Aalto University
2021/2022

Wine recognition using K-NN methods

Daniel Wohlrath

January 13, 2024

Contents

1 Introduction

A chemical analysis of wines grown in the same region in Italy by three different cultivators was conducted. This analysis was done in July, 1988, and was focusing on thirteen different measurements taken for different constituents found in the three types of wine. These thirteen different attributes are:

- alcohol,
- malic acid,
- ash,
- alcalinity of ash,
- magnesium,
- total phenols,
- flavanoids,
- nonflavanoid phenols,
- proanhocyanins,
- color intensity,
- hue,
- OD280/OD315 diluted,
- proline.

All of these attribute measurements are numeric values with precision up to 0.01 of a unit. The assumption is that the values are characteristic for each of the **three cultivators**. The original dataset is available for example at [1].

If a person is not an experienced sommelier, it is sometimes very challenging to distinguish between a cheaper version of a more expensive wine. It is also not uncommon to try sell bottles of wine during auctions for astronomical prices. In this project we will try to predict to which class (i.e. which cultivator's production) a wine belongs, based on the measurements of the 13 attributes. This could be used for example for clarification of wine's origin and if the seller at the auction is not trying to sell a cheap counterfeit.

In section 2 we explain the process of feature selection, different ML models, loss functions used for learning the hypothesis and computing training/validation/test errors. At the end of the section 2 we discuss the construction of training, validation and test sets. In section 3 we summarize the obtained results and conclude which model is the most suitable one. We support these conclusion with figures and tables. The python code is attached in the appendix.

2 Methods

2.1 Dataset

We further specify that the **datapoints** are samples of wines and the total number of datapoints is 178 (around 50 in each class). Moreover, the possible **features** are the 13 attributes shown in section 1 and they are numeric values. The **label** is a class which is represented by a cultivator. Labels are of categorical nature, that is, we have 3 different cultivators who can be represented as integeres $\{0, 1, 2\}$.

2.2 Process of feature selection

The correlation plot for all different 13 features[1] is shown in figure 1. This figure significantly implies that the dataset is not substantially 13-dimensional as many features are heavily correlated.

Hence, PCA[2] was performed in order to reduce the dimensionality of our problem. First, we scaled the features so the standard deviations are 1 and the means are 0. Then we conducted the PCA transformation with such scaled features. The ratio of total variance explained by the first two principal components is more than 55%. As this number is considered to be sensible in order to reduce the dimensions, we will conduct the rest of the analysis only with these two (scaled) principal components as features.

2.3 Hypothesis space

As we have 3 different classes to classify based on the values of features (see figure 2), we choose to apply the K-Nearest Neighbors method. The number $K \in \mathbb{N}$ of NNs used to determine the function value $h(x)$,

[1] ordered as shown in section 1

[2] Principal Component Analysis

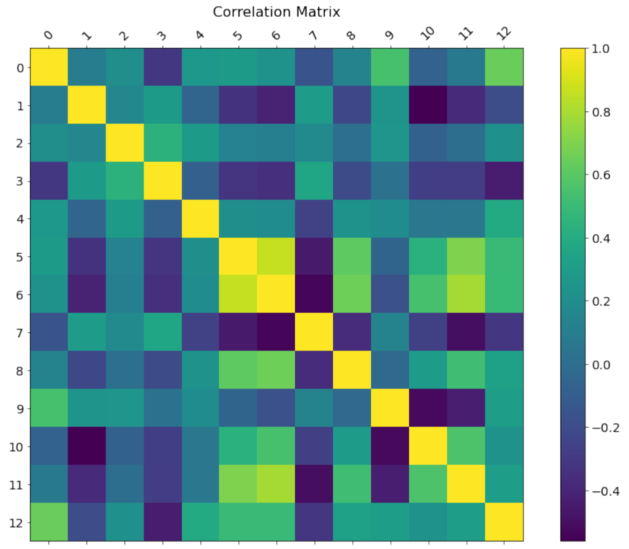

Figure 1: Correlation matrix for original features

where h is an element of the hypothesis space (1), is a hyperparameter of the method. Therefore, different values of K yield different ML methods[3]. In this report we used $K = 3, 5, 7$.

Usage of K-NN method is very sensible since our label space is consisting of finitely many values (categories), i.e. our label space is $\mathcal{Y} = \{0, 1, 2\}$.

K-NN method constructs a hypothesis space

$$\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y} \,|\, h \text{ is piecewise constant}\}, \tag{1}$$

where $\mathcal{X}$ is the feature space, in our case $\mathcal{X} = \mathbb{R}^2$.

2.4 Loss function

In order to learn the hypothesis, we chose the $0-1$ loss function, as this is a very sensible choice and is quite commonly used in problem of similar nature. Also it is recommended in MyCourses webpage.

The $0-1$ loss function is an indicator function of an event of classifying a datapoint incorrectly, mathematically written:

$$L\left((\mathbf{x}, y), h\right) = \begin{cases} 1 & \text{if } y \neq \hat{y}, \\ 0 & \text{else,} \end{cases}$$

where h is the hypothesis that is being learned.

