TERCER TALLER

Presentado por:

VALENTINA PIEDRAHITA GIL

JHOJAN STIVEN SANCHEZ PALADINES

Presentado a:

Ing ELIAS BUITRAGO BOLIVAR

Universidad ECCI

Ingeniería en Sistemas

Seminario Big Data & Gerencia de datos

Bogotá

2024

Subimos los datos:

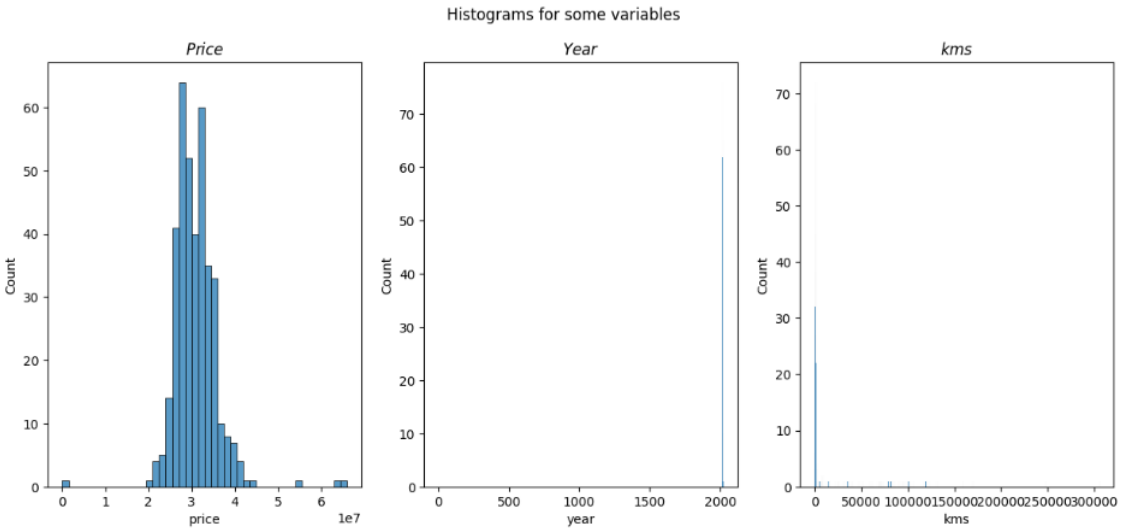
Load data

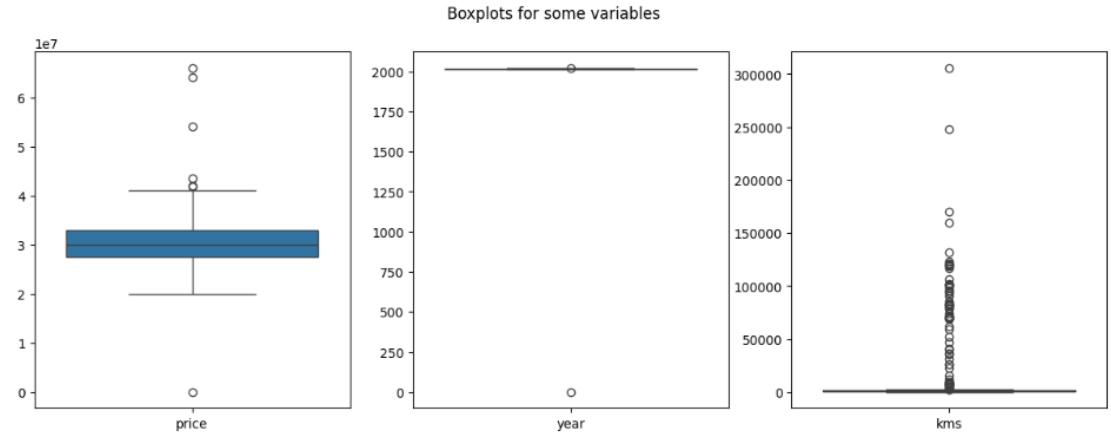
```
cols = ['model','price','year','kms','color','fueltype']
data = pd.read_csv('usedCarsCol_sail (1).csv', sep=',', names=cols, header=0, encoding='latin-1')
print(data.shape)
data.head()
```

(384, 6)

	model	price	year	kms	color	fueltype
0	Chevrolet Sail 1.4 Ls	$24.800.000	2014	117.000	Gris	Gasolina
1	Chevrolet Sail 1.4 Ltz 4 p	$28.000.000	2013	120.844	Gris	Gasolina
2	Chevrolet Sail 1.4 Ls	$33.900.000	2019	68.000	Plateado	Gasolina
3	Chevrolet Sail 1.4 Lt 4 p	$28.000.000	2015	70.000	Color not found	Gasolina
4	Chevrolet Sail 1.4 Ltz	$31.500.000	2016	69.000	Azul	Gasolina

Validamos los histogramas y los boxplots:





Multivariate lineal regression:

Multivariate lineal regression

```
[25] # Define model and prediction
     ols = LinearRegression()
     model1 = ols.fit(X_train, y_train)
     y_pred1 = model1.predict(X_test)
```

```
[26] # accuracy check
     rmse = MSE(y_test, y_pred1, squared=False)
     mae = MAE(y_test, y_pred1)
     r2 = r2_score(y_test, y_pred1)
     print("RMSE: %.2f" % rmse)
     print("MAE: %.2f" % mae)
     print("R2: %.2f" % r2)
```

```
RMSE: 2987701.84
MAE: 2236607.92
R2: 0.43
```

Light GBM:

```
[30] # accuracy check
     rmse = MSE(y_test, y_pred2, squared=False)
     mae = MAE(y_test, y_pred2)
     r2 = r2_score(y_test, y_pred2)
     print("RMSE: %.2f" % rmse)
     print("MAE: %.2f" % mae)
     print("R2: %.2f" % r2)
```

```
RMSE: 2861779.14
MAE: 2170874.56
R2: 0.47
```

Random Forest Regressor:

Random Forest Regressor

```
[34] from sklearn.ensemble import RandomForestRegressor
```

```
[35] model3 = RandomForestRegressor()
     model3.fit(X_train, y_train)
     y_pred3 = model3.predict(X_test)
```

```
[36] # accuracy check
     rmse = MSE(y_test, y_pred3, squared=False)
     mae = MAE(y_test, y_pred3)
     r2 = r2_score(y_test, y_pred3)
     print("RMSE: %.2f" % rmse)
     print("MAE: %.2f" % mae)
     print("R2: %.2f" % r2)
```

```
RMSE: 3331312.23
MAE: 2537112.90
R2: 0.29
```

Xgboost regressor:

```
[43] # Pred
     y_pred4 = model4.predict(X_test)

[44] # accuracy check
     rmse = MSE(y_test, y_pred4, squared=False)
     mae = MAE(y_test, y_pred4)
     r2 = r2_score(y_test, y_pred4)
     print("RMSE: %.2f" % rmse)
     print("MAE: %.2f" % mae)
     print("R2: %.2f" % r2)

     RMSE: 3952103.39
     MAE: 2932321.60
     R2: -0.00
```

Eliminamos los datos que no nos sirven:

El registro que está en 0 y los 6 primeros registros que tienen kms menores a 20.000.

