ML-MAJOR-AUGUST
ML-08-SPB2

Submitted On:
5/10/2022

Submitted By:
Sukrith Sunil
Amrita School Of Engineering
Amritapuri

INRODUCTION

In this project we perform,perform EDA(Exploratory Data Analysis) and apply a
suitable Classifier,Regressor or Clusterer and calculate the accuracy of the model on a data set CARS.csv

FUNCTIONS

dataorg()

graphs()

acc_check()

CSV FILE

https://drive.google.com/file/d/1-AyVrZz6vJtlq2f_dGn0tFjFBCpI8Cf6/view?usp=sharing

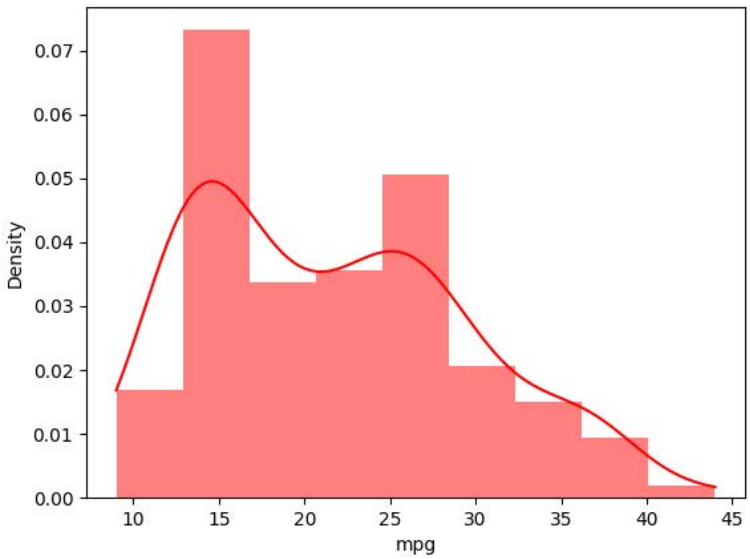
SOURCE CODE

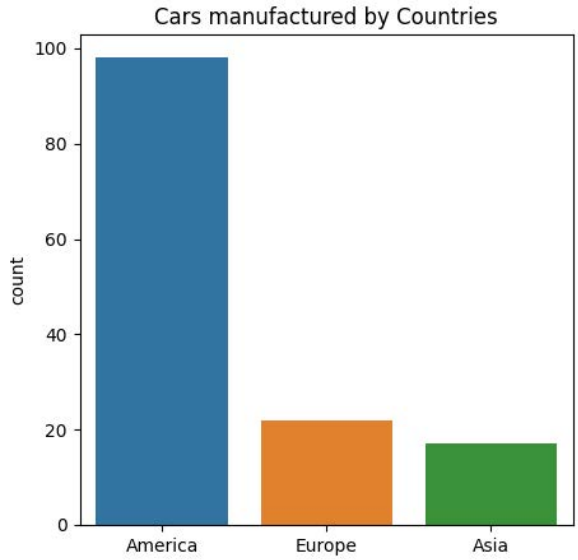
```
import pandas as pd
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from I  ython.display import display
import statsmodels as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.model_selection import
train_test_split,  ridSearchC  ,RandomizedSearchC
from sklearn.linear_model import  inearRegression,Ridge,  asso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import
Random  orestRegressor,  radient  oostingRegressor
from sklearn.metrics import r2_score,mean_squared_error
from sklearn import preprocessing
df_cars = pd.read_csv("CARS.csv")

def dataorg():
    df_cars.horsepower =
df_cars.horsepower.str.replace(' ',' a  ').astype(float)
    df_cars.horsepower.fillna(df_cars.horsepower.mean(),inplace=True)
    df_cars.horsepower = df_cars.horsepower.astype(int)
    df_cars.info()

    df_cars['car name'] = df_cars['car name'].str.replace('chevroelt chevrolet
chevy','chevrolet')
    df_cars['car name'] = df_cars['car name'].str.replace('maxda
mazda','mazda')
    df_cars['car name'] = df_cars['car name'].str.replace('mercedes mercedes
benz mercedes benz','mercedes')
    df_cars['car name'] = df_cars['car name'].str.replace('toyota
toyouta','toyota')
    df_cars['car name'] = df_cars['car name'].str.replace('vokswagen
volkswagen vw','volkswagen')
    df_cars.groupby(['car name']).sum().head()

    display(df_cars.describe().round(2))
    df_cars['origin'] = df_cars['origin'].replace(  1: '  SA', 2: 'Europe', 3: 'Asia'  )
    df_cars.head()
```

