# HACETTEPE UNIVERSITY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

## ELE489-FUNDAMENTALS OF MACHINE LEARNING

## HOMEWORK I

**BURAK ÇAYIRLI 2200357009**

Spring 2024-2025

# List of Figures

I downloaded Wine Dataset from the provided link and the graphs of some features in the following.

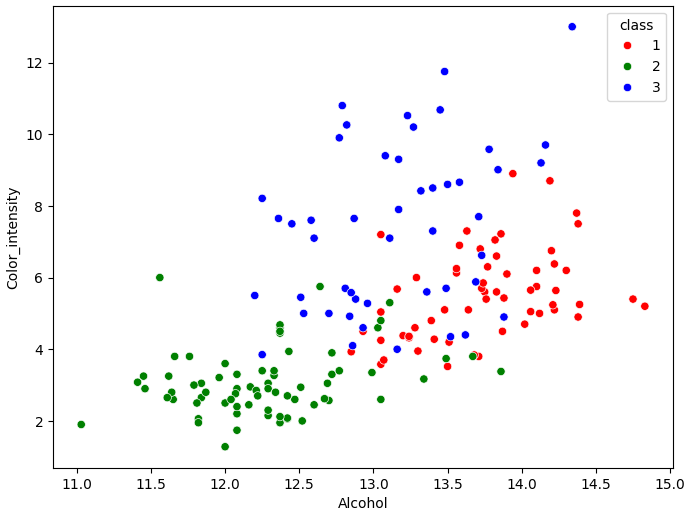


Figure 1: Alcohol vs Color Intensity Graph

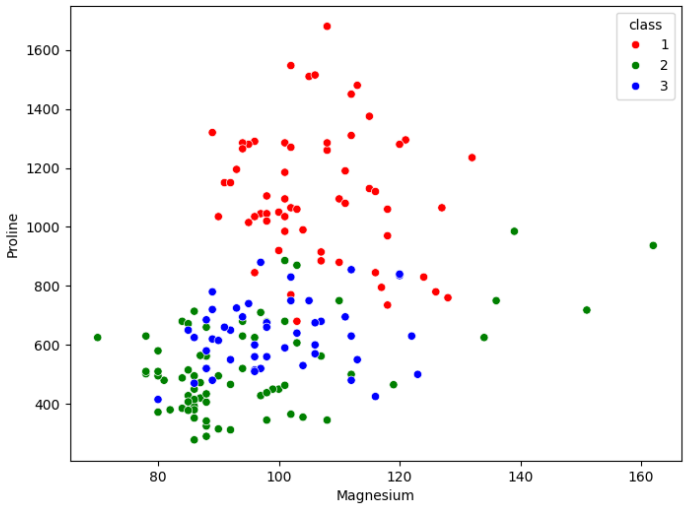


Figure 2: Magnesium vs Proline Graph

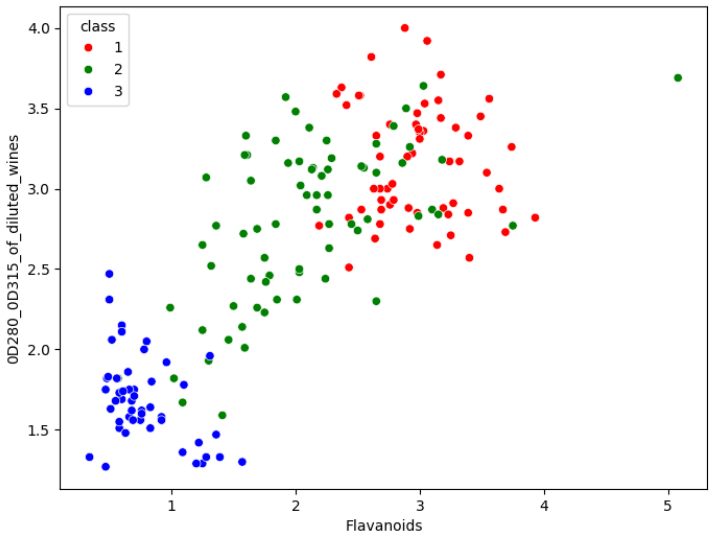


Figure 3: Flavanoids vs 0D280-0D315 of diluted wines Graph

Before processing the data, I normalized the data by using 'MinMaxScaler' function of 'sklearn'. I normalized the values between 0 and 5. After that, I split the data into four - 'X train','X test','Y train','Y test'-.Then I run my KNN code that is provided in my Github Repository. In my KNN code, I used two different distance metrics,Euclidean Distance and Manhattan Distance.Classification accuracy results for each distance metric in the following.Also results before normalization and after normalization in the following.

```
Accuracy for K=1: 0.7777777777777778
Accuracy for K=3: 0.7777777777777778
