# Hacettepe University

# Department of Electrical and Electronics Engineering

# ELE 489: Fundamentals of Machine Learning

# HW-1 Report

Kıvanç Ateş

2210357113

To gain insight about the data, I used some functions. I observed first 5 column of data with using "df.head()".

| | alcohol | malic_acid | ash | alcalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_intensity | hue | od280/od315_of_diluted_wines | proline | class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 14.23 | 1.71 | 2.43 | 15.6 | 127.0 | 2.80 | 3.06 | 0.28 | 2.29 | 5.64 | 1.04 | 3.92 | 1065.0 | 0.0 |
| 1 | 13.20 | 1.78 | 2.14 | 11.2 | 100.0 | 2.65 | 2.76 | 0.26 | 1.28 | 4.38 | 1.05 | 3.40 | 1050.0 | 0.0 |
| 2 | 13.16 | 2.36 | 2.67 | 18.6 | 101.0 | 2.80 | 3.24 | 0.30 | 2.81 | 5.68 | 1.03 | 3.17 | 1185.0 | 0.0 |
| 3 | 14.37 | 1.95 | 2.50 | 16.8 | 113.0 | 3.85 | 3.49 | 0.24 | 2.18 | 7.80 | 0.86 | 3.45 | 1480.0 | 0.0 |
| 4 | 13.24 | 2.59 | 2.87 | 21.0 | 118.0 | 2.80 | 2.69 | 0.39 | 1.82 | 4.32 | 1.04 | 2.93 | 735.0 | 0.0 |

Figure 1

I observed the dimensions about the data as (178,14) with using "df.shape". The shape (178,14) means that my data set has 178 samples in the dataset and this dataset includes 13 features and 1 class label. To generate statistic information about data set I used "df.describe()" function.

| | alcohol | malic_acid | ash | alcalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_intensity | hue | od280/od315_of_diluted_wines | proline | class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 | 178.000000 |
| mean | 13.000618 | 2.336348 | 2.366517 | 19.494944 | 99.741573 | 2.295112 | 2.029270 | 0.361854 | 1.590899 | 5.058090 | 0.957449 | 2.611685 | 746.893258 | 1.938202 |
| std | 0.811827 | 1.117146 | 0.274344 | 3.339564 | 14.282484 | 0.625851 | 0.998859 | 0.124453 | 0.572359 | 2.318286 | 0.228572 | 0.709990 | 314.907474 | 0.775035 |
| min | 11.030000 | 0.740000 | 1.360000 | 10.600000 | 70.000000 | 0.980000 | 0.340000 | 0.130000 | 0.410000 | 1.280000 | 0.480000 | 1.270000 | 278.000000 | 1.000000 |
| 25% | 12.362500 | 1.602500 | 2.210000 | 17.200000 | 88.000000 | 1.742500 | 1.205000 | 0.270000 | 1.250000 | 3.220000 | 0.782500 | 1.937500 | 500.500000 | 1.000000 |
| 50% | 13.050000 | 1.865000 | 2.360000 | 19.500000 | 98.000000 | 2.355000 | 2.135000 | 0.340000 | 1.555000 | 4.690000 | 0.965000 | 2.780000 | 673.500000 | 2.000000 |
| 75% | 13.677500 | 3.082500 | 2.557500 | 21.500000 | 107.000000 | 2.800000 | 2.875000 | 0.437500 | 1.950000 | 6.200000 | 1.120000 | 3.170000 | 985.000000 | 3.000000 |
| max | 14.830000 | 5.800000 | 3.230000 | 30.000000 | 162.000000 | 3.880000 | 5.080000 | 0.660000 | 3.580000 | 13.000000 | 1.710000 | 4.000000 | 1680.000000 | 3.000000 |

Figure 2

I use KDE plotting for each feature to check class overlap and visualize the features. It helps us understand the distribution and density of each class according to the given feature. If the curves overlap each other too much, we can say that there is overlap for this feature. In other words, this feature should not be preferred in distinguishing classes. I also tried to observe whether the inferences I made from the KDE drawings were correct by using boxplots. If the boxplots do not overlap each other, this feature is suitable for classification.