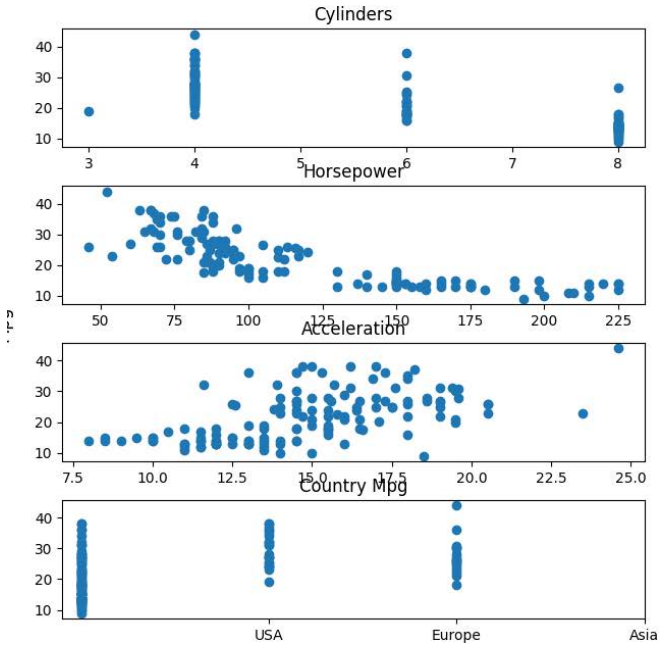
```
def graphs():

    sns_plot = sns.histplot(df_cars["mpg"], color="red", label="100% Equities", kde=True,
stat="density", linewidth=0)


    plt.figure(figsize=(10,6))
    sns.heatmap(df_cars.corr(),cmap=plt.cm.Reds,annot=True)
    plt.title('Heatmap displaying the relationship between the features of the data',
              fontsize=13)
    plt.show()



    fig, ax = plt.subplots(figsize = (5, 5))
    sns.countplot(x = df_cars.origin.values, data=df_cars)
    labels = [item.get_text() for item in ax.get_xticklabels()]
    labels[0] = 'America'
    labels[1] = 'Europe'
    labels[2] = 'Asia'
    ax.set_xtickladef graphs():

    sns_plot = sns.histplot(df_cars["mpg"], color="red", label="100% Equities", kde=True,
stat="density", linewidth=0)

bels(labels)
    ax.set_title("Cars manufactured by Countries")
    plt.show()

    fig, ax = plt.subplots(6, 2, figsize = (15, 13))
    sns.boxplot(x= df_cars["mpg"], ax = ax[0,0])
    sns.histplot(df_cars['mpg'], ax = ax[0,1])
    sns.boxplot(x= df_cars["cylinders"], ax = ax[1,0])
    sns.histplot(df_cars['cylinders'], ax = ax[1,1])
    sns.boxplot(x= df_cars["displacement"], ax = ax[2,0])
    sns.histplot(df_cars['displacement'], ax = ax[2,1])
    sns.boxplot(x= df_cars["horsepower"], ax = ax[3,0])
    sns.histplot(df_cars['horsepower'], ax = ax[3,1])
    sns.boxplot(x= df_cars["weight"], ax = ax[4,0])
    sns.histplot(df_cars['weight'], ax = ax[4,1])
    sns.boxplot(x= df_cars["acceleration"], ax = ax[5,0])
    sns.histplot(df_cars['acceleration'], ax = ax[5,1])
    plt.tight_layout()
```

```
    plt.figure(1)
    f,axarr = plt.subplots(4,2, figsize=(10,10))
    mpgval = df_cars.mpg.values
