Assignment skabelon - MAD

Mikkel Willén

November 2021

Exercise 1

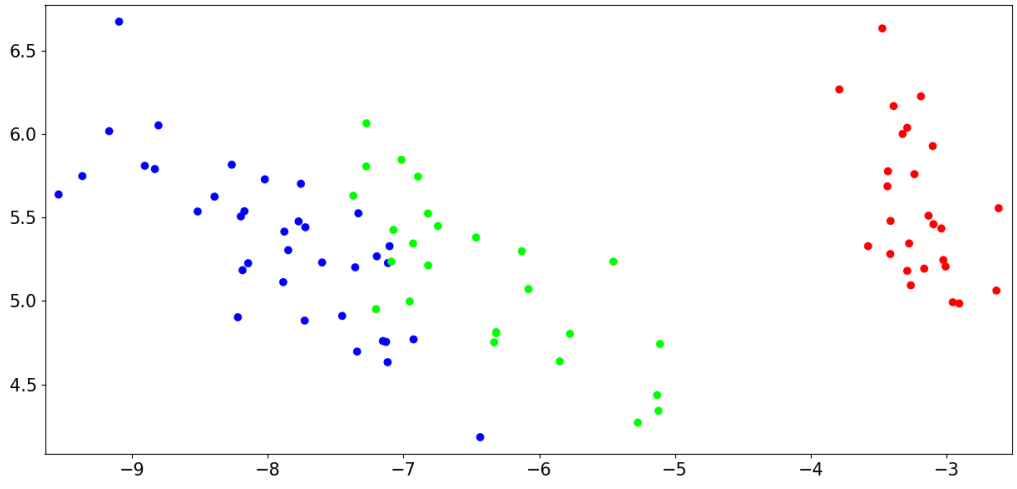
a)

Implementation af PCA

```
def __PCA(data):
    meanTrainingFeatures = np.mean(data.T, axis = 1)

    data_cent = data.T - meanTrainingFeatures.reshape((-1, 1))
    data_cent = np.cov(data_cent)
    PCevals, PCevecs = np.linalg.eigh(data_cent)
    PCevals = np.flip(PCevals, 0)
    PCevecs = np.flip(PCevecs, 1)
    # return
    return PCevals, PCevecs
```

Billede af plot.



b)

Implementation af KNN

```
def __kNNTest(trainingFeatures2D, trainingLabels, n_neighbors, validationFeatures2D,
              validationLabels):
    accuracy = 0.0
    for i in range(validationFeatures2D.shape[0]):
        valGuess = 0.0
        distance = []
        for j in range(trainingFeatures2D.shape[0]):
            distance.append(np.sqrt(
                          (trainingFeatures2D[j, 0] - validationFeatures2D[i, 0]) ** 2.0
                        + (trainingFeatures2D[j, 1] - validationFeatures2D[i, 1]) ** 2.0))
        smallestN = nsmallest(n_neighbors, distance)
        temp = []
        for j in range(n_neighbors):
            temp.append(trainingLabels[distance.index(smallestN[j])])
            valGuess = max(set(temp), key = temp.count)
        if(valGuess == validationLabels[i]):
            accuracy += 1.0

    accuracy /= len(validationFeatures2D)
    accuracy *= 100.0
    return accuracy

for n in range(1, 6):
    print('accuracy = ', __kNNTest(trainingFeatures2D, trainingLabels, n, validationFeatures2D,
```

c)

Præcision af forudsigelserne fra KNN test.

$$\begin{aligned}
\text{accuracy}_{k=1} &= 93.10344827586206\% \\
\text{accuracy}_{k=2} &= 96.55172413793103\% \\
\text{accuracy}_{k=3} &= 100.0\% \\
\text{accuracy}_{k=4} &= 100.0\% \\
\text{accuracy}_{k=5} &= 100.0\%
\end{aligned}$$

Jeg ville vælge $k = 3$, da der er færreste nærmeste naboer, men som stadig giver 100% præcision på vores data. $k = 4$ og $k = 5$ kan også bruges.