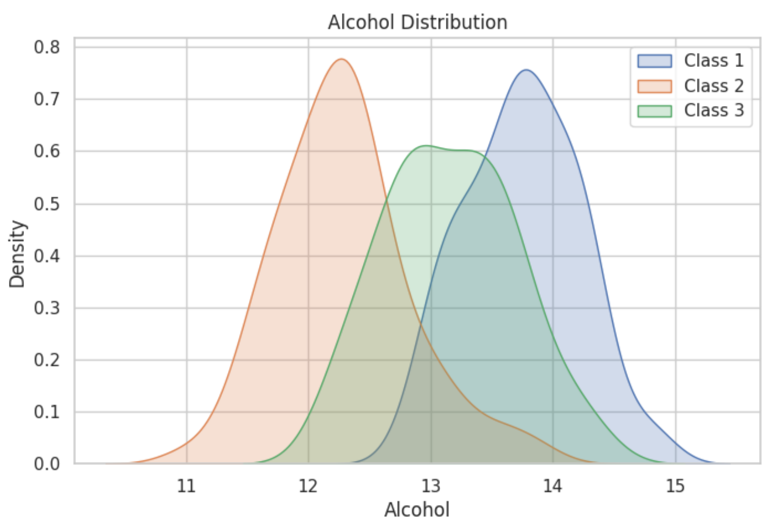



Figure 3

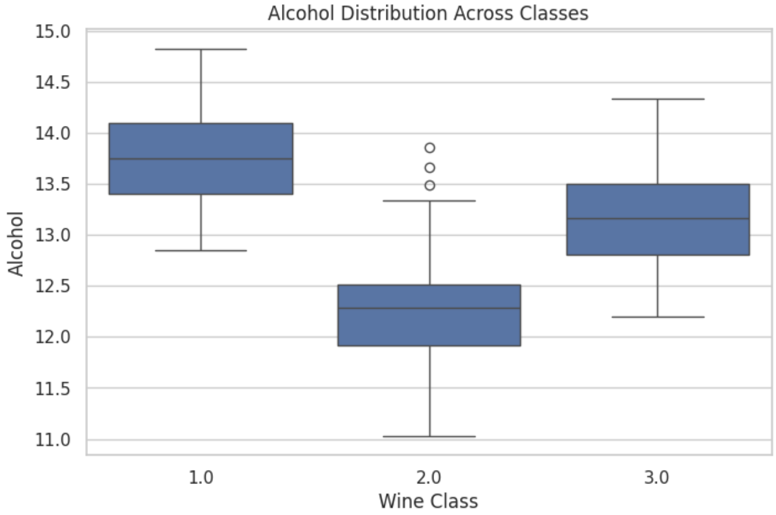



Figure 4

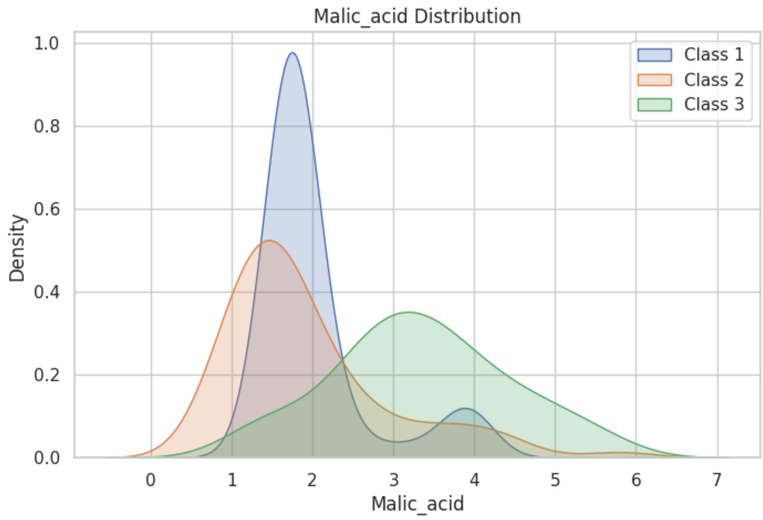



Figure 5

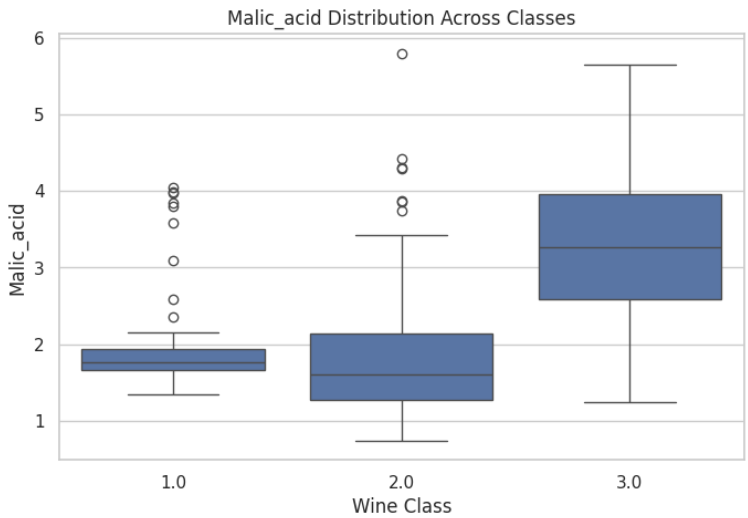



Figure 6

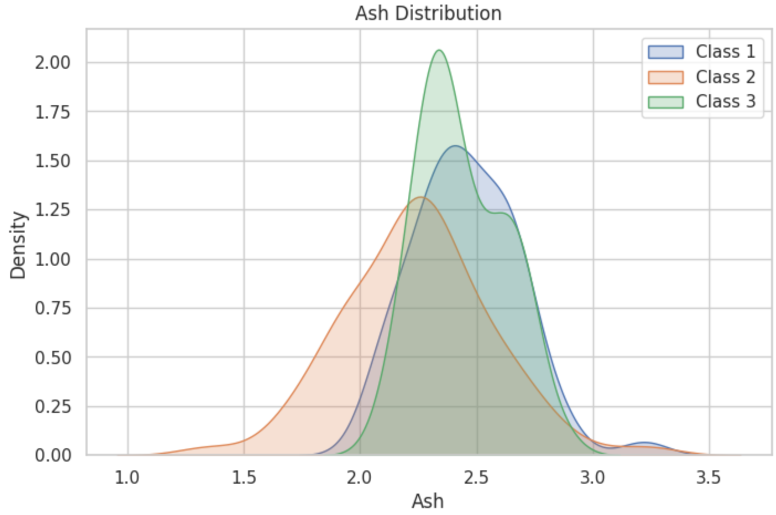



Figure 7

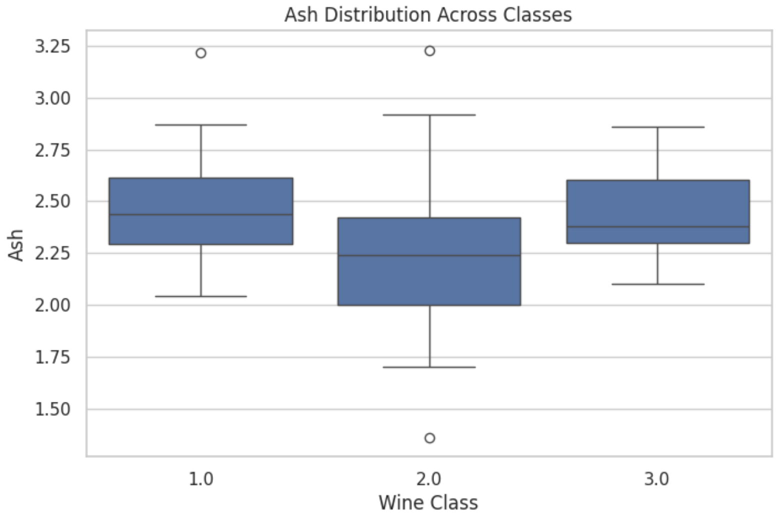



Figure 8

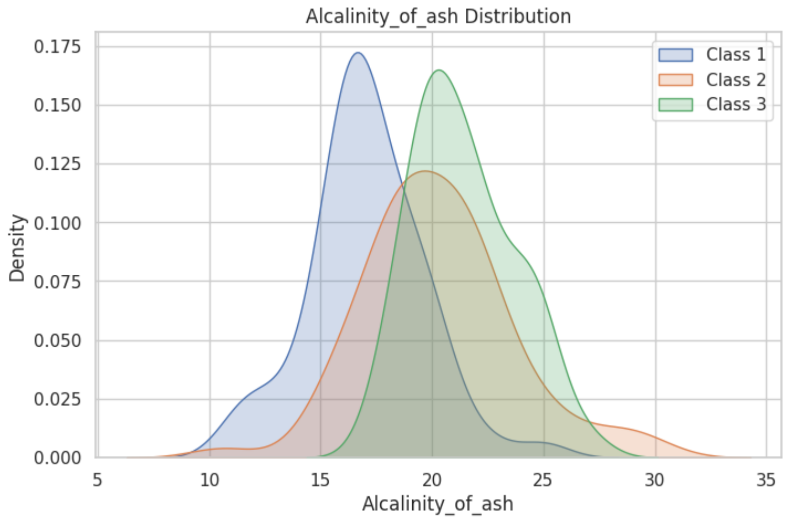



Figure 9

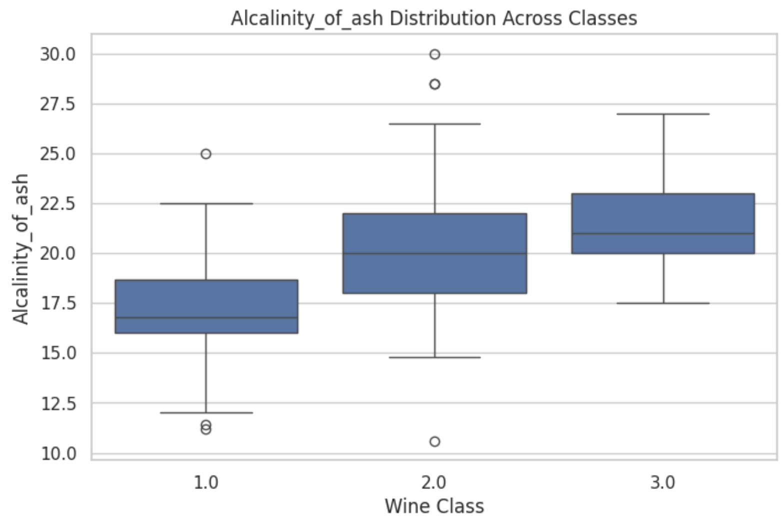



Figure 10

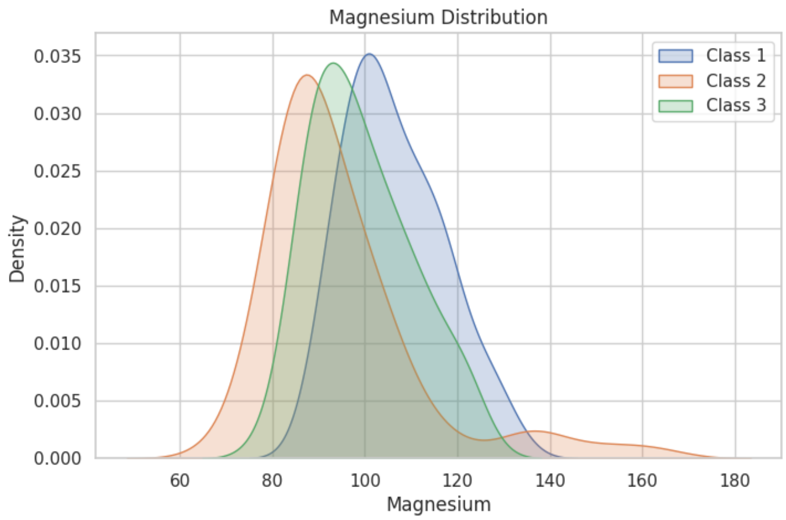



Figure 11

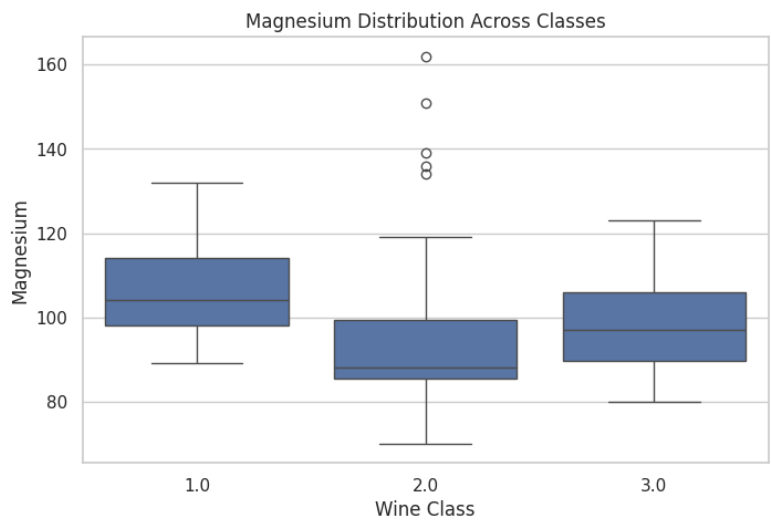



Figure 12

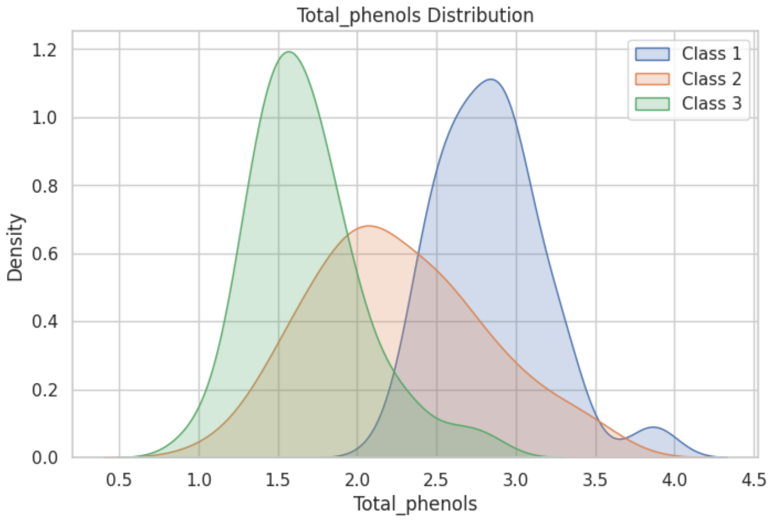



Figure 13

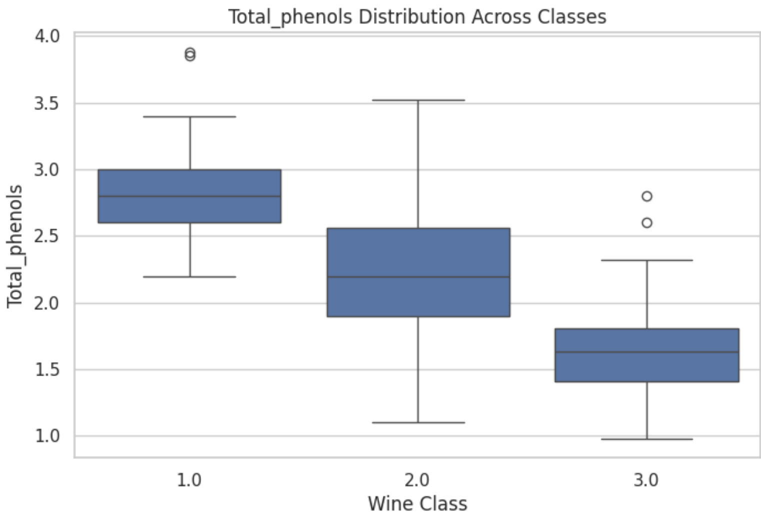



