Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

Wine recognition

1. Introduction

Wine can be classified through many methods, which include: appellation (term used to identify region and location where the grapes used for the wine were grown), vintage, sweetness, variety, style and vinification methods. The method may vary from one region to another or one country to another. The classifications may have been created by simply by growers' organizations to classify wines unofficially, and some are officially protected by wine laws inside the country (source: Wikipedia).

The most famous and used classifications include Bordeaux Wine Official Classification of 1855, German Wine Classification system, Classification of Saint-Émilion Wine and Cru Bourgeois of Bordeaux.

The purpose of the current project is to analyse only the following wine's chemical properties: (alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavonoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines and proline). It is known that the wines belong to three cultivars from the same region in Italy. We have the information of which cultivar the sample belongs for 130 wine samples, and another 48 wines that we need to predict its origin. With the help of machine learning techniques and data analysis, we will create a classification algorithm to predict from which cultivar they belong.

The language used was Python with the help of well-known packages, such as Seaborn, Numpy, Pandas and Sci-kit Learn.

Altogether with the best machine learning method, we will hand in our best logistic regression model, best KNN (K-Nearest Neighbours) classifier and best MLP (Multi-layer Perceptron) and its accuracies.

2. Methodology

All the techniques used in this project are considered either EDA (Exploratory Data Analysis) or machine learning classification algorithms.

First of all, we checked for the number of labels (cultivars) for each of the wines to check if there were considerable differences. As can be seen in Fig. 1 below, the number of labels for cultivars 1, 2 and 3 were 44, 51 and 35, respectively.

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

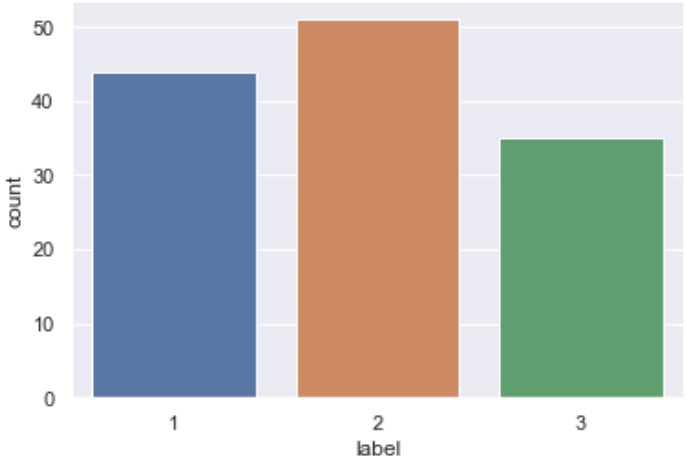

Fig. 1: Number of labels for each wine cultivar

After trying K-Nearest-Neighbours (using K = 1, 3 and 5) with a low predictive accuracies (as can be seen on table 1), we decided to transform the data. The data needs to be normalized in distance-based classifications, otherwise some features can be conditioned by features with a wider range of values when distances are computed. The method chosen was Box-Cox transformation, which is known to transform non-normal dependent variables into normal shape. After the transformation, the accuracy of the predictions increased by more than 20%.

KNN = 3 Classification report

	precision	recall	f1-score	support
1	0.70	0.70	0.70	10
2	0.59	0.77	0.67	13
3	0.83	0.50	0.62	10
micro avg	0.67	0.67	0.67	33
macro avg	0.71	0.66	0.66	33
weighted avg	0.70	0.67	0.66	33

Table 1: Accuracies for K-Nearest-Neighbours with N = 3 and without data transformation

After that the transformation, we moved onto new training techniques, such as Logistic Regression, MLP (Multi-layer Perceptron using 1 and 2 layers), Gaussian Naive Bayes and Random Forest. Some attempts were done using all the features, and after, trying a feature selection with the help of the Fishers' indexes, data analysis and Random Forest algorithm.

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

3. Data

The data set *wineITrain* consists of 178 observations, *wineTrainX*, with samples and features (13x130), *wineTrainY*, with the respective labels (1x130) and *wineTestX* (13x48). There were no labels for the last 48 observations.

