CARS PRICE PREDICTION

Presentado por:

TANIA JULIET HURTADO RAMÍREZ

JUAN CAMILO SANCHEZ FERNANDEZ

Presentado a:

Ing ELIAS BUITRAGO BOLIVAR

Universidad ECCI

Ingeniería en Sistemas

Seminario Big Data & Gerencia de datos

Bogotá

2024

DESARROLLO

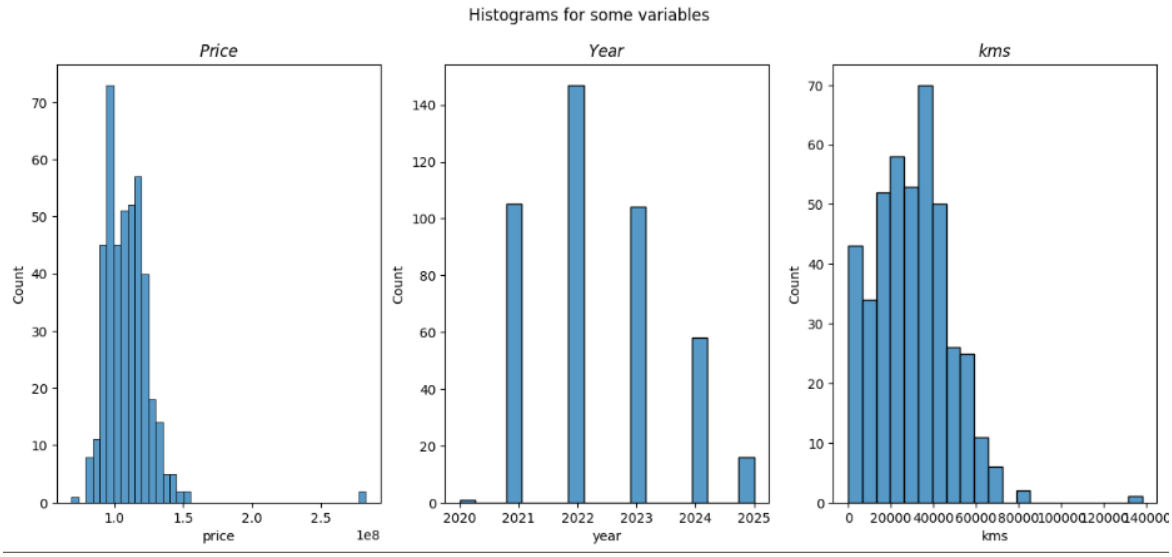
Descargamos lo datos del carro Mazda CX 30.

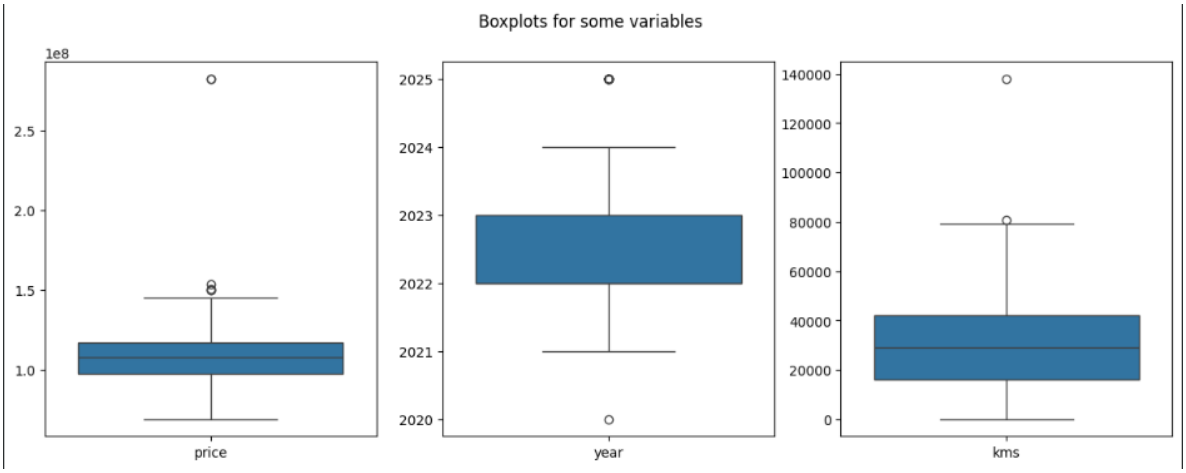
usedCarsCol_cx30.csv

En la primera ejecución, se pudo observar los siguientes valores en cada modelo:

	RMSE	R^2
Multivariate lineal regression	9,031,037.75	0.66
Light GBM	1,0313,830.33	0.55
Random Forest Regressor	15,788,588.55	-0.05
Xgboost regressor	15,022,401.73	0.05

De acuerdo a los siguientes histogramas y boxplots, se toma la decisión de eliminar los dos registros con mayor precio, uno con el modelo más antiguo y dos qué tienen un kilometraje mayor a 80.000 km.





