PROJECT REPORT ON :

“Used-Car Price Prediction Project”

SUBMITTED BY:

CHRIS CHHOTAI

ACKNOWLEDGMENT

I'd want to express my heartfelt gratitude to the "Flip Robo" team for providing me with the opportunity to work with such a beautiful dataset and for helping me develop my data analysis skills. And I'd like to express my heartfelt gratitude to Ms. Swati Mahaseth (SME Flip Robo), who has guided me through all of the challenges I've encountered while working on the project who has inspired me in many ways and encouraged me greatly with his wise words and unwavering support, resulting in a beautiful project.

Thank you so much to my "Data trained" academic team, who are the reason I am where I am now. Last but not least, my parents, who have supported me throughout my life. Thank you also to the many other people who have assisted me in completing the project, whether directly or indirectly.

1.INTRODUCTION

Business Problem Framing:

Predicting the price of a car is a fascinating and well-known topic. According to data obtained from the BiH Agency for Statistics, 921.456 automobiles were registered in 2014, with 84 percent of them being personal vehicles. This number has risen by 2.7 percent since 2013, and it is probable that this trend will continue, resulting in an increase in the number of cars in the future. This adds additional significance to the problem of the car price prediction. Because the price of an automobile is frequently determined by a number of distinct features and elements, accurate car price prediction necessitates specialist expertise. The most important ones are usually the brand and model, age, horsepower, and mileage. Due to frequent changes in the price of a fuel, the gasoline type used in the automobile as well as fuel consumption per mile have a significant impact on the price of a car. Exterior colour, door number, transmission type, dimensions, safety, air conditioning, interior, and whether or not it has navigation will all influence the automobile pricing. In this research, we used a variety of methodologies and procedures to improve the accuracy of used automobile price forecast.

We've seen a lot of changes in the car market as a result of the Covid 19's impact. Some cars are in high demand, making them expensive, while others are not, making them less expensive. One of our clients sells secondhand automobiles through tiny traders. Our customer is having issues with their previous car price estimation machine learning models due to market changes caused by the impact of Covid 19. As a result, they're on the lookout for new machine learning models based on new data. We must develop a paradigm for valuing automobiles.

Conceptual Background of the Domain Problem:

The manufacturer sets the price of new cars in the industry, with the government incurring some additional costs in the form of taxes. Customers who purchase a new car may rest comfortable that the money they spend will be well spent. However, due to rising new car prices and customers' inability to purchase new cars due to a lack of cash, used car sales are on the rise worldwide. There is a need for a system that can estimate the price of secondhand cars. While there are websites that provide this service, their forecast approach may not be the most accurate. Furthermore, several methods and algorithms may aid in the prediction of a used car's true market worth. When purchasing and selling, it's critical to understand their true market value.

Many people have been interested in the used automobile market at some point in their lives because they wanted to sell or acquire a used vehicle. It's a great mistake to pay too much or sell for less than the market value in this process.

The outcomes of this study may be of interest to one of the largest target groups. Used car vendors that have a better understanding of what makes a car popular and what are the most significant attributes for a used automobile may take this information into account and provide better service.

Review of Literature:

Even while the market for new cars has shrunk, the second-hand car market has continued to grow. According to Indian Blue Book's recent study on India's used car market, approximately 4 million used cars were purchased and sold in 2018-19. Both consumers and sellers have benefitted from the second-hand car industry. The majority of people choose to acquire old automobiles since they are less expensive and they can resale them after a few years of use for a profit. The cost of a used car is determined by a number of criteria, including the kind of gasoline, colour, model, mileage, transmission, engine, and number of seats. The market price of secondhand autos will continue to fluctuate. As a result, the assessment model for predicting the price of the used cars is needed.

Motivation for the Problem Undertaken:

There are websites that provide an estimate of a car's value. They might have a good model for forecasting. Having a second model, on the other hand, may aid them in providing a better prediction to their users. As a result, the model established in this study could aid online web services that determine the market worth of a used car.

2.Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem:

As a preliminary step, I scraped the necessary information from the carsdekho website. I retrieved data from several locations and saved it in excel format.

In this particular instance, my target column was car price, which was a continuous column. As a result, it's evident that this is a regression problem, and I'll need to employ all regression procedures to develop the model. The dataset contained null values. Also, I saw some unneeded entries in some of the columns, such as more than 50% null values in some columns, therefore I chose to remove such columns. If I leave those columns alone, the model will have a lot of skewness.

We used feature engineering to extract the appropriate feature format because we scraped the data from the Cardekho website and the raw data was not in the format. I employed plotting techniques such as distribution plots, bar plots, reg plots, strip plots, and count plots to gain a better understanding of the features. With these plots, I was able to better comprehend the relationship between the features. I also discovered outliers and skewness in the dataset, so I used the z-score approach to remove outliers and the yeo-johnson method to remove skewness. While building the model, I applied all of the regression procedures, then tuned and saved the best model. Finally, using the saved model, I was able to anticipate the car price.