After creating pairplots for all the possible relations between features (with the help of Seaborn library), we verified that some of them were good features and relations for classifying some of the labels:

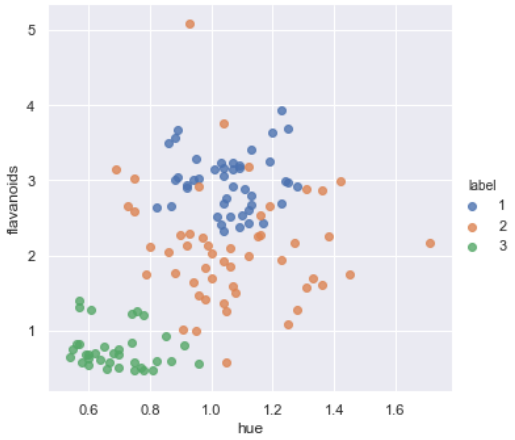

Fig. 2: Relation between 'flavanoids' and 'hue'

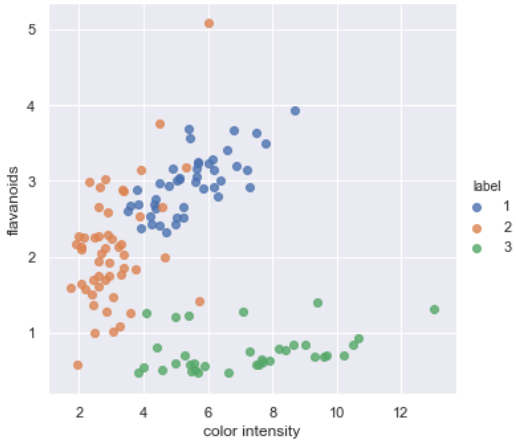

Fig. 3: Relation between 'flavanoids' and 'color intensity'

Analyzing Fig. 2 and 3, we can clearly see that label 3 can be separated and flavanoids can be considered an important feature with respect to classification. We also checked how correlated features were, because features that have a high correlation may interfere in some classification methods.

