

Project 2 - Training Impact Analyser

#importing the python libraies

Line no1 import numpy as np

!pip install numpy

Line no 2 import pandas as pd

!pip install pandas

Line no 3 import matplotlib.pyplot as plt

!pip install matplotlib

Line no 4 import seaborn as sns

!pip install seaborn

Line no 5 from scipy.stats import pearsonr

!pip install scipy

=> from scipy-> python library for scientific statistical analysis

=> stats->it is a sub module within scipy library

=> import pearsons

=> import-> we are calling external module

=> pearsonr -> pearsonr is a function within a scipy.stats

Line no 6 from sklearn.linear_model import LinearRegression

!pip install scikit-learn

=> sklearn.-> ml library

=> linear_model-> it is a sub module with sklearn library

(Ridge Regression (L1)

Lasso Regression (L2)

Elastic net (L1+L2))

=> import-> we are calling external module

=> LinearRegression-> it is class of sklearn.linear_model

Sample Data

Line no 7 data = {

Line no 8 'Training_Hour: [5,10,15,20,25,30,35,40,45,50],

Line no 9 'Performance_Score': [55,60,65,70,72,78,85,87,90,95]

Line no 10 }

Line no 11 df = pd.DataFrame(data)

=> df-> variable name

=> pd-> alias name for pandas

=> = -> assignment operator

=> DataFrame-> collection of row and column

=> data-> variable name for calling

Line no 12 pearson_corr,_ = pearson(df['Training_Hour'].df['Performance_Score'])

=> pearson_corr-> pearsonr return two value

=> _ -> means => means P Value

=> Pearsonr > 1st value of pearsonr function

=> df['Training_Hour']-> we are calling training hour from dataframe

=> df['Performance_Score']-> we are calling performance state from dataframe

#Pearson's Correlation

Line no 13 print(Pearson Correlation Coefficient: {pearson_corr:2f})

=> print()-> used to get output on console

=> .2f-> used to get 2 decimals place

scatter Plot With Regression Line

Line no 14 `sns.regplot(x=df['Training_Hour'],df['Performance_Score'],ci=None)`

=> `regplot`-> it is a function in seaborn used to create regression plot

=> `x=df['Training_Hour']`-> Independent Variable

=> `y= df['Performance_Score']`-> Dependent variable are output variable

=> `ci=None`-> Coefficient interval

=> `ci`-> it is a range of value which is used to estimate population

Line no 15 `plt.xlabel('Training Hour')`

=> `plt`-> `plt` alias for `plot`

=> `xlabel`-> label on x-axis

=> `'Training Hour'`-> label for x axis

Line no 16 `plt.ylabel('Performance Score')`

=> `plt`-> `plt` alias for `plot`

=> `ylabel`-> label on y-axis

=> `'Performance Score'`-> label for y axis

Line no 17 `plt.title('training hour vs pearson score')`

=> `plt`-> `plt` alias for `plot`

=> `title`-> title for graph

=> `'training hour vs pearson score'`-> text on title

Line no 18 `plt.show()`

#Method of Least Square

Line no 19 `X = df['Training_Hour'].value.reshape(-1,1)`

=> X-> variable name

=> `df['Training_Hour']`-> we are calling training hour with the help of df

=> `.values`->it is used to convert pandas series to numpy array

=> `.reshape`-> Function numpy used to change the shape of an array

=> `(-1,1)` => used to convert 1D to 2D array

Line no 20 `y = df['Performance_Score'].value.reshape(-1,1)`

=> y-> variable name

=> `df['Performance_Score']`-> we are calling performance score with the help of df

=> `.values`->it is used to convert pandas series to numpy array

=> `.reshape`-> Function numpy used to change the shape of an array

=> `(-1,1)` => used to convert 1D to 2D array

Line no 21 `model = LinearRegression()`

=> model-> variable name

=> `=` -> Assignment Operator

=> `LinearRegression()`-> it is a function

Line no 22 `model.fit(X,y)`

=> model-> object for linear regression

=> `.` -> refer to

=> `fit` -> it is a function of scikit learn

=> X-> it input data

=> y-> output data

Line no 23 `y_pred = model.predict(x)`

=> `y_pred`-> variable name to predict value for 'y axis'

=> `=` -> assignment operator

=> model -> model object for learn regression

=> `.` -> refer to

=> `predict` -> function used to train model to 'x'

plot the Least Square Regression

Line no 24

plt.scatter(df['Training_Hour'],df['Performance_Score'],color='Blue',label='Actual Data')

=>plt-> alias name for matplotlib

=> scatter-> scatter function use to plot scatter on the console

=> df -> It is data frame

=> df['Training_Hour'] -> we are calling the 'Training_Hour' by using df

=> df -> It is data frame

=>df['Performance_Score']-> we are calling the 'Performance_Hour' by using df

=> color-> used to set blue color for data point

=>label-> 'Actual Data'

Line no 25 plt.plot(df['Training_Hour'],y_pred,color='Red',label='Regression line')

=>plt-> alias name for matplotlib

=> plot-> it is function in matplotlib

=> df['Training_Hour'] -> we are calling the 'Training_Hour' by using df

=> y_pred-> prediction value for y_axis

=>df['Performance_Score']-> we are calling the 'Performance_Hour' by using df

=> color='Red'-> used to set blue color for regression line

=>label='Regression line' -> used to set label for legend

Line no 26 plt.xlabel('Training Hour')

Used to set label on x_axis which is training_hour

Line no 27 plt.ylabel('performance Scale')

Used to set label on y_axis which is performance scale

Line no 28 plt.title('Method of least square linear Regression')

Used to set title which is method of least square linear regression

Line no 29 plt.legend()

=>uses to plot legend on graph

Line 30 plt.show()

=>used to plot all diagram on graph

#Display Regression coefficient

Line 31 print('Slope(m):{model.coef_[0][0]:2F}')

=>print()-> used to show the output on console

=>f-> formatting string

=>slope(m)-> a string that represents output.

=>{model.coef_[0][0]:2F}->

=>model.coef-> slope for regression model

=>[0][0]-> regression model start from zero value

=>:2f=> two decimal place

Line no 32 print('intercept (b): {model.intercept[0][0]:2f}')

=>print()-> used to show the output on console

=>f-> formatting string

=>intercept(b)-> a string that represents output.

=>{model.intercept[0]:2F}->

=>model.intercept-> intercept for regression model

=>[0][0]-> regression model start from zero value

=>:2f=> two decimal place