INTRODUCTION

ABOUT DATASET

This dataset has been created by collecting data from 'cardekho.com' through Web-scrapping

The shape of this dataset is rows = 5300, variables = 8

we have taken many types of cars in this dataset like Sedan, SUV, MUV, Hatchback

. In this dataset has 8 variables

1. car_brand= car company name

2.fuel= which types of fuel does the car run on

3. type= car engine is automatic or manual type

4. kilometer= how much distanced covered by car

5. year= in which year car was bought

6.body = which types of cars like SUV, MUV Sedan Hatchback

7.Model = car's model name

8.price= car price, THIS IS TARGET VARIABLE

BUSINESS PROBLEM FRAMING

. Today, about 15 years ago there was no company that buy and sells used cars , but in a few years many such companies have come in the market that buy and sells used car. This Dataset is related to this

. Nowadays many people like to buy used car but while buying used car, maximum people want to see that the car is in good condition and not too old, there are many such reasons. some people keep their income in mind or save buy a used car

. There is to build a ML Regression Model on this Dataset so that we can easily know the price of car and for this we have to collecting mor and more data

because the more data there is, the more there will be a possibility to predict the price accurately

CONCEPTUAL BACKGROUND OF THIS DOMAIN

. This Dataset is related to car and if we have a good understanding about car or in this, then it helps us in creating a good data

. The Important point in this dataset is that on what basis the price of the used vehicles should be fixed and such problem can be solved only when there is more clear knowledge in this domain

. we have to make this Dataset such way that car price can be predicted very accurately and for this we should be more knowledge of all factors about car that price depends more on which type of factor

ANALYTICAL PROBLEM==

ANALYTICAL MODELING OF THE PROBLEM

. In this Dataset, the price and distanced travelled variables of used car apart from these all the variables were categorical and when I predicted the price by making ML model by one-hot-encoding the categorical variables, then the r2_score came to minus of Linear-Regression model

After that we more considered on the EDA part and we changed the values of categorical variables according to EDA along with car-price then we got some r2_score

DATA-PREPROCESSING

. In this we had to do some data cleaning

. In this process we have nothing to do because the dataset has not more variables all categorical type except one variable

. there were some categorical variables which I had label-encoding because after doing one-hot-encoding was coming very less r2_score

. There was nothing like Skewness, Outliers and Multicollinearity problem in this dataset

. Last process done on this dataset were Standard-Scaling

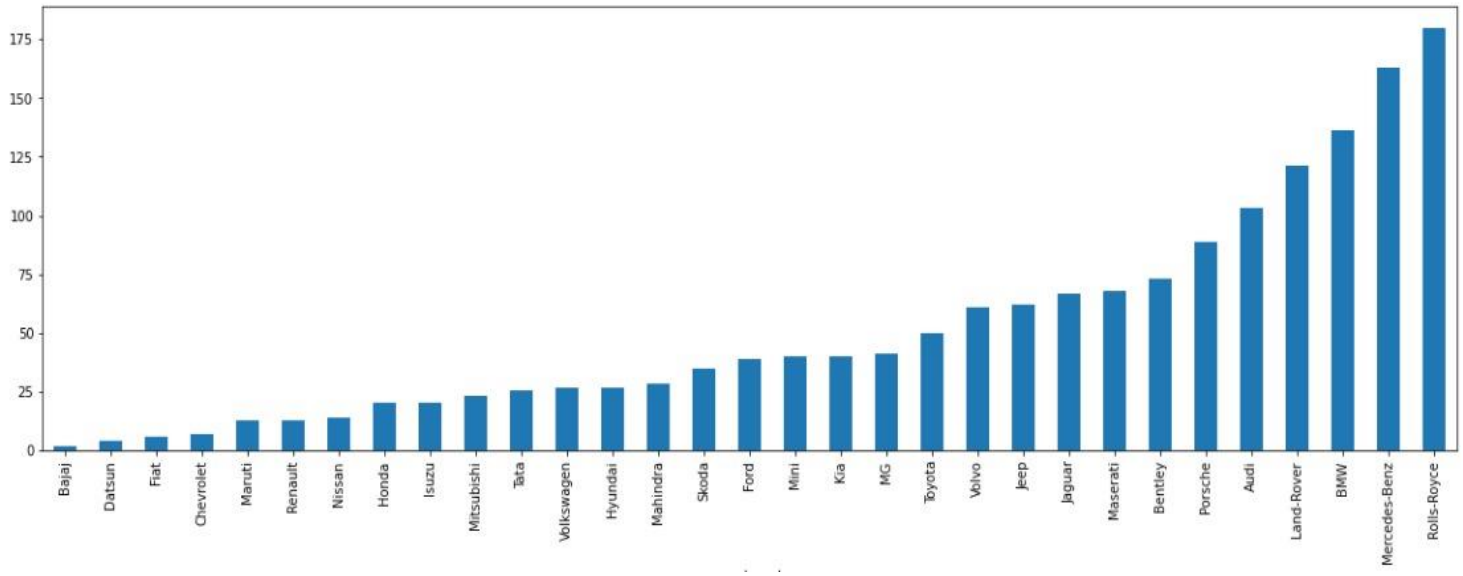
EDA on the Dataset

. I visualize the data using Matplotlib & seaborn

. All input-variable are randomly distributed with Target and no any relation between input-data and Target