    axarr[0,0].scatter(df_cars.cylinders.values, mpgval)
    axarr[0,0].set_title('Cylinders')
    axarr[0,1].scatter(df_cars.displacement.values, mpgval)
    axarr[0,1].set_title('Displacement')
    axarr[1,0].scatter(df_cars.horsepower.values, mpgval)
    axarr[1,0].set_title('Horsepower')
    axarr[1,1].scatter(df_cars.weight.values, mpgval)
    axarr[1,1].set_title('   eight')
    axarr[2,0].scatter(df_cars.acceleration.values, mpgval)
    axarr[2,0].set_title('Acceleration')
    axarr[2,1].scatter(df_cars["model year"].values, mpgval)
    axarr[2,1].set_title('   odel   ear')
    axarr[3,0].scatter(df_cars.origin.values, mpgval)
    axarr[3,0].set_title('Country    pg')

    axarr[3,0].set_xticks([1,2,3])
    axarr[3,0].set_xticklabels(["  SA","Europe","Asia"])
    axarr[3,1].axis("off")
    f.text( 0.01, 0.5, '   pg', va='center', rotation='vertical', fontsize = 12)
    plt.tight_layout()
    plt.show()

    sns.set(rc=  'figure.figsize':(11.  ,  .2  )  )
    cData_attr = df_cars.iloc[:, 0:  ]
    sns.pairplot(cData_attr, diag_kind='kde')

    df_cars.hist(figsize=(12,  ),bins=20)
    plt.show()
def acc_check():

      accuracy check

    feature_cols = ['mpg','displacement','horsepower','weight','acceleration']
       = df_cars[feature_cols]
    y = df_cars.cylinders
     _train,   _test, y_train, y_test = train_test_split(  , y, random_state=0)
    from sklearn.linear_model import   ogisticRegression
    logreg =   ogisticRegression()
    logreg.fit(  _train, y_train)
    y_pred_class = logreg.predict(  _test)
    from sklearn import metrics
    print(' n n t tAccuracy = ',(metrics.accuracy_score(y_test, y_pred_class))  100,'%')

   main pg

dataorg()

graphs()

acc_check()
```

