FLIP ROBO

Pre-Owned Cars Valuation Model

Submitted by:

Utsav Rastogi

ACKNOWLEDGMENT

This project was made possible with the help of my supervisor Swati Mahaseth. Under her guidance, I was able to complete the project on time and ample learning possibilities this project provided me with.

This project gave me the first glimpse of how real-world datasets are mined and what is actually need to be provided to the client in terms of solutions. This was my first-time experience with a dataset that I crafted for the client who required current market prices of used cars in major cities.

Data analysis and preprocessing was done keeping in mind how to make the code run efficiently and quickly with utilizing minimal memory. I had to refer to various articles on Medium, Analytics Vidhya, Kaggle and stack exchange to look for potential debugging solutions and visualization plots.

INTRODUCTION

Business Problem Framing

This project tries to shed light on an efficient way to price the used cars that can be utilized by car selling business owners to mark their products. The change in market due to covid 19 impact, client needs a machine learning model which is trained on current data from different cities.

Conceptual Background of the Domain Problem

There are a variety of used car seller who have already digitized their businesses by going online and are already selling various Indian cities. Companies like cardekho, cars24 and ola cars use different predictive models to ascertain the price of used cars. The buyer and seller can gain better information and prices using these models and create value for themselves.

Review of Literature

This dataset was mined from websites like cardekho and Ola cars. This data was further analyzed and cleaned with standard pre-processing steps. Then this dataset is saved in ".csv" format which will be subsequently used to create a predictive model. The model predicts the price of the car using several independent features. The model is based on a regression algorithm which are usually used to predict a continuous target variable.

The model with least difference in r2 score and cross validation score is finalized and saved in a pickle file. This model can be further integrated by webapps to check the price of any car in India.

Motivation for the Problem Undertaken

The objective behind this project was to provide the client with an overview of how used car prices can be marked efficiently with a statistical model. Cars have certainly become a necessary part of our daily lives. They represent social status, general and an urgent utility.

They have integrated people living far off much better than any other means of transportation. The degree of freedom and privacy this mode provides, no other means of travel can even compare. With the growing need of cars, old cars have become a burden on the environment if every time they need to be scrapped. Hence, selling is a fruitful option which can fulfil the dreams of many people with financial constraints and make it easier for them to carry on with their lives.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

Statistical analysis	Zscore was used to determine the outliers in the data and Correlation helped understand the relation between features and target. Skewness was also used to check the distribution and reduce it in the features which weren't gaussian.
Visual analysis	Libraries used: Matplotlib and Seaborn
Algorithms	Different Regression algorithms such as XGB Regressor & Lasso Regression were used which are all provided by sklearn library.

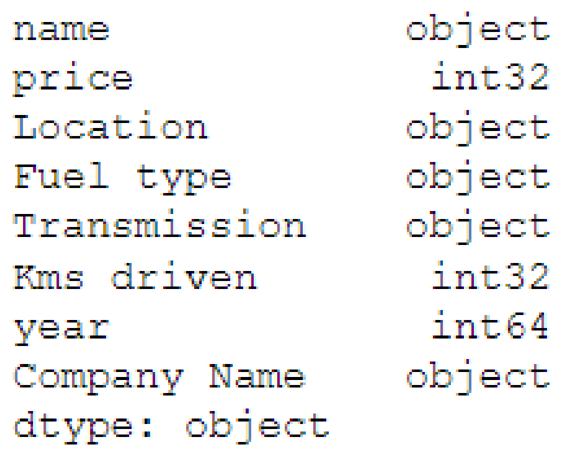
Data Sources and their formats

1. Data is sourced from:

Ola cars	3169 samples
CarDekho	2000 samples

2. Data is saved in a CSV format
3. Data Features:

8 Features with only two of them being numerical – “price”, ”Kms driven”, rest are categorical variables.



```
name             object
price             int32
Location         object
Fuel type        object
Transmission     object
Kms driven        int32
year              int64
Company Name     object
dtype: object
```

4. Null values present only in “Transmission”

Data Pre-processing Done

1. Removing the kms from Kms driven column
2. Removing the ‘₹’ from price
3. Converting price, year and Kms driven to integer format
4. Label encoding the features with dtype = ”object”
5. Imputing null values with Mode in Transmission column

Data Inputs- Logic- Output Relationships

Price is the dependent variable and year, Kms driven, name, Fuel type, Company Name, Transmission are the independent variables.

Highest relationship is between price and year the car manufactured.

Hardware and Software Requirements and Tools Used

Hardware: Windows 11
Software: Jupyter notebook
Libraries: seaborn, matplotlib, statsmodels, numpy, pandas, sklearn, scipy, pickle

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

The dataset was analyzed using the standard procedure where we indulge in statistical and graphical analysis of the dataset. Statistical methods like Zscore were used to clean data.

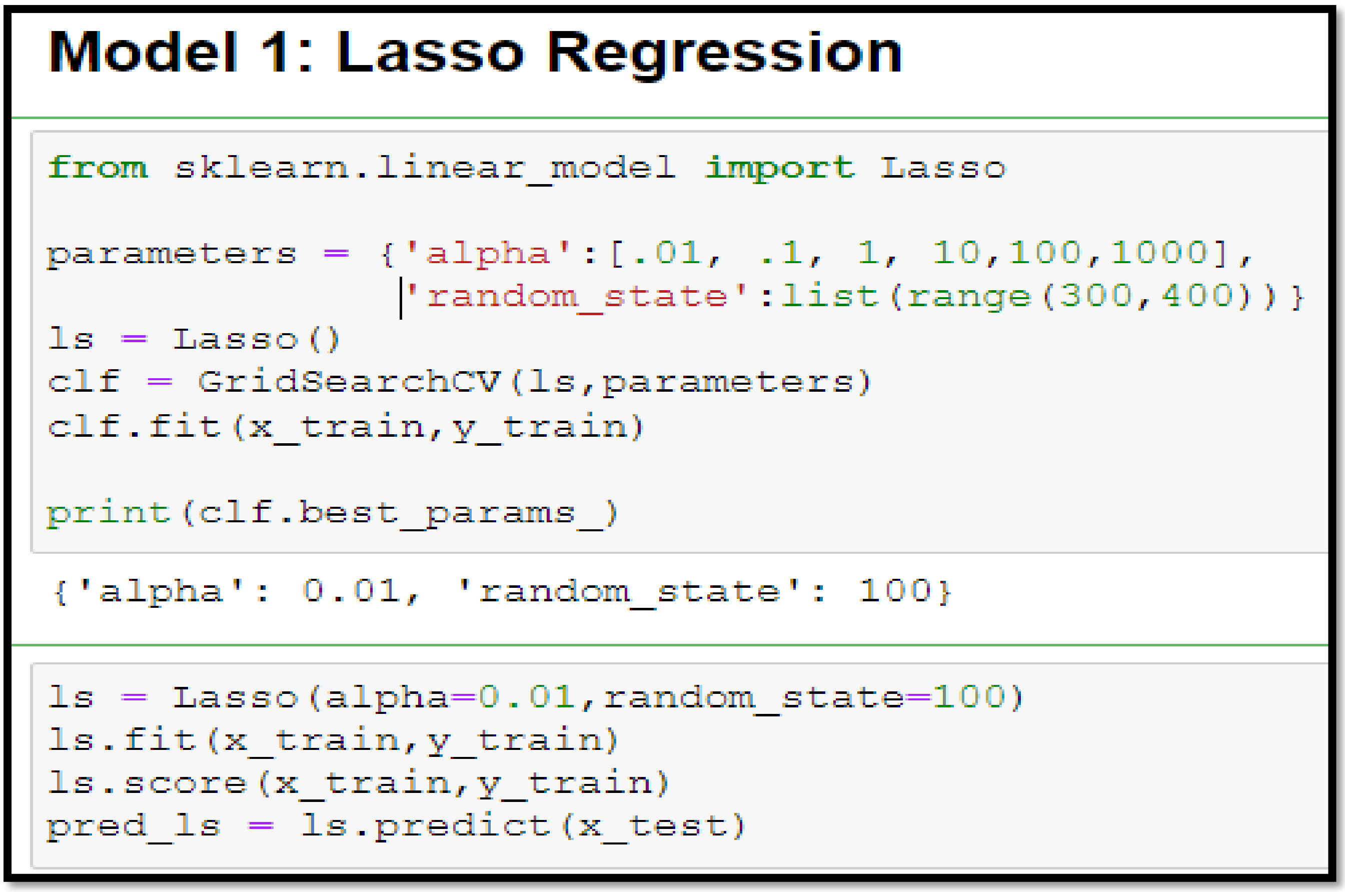
Best random state is found and is used to split the data to obtain an optimum score.

