Car Price Prediction using Regression

Data Mining and Knowledge Discovery (Prof. Faiz Hamid)

Group 7:

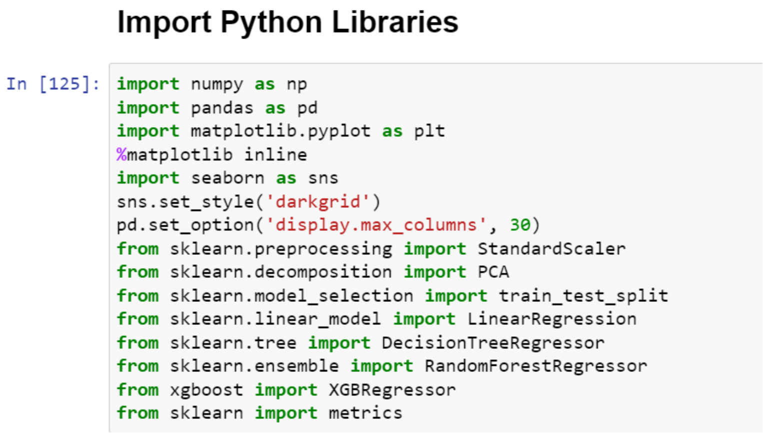
Abhay Singh (200013)	Dusane Atharv Sachin (200356)	Dibyojyotee Panda (200336)
Ayushi Singh (200258)	Harsh Vardhan Baudh (200423)	

Problem Statement

A Chinese automobile company Geely Auto aspires to enter the US market by setting up their manufacturing unit there and producing cars locally to give competition to their US and European counterparts

We are required to model the price of cars with the available independent variables. It will be used by the management to understand how exactly the prices vary with the independent variables. They can accordingly manipulate the design of the cars, the business strategy etc. to meet certain price levels. Further, the model will be a good way for management to understand the pricing dynamics of a new market.

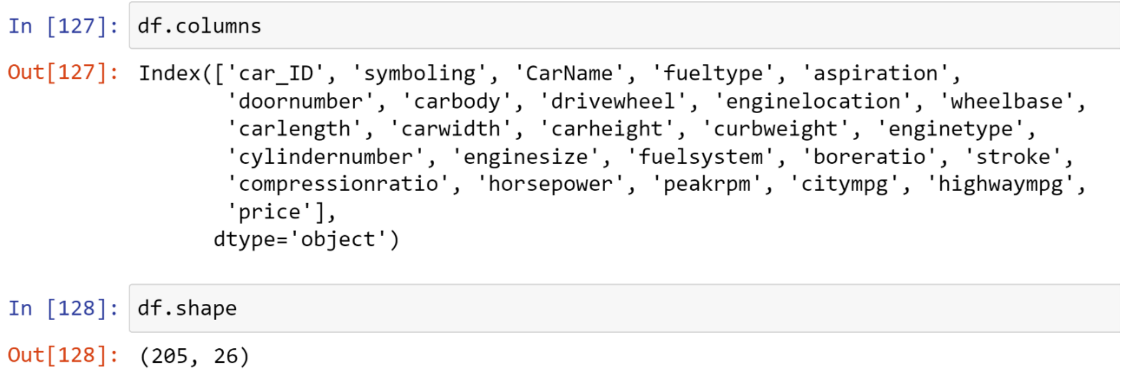
Setting Up Python Environment:



Import Python Libraries

```
In [125]: import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          %matplotlib inline
          import seaborn as sns
          sns.set_style('darkgrid')
          pd.set_option('display.max_columns', 30)
          from sklearn.preprocessing import StandardScaler
          from sklearn.decomposition import PCA
          from sklearn.model_selection import train_test_split
          from sklearn.linear_model import LinearRegression
          from sklearn.tree import DecisionTreeRegressor
          from sklearn.ensemble import RandomForestRegressor
          from xgboost import XGBRegressor
          from sklearn import metrics
```

Attributes: The series of attributes in the dataset can be found by using 'DataFrame.columns' in the python kernel. Attributes in the Telecom Churn dataset are as shown in the below screenshot from the notebook.
(*telecom is a copy of original dataframe object 'df')