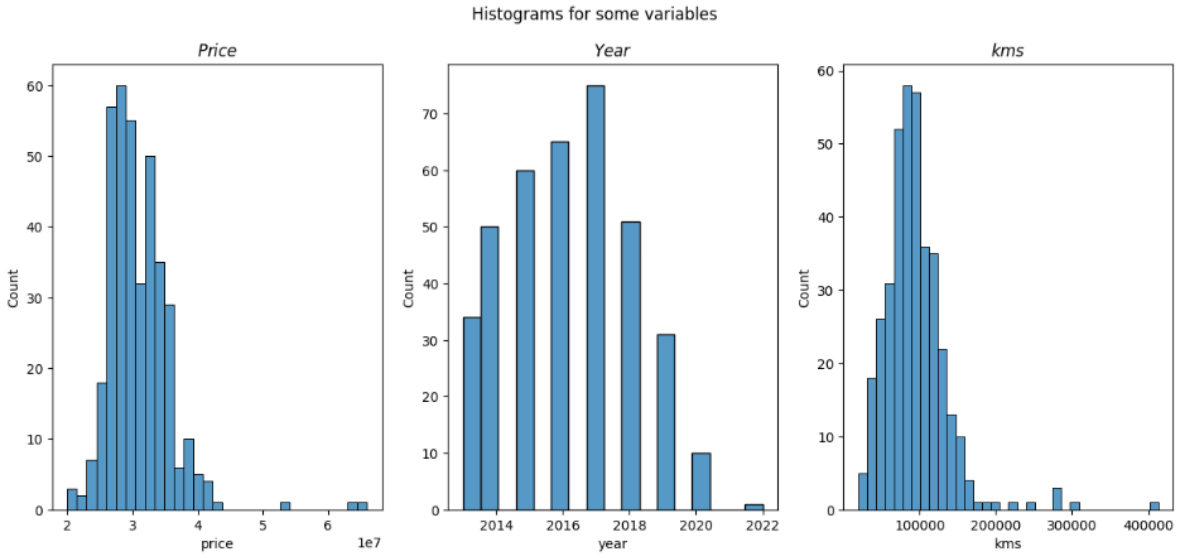
I19

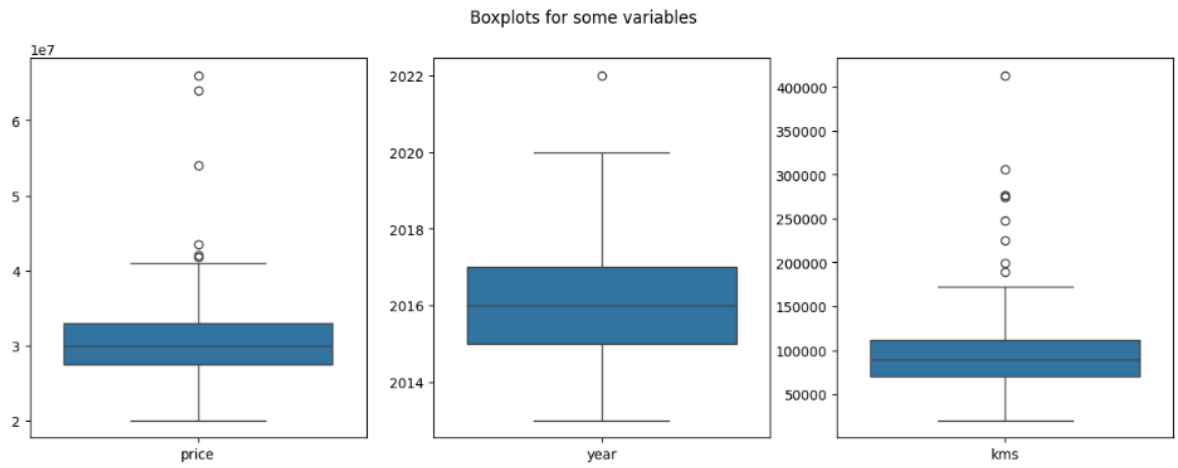
	A	B	C	D	E	F
1	car_model	price	year_model	kms	color	fueltype
2		0	0	0		
3	Chevrolet Sail 1.4 Ltz 5 p	$ 27.000.000	2013	199.000	Gris	Gasolina
4	Chevrolet Sail 1.4 Lt	$ 25.900.000	2013	151.100	Gris	Gasolina
5	Chevrolet Sail 1.4 Lt 4 p	$ 22.300.000	2013	150.000	Negro	Gasolina
6	Chevrolet Sail 1.4 Lt	$ 23.000.000	2013	146.000	Gris	Gasolina
7	Chevrolet Sail 1.4 Ls	$ 26.000.000	2013	138.000	Gris	Gasolina
8	Chevrolet Sail 1.4 Lt 5 p	$ 26.500.000	2013	135.000	Blanco	Gasolina
9	Chevrolet Sail 1.4 Ltz	$ 24.000.000	2013	135.000	Plateado	Gasolina
10	Chevrolet Sail 1.4 Ltz	$ 29.900.000	2013	131.687	Blanco	Gasolina
11	Chevrolet Sail 1.4 Lt	$ 28.000.000	2013	130.000	Negro	Gasolina
12	Chevrolet Sail 1.4 Lt 5 p	$ 30.900.000	2013	129.000	Plateado	Gasolina
13	Chevrolet Sail 1.4 Ltz	$ 24.500.000	2013	122.000	Gris	Gasolina
14	Chevrolet Sail 1.4 Ltz 4 p	$ 28.000.000	2013	120.844	Gris	Gasolina
15	Chevrolet Sail 1.4 Ls	$ 23.900.000	2013	120.000	Plateado	Gasolina
16	Chevrolet Sail 1.4 Lt	$ 26.000.000	2013	118.000	Rojo	Gasolina
17	Chevrolet Sail 1.4 Ls	$ 25.900.000	2013	115.000	Negro	Gasolina
18	Chevrolet Sail 1.4 Ltz	$ 28.500.000	2013	108.000	Negro	Gasolina
19	Chevrolet Sail 1.4 Ls	$ 25.000.000	2013	101.000	Gris	Gasolina
20	Chevrolet Sail 1.4 Ls	$ 30.900.000	2013	99.800	Negro	Gasolina
21	Chevrolet Sail 1.4 Ls	$ 28.500.000	2013	98.435	Gris	Gasolina

