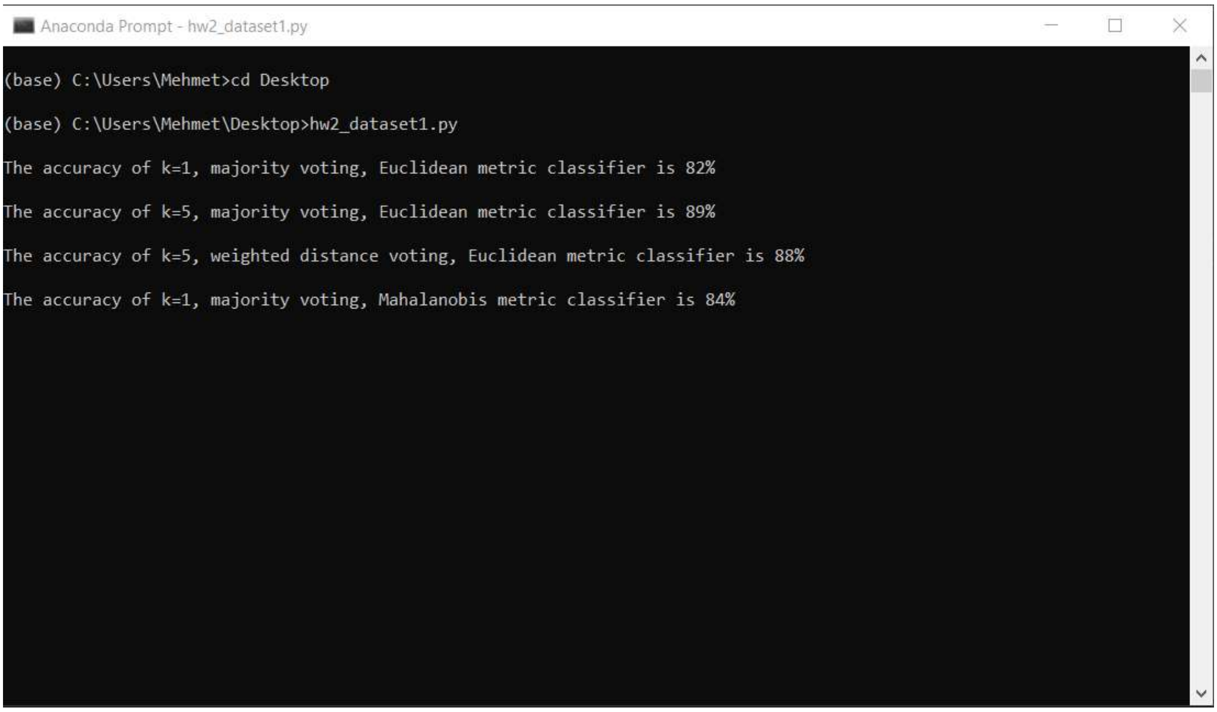
Pattern Recognition Methods
and
Introduction to Machine Learning