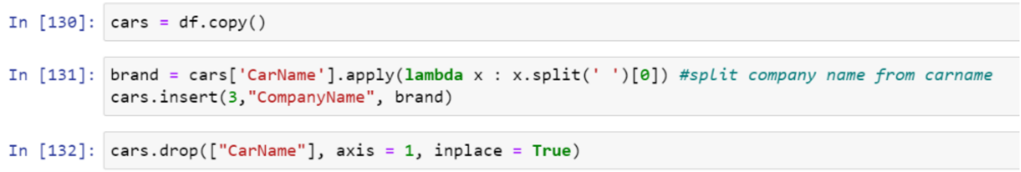
```
In [127]: df.columns

Out[127]: Index(['car_ID', 'symboling', 'CarName', 'fueltype', 'aspiration',
                 'doornumber', 'carbody', 'drivewheel', 'enginelocation', 'wheelbase',
                 'carlength', 'carwidth', 'carheight', 'curbweight', 'enginetype',
                 'cylindernumber', 'enginesize', 'fuelsystem', 'boreratio', 'stroke',
                 'compressionratio', 'horsepower', 'peakrpm', 'citympg', 'highwaympg',
                 'price'],
               dtype='object')

In [128]: df.shape

Out[128]: (205, 26)
```

EXPLORATORY DATA ANALYSIS

Data-Preprocessing

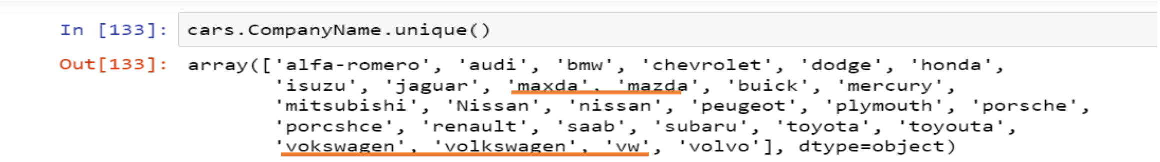
Data cleaning is fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. The given dataset contains 26 features and 205 objects.

1. Splitting company name from carname in order to study the market of brands as whole not individual cars.



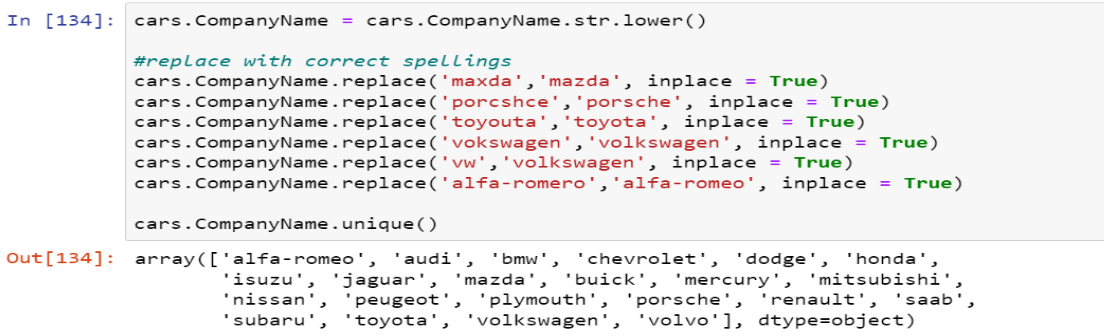
```
In [130]: cars = df.copy()

In [131]: brand = cars['CarName'].apply(lambda x : x.split(' ')[0]) #split company name from carname
          cars.insert(3,"CompanyName", brand)

In [132]: cars.drop(["CarName"], axis = 1, inplace = True)
```

2. Wrong spellings of company name e.g maxda and mazda are same, volkswagen,vokswagen and vw are same company. To correct this we replaced wrong names with correct ones.



```
In [133]: cars.CompanyName.unique()
Out[133]: array(['alfa-romero', 'audi', 'bmw', 'chevrolet', 'dodge', 'honda',
                 'isuzu', 'jaguar', 'maxda', 'mazda', 'buick', 'mercury',
                 'mitsubishi', 'Nissan', 'nissan', 'peugeot', 'plymouth', 'porsche',
                 'porcshce', 'renault', 'saab', 'subaru', 'toyota', 'toyouta',
                 'vokswagen', 'volkswagen', 'vw', 'volvo'], dtype=object)
```