H14

	A	B	C	D	E	F
1	car_model	price	year_model	kms	color	fueltype
2	Chevrolet Sail Ltz	$ 24.500.000	2013	1	Azul	Gasolina
3	Chevrolet Sail Ls	$ 27.900.000	2017	1	Azul	Gasolina
4	Chevrolet Sail 1.4l MecÃ¡nica	$ 37.500.000	2018	40	Blanco	Gasolina
5	Chevrolet Cruce	$ 32.000.000	2015	87	Gris	Gasolina
6	Chevrolet Sail 1.4 Ls	$ 26.500.000	2015	141	Plateado	Gasolina
7	Chevrolet Sail 1.4 Ltz Sport	$ 33.000.000	2014	172	Gris	Gasolina
8	Chevrolet Sail 1.4 Ltz	$ 40.000.000	2018	20.000	Gris	Gasolina
9	Chevrolet Sail 1.4 Ls	$ 36.700.000	2018	25.749	Color not found	Gasolina
10	Chevrolet Sail 1.4l MecÃ¡nica	$ 39.000.000	2019	25.954	Gris	Gasolina
11	Chevrolet Sail 1.4 Ls MecÃ¡nica	$ 33.500.000	2017	28.000	Azul	Gasolina
12	Chevrolet Sail 1.4 Ltz Limited	$ 38.000.000	2017	30.000	Gris	Gasolina
13	Chevrolet Sail 1.4 Ls	$ 35.500.000	2019	31.603	Plateado	Gasolina
14	Chevrolet Sail 1.4 Ls	$ 39.000.000	2020	32.000	Plateado	Gasolina
15	Chevrolet Sail 1.4 Lt 4 p	$ 32.000.000	2016	34.000	Blanco	Gasolina
16	Chevrolet Sail 1.4 Ls	$ 32.500.000	2016	34.000	Color not found	Gasolina
17	Chevrolet Sail 1.4 Ltz	$ 41.000.000	2018	35.847	Azul	Gasolina
18	Chevrolet Sail LS FE 1.4	$ 34.800.000	2018	36.557	Azul	Gasolina
19	Chevrolet Sail 1.4 Lt	$ 28.000.000	2014	36.800	Color not found	Gasolina
20	Chevrolet Sail 1.4 Lt	$ 32.000.000	2018	37.000	Plateado	Gasolina
21	Chevrolet Sail 1.4 Ls	$ 35.000.000	2020	38.000	Color not found	Gasolina

Los histogramas y los boxplots mejoraron con el cambio de los datos:





Volvemos a ejecutar los modelos:

Multivariate lineal regression:

Multivariate lineal regression

```
[71] # Define model and prediction
     ols = LinearRegression()
     model1 = ols.fit(X_train, y_train)
     y_pred1 = model1.predict(X_test)
```

```
[72] # accuracy check
     rmse = MSE(y_test, y_pred1, squared=False)
     mae = MAE(y_test, y_pred1)
     r2 = r2_score(y_test, y_pred1)
     print("RMSE: %.2f" % rmse)
     print("MAE: %.2f" % mae)
     print("R2: %.2f" % r2)
```

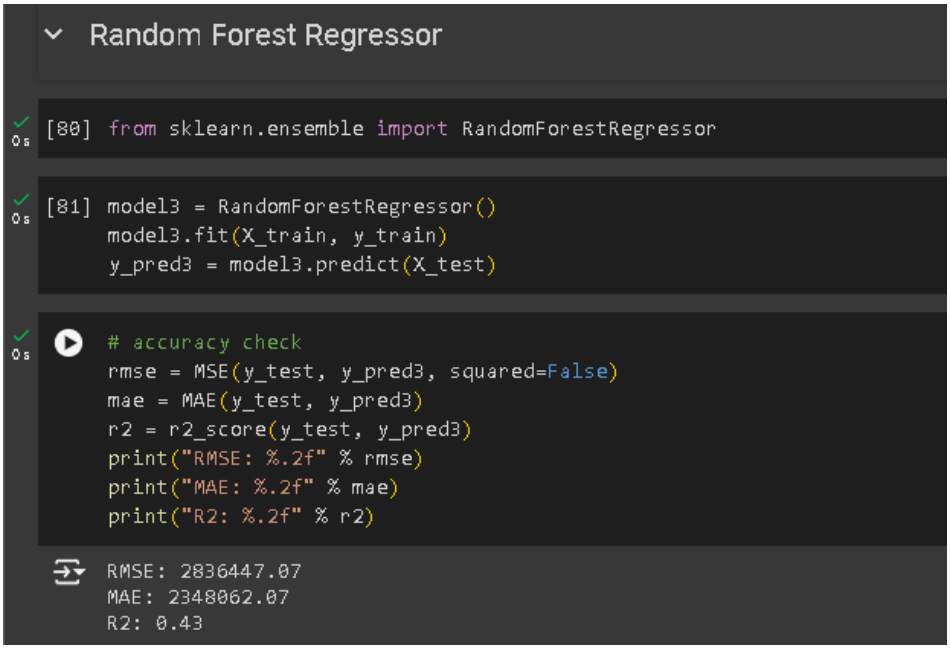
```
RMSE: 2524189.50
MAE: 1956062.97
R2: 0.55
```

Light GBM:

