IC SOLUTIONS & TAKE IT EASY ENGINEERS

INTERNSHIP PROJECT REPORT ON

"AUTOMOBILE PRICE PREDICTION"

BY

NAME:P KISHORE

USN:1KS17CS051

EMAIL:kishorep.shrivatsa@gmail.com

TUTOR: ABHISHEK C (IC SOLUTIONS)

ACKNOWLEDGEMENT

The AUTOMOBILE PRICE PREDICTION PROJECT would be incomplete without the mention of the people who made it possible and whose constant guidance crowned my effort with success.

I would like to extend my gratitude to IC SOLUTIONS & TAKE IT EASY ENGINEERS, for providing all the facilities to learn and understand the fundamentals of MACHINE LEARNING USING PYTHON.

I would like to extend my gratitude to ABHISHEK C (tutor) for constant guidance and inputs to furnish the AUTOMOBILE PRICE PREDICTION PROJECT, and IC SOLUTIONS for giving me this wonderful opportunity to upskill myself.

Finally, I extend my heart-felt gratitude to my family for their encouragement and support without which I would not have come so far.

Moreover, I thank all my friends for their invaluable support for guiding me when I was stuck at few points.

ABSTRACT

The machine learning field, which can be briefly defined as enabling computers make successful predictions using past experiences, has exhibited an impressive development recently with the help of the rapid increase in the storage capacity and processing power of computers. Together with many other disciplines, machine learning methods have been widely employed in bioinformatics.

In this project, we first make use of concepts of machine learning to predict the cost of an automobile based on given features. We make use of mathematical modules like NumPy for calculations, pandas for reading and analysing dataset, matplotlib and other libraries to visualize the relation between columns of dataset.

The project's goal is to predict the prices of automobiles and to return the accuracy scores of the algorithms used to perform this analyzing which method is most suitable for the given dataset.

INDEX: Pg.No

1.INTRODUCTION

The goal is to predict the price of a vehicle based on given set of features in given dataset. This plays a vital role in Automobile industry and business purposes. To achieve this goal, Artificial intelligence/ Machine learning regression algorithms are used. Machine learning algorithms take in labelled/normalized data, but in a realworld example, dataset will not be organized in most of the cases. Hence, we first normalize the given data and analyse with various plots to understand relation between each columns on another. Each model has its own pros and cons and, for a given dataset, each algorithm outperforms another, hence we can develop different models for same example and check which gives better accuracy (r2 score) and use that model. The model built, can be saved using a package called joblib and can be used in future in an application.

2.PROBLEM STATEMENT AND OBJECTIVE

- The model is given a dataset of automobile's features like make, body type and 24 more other features.
- Goal is to predict the price of an automobile based on given features.
- A ML model has to be built which can take in a dataset of previous records and predict new values based on that data.
- The data set may contain a lot of data which is incompatible or data which might not be helpful for training model.
- Data has to be normalized and analysed for better understanding of relation between data.
- Different models are built and the one which offers best r2 score is selected.

3. REQUIREMENT SPECIFICATION

SOFTWARE REQUIREMENTS:

OS - Windows, Mac, Ubuntu (Python 3 Kernel)

IDE - Jupyter-notebook (Anaconda distribution)

Libraries - NumPy, seaborn, pandas, sklearn, etc

HARDWARE REQUIREMENTS:

RAM – 2GB

PROCESSOR – 1 GHz

Hard disk – 1GB

ABOUT THE COMPANIES

ICS is a digital service provider that aims to provide software, designing and marketing solutions to individuals and businesses. At ICS, we believe that service and quality is the key to success.

We provide all kinds of technological and designing solutions from Billing Software to Web Designs or any custom demand that you may have. Experience the service like none other!

Some of our services include:

Development - We develop responsive, functional and super-fast websites. We keep User Experience in mind while creating websites. A website should load quickly and should be accessible even on a small view-port and slow internet connection.

Mobile Application - We offer a wide range of professional Android, iOS & Hybrid app development services for our global clients, from a start-up to a large enterprise.

Design - We offer professional Graphic design, Brochure design & Logo design. We are experts in crafting visual content to convey the right message to the customers.

Consultancy - We are here to provide you with expert advice on your design and development requirement.