. plotting car_brand' variable according to price, from which car_brand has highest price

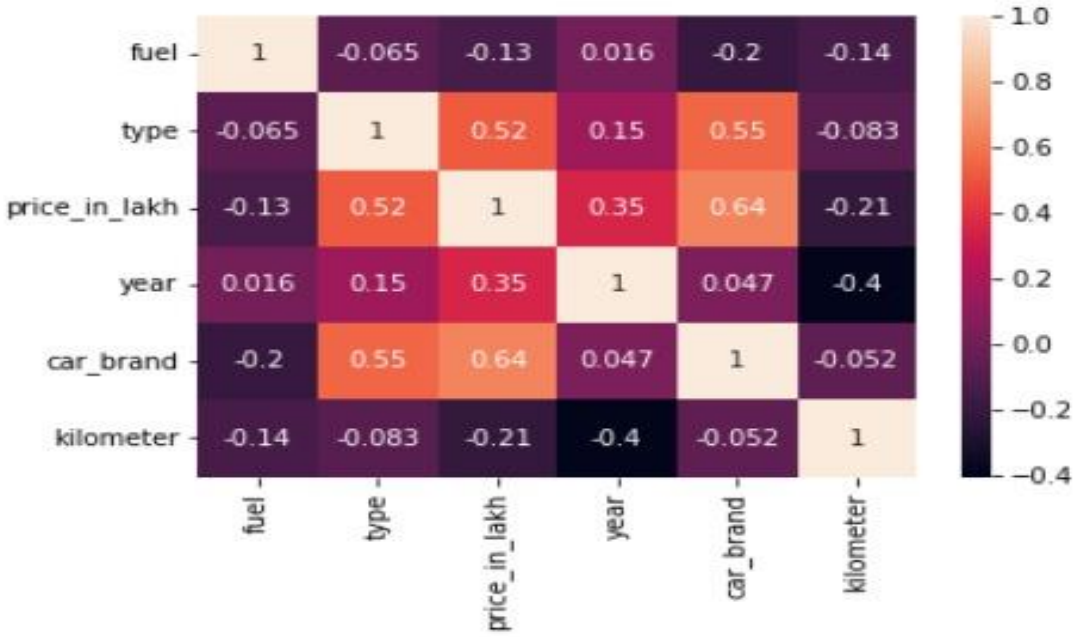
. The name of the car has been changed in she way that as the price of car is increasing the value of the name of car is also increasing, it will be easy to predict the price



. According this plot we will have to change the car_brand value for formed a relationship between Target & car_brand

. Like that we have change some categorical variables

. Analyzing the correlation then we saw that three variable are positive correlated with target and car_brand was more positive correlated with target(price)



HARDWARE AND SOFTWARE REQUIREMENT TOOLS

. we have done this entire project through (sklearn, pandas numpy, matplotlib, Seaborn)

. Load the dataset, and all data cleaning & analyzing part done by Pandas

. we have done Visualization the dataset through matplotlib, & seaborn

. All Machine Learning model have been created in this project through Sklearn library

MODEL DEVELOPMENT & EVALUATION

IDENTIFICATION OF PROBLEM-SOLVING APPROACHES

. After doing data-preprocessing and EDA , we first prepared the model by on-hot-encoding the categorical variable, then saw that the accuracy of the model was very low that of Linear-Regression, SVR

Then again, we used Label-encoding on the categorical variable and created the model on this then we got Linear Regression =57 and SVR = 70 percent r2_score

. All input-data randomly distributed with Target and very difficult to predict target in Regression problem without any correlation with target

. After plotting the dataset several times, I came to know that how to change the value of Categorical-variable

so that a positive relation can be formed with the Target(price) so that the target is predicted more accurately.