Spelling errors in 'CompanyName' column

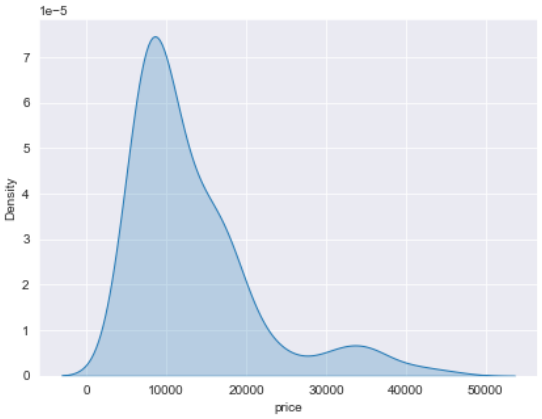


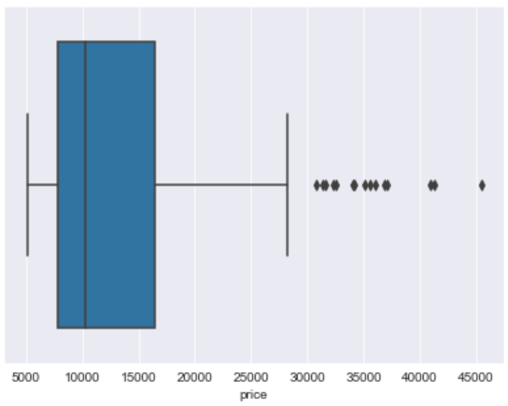
```
In [134]: cars.CompanyName = cars.CompanyName.str.lower()

          #replace with correct spellings
          cars.CompanyName.replace('maxda','mazda', inplace = True)
          cars.CompanyName.replace('porcshce','porsche', inplace = True)
          cars.CompanyName.replace('toyouta','toyota', inplace = True)
          cars.CompanyName.replace('vokswagen','volkswagen', inplace = True)
          cars.CompanyName.replace('vw','volkswagen', inplace = True)
          cars.CompanyName.replace('alfa-romero','alfa-romeo', inplace = True)

          cars.CompanyName.unique()
Out[134]: array(['alfa-romeo', 'audi', 'bmw', 'chevrolet', 'dodge', 'honda',
                 'isuzu', 'jaguar', 'mazda', 'buick', 'mercury', 'mitsubishi',
                 'nissan', 'peugeot', 'plymouth', 'porsche', 'renault', 'saab',
                 'subaru', 'toyota', 'volkswagen', 'volvo'], dtype=object)
```

3. No Duplicated or Null objects were found in the dataset.
4. **Feature Generation:** Created new feature '*fueleconomy'* with the help of existing features '*citympg' and* 'highwaympg'.

Understanding and Visualising Data

Price Distribution:



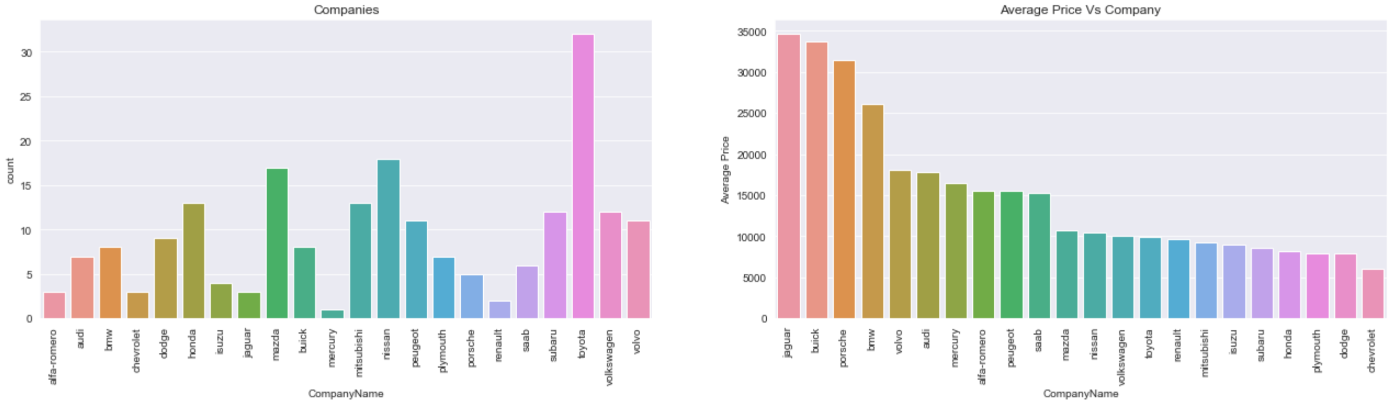


Numerical Discription of Price

Count:	205
Mean:	13276.71
Std:	7988.85
Min:	5118.00
25% :	7788.00
50% :	10295.00
75% :	16503.00
Max :	45400.00
IQR:	8715.00

Analyzing Categorical Variables:

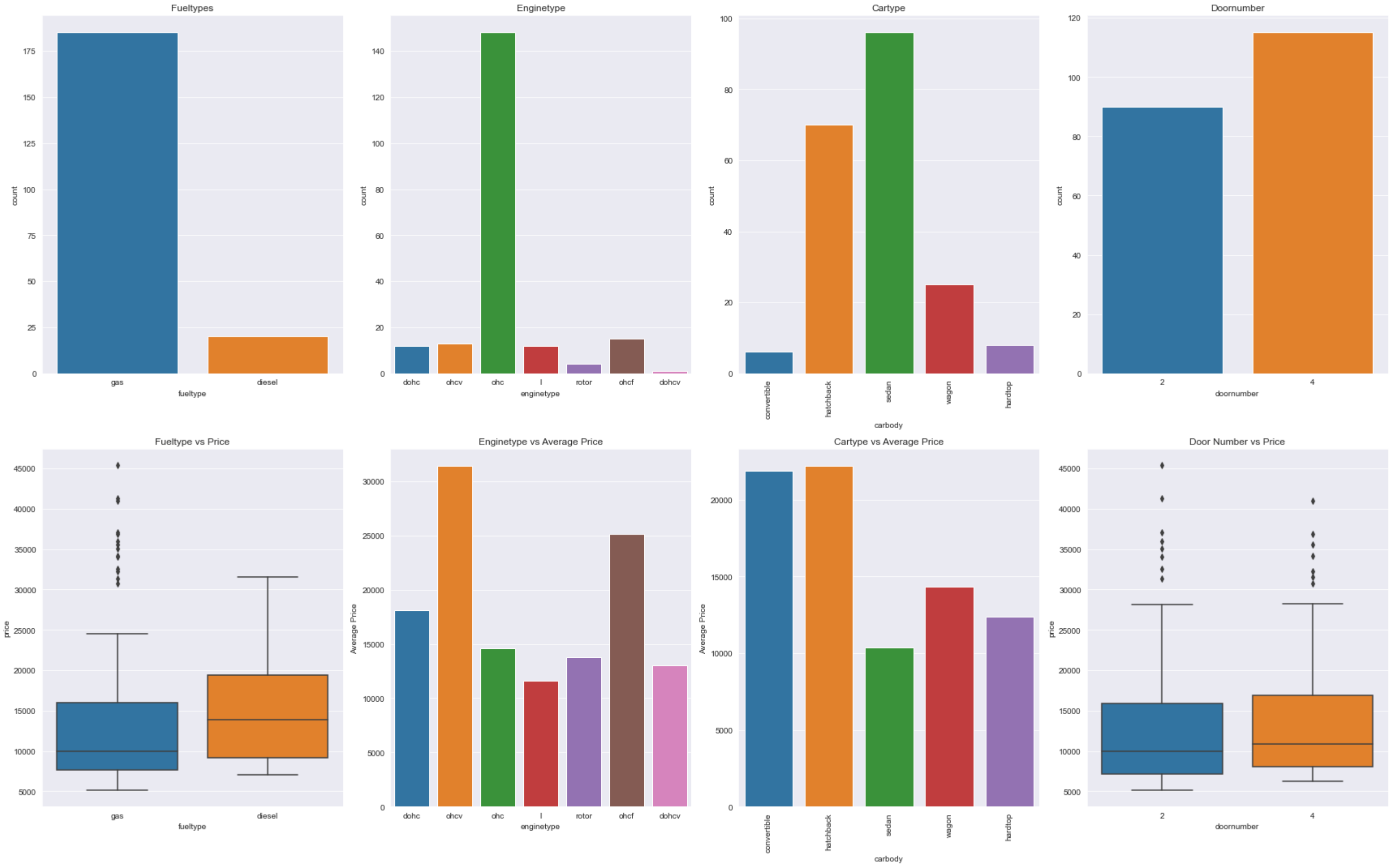
1. *Average Price of cars of each company:*



Inferences:

- `Toyota` makes the most buyed cars with `Average Price` around `10000`.
- `Jaguar`, `Buick`, and `Porsche` seems to have highest average price.