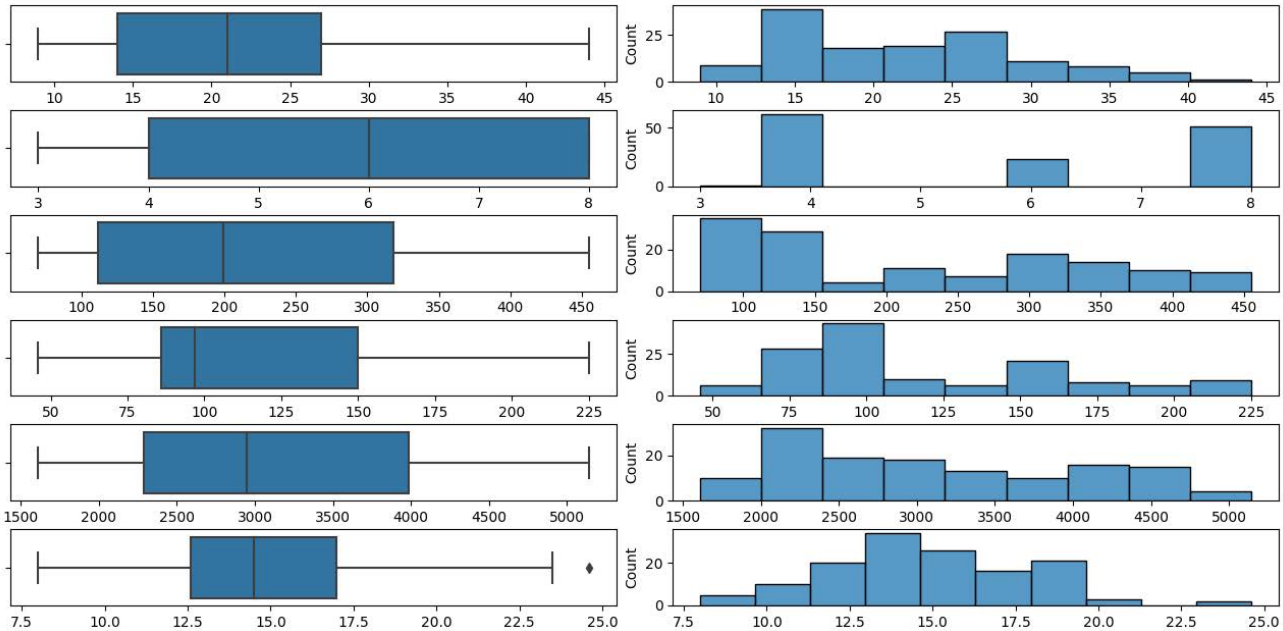
SCREENSHOTS

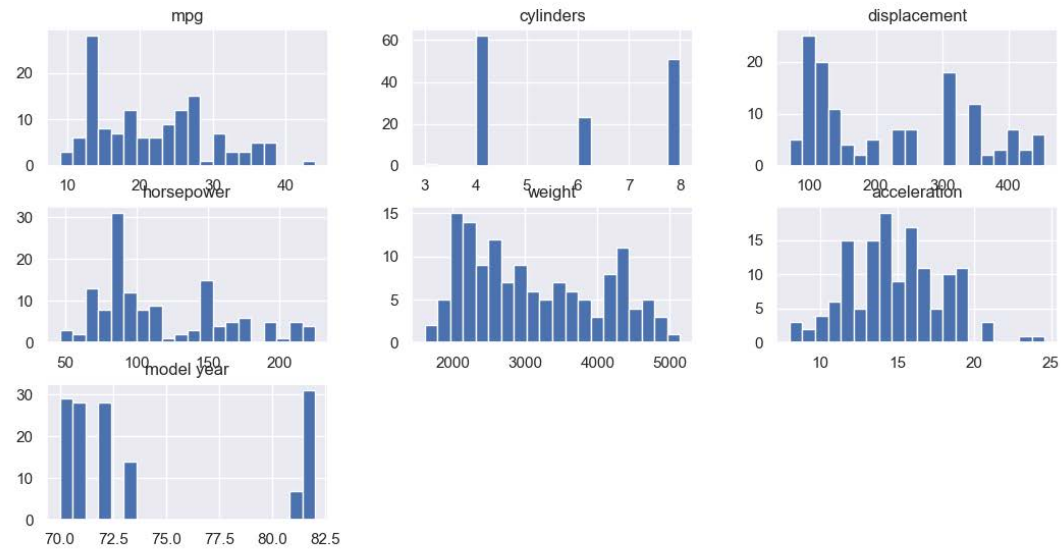




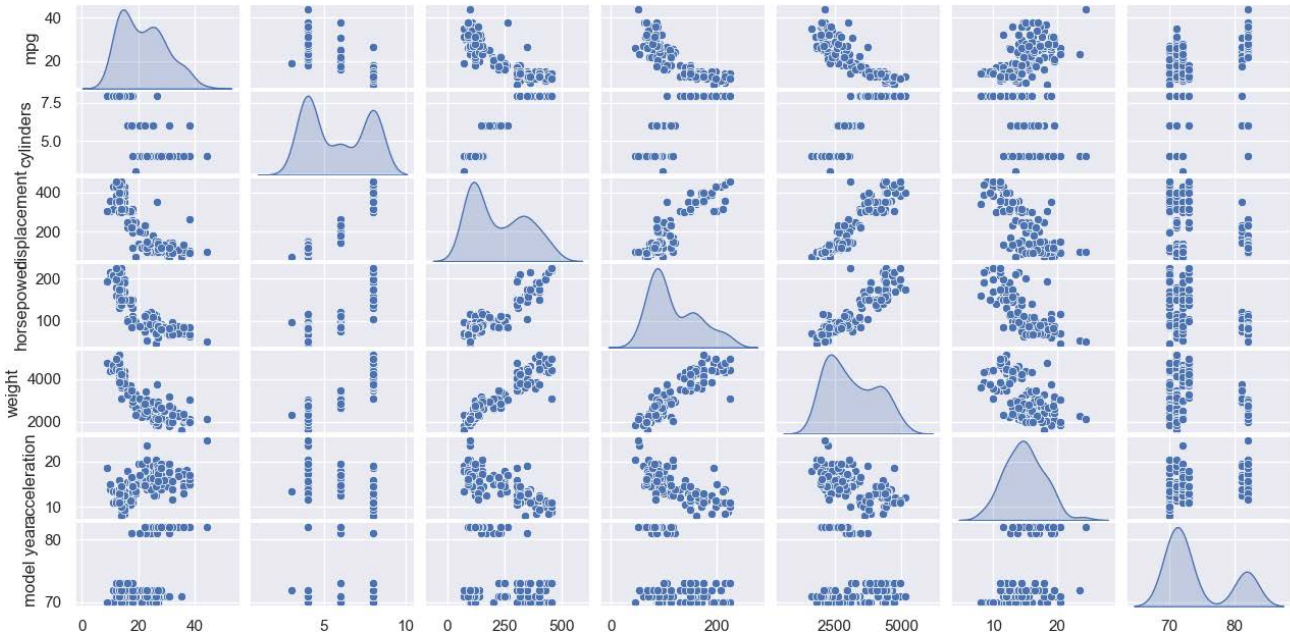


