Contents

[1- Using sci-kit learn or other python packages to implement a naive Bayesian classifier ( GaussianNB ) 2](#_Toc106133651)

[As a preprocessing step: 2](#_Toc106133652)

[1. a) using the train test split function in sci-kit learn to split the dataset into a training set, and a testing set. 2](#_Toc106133653)

[1. b) Use the classification report function to help you calculate precision, recall, and f1. 3](#_Toc106133654)

[1. c) Plot the decision boundary on the test set. 3](#_Toc106133655)

[2- Using sci-kit learn or other python packages to implement KNN classifier 8](#_Toc106133656)

[2. a) shuffling the dataset and splitting the dataset into a training set with 1000 samples, a validation set with 300 samples, and a testing set with 428 samples. 8](#_Toc106133657)

[2. b) Transforming the string values into numbers. 8](#_Toc106133658)

[2. c) Trying to use a different number of training samples to show the impact of several training samples. Use 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the training set for 10 separate KNN classifiers and show their performance (accuracy score) on the validation set and testing set. 10](#_Toc106133659)

[2.d) Using 100% of training samples, trying to find the best K value, and showing the accuracy curve on the validation set when K varies from 1 to 10. 11](#_Toc106133660)

[2. e) Analyzing the training time when using different numbers of training samples when k =2 and k=10 and using the full training data and with only 10% of it. 13](#_Toc106133661)

[2.d) Providing conclusions from the experiments of questions (c), (d), and (e) in this question. 14](#_Toc106133662)

## 1- Using sci-kit learn or other python packages to implement a naive Bayesian classifier ( GaussianNB )

### As a preprocessing step:

It's found that the number of doors and number of persons has the values more and 5more, so we replaced them with numerical values.

A screenshot of a computer

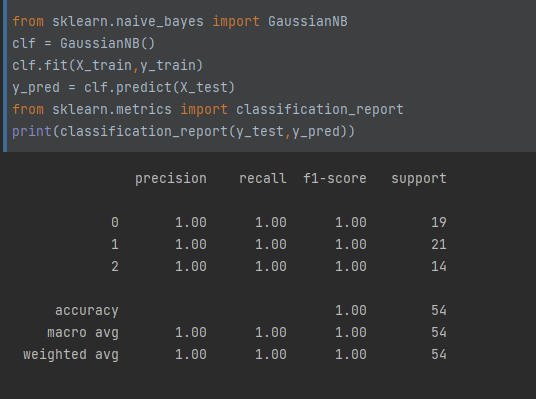
Description automatically generated

### 1. a) using the train test split function in sci-kit learn to split the dataset into a training set, and a testing set.

Text

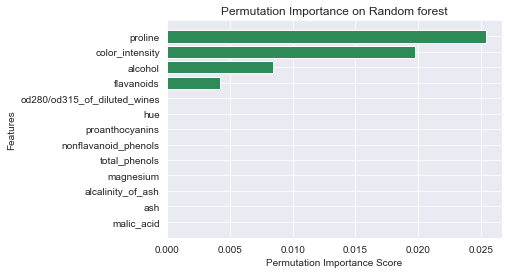
Description automatically generated

### 1. b) Use the classification report function to help you calculate precision, recall, and f1.



### 1. c) Plot the decision boundary on the test set.

We used permutation importance with the Random Forest classifier:

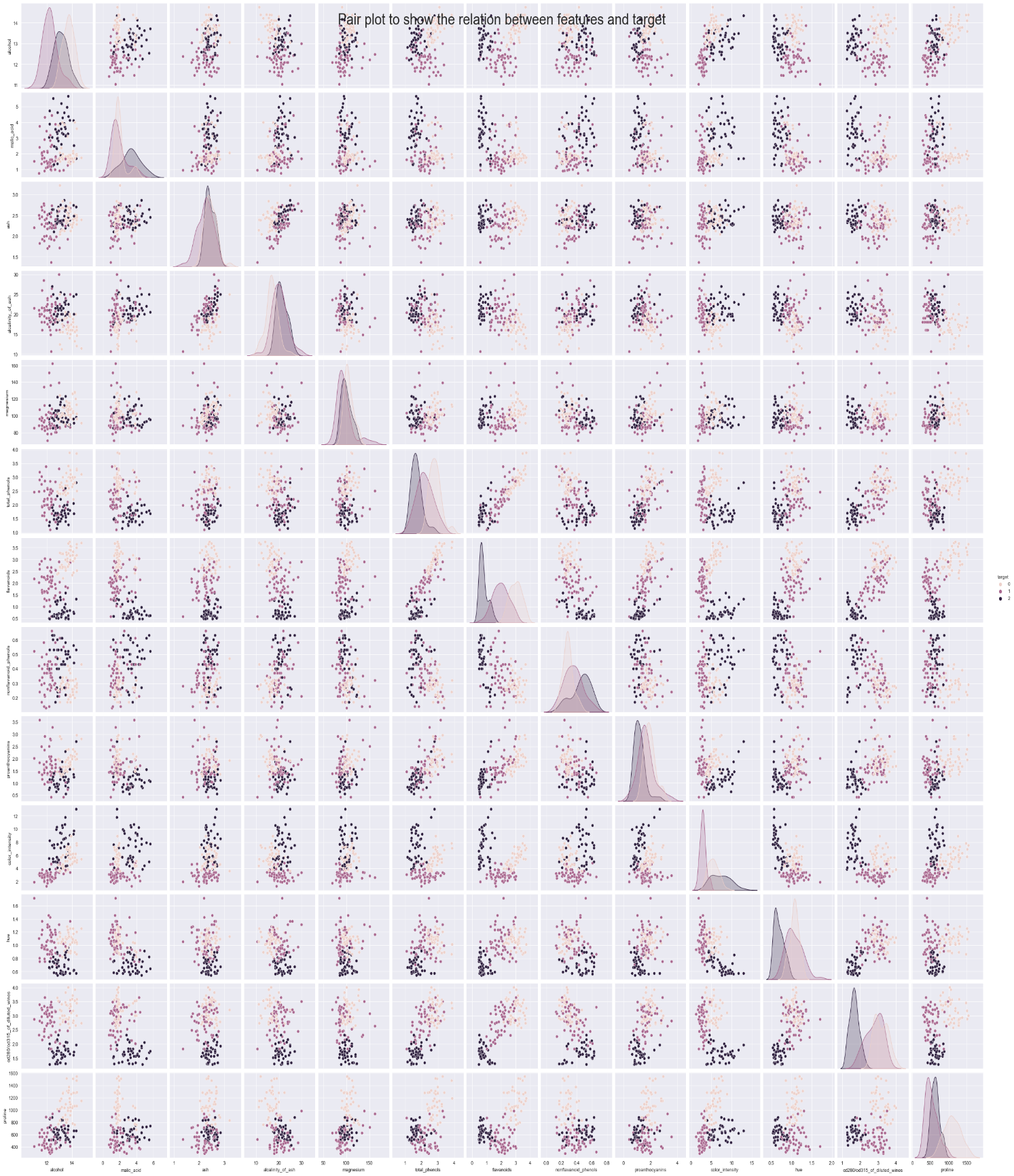


We also used Chi2 to get the best 2 features:

Graphical user interface

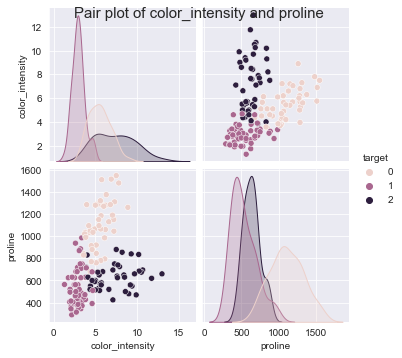
Description automatically generated with medium confidence

Showing the relation between features against the target to select the best features.



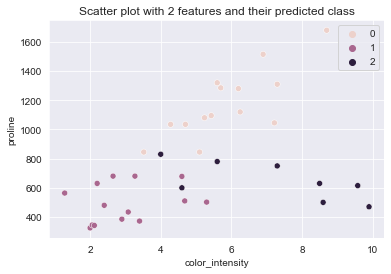
According to this figure, we can notice that these two features **[color\_intensity, proline ]** are linearly separable.

**As a zoom in to the selected features**



**So, The top selected features are [color\_intensity, proline ]**

Plotting Scatter plot to see the classification using our model and top selected features



Plotting decision boundaries using a new model after training with the selected features.

Chart, scatter chart

Description automatically generated

## 2- Using sci-kit learn or other python packages to implement KNN classifier

### 2. a) shuffling the dataset and splitting the dataset into a training set with 1000 samples, a validation set with 300 samples, and a testing set with 428 samples.

Graphical user interface, text

Description automatically generated

### 2. b) Transforming the string values into numbers.

First, we selected the categorical variables

A screenshot of a computer

Description automatically generated with medium confidence

Then we applied the Ordinal encoder

Text

Description automatically generated

And for the target column, we applied the label encoder

A picture containing background pattern

Description automatically generated

### Graphical user interface, text Description automatically generated2. c) Trying to use a different number of training samples to show the impact of several training samples. Use 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the training set for 10 separate KNN classifiers and show their performance (accuracy score) on the validation set and testing set.

**Visualization of the results:**

Chart, line chart

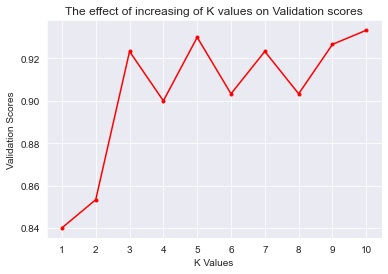
Description automatically generated

**So, we found that the accuracy increases when the dataset increases**

### 2.d) Using 100% of training samples, trying to find the best K value, and showing the accuracy curve on the validation set when K varies from 1 to 10.

Text

Description automatically generated

**Visualization of the results** 

**So, the best k =5**

2. e) Analyzing the training time when using different numbers of training samples when k =2 and k=10 and using the full training data and with only 10% of it. Text

Description automatically generated

**Visualization of the results:**

Chart, bar chart

Description automatically generated

The more the dataset size, the more training time and this makes sense

Chart, bar chart

Description automatically generated

The more the dataset size, the more testing time and this makes sense

### 2.d) Providing conclusions from the experiments of questions (c), (d), and (e) in this question.

Conclusion:

Regarding question (c)when we used 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the training set for 10 separate KNN we found that the accuracy increases when the dataset increases.

Regarding question (e)when we tried different values of k, we found that the best k=5.

Regarding question (d) after analyzing the training time and testing time we found that the more the dataset size, the more time required.