Car Price Prediction Project.

Submitted by:

K. Rahul Ramanujam

ACKNOWLEDEGMENT

1) The very basic idea behind gathering the data (Car price prediction) is to collect the details of the car and its economic value in the throbbing market. The new market situation has failed the already established models, thus, there is a utter need of new machine learning model which will predict the price for a new market.

2) The professional help that I've received was from our reporting manager Mr. Shubham Yadav, he has guided me the steps for resolving any problem by breaking down the process into a simpler manner.

3) The model building required a certain level of idea regarding the market's influence along with the other features on deciding the final pricing. The websites like CarDekho, Cars24 and other online retailer sites offer the best used car deals in the market. Thus, before going in for the model building, the sheer approach of the overall pricing and the respective influencing factors will help in better modelling.

INTRODUCTION

- Business Problem Framing

The business problem involved in this project is to create a new machine learning model for the newly Covid affected market. The market isn't what it used to be anymore; The new changes has riddled the minds of data scientists to come up with a new strategical approach in looking at the affected situation for a better solution in predicting the final selling price of the cars.

- Conceptual Background of the Domain Problem

The appropriate domain problems like customer retention projects, Housing projects are some of the best ones to refer and work. Those projects will give the basic idea of how the companies go through the concepts of retaining their position in the market by keeping the constant surveillance on their products and its reception from people. Customer sentiments are something that the firms value the most, they do whatever they can to please them. Thus, the current project will give a gist of how the final selling price works while considering the factors like 'The year of purchase', 'Kms driven' and other factors similar to those for the final reasonable pricing for the current affected market.

- Review of Literature

The basic approach which has intrigued me to work better in the project was the challenge, it has put me in a situation where I had to go through the used cars selling websites and look at the final pricing, factors involving to decide it.
The fact that the research required in order to understand the

market itself has the literature involved, like, the selling price of the cars has the following influencing features such as
Transmission: Whether it is manual or automatic type do decide the final pricing.
Owner: The no.of owners for a vehicle decides the final price.
Fuel: The type of fuel whether petrol or diesel will decide the price big time.
Kms driven, Year of purchase.
The above highlighted features are some the important ones which required research and taken down the notes of how it works during the final pricing.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  There aren't much mathematical calculations involved as the data is mostly of object datatype.

  The categorical features like Transmission, Fuel and Owner had to be converted to numerical as the machine model do not understand the strings data.

  The rest which are already numerical had outliers, the feature Selling Price and Kms_driven had outliers but they cannot be compromised since they signify how much the vehicle was used by the owner(s).

- Data Sources and their formats

  The data was web scraped from Cars24 website, it had raw data with symbols and other extra information, it required proofing before we take up to the modelling.

```
cars1.sample(5)
```

	name	year	selling_price	Km_driven	fuel	Transmission	owner
685	[Maruti Ciaz VXI MANUAL]	[2015]	₹5,05,199	33,936	Petrol	MANUAL	1st
1561	[Hyundai i10 ERA 1.1 IRDE MANUAL]	[2009]	₹1,71,099	1,19,014	Petrol	MANUAL	3rd
1135	[Ford Ecosport 1.5 TDCI TITANIUM PLUS MANUAL]	[2018]	₹9,30,499	25,143	Diesel	MANUAL	1st
1049	[Maruti Swift LXI MANUAL]	[2010]	₹3,69,199	46,079	Petrol	MANUAL	1st
840	[Maruti Ritz LXI MANUAL]	[2010]	₹2,12,499	1,21,252	Petrol + CNG	MANUAL	1st

There was rupee symbol, commas in price column, strings in owner column, brackets in year and name column. Thus, I had to edit that extra info out to make it easier.

	name	year	selling_price	km_driven	fuel	transmission	owner
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Manual	1st
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Manual	2nd
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Manual	3rd
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Manual	1st
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Manual	1st
...	...	...	...	...	...	...	...

Post-adjustments, the data like looked that shown in the pic above.

- ## Data Preprocessing Done

Data preprocessing is an integral part of the projects like these. The basic idea behind the data preprocessing process is to check for the missing values or any extra information present in the data. The presence of such can disturb the model in the end.

Thus, the data was checked for any missing values, it had none.

Then, the datatypes of the features were checked, there were 4 Object and 3 int datatype features.

