PROJECT REPORT ON

"CAR PRICE PREDICTION"

SUBMITTED BY

Ankit Dadarwala

ACKNOWLEDGMENT

I would like to express my special gratitude to "Flip Robo" team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Ms. Sapna Verma , she is the person who has helped me in tickets to get out of all the difficulties I faced while doing the project. And also Mr. Shubham Yadav (SME Flip Robo) who has inspired me in so many aspects and also encouraged me a lot with his valuable words and with his unconditional support I have ended up with a beautiful Project.

A huge thanks to my academic team "Data trained" who are the reason behind what I am today. Last but not least my parents who have been my backbone in every step of my life. And also thank you for many other persons who has helped me directly or indirectly to complete the project.

INTRODUCTION

1.1 Business Problem Framing:

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model

1.2 Conceptual Background of the Domain Problem

The digital revolution is disrupting used-car retailing—for the better. This new wave of digital retailing represents more than technology alone because it focuses a spotlight on the importance of the customer experience in the used-car-buying process. As revealed by our proprietary customer research, online providers are beginning to dilute traditional used-car dealers' positions and drive growth by empowering digitally savvy customers via three major capabilities:

- Complete end-to-end purchasing capabilities
- Extensive vehicle data and photos, along with effective search tools
- Unique delivery options

With the rise of digital players and potential incumbent-dealer consolidation on the horizon, the evolving market will feature new threats and opportunities for players trying to capture value in an already competitive environment. Furthermore, while the ways customers purchase vehicles are changing, it's also true that the needs of used-car buyers differ far more than those of new-vehicle purchasers. As a result, to generate a uniformly distinctive and differentiating customer experience, all used-car retailers must identify their target customer segments and rapidly develop the best approaches among a growing array of available options.

1.3 Review of Literature

The second-hand car market has continued to expand even as the reduction in the market of new cars. According to the recent report on India's pre-owned car market by Indian Blue Book, The second-hand car market has created the business for both buyers and sellers. Most of the people prefer to buy the used cars because of the affordable price and they can resell that again after some years of usage which may get some profit. The price of used cars depends on many factors like fuel type, colour, model, mileage, transmission, engine, number of seats etc., The used cars price in the market will keep on changing. Thus, the evaluation model to predict the price of the used cars is required.

1.4 Motivation for the Problem Undertaken

I have to model the used car price with the available independent variables. Further, the model will be a good way for the management to understand the price of car based on different variables. The **relationship between predicting price and the affecting factors** is an important motivating factor for predicting car price model.

Analytical Problem Framing

2.1 Mathematical/ Analytical modelling of the Problem

In this particular problem I had continues variables as my target column and it was having Price of the different models and company cars. So clearly it is a Regression base problem and I have to use all regressor algorithms while building the model. There were 88 null values in Transmission column in the dataset. To get better insight on the features I have used plotting like distribution plot, bar plot, Pie plot and count plot. With these plotting I was able to understand the relation between the features in better manner. Also, I found skewness in the dataset so I removed skewness using log method. I have used all the linear regression and Tree based algorithms while building model then tunned the best model and saved the best model. At last I have predicted the price using saved model.

2.2 Data Sources and their formats

The data was collected from online car retailers' websites like (cars24, Olx, CarDekho) as per instruction given by internship company – Flip Robo technologies in Data frame CSV format. It is hereby given to us for this exercise. In order to improve for our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

I am collecting data from Cars24.com in which I select 25 cities used car data, My dataset was having 4041 rows and 10 columns including target. In this parti cular dataset I have object and integer types of data. The information about fea tures is as follows.

Features Information:

1. Transmission:→ categorical feature with Automatic and Manual type val ues
2. Variant:→ categorical feature with different type of engine type values Li ke (LXI,VXI,VDI,ZDI,1.5L,2.0L etc.)
3. Km driven:→ int base column in that cars driven km show
4. Owner:→ categorical feature with value of 1st,2nd, etc..
5. Fuel type:→ In this category of fuel type like petrol, diesel..
6. Price:→ Target column int base
7. City:→ list of different cities of India
8. Mfg. year:→ manufacturing year of cars
9. Company name:→ List of the cars brand name
10. Model name:→ list of different car models of the different company

2.3 Data Pre-processing Done

1. As a first step I have imported required libraries and I have imported the dataset which was scrap and save in csv format.
2. Then I did all the analysis like checking shape, unique, value counts, info, duplicates value null values etc.
3. Then while looking into the value counts, I found some columns with name, model name which have insight details so replace str and create new columns with information
4. While checking for null values I found null values in the dataset.
5. I have also dropped Unnamed:0, column as I found they are useless.

2.4 Data Inputs- Logic- Output Relationships

- ✓ Since I had all numerical columns I have plotted distribution plot to see the distribution of each column data.
- ✓ Used Pie plot for value counts to show the highest category of values
- ✓ Used count plot as compare with price to show insight of the data
- ✓ Box plot to show the range of the cat price
- ✓ Line plot give the details of car price with respect to past to present years
- ✓ Heatmap to show the correlation with target and also inter correlation with other columns.

2.6 Hardware and Software Requirements and Tools Used

- ✓ Device name: HP Pavilion
- ✓ Processor: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz
- ✓ RAM: 8.00 GB
- ✓ System type: 64-bit operating system, x64-based processor
- ✓ Jupyter NoteBooks Version : 6.4.3
- ✓ Python3 vesrion : 3.8.8