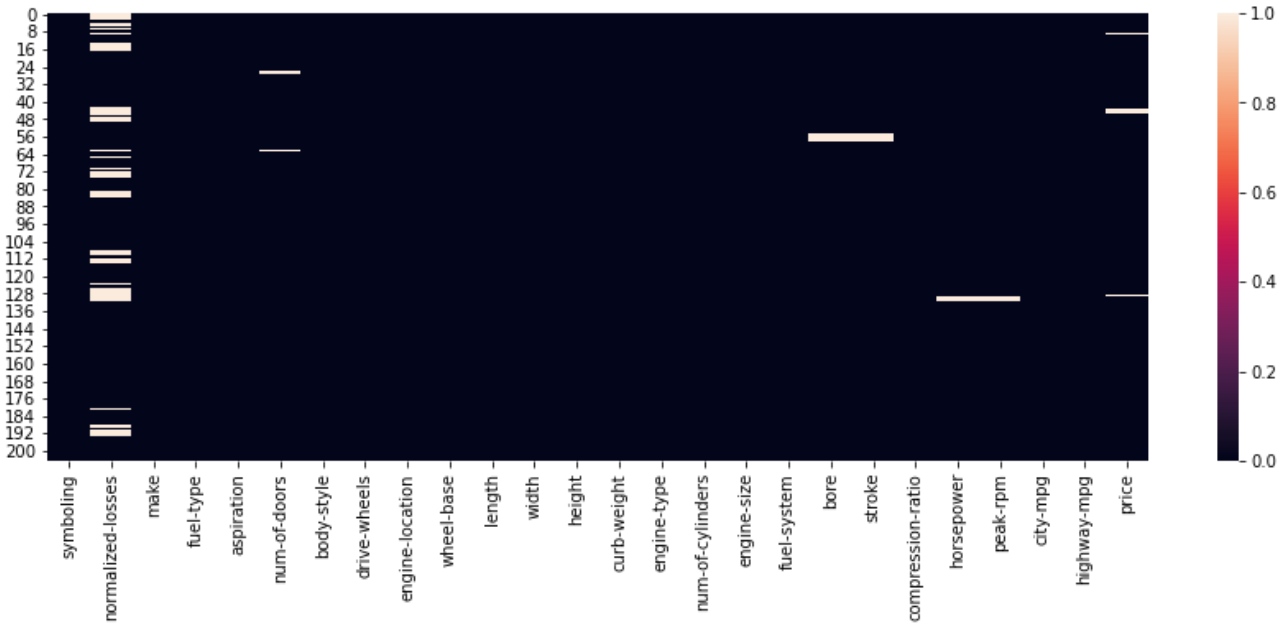
Videos - We create a polished professional video that impresses your audience.

4. EXPLORATORY DATA ANALYSIS

1. SEABORN HEATMAP

plt.figure(figsize = (15,5))

sns.heatmap(ad.isnull(), cmap='rocket')

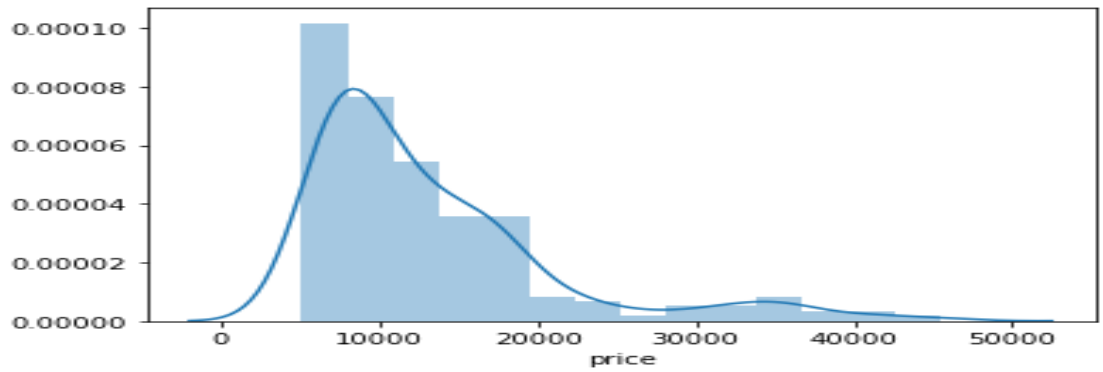


OBSERVATION:

The white stripes in the plot depicts the null values in respective column which we are going to be normalize with mean of that column.

