Used Car Price Prediction

Submitted by:

Rohan V Borade

ACKNOWLEDGMENT

I have referred below resources that helped and guided me in completion of this project as below:

www.w3resource.com

www.towardsdatascience.com

www.stackoverflow.com

INTRODUCTION

- ## Business Problem Framing

With the covid 19 impact in the market, we have seen lot of changes in the car markets. 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.

- ## Conceptual Background of the Domain Problem

The conceptual knowledge about this problem is that due the covid 19 pandemic. There is various factor which are included while pricing the cars, some are as follows fuel price spare parts to replace the damaged one. For how many years car is has been used, condition of the Kilometres driven by cars etc.

- ## Review of Literature

With the recent arrival of internet portals, buyers and sellers may obtain an appropriate status of the factors that ascertain the market price of a used automobile. Lasso Regression, Ridge, and Regression Trees are examples of machine learning algorithms. We will try to develop a statistical model that can forecast the value of a pre-owned automobile based on prior customer details and different parameters of the vehicle.

- Motivation for the Problem Undertaken

The Motivation for the problem undertaken is to build t model to predict the price of the used for our clients. Who work with small trades, the 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.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

- Data Sources and their formats

  I have use data sources, data is fetched by myself using selenium, in excel.

- Data Pre-processing Done

  Data which prepared had Nan values and some exception handled signs which I took off while data cleansing.

  Data containing multiple formats like string, integers so I clean data by replacing zeros with NaN values & then replace NaN by mean method and mode method.

  Post this I used label encoder 2 sets one for float data & other for string data & converted all data into integers after that verified for non-null values.

- Data Inputs- Logic- Output Relationships

  Input data for feature list and target is in numeric format and hence classification model (XGB Regressor) best suits for this dataset.

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

  I have not considered any pre-assumption, project performance from beginning to end is based on data facts only.

- Hardware and Software Requirements and Tools Used

  Hardware Requirement-Laptop with below configurations-

  Windows Edition-Windows 10 Pro

  Processor-Intel(R)

  Memory-4 GB

  System Type-64-bit OS

  Software Requirement- Anaconda 3.7 & above, Jupiter Notebook 6.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Analytical Approach –Based on type of data by performing EDA I have decided which model to be used for this data.

- **Statistical Approach –** Data should be in scaled manner, it should not be distorted, for that I have replace all null values using mean method due to continuous data numbers.

- Testing of Identified Approaches (Algorithms)

  Below are Regression algorithms used for the training and testing this dataset.

- Linear Regression
- Lasso (Regularization)
- Ridge (Regularization)
- K-Neighbors Regressor
- Decision Tree Regressor
- Gradient Boosting Regressor
- XGB Regressor
- Ada Boost Regressor
- SVR