Data Sources and their formats:

The information was gathered in excel format from the website cardekho.com. Selenium was used to scrape the data. The dataset

is saved as an excel file after the required features have been scraped.

In addition, my target dataset has 12608 rows and 20 columns. I have an object kind of data in this particular dataset that has been altered based on your study of the dataset. The following provides information on features.

Features Information:

- Fuel_type : Type of fuel used for car engine
- Car_Name : Name of the car with Year
- Running_in_kms : Car running in kms till the date
- Endine_disp : Engine displacement/engine CC
- Gear_transmission : Type of gear transmission used in car
- Milage_in_km/ltr : Overall milage of car in Km/ltr
- Seating_cap : Availability of number of seats in the car
- Max_power : Maximum power of engine used in car in bhp
- front_brake_type : type of brake system used for front-side wheels
- rear_brake_type : type of brake system used for back-side wheels
- Max_power : Maximum power of engine used in car in bhp
- front_brake_type : type of brake system used for front-side wheels
- rear_brake_type : type of brake system used for back-side wheels
- cargo_volume : the total cubic feet of space in a car's cargo area.
- height : Total height of car in mm
- width : Width of car in mm
- length : TOtal length of the car in mm
- Weight : Gross weight of the car in kg
- Insp_score : inspection rating out of 10
- top_speed : Maximum speed limit of the car in km per hours
- City_url : Url of the page of cars from a particular city
- Car_price : Price of the car

Data Preprocessing Done:

- As a preliminary step, I used selenium to scrape the relevant data from the cardekho website. I also imported the necessary libraries as well as the dataset, which was in Excel format.

- Then I performed every statistical analysis, such as checking shape, nunique, and value.

- Counts, information, and so forth.....

- When looking for null values, I discovered null values in the dataset and used the imputation approach to replace them.

- Unnamed:0, cargo volume, and Insp score columns have also been removed because they are no longer useful.

- Following that, as part of feature extraction, All of the columns' data types were transformed, and I was able to extract relevant information from the raw dataset. We believe that this data will be more useful to us than raw data.

Data Inputs- Logic- Output Relationships:

- I used a dist plot to see the distribution of skewness in each column data because I had numerical columns.
- For each pair of category features, I utilised a bar plot to show the relationship between label and independent features.

- To observe the relationship between numerical columns and the target column, I utilised a reg plot and a strip plot.
- I've noticed that maximum columns and target have a linear relationship.

3.DATA ANALYSIS AND VISUALIZATION:

1.Identification of possible problem-solving approaches (methods):

We must clean and prepare the data collected because it was not in the right format for our analysis. I used the z-score approach to eliminate outliers. I also used the yeo-johnson method to remove skewness. According to our understanding, we have removed all extraneous columns from the dataset. To check the correlation between dependent and independent features, use Pearson's correlation coefficient. In addition, I scaled the data using Standardisation. After scaling, we must use VIF to remove multicolinearity. After that, all regression procedures are used to develop a model.

2.Testing of Identified Approaches (Algorithms)

Because car price was my aim and it was a continuous column with an incorrect format that needed to be changed to a continuous float datatype column, this specific issue constituted a regression issue. And I built my model using all of the regression procedures. I discovered DecisionTreeRegressor as the best model with the least difference by looking at the difference between the r2 score and the cross validation score. We must also run many models to obtain the best model, and we

must use cross validation to avoid overfitting confusion. The regression algorithms I utilised in my project are listed below.

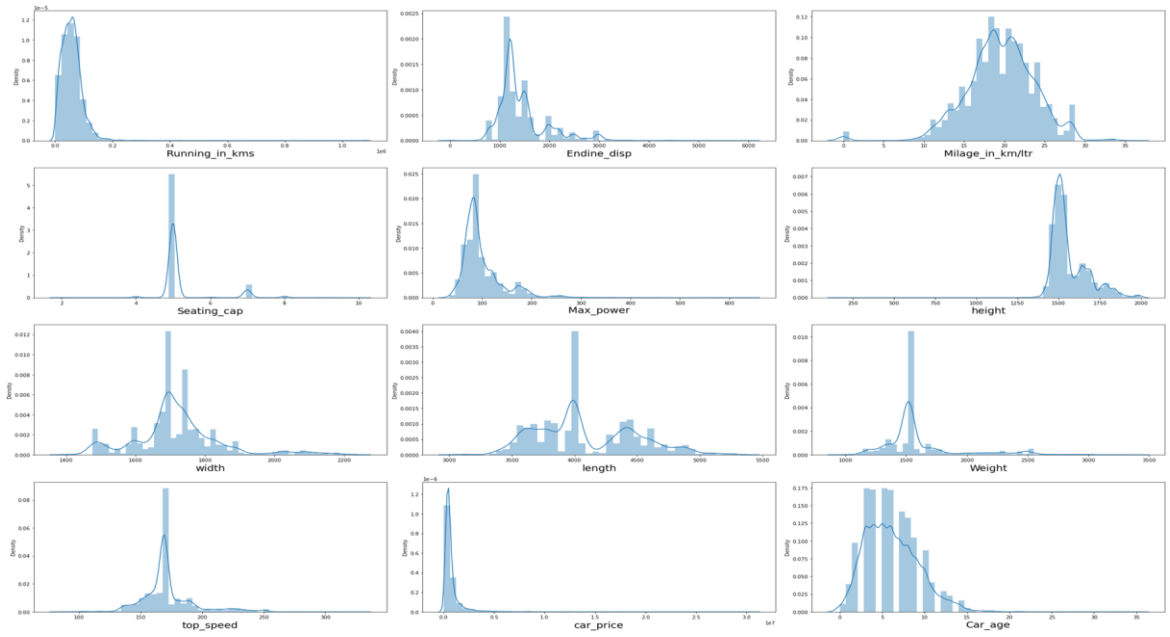
- RandomForestRegressor
- ExtrTrees Regressor
- GradientBoostingRegressor
- DecisionTreeRegressor
- BaggingRegressor