	A	B	C	D	E	F
1	car_model	price	year_model	kms	color	fueltype
2	Mazda CX-90 Grand Touring Signature	282650000	2024	0	Gris	Gasolina
3	Mazda CX-90 Grand Touring Signature	282600000	2024	0	Rojo	HÃ-brido
4	Mazda CX-30 2.5 Grand Touring Lx 4x4	153950000	2024	0	Gris	Gasolina
5	Mazda CX-30 2.5 Grand Touring Lx 4x4	150500000	2024	0	Gris	Gasolina
6	Mazda CX-30 2.5 Grand Touring Lx 4x4	150000000	2025	0	Blanco	Gasolina

	A	B	C	D	E	F
1	car_model	price	year_model	kms	color	fueltype
426	Mazda CX-30 2.0 Prime	83000000	2020	67522	Plateado	Gasolina

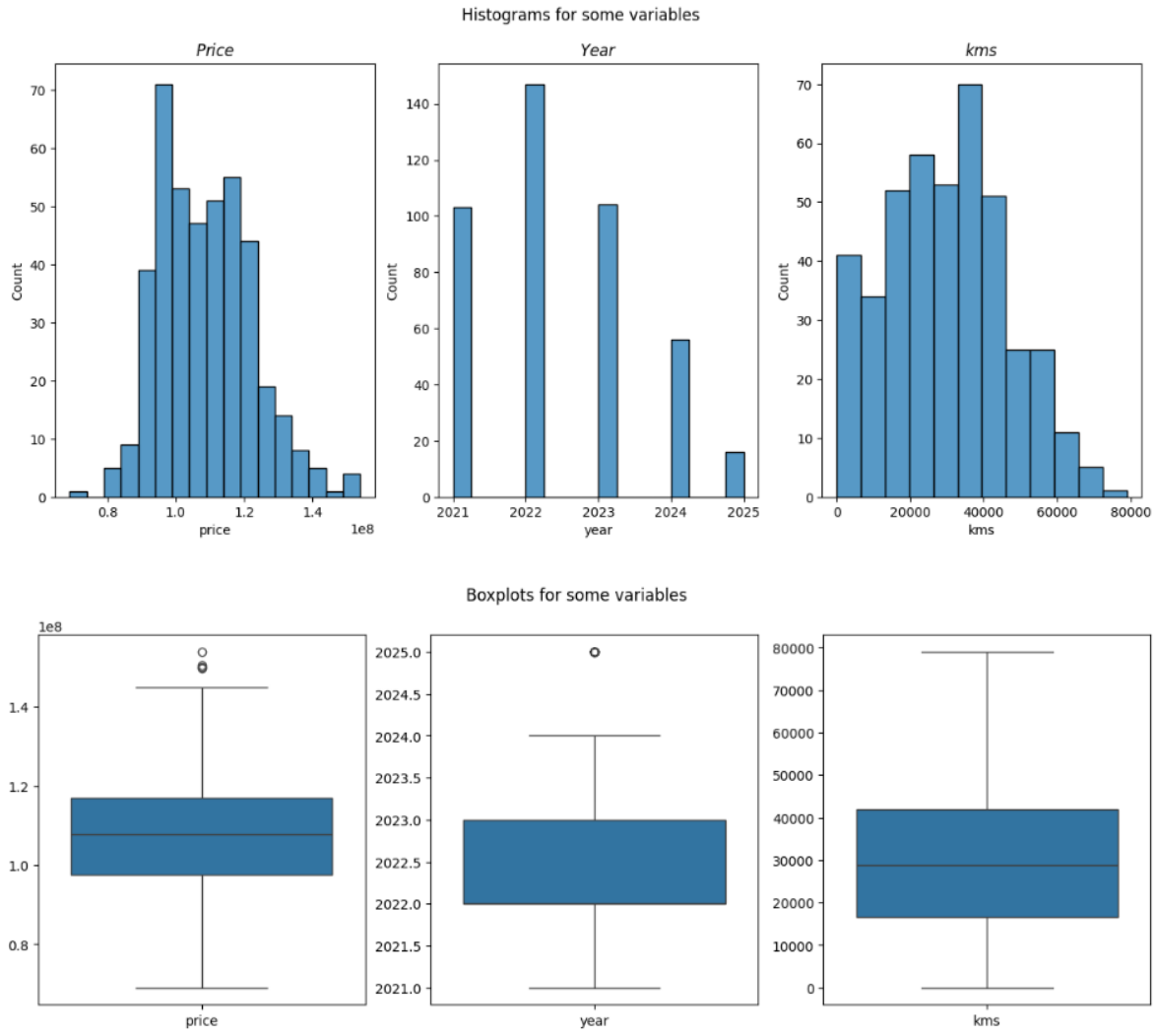
	A	B	C	D	E	F
1	car_model	price	year_model	kms	color	fueltype
2	Mazda CX-30 2.0 Touring At	80000000	2021	138000	Color not found	Gasolina
3	Mazda CX-30 Touring	86900000	2021	80901	Rojo	Gasolina

