USED CAR PRICE PREDICTION

Submitted by:

SURENDRAN G

INTRODUCTION

- ## Business Problem Framing

  Find the right price for the used cars based on various factors is a tough task for the user who wants to buy a used car in today's market. We are building an ML model which will take the required parameters as an input and return the car price based on its features.

- ## Conceptual Background of the Domain Problem

  This ML model will help to arrive the price of a used car based on factors requested. This would help the user to buy a used car at reasonable price and also helps the seller to derive a reasonable cost for the car that he/she is wish to sale.

- ## Review of Literature

  This research focuses on finding the best reasonable price for the used cars based on features like, Car name, Kilo meters driven, car release registration year, Transmission, Owner, fuel_type and seller type. We tried to extract how correlated all these features to the selling price and derived a model which gives the best accuracy.

  This model will help both buyer and seller of used cars to buy/sell a used car at reasonable price.

- ## Motivation for the Problem Undertaken

  Cardheko is a web service which connects the buyer and seller of a used car. This ML model will help the seller to sell his/her used car at reasonable price also helps the buyer to buy a used car at reasonable price which will gain the customer confidence on Cardheko

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  We had outliers in features like, selling price, kilo meters driven and Age of car. We had fixed the outliers using different techniques like IQR imputation and manual imputation. Mainly manual imputation applied on sales price based on Transmission type since Automatic cars generally sells at high price comparative to Manual Cars. Applying general IQR would be a good approach which will impact the model to provide wrong results.

  We had also remove the skewness of data using transformation techniques as follows.

  Sale price- Boxcox- sales price feature followed pareto distribution hence applied boxcox after applying all transformation with the help of probability plots

- Data Sources and their formats

  The complete Data set is scrapped from Cardheko web service using selenium. The scrapped data has 8 features excluding product URL are as follows.,

- Data Preprocessing Done

  We had done the following Data processing steps as follows,

  1. Handling outliers in numerical features
  2. Correcting skewness of numerical features
  3. Encoding for categorical variables

- Data Inputs- Logic- Output Relationships

  Final model will need an input as shown below with the below processing steps in order to get the best result. Every single input impacts the final model.

```
1 Chennai.drop('Sale_Price',axis=1)
```

	Car_Name	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age_of_car
0	Ford Endeavour 3.2 Trend AT 4X4	60000	Diesel	Dealer	Automatic	First owner	4

```
1 Chennai['Car_Name']=encoder.transform(Chennai['Car_Name'])
```

```
1 Chennai['Kms_Driven']=np.log(Chennai['Kms_Driven'])
```

```
1 Chennai
```

	Car_Name	Sale_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age_of_car
0	287	NaN	11.0021	Diesel	Dealer	Automatic	First owner	4

```
1 chennai_test=np.array([287,11.0021,4.0,0,1,0,0,0,1,0,0,1,0,1,0,0,0,0])
```

- State the set of assumptions (if any) related to the problem under consideration

  Assumptions on this model are as follow.,

  1. Selling price given for each product in Cardheko is reasonable
  2. Market price given for each product in Cardheko is correct

- Hardware and Software Requirements and Tools Used

  Software's used are python 3.7.6, selenium

  Libraries used are – Pandas , Numpy , Matplot , seaborn , scikit-learn

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  As said above intention of this model is to derive the best reasonable price. To achieve this we had collected data from Car dheko. Then the scrapped data went through data pre-processing steps as mentioned above.

  Since the expected output is a numerical feature, algorithms used to develop this model are regression models.

  We had tried regression models like, Linear regression, ADA boost regression, Random forest regression, KNN regression and XGBoost regression.

- Testing of Identified Approaches (Algorithms)

Algorithms used to test the data set and find the best performing models are as follows.,

1. Linear regression
2. ADA boost regression
3. Random forest regression
4. KNN regression
5. XGBoost regression.

- Run and Evaluate selected models
    a. Linear regression

```
LR=LinearRegression()
LR.fit(x_train,y_train)
LR_pred=LR.predict(x_test)
r2score=r2_score(y_test,LR_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, LR_pred))
print('Mean squared error: ', mean_squared_error(y_test,LR_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,LR_pred)))

```

```
R2score is :  0.7339007184356163

Errors:
mean Absolute error:  0.3328609898784561
Mean squared error:  0.19525141507934599
Root mean squared error:  0.44187262313855336
```