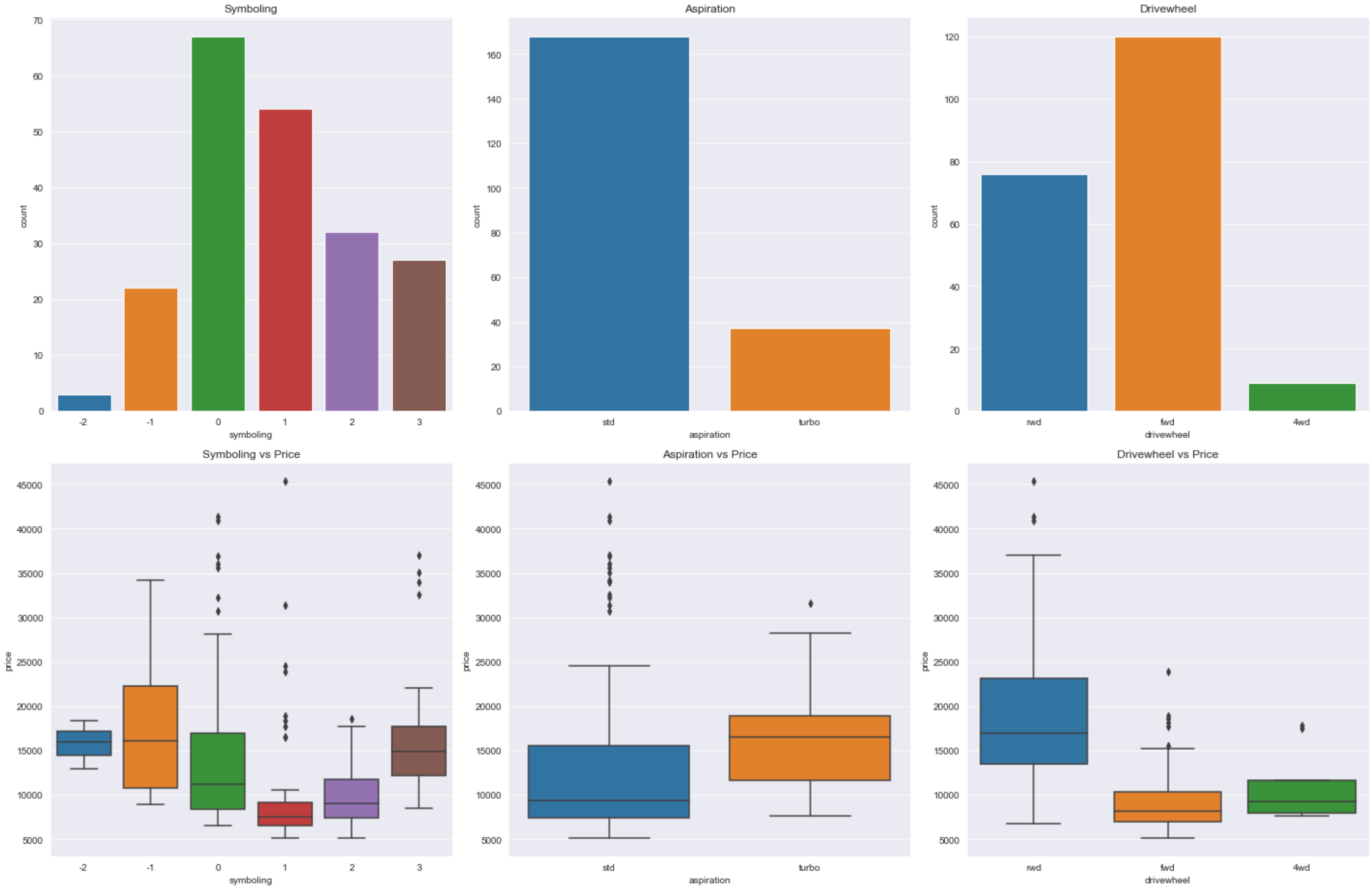
2. *Price vs fueltype, enginetype, cartype and door number*



Inference:

- Number of `gas` fueled cars are more than `diesel` fueled.
- Median price of `gas` fueled cars is lesser than `diesel` cars.
- `ohc` is the most preferred engine type with average price around 15000.
- `sedan` is the top car type prefered. `convertible` and `hatchback` cars have higher average price
- Cars with 4 `doors` were preferred over 2 doors

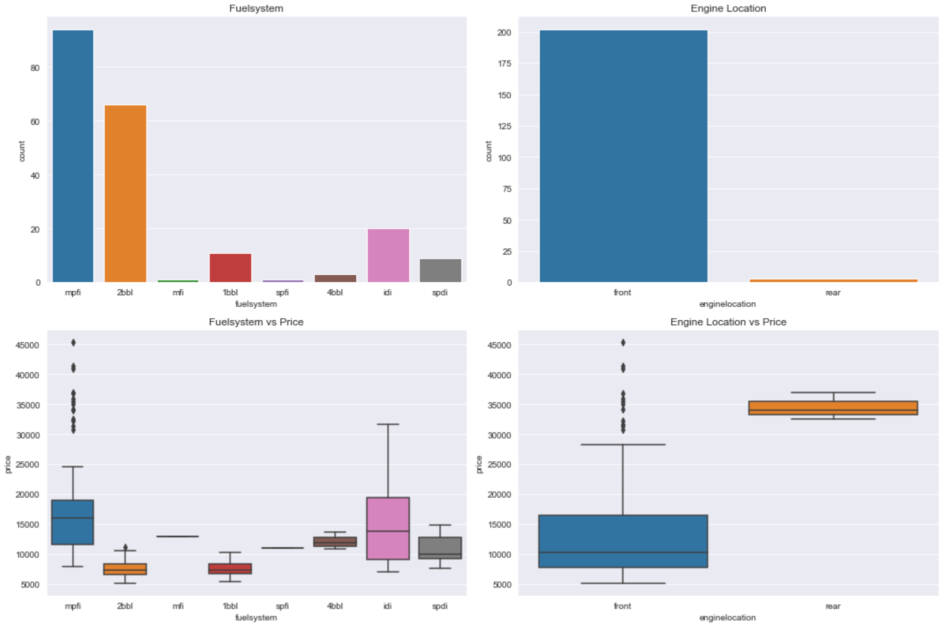
3. *Price vs Symboling, aspiration and drivewheel*



