FLIP ROBO

CAR PRICE PREDICTION

Submitted by:

SIDHARTH DAS

ACKNOWLEDGMENT

Firstly, the support I received to complete this project would be from my:

Mentor/SME @ FlipRobo Technologies, Ms.Sapna Verma, and

Data Trained Institute Inhouse Data Scientists and trainers.

Also referred to blogs and websites:

Stack Over Flow, Towards Data Science and Kaggle.

Professionals from Flip Robo Technologies and Data Trained Institute helped clear and give a clear picture on how to solve a particular problem.

INTRODUCTION

- Business Problem Framing

- Aim is to develop a new machine learning model to predict used car prices with respect to the current market conditions. 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. We have to make car price valuation model that can predict used car prices with respect to current market conditions by acquiring used car prices from websites.

- Conceptual Background of the Domain Problem

  A small understanding of economy and study on why the used car market rose post covid 19 would be very helpful.

- Review of Literature

  Scraped used car data from Car24 Website and tried to include most of the brands that are into car manufacturing in India. This study will help predict used car market accurately and help company to take necessary steps to maximize profits.

- Motivation for the Problem Undertaken

  The main motivation to undertake and create a model to predict used car price is to get updated or upgrade our understanding and get accurate price prediction post covid 19 impact.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  We have made sure all the columns do not have any type of symbols and commas so that we can easily pass them into a Machine Learning model. We have used labelencoder to encode the categorical columns and standard scaler to scale the data.

- Data Sources and their formats

  We have scraped data from Cars24 website.

- Data Preprocessing Done

  We removed index column and cleaned columns symbols and commas so that we can easily pass them into a Machine Learning model. We have used labelencoder to encode the categorical columns and standard scaler to scale the data.

- Data Inputs- Logic- Output Relationships

  If a certain car of certain brand has less age and have less kilo meters on it, the price of the car will be more. Yes, price also fluctuates with the brand and type of the car

- Hardware and Software Requirements and Tools Used
  a. Listing Requirements will start with a computer with i3 processing power and a dedicated GPU would be recommended as to process huge amount of data and save time. Anything more than i5 would be help.
  b. Use of Jupyter notebook is must, as this helps to carry out our whole project.
  c. Used Python language with Pandas, NumPy, Matplotlib, Seaborn. Pandas and NumPy helped in data importation and data wrangling. All the heavy work done was by using these two packages. Seaborn and Matplotlib was used to visualize and understand the data.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Used simple techniques that are essential and common for any project. Cleaning and visualizing of data, Pre-processing of data which included cleaning dropping of useless features. Lastly building a model by cross validating and by hyper parameter tuning.

- Testing of Identified Approaches (Algorithms)

  Listing down all the algorithms used for the training and testing:

  1) Decision Tree regressor.
  2) Random Forest regressor.
  3) AdaBoost regressor.
  4) Kneighbors regressor.
  5) Support Vector Regressor.
  6) Linear Regression.

  Gradient Boosting Regressor.

- Run and Evaluate selected models

  1) Linear Regression:

```
lr = LinearRegression()
lr.fit(x_train,y_train)
pred=lr.predict(x_test)
print("mean_squared_error:",mean_squared_error(y_test,pred))
print("mean_absolute_error:",mean_absolute_error(y_test,pred))
print("r2_score:",r2_score(y_test,pred))
print(lr.score(x_train,y_train))
```

```
mean_squared_error: 110654166570.68353
mean_absolute_error: 242788.96694069816
r2_score: 0.5221915492291604
0.31325030288925926
```

  2) Decision Tree regressor:

```
dt = DecisionTreeRegressor()
dt.fit(x_train,y_train)
pred = dt.predict(x_test)
print("mean_squared_error:",mean_squared_error(y_test,pred))
print("mean_absolute_error:",mean_absolute_error(y_test,pred))
print("r2_score:",r2_score(y_test,pred))
print(dt.score(x_train,y_train))
```

```
mean_squared_error: 38569202158.81047
mean_absolute_error: 85470.05735660848
r2_score: 0.8334568746745139
0.999999898169369
```

