



| Name of point | Height | weight | Points | Class | Distance from Unknown point  (170,57) | Rank |
| --- | --- | --- | --- | --- | --- | --- |
| A1 | 167 | 51 |  | 0 |  |  |
| A2 | 182 | 62 |  | 1 |  |  |
| A3 | 176 | 69 |  | 1 |  |  |
| A4 | 173 | 64 |  | 1 |  |  |
| A5 | 172 | 65 |  | 1 |  |  |
| A6 | 174 | 56 |  | 0 |  |  |
| A7 | 169 | 58 |  | 1 |  |  |
| A8 | 173 | 57 |  | 1 |  |  |
| A9 | 170 | 55 |  | 1 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

import matplotlib.pyplot as plt

x = [4, 5, 10, 4, 3, 11, 14 , 8, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

classes = [0, 0, 1, 0, 0, 1, 1, 0, 1, 1]

plt.scatter(x, y, c=classes)

plt.show()

from sklearn.neighbors import KNeighborsClassifier

data = list(zip(x, y))

knn = KNeighborsClassifier(n\_neighbors=1)

knn.fit(data, classes)

new\_x = 10

new\_y = 24

new\_point = [(new\_x, new\_y)]

prediction = knn.predict(new\_point)

plt.scatter(x + [new\_x], y + [new\_y], c=classes + [prediction[0]])

plt.text(x=new\_x, y=new\_y, s=f"new point, class: {prediction[0]}")

plt.show()

print(prediction)

runfile('C:/Users/Bala/sky/untitled88.py', wdir='C:/Users/Bala/sky')

[1]

import matplotlib.pyplot as plt

x = [4, 5, 10, 4, 3, 11, 14 , 8, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

classes = [0, 0, 1, 0, 0, 1, 1, 0, 1, 1]

plt.scatter(x, y, c=classes)

plt.show()

from sklearn.neighbors import KNeighborsClassifier

data = list(zip(x, y))

knn = KNeighborsClassifier(n\_neighbors=1)

knn.fit(data, classes)

new\_x = 8

new\_y = 21

new\_point = [(new\_x, new\_y)]

prediction = knn.predict(new\_point)

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()

print(prediction)

knn = KNeighborsClassifier(n\_neighbors=5)

knn.fit(data, classes)

prediction = knn.predict(new\_point)

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()

print(prediction)