Homework 2 - Report
Playing with k-NN

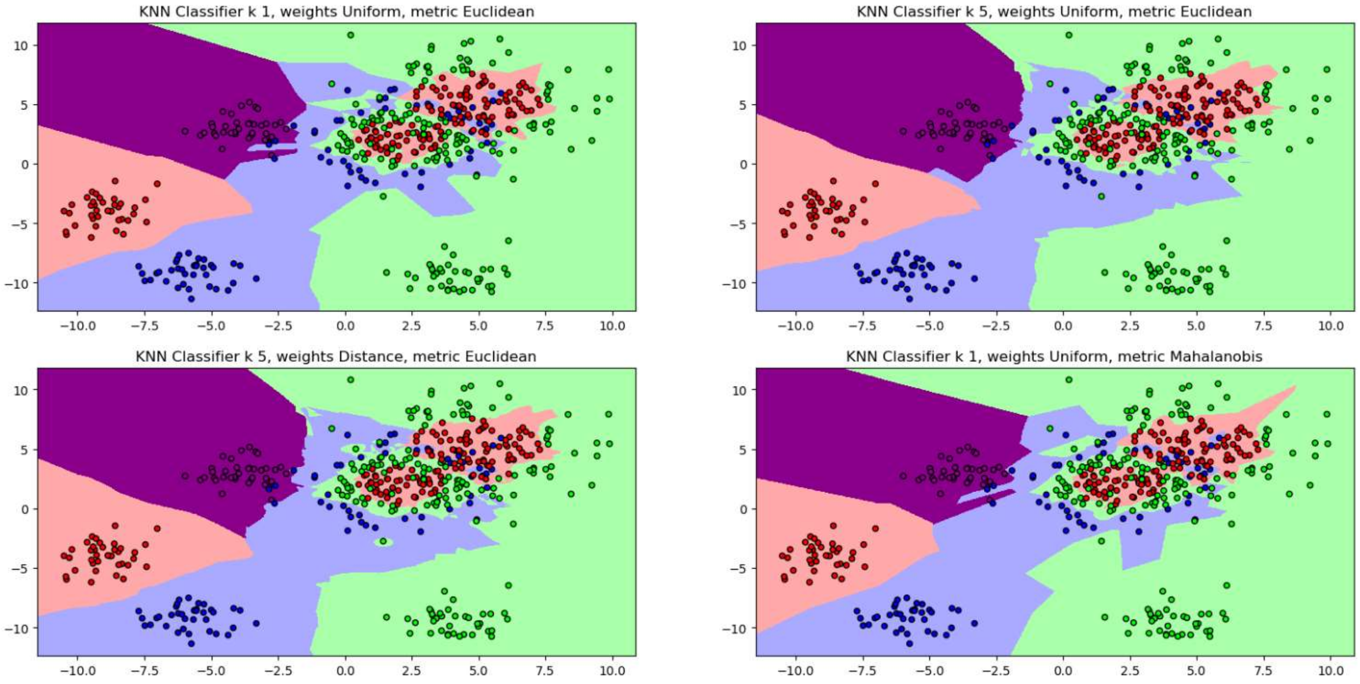
Prepared by: Mehmet Kapson

9.04.2019

Results for the first dataset are below:



```
(base) C:\Users\Mehmet>cd Desktop

(base) C:\Users\Mehmet\Desktop>hw2_dataset1.py

The accuracy of k=1, majority voting, Euclidean metric classifier is 82%

The accuracy of k=5, majority voting, Euclidean metric classifier is 89%

The accuracy of k=5, weighted distance voting, Euclidean metric classifier is 88%

The accuracy of k=1, majority voting, Mahalanobis metric classifier is 84%
```


KNN Classifier k 1, weights Uniform, metric Euclidean

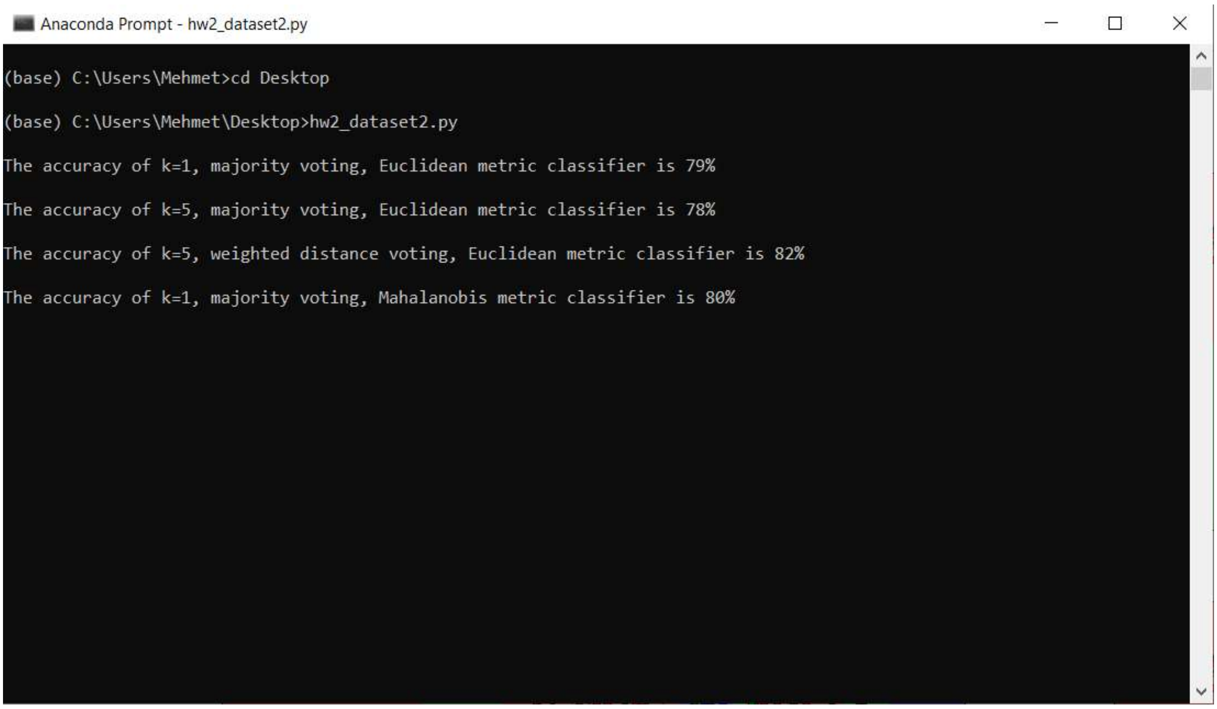
KNN Classifier k 5, weights Uniform, metric Euclidean

