*BMW USED CAR PRICE PREDICTION*

|  |  |  |
| --- | --- | --- |
| G.SRI HARSHA | HADHVIKA BODEPUDI | BVNSV RAHUL |
| School of Engineering and Applied Sciences University at Buffalo, The State University of New York New York, USA | School of Engineering and Applied Sciences University at Buffalo, The State University of New York New York, USA | School of Engineering and Applied Sciences University at Buffalo, The State University of New York New York, USA |

# Introduction

The prices of new cars in the industry are fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, there is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Even though there are websites that offers this service, their prediction method may not be the best. Besides, different models and systems may contribute on predicting power for a used car’s actual market value. It is important to know their actual market value while both buying and selling.

# DATA DESCRIPTION

The dataset is available in public and can be downloaded through <https://www.kaggle.com/mysarahmadbhat/bmw-used-car-listing?select=bmw.csv> . The dataset would return the following features like: Model, Year, Price, Transmission, Mileage, Fuel, Tax, MPG and Engine. We use this data for the price prediction using various regression techiques.

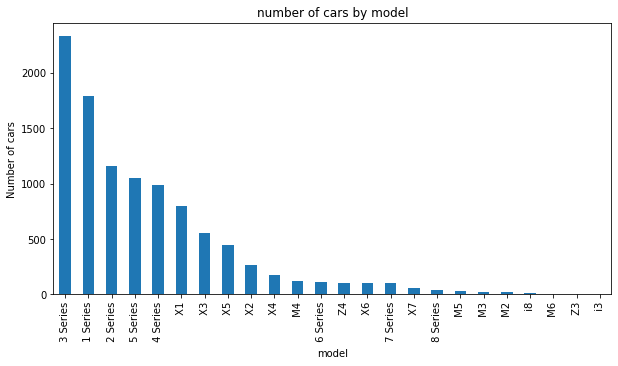
# AIM

To predict and build a linear model for calculating the best price for the used BMW car’s which the customer wants to buy.

# Exploratory Data Analysis

to visualize the data, we use seaborn and matplot libraries. Also, we use other pandas libraries to handle the dataframe such as Numpy, Sklearn. There are no null values present in the dataset but there are some outliers. So we need to remove the outliers using log transformation to get the accuracy of the prediction. Also, the values of tax and engine size is zero, so we change those value to null and make them zero. We removed the duplicate values present in the data.

After Removing the null values and duplicate values we now get into Data visualization. Now we plot a histogram between model and the number of cars in that model.



Now that the histogram is plotted, we also know that the price of any car is based on the number of miles that the car has travelled so we plot a scatter plot between miles and price.

Chart, scatter chart

Description automatically generated

From the above scatter plot we can observe that there is no linear relationship between miles and price which are the independent and dependent variable’s in our data set.

Now to find the relation between the dependent variable and independent variables we have to plot a “Heat map”.

Heat map generally gives us the correlation between the dependent variable and independent variables. Correlation is nothing but a value that describes how strongly the dependent variable and independent variable are correlated.

Correlation varies from 0-1, where 1 being they are highly correlated and 0 being week correlation between the variables. The heat map and the correlation table for our data set is as follows:

Table

Description automatically generated

A picture containing background pattern

Description automatically generated

Now that we have obtained and performed Exploratory data analysis, we can now start building our model.

# Test train Split

Now that the Data has been prepressed and scaled properly, we can build our required model. Before we do that, we have to split the data set into test and train datasets with desired test size. We split the data into X\_test, X\_train,y\_test,y\_train with the test size of 20% and train size as 80% of the data.

# Linear regression

Linear Regression tries to establish a linear relationship between the independent variables and the dependent variables. We also know that machine learning algorithms take only numeric values so we have to change the price column to numeric value as it is a “VARCHAR”. Now By importing SKlearn.LinearRegression. We fit the train data and predict with the help of test data set and obtain accuracy scores. The accuracy scores for linear regression are as follows:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

As it is a regression model, accuracy scores is not the correct metric to define weather our model is doing good or not so we take mean square error which says the error factor in our model and for linear regression we get an mean square error of “0.19”.

# Random Forest regressor

As we got a low accuracy score in linear regression but a well mean square error which resembles that our model is running well . So, we tried with the random forest regressor. The accuracy score have increased to 47% with mean square error value as 0.15.

# Graphical user interface, text, application, email Description automatically generated

# Support Vector Regression

# With the same train and test data we performed a SVR model with epsilon value as 0.2 . After performing it we got an accuracy score of 45% and mean square error value as 0.16.

Graphical user interface, text, application, email

Description automatically generated

# References

[1].SK-Learn:- <https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm> .

[2].Linear regression:-

🡪 <https://www.gs.washington.edu/academics/courses/akey/56008/lecture/lecture9.pdf>

🡪 <https://www.scribbr.com/statistics/simple-linear-regression/>

[3].Random Forrest:-

<https://corporatefinanceinstitute.com/resources/knowledge/other/random-forest/>

[4]. Support Vector Regression:-

<https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2>