Regression algorithms were used to create a model as the target variable was a continuous variable.

Testing of Identified Approaches (Algorithms)

1. Lasso Regression
2. Decision Tree Regressor
3. Random Forest Regressor
4. XGB regressor

Run and evaluate selected models



Model 1: Lasso Regression

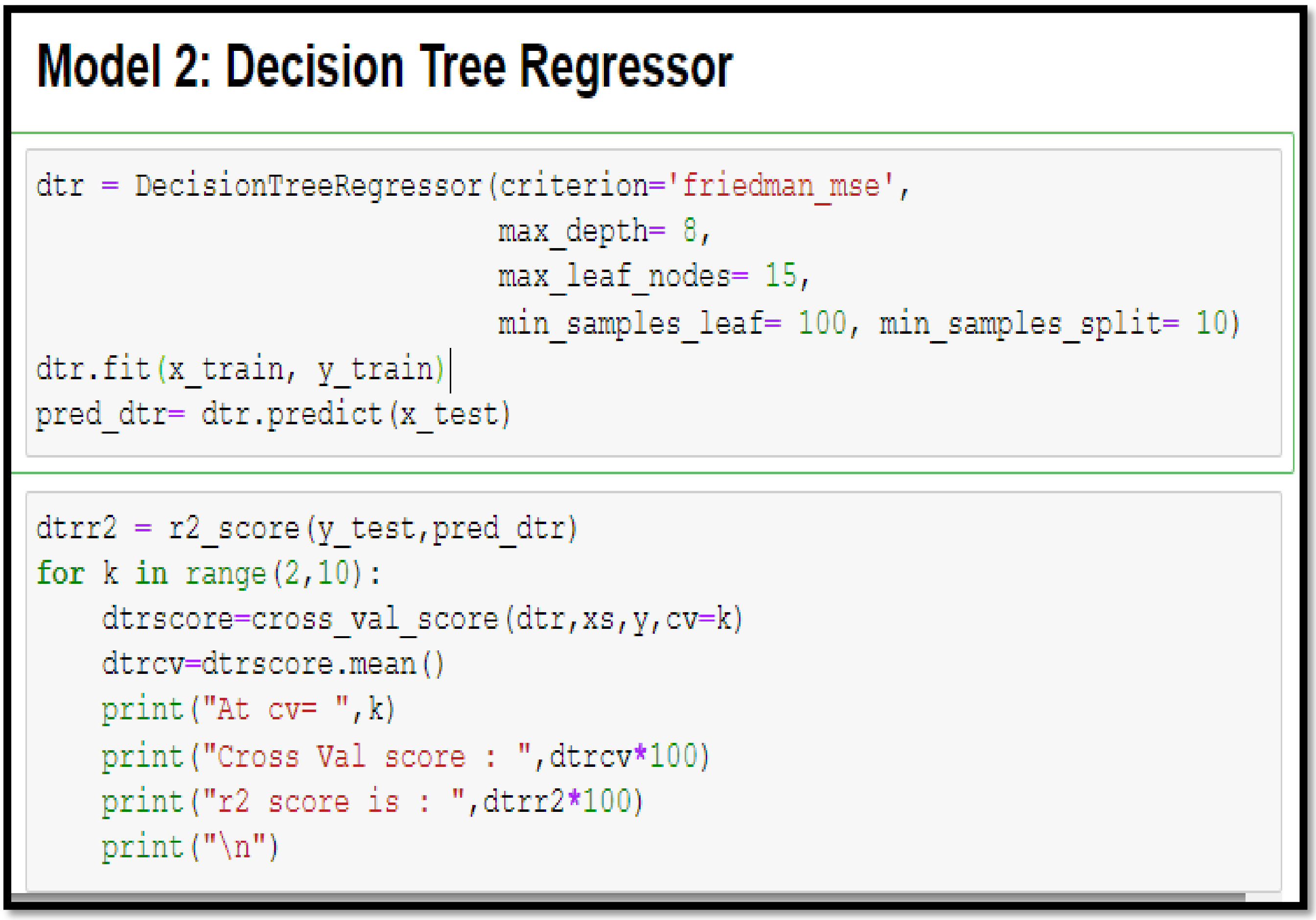
```
from sklearn.linear_model import Lasso

parameters = {'alpha':[.01, .1, 1, 10,100,1000],
              'random_state':list(range(300,400))}
ls = Lasso()
clf = GridSearchCV(ls,parameters)
clf.fit(x_train,y_train)

print(clf.best_params_)
```

```
{'alpha': 0.01, 'random_state': 100}
```

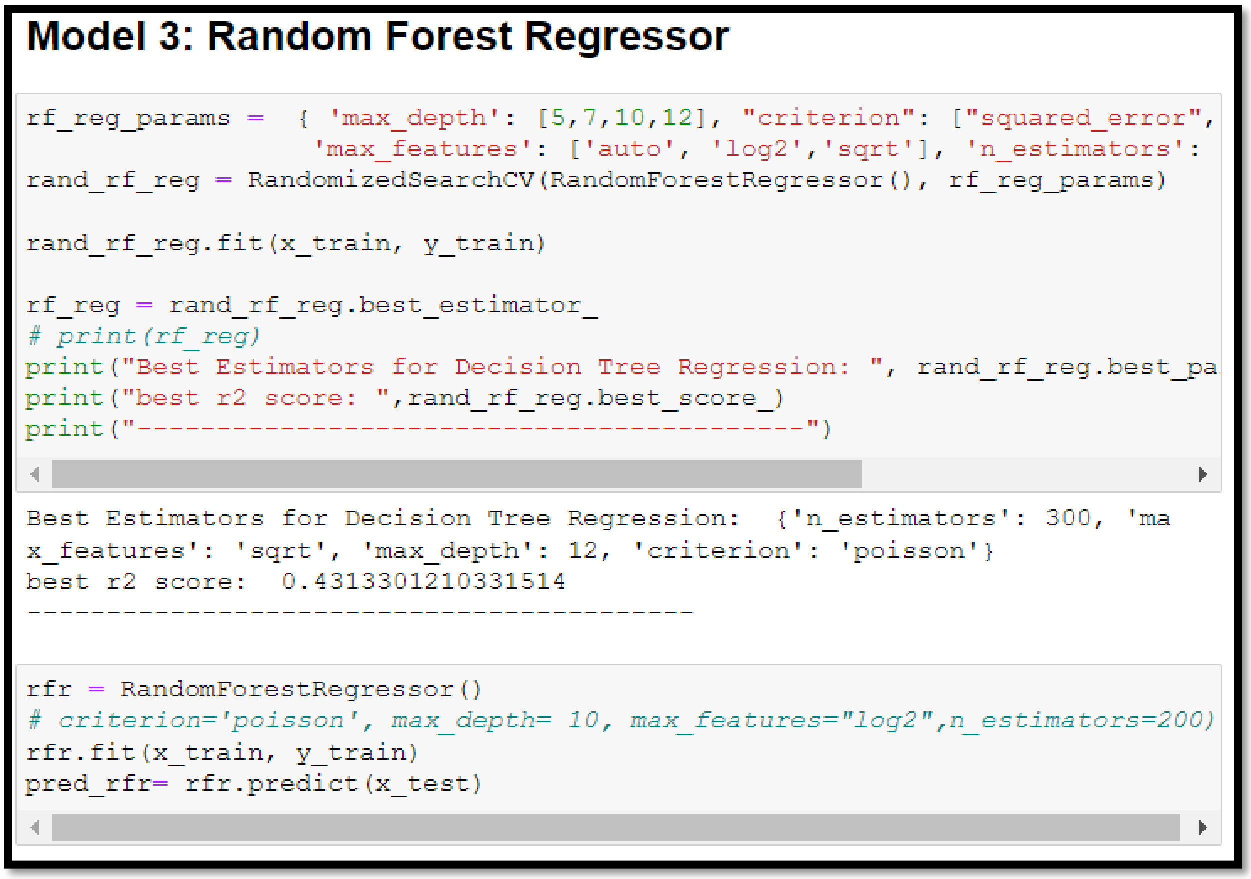
```
ls = Lasso(alpha=0.01,random_state=100)
ls.fit(x_train,y_train)
ls.score(x_train,y_train)
pred_ls = ls.predict(x_test)
```



Model 2: Decision Tree Regressor

```
dtr = DecisionTreeRegressor(criterion='friedman_mse',
                            max_depth= 8,
                            max_leaf_nodes= 15,
                            min_samples_leaf= 100, min_samples_split= 10)
dtr.fit(x_train, y_train)
pred_dtr= dtr.predict(x_test)
```

```
dtrr2 = r2_score(y_test,pred_dtr)
for k in range(2,10):
    dtrscore=cross_val_score(dtr,xs,y,cv=k)
    dtrcv=dtrscore.mean()
    print("At cv= ",k)
    print("Cross Val score : ",dtrcv*100)
    print("r2 score is : ",dtrr2*100)
    print("\n")
```