Accuracy for K=5: 0.8333333333333334
Accuracy for K=7: 0.7777777777777778
Accuracy for K=9: 0.75
Accuracy for K=11: 0.75
Accuracy for K=13: 0.75
Accuracy for K=15: 0.7222222222222222
Accuracy for K=17: 0.7777777777777778
Accuracy for K=19: 0.7777777777777778
Accuracy for K=21: 0.7777777777777778
Accuracy for K=23: 0.75
Accuracy for K=25: 0.75
Accuracy for K=27: 0.75
Accuracy for K=29: 0.75
```

Figure 4: Classification Accuracy Table for Euclidean Distance Before Normalization

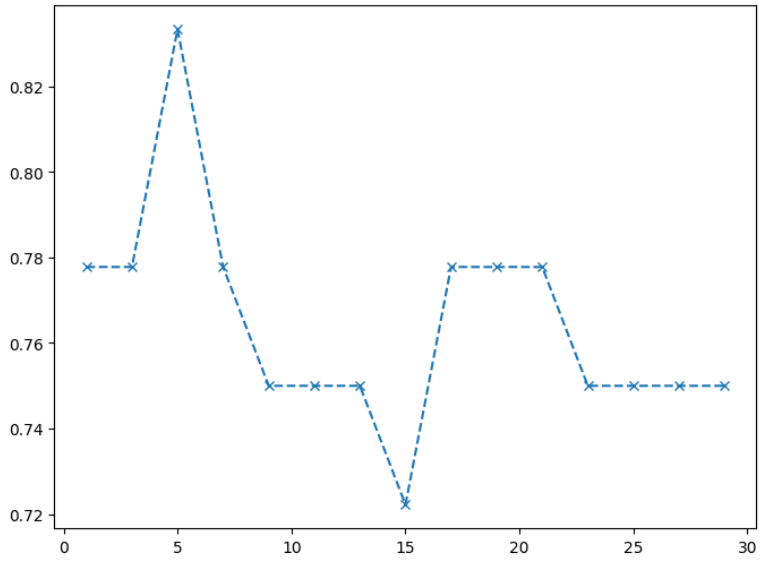


Figure 5: Classification Accuracy Graph for Euclidean Distance Before Normalization

```
Accuracy for K=1: 0.9166666666666666
Accuracy for K=3: 0.9722222222222222
Accuracy for K=5: 0.9722222222222222
Accuracy for K=7: 0.9444444444444444
Accuracy for K=9: 0.9444444444444444
Accuracy for K=11: 0.9444444444444444
Accuracy for K=13: 0.9722222222222222
Accuracy for K=15: 0.9444444444444444
Accuracy for K=17: 0.9444444444444444
Accuracy for K=19: 0.9722222222222222
Accuracy for K=21: 0.9722222222222222
Accuracy for K=23: 0.9722222222222222
Accuracy for K=25: 0.9444444444444444
Accuracy for K=27: 0.9166666666666666
Accuracy for K=29: 0.9166666666666666
```