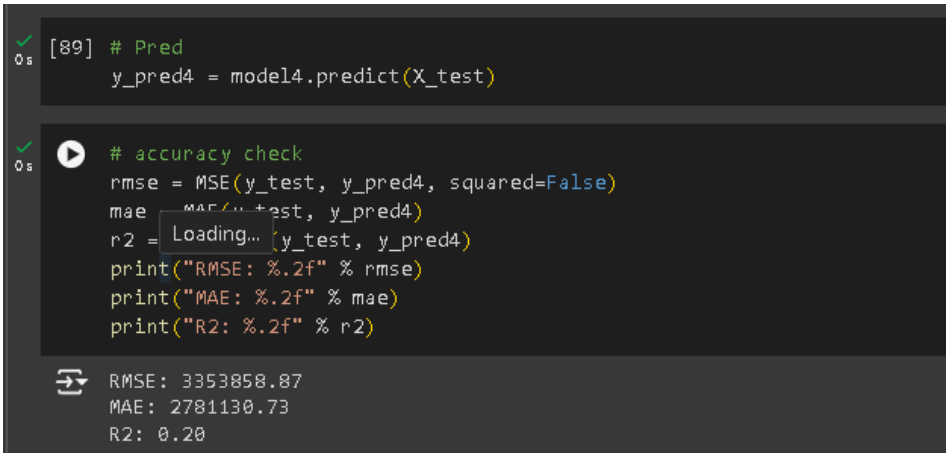
```
# accuracy check
rmse = MSE(y_test, y_pred2, squared=False)
mae = MAE(y_test, y_pred2)
r2 = r2_score(y_test, y_pred2)
print("RMSE: %.2f" % rmse)
print("MAE: %.2f" % mae)
print("R2: %.2f" % r2)
```

```
RMSE: 2637208.23
MAE: 2158436.91
R2: 0.50
```

Random Forest Regressor:



Xgboost regressor:



Vemos que todos los modelos mejoraron, ahora vamos a cambiar los parámetros de cada modelo, excepto del modelo “Multivariate lineal regression” ya que no tiene parámetros:

Light GBM:

Random Forest Regressor:

```
[81] model3 = RandomForestRegressor(n_estimators=500)
     model3.fit(X_train, y_train)
     y_pred3 = model3.predict(X_test)
```

Xgboost regressor:

```
[85] #Define model
     model4 = xgb.XGBRegressor(objective ='reg:squarederror',
                               booster='gbtree',
                               colsample_bytree = 1,
                               importance_type='gain',
                               learning_rate = 0.01,
                               max_depth = 5,
                               alpha = 5,
                               n_estimators = 500,
                               seed=123)
```

Volvemos a ejecutar los modelos:

Light GBM:

```
0s
# accuracy check
rmse = MSE(y_test, y_pred2, squared=False)
mae = MAE(y_test, y_pred2)
r2 = r2_score(y_test, y_pred2)
print("RMSE: %.2f" % rmse)
print("MAE: %.2f" % mae)
print("R2: %.2f" % r2)

RMSE: 2755588.01
MAE: 2203181.88
R2: 0.46
```

Random Forest Regressor:

```
0s
[103] # accuracy check
      rmse = MSE(y_test, y_pred3, squared=False)
      mae = MAE(y_test, y_pred3)
      r2 = r2_score(y_test, y_pred3)
      print("RMSE: %.2f" % rmse)
      print("MAE: %.2f" % mae)
      print("R2: %.2f" % r2)

RMSE: 2787387.82
MAE: 2308623.52
R2: 0.45
```

Xgboost regressor:

```
[111] # accuracy check
      rmse = MSE(y_test, y_pred4, squared=False)
      mae = MAE(y_test, y_pred4)
      r2 = r2_score(y_test, y_pred4)
      print("RMSE: %.2f" % rmse)
      print("MAE: %.2f" % mae)
      print("R2: %.2f" % r2)

      RMSE: 2825072.10
      MAE: 2281302.41
      R2: 0.43
```

Observamos que los modelos Light GBM y Random Forest Regressor bajaron su R2 y que el modelo Xgboost regressor mejoró, volvemos a realizar cambios en los parámetros:

Light GBM:

```
# Hyperparameters
params = {
    'task': 'train',
    'boosting': 'gbdt',
    'objective': 'regression',
    'num_leaves': 100,
    'learning_rate': 0.001,
    'metric': {'l2','l1'},
    'header' : 'true',
    'verbose': 0
}
```

Random Forest Regressor:

```
[102] model3 = RandomForestRegressor(n_estimators=800)
      model3.fit(X_train, y_train)
      y_pred3 = model3.predict(X_test)
```

Xgboost regressor:

```
[106] #Define model
      model4 = xgb.XGBRegressor(objective ='reg:squarederror',
                                booster='gbtree',
                                colsample_bytree = 1,
                                importance_type='gain',
                                learning_rate = 0.001,
                                max_depth = 5,
                                alpha = 5,
                                n_estimators = 800,
                                seed=123)
```

Ejecutamos los modelos:

Light GBM:

```
[135] # accuracy check
      rmse = MSE(y_test, y_pred2, squared=False)
      mae = MAE(y_test, y_pred2)
      r2 = r2_score(y_test, y_pred2)
      print("RMSE: %.2f" % rmse)
      print("MAE: %.2f" % mae)
      print("R2: %.2f" % r2)

      RMSE: 3554013.45
      MAE: 2823658.80
      R2: 0.10
```

