CAR PRICE PREDICTION

A Project Report submitted in partial fulfillment of

the requirement of under graduate Degree

Bachelor of Technology In

Computer Science Engineering

By

T N B K YESASVI	AP18110010199
V JAYANTH	AP18110010226
N GEETA KIRANMAI	AP18110010237
SHUBHAM VYAS	AP18110010242
T PAVAN KUMAR	AP18110010244

Department of Computer Science Engineering

SRM University, AP Amaravathi-522502

2020-2021

AIM:

To predict the price of a Car by taking its Company name, its Model name, Year of Purchase, and other parameters (like fuel type and number of kilometers that the car has been travelled).

ABSTRACT:

A car price prediction has been a high-interest research area, as it requires perceptible exertion and information. Significant numbers of different parameters are examined for reliable and exact prediction. To construct a model for predicting the price of used cars we apply the machine learning technique. Respective performances of different algorithms were then compared to find one that best suits the available data set. As we used the linear regression technique to built the model.

INTRODUCTION:

The car price predictor helps to predict the used car price based on the parameters/attributes. As it helps the customer to know whether the price is worth or not to purchase the used car not only to purchase the car but also people can use the website to check the price to sell they car to the valid price by giving the details of their car so that they can get to know the best price to sell the car. The car price predictor is done by using linear regression technique. As linear regression is a machine learning algorithm based on supervised learning. Supervised learning consists of a target/outcome variable which is used to be predicted from a given set of predictors. Using these set of variables, we generate a function that map the input to desired outputs. The training process continuous until the model achieves a desired level of accuracy on the training data.

Linear regression: It performs a regression task. A models target prediction value based on independent variables.

Problem survey:

According to a survey we get to know that now a day's everyone are interested in using cars but many of the people cannot effort to get a new car because the manufacturing cost of the new car is high and also some additional charges are added by the government in the form of taxes so, many customer may not be able to effort the price of a new car yet there is an opportunity to buy the same car with less cost by buying a used car. But they are many people who are selling the used car for a high price which that cannot be paid more for pre-owned. There is a need for car price prediction system to effectively determine the worthiness of the car using a variety of features. To get effective prediction we used machine learning model that linear regression technique.

DATASET:

The data set is scraped from kaggle.com by Balaka Biswas. Where she gathered the data from many web resource .The dataset is scraped on April 2020.So the data Is the fresh the data consist of 6 columns and 892 rows.

```
In [3]: car=pd.read_csv('quikr_car.csv')

In [4]: car.head()
```

Out[4]:

	name	company	year	Price	kms_driven	fuel_type
0	Hyundai Santro Xing XO eRLX Euro III	Hyundai	2007	80,000	45,000 kms	Petrol
1	Mahindra Jeep CL550 MDI	Mahindra	2006	4,25,000	40 kms	Diesel
2	Maruti Suzuki Alto 800 Vxi	Maruti	2018	Ask For Price	22,000 kms	Petrol
3	Hyundai Grand i10 Magna 1.2 Kappa VTVT	Hyundai	2014	3,25,000	28,000 kms	Petrol
4	Ford EcoSport Titanium 1.5L TDCi	Ford	2014	5,75,000	36,000 kms	Diesel

```
In [5]: car.shape
Out[5]: (892, 6)
```

The columns in the data set are:

NAME: The Model of the car.

COMPANY: The Company of the car.

YEAR: The year when the car model is released

PRICE: The price that the seller is going to sell.

KMS_DRIVEN: Total number of kilometers that the car has been travelled.

FUEL_TYPE: The car's fuel type.

Preprocessing:

Data preprocessing is an important step in the data mining process which is used to transform raw data in a useful and efficient format, so that we can use the data for further processes. The dataset we scrapped to be preprocessed as they are few null values in the kms_driven and fuel_type and the year should be in the integer type but in the scrapped data it in object type and has many non integer values not only the year but also the price is in the object type.

```
In [6]: car.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Data columns (total 6 columns):
 #  Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0  name        892 non-null    object
 1  company     892 non-null    object
 2  year        892 non-null    object
 3  Price       892 non-null    object
 4  kms_driven  840 non-null    object
 5  fuel_type   837 non-null    object
dtypes: object(6)
memory usage: 41.9+ KB
```

As if do not preprocess the data then we will have many errors in the output as we cannot get the accurate values. To avoid the errors we have to clean up the data and then we have to built a model of it.

After cleaning the data: The rows are reduced to 816

```
816 rows × 6 columns
```

```
In [20]: car.to_csv('Cleaned_Car_data.csv')
```

```
In [21]: car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   name        816 non-null    object
 1   company     816 non-null    object
 2   year        816 non-null    int32 
 3   Price       816 non-null    int32 
 4   kms_driven  816 non-null    int32 
 5   fuel_type   816 non-null    object
dtypes: int32(3), object(3)
memory usage: 28.8+ KB
```

IMPLEMENTATION:

As our project is based on linear regression First step is to extract features and labels. In the given data set everything is consider as feature except the price column. So we have to drop the price column and the price column set as target .