3) Random Forest Regressor:

```
rf = RandomForestRegressor()
rf.fit(x_train,y_train)
pred = rf.predict(x_test)
print("mean_squared_error:",mean_squared_error(y_test,pred))
print("mean_absolute_error:",mean_absolute_error(y_test,pred))
print("r2_score:",r2_score(y_test,pred))
print(rf.score(x_train,y_train))
```

```
mean_squared_error: 21013378521.557026
mean_absolute_error: 78432.70866167913
r2_score: 0.9092635176061552
0.9886824358698414
```

4) Knearest Neighbours Regressor:

```
kn = KNeighborsRegressor()
kn.fit(x_train,y_train)
pred = kn.predict(x_test)
print("mean_squared_error:",mean_squared_error(y_test,pred))
print("mean_absolute_error:",mean_absolute_error(y_test,pred))
print("r2_score:",r2_score(y_test,pred))
print(kn.score(x_train,y_train))
```

```
mean_squared_error: 78633459732.6394
mean_absolute_error: 165332.53366583542
r2_score: 0.6604580492718852
0.7291478026507958
```

5) SVR:

```
sv = SVR()
sv.fit(x_train,y_train)
pred = sv.predict(x_test)
print("mean_squared_error:",mean_squared_error(y_test,pred))
print("mean_absolute_error:",mean_absolute_error(y_test,pred))
print("r2_score:",r2_score(y_test,pred))
print(sv.score(x_train,y_train))
```

```
mean_squared_error: 260922995444.51953
mean_absolute_error: 317877.3980862304
r2_score: -0.12667435929035165
-0.11616297300253309
```

6) Ada Boost Regressor:

```
ab = AdaBoostRegressor()
ab.fit(x_train,y_train)
pred=ab.predict(x_test)
print("mean_squared_error:",mean_squared_error(y_test,pred))
print("mean_absolute_error:",mean_absolute_error(y_test,pred))
print("r2_score:",r2_score(y_test,pred))
print(ab.score(x_train,y_train))
```

```
mean_squared_error: 126624918913.78046
mean_absolute_error: 303723.36086382647
r2_score: 0.4532292979991047
0.5804474157006307
```

7) Gradient Boosting Regressor:

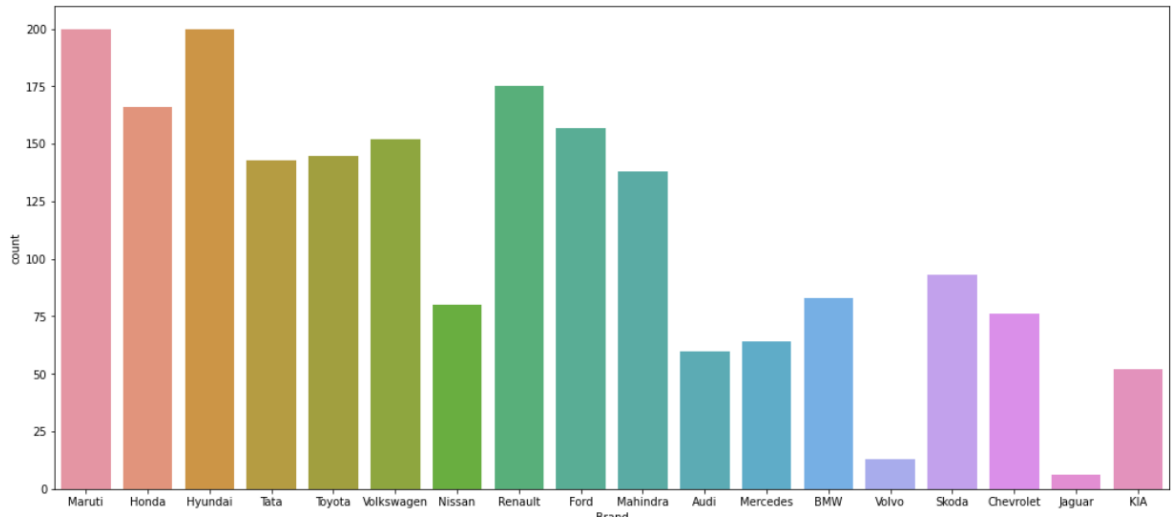
.