Outliers can be bothersome, ‘Selling price’ and ‘Kms_driven’ features have outliers which was also shown in boxplot. But they weren’t compromised as they do have their respective parts in the role of affecting the final selling price.

- Data Inputs- Logic- Output Relationships

  The input was nothing but a raw data scraped from the e-commerce site which has a selling price and the other features having their role in the final pricing of cars respectively. Initially there were columns of object datatype, later they was converted to int for the machine models to understand. The relationship between the input and output has been made to be more clear post text-processing as it had many outcasts which the model couldn't read, thus, getting rid of those made the job easy in modelling stage. The final model is now able to predict the price of cars.

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

  The input data which nothing but the web-scaped data from Cars24, the features were pretty self-explanatory. Thus, the values in the respective features were assumed to be what were there in real.
  Another assumption was taken while the outliers were being dealt with the help of boxplot. The assumption that these outliers may help in the final model was taken, since the age, capacity and pricing of a vehicle are under testimony with the features and its respective values.

- Hardware and Software Requirements and Tools Used

  I used Google Colab instead of jupyter notebook for this particular project. There is a data of 8.5K in length, it is tough for a local machine to do the modelling with such huge data thus, I had to use colab since it has GPU support.

  Since there were features which impacts the final price big time, the use of visualization was needed. Matplotlib and Seaborn packages were imported.

  For data modelling we used Random Forest, XGBRegressor, Linear Regressor, KNR libraries imported form the packages like skelearn.ensemble, xgboost, sklearn.neighbors, sklearn.linear_model packages respectively.

To test if there was any Variance or Bias influencing the final result, we had used Cross Validation Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  For the problem-solving, I had to first look at the data properly in order to see what are the tools or libraries required for it. Then, I had to use follow the standard procedure like getting rid of the null values (if any), Outliers, converting the object datatype to numerical.
  The statistical approach during the analysis of features for its influence over the final pricing was important before heading to the visualization. The analysis helped in understanding the data better, in drawing the better insights off the plots.

- Run and evaluate selected models

  1) **XGBoost**: The XGBRegressor is imported from xgboost package, it is an ensemble method which helps in boosting the right performance for a better prediction.

```
regressor = XGBRegressor(
    gamma=0,
    learning_rate=0.1,
    max_depth=5,
    n_estimators=1000,
    n_jobs=16,
    objective='reg:squarederror',
    subsample=0.8,
    scale_pos_weight=0,
    reg_alpha=0,
    reg_lambda=1
)

model = regressor.fit(trainX, trainY)
```

```
# predict X train
trainPredict = model.predict(trainX)

# predict X test
testPredict = model.predict(testX)
```

```
R-Squared : 0.8965744864824657
MAE : 121393.77318544942
MSE : 67307617283.07374
RMSE : 259437.11624028228
Median : 69175.4375
```

maxr2_score function is defined to apply the following models to get best

possible random state value, r2 score.

```
def maxr2_score(mod,X,y):
    max_r_score=0
    for r in range(42,100):
        trainX, testX, trainY, testY = train_test_split(X, y,random_state = r,test_size=0.20)
        mod.fit(trainX,trainY)
        y_pred = mod.predict(testX)
        r2_scr=r2_score(testY,y_pred)
        print("r2 score corresponding to ",r," is ",r2_scr)
        if r2_scr>max_r_score:
            max_r_score=r2_scr
            final_r=r
    print("max r2 score corresponding to ",final_r," is ",max_r_score)
    return final_r
```

Cross validation score

```
#Cross_val_score fucntion

from sklearn.model_selection import cross_val_score
def model_evaluation(model,X,y):
    c_scores=cross_val_score(model,X,y,cv=5,scoring="r2")
    print("Mean r2 score for regressor: ",c_scores.mean())
    print("standard deviation in r2 score for regressor: ",c_scores.std())
    print(c_scores)
```

2) **Random Forest Regressor**: This is an ensemble method as well, GridSearchCV aid for finding the best parameters helped the model to perform well.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
import warnings
warnings.filterwarnings("ignore")
rfr=RandomForestRegressor()
parameters = {"n_estimators":[10,100,500]}
clf = GridSearchCV(rfr, parameters, cv=5,scoring="r2")
clf.fit(X, y)
clf.best_params_
```

