```
#data = {

#'Temperature': ['Hot', 'Hot', 'Hot', 'Mild', 'Cool', 'Cool', 'Cool', 'Mild', 'Cool', 'Mild']

#'Humidity': ['High', 'High', 'High', 'High', 'Normal', 'Normal', 'Normal', 'High', 'Normal', 'Normal']

#The stpes to carry out kmeans include:
```

The steps in carrying out kmeans include :

- First you select the number of clusters (k)
- Then you select k points at random (finding clusters by selecting 3 random points. These 3 random points will act as centroids)
- Then you create the cluster by measuring the distance of each point to each of the centroids
- Then we calculate the coordinates of the new centroids
- Then we assess the quality of each centroid
- Then we repeat the past 3 steps (make 3 new clusters, calculate the center of each in relation to the centroid and assessing the quality) for around 4 iterations
- Then we plot an elbow graph in order to identify the most optimal point which we will pick as k

```
import numpy as np
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt
import pandas as pd


from sklearn.cluster import KMeans
kmeans = KMeans(init='k-means++')


figures = [['Hot', 'High'], ['Hot','High'], ['Hot','High'], ['Mild','High'], ['Cool','Normal'], ['Cool','Normal'], ['Cool','Normal'],
           ['Mild','High'], ['Cool','Normal'], ['Mild','Normal']]


figures
```

```
[['Hot', 'High'],
 ['Hot', 'High'],
 ['Hot', 'High'],
 ['Mild', 'High'],
 ['Cool', 'Normal'],
 ['Cool', 'Normal'],
 ['Cool', 'Normal'],
 ['Mild', 'High'],
 ['Cool', 'Normal'],
 ['Mild', 'Normal']]
```

```
type(figures)
```

```
list
```

```
titles = ['Temperature', 'Humidity']


titles
```

```
['Temperature', 'Humidity']
```

```
weather = pd.DataFrame(figures, columns=titles)


weather
```

	Temperature	Humidity
0	Hot	High
1	Hot	High
2	Hot	High
3	Mild	High
4	Cool	Normal

(Assumptions made in this program)

- If temperature is hot and humidity is high give 3
- If temperature is mild and humidity is high give 2
- If temperature is mild and humidity is normal give 1
- If temperature is cool and humidity is normal give 0

```
newFigures = [['Hot', 'High',3], ['Hot','High',3], ['Hot','High',3], ['Mild','High',2], ['Cool','Normal', 0], ['Cool','Normal', 0], ['Cool',
           ['Mild','High', 2], ['Cool','Normal', 0], ['Mild','Normal', 0]]
```

```
newTitles = ['Temperature', 'Humidity', 'Value']
```

```
uweather = pd.DataFrame(newFigures, columns=newTitles)
```

```
uweather
```

	Temperature	Humidity	Value
0	Hot	High	3
1	Hot	High	3
2	Hot	High	3
3	Mild	High	2
4	Cool	Normal	0
5	Cool	Normal	0
6	Cool	Normal	0
7	Mild	High	2
8	Cool	Normal	0
9	Mild	Normal	0

```
from sklearn import cluster
```

```
data = uweather.drop('Temperature', axis=1)
```

```
data
```

```
done = data.drop('Humidity', axis=1)
```

```
done
```

	Value
0	3
1	3
2	3
3	2
4	0
5	0
6	0
7	2
8	0
9	0

```
k_means = cluster.KMeans(n_clusters=2, max_iter=100, random_state=1)
```

```
k_means
```

```
KMeans(max_iter=100, n_clusters=2, random_state=1)
```

```
k_means.fit(done)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
```

```
KMeans(max_iter=100, n_clusters=2, random_state=1)
```

```
labels = k_means.labels_
```

```
labels
```

```
array([1, 1, 1, 1, 0, 0, 0, 1, 0, 0], dtype=int32)
```

```
done
```

	Value
0	3
1	3
2	3
3	2
4	0
5	0
6	0
7	2
8	0
9	0

```
outcome = pd.DataFrame(labels, index=uweather.Temperature, columns=['Cluster ID'])
```

```
outcome
```

Temperature	Cluster ID
Hot	1
Hot	1
Hot	1
Mild	1
Cool	0
Cool	0
Cool	0
Mild	1
Cool	0
Mild	0

```
k_means.cluster_centers_
```

```
array([[0. ],
       [2.6]])
```

```
import numpy as np
```