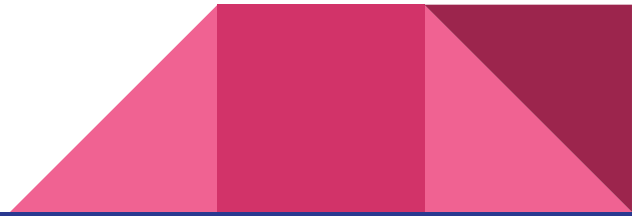
con menos columnas

```
3 mean_squared_error(y_train,random.predict(X_train),squared=False)
```

```
181412.98165376918
```

```
1 mean_squared_error(y_test,random.predict(X_test),squared=False)
```

```
261879.64095134562
```



Conclusión

El modelo con mejor resultado fue Random Forest