```
{'n_estimators': 500}
```

```
rfr = RandomForestRegressor(n_estimators = 500)
maxr2_score(rfr,X,y)
```

```
r2 score corresponding to  82  is  0.6767688501925008
r2 score corresponding to  83  is  0.7410729332826695
r2 score corresponding to  84  is  0.7662073543008759
r2 score corresponding to  85  is  0.7819853401161335
r2 score corresponding to  86  is  0.6809804549246458
r2 score corresponding to  87  is  0.7240203091082433
r2 score corresponding to  88  is  0.7564037432939649
r2 score corresponding to  89  is  0.7122976744702572
r2 score corresponding to  90  is  0.7824387752346115
r2 score corresponding to  91  is  0.7297478487035991
r2 score corresponding to  92  is  0.785505943229363
r2 score corresponding to  93  is  0.6951781863181808
r2 score corresponding to  94  is  0.7730225502830161
r2 score corresponding to  95  is  0.6878393011404772
r2 score corresponding to  96  is  0.6452742249544627
r2 score corresponding to  97  is  0.755490544283829
r2 score corresponding to  98  is  0.7841946231535506
r2 score corresponding to  99  is  0.771952705746686
max r2 score corresponding to  72  is  0.8507026579442764

72
```

Cross Validating RFR

```
#Lets chcek Random forest using n_estimators=500

rfr=RandomForestRegressor(n_estimators=500)

#Lets check the cross_val_score for Random Forest Regressor

print("Random Forest Regressor\n\n")
model_evaluation(rfr,X,y)
```

```
Random Forest Regressor


Mean r2 score for regressor:  0.7464015997400051
standard deviation in r2 score for regressor:  0.0653241464176357
[0.79232897 0.70728444 0.80169212 0.63562843 0.79507405]
```

3) **KNR**: K-Neighbors Regressors is a sequential model, with the help of GrisdSearchCV, the best parameters were used to get the best out the model's performance.

```
from sklearn.neighbors import KNeighborsRegressor
knr=KNeighborsRegressor()
parameters = {"n_neighbors":range(2,30)}
clf = GridSearchCV(knr, parameters, cv=5,scoring="r2")
clf.fit(X, y)
clf.best_params_
```

```
{'n_neighbors': 3}
```

```
knr=KNeighborsRegressor(n_neighbors=3)
maxr2_score(knr,X,y)
```

```
r2 score corresponding to  82  is  0.5563289231790982
r2 score corresponding to  83  is  0.6308323097041245
r2 score corresponding to  84  is  0.6464507914451221
r2 score corresponding to  85  is  0.6881466808214607
r2 score corresponding to  86  is  0.5740383678502088
r2 score corresponding to  87  is  0.6269255223131642
r2 score corresponding to  88  is  0.6937116804785142
r2 score corresponding to  89  is  0.612672906201355
r2 score corresponding to  90  is  0.6729395509854124
r2 score corresponding to  91  is  0.6092078465347966
r2 score corresponding to  92  is  0.6888345288292339
r2 score corresponding to  93  is  0.5425060619261377
r2 score corresponding to  94  is  0.7330838447645698
r2 score corresponding to  95  is  0.6050899049115666
r2 score corresponding to  96  is  0.5643540186730214
r2 score corresponding to  97  is  0.6885359681250602
r2 score corresponding to  98  is  0.6751611379447691
r2 score corresponding to  99  is  0.6678669560537625
max r2 score corresponding to  63  is  0.7353278706867139

63
```

Cross Validating KNR

```
knr=KNeighborsRegressor(n_neighbors=3)