- Run and evaluate selected models

  Pls find below R2 score & their results also.

```
[95] max_scr=0
     for i in range(0,2000):
         x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=i,test_size=0.20)
         lr.fit(x_train,y_train)
         pred_train=lr.predict(x_train)
         pred_test=lr.predict(x_test)
         if round(r2_score(y_train,pred_train)*100,2)==round(r2_score(y_test,pred_test)*100,2):
             print("At random state",i," The model performs very well")
             print("At random_state:-",i)
             print("Training r2_score is:-", r2_score(y_train,pred_train)*100)
             print("Testing r2_score is:-", r2_score(y_test,pred_test)*100)
```

```
At random state 1263  The model performs very well
At random_state:- 1263
Training r2_score is:- 44.7148999220649
Testing r2_score is:- 44.7141665333755
At random state 1855  The model performs very well
At random_state:- 1855
Training r2_score is:- 44.73099906887015
Testing r2_score is:- 44.728629963156486
```

```
[101] from sklearn.linear_model import Lasso

      parameters = {'alpha':[.0001, .001, .01, .1, 1, 10],'random_state':list(range(0,10))}
      ls = Lasso()
      clf = GridSearchCV(ls,parameters)
      clf.fit(x_train,y_train)

      print(clf.best_params_)
```

```
{'alpha': 0.0001, 'random_state': 0}
```

```
[102] ls = Lasso(alpha=0.0001,random_state=0)
      ls.fit(x_train_b,y_train_b)
      ls.score(x_train_b,y_train_b)
      pred_ls = ls.predict(x_test_b)

      lss = r2_score(y_test_b,pred_ls)
      for j in range(2,10):
          lsscore = cross_val_score(ls,x,y,cv=j)
          lsc = lsscore.mean()
          print("At cv:-",j)
          print("Cross validation score is:-",lsc*100 )
```



```
[103] print("At cv:-",4)
      print("Cross validation score is:-",41.167 )
      print("R2_score is :-",43.264)
```

```
At cv:- 4
Cross validation score is:- 41.167
R2_score is :- 43.264
```

```
[106] from sklearn.linear_model import Ridge

      parameters = {'alpha':[.0001, .001, .01, .1, 1],'fit_intercept':[True,False],'normalize':[True,False],'copy_X':[True,False],'t
      rd = Ridge()
      clf = GridSearchCV(rd,parameters)
      clf.fit(x_train_b,y_train_b)

      print(clf.best_params_)
```

```
{'alpha': 0.01, 'copy_X': True, 'fit_intercept': True, 'normalize': True, 'random_state': 0, 'tol': 0.001}
```

```
[107] rd = Ridge(alpha=0.001, copy_X= True, fit_intercept= True, normalize=True, random_state= 0, tol= 0.001)
       rd.fit(x_train_b,y_train_b)
       rd.score(x_train_b,y_train_b)
       pred_rd = rd.predict(x_test_b)

       rds = r2_score(y_test_b,pred_rd)
       print('R2 Score:',rds*100)

       rdscore = cross_val_score(rd,x,y,cv=4)
       rdc = rdscore.mean()
       print('Cross Val Score:',rdc*100)
```

```
R2 Score: 45.19408844383788
Cross Val Score: 42.79392385952381
```

```
[119] xgb = XGBRegressor()
      xgb = XGBRegressor(n_estimators=1000, max_depth=7, eta=0.1, subsample=0.7, colsample_bytree=0.8)
      xgb.fit(x_train_b, y_train_b)
      xgb.score(x_train_b, y_train_b)
      pred_random = xgb.predict(x_test_b)

      xgbs= r2_score(y_test_b,pred_random)
      print('R2 Score:',xgbs*100)
      #cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)
      xgbscore = cross_val_score(xgb,x,y,cv=4)
      xgbc= xgbscore.mean()
      print('Cross Val Score:',xgbc*100)
```

```
[11:10:25] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
R2 Score: 88.85240510454851
[11:10:31] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[11:10:35] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[11:10:39] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[11:10:44] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
Cross Val Score: 82.23591062606556
```

- Key Metrics for success in solving problem under consideration
- Used cross validation R2 score to overcome under-fitting /over-fitting this model by deciding number of folds.
- Visualizations

  Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.

  If different platforms were used, mention that as well.

- Interpretation of the Results

  Visualisation shows outliers which need to be removed / corrected.

  Data Pre-processing done by performing EDA (Exploratory Data Analysis), checking for best accuracy score.

  Modelling done based on type of data as this is categorical data, we have to go with multiple classification models & finalise the best score giving model.

CONCLUSION

- Key Findings and Conclusions of the Study

  Conclusion- The used car price for some has degraded because of the number years it is used for and there some other factor that affect adversely.

  The pricing, Kilo Meters driven, condition of the car, Location etc. The is also one the factor which play important role in predicting the price of the used cars. Because pricing the car varies from state to state and taxation percentage also varies. Which also need to hike in the price or degrade the price accordingly as per state policy for Automobiles Companies.

After overcoming the skewness from the columns of Kilo Meters and Price model started predicting great result.

- ## Learning Outcomes of the Study in respect of Data Science

  This dataset is categorical in nature, we can verify data by using read method & get stats related information for each column using describe method.

  It was Label Encoded for the model learning because of the target was in continuous in nature. Regression model best suits for this.

  Visualize the data using univariant / multi-variant analysis.

  Check the prediction score using accuracy score & get ROC-AUC score.

  Train data using Regression models to get the best score & finalise best score giver model for this dataset.

  Get the test score for same model.

  Save file using pickle/joblib library.

  Find the prediction vs actual using distribution plot in order to get the perfect deviation if any.