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

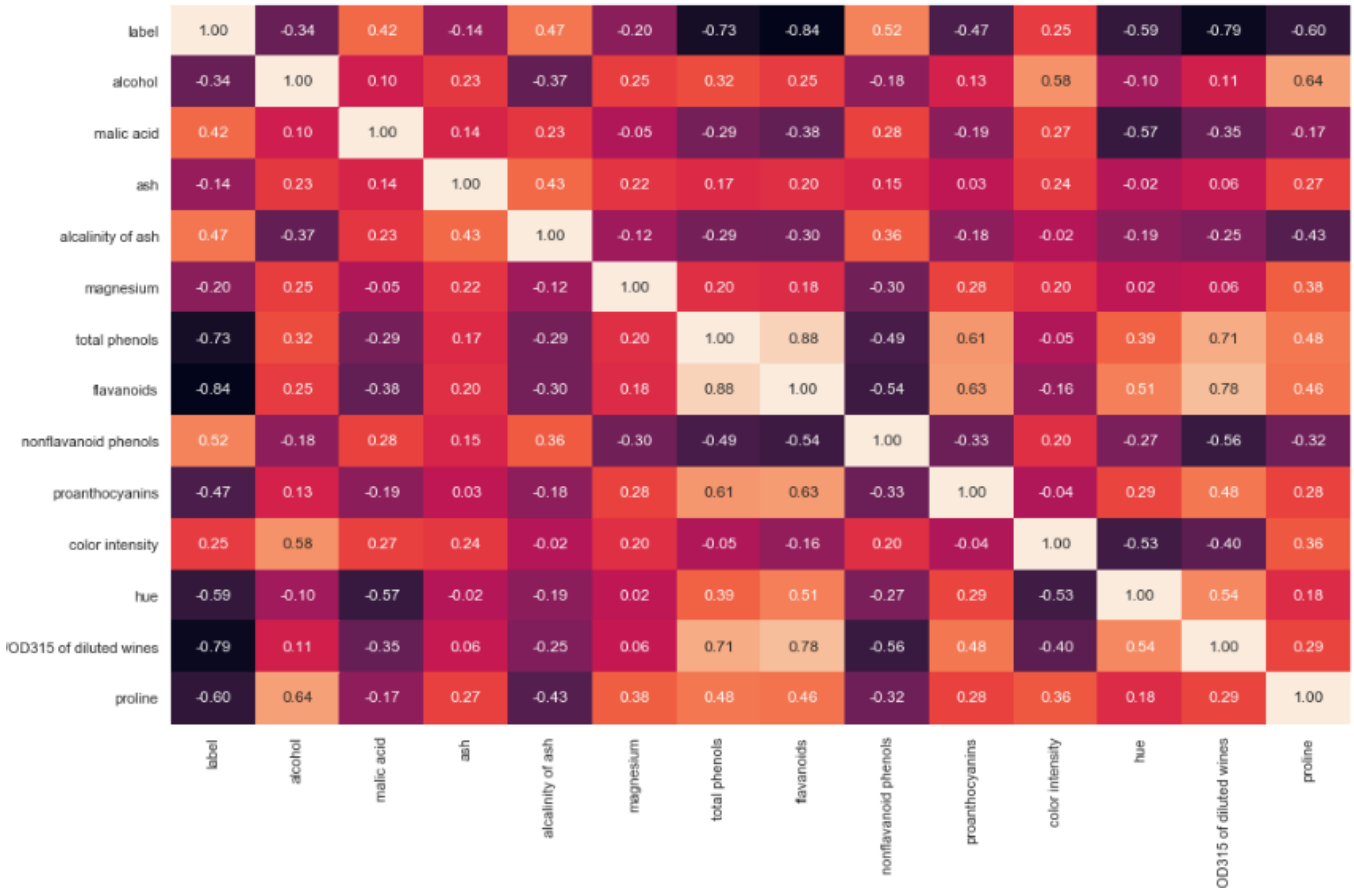

Fig. 4: Correlation between features

From the image above (Fig. 4), we can verify that ‘flavanoids’ and ‘total phenols’ are very correlated. Correlated features imply that there is very little information in linear combinations of the features. Removing the correlation simplifies the model and improves many algorithms, such as Logistic Regression which suffers from multicollinearity. Random Forest, which is also good at detecting interactions, can suffer from very correlated features and misinterpret or hide some interactions.

After testing ‘total phenols’ and ‘flavanoids’ with box plots, it was found that ‘flavanoids’ was the best feature to keep in the models for its capacity of separating labels 1 and 3. Therefore, several tests were made with and without using ‘total phenols’ to verify if improvements were achieved. Fig. 5 below shows exactly how flavonoid values differ for label 1 and 3 through a box plot.

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

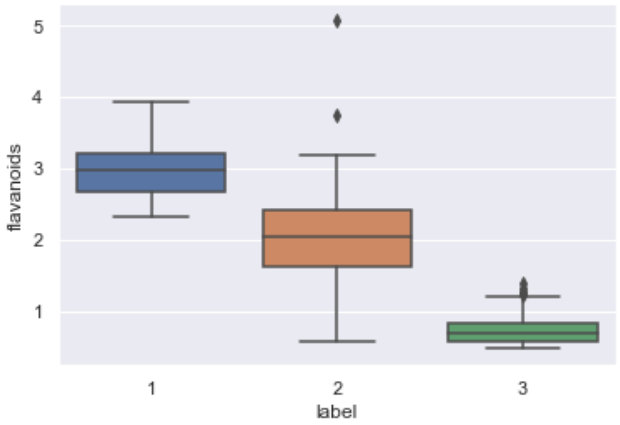

Fig. 5: box plot for flavonoids and its respective labels

We also verified feature importance's/scores from the dataset through Sklearn's Random Forest function called "feature_importances":

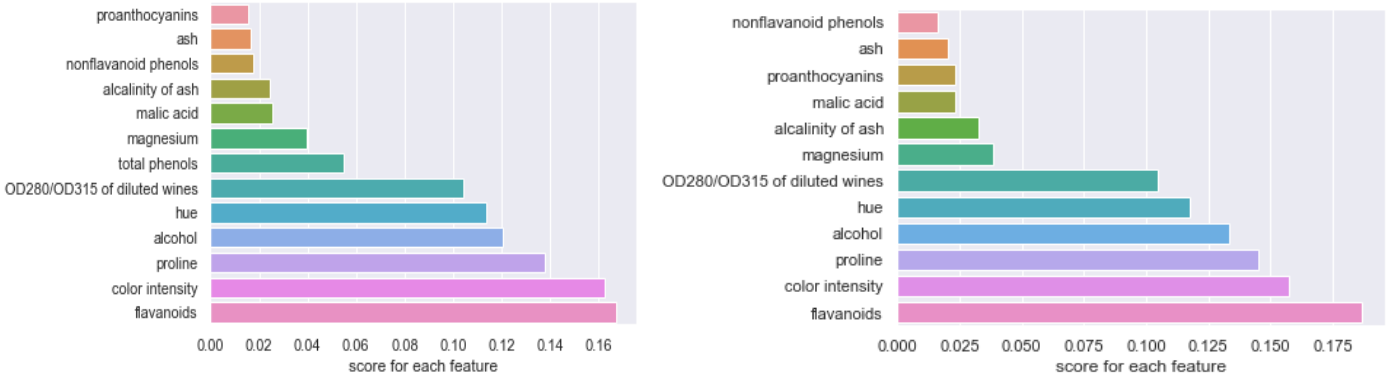

Fig. 6: On the left, features' importance's with 'total phenols' still included.
On the right, importance's without 'total phenols'

