Data Science

Working with Data

**Task II: Multiple Regression Analysis with Python**

**Part I:**

Data Exploration and Preparation:

Using the pandas library, load the dataset into a DataFrame and perform an initial exploration to understand the data structure and summary statistics.

**Python code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

dataframe = pd.read\_csv("D:\Airdata.csv")

dataframe.head()

**Results:**

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

**Python code:**

A screenshot of a computer

Description automatically generated

**Python code:**

#finding how many rows and columns

dataframe.shape

**Results:**

(8784, 8)

A white rectangular object with black text

Description automatically generated

**Python code:**

#finding general informations of each columns like Count, Mean, Standard deviation, Minimum values, Maximum values

dataframe.describe()

**Results:**

**A table with numbers and lines

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Python code:**

#drop duplicate values / rows

dataframe.drop\_duplicates(inplace=True)

dataframe.shape

**Results:**

(8784, 8)

Same output as earlier so it means there are no duplicate values / rows in this data source.

**A screenshot of a computer

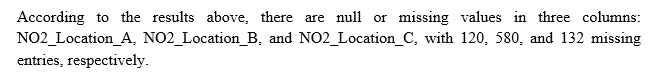
Description automatically generated**

Check for missing values and handle them appropriately, explaining your chosen method.

**Python code:**

A screenshot of a computer

Description automatically generated



**Python code:**

A screenshot of a computer

Description automatically generated

**Find NaN / null / missing values and replace with zero**

**Python code:**

**A screenshot of a computer

Description automatically generated**

Or, we can simply drop null values using the below Python code:

df2.dropna(inplace=True)

**Compare before and after:**

**Before:**

**A screenshot of a computer

Description automatically generated**

**After:**

**A screenshot of a graph

Description automatically generated**

Therefore, by substituting zero for null values, the data have remained unchanged.

The dependant variable is the Temperature and all other variables are independent.

**Python code:**

sns.jointplot (x=df2['Temperature'], y=df2['Humidity'], data=df2, size=5)

**Results:**

A screenshot of a graph

Description automatically generated

According to the joint plot shown above, humidity falls as temperature rises.

According to science, this is accurate because the relative humidity decreases as the temperature rises while the air's water vapor content stays constant.

|  |
| --- |
| (National Weather Service, 2023) References National Weather Service, 2023. *National Weather Service.* [Online]  Available at: https://www.weather.gov/lmk/humidity [Accessed 8 September 2023]. |

**Python code:**

x = df2['Temperature']

y = df2['Wind\_Speed']

plt.scatter(x,y)

plt.xlabel('Temperature')

plt.ylabel('Wind Speed')

plt.title('Temperature vs Windspeed')

plt.show()

**Results:**

A screen shot of a graph

Description automatically generated

**Python code:**

sns.scatterplot(x = "Temperature", y = "Wind\_Direction", hue = "Wind\_Speed" , data = df2)

**Results:**

A screen shot of a graph

Description automatically generated

According to the above 2 diagrams, the temperature drops as the wind speed increases.

|  |
| --- |
| According to science, the heat is taken out when the wind speed increases.  (National Weather Service, 2023) References National Weather Service, 2023. *National Weather Service.* [Online]  Available at: https://www.weather.gov/ama/WindChill [Accessed 8 Spetember 2023]. |

**Python code:**

import matplotlib.pyplot as plt

# visualizing the relationship between features and response using scatterplots

fig, axs = plt.subplots(1, 3, sharey=True)

df2.plot(kind='scatter', x='NO2\_Location\_A', y='Temperature', ax=axs[0], figsize=(16, 8))

df2.plot(kind='scatter', x='NO2\_Location\_B', y='Temperature', ax=axs[1])

df2.plot(kind='scatter', x='NO2\_Location\_C', y='Temperature', ax=axs[2])

**Results:**

A screenshot of a computer screen

Description automatically generated

It is clear from the three diagrams above that a high temperature causes the NO2 level to decrease.

|  |
| --- |
| The high temperatures cause low NO2 concentrations.  (Harkey, 2015) References Harkey, 2015. *Science Direct.* [Online]  Available at: https://www.sciencedirect.com/science/article/abs/pii/S0269749120368524 [Accessed 08 September 2023]. |

**The below Histogram will produce each and every column because here column names are not defined.**

**Python code:**

#histogram of each and every column

df2.hist()

A screenshot of a computer

Description automatically generated

**Histogram of Temperature column**

**Python code:**

#histogram of each and every column

plt.hist(df2['Temperature'])