---

    b. ADA boost regression

```
ADA=AdaBoostRegressor()
ADA.fit(x_train,y_train)
ADA_pred=ADA.predict(x_test)

r2score=r2_score(y_test,ADA_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, ADA_pred))
print('Mean squared error: ', mean_squared_error(y_test,ADA_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,ADA_pred)))
```

```
R2score is :  0.724638638663037

Errors:
mean Absolute error:  0.35004821750408727
Mean squared error:  0.20204750325945012
Root mean squared error:  0.4494969446608621
```

    c. Random forest regression

```
RFR=RandomForestRegressor()
RFR.fit(x_train,y_train)
RFR_pred=RFR.predict(x_test)

r2score=r2_score(y_test,RFR_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, RFR_pred))
print('Mean squared error: ', mean_squared_error(y_test,RFR_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,RFR_pred)))
```

```
R2score is :  0.9092054900914839

Errors:
mean Absolute error:  0.16982628781330744
Mean squared error:  0.06662083579050997
Root mean squared error:  0.2581101233785881
```

d. KNN regression

```
KNN=KNeighborsRegressor(n_neighbors=2)
KNN.fit(x_train,y_train)
KNN_pred=KNN.predict(x_test)
r2score=r2_score(y_test,KNN_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, KNN_pred))
print('Mean squared error: ', mean_squared_error(y_test,KNN_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,KNN_pred)))
```

```
R2score is :  0.9203705102864217

Errors:
mean Absolute error:  0.1415985986488276
Mean squared error:  0.058428457443469424
Root mean squared error:  0.2417197911704158
```

e. XGBoost regression.

```
XGB=XGBRegressor()
XGB.fit(x_train,y_train)
XGB_pred=XGB.predict(x_test)
r2score=r2_score(y_test,XGB_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, XGB_pred))
print('Mean squared error: ', mean_squared_error(y_test,XGB_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,XGB_pred)))
```

```
R2score is :  0.941510498136797

Errors:
mean Absolute error:  0.13364914956534207
Mean squared error:  0.04291690657313285
Root mean squared error:  0.20716396060399322
```

- Key Metrics for success in solving problem under consideration

  What were the key metrics used along with justification for using it? You may also include statistical metrics used if any.

The R-squared statistic provides a measure of fit. It takes the form of a proportion—the proportion of variance explained—and so it always takes on a value between 0 and 1. In simple words, it represents how much of our data is being explained by our model.


```
R2score is :  0.9203705102864217

Errors:
mean Absolute error:  0.1415985986488276
Mean squared error:  0.058428457443469424
Root mean squared error:  0.2417197911704158
```

```
from sklearn.model_selection import cross_val_score
print("Linear regression cross validation accuracy is :", cross_val_score(LR,x,y,cv=10).mean())
```

```
Linear regression cross validation accuracy is : 0.5145052636087221
```

```
print("Random forest regressor cross validation accuracy is :", cross_val_score(RFR,x,y,cv=10).mean())
```

```
Random forest regressor cross validation accuracy is : 0.7521723470006277
```

```
print("ADA boost regressor cross validation accuracy is :", cross_val_score(ADA,x,y,cv=10).mean())
```

```
ADA boost regressor cross validation accuracy is : 0.49750116497543573
```

```
print("KNN cross validation accuracy is :", cross_val_score(KNN,x,y,cv=10).mean())
```

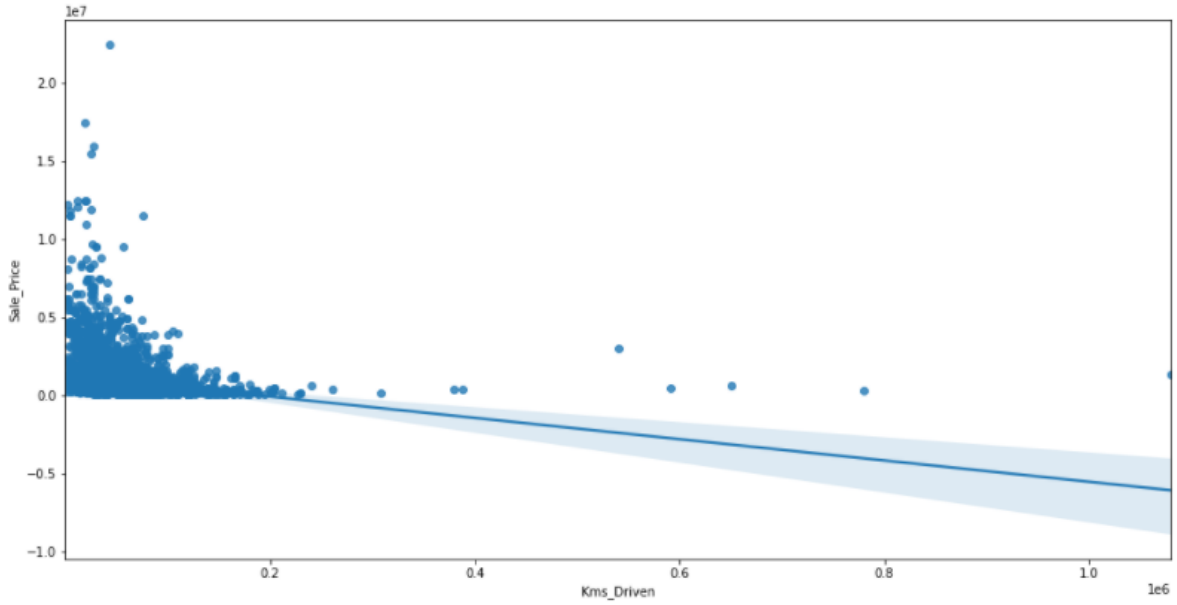
```
KNN cross validation accuracy is : 0.793111198302011
```

```
print("XGB cross validation accuracy is :", cross_val_score(XGB,x,y,cv=10).mean())
```