As can be seen on Fig. 6, the removal of 'total phenols' did not change the importance order of the remaining features. Also, the disparity between the first six features and the last six features increased.

4. Results

The project started out with an attempt of using K-Nearest-Neighbours, with an accuracy of 69% with KNN = 3 and 66% with KNN = 5. As a result, we transformed the data with Sklearn's *Power transform* function, which utilized the 'Box-Cox' method. We always measured whether the model is overfitting or underfitting through a graph comparing training score and 5-fold-cross-validation score. We chose 5 fold because the number of samples in the dataset is rather small (130), and a higher fold would result in really small testing groups. Every model was tested with and without out 'total phenols' feature based on its high correlation and worse classification properties than 'flavanoids'. Also, models were tested based on the

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

aforementioned “Feature Importance” function from Sklearn’s Random Forest function. We considered only the six features with the highest scores.

The next method applied was Sklearn’s Logistic Regression model with it’s multinomial property (the loss minimised is the multinomial loss fit across the entire probability distribution) and Gaussian Naive Bayes (GaussianNB). Interestingly, both models achieved very similar results across the board with high predictive score (between 97 and 98.01%).

As for the MLP models, we tried several number of nodes and layers (max. 3 layers and max. 10 nodes, which is the number of inputs, layers, minus the number of outputs, labels). Even though we tried 3 layers, one or two layers seemed enough for the problem not only because they had equal or better accuracy, but also because we should always aim for the simpler model. We found the best model to be a single-layer only, with 9 nodes. A lower number of nodes ended up underfitting (i.e. 3 or 4 nodes) and the optimal point seemed to be between 6 and 9 nodes.

Every model was tested with cross-validation with the help of Sklearn’s learning_curve function, which gives us great insight on the model as the number of samples increases over time.

The following graphs (Fig. 7-12) are correspondent to the best method for each model.

