Lab 3: D7054E

Sebastian Morel (sebmor-8)
Patrick Pilipiec (patpil-9)

Group 7

March 9, 2021

Introduction

The objective of Part A is to classify wine as either belonging to the group 1, 2, or 3. A total of 12 features is used in this predictive model. Part B is divided up into 2 sections under the category "Artificial Neural Networks". In the first part we are tasked to predict if the house prices are above or below median value, this will be achieved by using a python library called Keras. In the second part we will also be using Keras but implementing it onto another dataset.

1 Methodology

In this section, we will discuss the techniques and the preprocessing of the dataset.

1.1 Dataset

1.1.1 Part A

We make use of the Wine Data Set (from "https://archive.ics.uci.edu/ml/datasets/wine"). This dataset contains 178 observations with a total of 13 variables. These variables are:

1. Class
2. Alcohol
3. Malic acid
4. Ash
5. Alcalinity of ash
6. Magnesium
7. Total phenols
8. Flavanoids
9. Nonflavanoid phenols
10. Proanthocyanins
11. Color intensity
12. Hue
13. OD280/OD315 of diluted wines
14. Proline

All variables are numeric, meaning either integer of real values. As one of these variables contains the ordinal value of a class, this dataset can be used for classification.

1.1.2 Part B

For part B three datasets were used in total, the first dataset that was used is the Zillow's Home Value (from "https://drive.google.com/file/d/1GfvKA0qznNVknghV4botnNxyH-KvODOC/view"), this dataset contains 10 columns:

1. Lot Area
2. Overall Quality (Rated from 1-10)
3. Overall Condition (Rated from 1-10)
4. Total Basement Area
5. Number of full bathrooms
6. Number of half bathrooms
7. Number of bedrooms above ground
8. Total number of rooms above ground
9. Number of fireplaces
10. Garage Area
11. Above Median Price

The other two datasets are the UCI wine datasets (from "https://archive.ics.uci.edu/ml/datasets/wine+quality"), one dataset of white wine and the other for red wine. The dataset consists of 12 columns:

1. Fixed Acidity
2. Volatile Acidity
3. Citric Acid
4. Residual Sugar
5. Chlorides
6. Free Sulfur Dioxide
7. Total Sulfur Dioxide
8. Density
9. pH
10. Sulfates
11. Alcohol
12. Quality

1.2 Techniques

1.2.1 Part A

For the analyses, Python will be used. Pandas will be used to process the data in a dataframe. For visualization, we will use the packages Matplotlib and Seaborn. Finally, the predictive model is trained and tested using SciKit Learn. From the same package, we will use functionality to scale features and to evaluate the predictive power of this model. CRISP-DM is used to model this predictive assignment, meaning that we will follow the phases of the CRISP-DM framework. In addition, the predictve analysis is encapsulated in a Python class, where we will make plenty of use of class variables to store states, and methods to implement behaviors. We have selected to implement Naïve Bayes for these analyses.

1.2.2 Part B

In the first part we learn how to build a neural network with the help of the Keras library. Like for part A pandas was used to process the data into a dataframe. We also used SciKit Learn in order to normalise the data. Finally to plot the graphs for this part we used matplotlib.
For the second part we also used pandas to load in the csv datafile to convert it to a dataframe. Similar to part 1 we also used matplotlib to plot, but also used seaborn in order to plot a correlation heatmap. To preprocess the data we also here used SciKit learn, we also used SciKit learn to gather evaluation metrics on our model.

1.3 Pre-processing of the dataset

1.3.1 Part A

The data are already clean. Nevertheless, we explicitly check any missing values but observe that there are no missing values. Subsequently, we split the features of the dataframe into X and y sets, such that the X set holds all features, while y holds the class labels. As Naïve Bayes performs better when the features are scaled, we use the StandardScaler by SciKit-Learn to scale the features. Finally, we use train_test_split provided by SciKitLearn to create datasets for training and testing purposes. We specify a test size of 30% and choose 42 as the random state, which enables replication of these datasets.

1.3.2 Part B

For the first part of part B, the data is processed by first converting the csv file into arrays to allow the algorithm to process the data. We then split our data to into X and Y, X will be the input and Y is what we want to predict. Next we normalise our data by using SciKit learn. Lastly we split the data into a training set and a test set, with a test size of 0.3.
For the preprocessing of the data in the second part we used similar methods, although since we start off with two data sets on this one we start by merging the two datasets and differentiating them by having red wine as type 1 and white wine as type 0. We then split the data into X and Y, where X is the input, and Y what we want to predict which is the type of wine. Lastly like the first part we normalize our data by using SciKit learn.

2 Results

2.1 Part A

In Figure 1, we observe that Class 1 wine tends to have a higher percentage of alcohol. Class 2 has, relatively, the lowest percentage. Class 3 is in-between.