3.Key Metrics for success in solving problem under consideration:

I evaluated using the following metrics: • I utilised mean absolute error, which is the amount of the difference between an observation's prediction and the true value of that observation.
• One of the most often used measures for measuring the quality of forecasts is root mean square deviation.
• I utilised the r2 score to determine the accuracy of our model.

4.Visualizations:

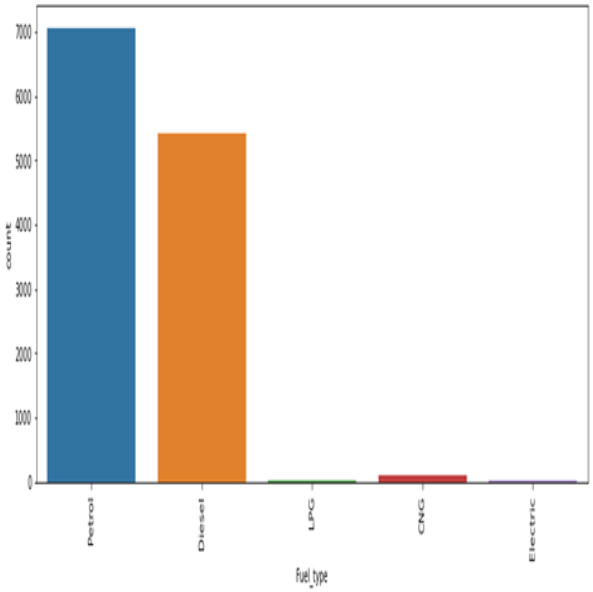
Univatiate Analysis for Numerical Column:

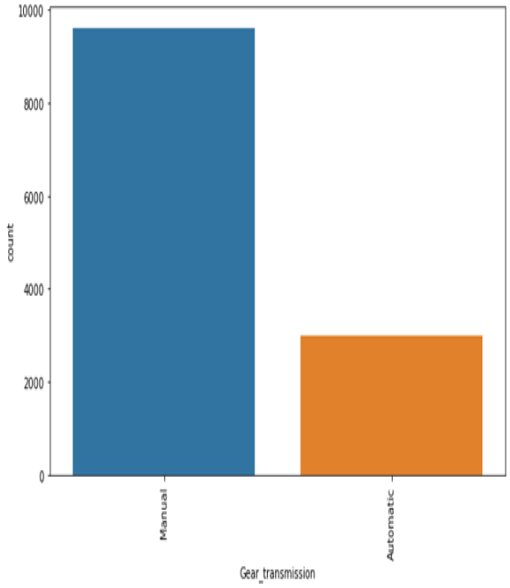


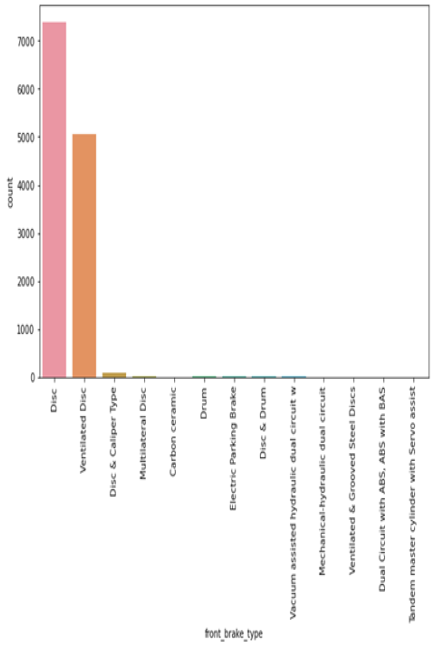
Observations:

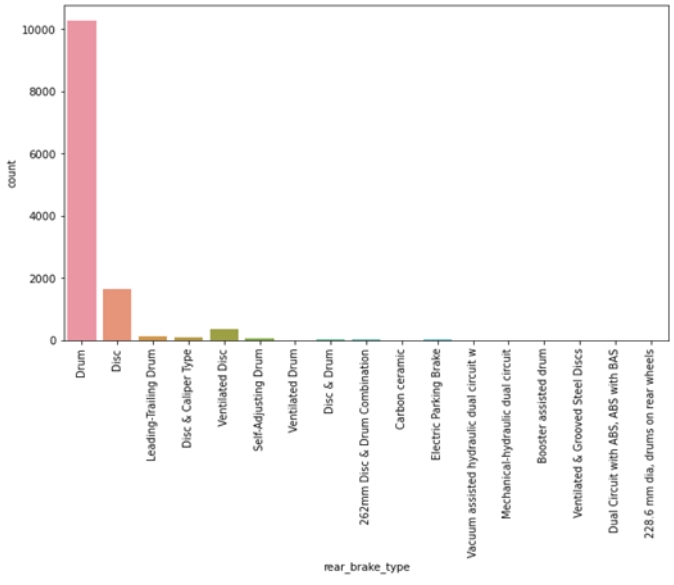
We can easily see that majority of the columns are skewed, so we must address them using appropriate approaches.

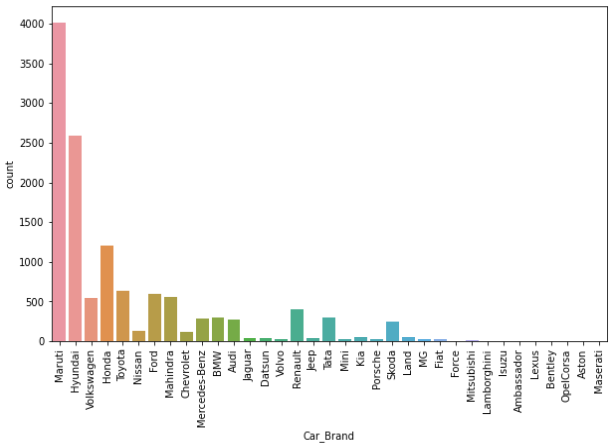
Univariate Analysis for Categorica Column:

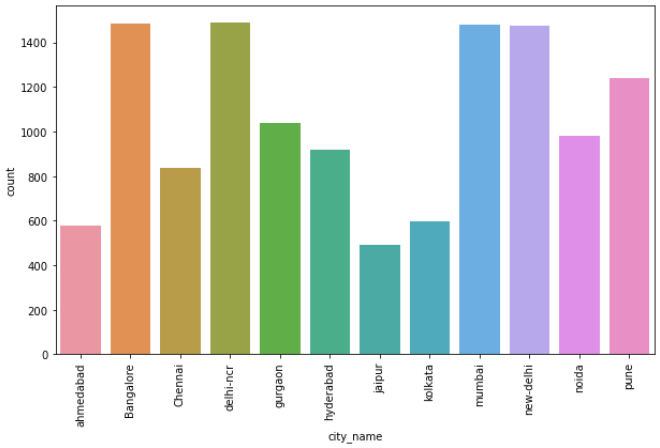








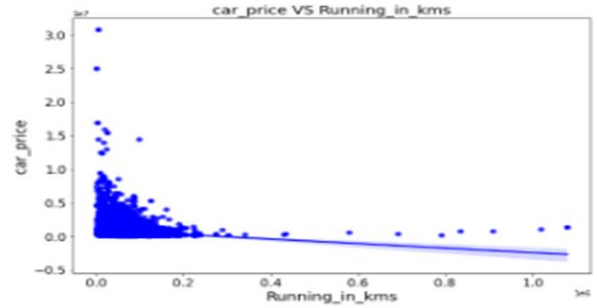


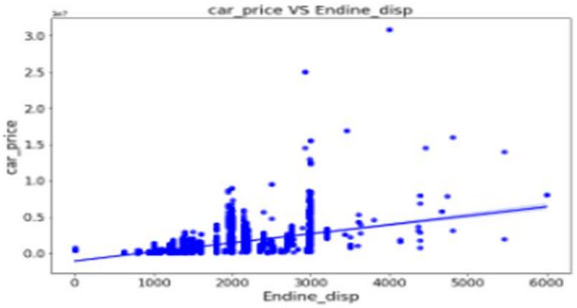


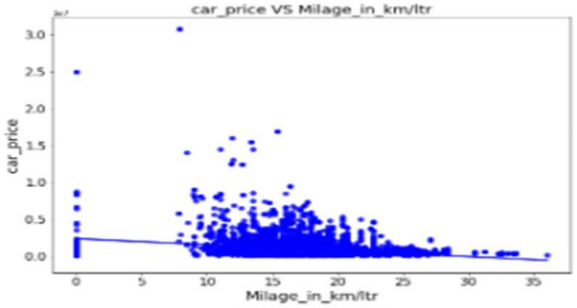
Observations:

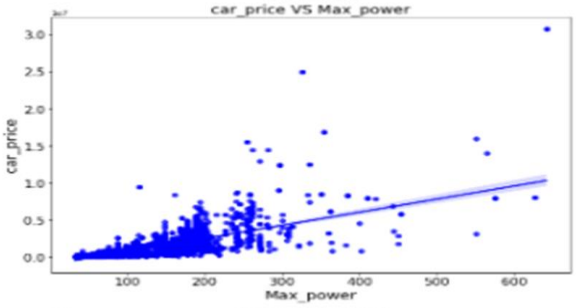
- ✓ The majority of automobiles are powered by gasoline or diesel.
- ✓ The majority of automobiles have a manual transmission.
- ✓ Cars with Disc front brakes are more common, followed by those with Ventilated Disc.
- ✓ The number of drum unusual break autos is increasing.
- ✓ Maruti Suzuki has the most cars on the market, followed by Hyundai.
- ✓ We can find the most automobiles for sale in Bangalore, Delhi-NCR, Mumbai, and New Delhi. Because these are the most inhabited areas.

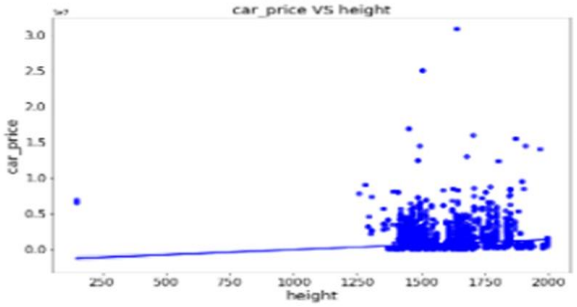
Bivariate Analysis for Numerical columns:

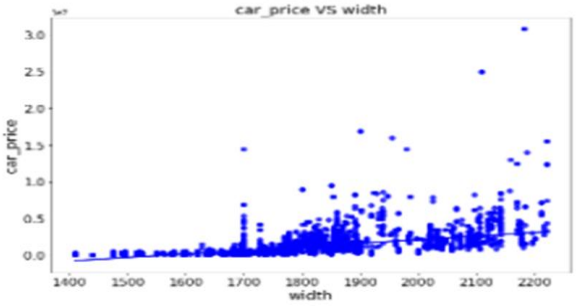


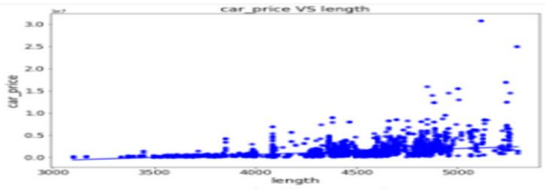


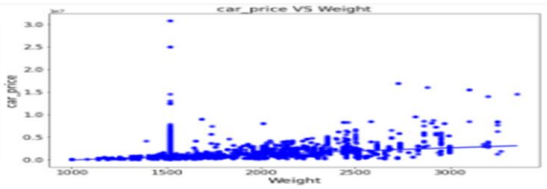


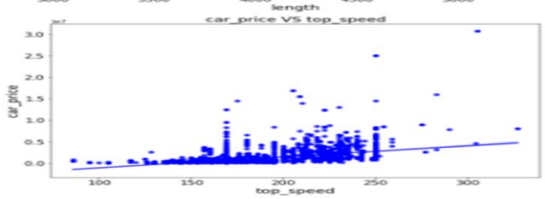


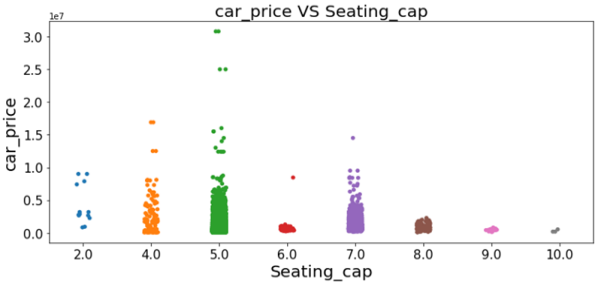


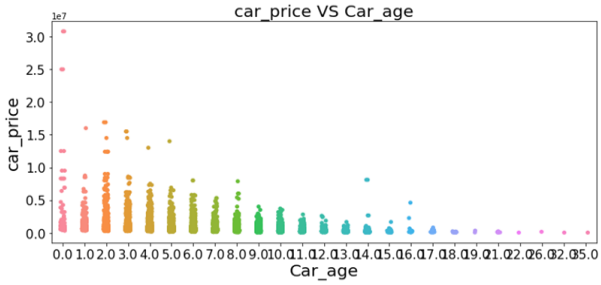








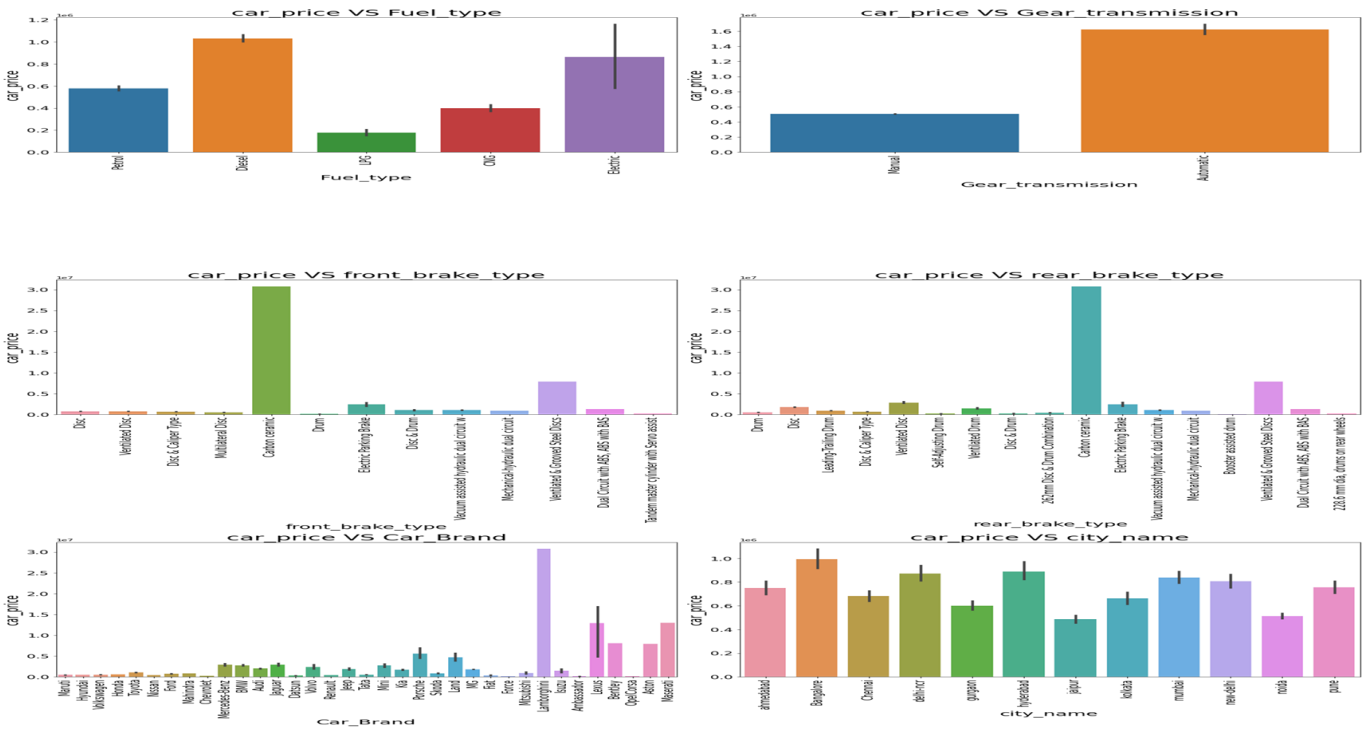




Observations:

- ✓ The majority of cars have fewer than 20 thousand kilometres travelled on them. And the cost of an automobile is expensive if it hasn't been driven much.
- ✓ The majority of automobiles have 1000-3000 Endine disp. In addition, the price of a 3000 Endine disp automobile is quite high.
- ✓ The majority of autos have a mileage of 10 to 25 kilometres. Also, milage has nothing to do with the cost of a car.
- ✓ The price of a car rises in tandem with the increase in Max power.
- ✓ The car price has no relationship with height.
- ✓ The price of a car rises in tandem with its breadth.
- ✓ The cost of a car rises in tandem with the length of the vehicle.
- ✓ The price of a car is likewise proportional to its weight.
- ✓ As top speed rises, so does the price of a car.
- ✓ Cars with 5 and 4 seats are the most expensive.
- ✓ The price of a car lowers as the age of the vehicle grows.

Bivariate Analysis for Categorical columns:



Observations:

- ✓ Diesel and electric cars are more expensive than gasoline, LPG, and CNG.
- ✓ Automobiles with automatic transmissions are more expensive than those with manual transmissions.
- ✓ Carbon Ceramic front brakes are more expensive than other types of brakes.
- ✓ Carbon Ceramic rear breaks are more expensive than other types of rear breaks.
- ✓ The sale price of Lamborghini cars is the highest.
- ✓ Because Bangalore, Hyderabad, and Delhi-NCR are densely populated cities, car rates are high.