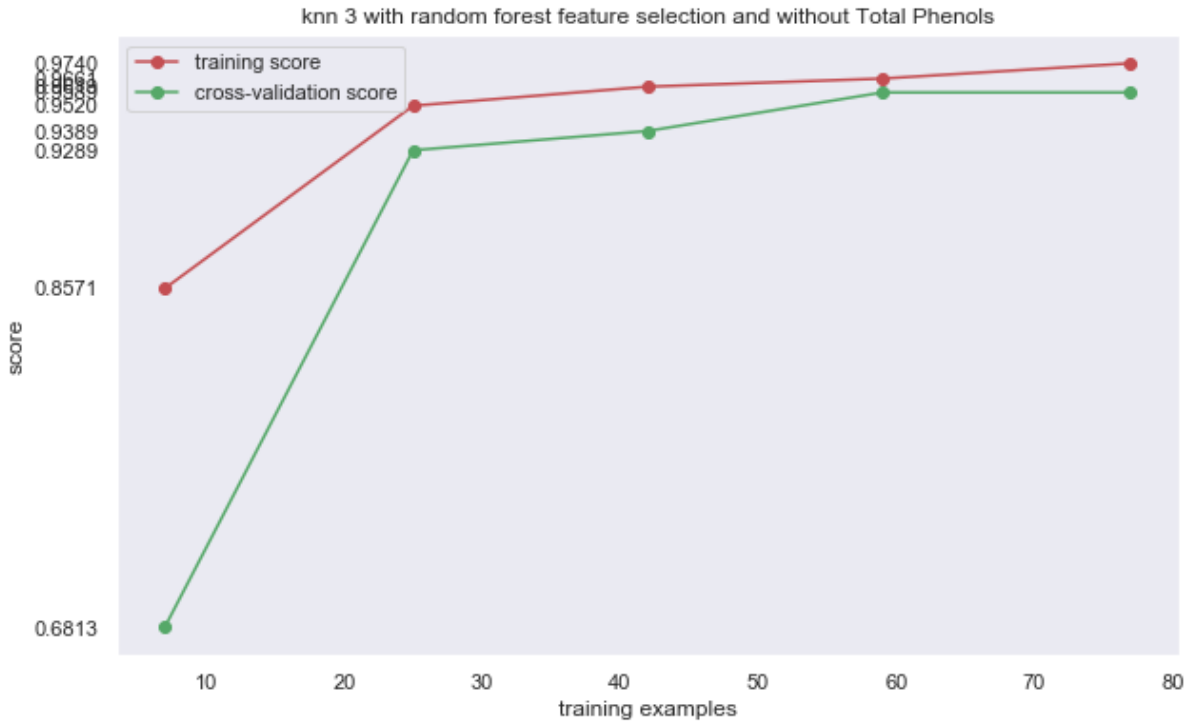

Fig. 7: KNN with N=3 with RF feature selection and without ‘total phenols’, score of 94.88%