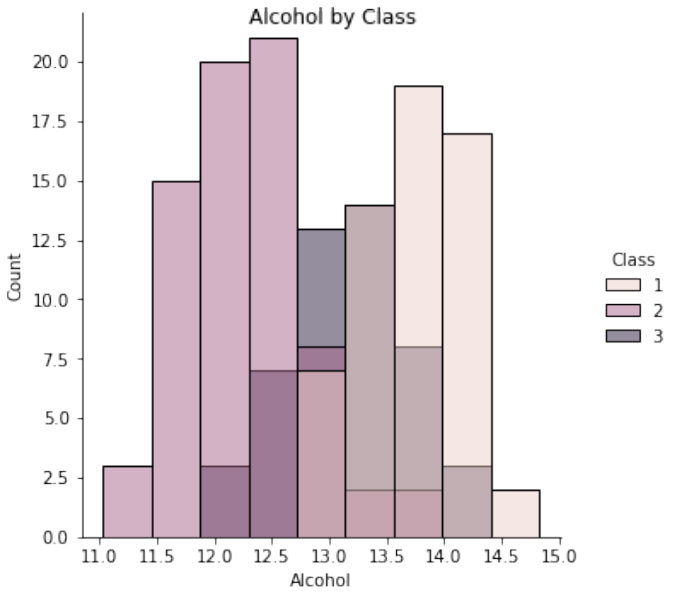

Figure 1: Alcohol by Class

There is not a clear distinction between the malic acid that wine contains (see Figure 2). In general, however, Class 2 and 3 wine tends to contain more malic acid compared to Class 1 wine.

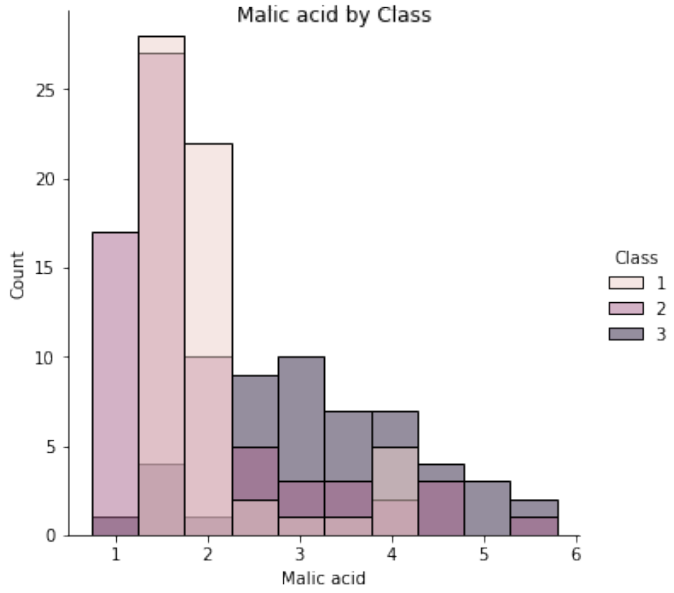

Figure 2: Malic acid by Class

It appears that Class 2 wine contains least ash (see Figure 3).

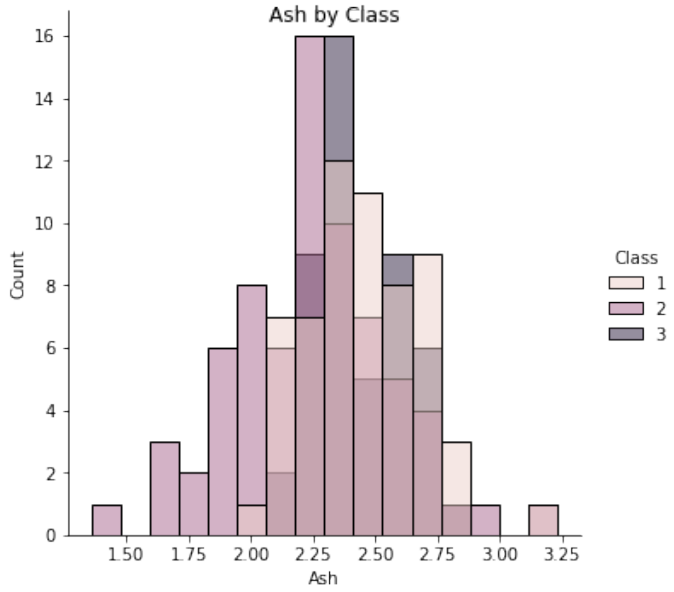

Figure 3: Ash by Class

The alcalinity of the ash in Class 1 wine is lower than for Class 2 and 3 wines. See Figure 4.