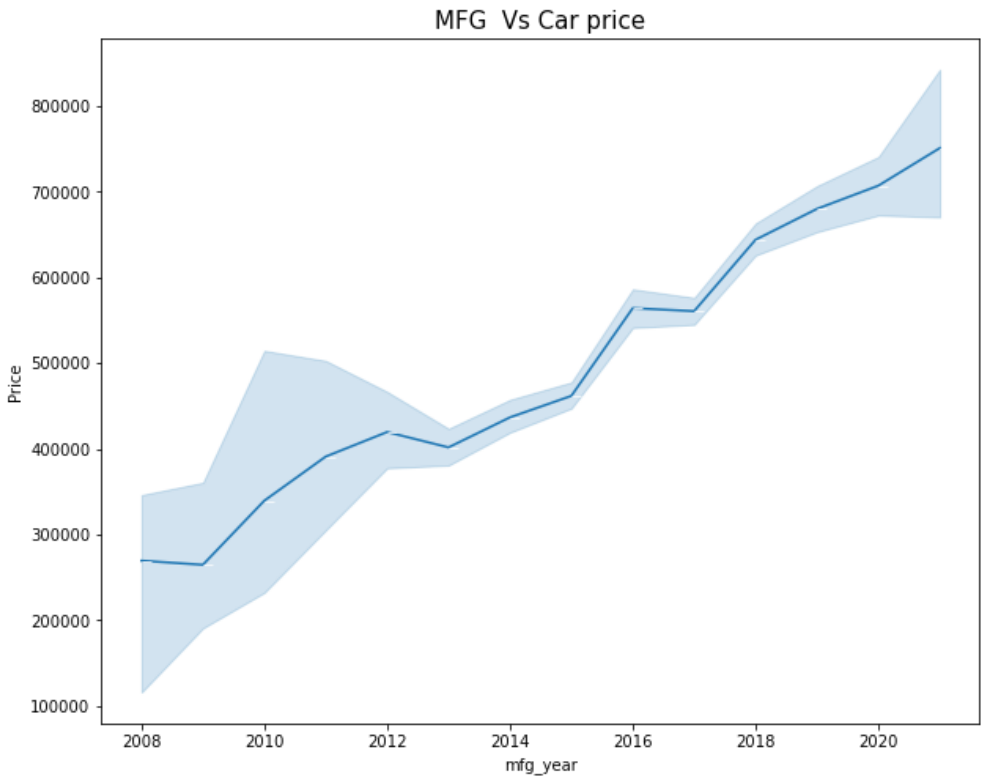
❖ **Libraries required** :-

- ✓ Import pandas as pd
- ✓ Import numpy as np
- ✓ Import matplotlib.pyplot as plt
- ✓ Import seaborn as sns
- ✓ Import warings
- ✓ From sklearn.model_selection import train_test_split,GridSearchCV
- ✓ from sklearn.preprocessing import StandardScaler
- ✓ from sklearn.ensemble import RandomForestRegressor
- ✓ from sklearn.ensemble import AdaBoostRegressor
- ✓ from sklearn.ensemble import GradientBoostingRegressor
- ✓ from sklearn.ensemble import BaggingRegressor
- ✓ from sklearn.tree import DecisionTreeRegressor

- ✓ from sklearn.svm import SVR
- ✓ from sklearn.neighbors import KNeighborsRegressor
- ✓ from linear_model import LinearRegression,SGDClassifier,Lasso,Ridge
- ✓ from xgboost import XGBRegressor
- ✓ from sklearn.metrics import mean squared error
- ✓ from sklearn.metrics import mean absolute error
- ✓ from sklearn.metrics import explain variance score
- ✓ from sklearn.model_selection import cross val score

Data Analysis and Visualization

3.1 Identification of possible problem-solving approaches (methods)



- As show in Line graph indicate the market of used cars price are increasing by years of manufacturing.
- Last Four years from 2016 price is rapidly increasing.

- Due to Impact of COVID-19 online car retailing business grow more by last two years, In that customer get best affordable price car in budget also get all related car details from websites
- Availability of all kind financial services from the site it is easy to buy.

3.2 Testing of Identified Approaches (Algorithms)

Since continues was my target and it was a Regression column , so this particular problem was Regression problem. And I have used all Regressor algorithms to build my model. By looking into the difference of R2 score score and cross validation score I found XGBOOST Regressor as a best model. Also to get the best model we have to run through multiple models and to avoid the confusion of overfitting we have go through cross validation and select best random state. Below are the list of Regressor algorithms I have used in my project.

1. Linear Regression
2. Lasso
3. Ridge
4. SGD Regressor
5. XGB Regressor
6. Decision Tree Regressor
7. SVM Regressor
8. KNeighbors Regressor
9. Random Forest Regressor
10. Gradient Boosting Regressor
11. Bagging Regressor
12. AdaBoost Regressor

3.3 Key Metrics for success in solving problem under consideration

- R-square is a comparison of residual sum of squares *(SSres)* with total sum of squares*(SStot)*. Total sum of squares is calculated by summation of squares of perpendicular distance between data points and the average line.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

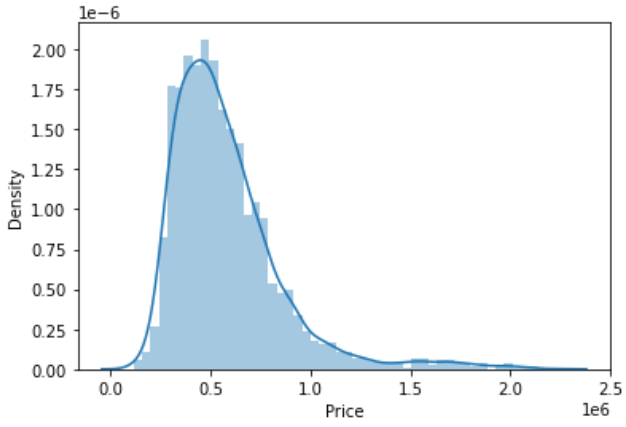
- Where *SSres* is the residual sum of squares and *SStot* is the total sum of squares.
- Absolute mean squared error metrics, **Absolute Error is the amount of error in your measurements.** It is the difference between the measured value and "true" value.

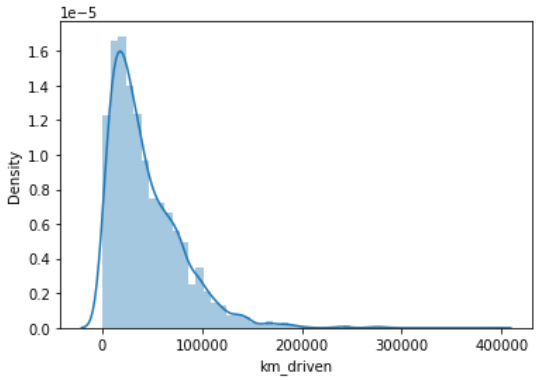
$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{x}|$$

- R-square is the main metric which I will use in this regression analysis.
- **Cross val score**: To run cross-validation on multiple metrics and also to return train scores, fit times and score times. Get predictions from each split of cross-validation for diagnostic purposes. Make a scorer from a performance metric or loss function.

