



CAR PRICE PREDICTION

Submitted by-
KAUSHIK VEER

ACKNOWLEDGMENT

All thanks to fliprobo technologies for providing me the opportunity to work on this project. I leaned alot from this project.

INTRODUCTION

- 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. This project contains two phase :-

- Data collection phase
- Model building phase

- Conceptual Background of the Domain Problem

First we need to collect data so in data collection phase We scraped 5000 used cars data from websites like cardekho, olx, etc. We need web scraping for this. We have to fetch data for different locations. The number of columns for data doesn't have limit, it's up to you and your creativity.

After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

- Review of Literature

This study investigates and explores the relationship between variables which affect consumer buying behaviour for used cars in different city. It also attempts to understand used car market in India.

The result of this study provides evidence and insights about the relationship between the variable which affects consumer buying behaviour for used cars. Apart from that, the study also provides valuable insight toward the understanding on how different factors provide the base for purchase intention and affects the consumer buying behaviour of used cars.

We will be finding these important features and how do they affect positively and negatively with sale price of the houses and use them to predict the housing prices using machine learning models.

- Motivation for the Problem Undertaken

My objective is to find the important attributes and how they affect the prices of the used cars using data science.

To use analytics in area of used cars market and see how we can use data science to perform various analysis to help the company to make better decisions and generate profits is a interesting task as I will be able to learn a lot in this field and gain experience using data science.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

This project contains two phase :-

- 1) Data collection phase
- 2) Model building phase

In data collection we scrapped 5000 used cars data to perform analysis from cardekho, olx, etc websites.

The data contains 5670 rows and 7 columns.

We need to predict the prices of used cars using the dataset

We need to check the relation of the variables with the price column and which are the most positive and negative aspects that affects the price.

Data contains Null values. You need to treat them using the domain knowledge and your own understanding.

Extensive EDA has been performed to gain relationships of important variable and price.

We will build Machine Learning models and determine the optimal values of Hyper Parameters.

We will find important features which affect the price positively or negatively.

- Data Sources and their formats

The data contains 5670 rows and 7 columns.

Following are the variables with description :-

- year & brand name = shows us the details of the model year and its brand name.
- Model = model name of the car.
- Fuel = fuel used by the car engine.
- Variant = variant of the car whether its automatic or manual.
- Kms = shows us the details of how much km the car has covered.
- Price = price of the car.

Here is the snapshot of the data :-

Unnamed: 0		year and brand	model	fuel	variant	kms	Price	year	brand
0	0	2016 Maruti Baleno	1.2 CVT Delta	Petrol	Automatic	25,735 kms	5.74	2016	Maruti Baleno
1	1	2015 Hyundai Grand i10	Sportz	Petrol	Manual	18,174 kms	4.5	2015	Hyundai Grand i10
2	2	2014 Hyundai i10	Sportz 1.1L	Petrol	Manual	45,195 kms	3.31	2014	Hyundai i10
3	3	2017 Maruti Alto K10	VXI Optional	Petrol	Manual	22,761 kms	3.34	2017	Maruti Alto K10
4	4	2015 Hyundai Grand i10	Asta Option	Petrol	Manual	22,819 kms	4.25	2015	Hyundai Grand i10
...
5665	5665	2012 Maruti Eeco	5 Seater AC BSIV	Petrol	Manual	30,000 kms	2.25	2012	Maruti Eeco
5666	5666	2018 Ford Figo	1.2P Titanium MT	Petrol	Manual	27,003 kms	4.9	2018	Ford Figo
5667	5667	2019 Hyundai Creta	1.6 SX Option	Petrol	Manual	19,000 kms	13	2019	Hyundai Creta
5668	5668	2018 Maruti Ignis	Alpha	Petrol	Manual	14,002 kms	4.75	2018	Maruti Ignis
5669	5669	2018 Toyota Innova Crysta	2.8 GX AT BSIV	Diesel	Automatic	1,12,000 kms	16.5	2018	Toyota Innova Crysta