**A screen shot of a graph

Description automatically generated**

**Distribution curves**

**Python code:**

#Distribution curves

plt.figure(figsize=(12,12))

plt.subplot(3,3,1)

sns.distplot(df2.Temperature)

plt.subplot(3,3,2)

sns.distplot(df2.Humidity)

plt.subplot(3,3,3)

sns.distplot(df2.Wind\_Speed)

plt.subplot(3,3,4)

sns.distplot(df2.Wind\_Direction)

plt.subplot(3,3,5)

sns.distplot(df2.NO2\_Location\_A)

plt.subplot(3,3,6)

sns.distplot(df2.NO2\_Location\_B)

plt.subplot(3,3,7)

sns.distplot(df2.NO2\_Location\_C)

**A group of blue graphs

Description automatically generated**

**A graph of a number of numbers

Description automatically generated with medium confidence**

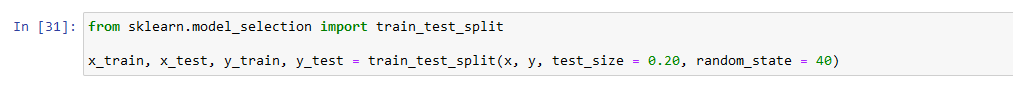
These show 2 graphs at once, a histogram and a trend.

Split the dataset into a training set (80%) and a testing set (20%) using the train\_test\_split function from the sklearn library.

**Python code:**

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

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.20, random\_state = 40)



**Part II:**

Multiple Regression Analysis and Model Evaluation:

Using the statsmodels or sklearn library, create a multiple regression model to predict temperature based on given factors. Include the Python code you used for creating the model.

Evaluate the performance of your model using appropriate metrics, such as R-squared, Adjusted R-squared, Mean Squared Error.

**Python code:**

import statsmodels.api as sm

model2 = sm.OLS(y\_train, x\_train).fit()

model2.summary()

A screenshot of a computer

Description automatically generated

**Using sklearn – Python code:**

from sklearn.metrics import r2\_score

r2 = r2\_score(y\_test, model2.predict(x\_test))

r2

A screenshot of a computer code

Description automatically generated

**Let’s consider two variables.**

**Temperature as the dependant variable.**

**Humidity as the independent variable.**

**Python code:**

df\_2 = df2[['Temperature', 'Humidity']]

# Using just the two attributes that were chosen from the dataset

df\_2.columns = ['Temperature', 'Humidity']

# show the first five rows

df\_2.head()

**Screenshot:**

A screenshot of a computer

Description automatically generated

**Checking the relationship between Temperature and Humidity using the Scatter Plot**

**Python code:**

**A screen shot of a graph

Description automatically generated**

According to the above, high temperatures cause the humidity to decrease.

#The model needs to be trained.

**A screenshot of a computer code

Description automatically generated**

**Predicted Value:**

**Python code:**

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

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

plt.plot(X\_test, y\_pred, color ='k')

plt.show()

**A graph with blue dots and a black line

Description automatically generated**

**Reducing the number of records and see the prediction:**

**Python code:**

**A screen shot of a graph

Description automatically generated**

**Checking the linear regression score:**

**Python code:**

**A screenshot of a computer program

Description automatically generated**

**Output:**

**Python code:**

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

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

plt.plot(X\_test, y\_pred, color ='k')

plt.xlabel("Temperature")

plt.ylabel("Humidity")

plt.show()

**A graph with blue dots

Description automatically generated**

**By using below 3 evaluation metrics, we can check the linear regression model.**

**Python code:**

**A screenshot of a computer program

Description automatically generated**

The above calculations say that,

The Mean Absolute Error (MAE) is approximately 8.5

The Mean Squared Error (MSE) is approximately 115.5

The Root Mean Squared Error (RSME) is approximately 10.7

A regression problem's average absolute difference between the anticipated and actual values is measured using the MAE.

The MSE risk approach allows us to express the average squared difference between a feature or variable's predicted and actual values.

The transition between values predicted by a model and actual values is calculated using RSME.

|  |
| --- |
| (javatpoint, 2021) References javatpoint, 2021. *javatpoint.* [Online]  Available at: https://www.javatpoint.com/rsme-root-mean-square-error-in-python [Accessed 8 September 2023]. |
| (educative, 2023) References educative, 2023. *educative.* [Online]  Available at: https://www.educative.io/answers/mean-absolute-error-in-sklearn [Accessed 8 September 2023]. |