- Key Metrics for success in solving problem under consideration

  We have used model score to understand how good the model has trained itself with the training data. We have used r2 score to understand the how close the data are to the fitted regression line.
- Visualizations

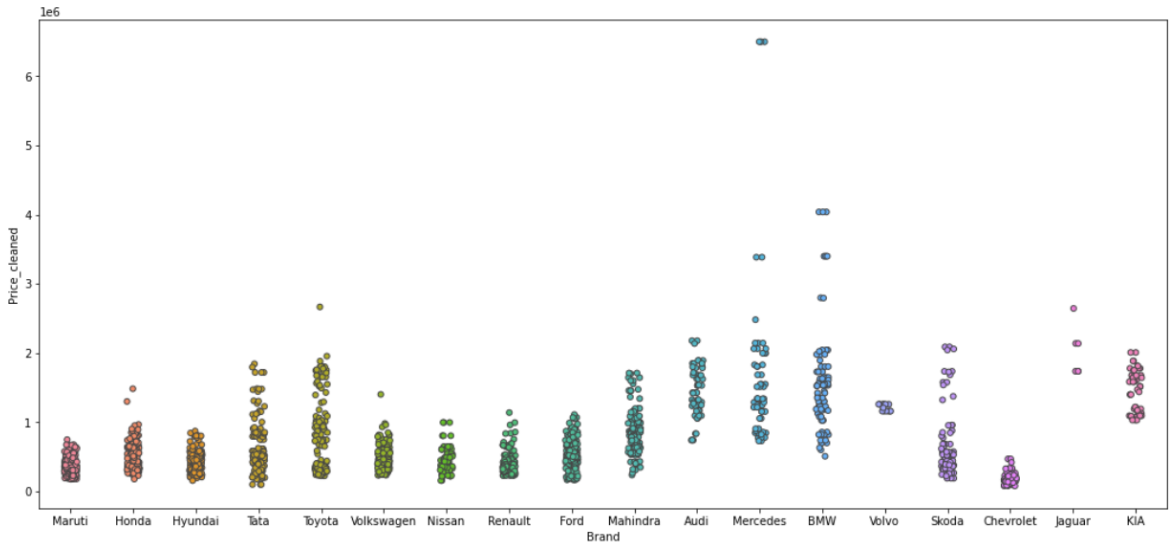
```
plt.figure(figsize=(18,8))
sns.countplot(car['Brand'])
```

```
<AxesSubplot:xlabel='Brand', ylabel='count'>
```




```
plt.figure(figsize=(18,8))
sns.stripplot(car['Brand'],car['Price_cleaned'], linewidth=1)
```

```
<AxesSubplot:xlabel='Brand', ylabel='Price_cleaned'>
```



- Interpretation of the Results

  If a car has less age and have run less kilometers, the price would be on higher side. The transmission, fuel type, location, variant and brand also equally important.

CONCLUSION

- Key Findings and Conclusions of the Study

  If a car has less age and have run less kilometers, the price would be on higher side. The transmission, fuel type, location, variant and brand also equally important.

- Learning Outcomes of the Study in respect of Data Science

  This study would help us understand which brand cars with certain a range of kilometers variant and other aspects to focus on so that we can create a better business model and take necessary steps to move forward.

- Limitations of this work and Scope for Future Work

  We would need to update our data and model so that we be up to date with current market and cars. A monthly refresh of the data would be necessary so that we do not outdate our understanding and result in wrong predictions to the relevant market.