- Data Pre-processing Done

We saw some nan values so we dropped them as the data loss is not more than 1%.

we can see that there are some null values present in the price column so we will be removing the null values

```
] : #dropping the null values  
df.dropna(inplace=True)
```

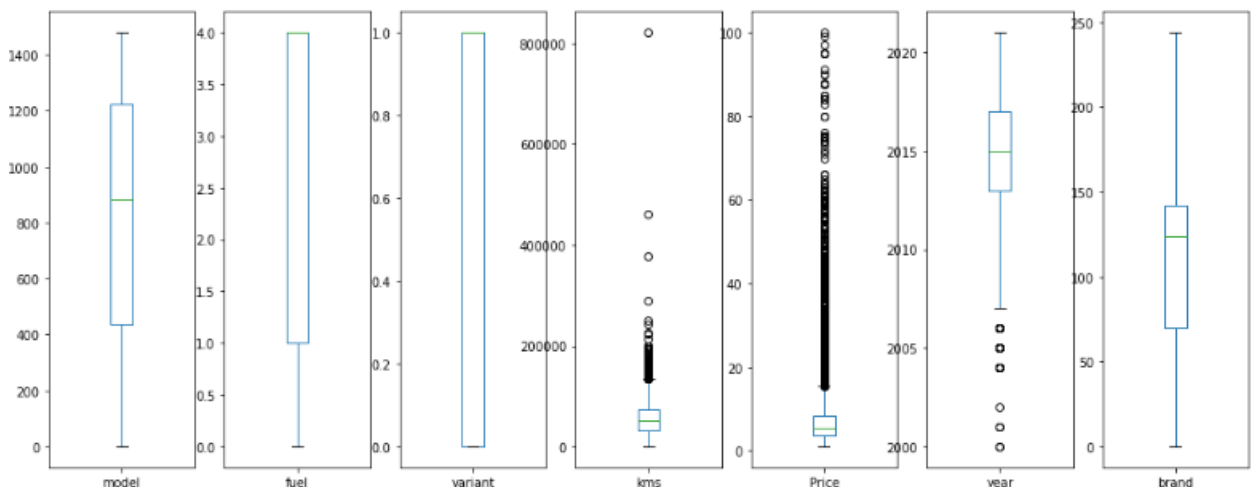
```
] : df
```

we will drop the columns unnamed and year and brand
unnamed is nothing but a serial number to the entry
we already split the year and brand into two different columns.

```
: df=df.drop(['year and brand','Unnamed: 0'],axis=1)  
df
```

```
:  
      model  fuel  variant  kms  Price  year  brand  
0  1.2 CVT Delta  Petrol  Automatic  25735  5.74  2016  Maruti Baleno  
1      Sportz  Petrol   Manual  18174  4.50  2015  Hyundai Grand i10
```

We can see that there are some outliers present in the year column and kms column so we used zscore method to remove the outliers.



We used label encoder for encoding the categorical columns for converting its object type data to numerical so that it will help us in prediction models.

We also saw some skewness present in the data so we used power transform method to remove skewness.

```
] : #using power transform method to remove skewness.  
from sklearn.preprocessing import power_transform  
x[['model', 'fuel', 'variant', 'year', 'brand', 'kms']] = power_transform(x[['model', 'fuel', 'variant', 'year', 'brand', 'kms']], method='yeo-johnson', return_df=True)  
x
```

	model	fuel	variant	kms	year	brand
0	-1.832044	0.886997	-1.559921	-0.887564	0.365395	0.178083
1	0.778853	0.886997	0.641058	-1.259981	0.033570	-0.783340
2	0.780961	0.886997	0.641058	-0.142430	-0.290780	-0.576404

- Hardware and Software Requirements and Tools Used

Hardware specifications are :-

Ryzen 5

16gb ram

RTX 2070 super graphics

Software used :-

Operating system : windows 10

Jupyter notebook and anaconda navigator- for coding and using data analytics tools and libraries.