Figure 14

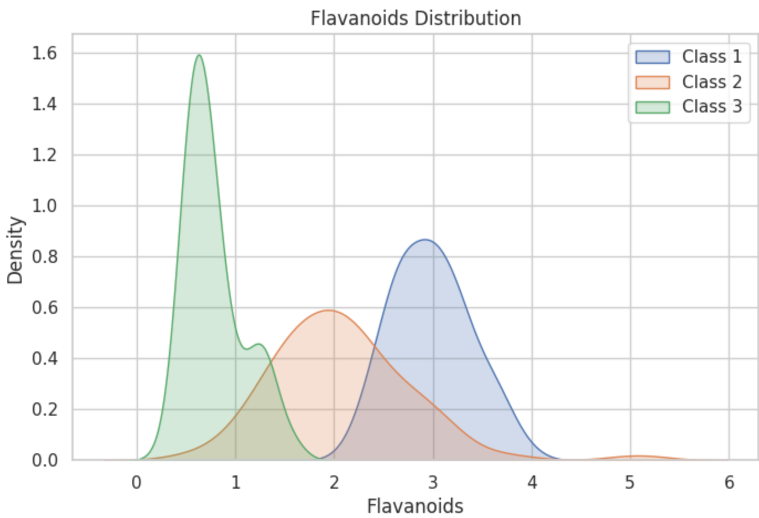



Figure 15

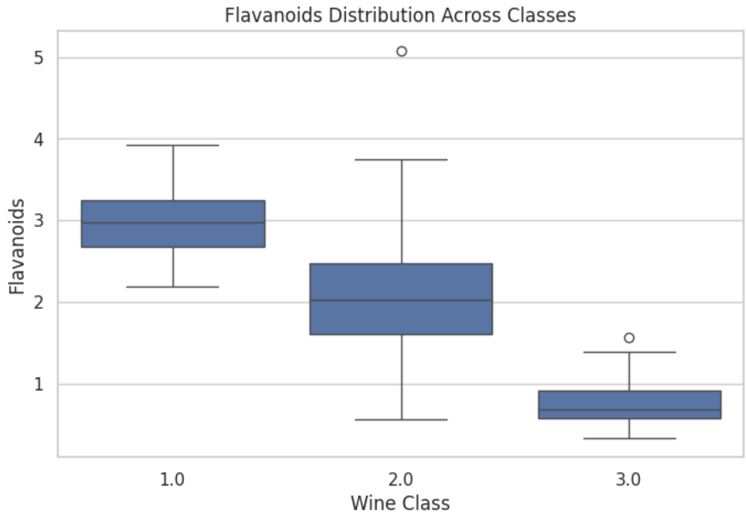



Figure 16

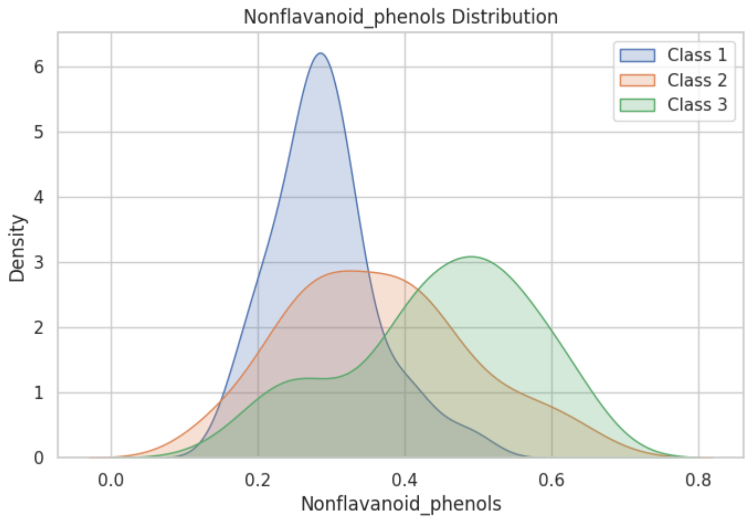



Figure 17

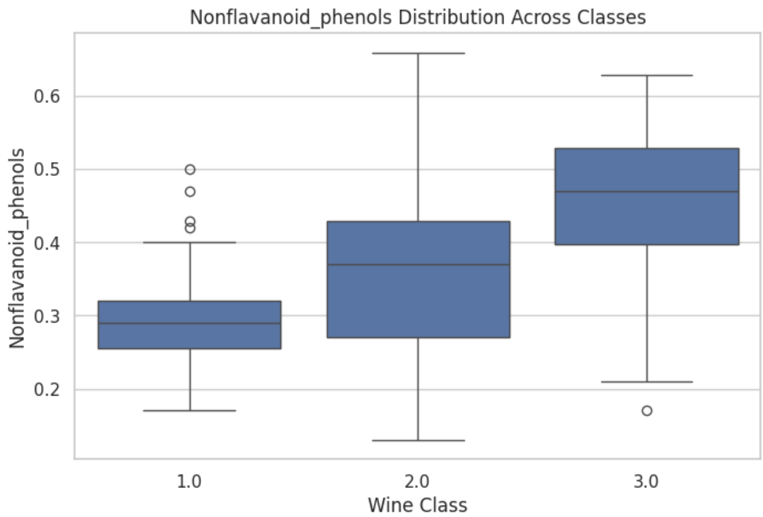



Figure 18

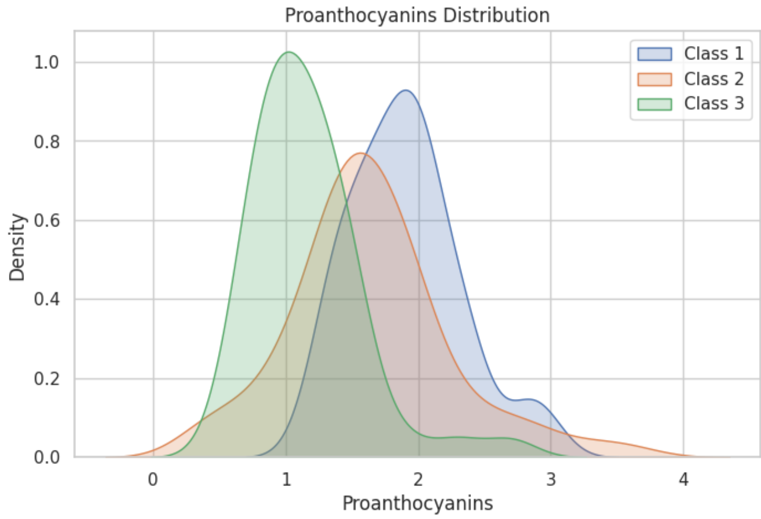



Figure 19

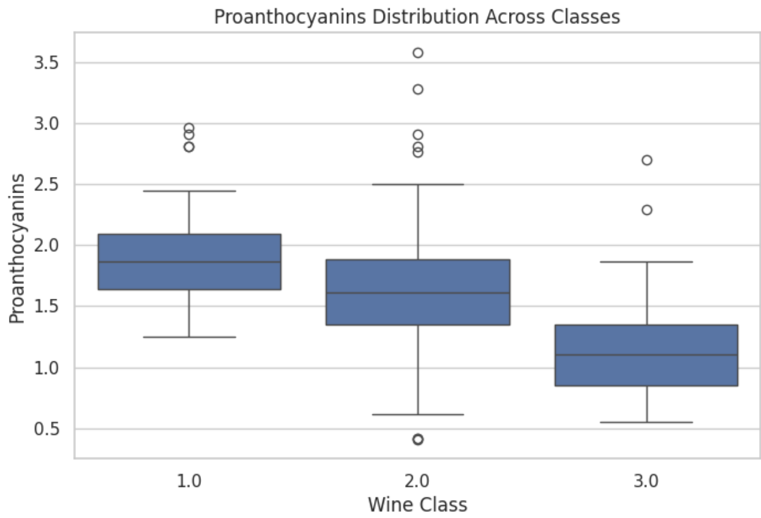



Figure 20

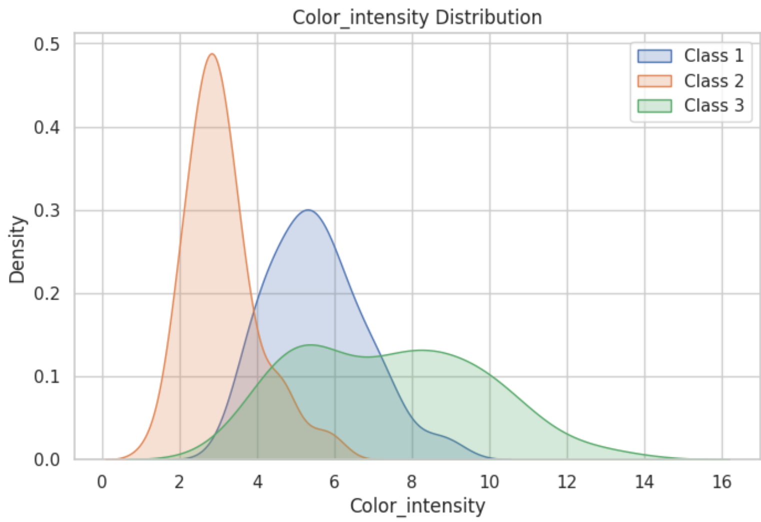



Figure 21

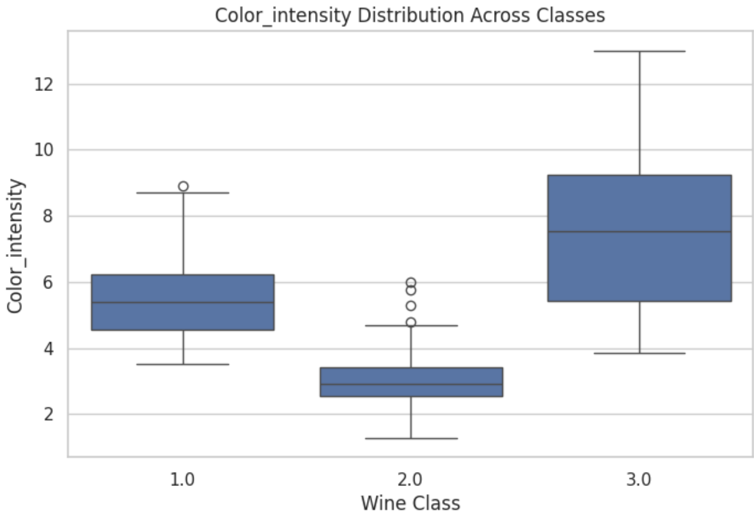



Figure 22

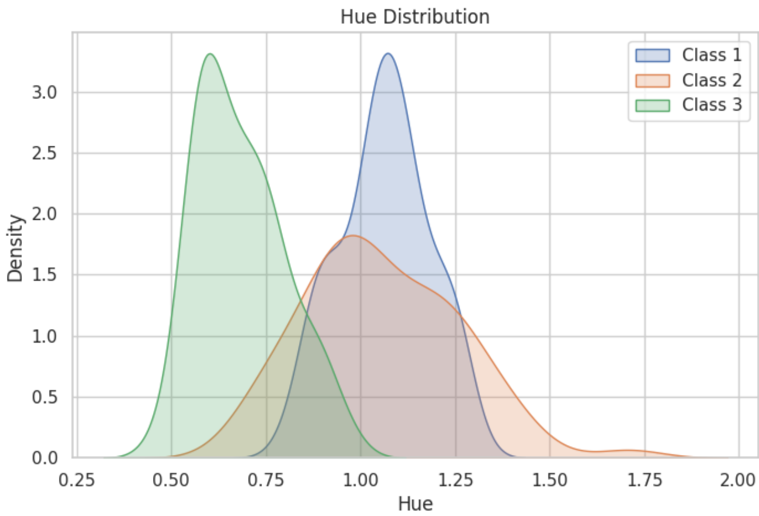



Figure 23

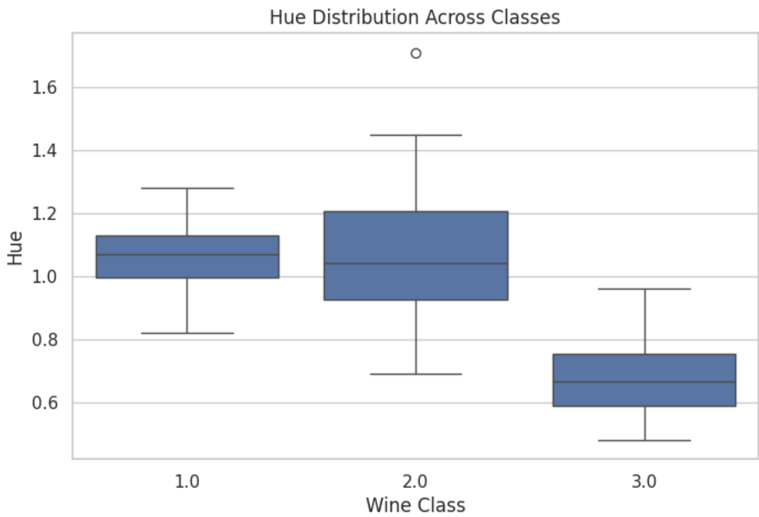



Figure 24

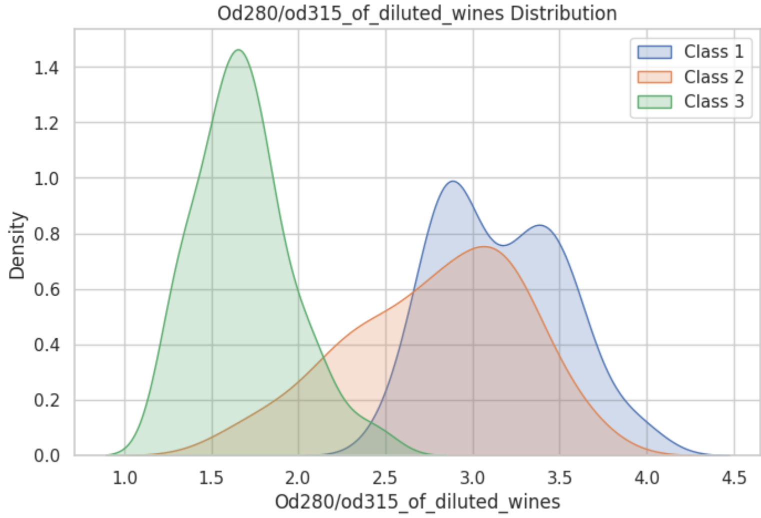