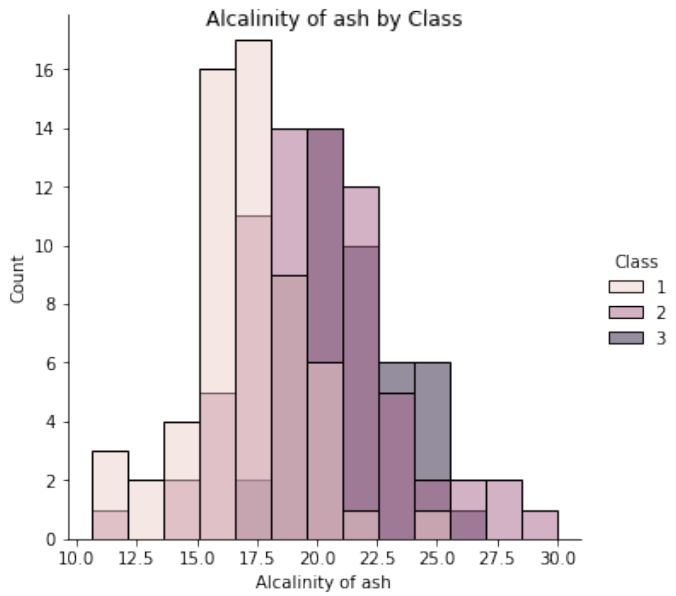

Figure 4: Alcalinity of ash by Class

Class 2 and 3 wines are more likely to contain less magnesium, compared to Class 1 wine (see Figure 5).

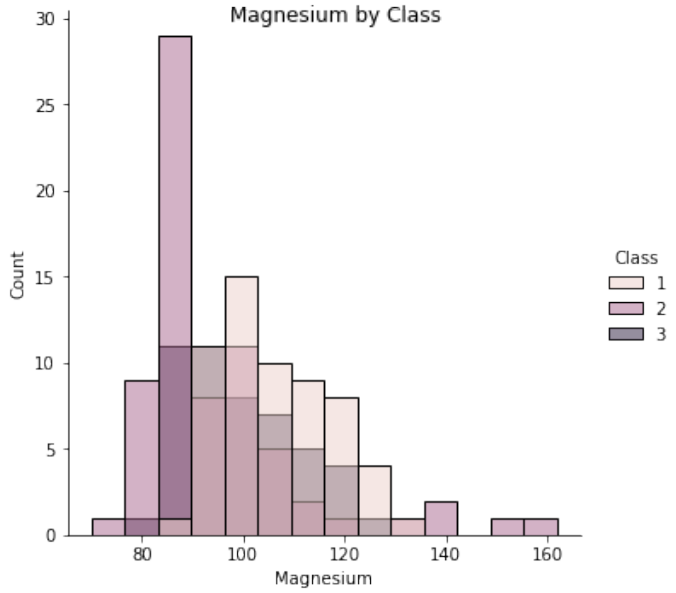

Figure 5: Magnesium by Class

Similarly, Class 2 and 3 wines more often contain less total phenols, compared to Class 1 wine (see Figure 6).

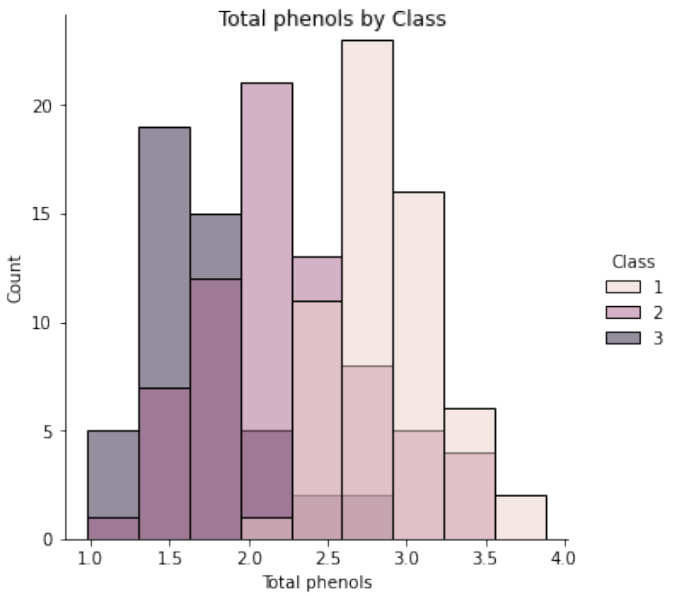

Figure 6: Total phenols by Class

Again, Class 2 and 3 wines more often contain less flavanoids than Class 1 wines (see Figure 7).

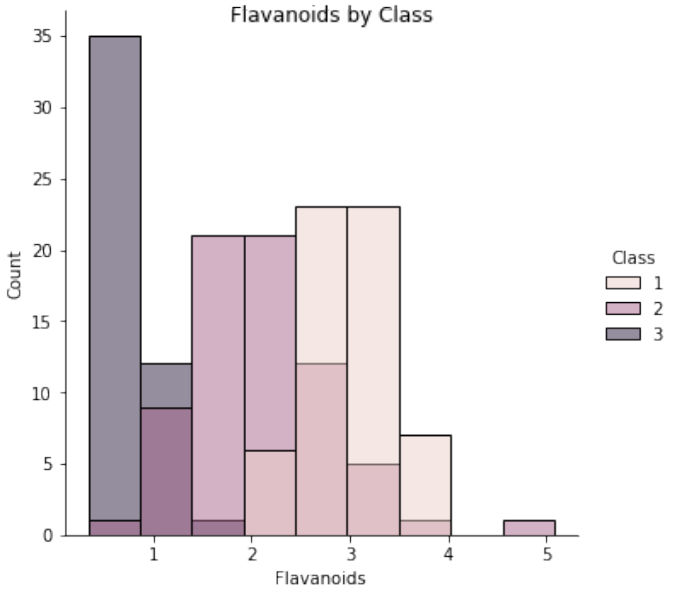

Figure 7: Flavanoids by Class

However, as is illustrated in Figure 8, nonflavanoid phenols are lower in Class 1 wine, compared to Class 2 an 3.