5.Run and evaluate Selected Models:

A. Model Building

1. RandomForestRegressor:

1. Random Forest Regressor:

In [108]:

```
RFR=RandomForestRegressor()
RFR.fit(X_train,y_train)
pred=RFR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(RFR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 96.43201079706564
mean_squared_error: 8748989647.849602
mean_absolute_error: 50651.823611583386
root_mean_squared_error: 93536.03395403079

Cross validation score : 93.02022344365139

R2_Score - Cross Validation Score : 3.4117873534142547
```

- RandomForestRegressor has given me 96.46% r2_score and the difference between r2_score and cross validation score is 3.41%, but still we have to look into multiple models.

2. GradientBosstingRegressor:

2. Gradient Boosting Regressor:

In [109]:
```
GBR=GradientBoostingRegressor()
GBR.fit(X_train,y_train)
pred=GBR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(GBR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 94.01295066999727
mean_squared_error: 14680714999.440113
mean_absolute_error: 73887.91896207062
root_mean_squared_error: 121164.00042685993

Cross validation score : 90.19383240524806

R2_Score - Cross Validation Score : 3.819118264749207
```

- GradientBoostingRegressor is giving me 94.01% r2_score and the difference between r2_score and cross validation score is 3.81%.

3. DecisionTreeRegressor:

3. Decision Tree Regressor:

In [110]:
```
DTR=DecisionTreeRegressor()
DTR.fit(X_train,y_train)
pred=DTR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(DTR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 92.55981819497732
mean_squared_error: 18243909913.384327
mean_absolute_error: 62459.27343976778
root_mean_squared_error: 135070.0185584659

Cross validation score : 88.24308032682848

R2_Score - Cross Validation Score : 4.316737868148849
```

- DecisionTreeRegressor is giving me 92.55% r2_score and the difference between r2_score and cross validation score is 4.13%.

4. BaggingRegressor:

4. Bagging Regreesor:

In [111]:
```
BR=BaggingRegressor()
BR.fit(X_train,y_train)
pred=BR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(BR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 95.52124404563119
mean_squared_error: 10982261226.517046
mean_absolute_error: 55365.77981615868
root_mean_squared_error: 104796.28441179128

Cross validation score : 92.49889681736596

R2_Score - Cross Validation Score : 3.0223472282652324
```

- BaggingRegressor is giving me 95.52% r2_score and the difference between r2_score and cross validation score is 3.02%.

B. Hyper Parameter Tuning:

Hyper Parameter Tuning for best model:

In [300]:
```
#importing necessary libraries
from sklearn.model_selection import GridSearchCV
```

In [301]:
```
parameter = {'criterion':['squared_error', 'friedman_mse', 'absolute_error', 'poisson'],
             'max_features':[10],
             'min_samples_split':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],
             'max_depth':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]}
```

In [302]:
```
GCV=GridSearchCV(DecisionTreeRegressor(),parameter,cv=10)
```

In [303]:
```
GCV.fit(X_train,y_train)
```

Out[303]:
```
GridSearchCV(cv=10, estimator=DecisionTreeRegressor(),
             param_grid={'criterion': ['squared_error', 'friedman_mse',
                                       'absolute_error', 'poisson'],
                         'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
                                       13, 14, 15],
                         'max_features': [10],
                         'min_samples_split': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                                               11, 12, 13, 14, 15]})
```

In [304]:
```
GCV.best_params_
```

Out[304]:
```
{'criterion': 'friedman_mse',
 'max_depth': 13,
 'max_features': 10,
 'min_samples_split': 7}
```

In [305]:
```
Best_mod=DecisionTreeRegressor(criterion='friedman_mse',max_depth=15,max_features='auto',min_samples_split=4,splitter='random')
Best_mod.fit(X_train,y_train)
pred=Best_mod.predict(X_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2_Score: 93.0703772172476
mean_squared_error: 18085987578.08605
mean_absolute_error: 71835.24767115884
RMSE value: 134484.15363189095
```

After hyper parameter tuning of the model we got an accuracy of 93.6%.

I have choosed all parameters of DecisionTreeRegressor, after tunning the model with best parameters I have incresed my model accuracy from 92.55% to 93.6%.

C. Saving the model and Predictions:

Saving the model:

In [306]:
```
# Saving the model using .pkl
import joblib
joblib.dump(Best_mod,"Car_Price.pkl")
```

Out[306]:
```
['Car_Price.pkl']
```

Predictions:

In [307]:
```
# Loading the saved model
model=joblib.load("Car_Price.pkl")

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

Out[307]:
```
array([ 379000.        , 1650000.        ,  378333.33333333, ...,
         557000.        ,  204227.27272727,  416562.5       ])
```

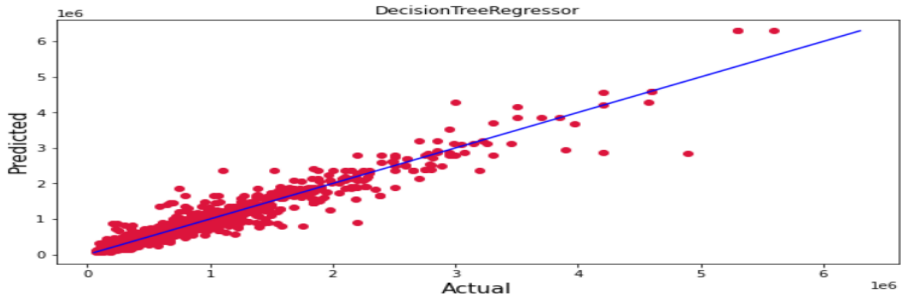
In [308]:
```
pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"])
```

Out[308]:

	0	1	2	3	4	5	6	7	8	9	10	11
Predicted	379000.0	1650000.0	378333.333333	313333.333333	549173.913043	451500.0	554500.0	554500.0	1683000.0	708448.979592	900000.0	401578.947368
Actual	379000.0	1650000.0	465000.000000	435000.000000	550000.000000	450000.0	550000.0	550000.0	1500000.0	643000.000000	1125000.0	385000.000000

✓ **I have predicted the Car Price using saved model, and the predictions look good. The Predicted values are almost same as actual values.**

```
In [309]: plt.figure(figsize=(10,5))
          plt.scatter(y_test, prediction, c='crimson')
          p1 = max(max(prediction), max(y_test))
          p2 = min(min(prediction), min(y_test))
          plt.plot([p1, p2], [p1, p2], 'b-')
          plt.xlabel('Actual', fontsize=15)
          plt.ylabel('Predicted', fontsize=15)
          plt.title("DecisionTreeRegressor")
          plt.show()
```



- **Plotting Actual vs Predicted, To get better insight. Bule line is the actual line and red dots are the predicted values.**

Interpretation of the Results:

- The information was taken from the cardekho website.
- The dataset was difficult to work with because it included 20 characteristics and 12608 samples.
- To begin, the datasets contained any null values, therefore I utilised the imputation approach to replace them.
- And because all of the features included a large number of superfluous entries, I utilised feature extraction to retrieve the needed format of variables.
- Furthermore, correct plotting for specific types of characteristics can aid us in gaining a better understanding of the data. Because the dataset contains both numerical and categorical columns, I used reg plots, strip plots, and bar plots to visualise the relationship between target and characteristics.

- I observe a large number of outliers and skewness in the data, thus we've chosen appropriate strategies to deal with them. We may wind up with a lousy model with reduced accuracy if we overlook the outliers and skewness.
- Then scaling the dataset has a positive impact, as it aids the model in not becoming baised. We must choose Standardisation because we have removed outliers and skewness from the dataset.
- To extract the best model out of the dataset, we need to use many models while developing the model.
- We must also use multiple measures such as mse, mae, rmse, and r2 score to assist us choose the optimum model.
- With a r2 score of 91.79 percent, I discovered DecisionTreeRegressor to be the best model. In addition, by using hyper parameter tuning, I was able to improve the accuracy of the best model.
- Finally, using the saved model, I was able to anticipate the price of a used car. It was fantastic!! that I was able to come close to the forecasts orignal values.

CONCLUSION :

Key Findings and Conclusions of the Study:

We employed machine learning techniques to estimate used automobile prices in this project report. The process for analysing the dataset and determining the correlation between the features has been described in detail. As a result, we can choose traits that are both connected and independent in nature. These feature sets were then fed into five algorithms, with the best model undergoing hyper parameter tuning, resulting in increased accuracy. As a result, we calculated each model's performance using several performance measures and compared them using those criteria. The best model

was then saved, and the used car price was forecasted. The fact that the projected and actual values were practically identical was a plus.

Learning Outcomes of the Study in respect of Data Science:

The dataset was found to be fairly fascinating to work with because it contains a wide range of data and was scraped from the Cardekho website using Selenium.

Because of advancements in computing technology, it is now possible to evaluate social data that could not previously be gathered, processed, or analysed. In used automobile price research, new machine learning analytical tools can be applied. The power of visualisation has aided us in comprehending data through graphical representation, allowing me to comprehend what the data is attempting to convey. One of the most critical phases in removing unrealistic and null values is data cleaning. This is an exploratory study that compares five machine learning techniques for estimating used automobile price prediction.

To summarise, the use of machine learning to estimate the price of a used car is still in its early stages. We hope that our study has made a tiny step forward in terms of methodological and empirical contributions to crediting online platforms, as well as introducing an alternate way to used automobile pricing valuation. Incorporating extra used automobile data from a bigger economic background with more attributes could be a future study path.

Limitations of this work and Scope for Future Work:

- The first disadvantage is scraping the data because it is a constantly changing procedure.

- Additional outliers and skewness will diminish our model's accuracy, which will be followed by more outliers and skewness.
- We've also done our best to handle outliers, skewness, and null values. So, even after dealing with all of these flaws, it appears that we have obtained a 93.60 percent accuracy.

- Also, this research will not cover all Regression methods; rather, it will focus on the algorithm chosen, starting with the most fundamental ensembling approaches and progressing to the most advanced.