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

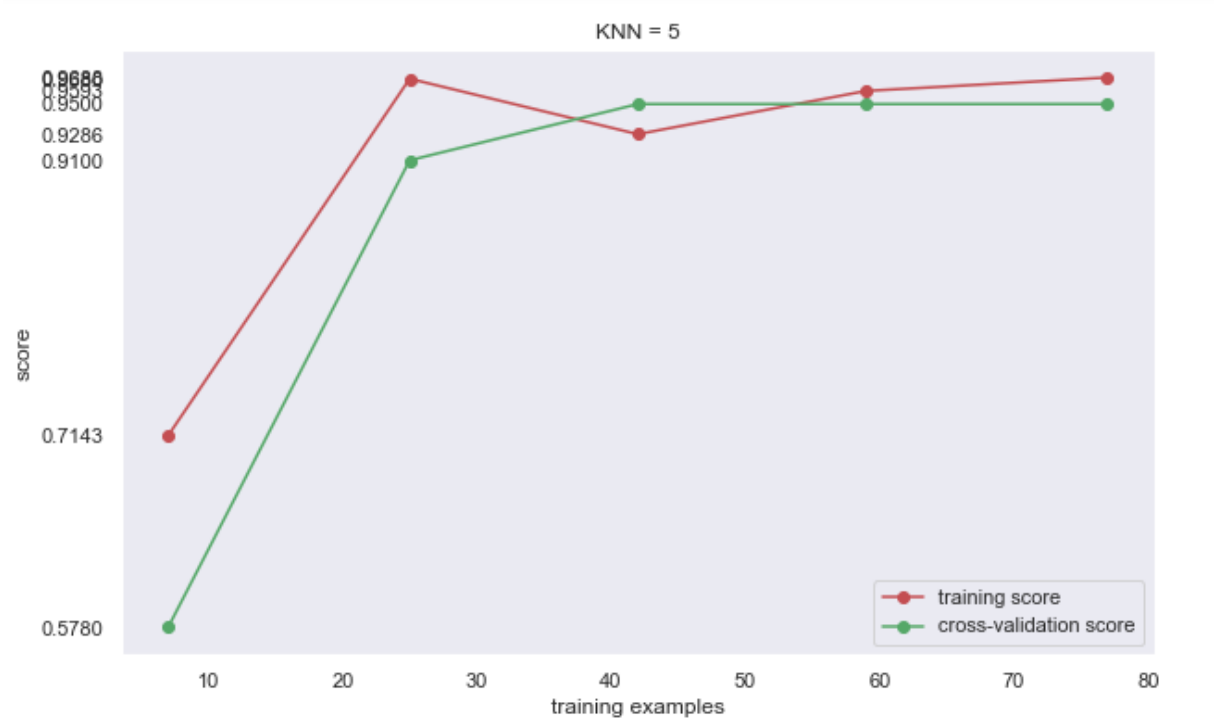

Fig. 8: KNN without 'total phenols' and N=5, score of 95%

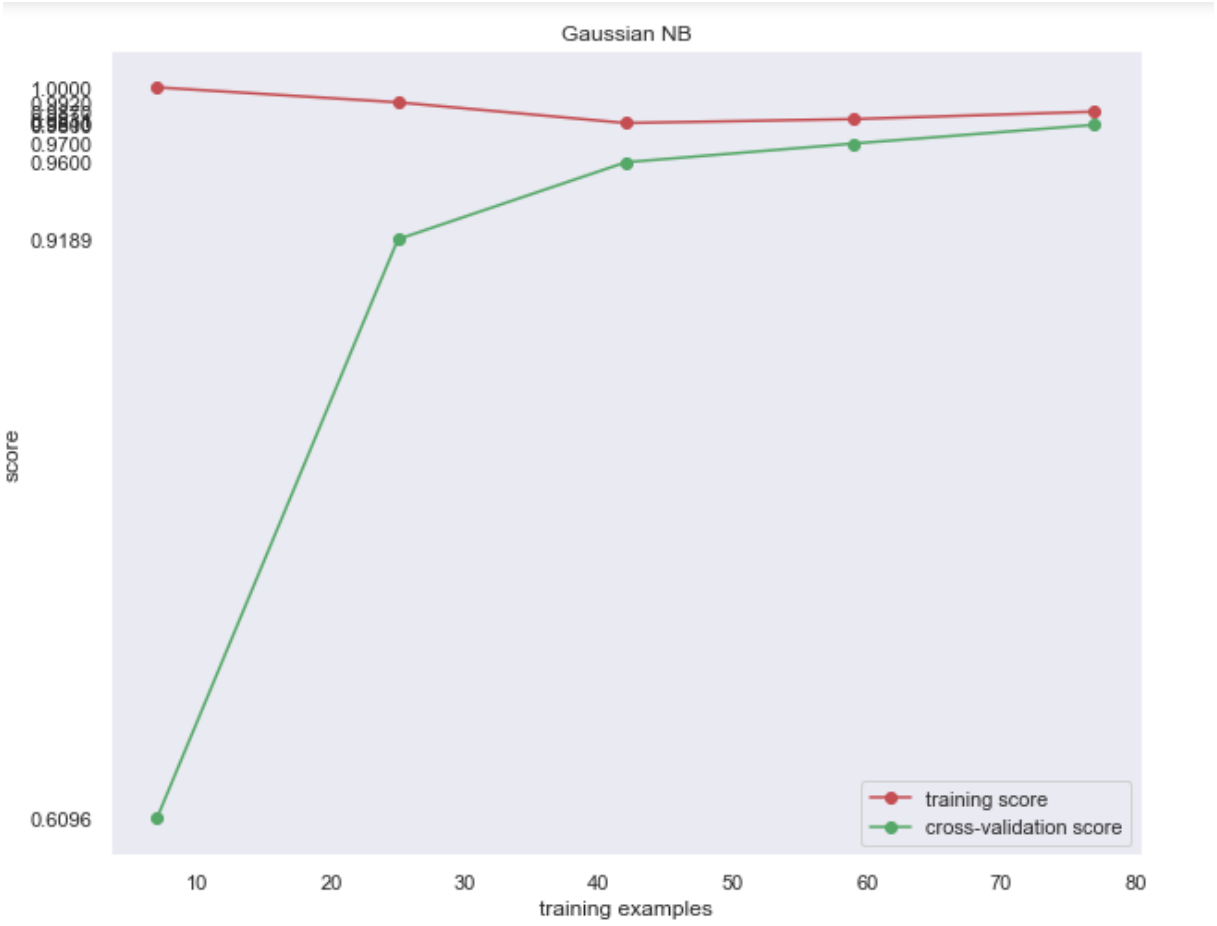

Fig. 9: Gaussian Naïve Bayes without 'total phenols', predictive accuracy of 98%