KNN Classifier k 5, weights Distance, metric Euclidean

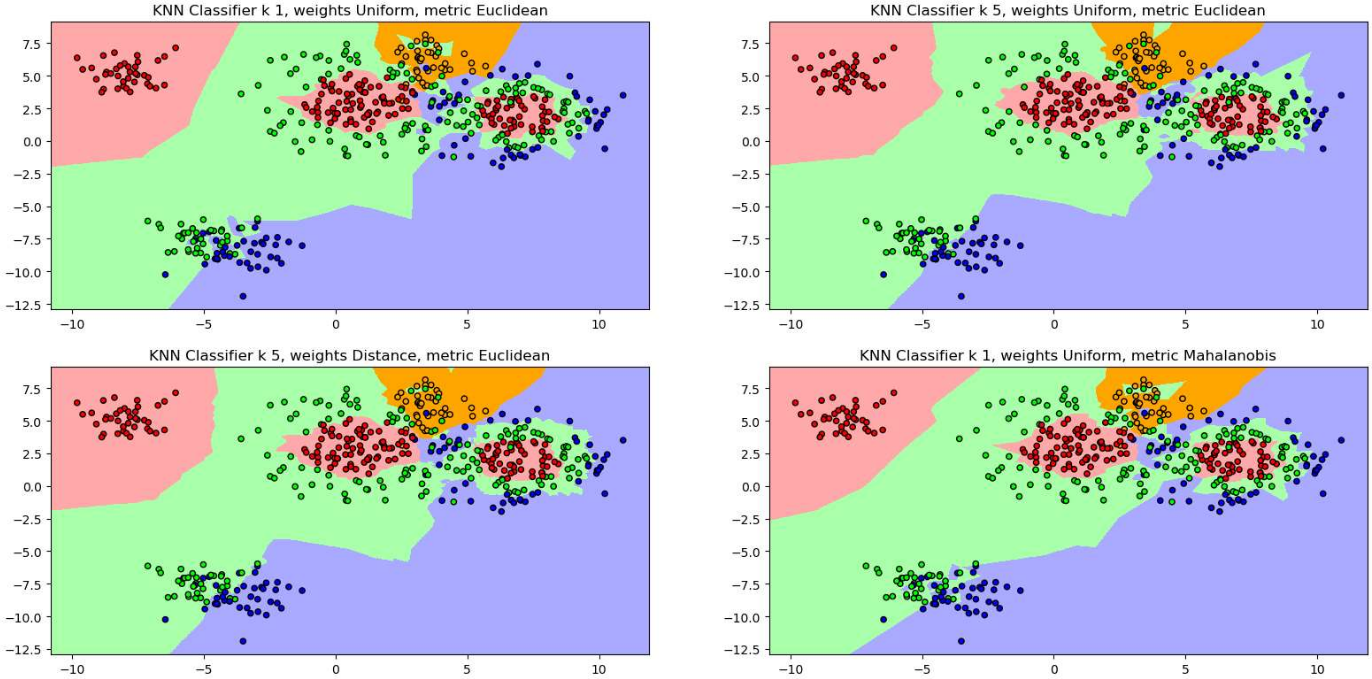
KNN Classifier k 1, weights Uniform, metric Mahalanobis

Just increasing k from 1 to 5, increased the accuracy of the k-NN classifier from 82% to 89%. For small k values, the classification results are sensitive to the data sparseness and the noisy points. That is the reason for this increase. While k=1, just changing Euclidean metric to Mahalanobis metric, increased the accuracy of the k-NN classifier from 82% to 84%. It is because of the Mahalanobis metric taking into account the correlations of the data set. While k=5, just changing weight Uniform to Distance, decreased the accuracy of the k-NN classifier from 89% to 88%.