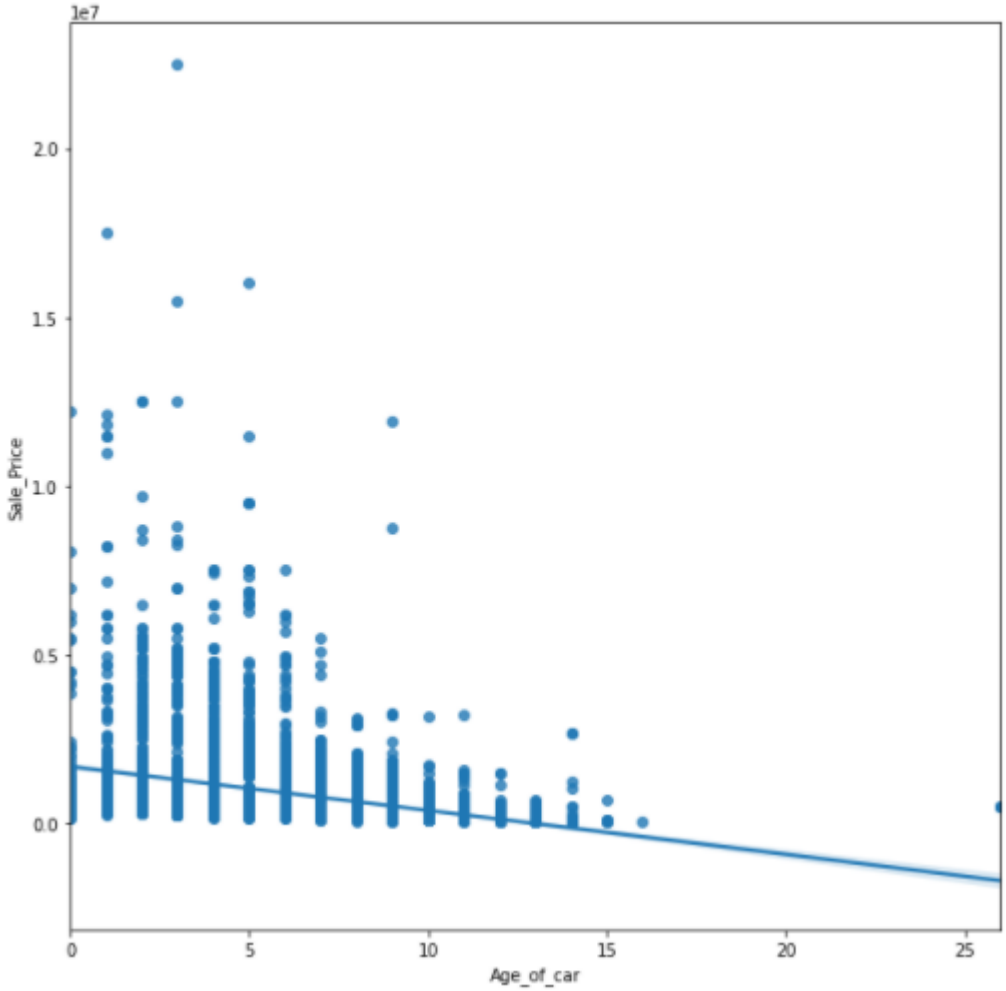
```
XGB cross validation accuracy is : 0.8425295750964314
```

