# -\*- coding: utf-8 -\*-

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import data\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

import country as cou

from IPython import get\_ipython

ipy = get\_ipython()

if ipy is not None:

ipy.run\_line\_magic('matplotlib', 'inline')

CMR = pd.read\_csv("CMR Dataset-REGRESSION.csv")

CMR.head()

plt.show()

cols= ['elpca','ele','twe','twpca' ,'twen' ,'enggpcmaverage','semenggt', 'label']

X1=CMR[cols]

feature\_cols = ['elpca','ele','twe','twpca' ,'twen' ,'enggpcmaverage','semenggt']

X = CMR['elpca'].values.reshape(-1,1)

y = CMR.label

correlation\_matrix = X1.corr().round(2)

sns.heatmap(data=correlation\_matrix, annot=True)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

regressor = LinearRegression()

regressor.fit(X\_train, y\_train) #Data the algorithm

#To retrieve the intercept:

print(regressor.intercept\_)

#For retrieving the slope:

print(regressor.coef\_)

y\_pred = regressor.predict(X\_test)

print(y\_pred)

CMR1= pd.read\_csv("R-TEST1.csv")

CMR1.head()

CMR1.plot(x='semcgpa', y='career', style='o')

plt.title('CGPA vs Career')

plt.xlabel('CGPA')

plt.ylabel('CAREER')

plt.show()

colors = ("red", "green", "blue")

CMR2= pd.read\_csv("R-TEST1.csv")

CMR2.head()

CMR2.plot(x='coserver', y='co', style='o')

plt.title('Coserver vs Extra asn')

plt.xlabel('Coserver')

plt.ylabel('ASN')

plt.show()

df = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

df

df1 = df.head(25)

df1.plot(kind='pie',figsize=(16,10))

plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')

plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')

plt.show()

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

sns.distplot(CMR['label'], bins=30)

plt.show()

plt.figure(figsize=(20, 5))

plt.scatter(X\_test, y\_test, color='gray')

plt.plot(X\_test, y\_pred, color='red', linewidth=2)

plt.show()