Subimos el archivo modificado.

usedCarsCol_cx30_
Modificado.csv

En la segunda ejecución, se pudo observar los siguientes valores en cada modelo:

	RMSE	R^2
Multivariate lineal regression	10,592,774.06	0.47
Light GBM	9,752,553.14	0.55
Random Forest Regressor	10,161,377.45	0. 51
Xgboost regressor	11,641,410.19	0.36



Evidenciamos que los modelos Multivariate lineal regression y Light GBM empeoraron, pero los modelos Random Forest Regressor y Xgboost regressor mejoraron.

En el primer modelo no hay parámetros, por lo tanto, se dejará tal cual está.

Y en los otros tres modelos haremos los siguientes cambios:

Light GBM

```
# Hyperparameters
params = {
    'task': 'train',
    'boosting': 'gbdt',
    'objective': 'regression',
    'num_leaves': 30,
    'learning_rate': 0.01,
    'metric': {'l2','l1'},
    'header' : 'true',
    'verbose': 0
}
```

Random Forest Regressor

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

Xgboost regressor

```
#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 = 400,
                          seed=123)
```

Volvemos a ejecutarlos:

	RMSE	R^2
Multivariate lineal regression	10,592,774.06	0.47
Light GBM	10,578,119.76	0.47
Random Forest Regressor	10.248.659.51	0. 50
Xgboost regressor	10,620,903.78	0.46

Se observa que el modelo Light GBM experimentó un aumento en el RMSE, mientras que el Random Forest Regressor mostró una mejora y el XGBoost Regressor también aumento en el RMSE.

Volvemos a realizar cambios:

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

```
model3 = RandomForestRegressor(n_estimators=1000)
model3.fit(X_train, y_train)
y_pred3 = model3.predict(X_test)
```

Xgboost regressor

```
#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 = 100,
                          seed=123)
```

Ejecutamos:

	RMSE	R^2
Multivariate lineal regression	10,592,774.06	0.47
Light GBM	13,748,960.06	0.47
Random Forest Regressor	10,287,987.75	0. 50
Xgboost regressor	13,853,024.67	0.09

Los tres modelos Light GBM, Random Forest Regressor, Xgboost regressor presentaron una desmejora, volvemos a cambiar los parámetros:

Light GBM

```
# Hyperparameters
params = {
    'task': 'train',
    'boosting': 'gbdt',
    'objective': 'regression',
    'num_leaves': 20,
    'learning_rate': 0.1,
    'metric': {'l2','l1'},
    'header' : 'true',
    'verbose': 0
}
```

Random Forest Regressor

```
model3 = RandomForestRegressor(n_estimators=100)
model3.fit(X_train, y_train)
y_pred3 = model3.predict(X_test)
```

Xgboost regressor

```
#Define model
model4 = xgb.XGBRegressor(objective ='reg:squarederror',
                          booster='gbtree',
                          colsample_bytree = 1,
                          importance_type='gain',
                          learning_rate = 0.1,
                          max_depth = 5,
                          alpha = 5,
                          n_estimators = 100,
                          seed=123)
```

	RMSE	R^2
Multivariate lineal regression	10,592,774.06	0.47
Light GBM	9,643,312.85	0.56
Random Forest Regressor	10,297,626.32	0. 50
Xgboost regressor	10,911,876.80	0.43

En los tres modelos se evidencia una mejora, ya que tenemos muy pocos datos, entre más pequeños pongamos los parámetros, mejor funciona.