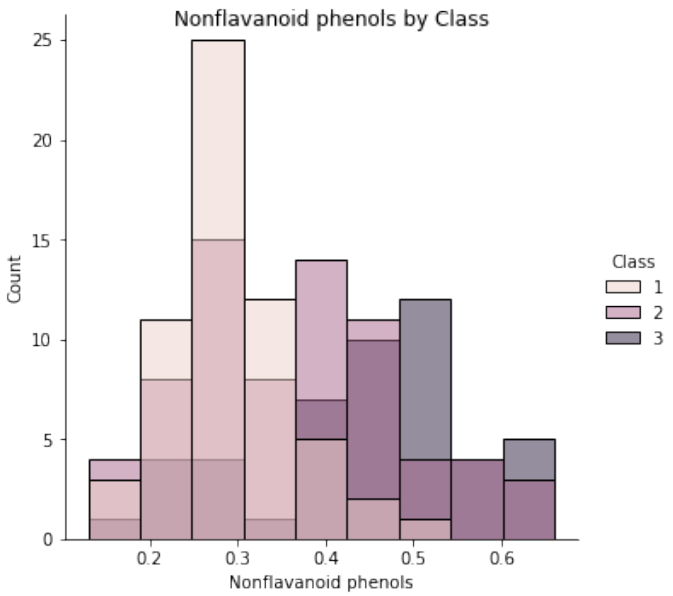

Figure 8: Nonflavanoid phenols by Class

Furthermore, proanthocyanins are higher in Class 1 wine (see Figure 9).

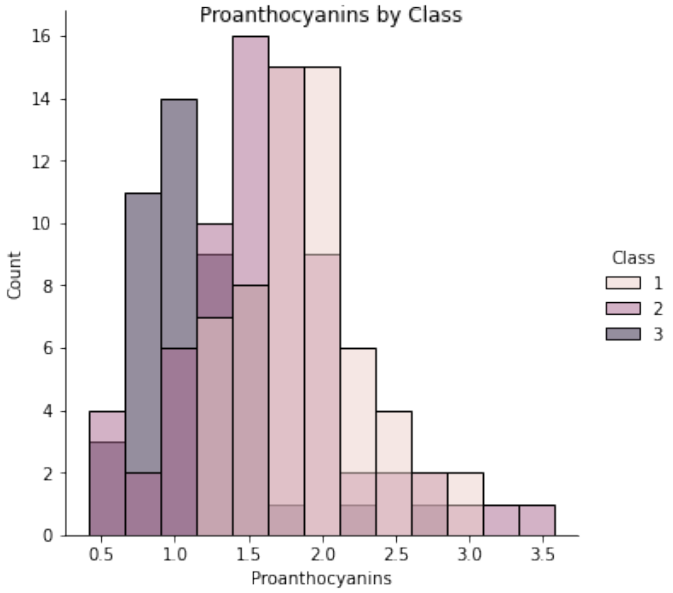

Figure 9: Proanthocyanins phenols by Class

The color intensity of wine is lower for Class 2. The intensity is highest for Class 3 wine. Class 1 wine more often has a color intensity that is in the middle. See Figure 10.

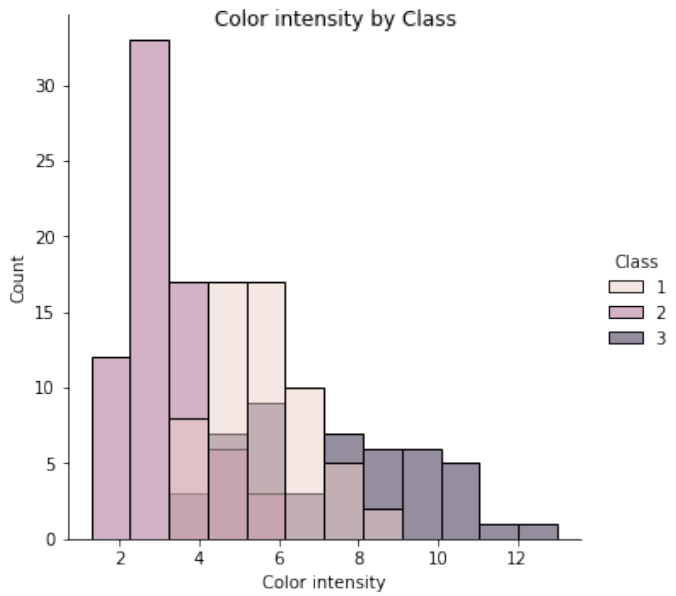

Figure 10: Color intensity by Class

The hue of Class 3 wine is lowest. There is no clear distinction between the hue of Class 1 and 2 wine. See Figure 11.