Libraries used are matplotlib and seaborn for visualization purpose and scipy and sklearn for building models and pre-processing of data.

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

We saw that the data have some categorical data for which we used label encoder to convert the object type data to numerical.

We found null or nan values present in the data so we dropped them as the data loss was less than 1%.

We found outliers and removed them with zscore method as we cannot proceed to model building without removing the outliers.

We removed the skewness from variables using power-transform method.

We used standard scaler to scale the data.

- **Testing of Identified Approaches (Algorithms)**

We split the data into into X and Y where x hold all the input variables after perfroming all the EDA and cleansing of data and Y holds the output variable which is price column.

The data is in float type which clearly indicates that it's a regression problem and will need regression models to train test and predict the output.

We used train test split to split the data and perform machine learning tasks.

- Run and Evaluate selected models

We used `train_test_split` to split the data into `xtrain`, `xtest` and `ytrain`, `ytest` with test size set to 0.20 which is 20% of the data set. It means that `ytest` and `xtest` will hold 20% of the random data which the model will use for testing purpose and predict the data for the `xtest` and check how accurate it is with `ytest`.

Below is the snapshot of the code :-

```
] : #making a List of the regression models which are to be tested with the data set
models=[GradientBoostingRegressor(),LinearRegression(),Ridge(),BayesianRidge(),SGDRegressor(),SVR(),
AdaBoostRegressor(),KNeighborsRegressor(),RandomForestRegressor(),BaggingRegressor(),
DecisionTreeRegressor(),ExtraTreesRegressor()]

] : #making a for loop to check the models and their mean cross_val score with scoring set to r2
for i in models:
    xtrain,xtest,ytrain,ytest=train_test_split(x,y,random_state=82,test_size=0.20)
    score=cross_val_score(i,xtrain,ytrain,cv=5,scoring='r2').mean()
    i.fit(xtrain,ytrain)
    ypred=i.predict(xtest)
    if r2_score(ytest,ypred)>score:
        diff=r2_score(ytest,ypred)-score
    else:
        diff=score-r2_score(ytest,ypred)
    print(i)
    print('mean cross_val_score',score)
    print('r2',r2_score(ytest,ypred))
    print('diff',diff)
    print('mean_abs_error',mean_absolute_error(ytest, ypred))
    print('\n')
```

We included cross val score and cv set to 5 and checking the mean score. Random state is set to 85, we will be checking the r2 score and difference between the r2 score and cross val mean score. Where diff represents the difference in cross_val mean score and r2_score and check the difference. We can't choose the model if the difference is high as it leads to overfitting and underfitting problems.

Below is the image of the results that we got from the above code:-

```

GradientBoostingRegressor()
mean cross_val_score 0.7386624908686511
r2 0.8333042812830935
diff 0.09464179041444243
mean_abs_error 2.1505531050120816

LinearRegression()
mean cross_val_score 0.3407474070410016
r2 0.41737142351620193
diff 0.07662401647520034
mean_abs_error 4.380030928841576

Ridge()
mean cross_val_score 0.34075166383788824
r2 0.4173852520065292
diff 0.07663358816864096
mean_abs_error 4.379549729935338

BayesianRidge()
mean cross_val_score 0.3408016133900258
r2 0.41755063863873054
diff 0.07674902524870475
mean_abs_error 4.3736895528676785

SGDRegressor()
mean cross_val_score 0.3393485835793144
r2 0.41874061234738014
diff 0.07939202876806573
mean_abs_error 4.331451408445702

SVR()
mean cross_val_score 0.40985021219438833
r2 0.5491764020773742
diff 0.13932618988298584
mean_abs_error 2.6575853821110718