. and last, we select (car-brand & type) variable for changing the values of this variables manually, didn't used Label-Encoding

. If we add one or two variable more in this dataset than most we can get more r2_accuracy

. If the car 'model' variable also changed to the car-price, then there was some possibility and more r2_score would come

TRAING & TESTING OF IDENTIFIED APPROACHES

. First, we do Train_test_split on input-data & target-attribute then data split into train & test data, fit the model on train data then test the model accuracy on test data

. we used different types of algorithms like Linear-Regression, KNN Regressor, RandomForest-Regressor, for making a model.

. There is trained the model with these all algorithms for have best r2-score

. we got the highest r2_score & minimum mean_squared_error on this Dataset from RandomForest Model

. I used different types of metrics to find which one model is perform better on this dataset

. r2_score, mean_squared_error, mean_absolute_error and root_mean_squared_error we used all these metrics

. we used cross validation with KFold on all five model, cross_validation_score of RandomForest model is highest compare to all models

. We done Hyperparameter tuning over RandomForest_Model because 'r2_score' & cross_validation_score of

RUN & EVALUATE SELECTED MODELS

. Linear-Regression model = 57 percent r2_score, The score for this model is so low because the input-data was very little positively correlated with Target

. We showing the model

```
:
##                                              LINEAR REGRESSION
from sklearn.linear_model import LinearRegression
x_train,x_test,y_train,y_test=train_test_split(scaled1, y2,random_state = 59,test_siz

LR = LinearRegression()
LR.fit(x_train, y_train)
LR_pred= LR.predict(x_test)
print('r2_score=',r2_score(y_test, LR_pred))
print('mean_squared_erro=', mean_squared_error(y_test, LR_pred))
print('mean_absolute_error=', mean_absolute_error(y_test, LR_pred))
print('root_mean_squared_erro=', np.sqrt(mean_squared_error(y_test, LR_pred)))
```

```
r2_score= 0.5770635878066761
mean_squared_erro= 77.3071982541808
mean_absolute_error= 5.390186208620437
root_mean_squared_erro= 8.792451208518635
```

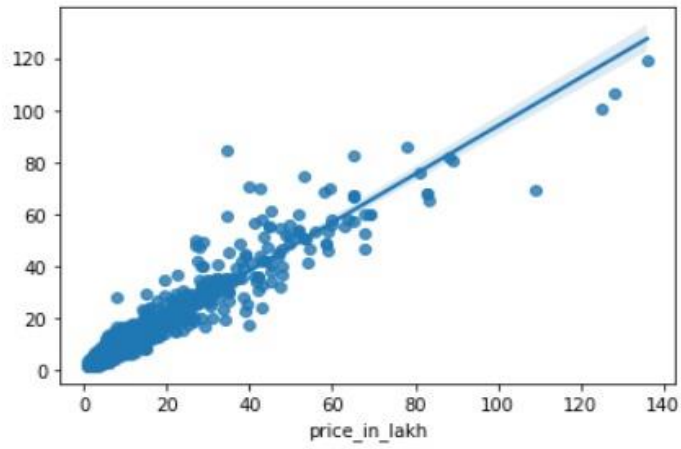
. We showing the RandomForest model result

```
:
##                                                RANDOMFOREST REGRESSOR
from sklearn.ensemble import RandomForestRegressor
x_train,x_test,y_train,y_test=train_test_split(scaled1, y2,random_state = 59,test_size=0.

RF = RandomForestRegressor()
RF.fit(x_train, y_train)
RF_pred= RF.predict(x_test)
print('r2_score=',r2_score(y_test, RF_pred))
print('mean_squared_erro=', mean_squared_error(y_test, RF_pred))
print('mean_absolute_error=', mean_absolute_error(y_test, RF_pred))
print('root_mean_squared_erro=', np.sqrt(mean_squared_error(y_test, RF_pred)))
```

```
r2_score= 0.9145610754153911
mean_squared_erro= 15.617108603236561
mean_absolute_error= 1.884420336176363
root_mean_squared_erro= 3.951848757636932
```

```
:
## PLOTTING REGPLOT  OF ( y_test and predicted_value ) with RandomForest model

sns.regplot(y_test, RF.predict(x_test))
```

```
: <AxesSubplot:xlabel='price_in_lakh'>
```



. We using cross validation for evaluating the model

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
kfold= KFold(n_splits=12, shuffle=True, random_state=59)
```

```
## CROSS_VALIDATION WITH  RANDOMFOREST
(cross_val_score(RF, x_train,y_train, cv=kfold)).mean()
```

```
0.8351589435684749
```

```
## CROSS_VALIDATION WITH  KNN
(cross_val_score(knn, x_train,y_train, cv=kfold)).mean()
```

```
0.7484786199784838
```

```
## CROSS_VALIDATION WITH  SVR
(cross_val_score(svm, x_train,y_train, cv=kfold)).mean()
```

```
0.6438795840367649
```