

```
In [13]: # Assignment 4 - Active In-Class Learning!!!!

import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier

# Values from ppt are changed as per my input
x = [4, 8, 12, 4, 3, 11, 16, 18, 10, 14]
y = [25, 22, 24, 19, 16, 25, 24, 20, 21, 23]
classes = [0, 1, 1, 0, 0, 1, 1, 0, 0, 0]

# Here we are printing the x, y values together as co-ordinates in the graph
data = list(zip(x, y))
data
```

```
Out[13]: [(4, 25),
          (8, 22),
          (12, 24),
          (4, 19),
          (3, 16),
          (11, 25),
          (16, 24),
          (18, 20),
          (10, 21),
          (14, 23)]
```

```
In [14]: # Based on the document scores and target class, we fit KNN method for 1 nearest neighbor:

knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(data, classes)
```

```
Out[14]: KNeighborsClassifier(n_neighbors=1)
```

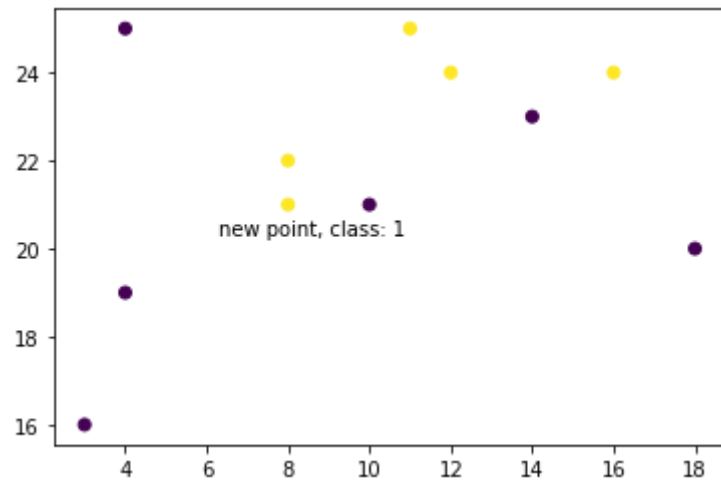
```
In [15]: # take new X and Y arrays values, then redict knn on new document score to obtain a calss of 0 or 1  
# Basically applying same KNN object to identify the class of new, unforeseen documents
```

```
new_x = 8  
new_y = 21  
new_point = [(new_x, new_y)]  
prediction = knn.predict(new_point)  
print(prediction)
```

```
[1]
```

```
In [16]: # plotting graph
```

```
plt.scatter(x + [new_x], y + [new_y], c=classes + [prediction[0]])  
plt.text(x=new_x-1.7, y=new_y-0.7, s=f"new point, class: {prediction[0]}")  
plt.show()
```



```
In [17]: # when we vary the number of neighbors to 5, the number of scores used to classify our new document scores is 5

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(data, classes)
prediction = knn.predict(new_point)
print(prediction)
```

[1]

```
In [18]: # plotting graph

plt.scatter(x + [new_x], y + [new_y], c=classes + [prediction[0]])
plt.text(x=new_x-1.7, y=new_y-0.7, s=f"new point, class: {prediction[0]}")
plt.show()
```