2.DISTPLOT

sns.distplot(ad['price'])

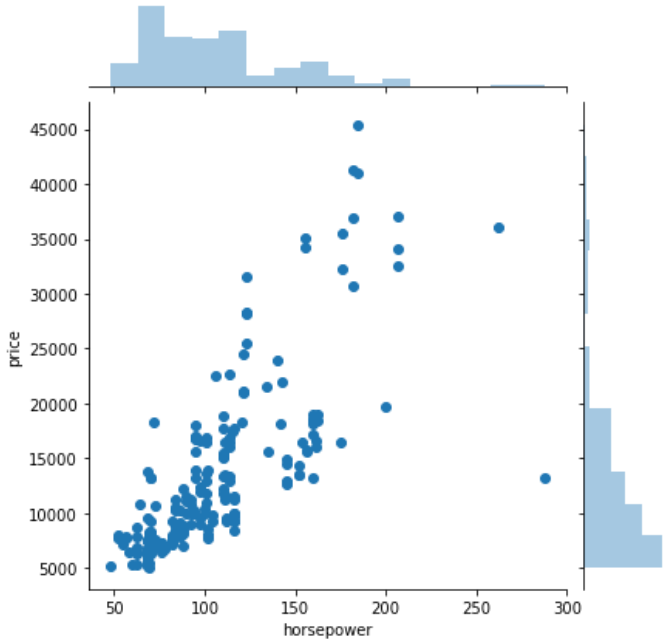


OBSERVATION:

Data points follow a roughly straight-line trend, the variables have an approximately linear relationship.

3.JOINT PLOT

sns.jointplot(x='horsepower',y='price',data=ad)



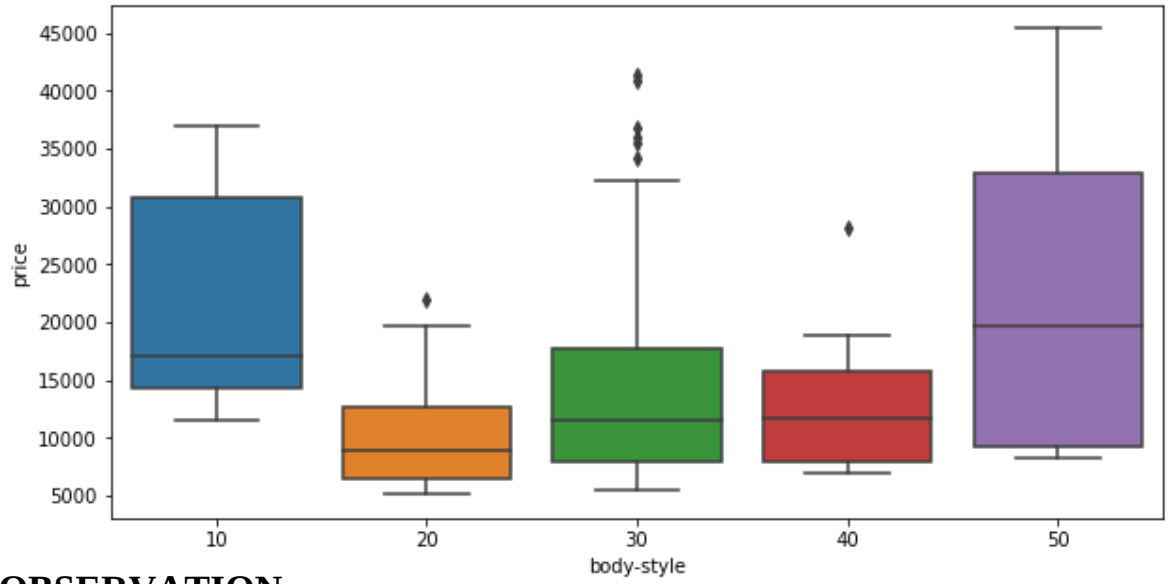
OBSERVATION:

Above plot depicts distribution of horsepower vs price, and price is concentrated more in the range of 5k-20k which approximately has horsepower in range of 50-150hp.

4.BOX PLOT

plt.figure(figsize=(10,5))

sns.boxplot(x='body-style',y='price',data=ad)



OBSERVATION:

10-Convertible, 20-Hatchback, 30-Sedan, 40-Wagon, 50-Hardtop.

