Rainfall prediction using Linear Regression

1) Cleaning the data

In [ ]:

```
# importing libraries
import pandas as pd
import numpy as np
import sklearn as sk
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [ ]:

```
# อ่านข้อมูลที่เตรียมไว้
data = pd.read_csv('austin_weather.csv')
data.head()
```

In [ ]:

```
# ลบ features ที่ไม่ต้องการ
data = data.drop(['Events', 'Date', 'SeaLevelPressureHighInches',
                  'SeaLevelPressureLowInches'], axis = 1)
data.head()
```

In [ ]:

```
data.shape
```

In [ ]:

```
# some values have 'T' which denotes trace rainfall (row number 6,8,9)
# we need to replace all occurrences of T with 0
# so that we can use the data in our model
data = data.replace('T', 0.0)
data
```

In [ ]:

```
# the data also contains '-' which indicates no  (row number 176-179)
# or NIL. This means that data is not available
# we need to replace these values as well.
data = data.replace('-', 0.0)
```

In [ ]:

```
# save the data in a csv file
data.to_csv('austin_final.csv')
```

2) Scikit-learn’s linear regression model

In [ ]:

```
# importing libraries
import pandas as pd
import numpy as np
import sklearn as sk
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [ ]:

```
# read the cleaned data
data = pd.read_csv('austin_final.csv')
```

In [ ]:

```
# the features or the 'x' values of the data
# these columns are used to train the model
# the last column, i.e, precipitation column
# will serve as the label
X = data.drop(['PrecipitationSumInches'], axis = 1)
```

In [ ]:

```
# the output or the label.
Y = data['PrecipitationSumInches']
# reshaping it into a 2-D vector
Y = Y.values.reshape(-1, 1)
```

In [ ]:

```
# consider a random day in the dataset
# we shall plot a graph and observe this day (798)
day_index = 798
days = [i for i in range(Y.size)]
```

In [ ]:

```
# initialize a linear regression classifier
clf = LinearRegression()
```

In [ ]:

```
# train the classifier with our
# input data.
clf.fit(X, Y)
```

In [ ]:

```
# give a sample input to test our model
# this is a 2-D vector that contains values
# for each column in the dataset.
inp = np.array([[74], [60], [45], [67], [49], [43], [33], [45],
                [57], [29.68], [10], [7], [2], [0], [20], [4], [31]])
inp = inp.reshape(1, -1)
```

In [ ]:

```
# print the output.
print('The precipitation in inches for the input is:', clf.predict(inp))
```

In [ ]:

```
# plot a graph of the precipitation levels
# versus the total number of days.
# one day, which is in red, is
# tracked here. It has a precipitation
# of approx. 2 inches.
print("the precipitation trend graph: ")
plt.scatter(days, Y, color = 'g')
plt.scatter(days[day_index], Y[day_index], color ='r')
plt.title("Precipitation level")
plt.xlabel("Days")
plt.ylabel("Precipitation in inches")
plt.show()
```

In [ ]:

```
# filter data for displaying
x_vis = X.filter(['TempAvgF', 'DewPointAvgF', 'HumidityAvgPercent',
                  'SeaLevelPressureAvgInches', 'VisibilityAvgMiles',
                  'WindAvgMPH'], axis = 1)
```

In [ ]:

```
x_vis.columns
```

In [ ]:

```
x_vis.columns.size
```

In [ ]:

```
# plot a graph with a few features (x values)
# against the precipitation or rainfall to observe
# the trends

print("Precipitation vs selected attributes graph: ")
for i in range(x_vis.columns.size):
    plt.subplot(3, 2, i + 1) #Add a subplot to the current figure; nrows=3,ncols=2,index=i+
    #x=days, y=x_vis.columns.values (between 0 and 99)
    plt.scatter(days,
                x_vis[x_vis.columns.values[i][:100]],
                color = 'g')

    plt.scatter(days[day_index],
                x_vis[x_vis.columns.values[i]][day_index],
                color ='r')

    plt.title(x_vis.columns.values[i])

plt.show()
```

A day (in red) having precipitation of about 2 inches is tracked across multiple parameters (the same day is

tracker across multiple features such as temperature, pressure, etc).

The x-axis denotes the days and the y-axis denotes the magnitude of the feature such as temperature, pressure, etc.

From the graph, it can be observed that rainfall can be expected to be high when the temperature is high and humidity is high.