In order to compute training, validation and test errors, we used the most commonly used metric: *accuracy*. It is a natural choice for deciding whether a classifier is performing poorly or not. The higher the accuracy, the better is the model predicting.

2.5 Model validation

We split the data into a training set, a validation set and a test set using the `train_test_split` function (single split) from `sklearn` library. The number of datapoints in each set is given by these following proportions[4]: 40%, 48%, 12% (respectively). This way, we leave a reasonable proportion of the dataset aside for each of the model validation stages, so we can assess the models performance in an easy way. This way of dividing the dataset into training, validation and test sets is a common one, as we want the validation set to be usually around the same size as the training set.

[3] Approved by a TA

[4] The total number of datapoints is given in subsection 2.1

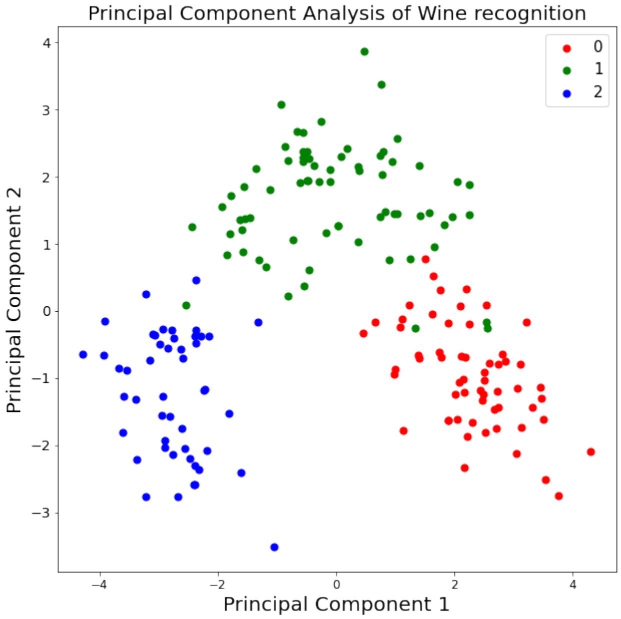

Figure 2: Scatter plot of the whole dataset after PCA

K	3	5	7
Training accuracy	0.9718	0.9437	0.9577
Validation accuracy	0.9765	0.9765	0.9647

Table 1: Comparison of training and validation accuracies for different models

3 Results and conclusion

The performance of different models is shown in the table 1. According to validation accuracies, there are two equally good models, with $K = 5$ and $K = 3$. Both have the validation accuracy 0.97647 but $K = 3$ model has higher training accuracy. That means that it is more capable of predicting the correct labels. The graphical visualization of classification of the validation set using $K = 3$ is shown in the figure 3.

Every model has the training accuracy slightly lower than the validation accuracy, this is probably caused due to the small number of datapoints. If we had more datapoints, we could increase the size of the validation set, which would likely result in different validation accuracies for different model.

Based on the training and validation accuracies, we conclude that $K = 3$ is the best performing model, since the values are the highest. The computed test accuracy[5] for the best method ($K = 3$) is 0.9545. This implies that the model performs outstandingly well. On average, less than 1 out of 20 wine samples is classified incorrectly.

In order to obtain even better performing ML model, we could use another loss function for computing train and validation errors. Even though accuracy is commonly used one, another choices could reveal some hidden relations. Figures 2 and 3 provide hints of existence of clusters. Trying clustering could also lead to interesting discoveries.

[5] it is explained in section 2 how test set is constructed

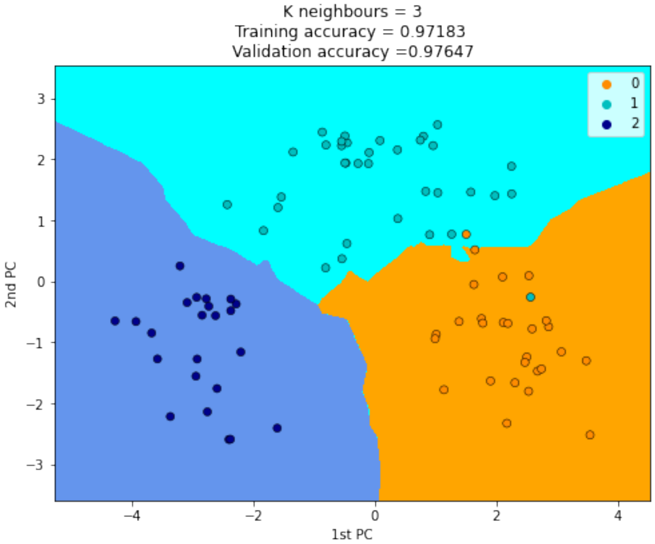

Figure 3: Classification of validation set for $K = 3$

References

[1] Uci ml wine recognition. `https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data`. Accessed: 2022-02-09.