Figure 6: Classification Accuracy Table for Euclidean Distance After Normalization

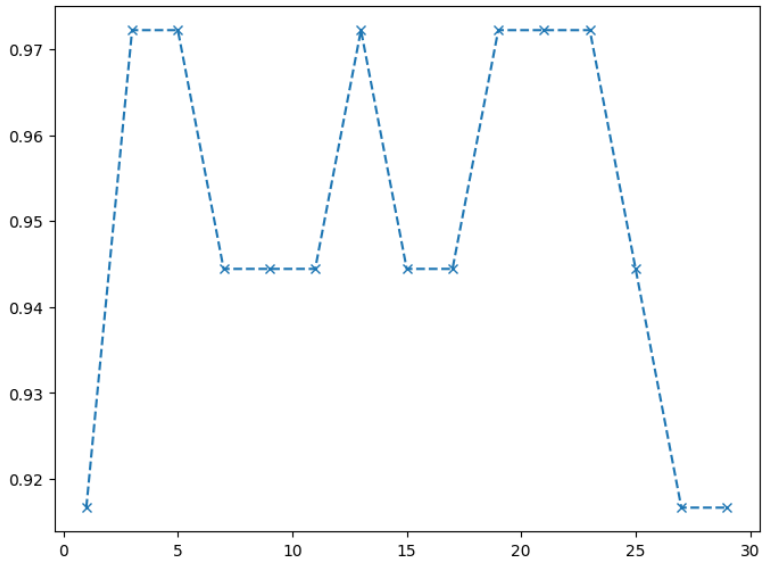


Figure 7: Classification Accuracy Graph for Euclidean Distance After Normalization

```
Accuracy for K=1: 0.8333333333333334
Accuracy for K=3: 0.8333333333333334
Accuracy for K=5: 0.8333333333333334
Accuracy for K=7: 0.7777777777777778
Accuracy for K=9: 0.7777777777777778
Accuracy for K=11: 0.7777777777777778
Accuracy for K=13: 0.75
Accuracy for K=15: 0.75
Accuracy for K=17: 0.75
Accuracy for K=19: 0.75
Accuracy for K=21: 0.75
Accuracy for K=23: 0.75
Accuracy for K=25: 0.7777777777777778
Accuracy for K=27: 0.7777777777777778
Accuracy for K=29: 0.75
```

Figure 8: Classification Accuracy Table for Manhattan Distance Before Normalization