Random Forest Regressor:

```
[141] # accuracy check
      rmse = MSE(y_test, y_pred3, squared=False)
      mae = MAE(y_test, y_pred3)
      r2 = r2_score(y_test, y_pred3)
      print("RMSE: %.2f" % rmse)
      print("MAE: %.2f" % mae)
      print("R2: %.2f" % r2)

      RMSE: 2774826.09
      MAE: 2295755.34
      R2: 0.45
```

Xgboost regressor:

```
      # accuracy check
      rmse = MSE(y_test, y_pred4, squared=False)
      mae = MAE(y_test, y_pred4)
      r2 = r2_score(y_test, y_pred4)
      print("RMSE: %.2f" % rmse)
      print("MAE: %.2f" % mae)
      print("R2: %.2f" % r2)

      RMSE: 2880819.01
      MAE: 2287143.32
      R2: 0.41
```

Los modelos Light GBM y Xgboost regressor no les favoreció el cambio de parámetros, pero el modelo Random Forest Regressor se mantuvo igual, volvemos a cambiar los parámetros y a editar los datos.

Light GBM:

```
# Hyperparameters
params = {
    'task': 'train',
    'boosting': 'gbdt',
    'objective': 'regression',
    'num_leaves': 20,
    'learning_rate': 0.001,
    'metric': {'l2','l1'},
    'header' : 'true',
    'verbose': 0
}
```

Random Forest Regressor:

```
[140] model3 = RandomForestRegressor(n_estimators=200)
      model3.fit(X_train, y_train)
      y_pred3 = model3.predict(X_test)
```

Xgboost regressor:

```
[144] #Define model
      model4 = xgb.XGBRegressor(objective ='reg:squarederror',
                                booster='gbtree',
                                colsample_bytree = 1,
                                importance_type='gain',
                                learning_rate = 0.01,
                                max_depth = 5,
                                alpha = 5,
                                n_estimators = 100,
                                seed=123)
```

Eliminamos los 3 registros con mayor precio y los dos registros con mayor km.

I400

	A	B	C	D	E	F
1	car_model	price	year_mod	kms	color	fueltype
2	Hchevrolet Sail Lt 2016	38000000	2016	413000	Amarillo	Gasolina
3	Chevrolet Sail 1.4 Ls MecÃ¡nica	26700000	2017	305705	Gris	Gasolina
4	Chevrolet Chevy Chevytaxy Premium	31000000	2019	277000	Color not fou	Gasolina
5	Chevrolet Sail 2017	54000000	2017	275000	Amarillo	Gasolina
6	Chevrolet Sail 1.4 Ls MecÃ¡nica	66000000	2017	275000	Amarillo	Gasolina
7	Chevrolet Sail 1.4 Lt	28000000	2019	247628	Gris	Gasolina
8	Chevrolet Sail 1.4l MecÃ¡nica	43500000	2019	224690	Amarillo	Gasolina
9	Chevrolet Sail 1.4 Ltz 5 p	27000000	2013	199000	Gris	Gasolina
10	Chevrolet Sail 1.4 Ltz	32000000	2016	189000	Azul	Gasolina
11	Chevrolet Chevy Sail	64000000	2019	172000	Color not fou	Gasolina
12	Chevrolet Sail 1.4 Ltz	30000000	2015	169952	Gris	Gasolina
13	Chevrolet Sail 1.4 Ls	27000000	2015	162000	Gris	Gasolina
14	Chevrolet Sail 1.4 Ls	26990000	2015	160700	Color not fou	Gasolina
15	Chevrolet Sail 1.4 Ltz	27500000	2015	159558	Blanco	Gasolina
16	Chevrolet Sail 1.4 Ls	33000000	2017	155000	Gris	Gasolina
17	Chevrolet Sail 1.4 Ltz	28500000	2014	152400	Gris	Gasolina
18	Chevrolet Sail 1.4 Ls	27000000	2014	152000	Color not fou	Gasolina
19	Chevrolet Sail 1.4 Lt 4 p	28000000	2016	151160	Rojo	Gasolina
20	Chevrolet Sail 1.4 Lt	25900000	2013	151100	Gris	Gasolina