Model 3: Random Forest Regressor

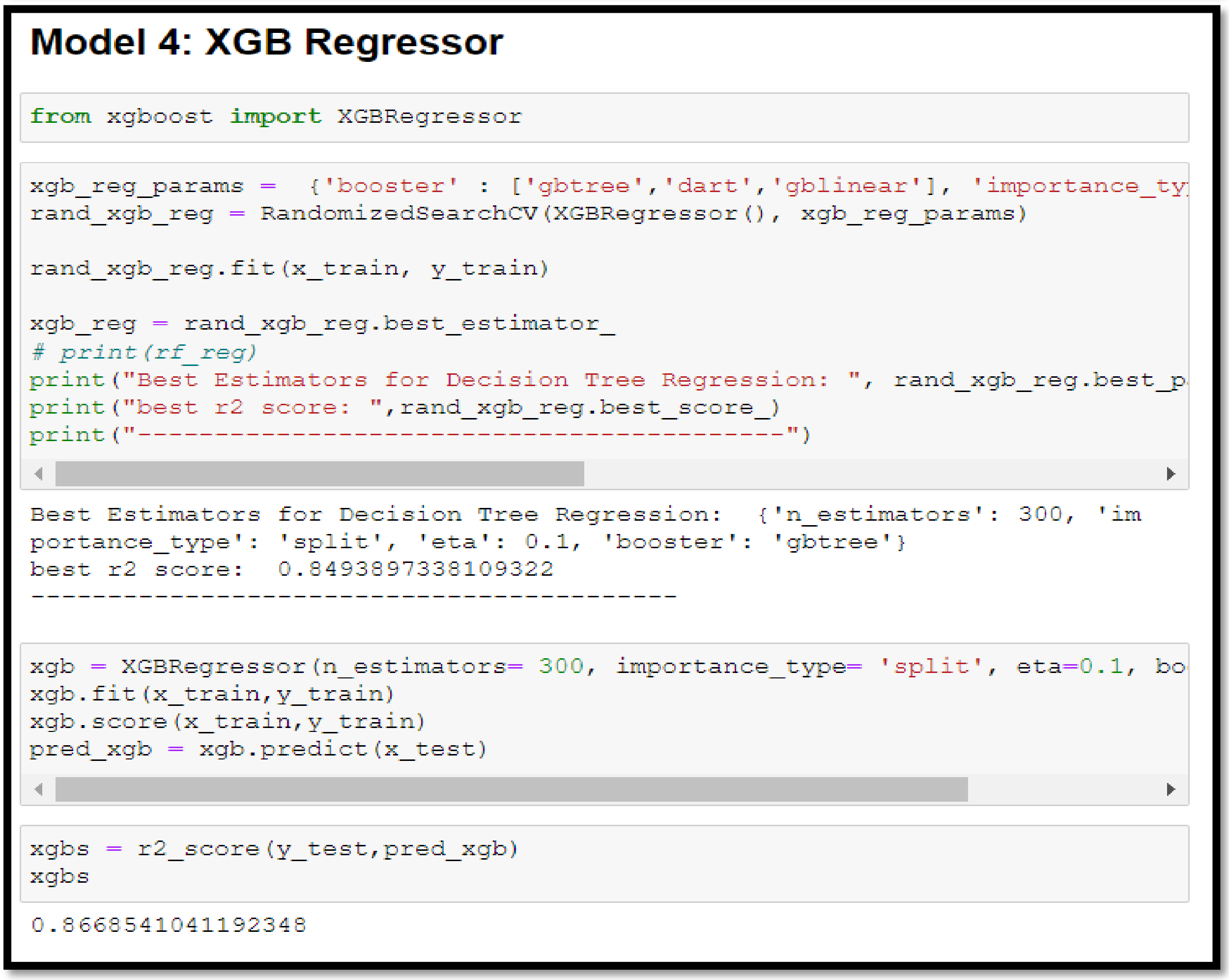
```
rf_reg_params =  { 'max_depth': [5,7,10,12], "criterion": ["squared_error",
                   'max_features': ['auto', 'log2','sqrt'], 'n_estimators':
rand_rf_reg = RandomizedSearchCV(RandomForestRegressor(), rf_reg_params)

rand_rf_reg.fit(x_train, y_train)

rf_reg = rand_rf_reg.best_estimator_
# print(rf_reg)
print("Best Estimators for Decision Tree Regression: ", rand_rf_reg.best_pa
print("best r2 score: ",rand_rf_reg.best_score_)
print("------------------------------------------")
```

```
Best Estimators for Decision Tree Regression:  {'n_estimators': 300, 'ma
x_features': 'sqrt', 'max_depth': 12, 'criterion': 'poisson'}
best r2 score:  0.4313301210331514
------------------------------------------
```

```
rfr = RandomForestRegressor()
# criterion='poisson', max_depth= 10, max_features="log2",n_estimators=200)
rfr.fit(x_train, y_train)
pred_rfr= rfr.predict(x_test)
```



Model 4: XGB Regressor

```
from xgboost import XGBRegressor
```

```
xgb_reg_params =  {'booster' : ['gbtree','dart','gblinear'], 'importance_ty
rand_xgb_reg = RandomizedSearchCV(XGBRegressor(), xgb_reg_params)

rand_xgb_reg.fit(x_train, y_train)

xgb_reg = rand_xgb_reg.best_estimator_
# print(rf_reg)
print("Best Estimators for Decision Tree Regression: ", rand_xgb_reg.best_p
print("best r2 score: ",rand_xgb_reg.best_score_)
print("------------------------------------------")
```

```
Best Estimators for Decision Tree Regression:  {'n_estimators': 300, 'im
portance_type': 'split', 'eta': 0.1, 'booster': 'gbtree'}
best r2 score:  0.8493897338109322
------------------------------------------
```

```
xgb = XGBRegressor(n_estimators= 300, importance_type= 'split', eta=0.1, bo
xgb.fit(x_train,y_train)
xgb.score(x_train,y_train)
pred_xgb = xgb.predict(x_test)
```

```
xgbs = r2_score(y_test,pred_xgb)
xgbs
```

```
0.8668541041192348
```

r2 Score:

Model	Score
Decision Tree Regressor	66.91602137576673
Random Forest Regressor	66.91602137576673
Lasso Regression	36.42172238391035
XGB Regressor	86.68541041192348

Cross Validation Score:

Model	Score
Decision Tree Regressor	57.107691100804125
Random Forest Regressor	71.94726181955497
Lasso Regression	30.41666056374264
XGB Regressor	83.45996117921885

Key Metrics for success in solving problem under consideration

The least difference between r2score and cv score was seen in gradient boosting regressor hence proving least fitting problem. Hence it is considered for modelling the data under consideration (test data).

The r2 score was used.

R-squared is a metric of correlation. Correlation is measured by “r” and it tells us how strongly two variables can be related.

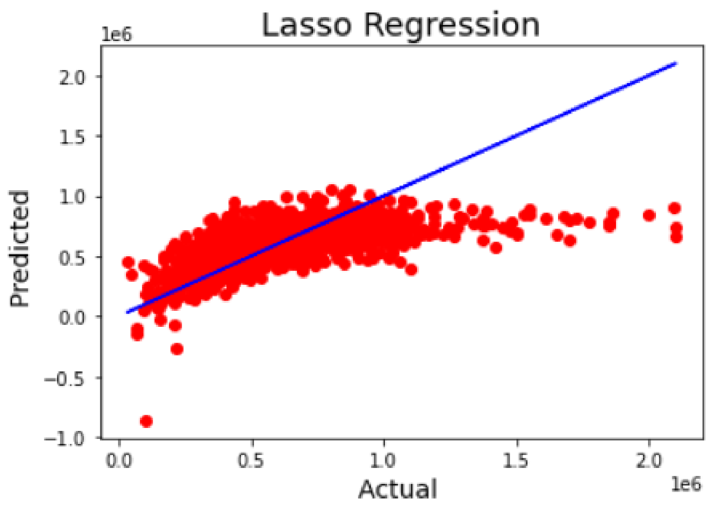
A correlation closer to +1 means a strong relationship in the positive direction, while -1 means a stronger relationship in the opposite direction.