Figure 25

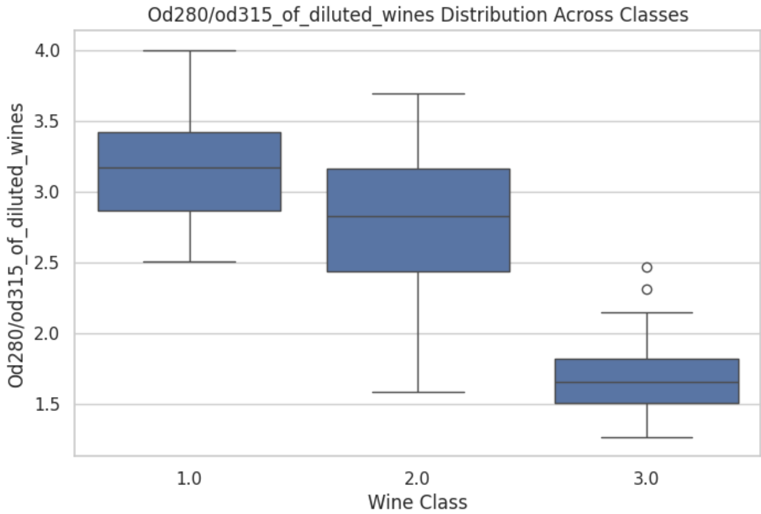



Figure 26

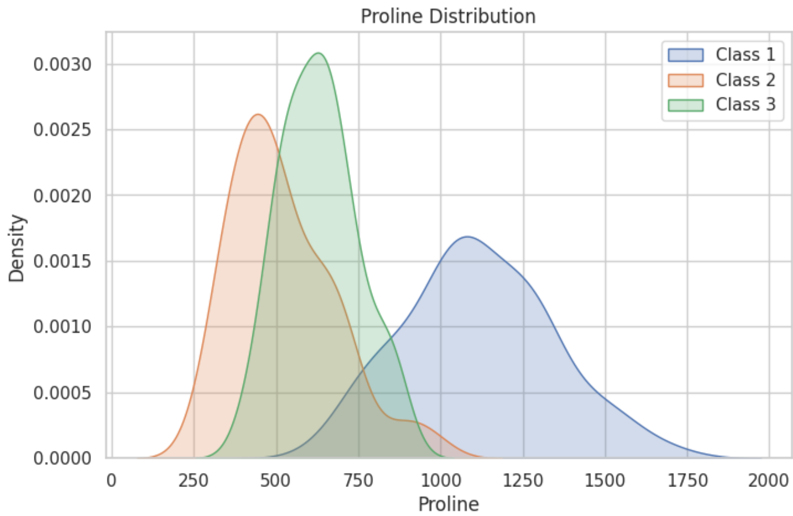



Figure 27

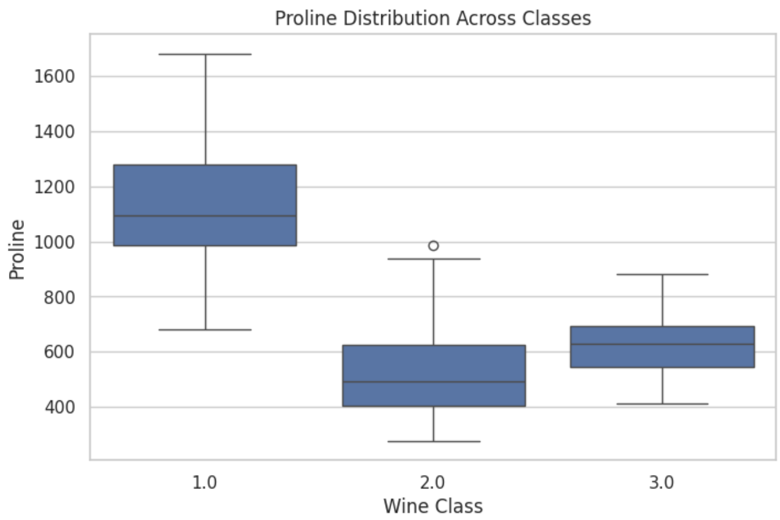



Figure 28

When we examine the Overlap status of the features, we see that the features that will give the most accurate results in class determination are Alcohol Distribution, Total Phenols Distribution, Flavonoids Distribution. The features that may give us incorrect results in determining the class are Ash Distribution, Alkalinity of Ash Distribution, Hue Distribution. We should not select features that contain overlap when determining the class.