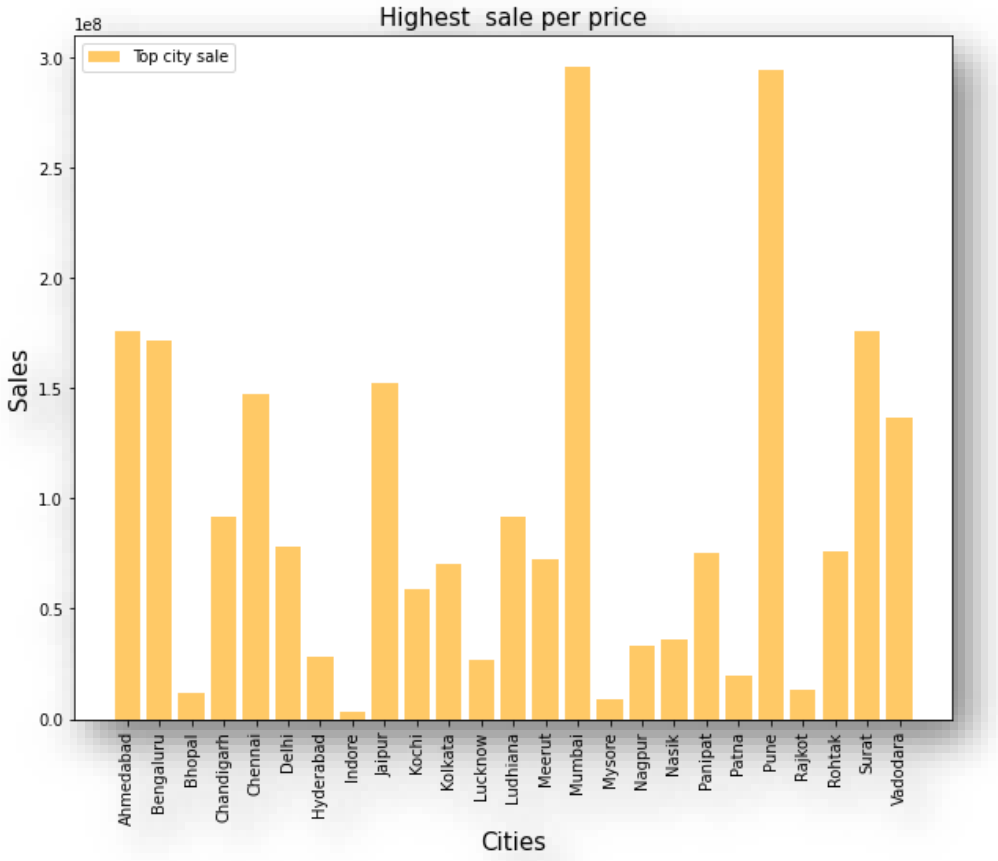
3.4 Visualizations

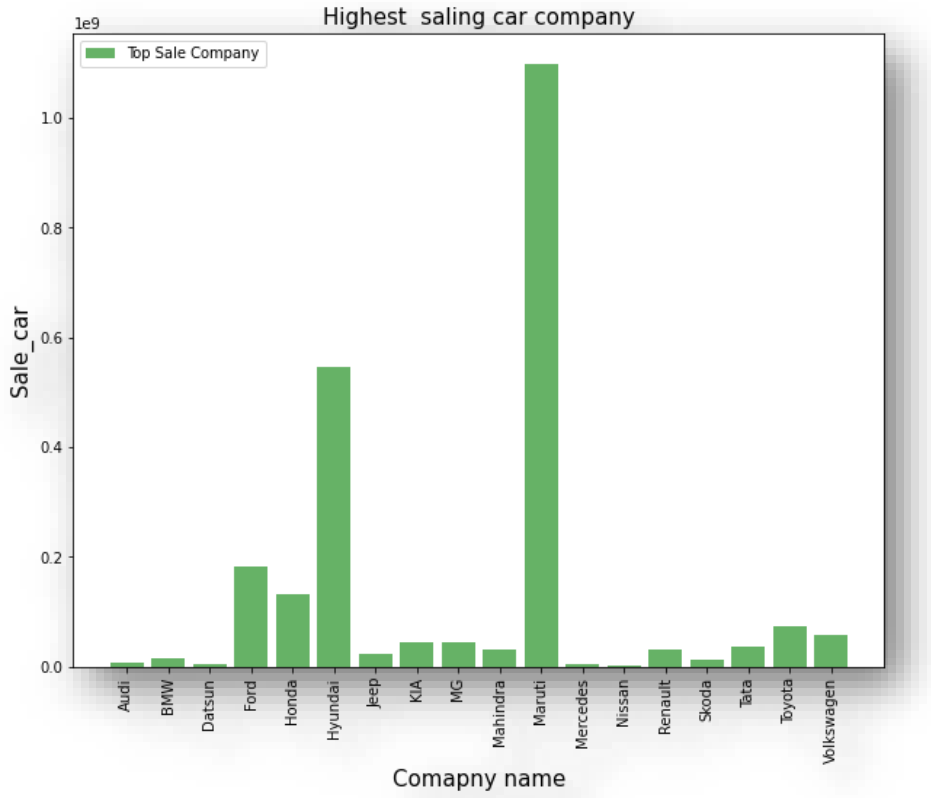
I have used bar, pie, count, line, distribution, heatmap plots to see the relation of independent feature with target and I have used 2 types of plots for numerical columns one is distribution plot for univariate and bar plot for bivariate analysis.