Inference:

1. Mostly sold cars have `symboling` values between `0` and `1`.
2. Cars with lower symboling values have higher price.
3. Aspiration with `turbo` have higher interquartile range of price than `std`
4. Most high range cars seems to prefer `rwd`(rear-wheel-drive) drivewheel

4. Fuelsysytem and enginelocation

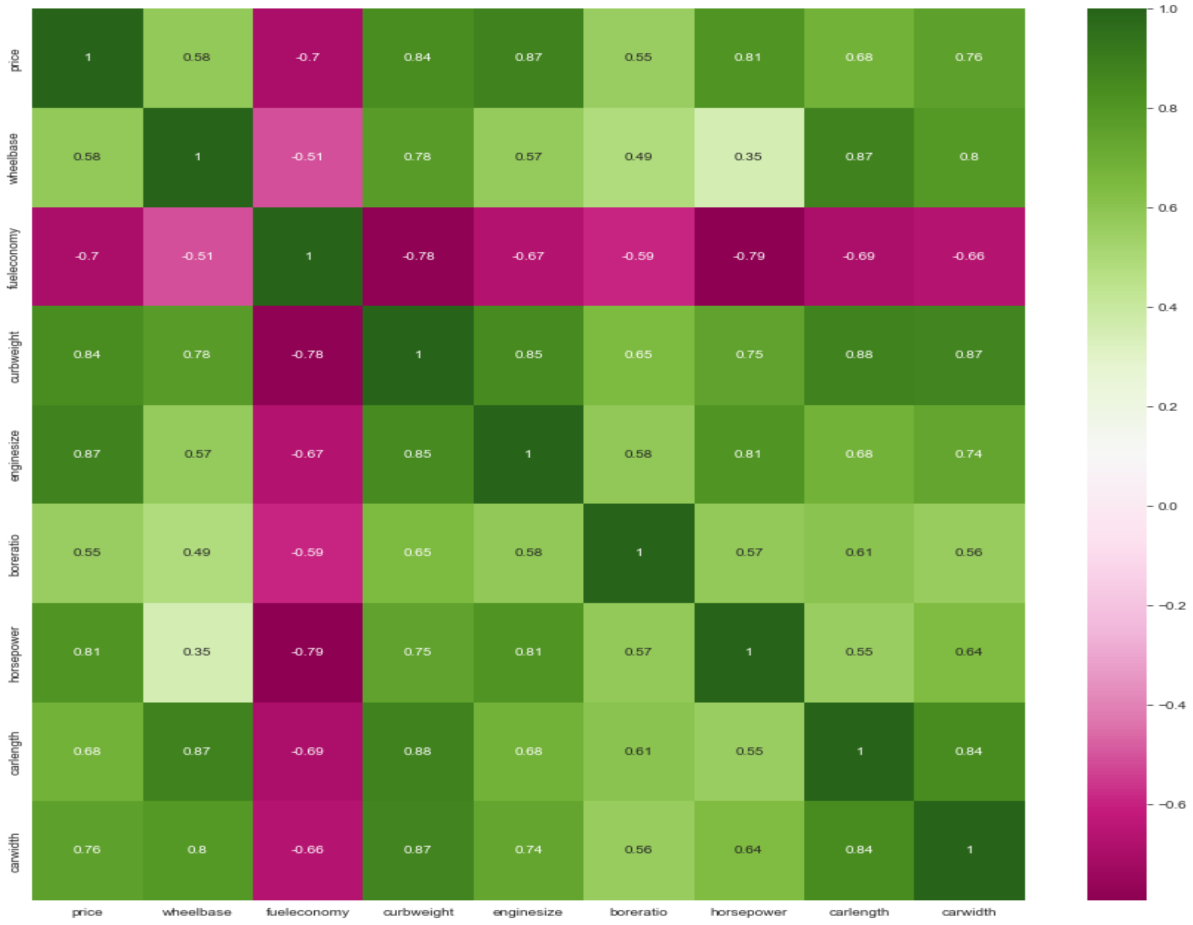


Inference :

- `mpfi` and `2bbl` are most common type of fuel systems. `mpfi` and `idi` having the highest price range. But there are few data for other categories to derive any meaningful inference.
- Most cars have enginelocation as `front`. Very few datapoints for further infeence to be made.