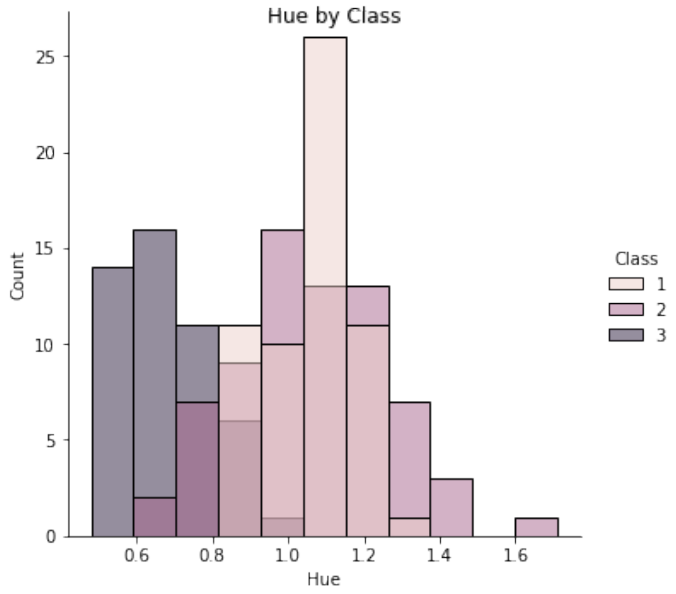

Figure 11: Hue by Class

The dataset does not contain information about the meaning of the feature "OD280/OD315 of diluted wines". Although the distribution of this variable by Class is presented in Figure 12, we cannot interpret this.

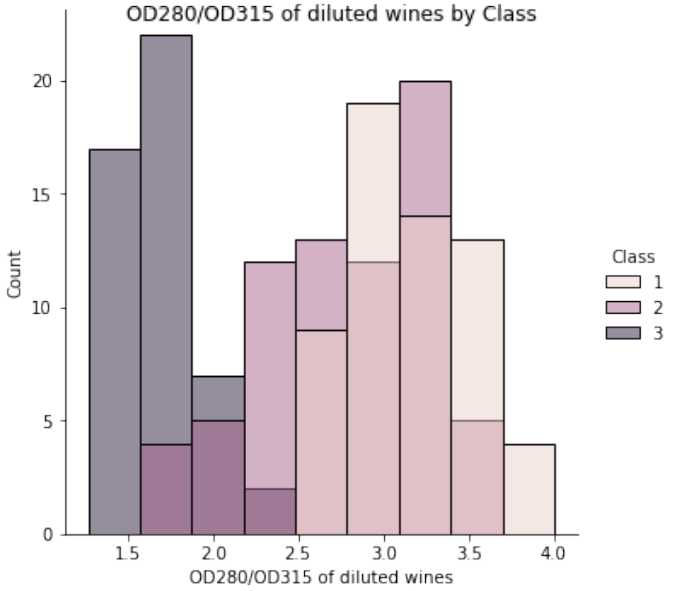

Figure 12: OD280/OD315 of diluted wines by Class

Lastly, Class 1 wine has a higher proline than Class 2 and 3 wine (see Figure 13).

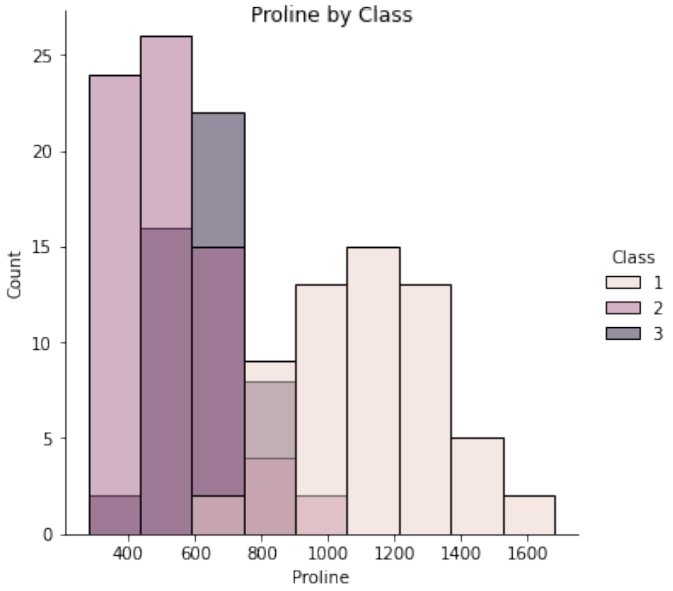

Figure 13: Proline by Class

All these 12 features were included in the Naïve Bayes model to predict the outcome variable Class. Overall, we find that the trained Naïve Bayes classifier was able to predict the outcome variable perfectly. The accuracy score was 1.0. The confusion matrix, which is presented in Table 1, illustrates that no classification errors were made when the trained model was tested on the test set. In this matrix, the true labels are provided on the x-axis, while the predicted labels are presented on the y-axis.