```

```

AdaBoostRegressor()
mean cross_val_score 0.32628208542883386
r2 -0.13775564838629117
diff 0.46403773381512503
mean_abs_error 7.026973807326548

KNeighborsRegressor()
mean cross_val_score 0.5371545589272048
r2 0.7457970021885725
diff 0.20864244326136772
mean_abs_error 2.181066430469442

RandomForestRegressor()
mean cross_val_score 0.7935737123978378
r2 0.8986855212353098
diff 0.10511180883747195
mean_abs_error 1.345129769285925

BaggingRegressor()
mean cross_val_score 0.7810836476393244
r2 0.8997751492990188
diff 0.11869150165969444
mean_abs_error 1.3952728921506599

DecisionTreeRegressor()
mean cross_val_score 0.682825264330047
r2 0.8140769438094445
diff 0.13125167947939753
mean_abs_error 1.6122099202834366

ExtraTreesRegressor()
mean cross_val_score 0.7660888233564749
r2 0.8851413617815753
diff 0.11905253842510044
mean_abs_error 1.4638561854148218

```

from the above results what we found is that our best performing models:

Bagging regressor is providing us with the accuracy of close to 90% and least cross val difference.

random forest is giving us the accuracy of close to 90% with less cross val difference.

extratrees is giving us the accuracy of 88%

- Key Metrics for success in solving problem under consideration

We used r2_score for checking accuracy of the model.

We used cross val score and checking the mean score with sv set to 5 and checking the difference in the mean cross val score and r2 score to avoid other problems and choose the models with best accuracy and least difference.