Above plot tells us many things:

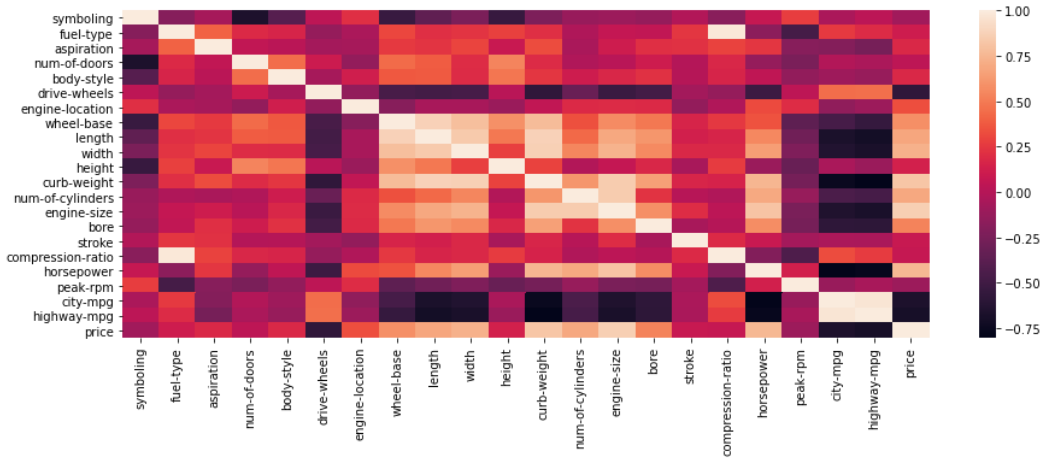
1.Hatchback cars are the least expensive ranging from 7.5k-13.5k and hardtops are the most expensive ranging from 10k-33.5k.

2.Most people prefer hardtops first, convertible next and the others the least.

5.HEAT MAP

plt.figure(figsize=(15,5))

sns.heatmap(ad.corr())

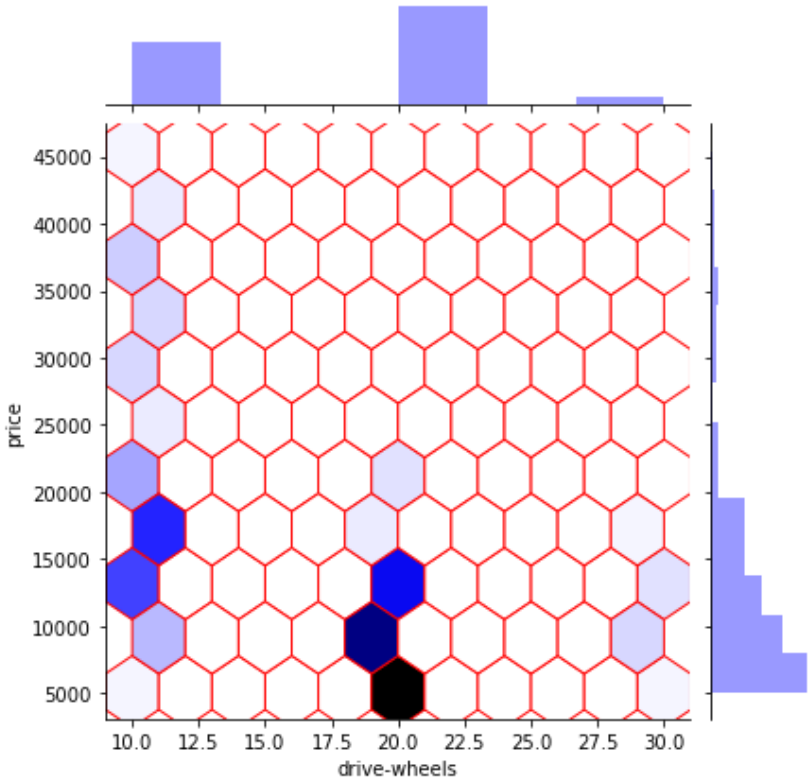


OBSERVATION:

Data points follow a roughly straight-line trend, the variables have an approximately linear relationship.