As seen in Figure 29, the dataset does not contain missing data.

```
alcohol                         0
malic_acid                      0
ash                             0
alcalinity_of_ash               0
magnesium                       0
total_phenols                   0
flavanoids                      0
nonflavanoid_phenols            0
proanthocyanins                 0
color_intensity                 0
hue                             0
od280/od315_of_diluted_wines    0
proline                         0
class                           0
dtype: int64
```

Figure 29

I separated the data so that the features in the "x" variable and the class in the "y" variable. And I separated these variables as "X_train", "X_test", "y_train", "y_test". I observed the dimension of "X_train" as (142, 13), "X_test" as (36, 13), "y_train" as (142,), "y_test" as (36,).

For k-NN Standardization is usually better than Normalization. Therefore, I used Standardization. Standardization makes the mean 0 and standard deviation 1, which prevents large-scale features from dominating small-scale ones.

I tested the k-NN algorithm using different k and different distance finding methods. I first did what was requested using Euclidean Distance.

```
K=1 → Accuracy: 0.9167
K=3 → Accuracy: 0.9444
K=5 → Accuracy: 0.9722
K=7 → Accuracy: 1.0000
K=9 → Accuracy: 1.0000
K=11 → Accuracy: 1.0000
K=13 → Accuracy: 1.0000
```

Figure 30

Accuracy is shown in Figure 30 for different “K” values. When I increase the “K”, I observe the accuracy goes to one. So, we can say that when we increase the “K” value, accuracy increases. Accuracy plot with respect to “K” is shown in Figure 31.

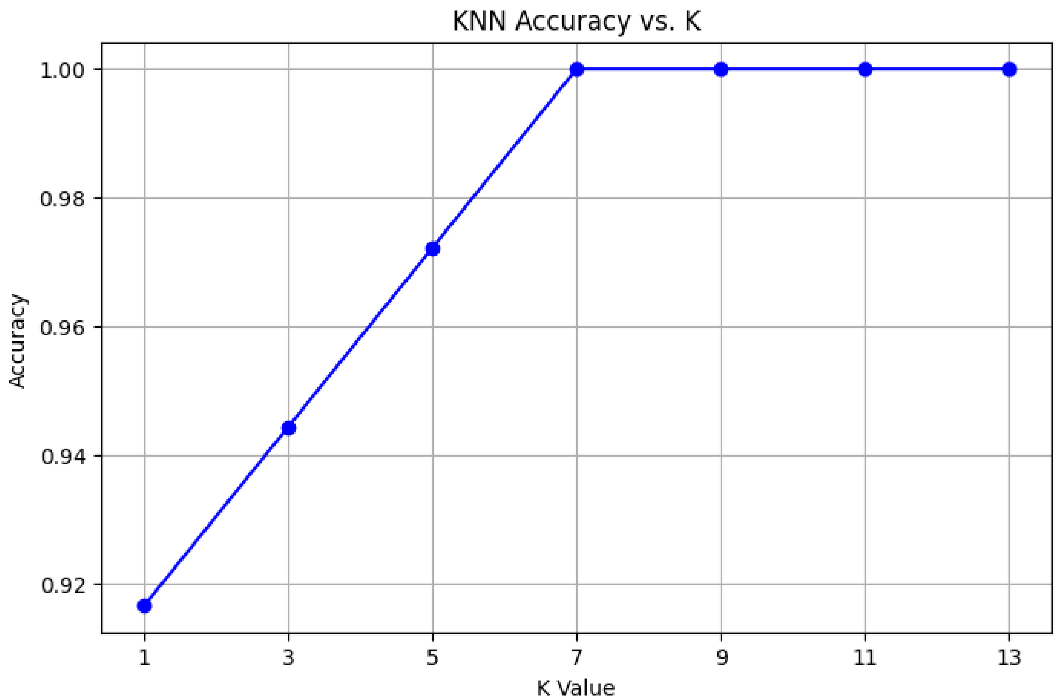
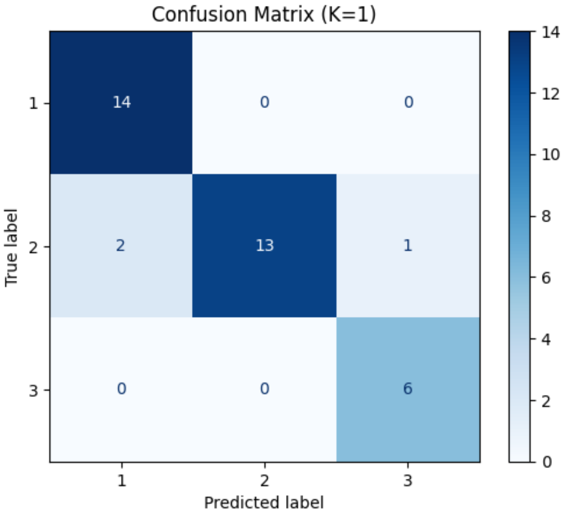
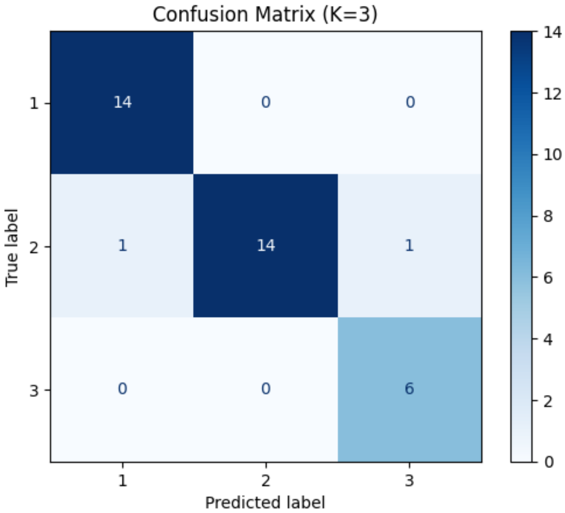
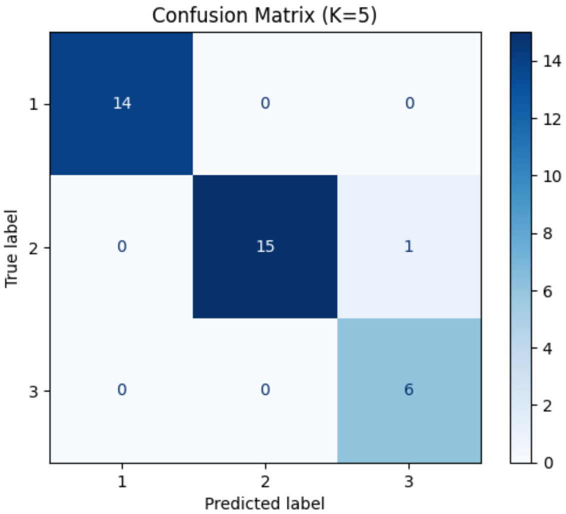


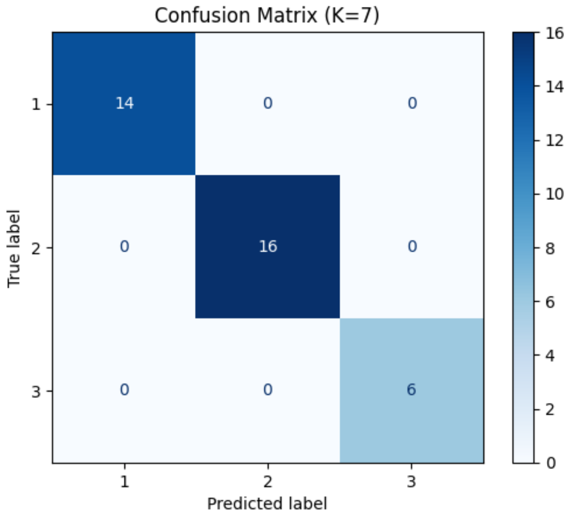

Figure 31

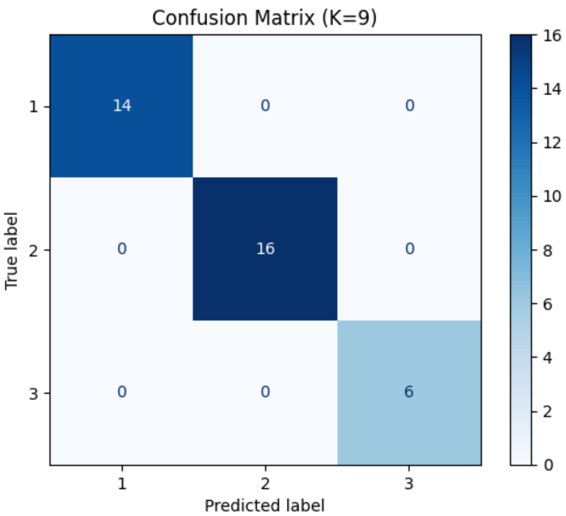
When value of “K” becomes 7, accuracy stays at 1. Confusion Matrix plots are shown below for different “K” values.