Extracting Training Data

```
In [32]: X=car[['name','company','year','kms_driven','fuel_type']]
         y=car['Price']
```

The regression problem is measured using the r2_score and also onehotencoder. As the r2_score, linear regression and onehotencoder[1] are imported from sklearn. As once the data is fit into the object of onehotencoder we have to transfer all the x_train and x_test using onehotencoder as it is bit difficult we are going to use sklearn column transformer[2] and pipeline.

Applying Train Test Split

```
In [35]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

```
In [74]: from sklearn.linear_model import LinearRegression
```

```
In [75]: from sklearn.preprocessing import OneHotEncoder
         from sklearn.compose import make_column_transformer
         from sklearn.pipeline import make_pipeline
         from sklearn.metrics import r2_score
```

Creating an OneHotEncoder object to contain all the possible categories

```
In [39]: ohe=OneHotEncoder()
         ohe.fit(X[['name','company','fuel_type']])
```

```
Out[39]: OneHotEncoder()
```

Creating a column transformer to transform categorical columns

```
In [52]: column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categories_),['name','company','fuel_type']),
                                             remainder='passthrough')
```

Fitting the data into the pipeline this helps to transfer the raw date from one end and we will get all the prediction from the other end. And after dumping we can use it in the web page even without using onehotencoder.

Linear Regression Model

```
In [54]: lr=LinearRegression()
```

Making a pipeline

```
In [55]: pipe=make_pipeline(column_trans,lr)
```

Fitting the model

```
In [59]: pipe.fit(X_train,y_train)
```

```
Out[59]: Pipeline(steps=[('columntransformer',
                          ColumnTransformer(remainder='passthrough',
                                            transformers=[('onehotencoder',
                                                           OneHotEncoder(categories=[array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi A4 2.0', 'Audi A6 2.0',
       'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Series',
       'BMW 5 Series', 'BMW 7 Series', 'BMW X1', 'BMW X1 sDrive20d',
       'BMW X1 xDrive20d', 'Chevrolet Beat', 'Chevrolet Beat...
                                                                                  array(['Audi', 'BMW', 'Chevrolet', 'Datsun', 'Fiat', 'Force', 'Ford',
       'Hindustan', 'Honda', 'Hyundai', 'Jaguar', 'Jeep', 'Land',
       'Mahindra', 'Maruti', 'Mercedes', 'Mini', 'Mitsubishi', 'Nissan',
       'Renault', 'Skoda', 'Tata', 'Toyota', 'Volkswagen', 'Volvo'],
      dtype=object),
                                                                                  array(['Diesel', 'LPG', 'Petrol'], dtype=object)]),
                                                                     ['name', 'company',
                                                                      'fuel_type'])])),
                         ('linearregression', LinearRegression())])
```

For predicting we are using r2_score As the data set is too small that's why the different train test splits are resulting different values of r2_score. So we are training the data using the random state.

The best model is found at a certain random state

```
In [67]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=np.argmax(scores))
         lr=LinearRegression()
         pipe=make_pipeline(column_trans,lr)
         pipe.fit(X_train,y_train)
         y_pred=pipe.predict(X_test)
         r2_score(y_test,y_pred)
```

```
Out[67]: 0.920088412025344
```

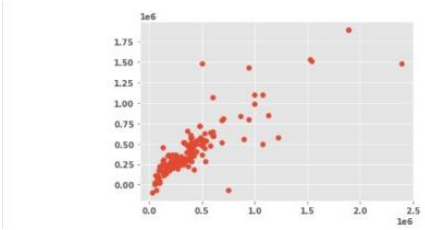
Now we are going the dumped the pipeline using pickle.

```
In [68]: import pickle
```

```
In [69]: pickle.dump(pipe,open('LinearRegressionModel.pkl','wb'))
```

The Website for the car price prediction used the previous data. The cleaned data is used and the pickle file extracted the model is used. For the application we attached the html file to the application file and CSS file is attached to the html. From the cleaned car file we read the values with unique categories and before sending the date or predicting the data we make a predict function and passed few arguments and by passing the argument request. Loading the linear regression model we created an object and loaded the pickle file in to it.

Resultant graph: The graph for the price and the prediction.



Conclusion:

We used a linear regression model to predict the car price and the training data used few python techniques. Its purposes was to predict the prices of used cars by using a dataset. The set is analyzed with 5 predictors and with many observations. With the help of the data visualizations and exploratory data analysis, the dataset was explored deeply. Concluding that the linear regression model gave the best prediction values for predicting the used car price

Reference:

1. https://www.kaggle.com/balaka18/quikr-cars-scraped/version/1
2. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
3. https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html