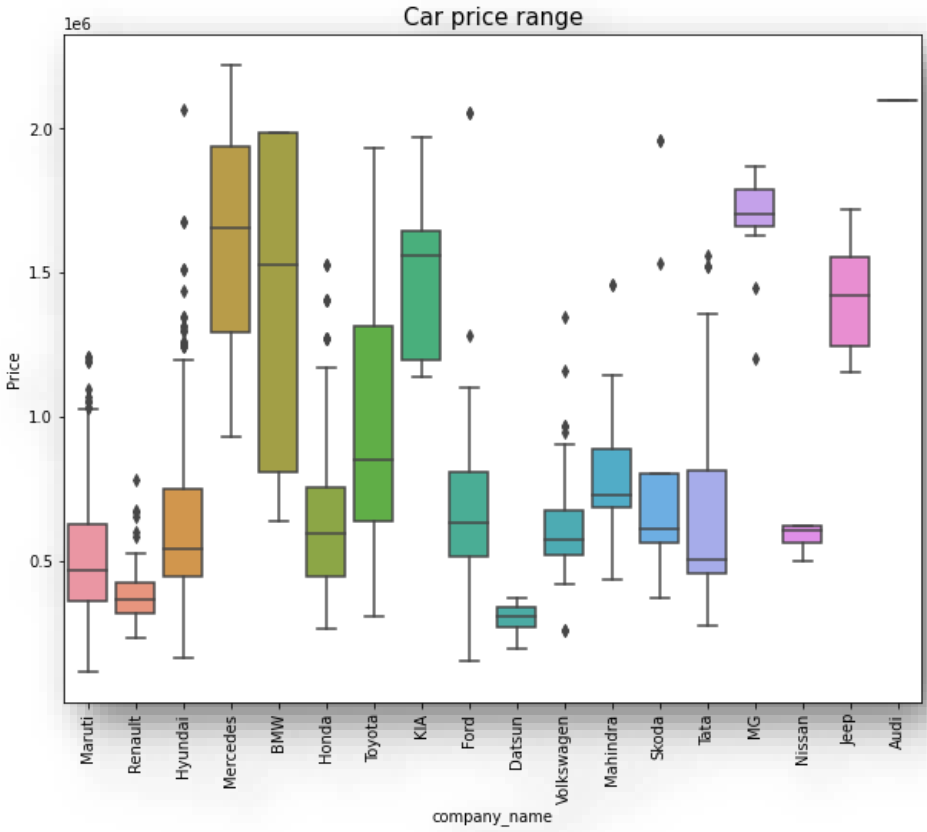


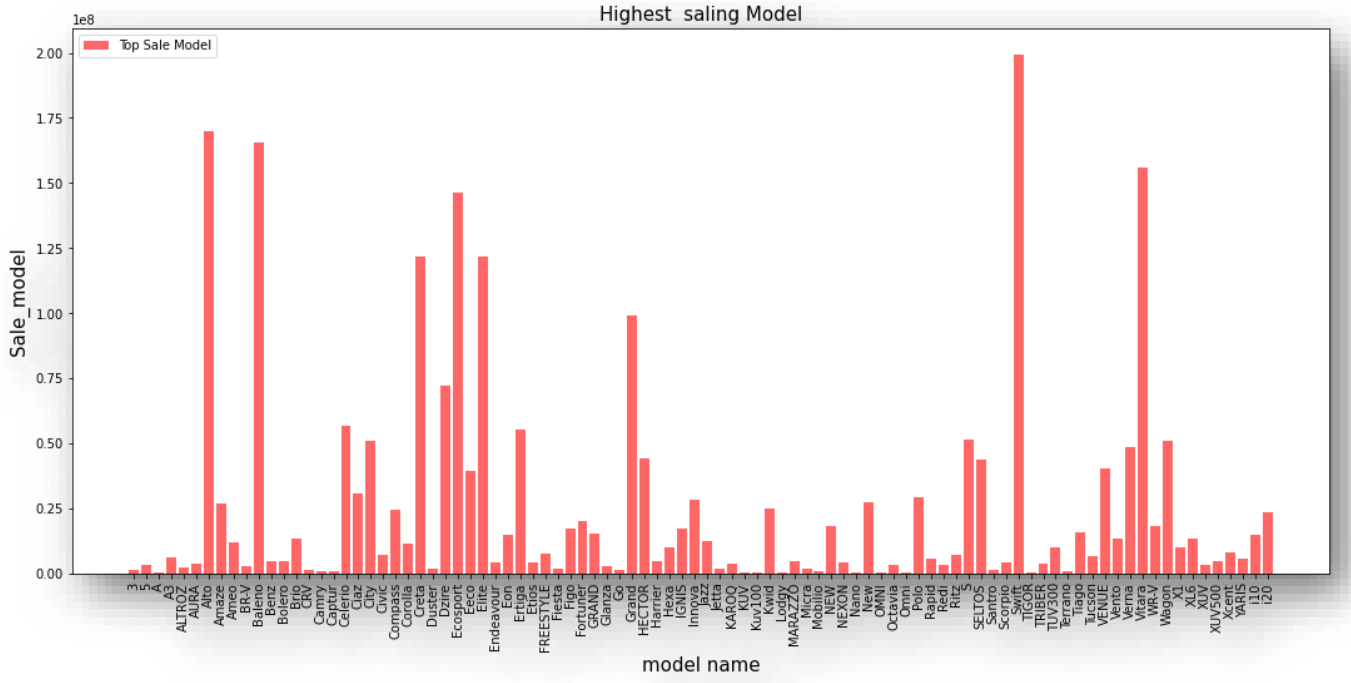


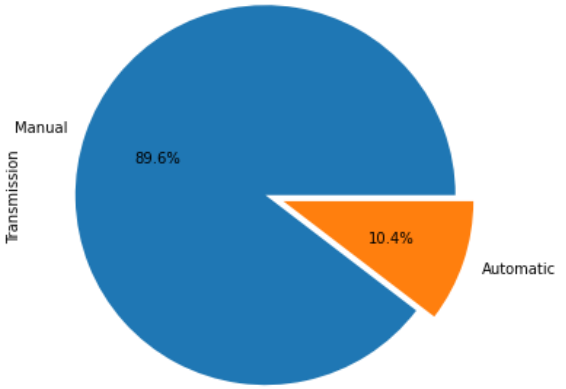
✓ We can clearly see that there is right positive skewness in the price and km driven columns so we have to treat them using suitable methods.