A value closer to 0 means that there is not much of a relationship between the variables. R-squared is closely related to correlation

1. Lasso Regression

```
print('MAE:', metrics.mean_absolute_error(y_test, pred_ls))
print('MSE:', metrics.mean_squared_error(y_test, pred_ls))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred_ls)))
```

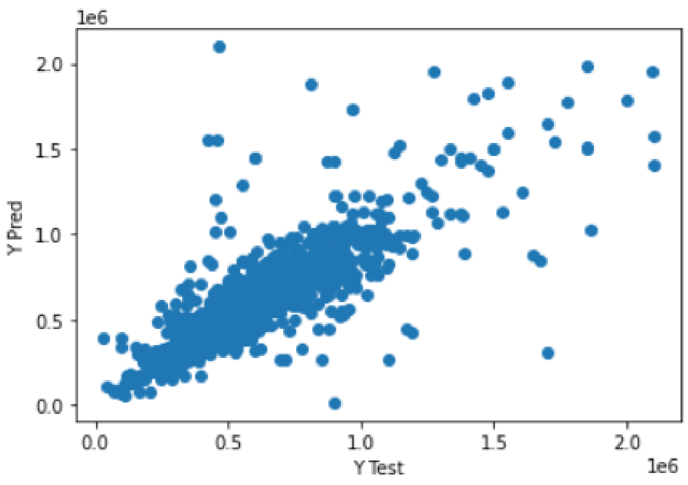
```
MAE: 164343.86474122625
MSE: 53627530571.68836
RMSE: 231576.187402091
```



2. Decision Tree Regressor

```
print('MAE:', metrics.mean_absolute_error(y_test, pred_dtr))
print('MSE:', metrics.mean_squared_error(y_test, pred_dtr))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred_dtr)))
```

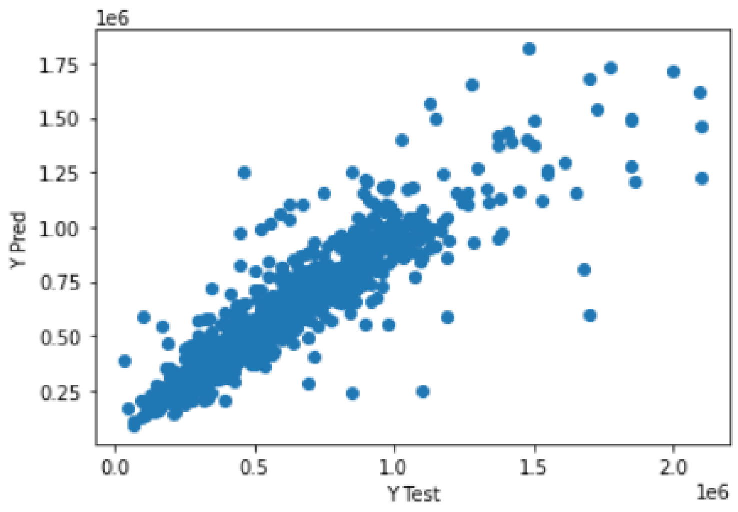
```
MAE: 93147.8097133758
MSE: 27905947465.540604
RMSE: 167050.7332086292
```



3. Random Forest Regressor

```
print('MAE:', metrics.mean_absolute_error(y_test, pred_rfr))
print('MSE:', metrics.mean_squared_error(y_test, pred_rfr))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred_rfr)))
```

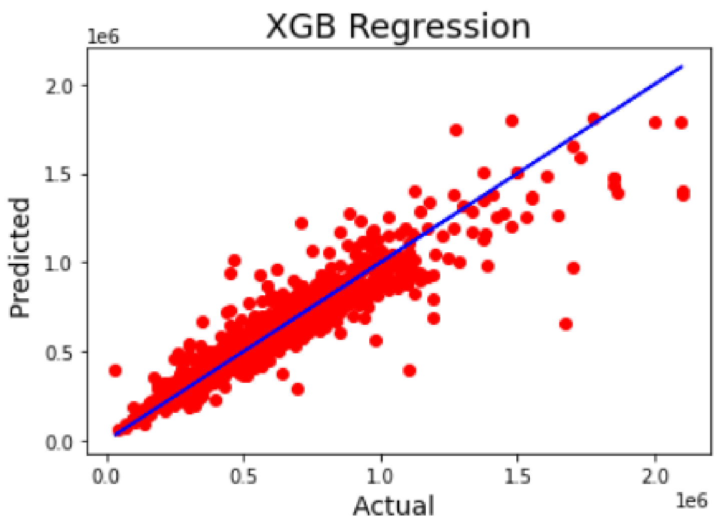
```
MAE: 74629.86048964968
MSE: 15843003619.866734
RMSE: 125868.99387802674
```



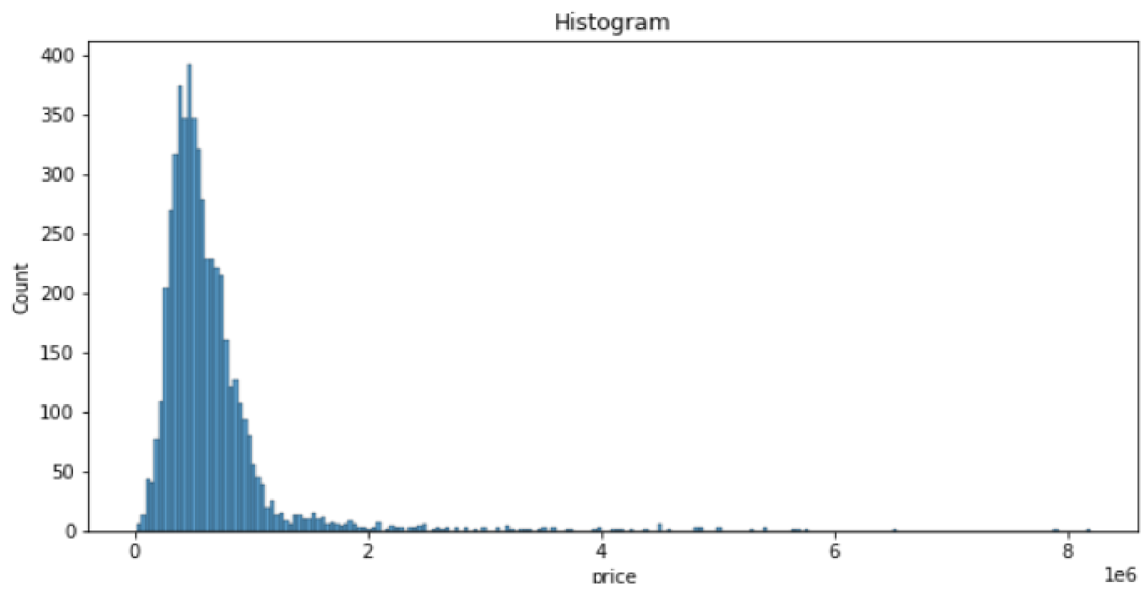
4. XGB Regressor

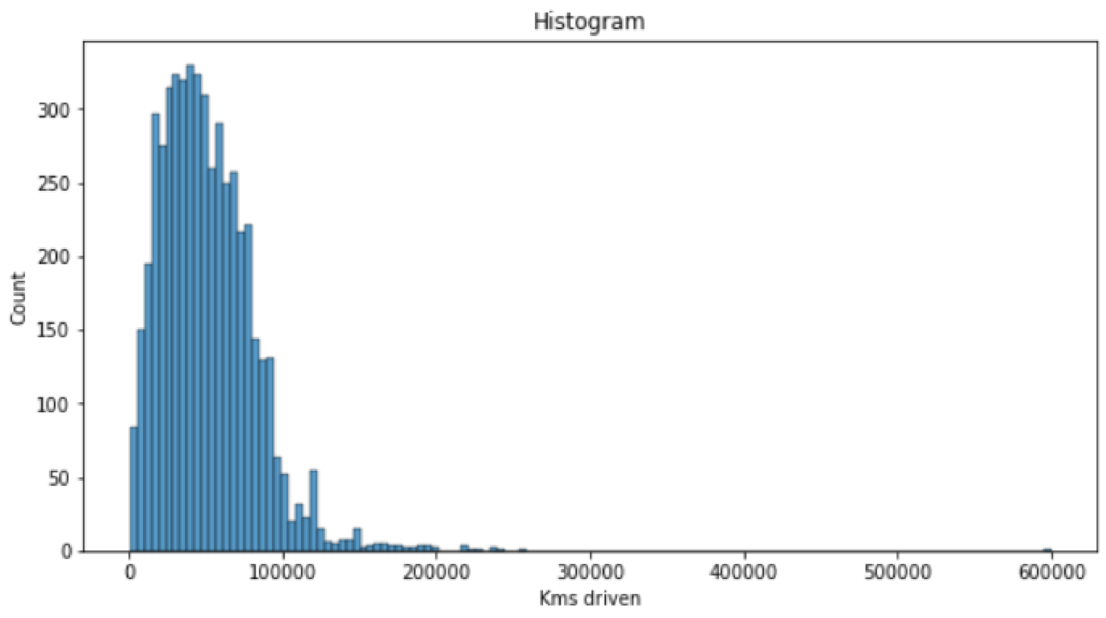
```
print('MAE:', metrics.mean_absolute_error(y_test, pred_xgb))
print('MSE:', metrics.mean_squared_error(y_test, pred_xgb))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred_xgb)))
```

```
MAE: 63891.28313159335
MSE: 11230699964.784185
RMSE: 105974.99688503976
```



