Computer Engineering Department National University of Technology Islamabad, Pakistan

Introduction to Data Mining

Practice Exercise 01

Name: Muhammad Sami Uddin Rafay

Roll Number: F18604013

Submitted To: Dr. Kamran Javed

Date: 11 November 2020

Practice Exercise 01

Understand and Analyse Weather Data.

Objective:

- Understanding complexity of data and importance of pre-processing.

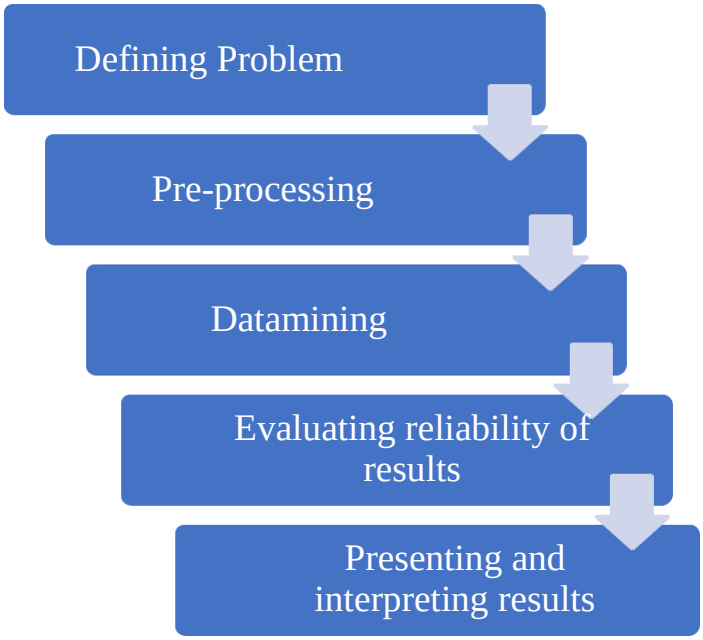
Equipment/Software Required:

- Python (Spyder 4.0 Anaconda Distribution)
- Weather data file (.csv) from Kaggle.

Background:

Data mining:

Data mining is defined as a process used to extract usable data from a larger set of any raw data. It implies analysing data patterns in large batches of data using one or more software. The data mining process breaks down into five steps:



Data mining has a long history. It emerged with computing in the 1960s through the 1980s. Historically, data mining was an intensive manual coding process and it still involves coding ability and knowledgeable specialists to clean, process, and interpret data mining results today. Data specialists need statistical knowledge and some programming language knowledge to complete data mining techniques accurately.

Significance:

Data mining find its application across various industries such as market analysis, business management, fraud inspection, corporate analysis and risk management, among others.

Key features of data mining are:

- Automatic pattern predictions based on trend and behaviour analysis.
- Prediction based on likely outcomes.
- Creation of decision-oriented information.
- Focus on large data sets and databases for analysis.
- Clustering based on finding and visually documented groups of facts not previously known.

Tasks:

- Visualize weather data features.
- Find correlation of temperature with all other features.
- Visualize a comparative bar plot of correlation values.

Code:

```
# importing neccessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# reading "weatherHistory.csv" using read_csv under pandas
Weather_Dataset=pd.read_csv("D:\weatherHistory.csv")

# printing head of Weather_Dataset
print("\n")
print("Head of Weather_Dataset")
print(Weather_Dataset.head())

# storing each column of Weather_Dataset in a separate variable
Temperature = Weather_Dataset["Apparent Temperature (C)"]

Humidity = Weather_Dataset["Humidity"]

Wind_Speed= Weather_Dataset["Wind Speed (km/h)"]

Wind_Bearing = Weather_Dataset["Wind Bearing (degrees)"]

Visibility = Weather_Dataset["Visibility (km)"]

Loud_Cover = Weather_Dataset["Loud Cover"]

Pressure = Weather_Dataset["Pressure (millibars)"]

D_Summary = Weather_Dataset["Daily Summary"]
print("\n")

# printing correlation coefficient of temperature with all vaiables
print("Correlation Coefficient between Temperature and Humidity")
print(np.corrcoef(Temperature, Humidity))
print("\n")
```

```
print("Correlation Coefficient between Temperature and Wind_Speed")
print(np.corrcoef(Temperature, Wind_Speed))
print("\n")
print("Correlation Coefficient between Temperature and Wind_Bearing")
print(np.corrcoef(Temperature, Wind_Bearing))
print("\n")
print("Correlation Coefficient between Temperature and Visibility")
print(np.corrcoef(Temperature, Visibility))
print("\n")
print("Correlation Coefficient between Temperature and Loud_Cover")
print(np.corrcoef(Temperature, Loud_Cover))
print("\n")
print("Correlation Coefficient between Temperature and Pressure")
print(np.corrcoef(Temperature, Pressure))
print("\n")
```