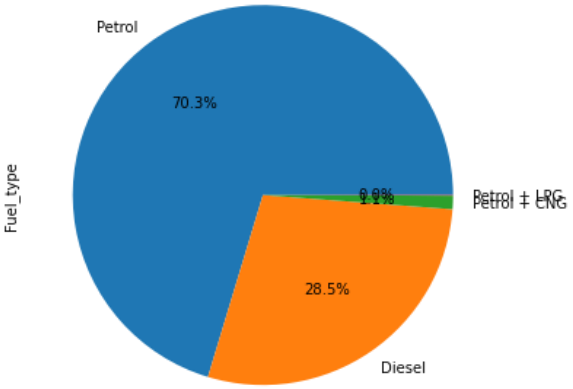

Highest sale per price
1e8
Top city sale
Sales
Cities
Ahmedabad
Bengaluru
Bhopal
Chandigarh
Chennai
Delhi
Hyderabad
Indore
Jaipur
Kochi
Kolkata
Lucknow
Ludhiana
Meerut
Mumbai
Mysore
Nagpur
Nasik
Panipat
Patna
Pune
Rajkot
Rohtak
Surat
Vadodara

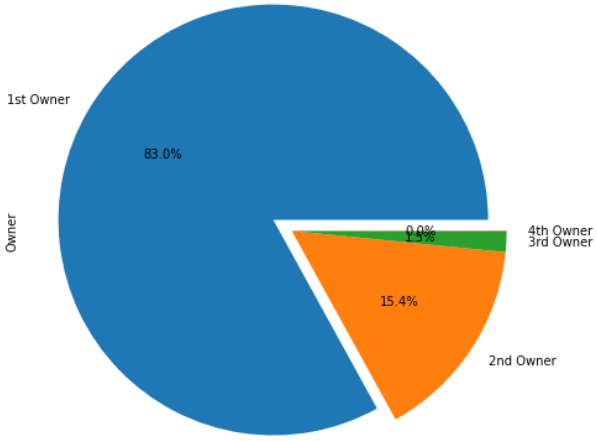

Highest saling car company
1e9
Top Sale Company
Sale_car
Comapny name
Audi
BMW
Datsun
Ford
Honda
Hyundai
Jeep
KIA
MG
Mahindra
Maruti
Mercedes
Nissan
Renault
Skoda
Tata
Toyota
Volkswagen

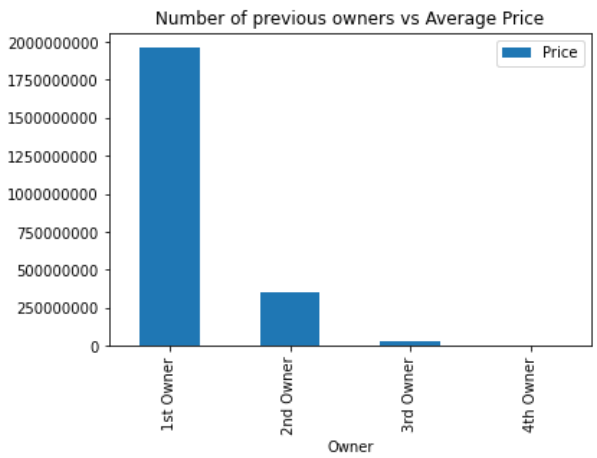

Car price range
1e6
Price
company_name
Maruti
Renault
Hyundai
Mercedes
BMW
Honda
Toyota
KIA
Ford
Datsun
Volkswagen
Mahindra
Skoda
Tata
MG
Nissan
Jeep
Audi