Light GBM

```
# Hyperparameters
params = {
    'task': 'train',
    'boosting': 'gbdt',
    'objective': 'regression',
    'num_leaves': 10,
    'learning_rate': 0.0001,
    'metric': {'l2','l1'},
    'header' : 'true',
    'verbose': 0
```

Random Forest Regressor

```
model3 = RandomForestRegressor(n_estimators=100,
                               max_depth=5,
                               min_samples_split=2,
                               min_samples_leaf=1,
                               max_features='auto')
model3.fit(X_train, y_train)
y_pred3 = model3.predict(X_test)
```

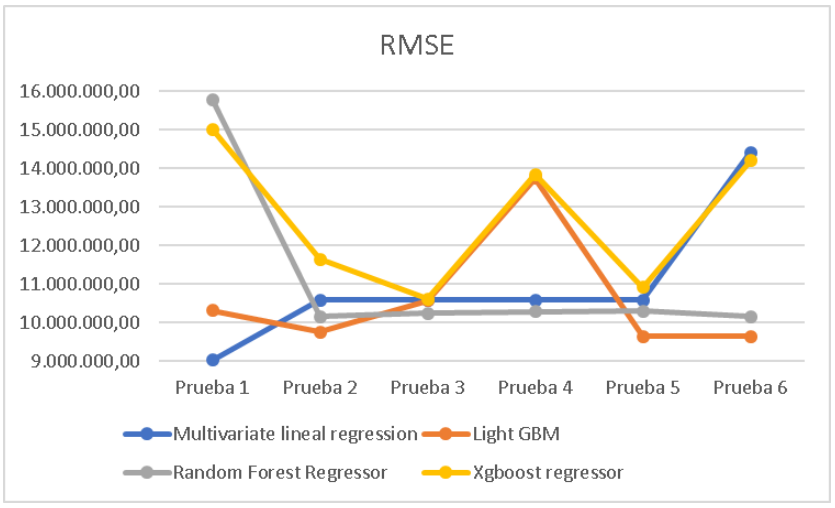
Xgboost regressor

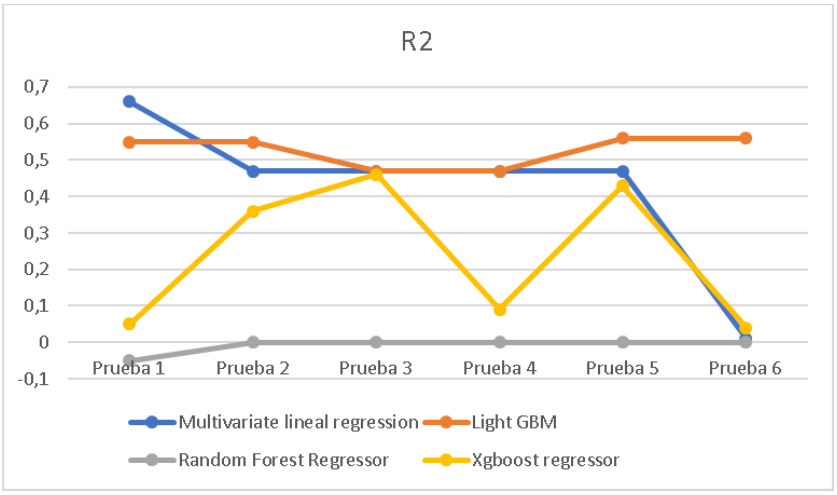
```
#Define model
model4 = xgb.XGBRegressor(objective ='reg:squarederror',
                          booster='gbtree',
                          colsample_bytree = 0.1,
                          importance_type='gain',
                          learning_rate = 0.001,
                          max_depth = 1,
                          alpha = 0.5,
                          n_estimators = 100,
                          seed=123)
```

Ejecutamos:

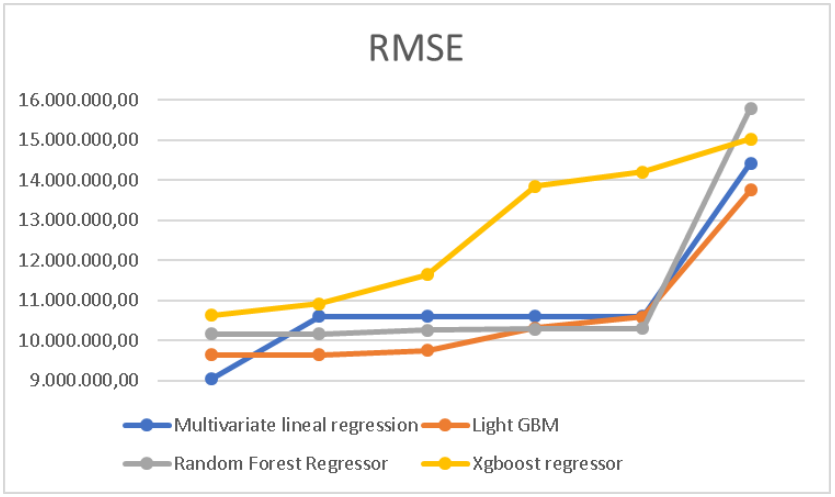
	RMSE	R^2
Multivariate lineal regression	14,419,994.80	0.01
Light GBM	9,643,312.85	0.56
Random Forest Regressor	10,163,364.98	0. 51
Xgboost regressor	14,199,961.85	0.04

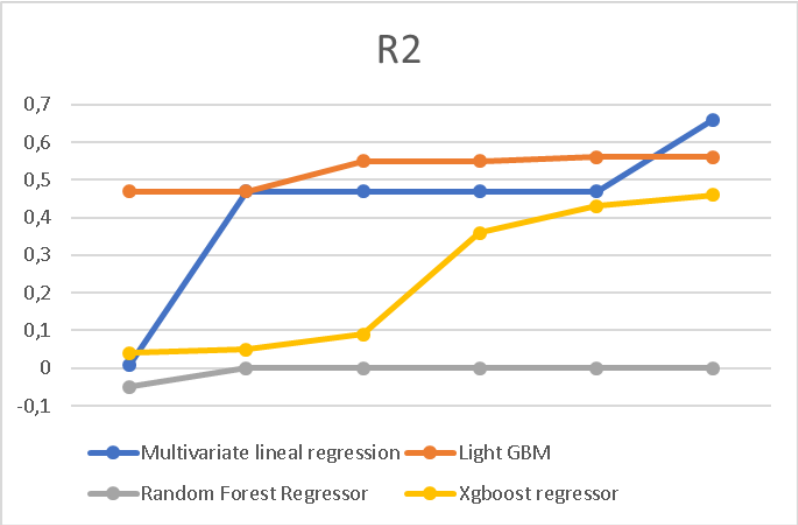
Adjunto graficas donde se ve el resultado de cada prueba.





Para poder sacar mejor las conclusiones, ordené los resultados del peor al mejor.





CONCLUSIONES

1. Modelo de Regresión Lineal Multivariante:
   - Inicialmente, mostró un RMSE de 9,031,037.75 y un R^2 de 0.66.
   - En pruebas posteriores, su rendimiento fluctuó, mostrando en algunos casos un empeoramiento significativo en el RMSE (hasta 14,419,994.80) y en R^2 (hasta 0.01) debido a variaciones en los datos.
2. Modelo Light GBM:
   - Presentó un RMSE inicial de 10,313,830.33 y un R^2 de 0.55.
   - Después de ajustes y modificaciones, el modelo mejoró con un RMSE de 9,643,312.85 y un R^2 de 0.56, indicando una mejor precisión con parámetros más pequeños.
3. Random Forest Regressor:
   - Inicialmente, tuvo un rendimiento pobre con un RMSE de 15,788,588.55 y un R^2 de -0.05.
   - Se observó una mejora con un RMSE de 10,163,364.98 y un R^2 de 0.51, aunque en algunas pruebas después el rendimiento empeoró.
4. XGBoost Regressor:
   - Comenzó con un RMSE de 15,022,401.73 y un R^2 de 0.05.

- El modelo presentó variaciones en su rendimiento, con un RMSE que varió entre 10,911,876.80 y 13,853,024.67, y un R^2 que osciló entre 0.04 y 0.46. Estas variaciones sugieren que el modelo funciona bien con un rate learning menor y “n_estimators” en 400.

5. Todos los modelos mostraron sensibilidad a los parámetros y a la cantidad de datos. Es crucial realizar múltiples iteraciones y ajustes para optimizar los modelos de predicción.
6. El modelo Light GBM demostró ser el más consistente en términos de mejora con ajustes en los parámetros.
7. Los modelos Random Forest y XGBoost, aunque inicialmente tuvieron un rendimiento pobre, mejoraron con ajustes, destacando la importancia de la selección y ajuste de parámetros adecuados.