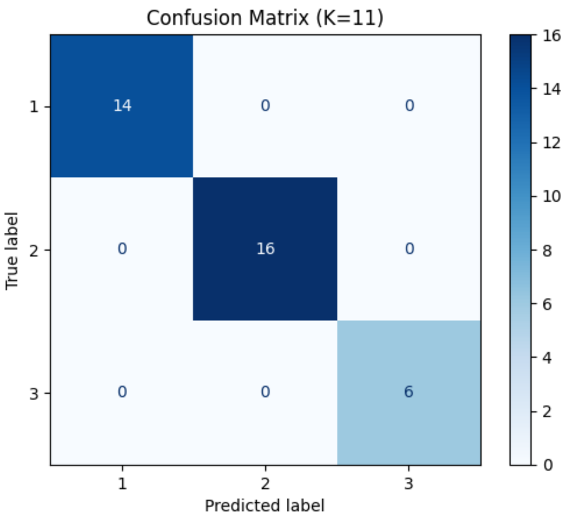

Confusion Matrix (K=1)
14 0 0
2 13 1
0 0 6
True label
Predicted label


Confusion Matrix (K=3)
14 0 0
1 14 1
0 0 6
True label
Predicted label


Confusion Matrix (K=5)
14 0 0
0 15 1
0 0 6
True label
Predicted label


Confusion Matrix (K=7)
14 0 0
0 16 0
0 0 6
True label
Predicted label


Confusion Matrix (K=9)
14 0 0
0 16 0
0 0 6
True label
Predicted label


Confusion Matrix (K=11)
14 0 0
0 16 0
0 0 6
True label
Predicted label

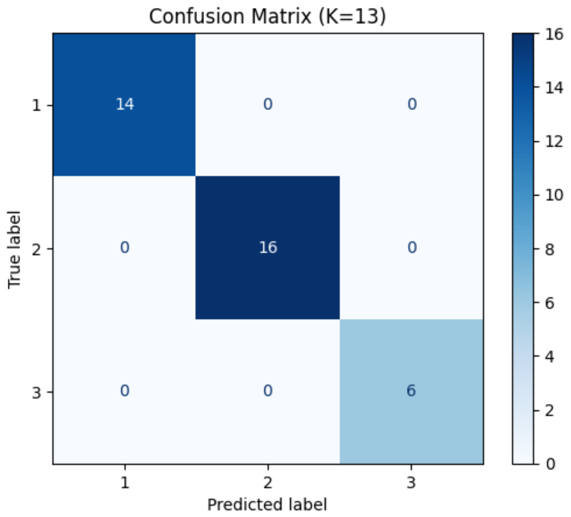



Figure 32

When we examine the Confusion Matrices, it is observed that all the predictions made from K = 7 onwards overlap with the real classes. In other words, all the test data we process are classified correctly.

Precision Reports are shown below.

K=1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.88 | 1.00 | 0.93 | 14 |
| Class 2 | 1.00 | 0.81 | 0.90 | 16 |
| Class 3 | 0.86 | 1.00 | 0.92 | 6 |
| accuracy | | | 0.92 | 36 |
| macro avg | 0.91 | 0.94 | 0.92 | 36 |
| weighted avg | 0.93 | 0.92 | 0.92 | 36 |

K=3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.88 | 0.93 | 16 |
| Class 3 | 0.86 | 1.00 | 0.92 | 6 |
| accuracy | | | 0.94 | 36 |
| macro avg | 0.93 | 0.96 | 0.94 | 36 |
| weighted avg | 0.95 | 0.94 | 0.94 | 36 |

K=5

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 0.94 | 0.97 | 16 |
| Class 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 |

K=7

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 1.00 | 1.00 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 1.00 | 36 |
| macro avg | 1.00 | 1.00 | 1.00 | 36 |
| weighted avg | 1.00 | 1.00 | 1.00 | 36 |

K=9

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 1.00 | 1.00 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 1.00 | 36 |
| macro avg | 1.00 | 1.00 | 1.00 | 36 |
| weighted avg | 1.00 | 1.00 | 1.00 | 36 |

K=11

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 1.00 | 1.00 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 1.00 | 36 |
| macro avg | 1.00 | 1.00 | 1.00 | 36 |
| weighted avg | 1.00 | 1.00 | 1.00 | 36 |

K=13

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 1.00 | 1.00 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 1.00 | 36 |
| macro avg | 1.00 | 1.00 | 1.00 | 36 |
| weighted avg | 1.00 | 1.00 | 1.00 | 36 |

As we found in the previous results, it can be observed from the "precision" and “recall” columns that when K = 7, all class predictions are correct. For all class predictions are correct, "precision" and “recall” columns values should equal to 1 for all classes.

Secondly, I used the Manhattan Distance method. As in the Euclidean Distance method, in this method, all test data are classified correctly at only K=7. For all other K values, accuracy is different from 1. If Manhattan Distance is to be used for the kNN algorithm, it would be best to choose K as 7 so that all predictions are correct. We can observe this from the Accuracy vs "K" graph, Confusion matrices and Precision Reports.

```
K=1 → Accuracy: 0.9444
K=3 → Accuracy: 0.9722
K=5 → Accuracy: 0.9722
K=7 → Accuracy: 1.0000
K=9 → Accuracy: 0.9722
K=11 → Accuracy: 0.9722
K=13 → Accuracy: 0.9722
```

Figure 33

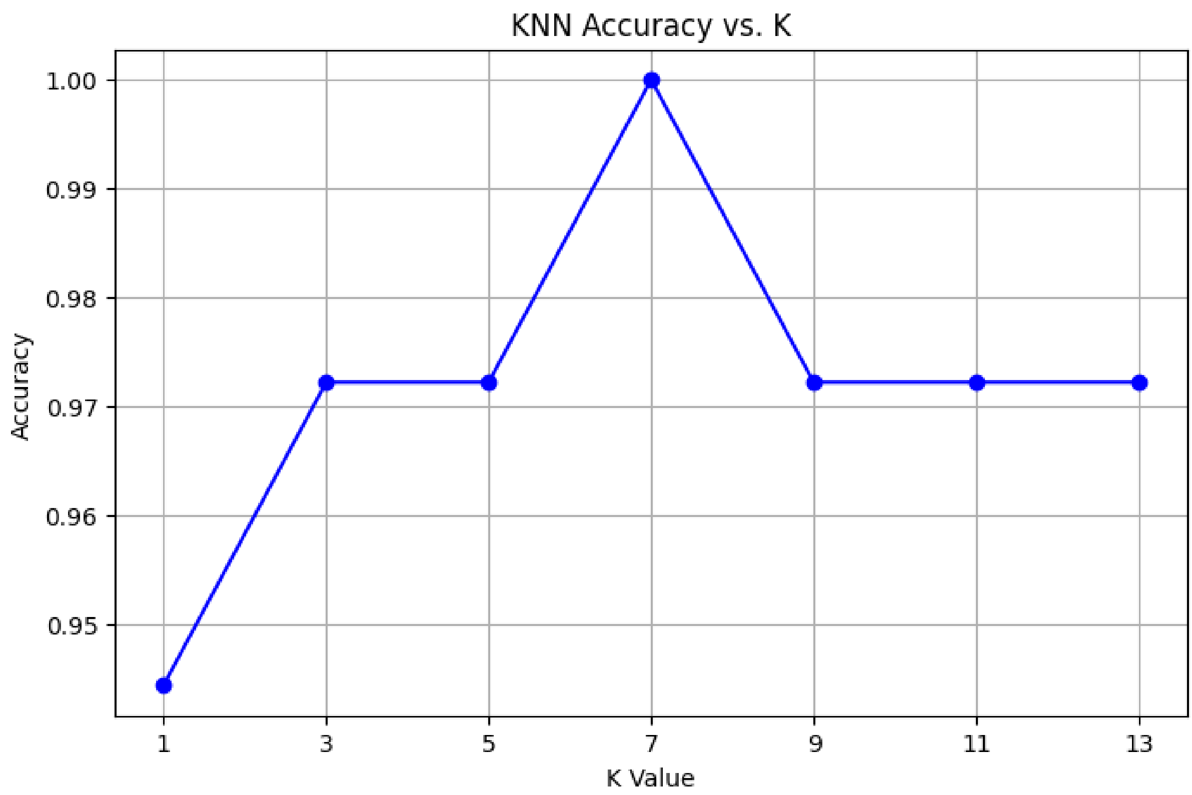
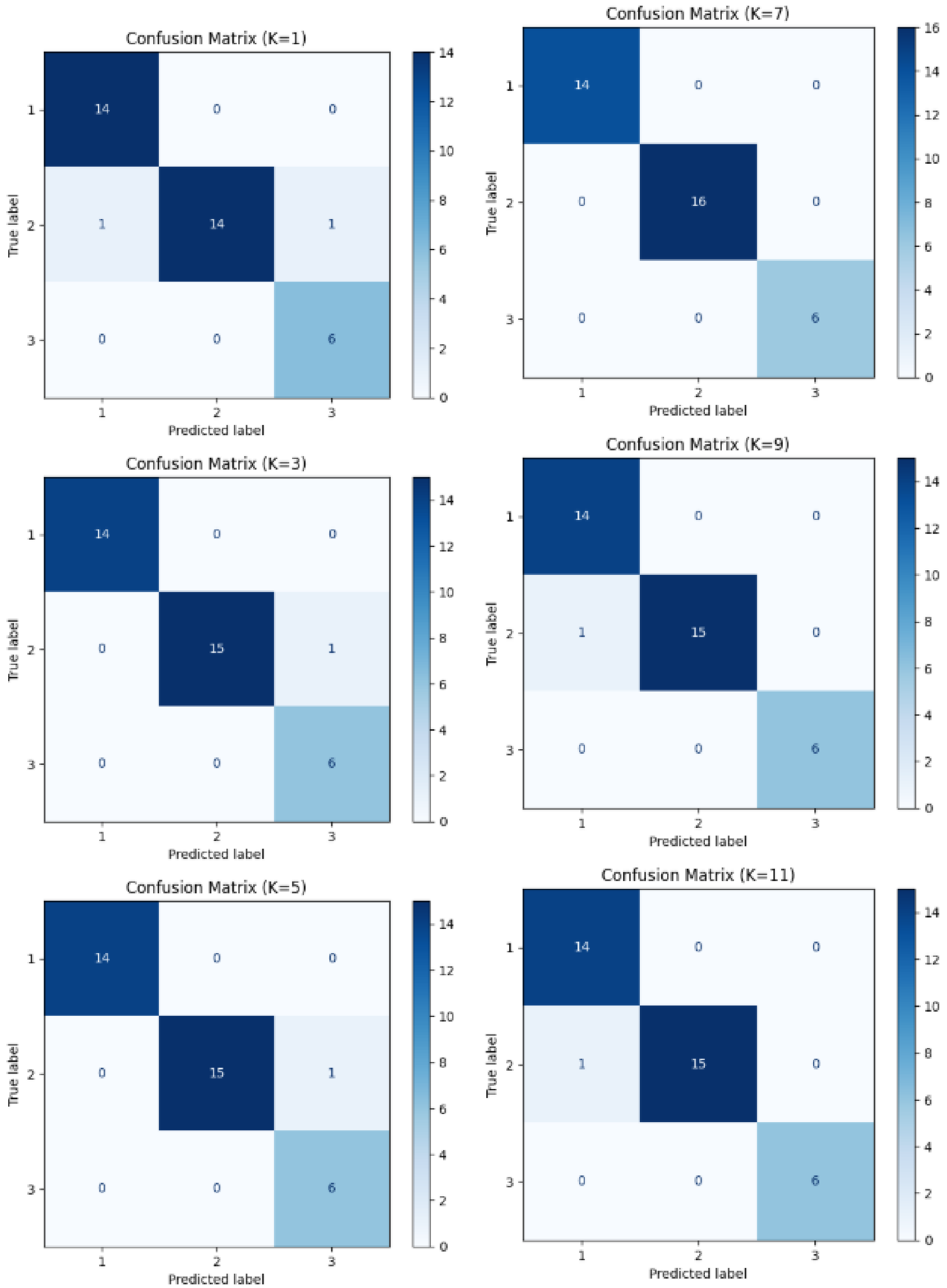