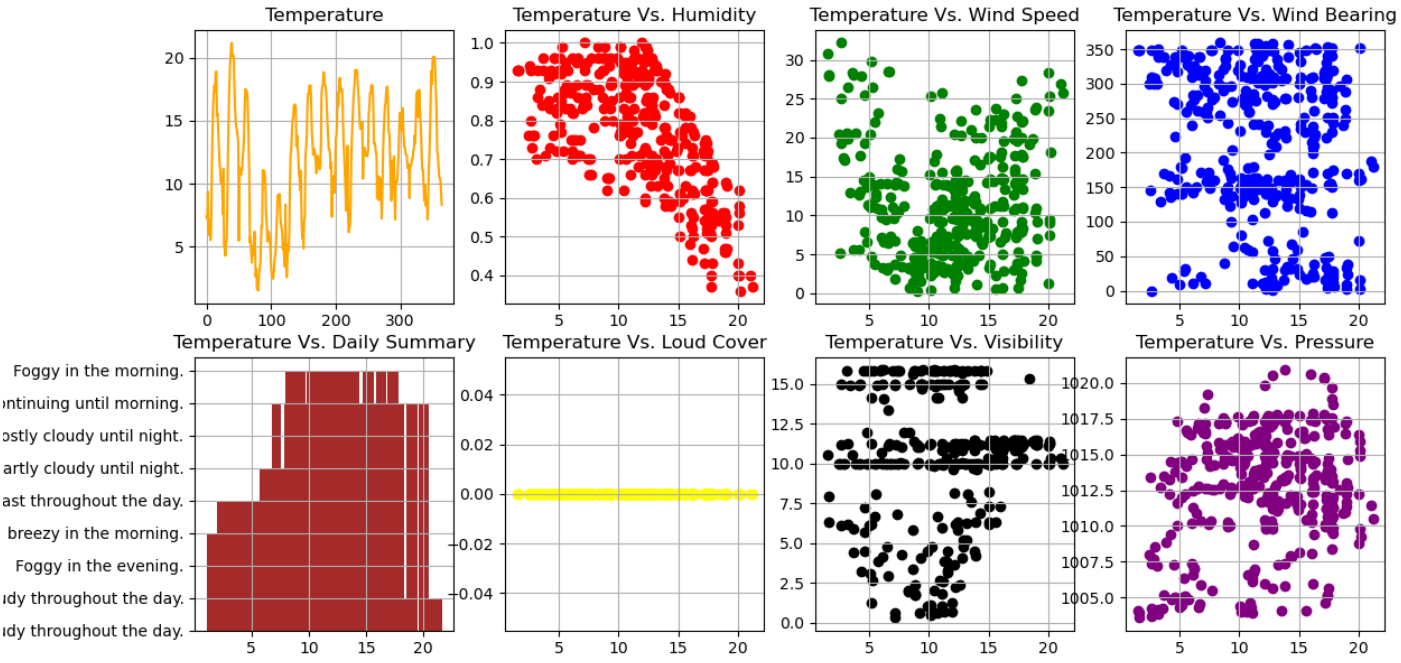
 # Sub-plotting all correlations with temperatures using scatter plot

```
plt.figure(2, figsize=(20,10))
plt.subplot(241)
plt.plot(Temperature[0:365],color='orange')
plt.grid()
plt.title("Temperature")

plt.subplot(242)
plt.scatter(Temperature[0:365],Humidity[0:365],color='red')
plt.grid()
plt.title("Temperature Vs. Humidity")

plt.subplot(243)
plt.scatter(Temperature[0:365],Wind_Speed[0:365],color='green')
plt.grid()
plt.title("Temperature Vs. Wind Speed")

plt.subplot(244)
plt.scatter(Temperature[0:365],Wind_Bearing[0:365],color='blue')
plt.grid()
plt.title("Temperature Vs. Wind Bearing")

plt.subplot(245)
plt.bar(Temperature[0:365],D_Summary[0:365],color='brown')
plt.title("Temperature Vs. Daily Summary")
plt.grid()

plt.subplot(246)
plt.scatter(Temperature[0:365],Loud_Cover[0:365],color='yellow')
plt.grid()
plt.title("Temperature Vs. Loud Cover")

plt.subplot(247)
plt.scatter(Temperature[0:365],Visibility[0:365],color='black')
plt.grid()
plt.title("Temperature Vs. Visibility")

plt.subplot(248)
plt.scatter(Temperature[0:365],Pressure[0:365],color='purple')
```

```
plt.grid()
plt.title("Temperature Vs. Pressure")

plt.show()
```

Output:

```
Head of Weather_Dataset
                Formatted Date  ...                Daily Summary
0  2006-04-01 00:00:00.000 +0200  ...  Partly cloudy throughout the day.
1  2006-04-01 01:00:00.000 +0200  ...  Partly cloudy throughout the day.
2  2006-04-01 02:00:00.000 +0200  ...  Partly cloudy throughout the day.
3  2006-04-01 03:00:00.000 +0200  ...  Partly cloudy throughout the day.
4  2006-04-01 04:00:00.000 +0200  ...  Partly cloudy throughout the day.

[5 rows x 12 columns]


Correlation Coefficient between Temperature and Humidity
[[ 1.       -0.602571]
 [-0.602571  1.      ]]


Correlation Coefficient between Temperature and Wind_Speed
[[ 1.        -0.0566497]
 [-0.0566497  1.       ]]


Correlation Coefficient between Temperature and Wind_Bearing
[[1.         0.02903052]
 [0.02903052 1.        ]]


Correlation Coefficient between Temperature and Visibility
[[1.         0.38171847]
 [0.38171847 1.        ]]


Correlation Coefficient between Temperature and Loud_Cover
[[ 1. nan]
 [nan nan]]


Correlation Coefficient between Temperature and Pressure
[[ 1.00000000e+00 -2.18999786e-04]
 [-2.18999786e-04  1.00000000e+00]]
```



Results and Discussions:

In this practical I learned to read .csv file using read_csv() command of Pandas library, to calculate correlation of different weather variables with temperature using corrcoef(x,y) command of NumPy library and plotted all correlations in subplots(Scatter) using matplotlib library of python.

Conclusion:

The correlation coefficient is a statistical measure of the strength of the relationship between the relative movements of two variables. The values range between -1.0 and 1.0. A calculated number greater than 1.0 or less than -1.0 means that there was an error in the correlation measurement. A correlation of -1.0 shows a perfect negative correlation, while a correlation of 1.0 shows a perfect positive correlation. A correlation of 0.0 shows no linear relationship between the movement of the two variables.