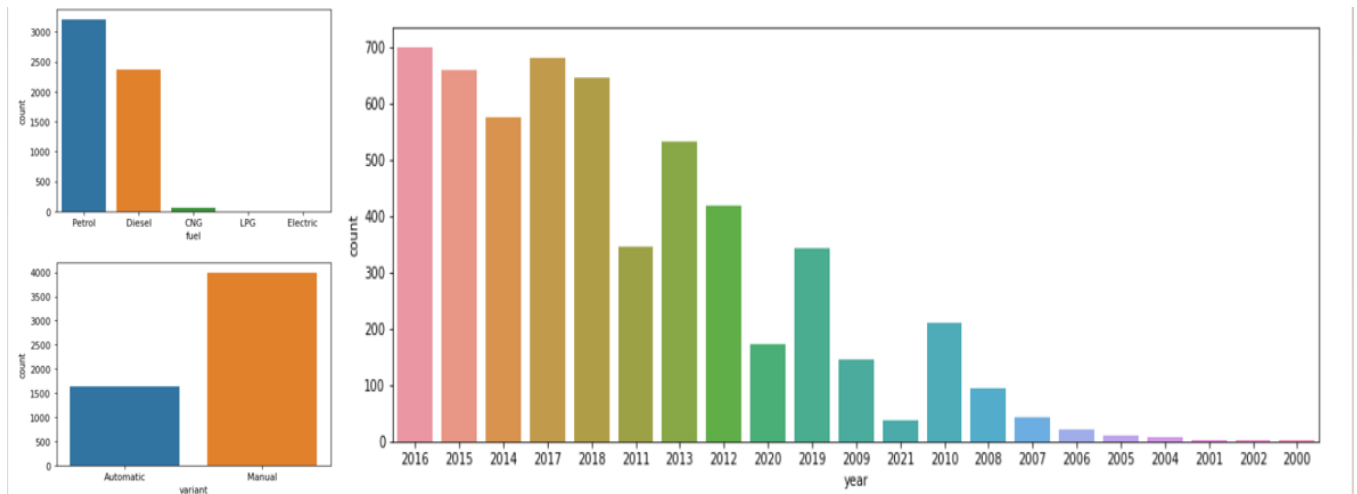
We used hyper tuning the models :-

Bagging regressor	Random forest Regressor
<pre>]: parameters={'n_estimators':[10,500], 'max_features':[0.001,0.01,0.1,1.0]} clf=GridSearchCV(BaggingRegressor(),parameters,cv=5,scoring='r2') clf.fit(x,y) clf.best_params_ </pre>	<pre>]: parameters={'n_estimators':[10,50,100], 'criterion':['mse', 'mae']} clf = GridSearchCV(RandomForestRegressor(), parameters,scoring="r2") clf.fit(x,y) clf.best_params_ </pre>
<pre>]: {'max_features': 1.0, 'n_estimators': 500} </pre>	<pre>]: {'criterion': 'mse', 'n_estimators': 100} </pre>
TESTING ON THE WHOLE DATASET	TESTING ON THE WHOLE DATA
<pre>]: x_train,x_test,y_train,y_test=train_test_split(x, y,random_state = 82,test_size=0.20,shuffle=True) br=BaggingRegressor(n_estimators=500,max_features=1.0) br.fit(x_train,y_train) y_pred=br.predict(x_test) score=r2_score(y_test,y_pred) print('r2_score : ',score) </pre>	<pre>]: x_train,x_test,y_train,y_test=train_test_split(x, y,random_state=82,test_size=0.20,shuffle=True) rfr=RandomForestRegressor(n_estimators=100,criterion='mse',max_features='log2') rfr.fit(x_train,y_train) y_pred=rfr.predict(x_test) score=r2_score(y_test,y_pred) print('r2_score : ',score) </pre>
<pre> r2_score : 0.9762695310512848 </pre>	<pre> r2_score : 0.975856357321024 </pre>
TRAINED DATA	TRAINED DATA
<pre>]: x_train,x_test,y_train,y_test=train_test_split(x, y,random_state = 82,test_size=0.20,shuffle=True) br=BaggingRegressor(n_estimators=500,max_features=1.0) br.fit(x_train,y_train) y_pred=br.predict(x_test) score=r2_score(y_test,y_pred) print('r2_score : ',score) print('mean_abs_error',mean_absolute_error(ytest, ypred)) </pre>	<pre>]: xtrain,xtest,ytrain,ytest=train_test_split(x, y,random_state=82,test_size=0.20,shuffle=True) rfr=RandomForestRegressor(n_estimators=50,criterion='mse',max_features='log2') rfr.fit(xtrain,ytrain) ypred=rfr.predict(xtest) score=r2_score(ytest,ypred) print('r2_score : ',score) print('mean_abs_error',mean_absolute_error(ytest, ypred)) </pre>
<pre> r2_score : 0.9018295094214519 mean_abs_error 1.5100700493483485 </pre>	<pre> r2_score : 0.8895618384501681 mean_abs_error 1.5020647032772365 </pre>

What we observed was:-

- After hyper tuning we got close to 90% accuracy and with less error with random forest regressor.
- after hyper tuning on Bagging regressor we got the accuracy of 90%.