Analyzing Numerical Attributes

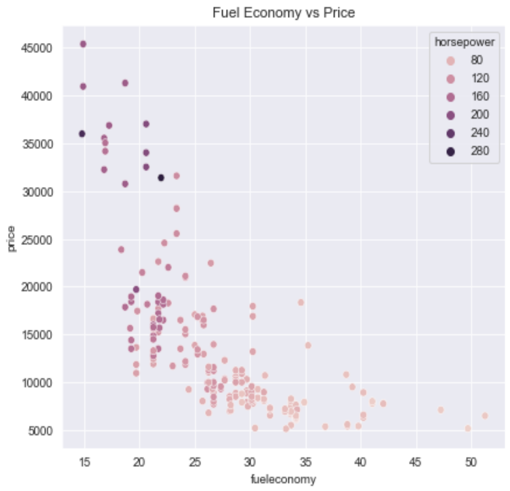
Correlation Heatmap:

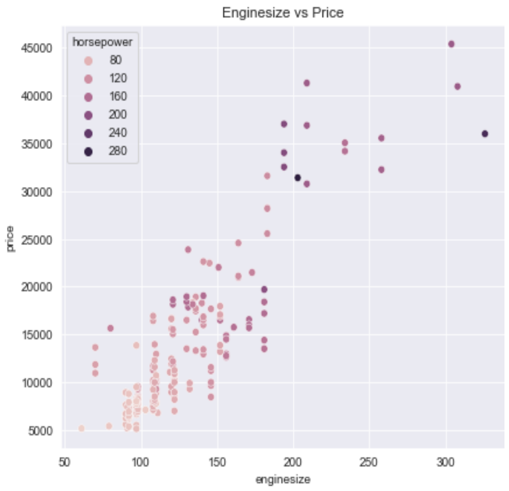


Inference:

- `enginesize`, `boreratio`, `horsepower`, `wheelbase` - have a significant positive correlation with price.
- `citympg`, `highwaympg` - have a significant negative correlation with price.
- `carwidth`, `carlength` and `curbweight` seems to have a poitive correlation with `price`.
- `carheight` doesn't show any significant trend with price.

Enginesize and horsepower with price





inference :

- `fueleconomy` has an obvious `negative correlation` with price and is significant.
- Cars with more `horsepower` have higher `price` and low `fueleconomy`.
- `enginesize` is highly correlated with `price`

Model Building and Evaluation

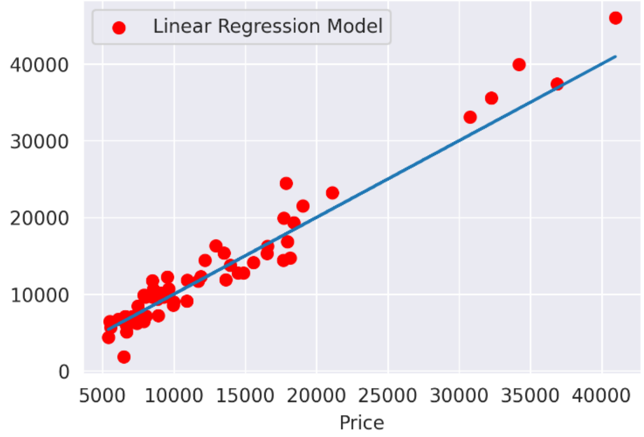
Before fitting data to our models, we need to follow below steps

- Getting Dummy variables to convert categorical attributes into numeric features.
- Scaling data using standard scaler to remove mean and making each feature variance to 1.
- Performing Dimensionality Reduction using Principal Component Analysis.
- Splitting data into 70% training and 30% test set.

1: Linear Regression Model

Metrics:

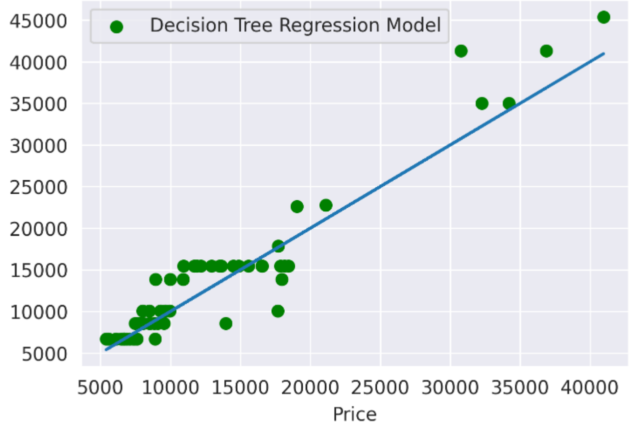
```
Mean Absolute Error: 1613.5
Mean Squared Error: 4443660.48
Root Mean Squared Error: 2108.0
Mean Absolute Percentage Error: 0.14
R2 score: 0.94
```