#Lets check the cross_val_score for Random Forest Regressor

print("KNR \n\n")
model_evaluation(knr,X,y)
```

```
KNR


Mean r2 score for regressor:  0.6536035356661914
standard deviation in r2 score for regressor:  0.05595409332593448
[0.6672741  0.66550248 0.69144038 0.54481591 0.69898481]
```

- Visualizations:

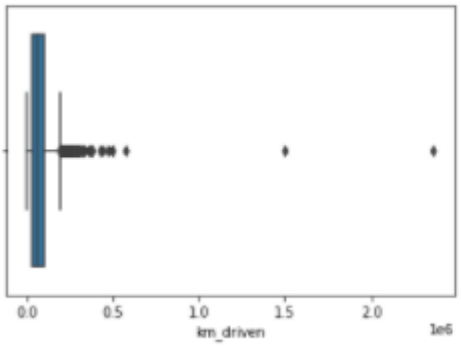
The important aspects of the project were the inferences from the data

which was important to understand the relation among the features involved. Some of the important inferences were drawn are shown below.

1) Uni-Variate Analysis:

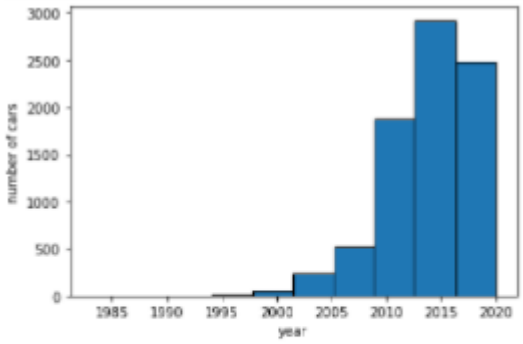
```
#kms driven outlier visualized
sns.boxplot(x=cars['km_driven'])
plt.xlabel('km_driven')
```

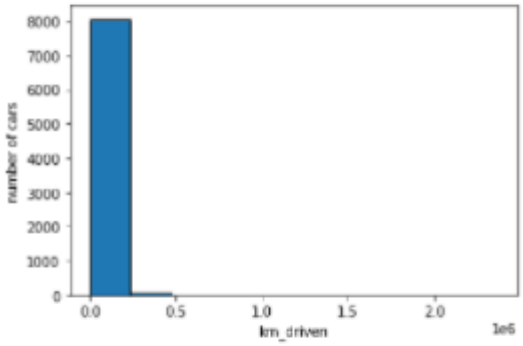
```
Text(0.5, 0, 'km_driven')
```

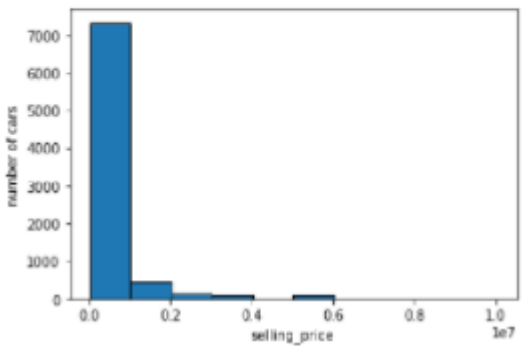


The amount of distance driven in kms by a car can be varied. The above plot evidents the possibility of highest possible distance covered by a car, i.e. 23,60,457 kms is the max value in that column, next is around 15lac kms of distance.

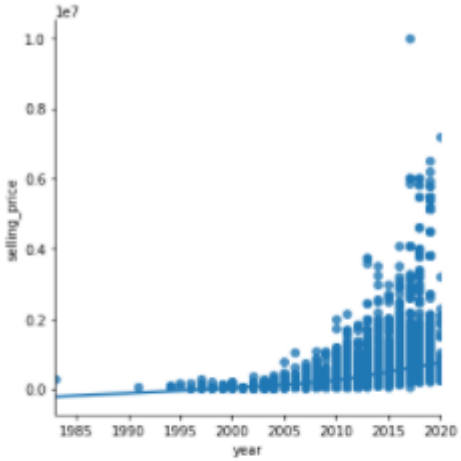
2) Bi-Variate Analysis







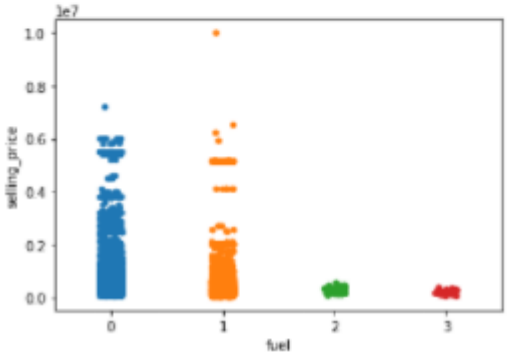
The above histogram shows the no.of cars sold with respective to the features



Clearly, the plot suggests that the selling price hikes with respect to year ele

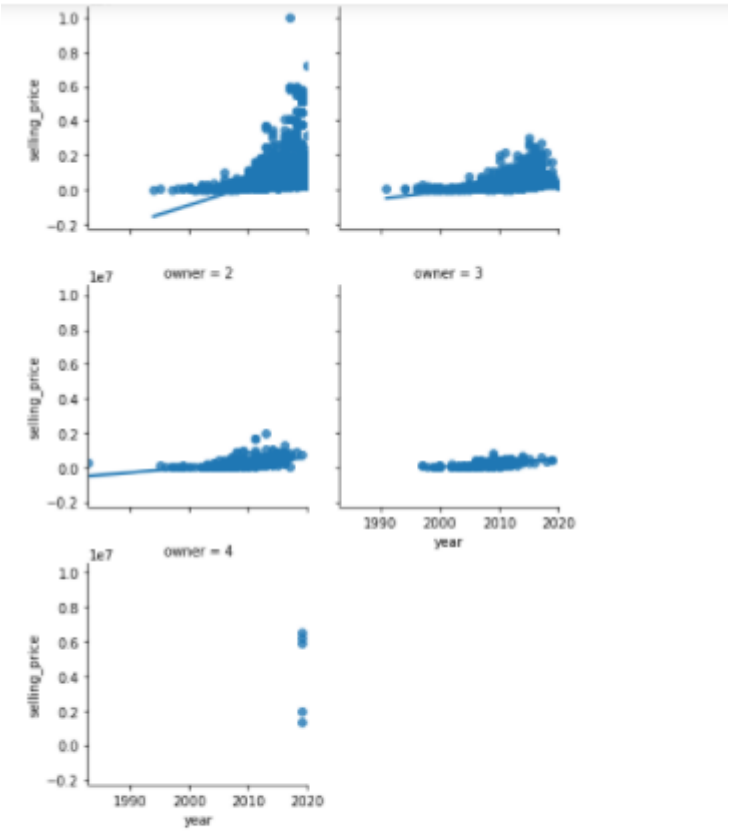
Newer the model is, more will be the selling price. It's natural to assume the