- Visualizations

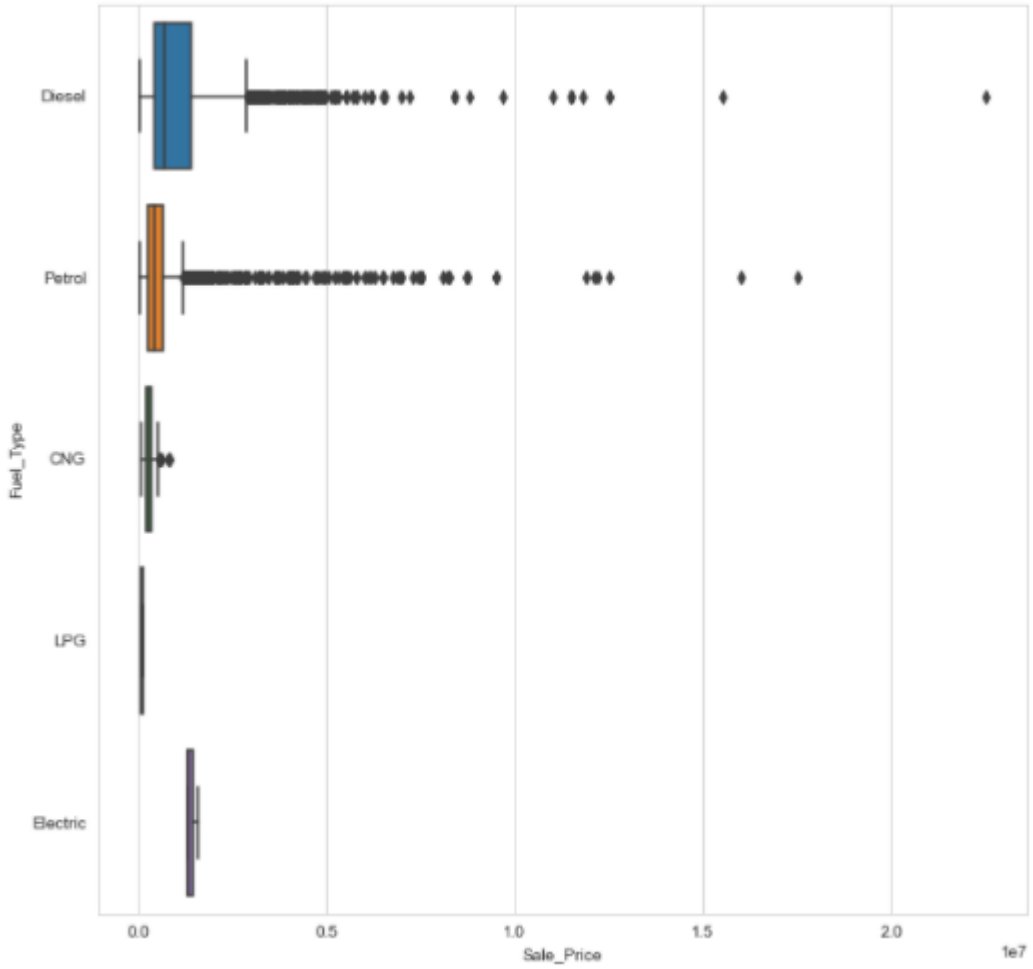
Sale price is reduced when kilo meter driven increase as shown below.



