**CAR DEKHO**

**PRICE PREDICTION**

**PROJECT USING PYTHON**

**Submitted by**

VenkataSubramanian Sriganesh

MD112

**Introduction:**

The project uses the Linear prediction models to predict the price of the used cars based on the car details which we extracted from the CarDekho website.

CarDekho.com is India's leading car search venture that helps users buy cars that are right for them. Its website and app carry rich automotive content such as expert reviews, detailed specs and prices, comparisons as well as videos and pictures of all car brands and models available in India.

**Objectives:**

Our objective is to understand the data and pre-process them to make it ready for building a prediction model.

We will be using the Data feature engineering and Linear Regression algorithms to select the dependent features and predict the car price based on the features.

**Data Set Description:**

The datasets are provided to us based on the cities.

The datasets contain the json data for the car details.

Features:

1. New Car Detail:
   1. it (integer): Ignition type.
   2. ft (string): Fuel type (e.g., Petrol).
   3. bt (string): Body type (e.g., Hatchback).
   4. km (string): Kilometers driven.
   5. transmission (string): Transmission type (e.g., Manual).
   6. ownerNo (integer): Number of previous owners.
   7. owner (string): Ownership details.
   8. oem (string): Original Equipment Manufacturer (e.g., Maruti).
   9. model (string): Car model (e.g., Maruti Celerio).
   10. modelYear (integer): Year of car manufacture.
   11. centralVariantId (integer): Central variant ID.
   12. variantName (string): Variant name.
   13. price (string): Price of the used car.
2. New Car Overview:
   1. heading (string): Car overview heading.
   2. top (list of dictionaries): Top details, including keys like registration year, insurance validity, fuel type, etc.
   3. bottomData (None): Additional bottom data (currently not available).
3. New Car Feature:
   1. heading (string): Features heading.
   2. top (list of strings): Top features.
   3. data (list of dictionaries): Detailed feature information categorized by comfort, interior, exterior, safety, etc.
4. New Car Specs:
   1. heading (string): Specifications heading.
   2. top (list of dictionaries): Top specifications like mileage, engine, max power, torque, etc.
   3. data (list of dictionaries): Detailed engine and transmission information, dimensions, capacity, and miscellaneous details.

**Methodology:**

1. **Data Loading:**
   1. We will be using the below libraries:
      1. Pandas
      2. Numpy
      3. Json
      4. Ast
   2. Combining the dataset based on the cities to one dataset named **'cars\_details.csv'**
   3. With the help **json\_columns** and **flatten\_json** functions, we are converting the json data set to csv type dataset with columns separated.
   4. Save the columns separated dataset as **cols\_cars\_details.csv**
2. **Data Analysis:**
   1. For numerical data:
      1. We will be using **Mean Strategy Imputing** method to fill the missing values.
      2. **StandardScaler** method is being used to perform the scaling operation
      3. The outliers are removed by the function **remove\_outliers** created
   2. For Categorical data:
      1. The missing values in the categorical data are being replaced by the **Most Frequent value** of the columns.
      2. The data is then encoded using the **OneHotEncoding** Method to get better understanding.
      3. The reason why **LabelEncoding** is not used is because the encoded values might vary the prediction values and provide the least r2 score.
   3. Data Visualization:
      1. Libraries used are:
         1. Pandas
         2. Matplotlib
         3. Seaborn
      2. We visualized the data for the numerical columns and the outliers are found from the box plots from the **data\_visualization.ipynb** file
3. **Feature Engineering:**
   1. The most correlated dependent features are dropped using the below code block. This reduces the features to be considered for the model development:

corr\_matrix = X.corr().abs()

upper = corr\_matrix.where(np.triu(np.ones(corr\_matrix.shape), k=1).astype(bool))

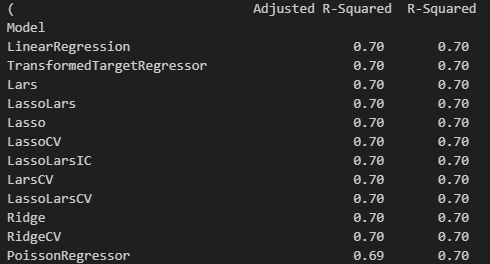
to\_drop = [column for column in upper.columns if any(upper[column] > 0.95)]

X = X.drop(to\_drop, axis=1)

* 1. For selecting the features which are having highest dependency for price prediction, we are using **SelectKBest** to select k number of the features for prediction. The columns which are being used are:

Engine Displacement, Torque, Wheel Size, No of Cylinder, Cruise Control.1, Engine Start Stop Button, Steering Wheel Gearshift Paddles, transmission\_Manual, oem\_Mercedes-Benz, model\_Mercedes-Benz AMG G 63, model\_Toyota Land Cruiser 300, km

1. **Model Development:**
   1. Linear Regression algorithms:
      1. **LazyRegressor()** method is being used to perform prediction and provide the evaluation metrics for multiple algorithms



* + 1. Model Selection:

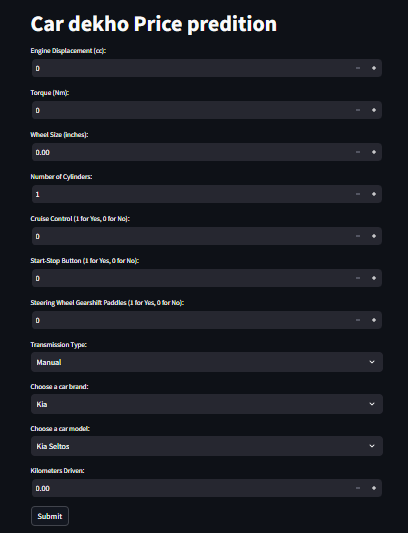
With the above results, we can see that the **LinearRegression** model has highest r2\_score compared to others. Hence we are training the Linear Regression model and saving it as **best\_prediction\_model.pkl** to load it and use it for the future use.

1. **Streamlit application:**

Libraries used are Pandas, Streamlit and pickle

The streamlit application is used to develop interactive front end for the python application created.

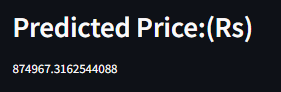
We will be getting the users input as below screenshot:



Once the users input are provided to the streamlit application, the data is then converted to a dataframe X in back end.

Then using the pickle the **best\_prediction\_model.pkl,**  the model is loaded and used for predicting the price of the car.

The sample output is as below:



1. **Conclusion:**

The streamlit application for the Car dekho price prediction is developed with R2 score of 0.7 and the value is being adjusted to provide more accurate value.