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

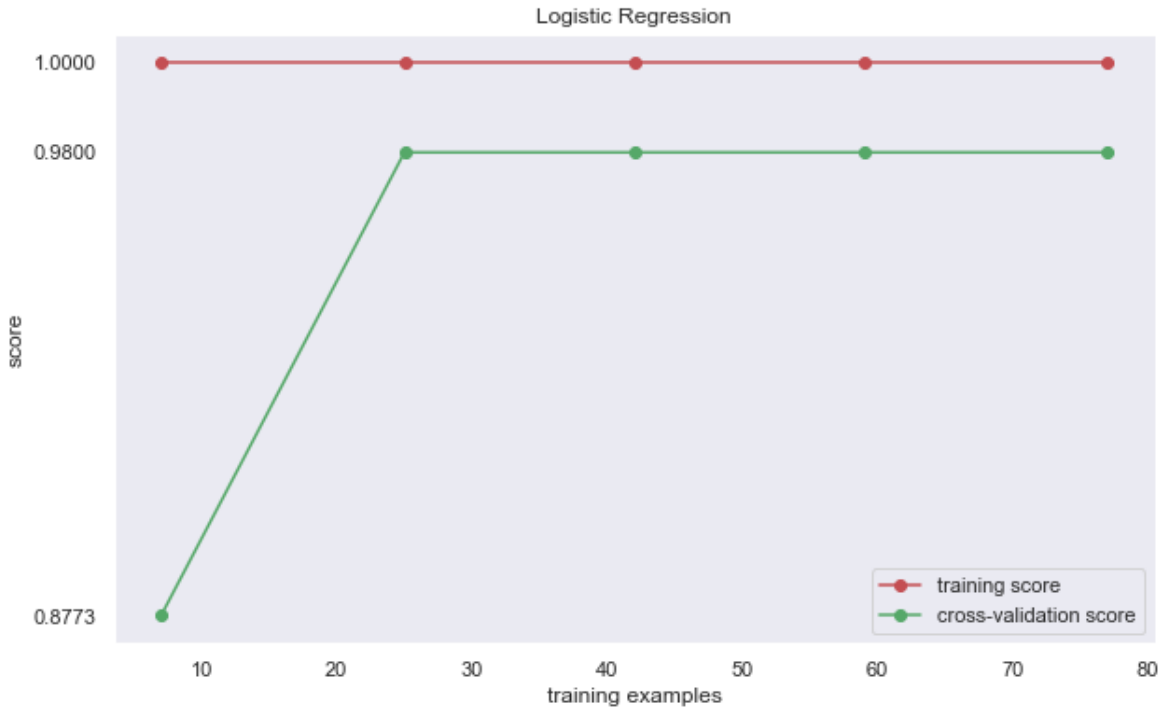

Fig. 10: Logistic Regression without 'total phenols' feature, score of 98.01%

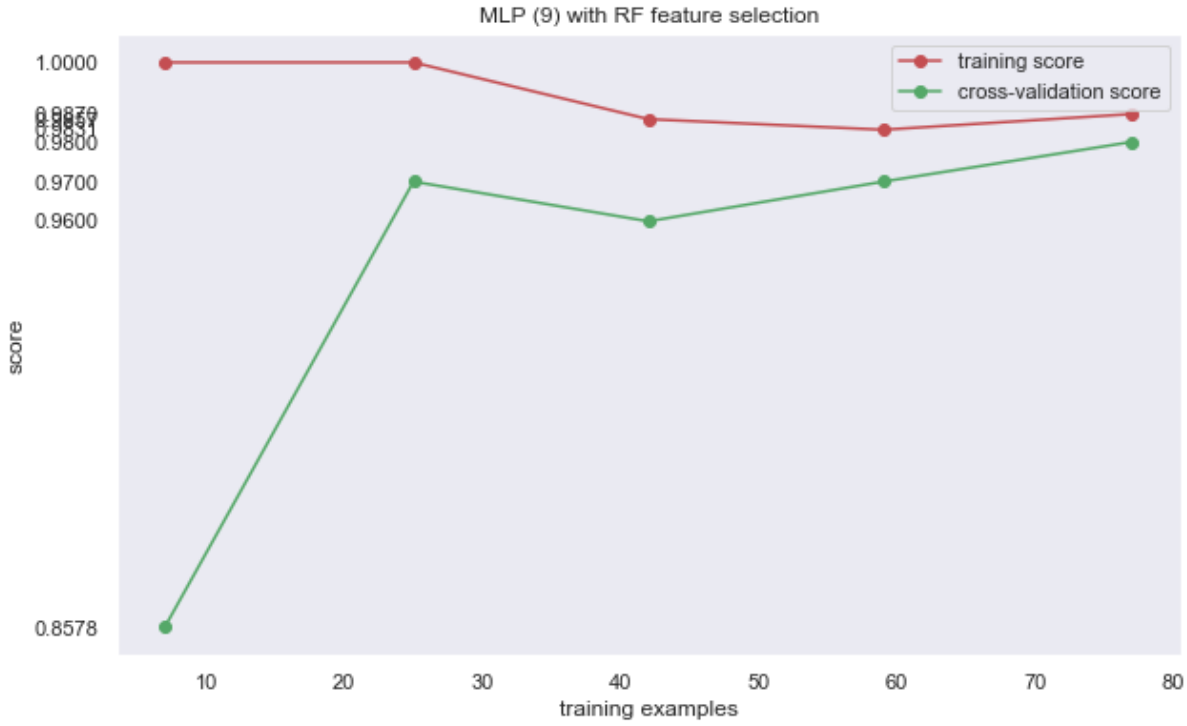

Fig. 11: MLP (9) with RF feature selection and without 'total phenols', score of 98.01%