Table 1: Confusion matrix for Naïve Bayes

	Class 1	Class 2	Class 3
Class 1	19	0	0
Class 2	0	21	0
Class 3	0	0	14

2.2 Part B

2.2.1 Part 1

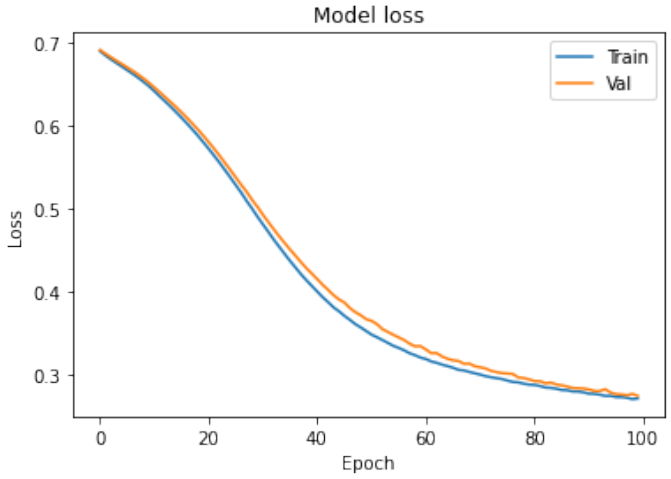

Figure 14: Loss over epoch for the train and validation set

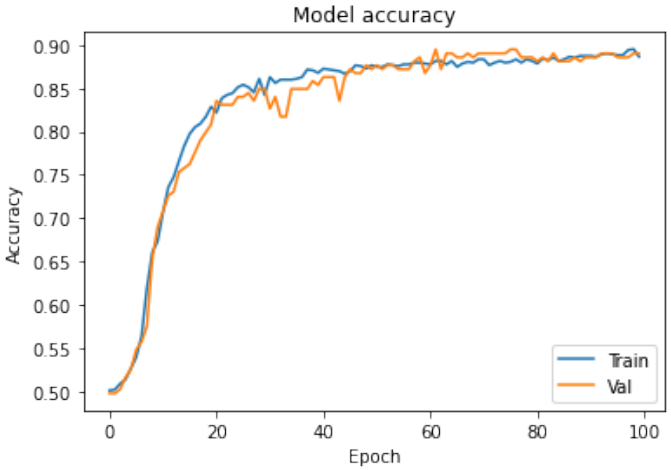

Figure 15: Accuracy over epoch for the train and validation set

Our first model's loss and accuracy is presented by figure 14 and figure 15 respectively. This model has the hidden layers with two layers having 32 neurons with relu activation, and the output layer having 1 neuron with sigmoid activation.

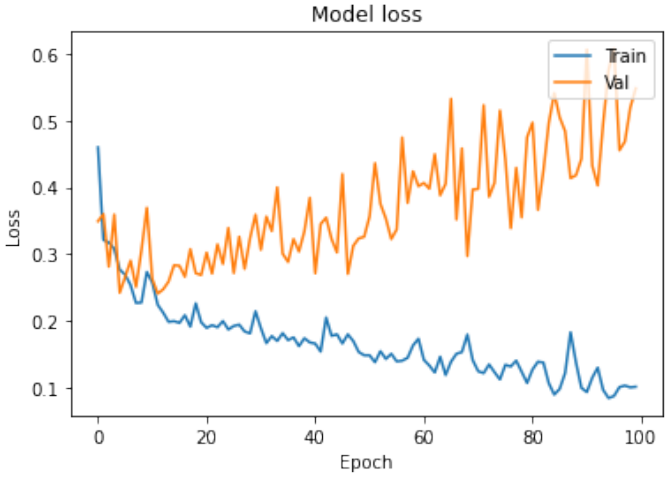

Figure 16: Loss over epoch for the train and validation set when the model is overfitting

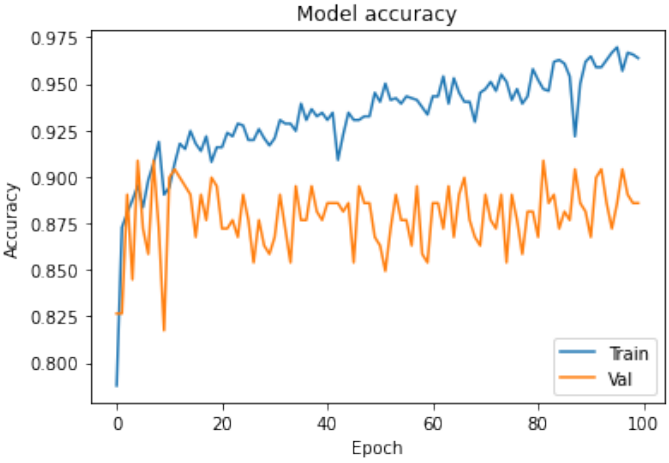

Figure 17: Accuracy over epoch for the train and validation set when the model is overfitting

The second model's loss and accuracy is represented by figure 16 and figure 17 which illustrates how the graphs would look like when the model is over-fitting. This model has 5 layers with the hidden layers having 1000 neurons with relu activation, and the output layer having 1 neuron with sigmoid activation.