Visualizations



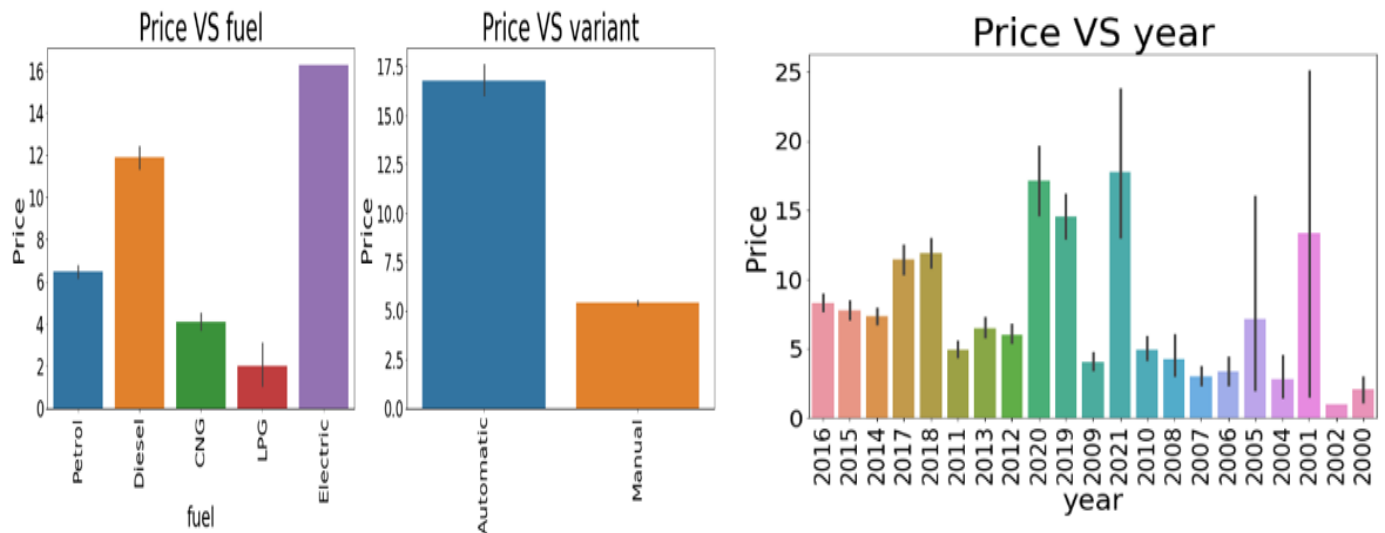
observations:

Most of the used cars are in petrol and then diesel. we can also see some of the cars with cng fuel, as the market is growing for electric vehicles we dont see a lot of electric cars being sold as used cars.

manual cars are what cost low when it comes to used cars and it is more likely to be sold in used cars compared to automatic.

most of the used cars for sale comes from the year 2016, 2017, 2018¶

we compared other variables with price and how they affect each other below are the images and observations:-



observation from the above graphs:

What we saw from price vs fuel graph is that lpg used cars have the lowest price compared to all other cars as there are very less lpg pumps and demand for them is less

Price for diesel engine cars is more as diesel is less expensive than petrol and widely available

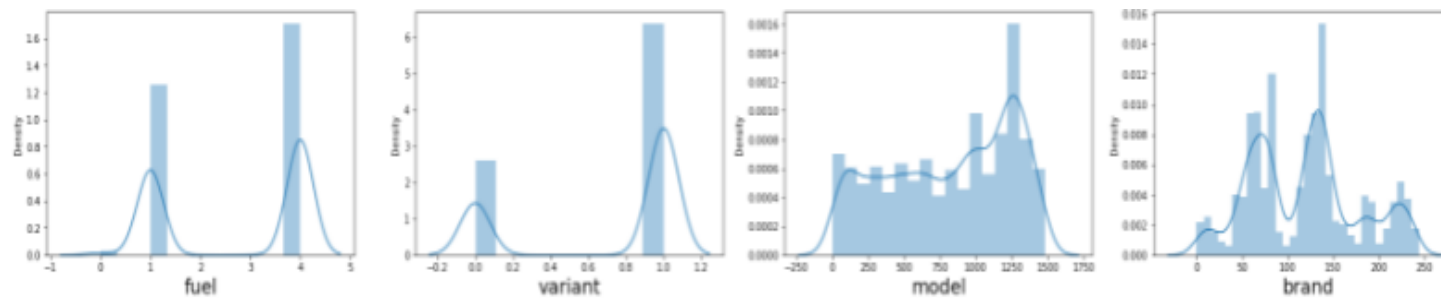
Price for petrol cars is moderate

Price for electric cars is the highest as they are newly introduced in the market and it cost very less compared to other fuels.