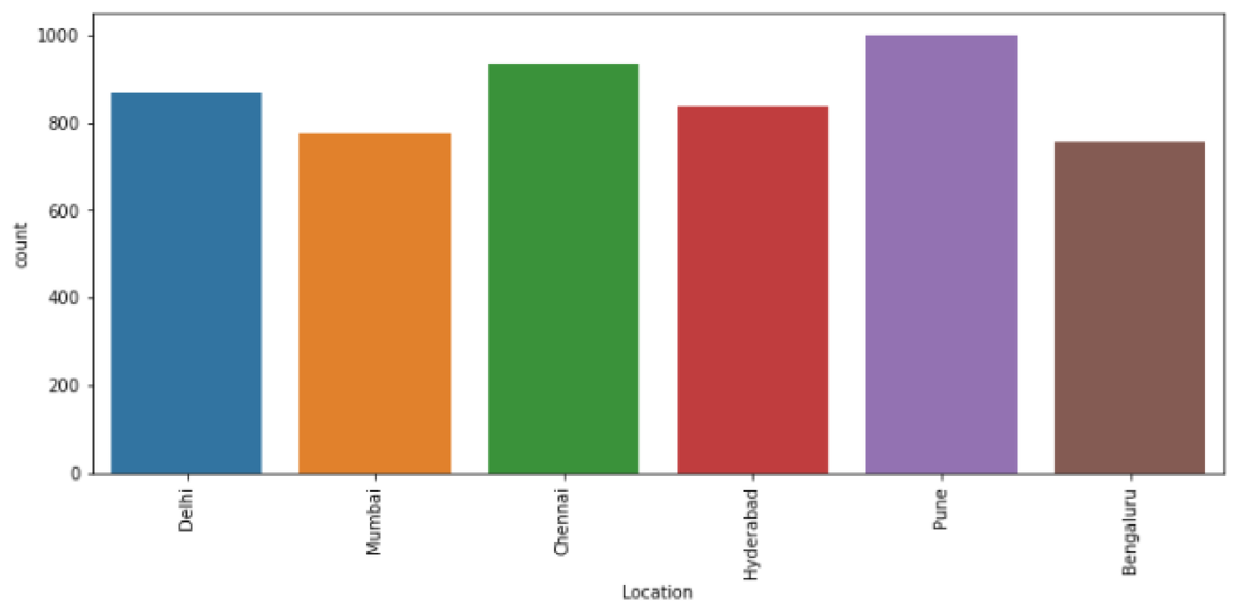
Visualizations



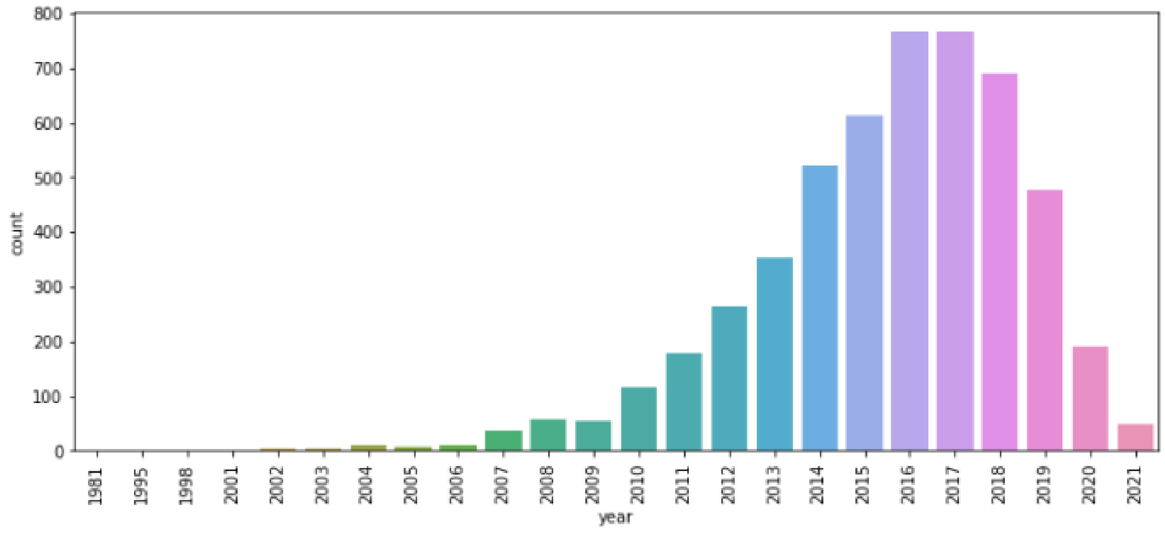


Observations:

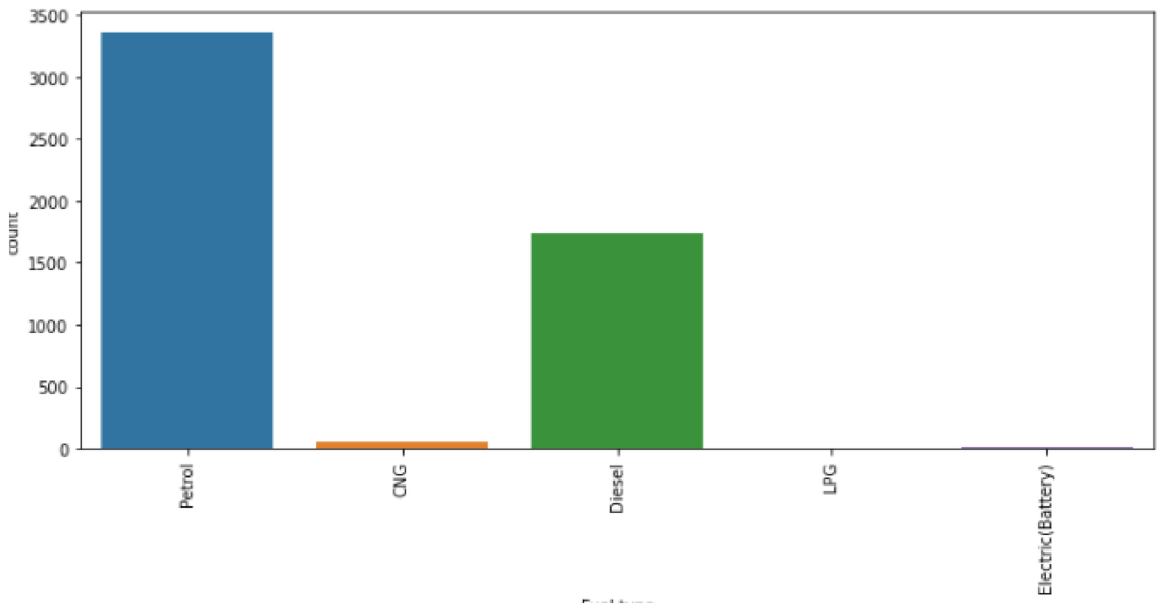
- Majority of the price is centred around 500,000
- Maximum distance driven by cars hovers around 45000 km



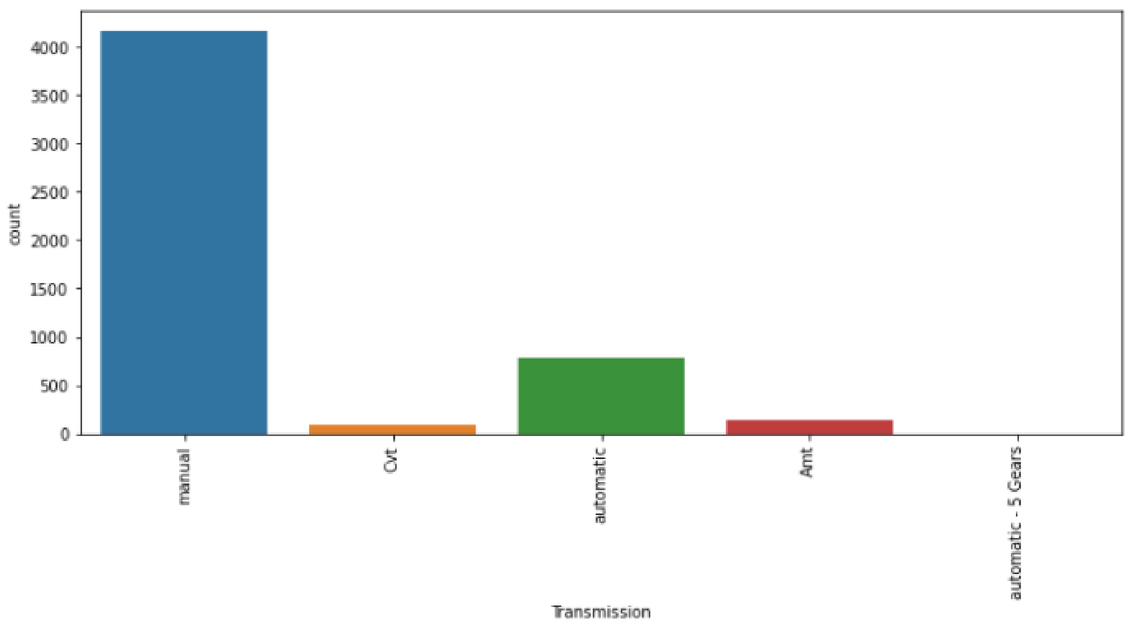
count
0
200
400
600
800
1000
Delhi
Mumbai
Chennai
Hyderabad
Pune
Bengaluru
Location