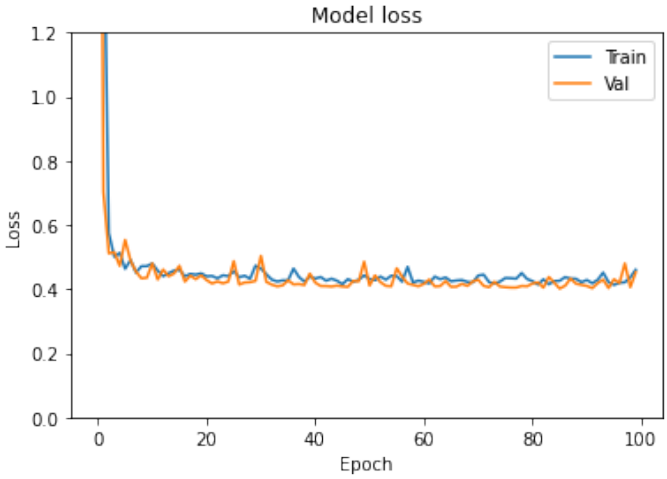

Figure 18: Loss over epoch for the train and validation set after treating the overtfit model

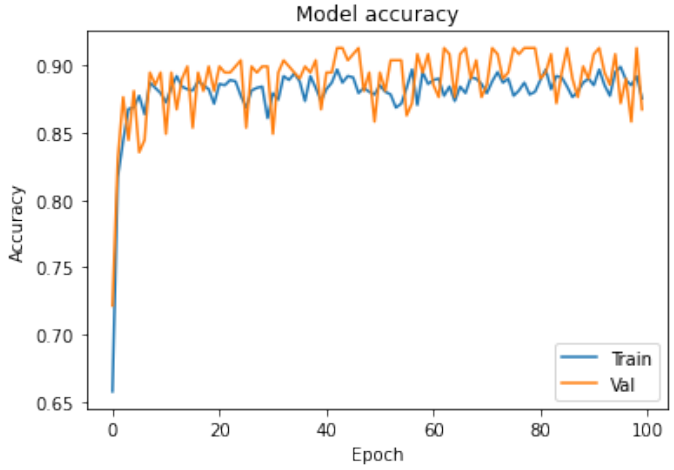

Figure 19: Accuracy over epoch for the train and validation set after treating the overtfit model

For our third model we implemented L2 regularization and dropout on the second model to reduce the over-fitting. Figure 18 and figure 19 show the loss and accuracy respectively after implementing those two strategies onto the second model.

2.2.2 Part 2

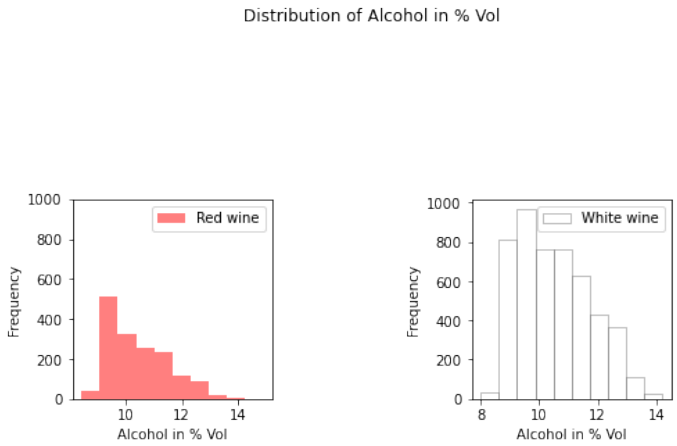

Figure 20: Histogram comparing the frequency for alcohol in % Vol between red whine and white whine.

From figure 20 we can see that the amount of alcohol in each wine are relatively the same in the dataset. The most common alcohol % for the wines is around 9-10

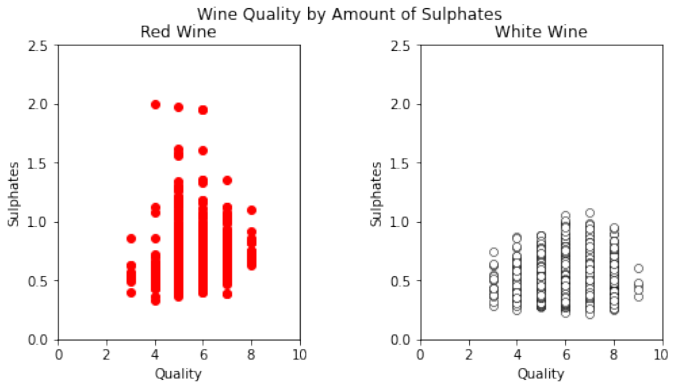

Figure 21: Scatterplot showing the amount of sulphate for each wine against the quality of the wines.