6.JOINT PLOT

sns.jointplot(x='drive-wheels',y='price',kind='hex',data=ad,color='b',edgecolor='r',linewidth=1)



OBSERVATION:

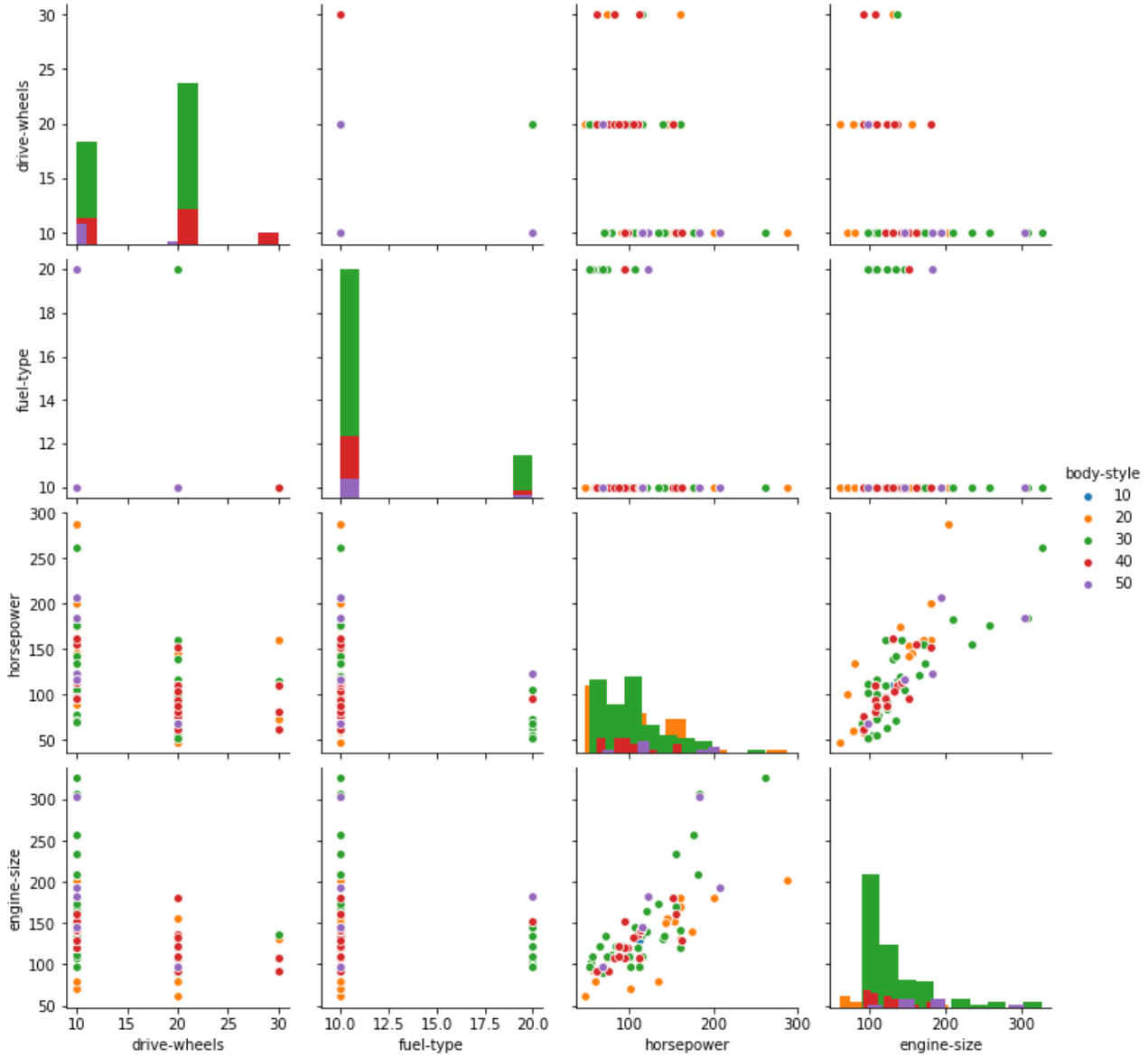
10-rwd, 20-fwd, 30-4wd.

Above plot tells us that rear-wheel drive cars are the most expensive ranging from 5k-45k(most cars in 13k-18k range), followed by front wheel drive ranging from 5k-25k(most cars in 5k-15k range) lastly 4wd cars ranging from 7k-18k.