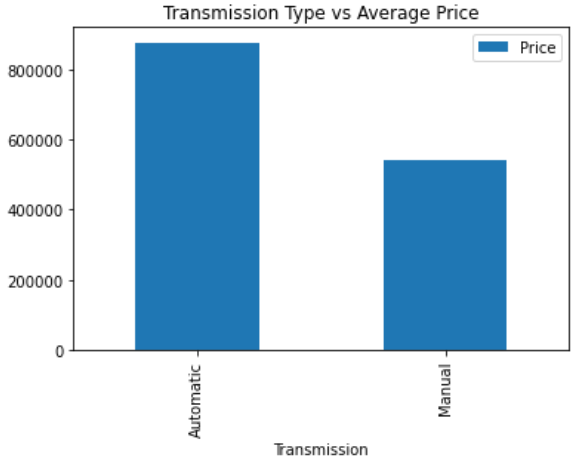

Highest saling Model
1e8
Top Sale Model
Sale_model
model name

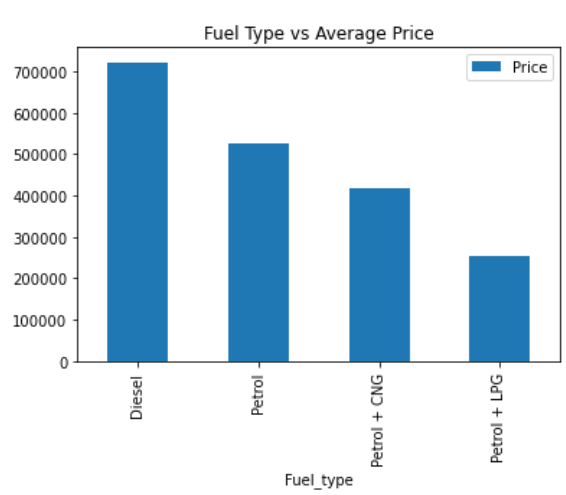

Manual
89.6%
Transmission
10.4%
Automatic

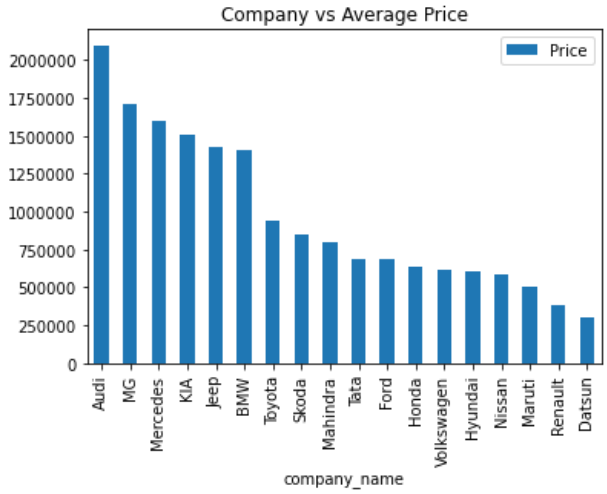

Petrol
70.3%
Fuel_type
Petrol + LPG
Petrol + CNG
28.5%
Diesel

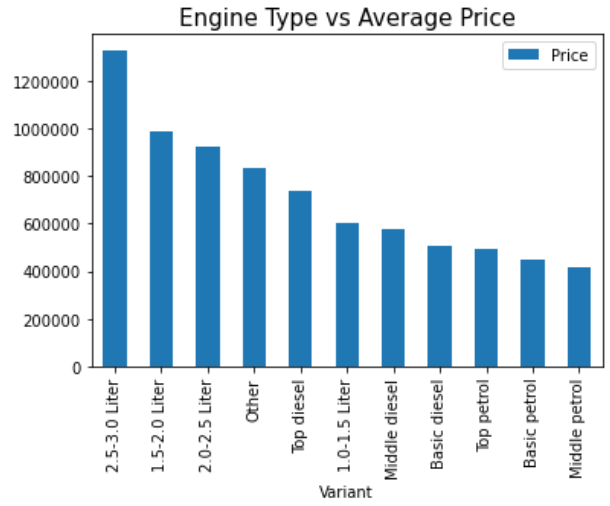

1st Owner
83.0%
Owner
4th Owner
3rd Owner
15.4%
2nd Owner


Number of previous owners vs Average Price
Price
2000000000
1750000000
1500000000
1250000000
1000000000
750000000
500000000
250000000
0
1st Owner
2nd Owner
3rd Owner
4th Owner
Owner


Transmission Type vs Average Price
Price
800000
600000
400000
200000
0
Automatic
Manual
Transmission







Observations:

- ✓ From the 1st graph we saw that highest used car selling cities are Mumbai and Pune from Cars24.com
- ✓ Ahmedabad, Surat, Bengaluru, Jaipur, Chennai are 2nd most cities of used car selling.
- ✓ Bhopal, Indore, Mysore, Rajkot, Patna in that cities used car selling is very low as cars24.com has improve selling in that city.
- ✓ Delhi have restriction of using cars like for Diesel engine Max used 10 years, for Petrol engine Max used 15 years.
- ✓ Maruti is top selling company in used cars because of it's reliability and Trustworthy of customers also have all types of cars like commercial to luxurious.
- ✓ Hyundai, Honda, Ford also have good market in selling cars.

- ✓ Audi, BMW, Mercedes are the luxurious brands so it can't be affordable to common peoples, also high maintenance cost so sells are less.
- ✓ Jeep, KIA, MG Hector, Mahindra are the SUV brands so behalf of new purchasing give more cost people liked to buy second hand at affordable price.
- ✓ Top used cars selling models are Alto, Baleno, Swift, Vitara all are from Suzuki brand.
- ✓ From the Hyundai Creta, Grand i10, i20, Verna most selling models.
- ✓ Maruti, Renault, Hyundai, Honda, TATA, Datsun, Skoda, Mahindra, Volkswagen are low range to medium price range brands.
- ✓ BMW, Mercedes, Toyota, Jeep, Audi, KIA, MG are high price range brands.
- ✓ From the pie chat mostly manual transmission driven cars are used by people about to 90%
- ✓ 70% of Petrol fuel type car are preferred by customers because of low running and maintenance cost.
- ✓ 30% of Diesel fuel type car are preferred by customer in most SUV or Luxurious cars.
- ✓ Most used cars are preferred of 1st owners about to 83%.
- ✓ From the bat plot price must decreasing by 1st owner to 4th owner
- ✓ Price of Diesel cars is high than petrol cars.
- ✓ CNG/LPG+ Petrol cars price is low preference of that type of fuel cars are low as chances of fire in engine or any king of accidents due high heat.
- ✓ As compare of transmission Automatic cars transmission have high price due comfortability of drive
- ✓ As compare in engine capacity 2.3L to 3.0L engine price is high
- ✓ Also, Diesel engine models price is high than Petrol engine

3.5 Run and evaluate select models

1). Linear Regressor

```
# selecting best random state
for i in range(1,100):
    x_train,x_test,y_train,y_test=train_test_split(x1,y,test_size=0.3,random_state=i)
    lr.fit(x_train,y_train)
    pred_tr=lr.predict(x_train)
    pred_te=lr.predict(x_test)
    if round(r2_score(y_train,pred_tr)*100,1)== round(r2_score(y_test,pred_te)*100,1):
        print('\n Random State',i)
        print('r2_score TR',r2_score(y_train,pred_tr)*100)
        print('r2_score TE',r2_score(y_test,pred_te)*100)
```

```
 Random State 4
r2_score TR 48.37175487312435
r2_score TE 48.385433040692924

 Random State 36
r2_score TR 48.406241093630385
r2_score TE 48.394085209199424
```

---

2). Decision Tree Regressor

```
# decision tree regressor
dtr = DecisionTreeRegressor(random_state=4)
dtr.fit(x_train, y_train)
y_pred_dt = dtr.predict(x_test)
print('Train score: ',dtr.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_dt)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_dt))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_dt)))
scr = cross_val_score(dtr, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  100.0
R2_score : 87.52875903156809
error1:
: 0.08256644492478864
RSME:
: 0.14979294003485719
Cross value score:  83.30824628800451
```

---

3). SVM Regressor

```
# support vector regressor
svr = SVR()
svr.fit(x_train, y_train)
y_pred_sv = svr.predict(x_test)
print('Train score: ',svr.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_sv)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_sv))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_sv)))
scr = cross_val_score(svr, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  81.17937823034943
R2_score : 74.19972365651226
error1:
: 0.1537735957941263
RSME:
: 0.2154510864178257
Cross value score:  71.22749207701989
```

---

4). KNeighbors Regressor

```
# Kneiarest neighbors
knn = KNeighborsRegressor()
knn.fit(x_train, y_train)
y_pred_kn = knn.predict(x_test)
print('Train score: ',knn.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_kn)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_kn))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_kn)))
scr = cross_val_score(knn, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  81.4695656239979
R2_score : 71.2878256627318
error1:
: 0.1541983384507638
RSME:
: 0.22728435938694197
Cross value score:  69.3521056652047
```