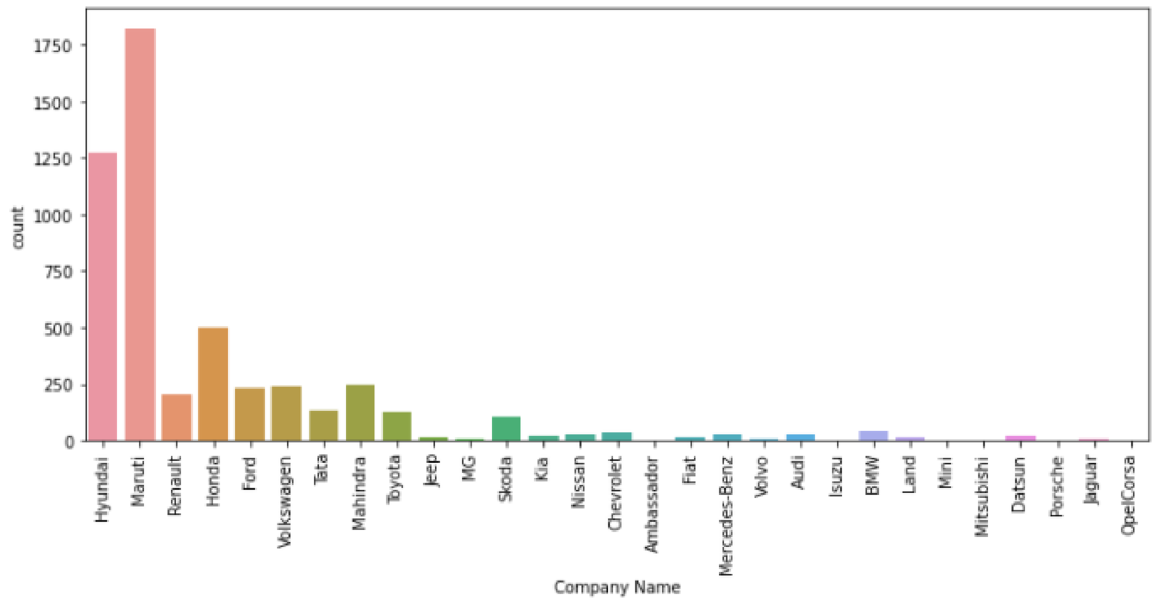


count
0
100
200
300
400
500
600
700
800
1981
1995
1998
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
year



count
0
500
1000
1500
2000
2500
3000
3500
Petrol
CNG
Diesel
LPG
Electric(Battery)

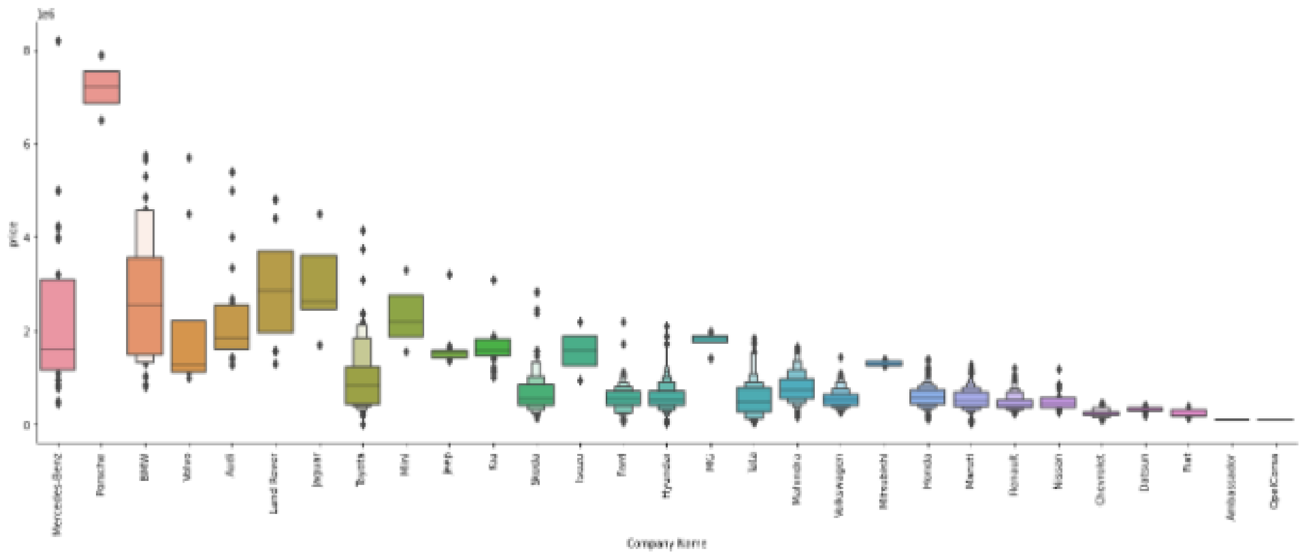




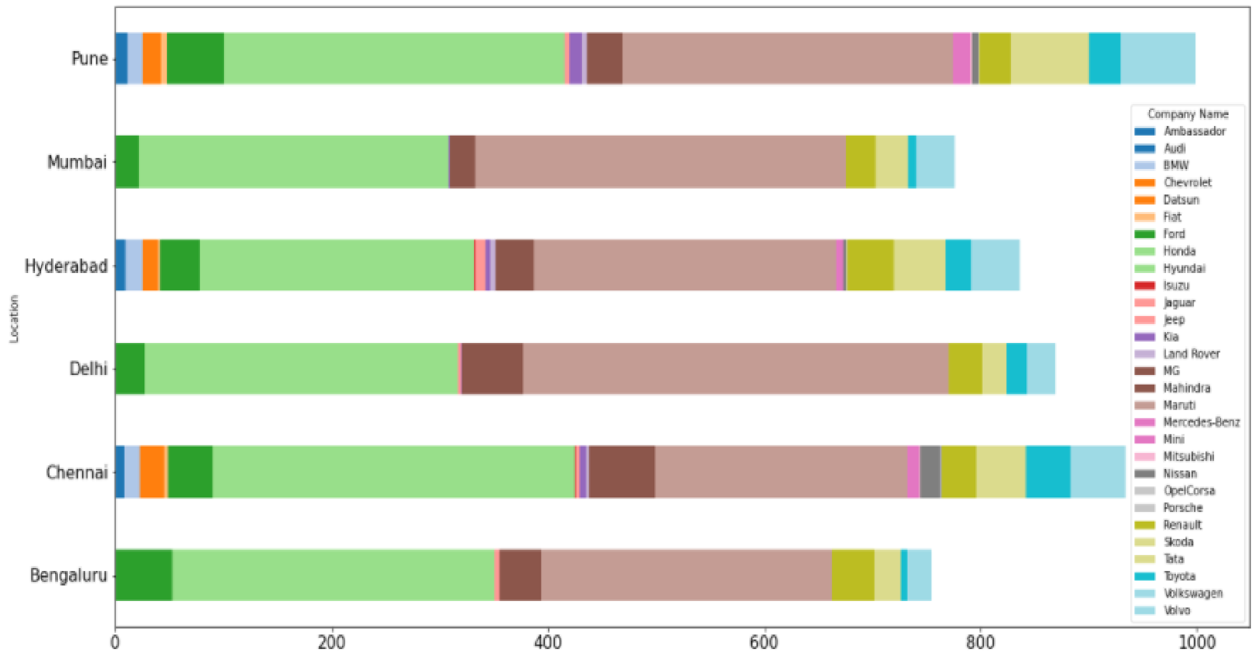
Observations:

- Out of the 5k cars, Maruti Suzuki and Hyundai are the most used car companies which has been put up for sale, indicating high penetration in the Indian market compared to its rivals.
- Manual Transmission is abundant as the cars tend to be cheaper which makes sense for the Indian market
- We have almost equal number of cars available from all the 5 major cities.
- Petrol cars have the highest quantity to be listed for sale, as petrol have an extra benefit compared to Diesel - it can be converted to run on CNG,

which is almost 50% cheaper than petrol and diesel and cleaner for the environment.
- Most cars are relatively new with majority belonging to 2014 to 2019



Observation:
- We can see majority of the companies have outlier prices.
-Luxurious brands will logically be expensive than commercial car brands, like Porsche is the one of the most expensive car companies globally.