I400

	A	B	C	D	E	F
1	car_model	price	year_mod	kms	color	fueltype
2	Chevrolet Sail 1.4 Ls MecÃ¡nica	66000000	2017	275000	Amarillo	Gasolina
3	Chevrolet Chevy Sail	64000000	2019	172000	Color not fou	Gasolina
4	Chevrolet Sail 2017	54000000	2017	275000	Amarillo	Gasolina
5	Chevrolet Sail 1.4l MecÃ¡nica	43500000	2019	224690	Amarillo	Gasolina
6	Chevrolet Sail 1.4 Ltz	42000000	2020	40708	Negro	Gasolina
7	Chevrolet Sail 1.4 Ltz	41800000	2020	49000	Gris	Gasolina
8	Chevrolet Sail 1.4 Ltz	41000000	2018	35847	Azul	Gasolina
9	Chevrolet Sail 1.4 Ltz Mecanico	40900000	2018	59117	Blanco	Gasolina
10	Chevrolet Sail Ls 1.4 2019	40500000	2019	38000	Blanco	Gasolina
11	Chevrolet Sail 1.4 Ls	40000000	2016	48000	Plateado	Gasolina
12	Chevrolet Sail 1.4 Ltz	40000000	2018	20000	Gris	Gasolina
13	Chevrolet Sail 1.4 Ltz	39900000	2019	81500	Gris	Gasolina
14	Chevrolet Sail 1.4 Ltz	39800000	2020	45100	Rojo	Gasolina
15	Chevrolet Sail 1.4 Ltz	39000000	2020	62500	Plateado	Gasolina
16	Chevrolet Sail 1.4 Ls	39000000	2020	32000	Plateado	Gasolina
17	Chevrolet Sail 1.4l MecÃ¡nica	39000000	2019	25954	Gris	Gasolina
18	Chevrolet Sail 1.4 Ls+aa	38900000	2020	39500	Plateado	Gasolina
19	Chevrolet Sail 1.4 Ltz	38400000	2017	47000	Azul	Gasolina
20	Hchevrolet Sail Lt 2016	38000000	2016	413000	Amarillo	Gasolina
21	Chevrolet Sail 1.4 Ltz	38000000	2020	70000	Rojo	Gasolina

Volvemos a ejecutar:

Multivariate lineal regression:

```
[194] # accuracy check
      rmse = MSE(y_test, y_pred1, squared=False)
      mae = MAE(y_test, y_pred1)
      r2 = r2_score(y_test, y_pred1)
      print("RMSE: %.2f" % rmse)
      print("MAE: %.2f" % mae)
      print("R2: %.2f" % r2)

RMSE: 3021041.23
MAE: 2324922.01
R2: 0.40
```

Light GBM:

```
[198] # accuracy check
      rmse = MSE(y_test, y_pred2, squared=False)
      mae = MAE(y_test, y_pred2)
      r2 = r2_score(y_test, y_pred2)
      print("RMSE: %.2f" % rmse)
      print("MAE: %.2f" % mae)
      print("R2: %.2f" % r2)

RMSE: 3792751.68
MAE: 3186999.18
R2: 0.06
```

Random Forest Regressor:

```
# accuracy check
rmse = MSE(y_test, y_pred3, squared=False)
mae = MAE(y_test, y_pred3)
r2 = r2_score(y_test, y_pred3)
print("RMSE: %.2f" % rmse)
print("MAE: %.2f" % mae)
print("R2: %.2f" % r2)

RMSE: 3203594.40
MAE: 2432105.42
R2: 0.33
```

Xgboost regressor:

```
# accuracy check
rmse = MSE(y_test, y_pred4, squared=False)
mae = MAE(y_test, y_pred4)
r2 = r2_score(y_test, y_pred4)
print("RMSE: %.2f" % rmse)
print("MAE: %.2f" % mae)
print("R2: %.2f" % r2)
```

```
RMSE: 3272782.34
MAE: 2604864.86
R2: 0.30
```

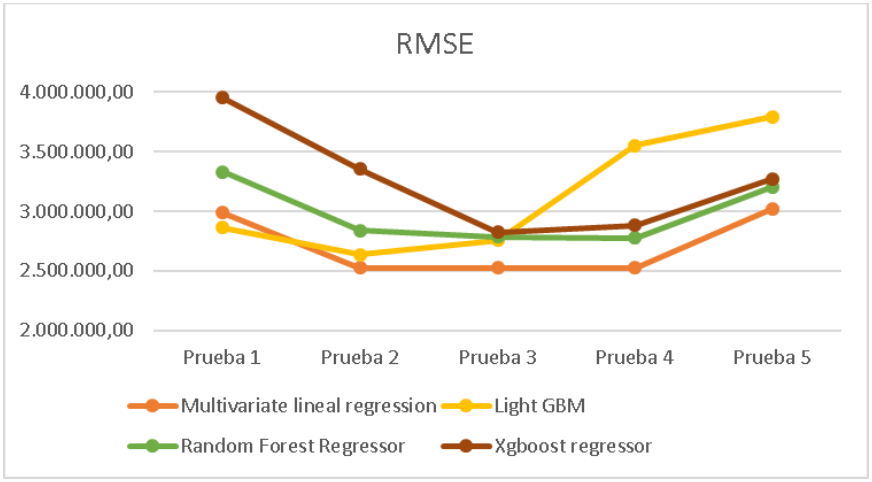
Los tres modelos presentaron cambios negativos.