Results for the second dataset are below:

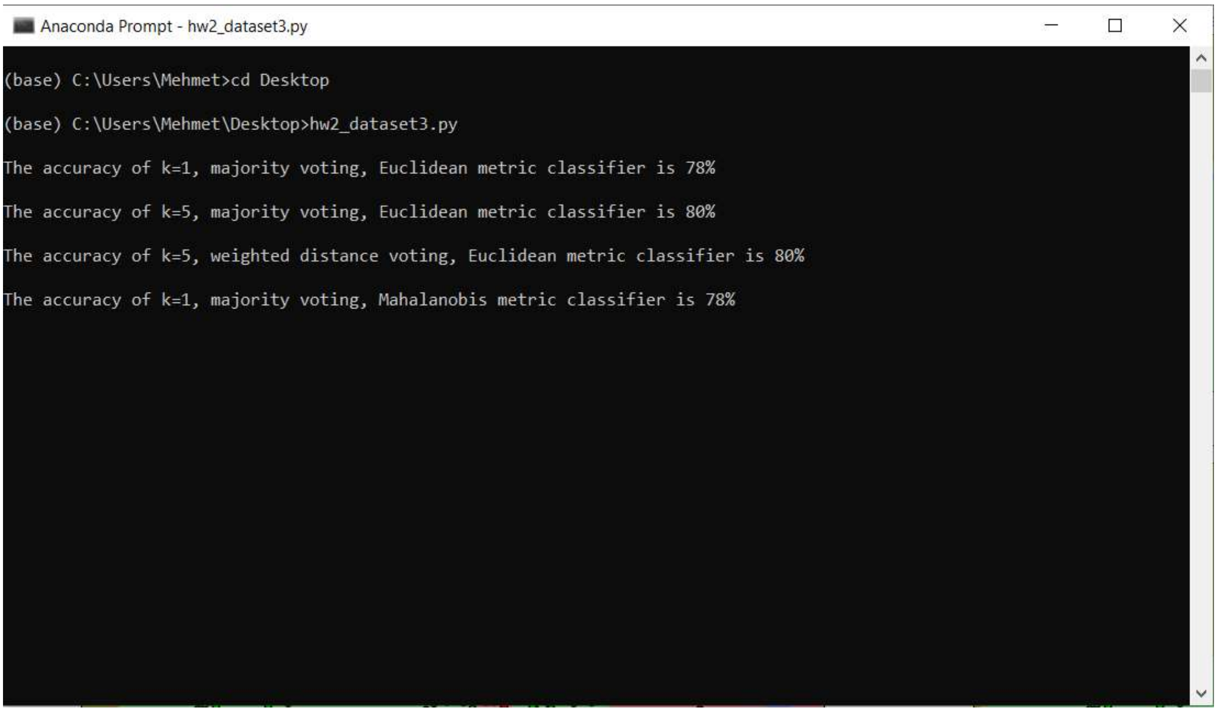


```
(base) C:\Users\Mehmet>cd Desktop

(base) C:\Users\Mehmet\Desktop>hw2_dataset2.py

The accuracy of k=1, majority voting, Euclidean metric classifier is 79%

The accuracy of k=5, majority voting, Euclidean metric classifier is 78%

The accuracy of k=5, weighted distance voting, Euclidean metric classifier is 82%

The accuracy of k=1, majority voting, Mahalanobis metric classifier is 80%
```