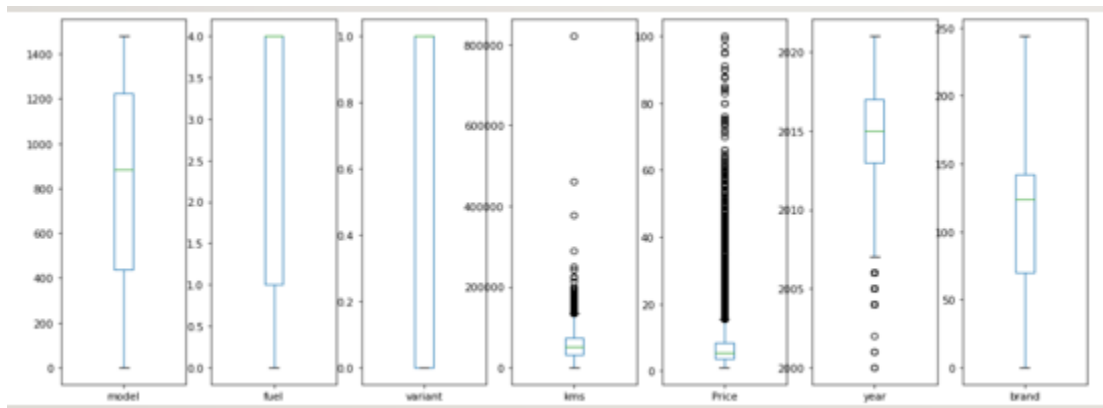
Price for automatic cars is more compared to manual cars as automatic cars provide ease in driving and have high demand compared to manual cars.

When we checked price vs year graph we found that cars are expensive if the model year is from 2015-2021

Older the car, less is the price of the car and vice-versa.



we can see biomodal distribution in the graphs.



we can see that there are some outliers present in the year column and kms column and we used zscore method to remove the outliers.



observation:

Brand and model columns are very less correlated to the price column.

variant year and fuel are very highly correlated to the price column and variant being the highest compared to the other two.

- Interpretation of the Results

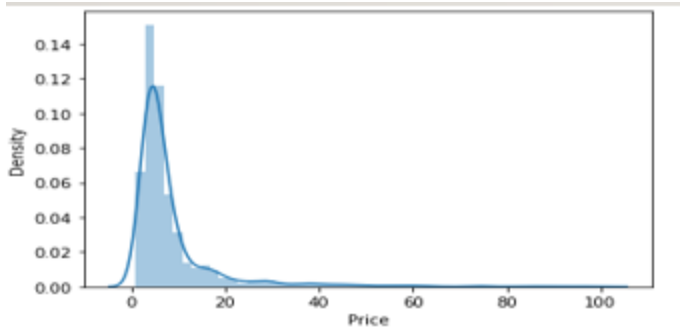
1) We saw that the data was not in its finest form we need to clear some nan values and outliers and clean the data.

2) We also saw biasing in the data which we removed using power transform method.