---

5). Random Forest Regressor

```
# Random forest regressor
rfr = RandomForestRegressor(random_state=15)
rfr.fit(x_train, y_train)
y_pred_rf = rfr.predict(x_test)
print('Train score: ',rfr.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_rf)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_rf))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_rf)))
print("Explain variance score =", round(explained_variance_score(y_test, y_pred_rf), 2))
scr = cross_val_score(rfr, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  98.65618433528395
R2_score : 91.40328452273639
error1:
: 0.07795861596451607
RSME:
: 0.12436632069690802
Explain variance score = 0.91
Cross value score:  89.82047086694382
```

---

6). Lasso and Ridge Regressor

```
# lasso regressor
l1=Lasso(alpha=0.0001,fit_intercept=True,normalize=True,precompute=False)
l1.fit(x_train, y_train)
y_pred_l1 = l1.predict(x_test)
print('Train score: ',l1.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_l1)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_l1))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_l1)))
```

```
Train score:  48.233081437541045
R2_score : 48.32271621604923
error1:
: 0.23311029234872185
RSME:
: 0.3049203073346656
```

```
# Ridge regressor
l2=Ridge(alpha=1,solver='sag')
l2.fit(x_train, y_train)
y_pred_l2 = l2.predict(x_test)
print('Train score: ',l2.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_l2)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_l2))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_l2)))
```

```
Train score:  48.37174023548671
R2_score : 48.38960518794414
error1:
: 0.23364917804675828
RSME:
: 0.30472290522174966
```

---

7). XGBOOST Regressor

```
# XGboost regressor
xg=XGBRegressor()
xg.fit(x_train, y_train)
y_pred_xg = xg.predict(x_test)
print('Train score: ',xg.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_xg)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_xg))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_xg)))
print("Explain variance score =", round(explained_variance_score(y_test, y_pred_xg), 2))
```

```
Train score:  99.1291508149418
R2_score : 94.48447700276058
error1:
: 0.06738516785197729
RSME:
: 0.0996161744183057
Explain variance score = 0.95
```

---

8). SGD Regressor

```
: # SGD Regressor
sd=SGDRegressor()
sd.fit(x_train, y_train)
y_pred_sd = sd.predict(x_test)
print('Train score: ',sd.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_sd)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_sd))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_sd)))
print("Explain variance score =", round(explained_variance_score(y_test, y_pred_sd), 2))
scr = cross_val_score(sd, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  48.24757442691423
R2_score : 48.23491312955445
error1:
: 0.2333472233669544
RSME:
: 0.30517923717350626
Explain variance score = 0.48
Cross value score:  43.79208216345381
```

---

9). Ada Boost Regressor

```
# Ada boost regressor
adr=AdaBoostRegressor()
adr.fit(x_train, y_train)
y_pred_ad = adr.predict(x_test)
print('Train score: ',adr.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_ad)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_ad))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_ad)))
print("Explain variance score =", round(explained_variance_score(y_test, y_pred_ad), 2))
scr = cross_val_score(adr, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  65.29943576486144
R2_score : 63.19096855059569
error1:
: 0.20697592893915562
RSME:
: 0.2573437970431165
Explain variance score = 0.64
Cross value score:  60.45266948084781
```

---

10).Gradient Boosting Regressor

```
# Gradient boosting regressor
gd=GradientBoostingRegressor()
gd.fit(x_train, y_train)
y_pred_gd = gd.predict(x_test)
print('Train score: ',gd.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_gd)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_gd))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_gd)))
print("Explain variance score =", round(explained_variance_score(y_test, y_pred_gd), 2))
scr = cross_val_score(gd, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  87.6310838973211
R2_score : 83.71178067237418
error1:
: 0.11786018467736281
RSME:
: 0.1711879983421643
Explain variance score = 0.84
Cross value score:  82.65310528521974
```

11). Bagging Regressor

```
# Bagging regressor
bg=BaggingRegressor()
bg.fit(x_train, y_train)
y_pred_bg = bg.predict(x_test)
print('Train score: ',bg.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_bg)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_bg))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_bg)))
print("Explain variance score =", round(explained_variance_score(y_test, y_pred_bg), 2))
scr = cross_val_score(bg, x1,y, cv=7,scoring='r2')
print('Cross value score: ',scr.mean()*100)
```

```
Train score:  98.16075016981023
R2_score : 91.06140680140487
error1:
: 0.08120033379931384
RSME:
: 0.12681513719727766
Explain variance score = 0.91
Cross value score:  88.55319766213842
```

12). Hyper Parameter Tuning for selected best model

```
58]: # Hyper tunning parameter for select model
     pr={'max_depth':[3,5,7],'n_estimators':[50,70,100,300,500],'learning_rate':[0.001,0.01,0.1,],
        'subsample':[0.6,0.7,1],'colsample_bytree':[0.5,.8,1]}
     grid = GridSearchCV(xg, pr,scoring="r2")
     grid.fit(x1,y)
     print(grid.best_params_)
```

```
{'colsample_bytree': 0.8, 'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 500, 'subsample': 0.6}
```

13). Result after Hyper parameter tuning

```
xg1=XGBRegressor(colsample_bytree=0.8,learning_rate=0.1,max_depth=7,n_estimators=500,subsample=0.6)
xg1.fit(x_train, y_train)
y_pred_xg1 = xg1.predict(x_test)
print('Train score: ',xg1.score(x_train, y_train)*100)
print('R2_score :',r2_score(y_test,y_pred_xg1)*100)
print('error1:\n:',mean_absolute_error(y_test,y_pred_xg1))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,y_pred_xg1)))
print("Explain variance score =", round(explained_variance_score(y_test, y_pred_xg1), 2))
```