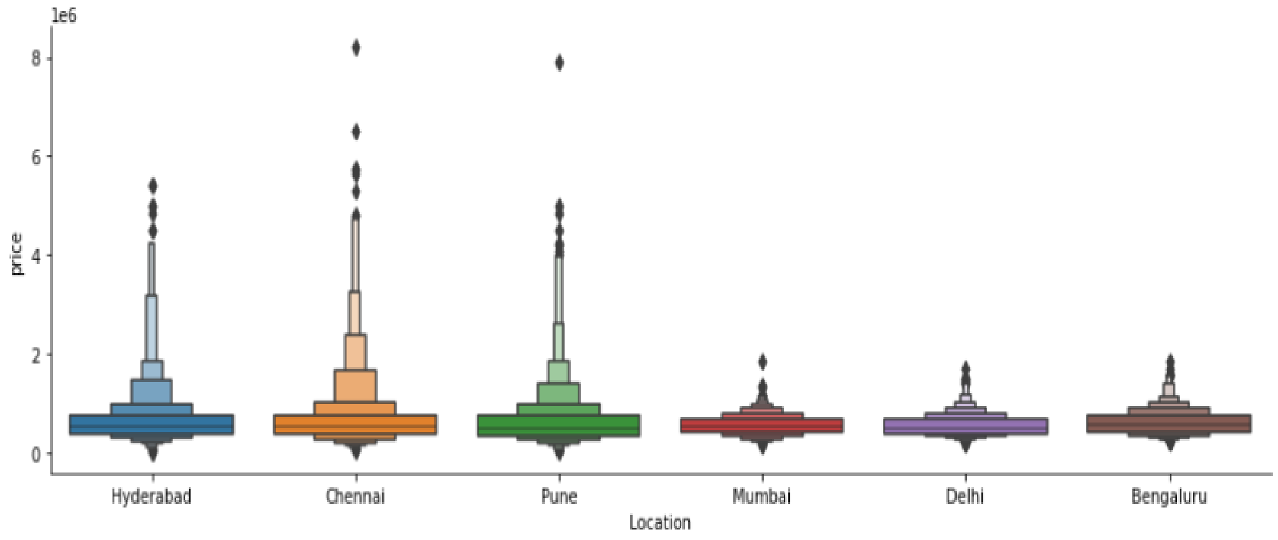


Observation:

- Chennai has more Hyundai's listed than Maruti's indicating the famous car company in the city.

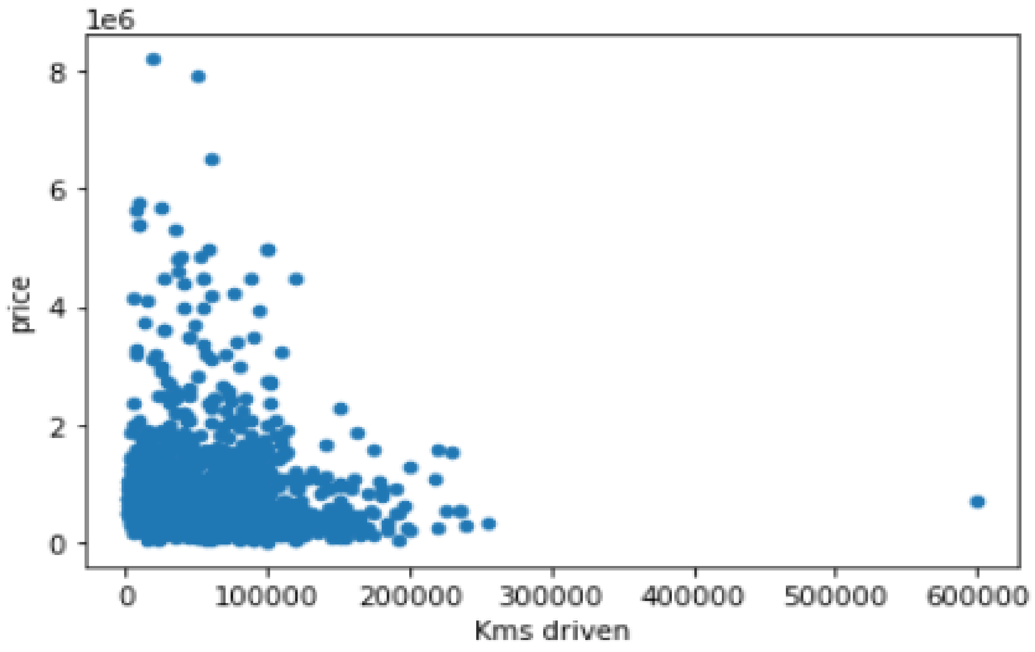
- Delhi has the largest number of Maruti's listed

- Chennai also tops the list in terms of Toyota's listed.



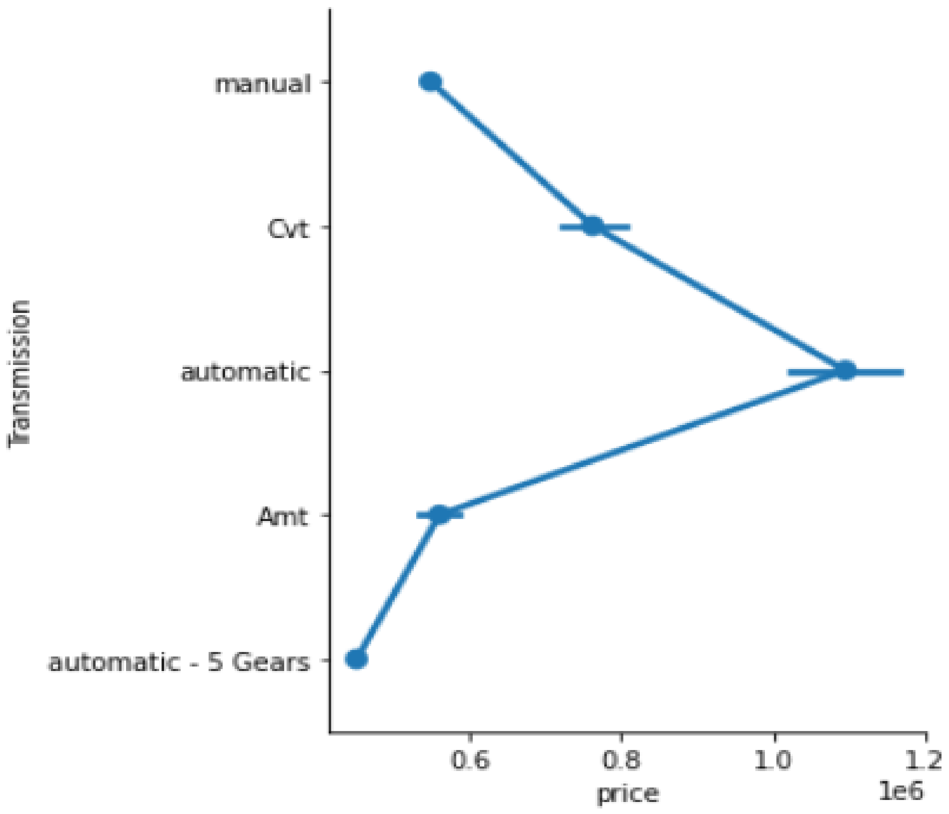
Observation:
-Mumbai, Delhi and Bengaluru have only inexpensive cars listed whereas the other 3 definitely have a lot of expensive cars listed as well.

-All the cities have almost equal average listed price.