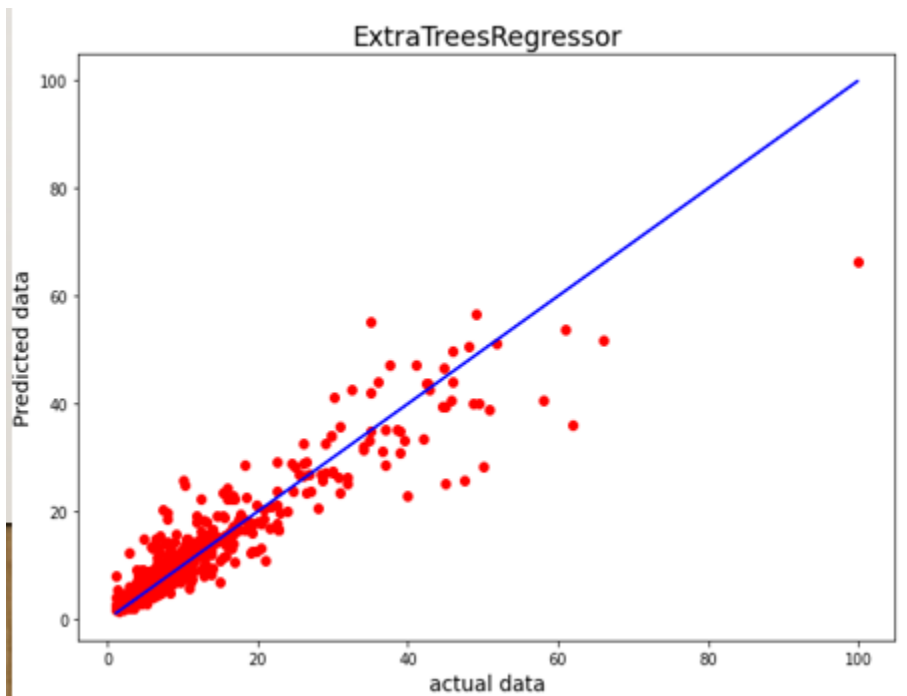
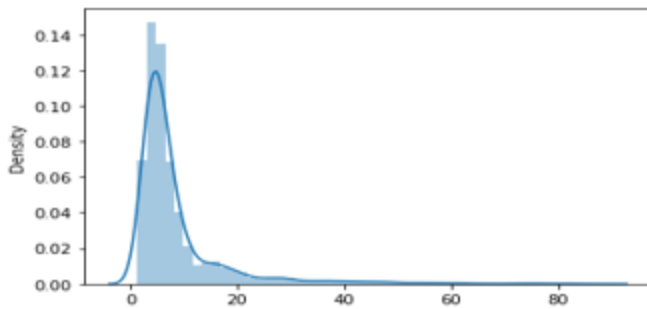
3) From visualization we found the positively correlated variables such as year and negatively correlated variables such as variant with the output variable which is price.

4) We found the best performing model after checking the metrics such as r^2 _score, cross val mean score, and mean_squared_error which is Bagging Regressor with the accuracy of 90% after hyper tuning.

5) After selecting the models we compared the model's with the predicted values and actual value.



:AxesSubplot:ylabel='Density'>



We can see that the actual data and predicted data are very close to each other. This indicates that our model is giving good results with accuracy of 90%.

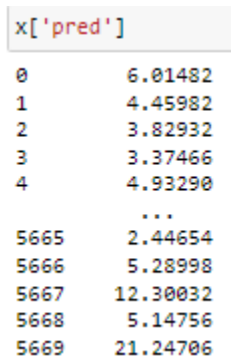
Conclusion

- Key Findings and Conclusions of the Study

We had to perform analysis on the train data and clean the data and build a model which will predict the used car prices for the test data.

We did some visualization and pre-processing and after scaling the data we were able to build a model with 90% accuracy.

We used the model to predict used cars prices for the data. Here is the image of the results:-



```
x['pred']  
0      6.01482  
1      4.45982  
2      3.82932  
3      3.37466  
4      4.93290  
...  
5665    2.44654  
5666    5.28998  
5667   12.30032  
5668    5.14756  
5669   21.24706
```

- Learning Outcomes of the Study in respect of Data Science

We found the important features and how do they affect positively and negatively with sale price of the used cars and used them to predict the used cars prices using machine learning models.

I also found out that there are so many variables which can affect the used cars prices which was something new to me.

People tend to check all these variables before buying the pre-owned cars and how the most important features can influence buying decision of a human being.

I learned that using data science it can help the business to make better strategies and focus on the areas which affects the output variables and make better decisions.

- Limitations of this work and Scope for Future Work

We can see the data was not in its finest form with some more analysis we can still improve the accuracy of the model.

The goal was to achieve 100% accuracy in predicting the house prices but we ended up with the accuracy of 90%.

Due to lack of experience I was not able to reach my goal but with working on more such projects and gaining more experience will help me to grow and develop more valuable skills to reach my goal in future.