2: Decision Tree Regression Model

Metrics:

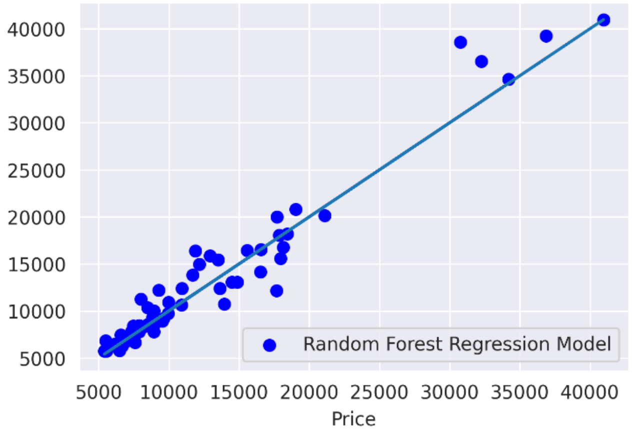
```
Mean Absolute Error: 1813.25
Mean Squared Error: 7235251.61
Root Mean Squared Error: 2689.84
Mean Absolute Percentage Error: 0.14
R2 score: 0.91
```



3: Random Forest Regression Model

Metrics:

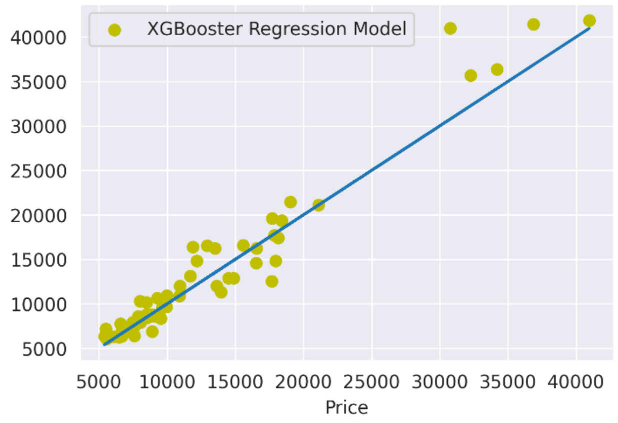
```
Mean Absolute Error: 1281.13
Mean Squared Error: 3812013.65
Root Mean Squared Error: 1952.44
Mean Absolute Percentage Error: 0.1
R2 score: 0.95
```



4: Extreme Gradient Boosting (XGBRegression)

Metrics:

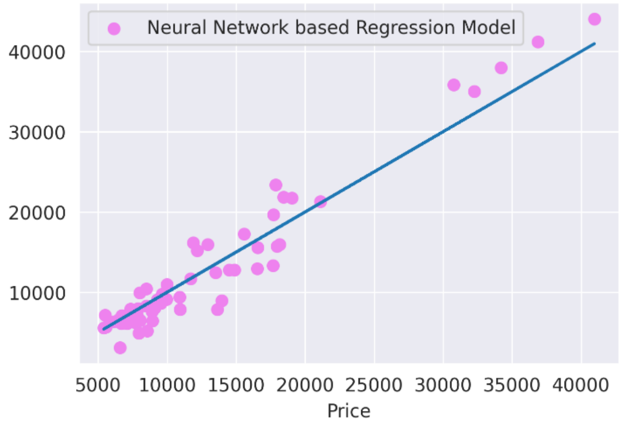
```
Mean Absolute Error: 1352.34
Mean Squared Error: 4629922.31
Root Mean Squared Error: 2151.73
Mean Absolute Percentage Error: 0.1
R2 score: 0.94
```



5: Neural Network based Regression Model

Metrics:

```
Mean Absolute Error: 1825.35
Mean Squared Error: 5721604.3
Root Mean Squared Error: 2391.99
Mean Absolute Percentage Error: 0.15
R2 score: 0.93
```



Result Interpretation:

- This problem statement was identified as Regression problem.
- Used several regression models to predict the price of the cars with least error.
- The Multiple Linear Regression, Decision Tree Regression, Random Forest Regression and XGBRegression performs very well on the training as well as test sets with R2 regression scores of 0.94, 0.91, 0.95 and 0.94 respectively.
- Among all models the Random Forest Regression technique performs best, but the difference is not much. Any of the three models can be used by the company.
- We also tried regression using Sequential Neural Networks, produced output has a Root mean square error value of 2391.99 and R2 score of 0.93 which is also a good score.