```
#stripplot comparing selling price and fuel type
sns.stripplot(y=cars['selling_price'],
              x=cars['fuel'])
plt.show()
```



First plot, it explains the relation like how much the feature 'Year' influences the final 'selling price'.
Second plot, it shows how the type of fuel can impact the final selling price of cars.

3) Multi-Variate Analysis:



No.of owners and the year of purchase affects the selling price in the following wise:

1) lesser the count of owners, earlier year of purchase will hike the selling price

2) More the count of owners, late year of purchase will depreciate the selling price.

- Interpretation of the Results

  Post Visualization, it is clear that each feature impacts the final selling price differently. The inferences drawn from the plots have helped to rank the importance for the better modelling.

  The categorical features were plotted with the help of histograms and that has helped to understand each categorical value's impact on the selling price of car.

  Machine model's success rate can be dependent on how the data is split, how the values of features are sorted and many other factors. The final models did yield better results.

CONCLUSION

- Key Findings and Conclusions of the Study

1) I've learned that the type of data can play a major part in the final prediction. More refined the data is, better will be the prediction by the best-chosen model.
2) The role of ensemble models such as XGBoost, Random Forest plays a crucial part in the projects like these, the hyperparameters that were used in gradient boosting have played a major part in better prediction.

- Learning Outcomes of the Study in respect of Data Science

The market and its structure aren't rigid but more like a fluid/dynamic in nature. The recent pandemic situation is a best example to prove that market isn't structured but a volatile one which depends upon various factors.

The role of a Data Scientist has come in handy for such markets from falling apart, we do research on the previous market's situation and its revenue building tactics to come up with a strategic plan and the respective appropriate model to predict the right pricing for the current affected market

I've learned that the projects which has big data like this one, must be dealt with utmost planning, since, the training time can take away all the time we have, thus, we need to choose those parameters which will work efficiently while taking less time to finish it.

- Limitations of this work and Scope for Future Work

  The only limitation I felt was to web-scrape the whole data by ourselves and then applying the modelling to it. Some websites do not stack their products well, they can be clumsy and ununiform. I expect that the clients to provide us with the data (clumsy or refined, anything that can be useful) to us.