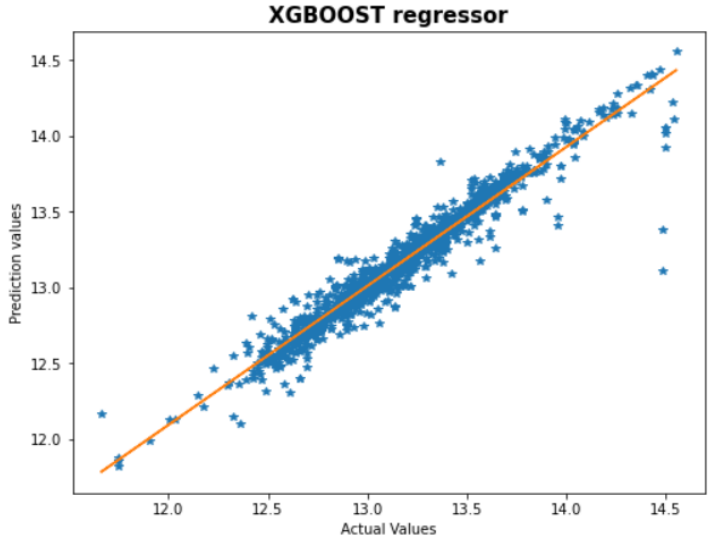
```
Train score:  99.89767115386081
R2_score : 93.69417762726573
error1:
: 0.06262646917632571
RSME:
: 0.10651416718848741
Explain variance score = 0.94
```

- **I have chosen all parameters of XGBoost Regressor, after tunning the model with best parameters I have R2 score 93.69%**

14). Regression and Distribution plot of Prediction

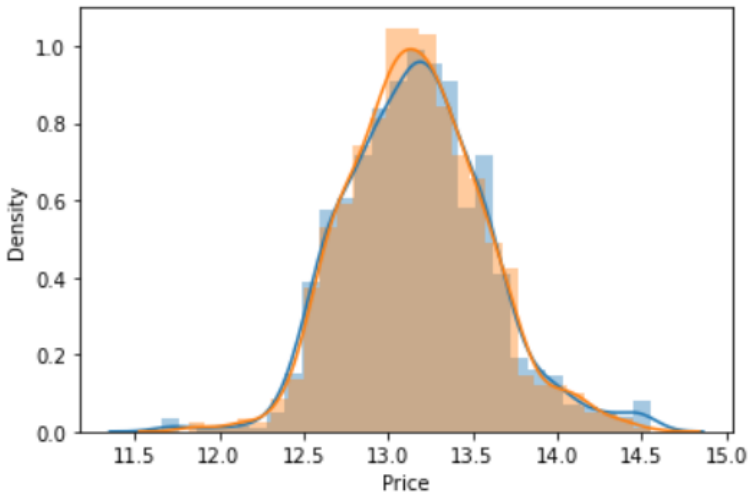
```
x=np.array(y_test)
y=np.array(xg1.predict(x_test))
plt.figure(figsize=(8,6))
plt.plot(x,y,'*')
m,b=np.polyfit(x,y,1)
plt.plot(x,m*x+b,)
plt.xlabel('Actual Values')
plt.ylabel('Prediction values')
plt.title('XGBOOST regressor',{'fontweight':'bold','fontsize':15})
```

```
Text(0.5, 1.0, 'XGBOOST regressor')
```



```
# Distribution plot of Prediction
sns.distplot(y_test)
sns.distplot(xg1.predict(x_test))
```

```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



15). Saving Model for Prediction

=> I have saved my best model using .obj as follows.

In [1971]:
```
# saving model
import joblib
joblib.dump(xg1,'Carprice.obj')
```

Out[1971]:
```
['Carprice.obj']
```

In [1973]:
```
# Loading the saved model
model=joblib.load("Carprice.obj")

#Prediction
prediction = model.predict(x_test)
```

In [1974]:
```
# prediction
pd.set_option('max_columns',5000)
pd.DataFrame([model.predict(x_test)[:],y_test[:]],index=["Predicted","Actual"])
```

Out[1974]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	
Predicted	13.858259	12.849697	13.055937	13.247395	13.064183	13.279303	13.075999	13.485873	12.816879	13.895761	13.000569	12.873230	13.392806	12.7
Actual	13.896998	12.899220	13.087772	13.247409	13.152116	13.335861	13.045050	13.497232	12.818552	13.912274	12.978262	12.918042	13.413689	12.7

3.6 Interpretation of the Results

- ✓ Scraping the all used cars data from websites and from different cities are challenging.
- ✓ Firstly, the datasets were not having duplicates value
- ✓ But there was null values in transmission columns handle it with mode method.
- ✓ And proper plotting for proper type of features will help us to get better insight on the data. I found maximum categorical columns in the dataset so I have chosen bar, count, pie plot to see the relation between target and features.
- ✓ I notice a skewness in the data so we have chosen proper methods to deal with skewness. If we skewness we may end up with a bad model which has less accuracy.
- ✓ Then scaling dataset has a good impact like it will help the model not to get biased. Since we have not removed outliers and skewness.
- ✓ We have to use multiple models while building model using dataset as to get the best model out of it.
- ✓ And we have to use multiple metrics like R2 score, MSE, RMSE and cross val score which will help us to decide the best model.
- ✓ I found XGBoost Regressor as the best model with 93.69.% R2 score..
- ✓ At last I have predicted car price using saved model. It was good!! that I was able to get the predictions near to actual values.

CONCLUSION

4.1 Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict the used car price. We have mentioned the step-by-step procedure to analyse the dataset and finding the correlation between the features. Thus we can select the features which are correlated to each other and are independent in nature. These feature set were then given as an input to four algorithms and a hyper parameter tunning was done to the best model and the accuracy has been improved. Hence we calculated the performance of each model using different performance metrics and compared them based on these metrics. Then we have also saved the best model and predicted the label. It was good the predicted and actual values were almost same.

4.2 Limitations of this work and Scope for Future Work

- ✓ First draw back is the length of the dataset it is small because of scraping data.
- ✓ Followed by a greater number of skewness these two will reduce our model accuracy.
- ✓ Data scrap from the different websites so it can be change at any time so also we have update model with new data at time.
- ✓ Also, we have tried best to deal with skewness and feature insight. So it looks quite good that we have achieved a accuracy of 93.69% even after dealing all these drawbacks.
- ✓ Also, this study will not cover all Regression algorithms instead, it is focused on the chosen algorithm, starting from the basic ensemble techniques to the advanced ones.