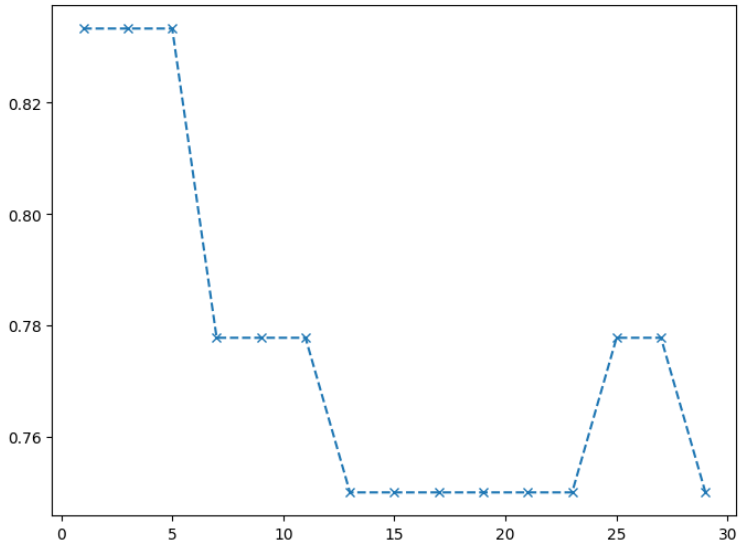


Figure 9: Classification Accuracy Graph for Manhattan Distance Before Normalization

```
Accuracy for K=1: 0.9166666666666666
Accuracy for K=3: 0.9444444444444444
Accuracy for K=5: 0.9722222222222222
Accuracy for K=7: 0.9722222222222222
Accuracy for K=9: 0.9722222222222222
Accuracy for K=11: 0.9444444444444444
Accuracy for K=13: 0.9722222222222222
Accuracy for K=15: 0.9444444444444444
Accuracy for K=17: 0.9444444444444444
Accuracy for K=19: 0.9444444444444444
Accuracy for K=21: 0.9444444444444444
Accuracy for K=23: 0.9444444444444444
Accuracy for K=25: 0.9444444444444444
Accuracy for K=27: 0.9444444444444444
Accuracy for K=29: 0.9444444444444444
```

Figure 10: Classification Accuracy Table for Manhattan Distance After Normalization

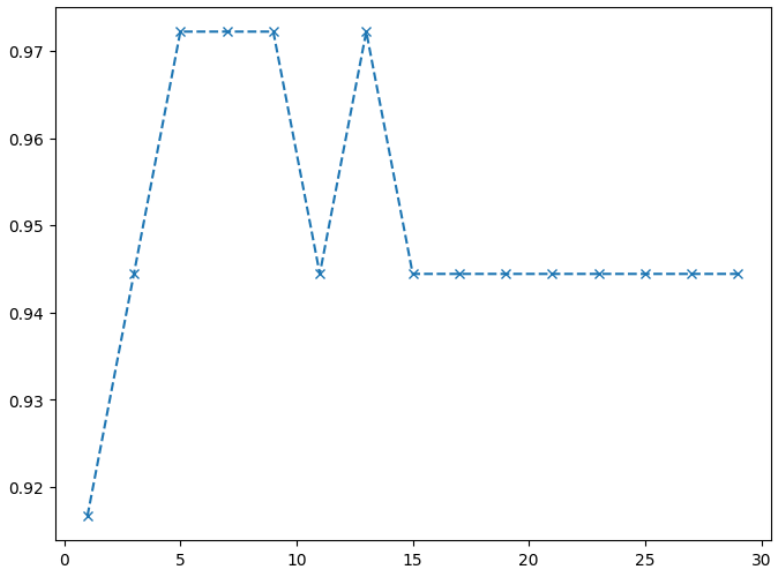


Figure 11: Classification Accuracy Graph for Manhattan Distance After Normalization

Before analyzing the impact of K, I wanted to examine the effect of preprocessing. Initially, the accuracy values were very low due to the lack of normalization. Since unnormalized data gives higher weight to features with larger scales,some important features for distinguishing the class became less significant because of their low distances. As a result, the model performed poorly and produced low accuracy scores. However, after normalization the accuracy scores significantly increased regardless of the K value. Now, it is time to analyze the effect of the K values. I used odd K values to ensure test data has one class label, and I checked accuracy values for odd numbers between 1 and 29. I saw that the value of K affects accuracy scores significantly as you can see from the Figure 7 and Figure 11. While using Euclidean Distance after K=23 accuracy scores significantly dropped. For Manhattan Distance after K=15 accuracy scores dropped and remained same. This shows us after a certain value algorithm does not work properly it works on big data and cannot make an accurate predictions. To get rid of this I choose my K value as 5 according to the graphs. I created my confusion matrix and clasififcation report according to this K value.

```
Confusion Matrix
[[14  0  0]
 [ 0 15  1]
 [ 0  0  6]]
Classificiton Report
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        14
           2       1.00      0.94      0.97        16
           3       0.86      1.00      0.92         6

    accuracy                           0.97        36
   macro avg       0.95      0.98      0.96        36
weighted avg       0.98      0.97      0.97        36
```

Figure 12: Confusion Matrix and Classfication Repoort

My Github Link Below

`https://github.com/burakcayirli/ELE-489_Homework_1`