From figure 21 we can see that red whine has alot more sulfate than the white wine, and despite the sulfate level being low for the white wine it still has a quality of 9.

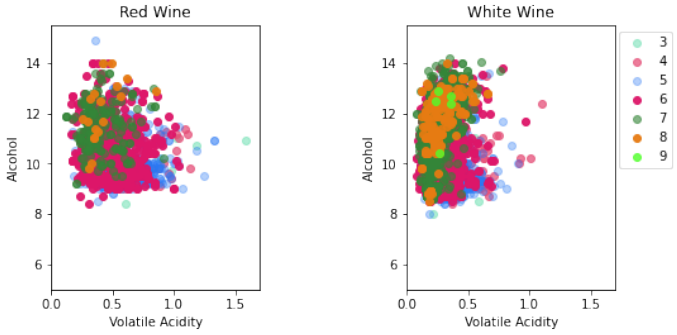

Figure 22: Scatter plot for alcohol level vs acidity level color distinguished by the quality

Lastly we chose to analyse if the acidity and alcohol level has an affect on the quality of the wines from the dataset in figure 22. In the figure we can see some outliers that have a higher acidity level especially for the red wines. Although most of the datapoints are too clustered to draw a clear conclusion.

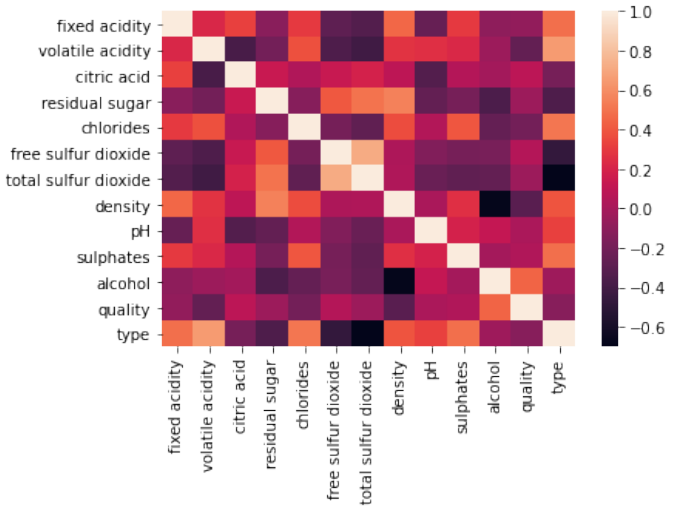

Figure 23: Correlation matrix to see correlation of features between red and white wine.

The four most prominent correlations that sticks out in Figure 23 are, type and volatile acidity, type and sulphates, residual sugar and density, and free sulfur dioxide and total sulfur dioxide (obviously).

```
Loss: 0.02372116409242153
Accuracy: 99.53380227088928 %
Confusion Matrix: [[1587    1]
 [   9  548]]
Precision Score: 0.9981785063752276
Recall Score: 0.9838420107719928
F1 Score: 0.9909584086799277
Cohen's Kappa: 0.9878179265599869
```

Figure 24: Evaluation of our model.

The results that are showcased in Figure 24 are very good. From the confusion matrix we can see that our model only misclassified 1 white wine as red wine, and 9 red wines as white wines.

3 Conclusion

3.1 Part A

We conclude that it is perfectly possible to create a predictive model using Naïve Bayes, and to classify wine to one of the three classes based on the provided 12 features. No classification errors were made, meaning that there is no need to further improve this predictive model. Also, Naïve Bayes is an algorithm that is fairly efficient. If improvements were to be suggested, one may try to achieve the same performance using fewer features. Also, Principal Component Analysis could be used to reduce the dimensionality of the data by encapturing the variance of these features in a fewer number of features. However, strictly speaking, as the dataset is small and it only contains few features, such improvements are not relevant or necessary.

3.2 Part B

For the first part we see that the optimal model is the first one. In the first model we use dense layers, the two hidden layers have 32 neurons with ReLu activation, the output layer having 1 neuron with Sigmoid activation, and the input layer consists of 10 input features, the model got an accuracy of 87.21%. When adding more neurons and layers like in our second model the model will start to overfit because the model becomes too complex, thus the first model gives us the best performance.
For the second part we went deeper into our analysis of the dataset which gave us an informative insight into the wine dataset we would be working on. For one we had almost three times more white wines than red wines, this can be reflected in the Confusion Matrix, because the model has been trained with more white wine data and thus the model misclassified more red wines as white wines. In conclusion we find that Keras is a very good library and even with a few lines of code we managed to achieve a very high accuracy on our model.