Lab 3 (weather dataset)

June 11, 2020

1 ML program 3

1.1 Demonstrating K-Means algorithm

1.1.1 Importing all necessary packages

[8]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans
%matplotlib inline
```

1.1.2 Reading the data

Sample basically takes only 200 samples from the population (To make it easy to visualize and understand

[10]:

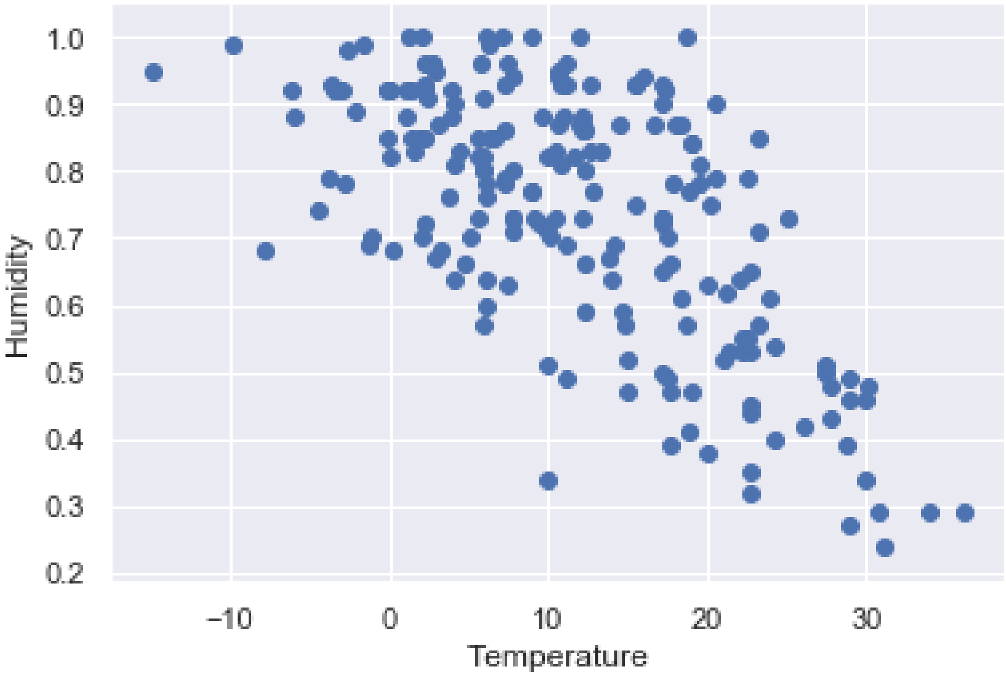
```
data = pd.read_csv('weatherHistory.csv')
sample = data.sample(200)
df = pd.DataFrame(sample, columns = ('Temperature (C)', 'Humidity'))
print(df)
```

```
       Temperature (C)  Humidity
30794        10.850000      0.81
83813        22.766667      0.65
12259         1.466667      0.85
51922        22.172222      0.55
61945        17.116667      0.65
...                ...       ...
67608        12.200000      0.88
14595         5.855556      0.82
94792         3.022222      0.87
8248         26.011111      0.42
10444        -2.100000      0.89

[200 rows x 2 columns]
```

1.1.3 Displaying the raw data as a scatter plot. We have considered 2 features - Humidity and temperature

```
[11]: plt.scatter(df['Temperature (C)'], df['Humidity'])
      plt.xlabel('Temperature')
      plt.ylabel('Humidity')
      plt.show()
```



1.1.4 Copying df into X and using KMeans function (sklearn) and running the algorithm on X. Number of clusters (here) is 4.

```
[12]: X = df.copy()
      kmeans = KMeans(4)
      kmeans.fit(X)
```

```
[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
             random_state=None, tol=0.0001, verbose=0)
```

1.1.5 Setting a column to show predcited (Assigned) classes for display purpose

```
[14]: clusters =X.copy()
      clusters['pred'] = kmeans.fit_predict(X)
      print(clusters)
```

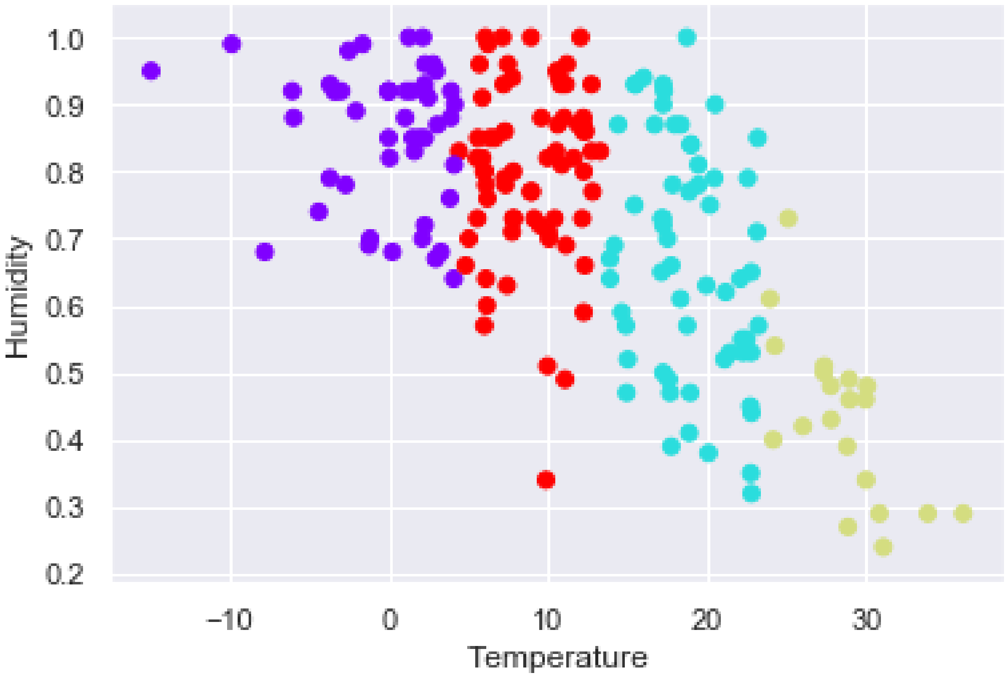
```
       Temperature (C)  Humidity  pred
30794        10.850000      0.81     3
83813        22.766667      0.65     1
12259         1.466667      0.85     0
51922        22.172222      0.55     1
61945        17.116667      0.65     1
...                ...       ...   ...
67608        12.200000      0.88     3
14595         5.855556      0.82     3
94792         3.022222      0.87     0
8248         26.011111      0.42     2
10444        -2.100000      0.89     0

[200 rows x 3 columns]
```

1.1.6 Plotting the data using matplotlib based on assigned clusters.

1.1.7 We have used the clusters with 'pred' column dataframe for plot

```
[15]: plt.scatter(clusters['Temperature (C)'], clusters['Humidity'],␣
        ↪c=clusters['pred'], cmap='rainbow')
      plt.xlabel('Temperature')
      plt.ylabel('Humidity')
      plt.show()
```


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Humidity
−10
0
10
20
30
Temperature

```
[ ]:
```