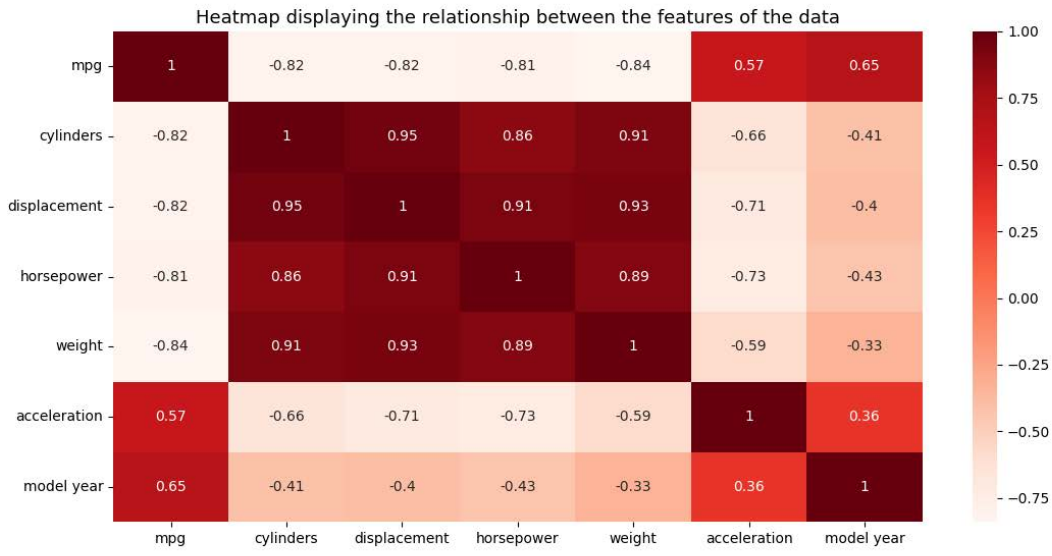
SCREENSHOTS





SCREENSHOTS






```
Accuracy =  88.57142857142857 %
```