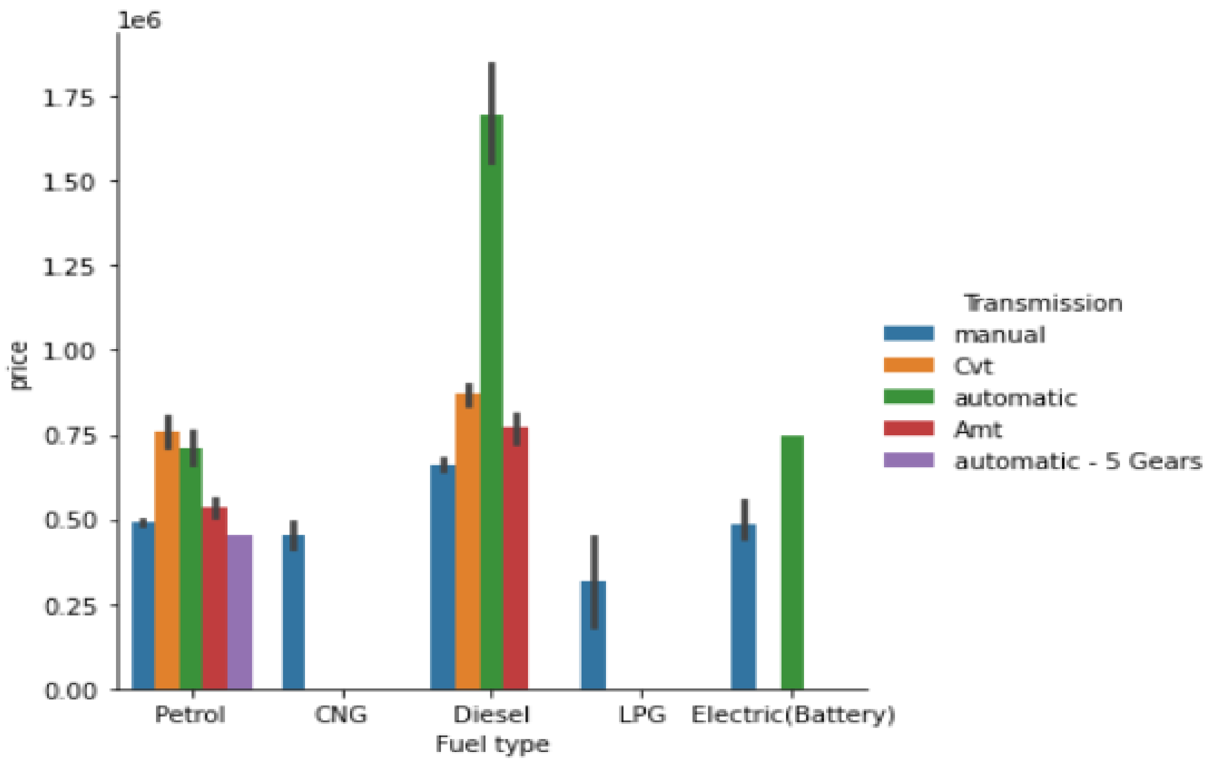
Observation:

- Lesser driven vehicles will obviously cost more which is what the graph depicts.



observation:

- Automatic cars are the most expensive with CVT coming in at second.



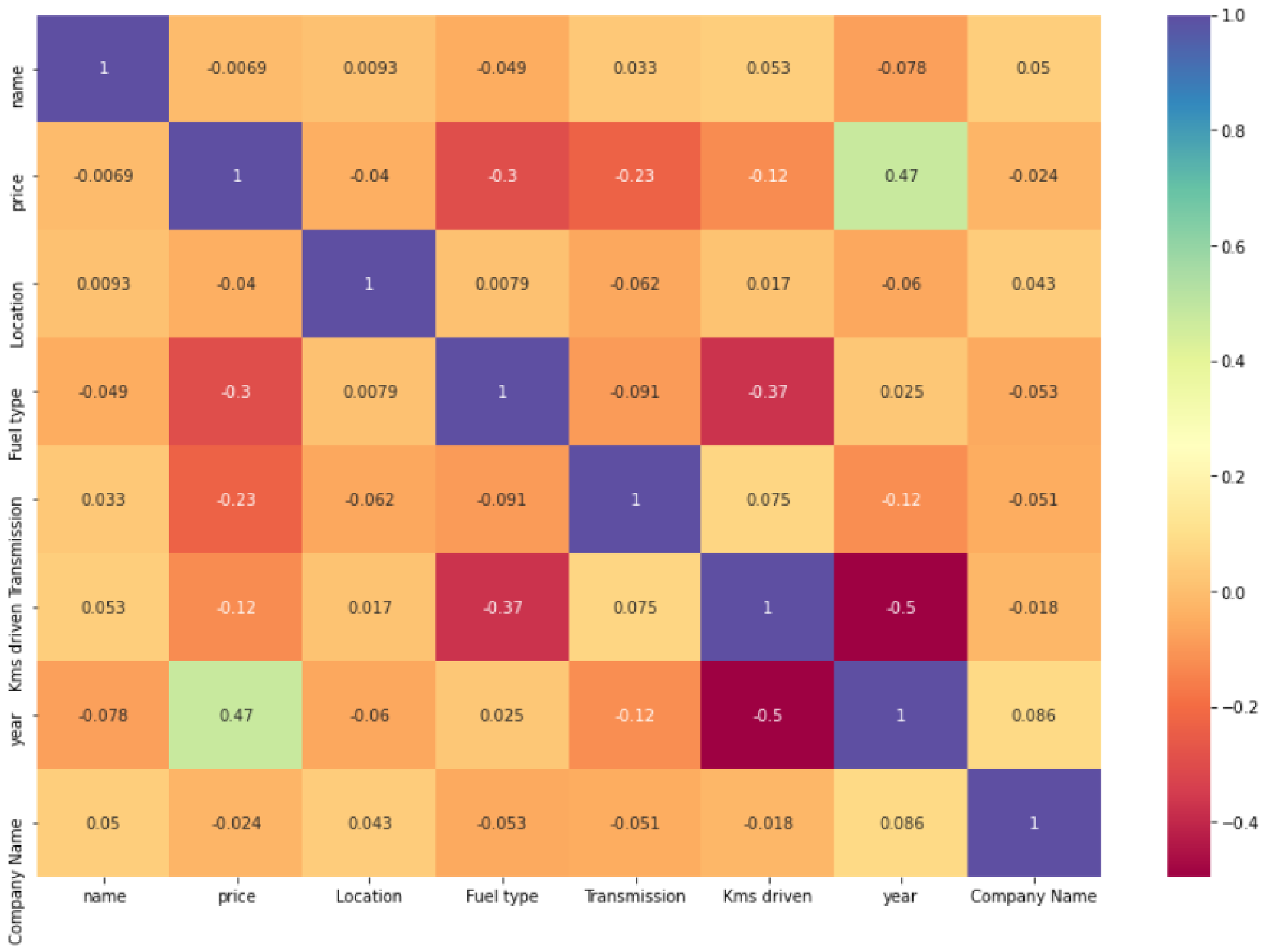
Observations:

- Manual cars with LPG are the cheapest combo available

- Automatic and Diesel is the highest priced option listed

- Electric cars have only manual or automatic transmission

- CVT transmission (manual with no clutch pedal) is the second highest in terms of price in Diesel variant, but cars having Petrol as their fuel type, it's the most expensive type of transmission.



	name	price	Location	Fuel type	Transmission	Kms driven	year	Company Name
name	1	-0.0069	0.0093	-0.049	0.033	0.053	-0.078	0.05
price	-0.0069	1	-0.04	-0.3	-0.23	-0.12	0.47	-0.024
Location	0.0093	-0.04	1	0.0079	-0.062	0.017	-0.06	0.043
Fuel type	-0.049	-0.3	0.0079	1	-0.091	-0.37	0.025	-0.053
Transmission	0.033	-0.23	-0.062	-0.091	1	0.075	-0.12	-0.051
Kms driven	0.053	-0.12	0.017	-0.37	0.075	1	-0.5	-0.018
year	-0.078	0.47	-0.06	0.025	-0.12	-0.5	1	0.086
Company Name	0.05	-0.024	0.043	-0.053	-0.051	-0.018	0.086	1

Observation:

- No multicollinearity can be observed as the values are below 0.6.

CONCLUSION

Key Findings and Conclusions of the Study

The observations explained above, clearly indicates the factors that influence the price of the car in each city. Business owners can easily determine the price of the product by deploying this model on the web.

Year is the most positively corelated variable whereas Kms driven is the most negatively correlated variable which makes sense as 2015 Honda city with 35000 kms will cost less than 2019 Honda city with 15000 kms.

Learning Outcomes of the Study in respect of Data Science

This dataset sheds light on the power of visualization and EDA. As the data is largely limited which makes it expectedly inconclusive to perform exploratory data analysis and reach any fruitful conclusions. This is first opportunity I have received to merge data mining and machine learning in a single project.

Limitations of this work and Scope for Future Work

The dataset is severely limited and can be expanded extensively with data mining. The number of features and datapoints is less. To have an accurate model, more data needs to be mined with increasing the computing capacity which can run computations rather quickly with assisting in hyperparameter tuning and visualizations