7.PAIR PLOT

plt.figure(figsize=(20,20))

sns.pairplot(ad[['drive-wheels','body-style','fuel-type','horsepower','engine-size']],hue='body-style',diag_kind='hist')



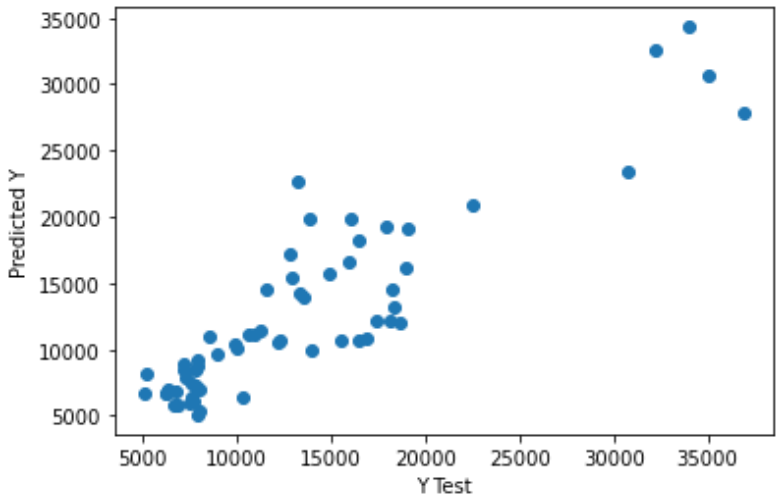
OBSERVATION:

Above plot tells us the relationship between the selected columns mentioned in the code. It helps us understand the spread of data w.r.t. above columns. (works on numeric columns only).

5. PREPARING MACHINE LEARNING MODEL

5.1. LINEAR REGRESSION

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
X = ad.drop(['make','engine-type', 'fuel-system','price'], axis=1)
y = ad['price']
lr = LinearRegression()
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)
lr.fit(X_train, y_train)
plt.scatter(y_test,y_pred) plt.xlabel('Y test)
plt.ylabel('Predicted Y')
```



Training multiple models to get best r2 score

```
best_r2 = 0
for i in range(44):
    X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)
.  lr.fit(X_train, y_train)
   y_pred = lr.predict(X_test)
   if best_r2 < r2_score(y_test, y_pred):
        best_r2 = r2_score(y_test, y_pred)
print(best_r2)
```

Best r2 score = 0.901393

5.2. SUPPORT VECTOR MACHINES

```
from sklearn.model_selection import train_test_split
from sklearn.externals import joblib from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV
X = ad.drop(['fuel-system','make','engine-type','price'], axis=1)
y = ad['price']
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)
lr.fit(X_train, y_train)
param_grid = {'C': [0.1,1, 10, 100, 1000], 'gamma': [1,0.1,0.01,0.001,0.0001],
'kernel': ['linear']}
grid = GridSearchCV(SVR(),param_grid,verbose=8)
grid.fit(X_train,y_train)

grid_pred = grid.predict(X_test)

r2_score(y_test, grid_pred)
```

Best r2 score = 0.111793

5.3. DECISION TREE

```
from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier() dtree.fit(X_train,y_train)

pred_dtree = dtree.predict(X_test)

r2_score(y_test, pred_dtree)
```

r2 score = 0.907748

This can be further improvised by using random forest.

5.4 RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(X_train,y_train)
pred_rf = rfc.predict(X_test)
r2_score(y_test, pred_rf)
```

r2 score = 0.827657

Training multiple models to get best r2 score.

```
bestr2 = 0
for i in range(22):
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)
rfc.fit(X_train, y_train)
pred_rf = rfc.predict(X_test)
  if bestr2 < r2_score(y_test, pred_rf):
      bestr2 = r2_score(y_test, pred_rf)
      print(best_rfc_r2)
```

Best r2 score = 0.934109

6. ML MODEL CHART

SL NO.	ALGORITHM	r2_score
1	Random Forest	0.9341
2	Decision Tree	0.9078
3	Linear Regression	0.9014

HURDLES:

As a beginner to ML and data science concepts, analysing the data and drawing conclusion from plots were challenging but things like data cleaning was the toughest part it needed a lot of understanding of built-in libraries and basics which I've implemented using functions. The other options like label encoding would be a better option which ill definitely implement in upcoming projects.

CONCLUSION:

Best r2_score was achieved with following algorithms:

1 – Random Forest (0.9341)

2 – Decision Tree (0.9078)

3 – Linear regression (0.9014)

BIBILOGRAPHY:

- https://medium.com/data-science-group-iitr/linear-regression-back-to-basics-e4819829d78b
- https://medium.com/python-in-plain-english/exploratory-data-analysis-in-python-50fd19912155
- https://medium.com/pursuitnotes/support-vector-regression-in-6-steps-with-python-c4569acd062d
- https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991
- https://www.kaggle.com/pratsiuk/valueerror-unknown-label-type-continuous.
- Resources provided by instructor(day4-day7).