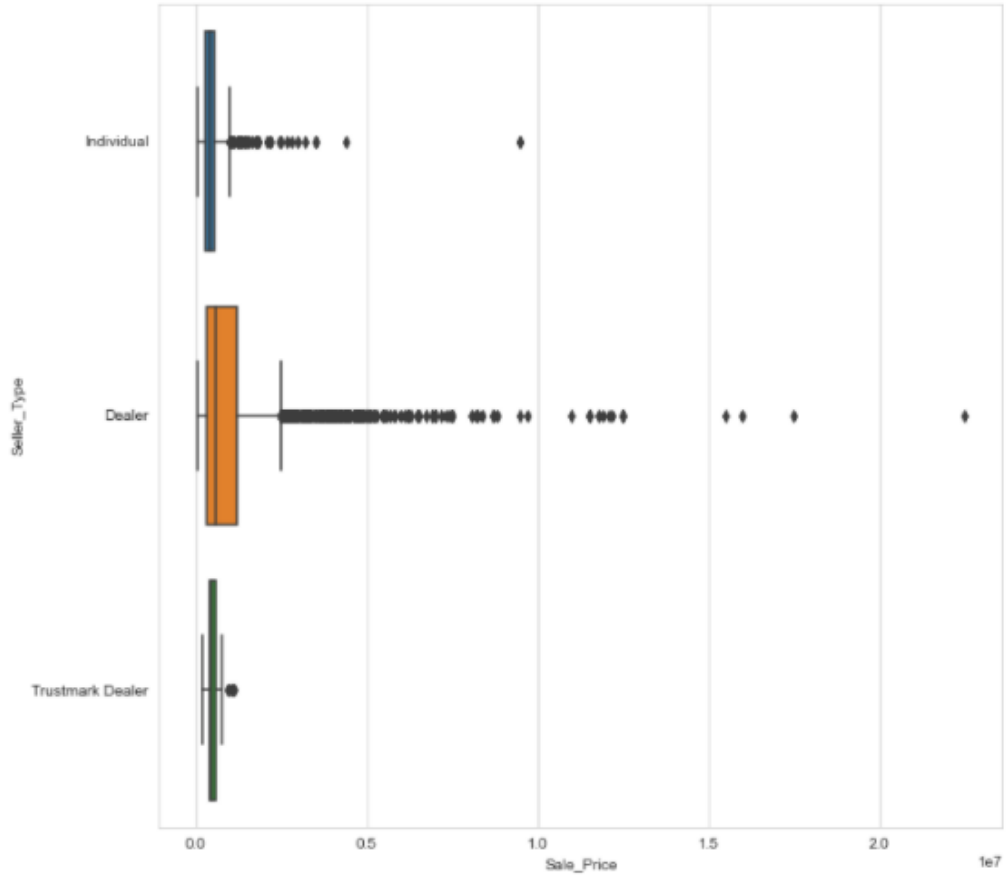
Sale price is reducing when the age of car increases.



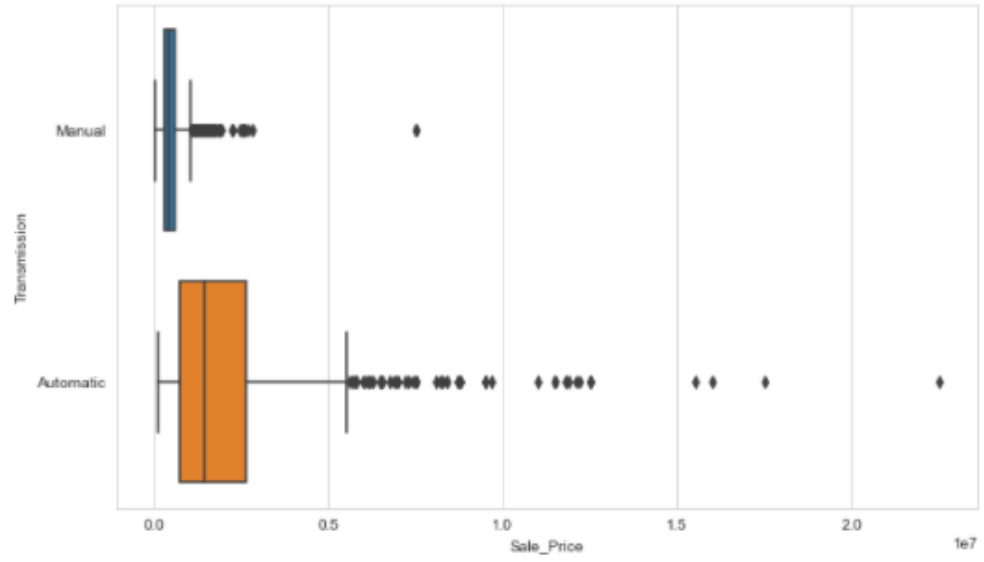
Diesel and petrol cars resale value is high.



Most of the cars are sold by delears and the high rate cars are also sold by Dealers.



Automatic cars are at high resale value, logic is current market price too is high.



CONCLUSION

- Key Findings and Conclusions of the Study
  1. Automatic cars are at high price
  2. Car price reduces as the age increases
  3. Car price reduces when kilo meters driven increases

- Learning Outcomes of the Study in respect of Data Science

  I had a chance to do an end to end project starts from data scrapping based on requirement and develop a model which can be deployed in different platforms.

  I had a chance to learn python techniques and advance excel techniques to clean a data I scrapped.

- Limitations of this work and Scope for Future Work

  Now the model is developed and tested for used car price in Delhi, Car price between cities may vary which can be accommodated in the future.

  Future scope can be adding current market price for all the data set and train model with current market price which may help us to get the best saleable price.