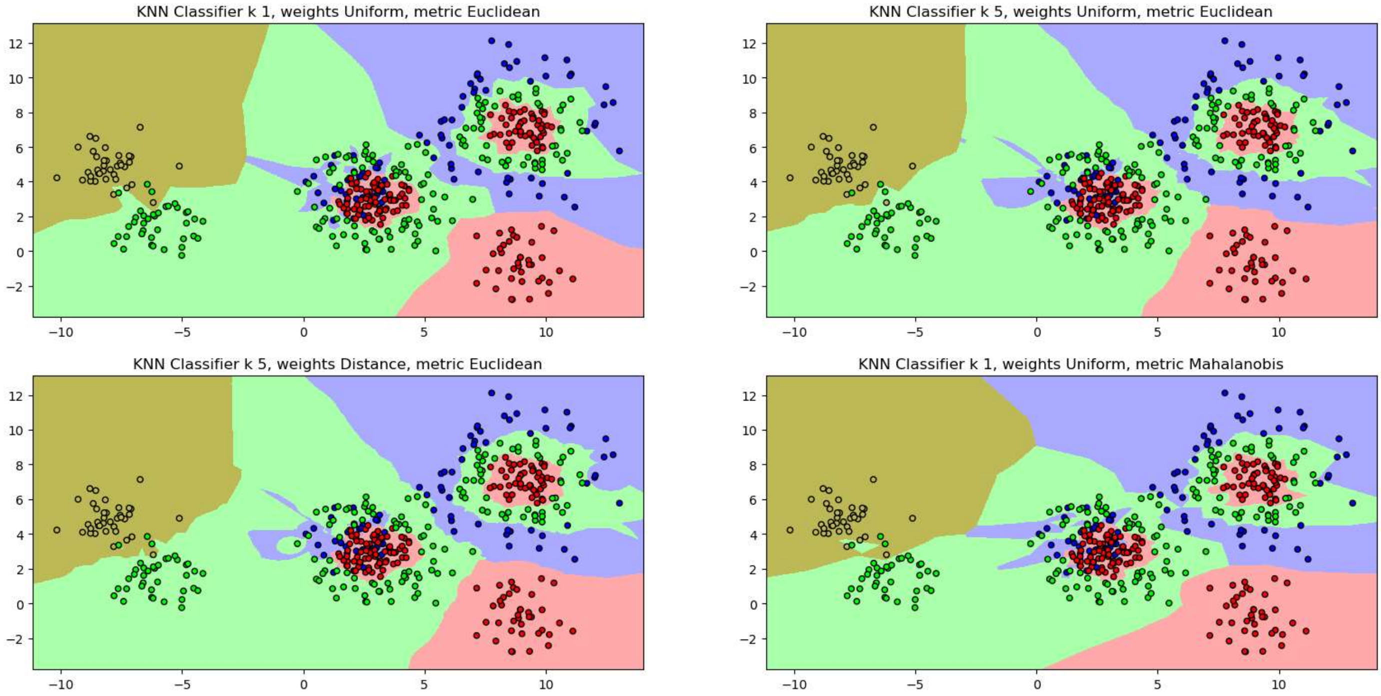


As you can see, data set 2 is more complex than data set 1. Thus; the accuracies of the classifiers are lower. Just increasing k from 1 to 5, decreased the accuracy of the k-NN classifier from 79% to 78%. This happened because of outliers. While k=1, just changing Euclidean metric to Mahalanobis metric, increased the accuracy of the k-NN classifier from 79% to 80%. It is because of the Mahalanobis metric taking into account the correlations of the data set. While k=5, just changing weight Uniform to Distance, increased the accuracy of the k-NN classifier from 78% to 82%.

Results for the third dataset are below:



```
(base) C:\Users\Mehmet>cd Desktop

(base) C:\Users\Mehmet\Desktop>hw2_dataset3.py

The accuracy of k=1, majority voting, Euclidean metric classifier is 78%

The accuracy of k=5, majority voting, Euclidean metric classifier is 80%

The accuracy of k=5, weighted distance voting, Euclidean metric classifier is 80%

The accuracy of k=1, majority voting, Mahalanobis metric classifier is 78%
```


KNN Classifier k 1, weights Uniform, metric Euclidean

KNN Classifier k 5, weights Uniform, metric Euclidean

KNN Classifier k 5, weights Distance, metric Euclidean

KNN Classifier k 1, weights Uniform, metric Mahalanobis

In data set 3, the accuracies of the classifiers are even lower than data set 2. Just increasing k from 1 to 5, increased the accuracy of the k-NN classifier from 78% to 80%. It is the same situation which happened in data set 1. While k=1, just changing Euclidean metric to Mahalanobis metric, did not change the accuracy of the k-NN classifier. While k=5, just changing weight Uniform to Distance, did not change the accuracy of the k-NN classifier in this data set.