Figure 34


Confusion Matrix (K=1)
True label
Predicted label
14 0 0
1 14 1
0 0 6
Confusion Matrix (K=7)
True label
Predicted label
14 0 0
0 16 0
0 0 6
Confusion Matrix (K=3)
True label
Predicted label
14 0 0
0 15 1
0 0 6
Confusion Matrix (K=9)
True label
Predicted label
14 0 0
1 15 0
0 0 6
Confusion Matrix (K=5)
True label
Predicted label
14 0 0
0 15 1
0 0 6
Confusion Matrix (K=11)
True label
Predicted label
14 0 0
1 15 0
0 0 6

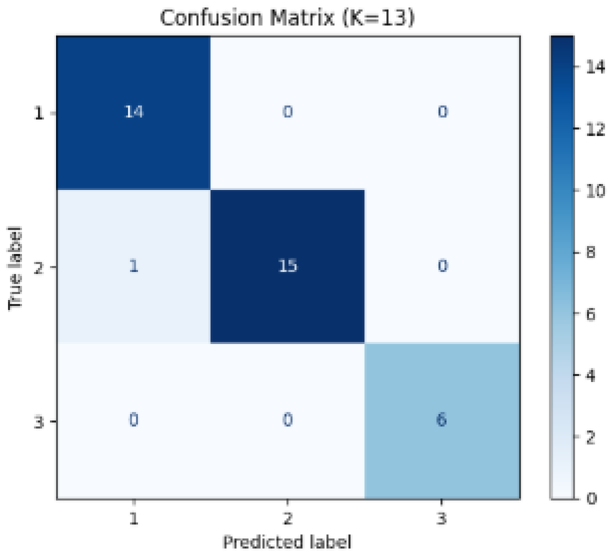



Figure 35

K=1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.88 | 0.93 | 16 |
| Class 3 | 0.86 | 1.00 | 0.92 | 6 |
| accuracy | | | 0.94 | 36 |
| macro avg | 0.93 | 0.96 | 0.94 | 36 |
| weighted avg | 0.95 | 0.94 | 0.94 | 36 |

K=3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 0.94 | 0.97 | 16 |
| Class 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 |

K=5

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 0.94 | 0.97 | 16 |
| Class 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 |

K=7

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 1.00 | 1.00 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 1.00 | 36 |
| macro avg | 1.00 | 1.00 | 1.00 | 36 |
| weighted avg | 1.00 | 1.00 | 1.00 | 36 |

K=9

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.94 | 0.97 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 0.97 | 36 |
| macro avg | 0.98 | 0.98 | 0.98 | 36 |
| weighted avg | 0.97 | 0.97 | 0.97 | 36 |

K=11

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.94 | 0.97 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 0.97 | 36 |
| macro avg | 0.98 | 0.98 | 0.98 | 36 |
| weighted avg | 0.97 | 0.97 | 0.97 | 36 |

K=13

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.94 | 0.97 | 16 |
| Class 3 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 0.97 | 36 |
| macro avg | 0.98 | 0.98 | 0.98 | 36 |
| weighted avg | 0.97 | 0.97 | 0.97 | 36 |

Figure 36

Lastly, I used the Chebyshev Distance method. When Chebyshev Distance is used for the kNN algorithm, it is seen that accuracy for the tested k values is never 1. If Chebyshev Distance is to be used, k = 1,3,5,9 values will give more accurate results than other values.

```
K=1 → Accuracy: 0.9444
K=3 → Accuracy: 0.9444
K=5 → Accuracy: 0.9444
K=7 → Accuracy: 0.9167
K=9 → Accuracy: 0.9444
K=11 → Accuracy: 0.9167
K=13 → Accuracy: 0.9167
```