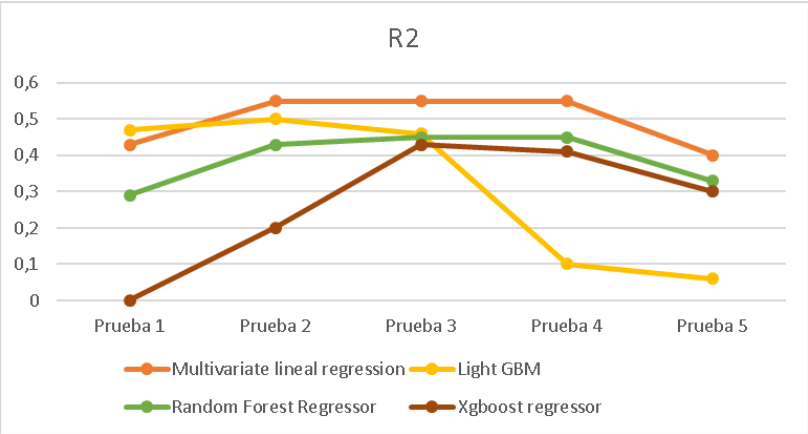
Adjuntamos las tablas con los resultados de cada prueba:

RMSE	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
Multivariate lineal regression	2.987.701,84	2.524.189,50	2.524.189,50	2.524.189,50	3.021.041,23
Light GBM	2.861.779,14	2.637.208,23	2.755.588,01	3.554.013,45	3.792.751,68
Random Forest Regressor	3.331.312,23	2.836.447,07	2.787.387,82	2.774.826,09	3.203.594,40
Xgboost regressor	3.952.103,39	3.353.858,87	2.825.072,10	2.880.819,01	3.272.782,34

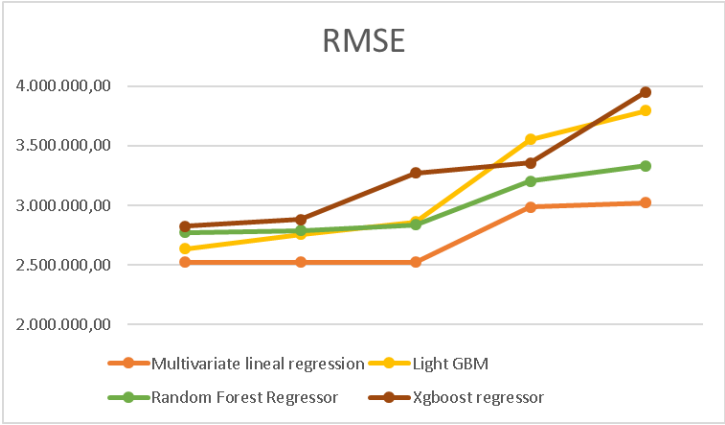
R2	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
Multivariate lineal regression	0,43	0,55	0,55	0,55	0,4
Light GBM	0,47	0,50	0,46	0,10	0,06
Random Forest Regressor	0,29	0,43	0,45	0,45	0,33
Xgboost regressor	0,00	0,20	0,43	0,41	0,3

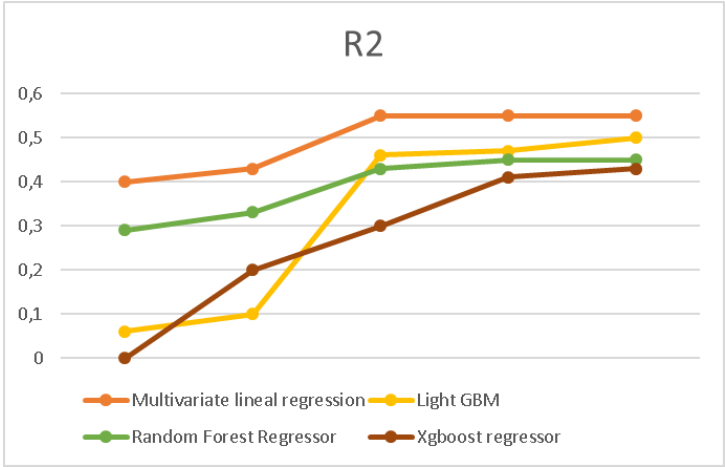
Graficas:





Organizamos de menos a mayor los resultados y las gráficas son las siguientes:





CONCLUSIONES

1. El modelo Multivariate lineal regression aunque sea simple, fue el que mejor R2 tuvo, así que es confiable.
2. Los modelos Light y XGBoost fueron los que peor les fue en las pruebas, así que se necesita un mejor manejo en los parámetros o en la data para poder hacer que funcionen de manera óptima.

3. La limpieza de la data es fundamental, ya que cuando se limpió los datos por primera vez, mejoraron los modelos, pero cuando se modificaron por segunda vez, todos los modelos empeoraron su rendimiento.
4. Los parámetros dependen de la cantidad de datos que tengamos, así que se deben buscar los adecuados para poder tener mejor rendimiento.
5. El modelo Random Forest Regressor presentó mejora después de cambiar su parámetro n_estimators con un valor más bajo y luego de limpiar la data.