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

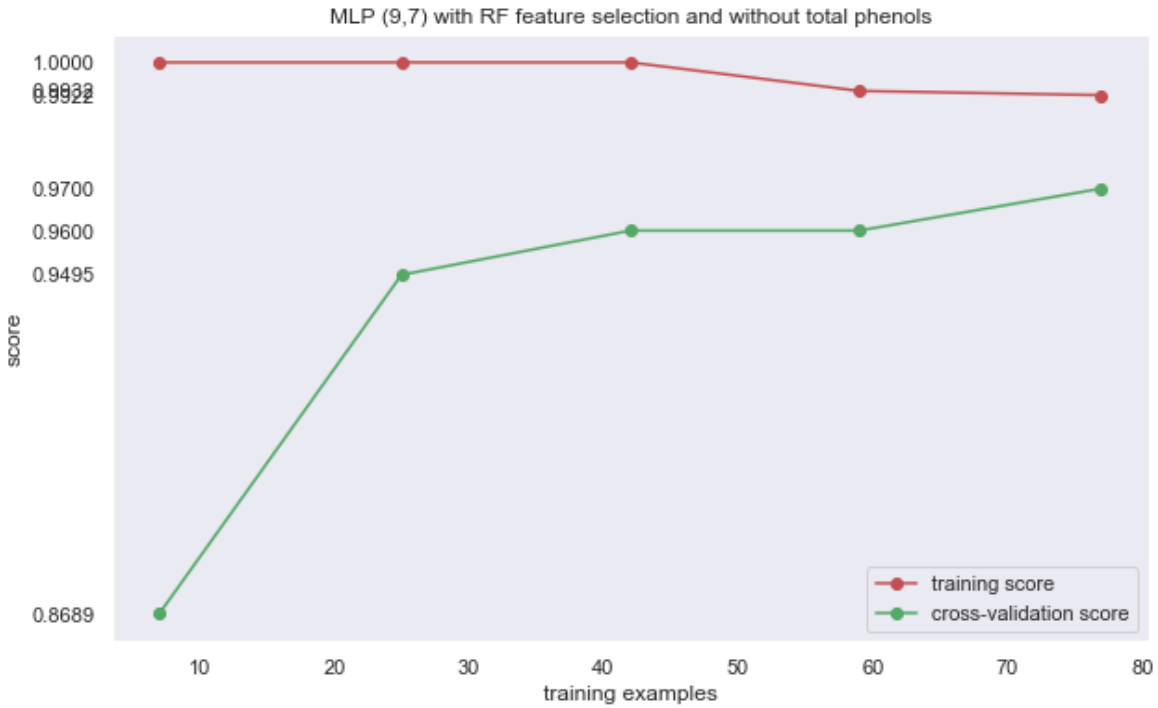

Fig. 12: MLP (9,7) without 'total phenols' and with RF feature selection, score of 97%

All the results were put up together in the Table 2 below.

	Full model	Without 'Total phenols' feature	Random Forest feature selection
KNN = 3	91,99%	93%	**94.88%**
KNN = 5	91,94%	**95%**	94.9%
Logistic Regression	97%	**98.01%**	97%
Gaussian NB	97%	**98.01%**	98%
MLP(9)	95%	96%	**98.01%**
MLP(9,7)	96%	**97%**	**97%**

Table 2. Cross validation (cv=5) for each model with respective feature selection technique.

As said before, there is a strong relation between Logistic Regression model and Gaussian Naïve Bayes models, and Multi-layer Perceptron performed very similar with respect to 1 and 2 layers. Since there were no major improvements in 3 layers and above, they were discarded. The K-Nearest Neighbours models also performed better than expected.

Ankit Jadav, ankitjadav94@gmail.com
Lucas Lucian, lucaslucian471@gmail.com
Fredrik Svanström, fresva16@student.hh.se

5. Discussion

So this project can be considered as a rather simple classification problem, with low number of features and low number of samples. However, it was intriguing that none of the models tested managed to achieve more than 98.01% accuracy, which could indicate that some samples weren't fit for classification algorithms, being nearly impossible to classify accordingly.

More tests could be done with a higher number of samples in order to further verify and increase the models' predictive power, especially Multi-layer Perceptron.