Figure 37

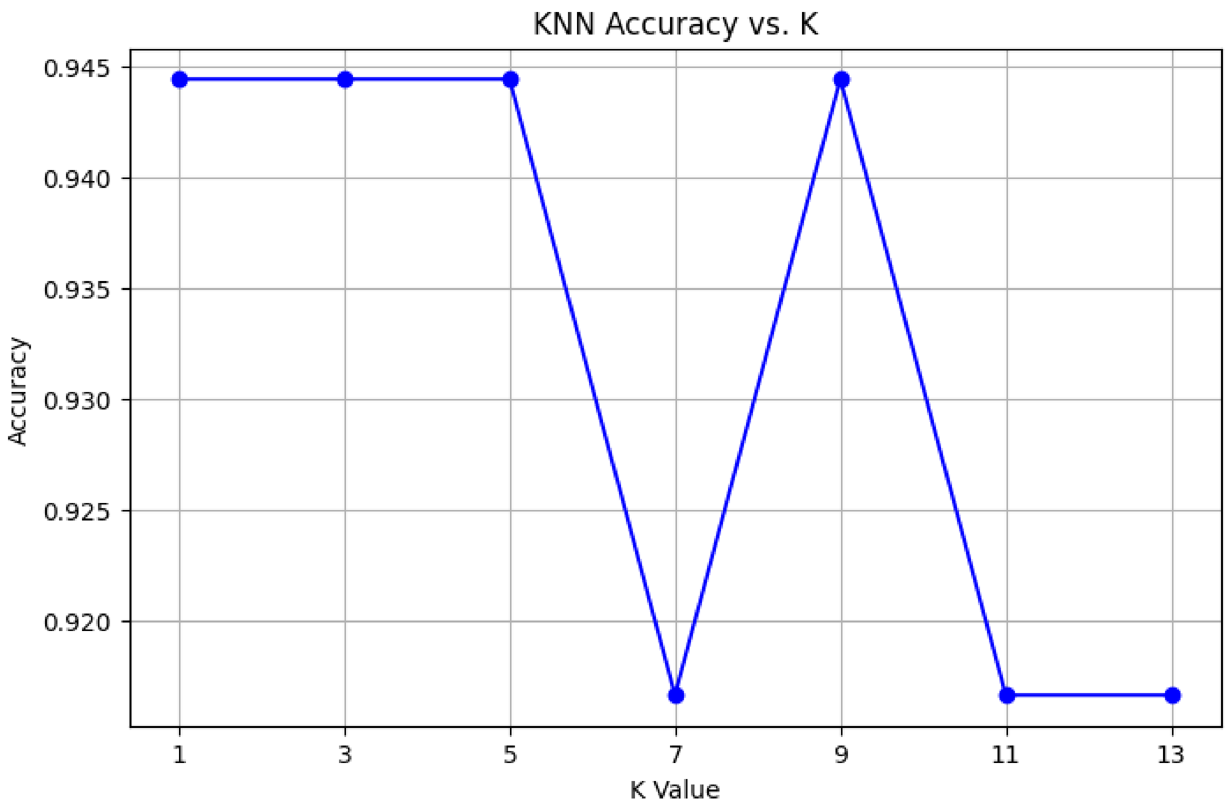
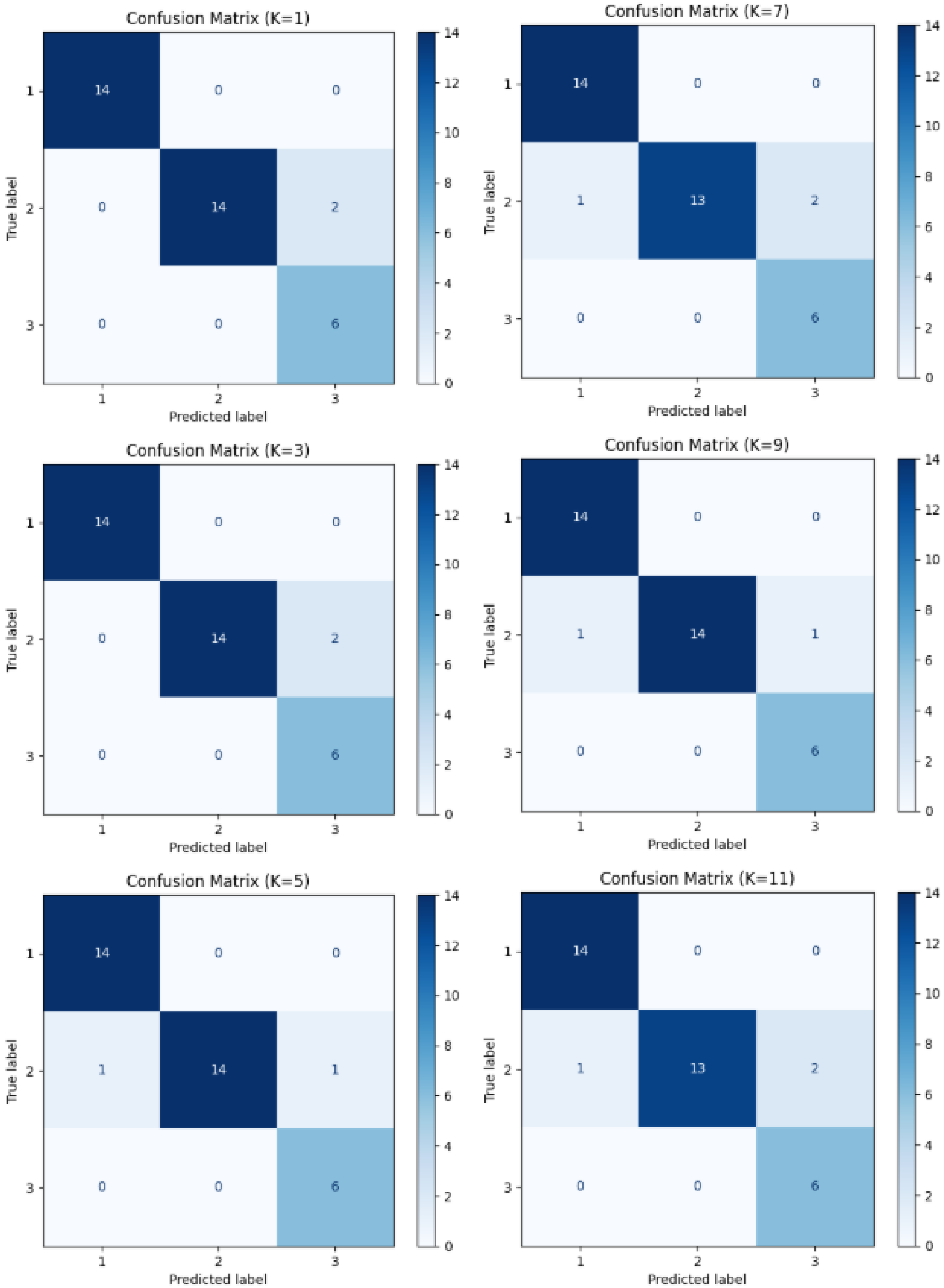



Figure 38


Confusion Matrix (K=1)
14 0 0
0 14 2
0 0 6
True label
Predicted label
Confusion Matrix (K=7)
14 0 0
1 13 2
0 0 6
True label
Predicted label
Confusion Matrix (K=3)
14 0 0
0 14 2
0 0 6
True label
Predicted label
Confusion Matrix (K=9)
14 0 0
1 14 1
0 0 6
True label
Predicted label
Confusion Matrix (K=5)
14 0 0
1 14 1
0 0 6
True label
Predicted label
Confusion Matrix (K=11)
14 0 0
1 13 2
0 0 6
True label
Predicted label

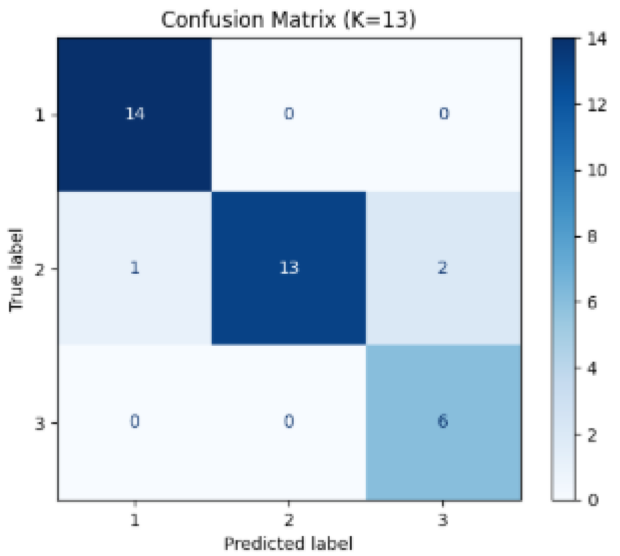



Figure 39

K=1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 0.88 | 0.93 | 16 |
| Class 3 | 0.75 | 1.00 | 0.86 | 6 |
| accuracy | | | 0.94 | 36 |
| macro avg | 0.92 | 0.96 | 0.93 | 36 |
| weighted avg | 0.96 | 0.94 | 0.95 | 36 |

K=3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 1.00 | 1.00 | 1.00 | 14 |
| Class 2 | 1.00 | 0.88 | 0.93 | 16 |
| Class 3 | 0.75 | 1.00 | 0.86 | 6 |
| accuracy | | | 0.94 | 36 |
| macro avg | 0.92 | 0.96 | 0.93 | 36 |
| weighted avg | 0.96 | 0.94 | 0.95 | 36 |

K=5

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.88 | 0.93 | 16 |
| Class 3 | 0.86 | 1.00 | 0.92 | 6 |
| accuracy | | | 0.94 | 36 |
| macro avg | 0.93 | 0.96 | 0.94 | 36 |
| weighted avg | 0.95 | 0.94 | 0.94 | 36 |

K=7

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.81 | 0.90 | 16 |
| Class 3 | 0.75 | 1.00 | 0.86 | 6 |
| accuracy | | | 0.92 | 36 |
| macro avg | 0.89 | 0.94 | 0.91 | 36 |
| weighted avg | 0.93 | 0.92 | 0.92 | 36 |

K=9

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.88 | 0.93 | 16 |
| Class 3 | 0.86 | 1.00 | 0.92 | 6 |
| accuracy | | | 0.94 | 36 |
| macro avg | 0.93 | 0.96 | 0.94 | 36 |
| weighted avg | 0.95 | 0.94 | 0.94 | 36 |

K=11

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.81 | 0.90 | 16 |
| Class 3 | 0.75 | 1.00 | 0.86 | 6 |
| accuracy | | | 0.92 | 36 |
| macro avg | 0.89 | 0.94 | 0.91 | 36 |
| weighted avg | 0.93 | 0.92 | 0.92 | 36 |

K=13

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Class 1 | 0.93 | 1.00 | 0.97 | 14 |
| Class 2 | 1.00 | 0.81 | 0.90 | 16 |
| Class 3 | 0.75 | 1.00 | 0.86 | 6 |
| accuracy | | | 0.92 | 36 |
| macro avg | 0.89 | 0.94 | 0.91 | 36 |
| weighted avg | 0.93 | 0.92 | 0.92 | 36 |

Figure 40

When Chebyshev Distance, Manhattan Distance and Euclidean Distance methods are compared, it is seen that Euclidean Distance is the most suitable method for this data set. In other methods, the accuracy is 1 for the tested k values and the values close to 1 are limited. In the Euclidean Distance method, accuracy is always 1 after a certain k value. A 1 for accuracy indicates that all predictions are classified correctly.

As a result, I evaluated a data set of 178 samples given in this assignment. I evaluated the features of the data according to the overlap formation status. I separated 20 percent of our samples as test data and tested the kNN method for different distance finding methods and different k values. I found the most suitable distance finding method as Euclidean Distance and I found that choosing k as 7 should be the most suitable choice for this method. It is important for the speed of the algorithm that k is not chosen as a very high value.

GitHub link is given below.

https://github.com/kivancates/kNN-algorithm-ele489