# Lab Work 1: "Feature Engineering"

## General Remarks

- Please **work in groups** of **two or three** people!
- **Clearly mark the code sections that were created with the help of generative AI!**
- Apply train/test data split of **80/20** for every task
- Please use the appropriate sklearn libs

### 1. Report

- Must be **submitted 3 days before** the presentation
- A written project report (pdf) must contain the results of all tasks (task 1, task 2, task 3 and task 4.1) clearly readable
- Submit the code (pdf), commented and easy to read

### 2. Presentation

- Show and execute the submitted code
- Explain the tasks and show their results
- **ppt presentation (15 min.) about the results of your findings**

## Introduction

The overall goal is to investigate and assess different feature engineering techniques, in particular attribute reduction techniques, which can be applied to binary classification problems. Accuracy will be used for the evaluation.

For the first task one starts by assessing the importance of "good" feature. For this the data is manipulated to get a clearer separation of the two classes (Y=0 and Y=1) for one attribute. The goal here is to investigate whether this may lead to an improvement of the Accuracy.

Then two preprocessing techniques are applied to the original data set and the evaluation is performed again and compared with the results obtained without preprocessing.

**After completion you have learned:**

- Study distribution of attributes and its impact on accuracy
- Applying PCA and RFE feature engineering techniques and its impact on accuracy
- Working with grid search
- Using proper sklearn libs

# Data Set

You will use the so-called "Wine Quality Dataset" from Kaggle[1]. It is a publicly available data set that contains 1400 samples of a Portuguese red wine with 11 attributes, i.e. physio-chemical data and a quality ranking. The quality ("Y") is based on sensory data, i.e. not evaluated by humans, and has a range of scores from 1 to 8.

The target Y of the classification task is the quality. To use the quality for a binary classification task, the original data set the quality was grouped into two groups. The first group contains the "bad" wines, all samples with a quality <=5. The second group contains the "good" wines with a quality > 5. Since in most cases it is easier to work with numbers than with strings, the first group is from now on referred to as quality = 0 and the second group as quality = 1. The modified data is stored in the file "**Wine_Test_01.csv**". Fig. 1 shows the first 10 rows of the data set with its 11 attributes and in the last column the changed quality, the target Y, contains only 0s and 1s.

| fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol | quality |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 | 0 |
| 7.8 | 0.88 | 0.00 | 2.6 | 0.098 | 25.0 | 67.0 | 0.9968 | 3.20 | 0.68 | 9.8 | 0 |
| 7.8 | 0.76 | 0.04 | 2.3 | 0.092 | 15.0 | 54.0 | 0.9970 | 3.26 | 0.65 | 9.8 | 0 |
| 11.2 | 0.28 | 0.56 | 1.9 | 0.075 | 17.0 | 60.0 | 0.9980 | 3.16 | 0.58 | 9.8 | 1 |
| 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 | 0 |
| 7.4 | 0.66 | 0.00 | 1.8 | 0.075 | 13.0 | 40.0 | 0.9978 | 3.51 | 0.56 | 9.4 | 0 |
| 7.9 | 0.60 | 0.06 | 1.6 | 0.069 | 15.0 | 59.0 | 0.9964 | 3.30 | 0.46 | 9.4 | 0 |
| 7.3 | 0.65 | 0.00 | 1.2 | 0.065 | 15.0 | 21.0 | 0.9946 | 3.39 | 0.47 | 10.0 | 1 |
| 7.8 | 0.58 | 0.02 | 2.0 | 0.073 | 9.0 | 18.0 | 0.9968 | 3.36 | 0.57 | 9.5 | 1 |
| 6.7 | 0.58 | 0.08 | 1.8 | 0.097 | 15.0 | 65.0 | 0.9959 | 3.28 | 0.54 | 9.2 | 0 |

Figure 1: Excerpt of the modified data set with the now binary attribute "quality".

# Task 1: Classification with the Unmodified Attributes

## Tasks

a) Display a histogram for each attribute
b) Show two histograms for each attribute with respect to Y. Please **comment** on the expected performance of the classifier. Please **explain** how you came to this conclusion.
c) Perform **10 runs** of modeling and test. In **each run**, split the dataset **randomly** into a training and test set (somewhat like k-fold cross-validation).
   - For each run a grid search with 10 different sets of hyperparameters, employing the Support Vector Machine (SVM), should be performed.
   - Display the best hyperparameter for each run.
   - With the best hyperparameter for each run, display the obtained Accuray of the **test data**.
   - Display the average accuracy with its respective standard deviation.

[1] https://www.kaggle.com/datasets/yasserh/wine-quality-dataset

# Task 2: Density Analysis

This task does not deal with attribute reduction but should give you a feeling how important "good" features are. For this, you delete some data samples of the given data set "**Wine_Test_01.csv**" such that a clear distinction between the two classes (Y=0 and Y=1) for one attribute is achieved. Following this data sample deletion, the obtained performance of the classifier should be studied and compared with task 1.

## Basic knowledge and approach

Let's consider the two attributes *fixed acidity* and *volatile acidity*. Fig 2 shows the distribution of both attributes regarding Y = 0 and Y = 1. We notice that there is an overlap for both classes for each attribute. After deleting some data samples, the distributions show a stronger separation (see Fig. 3) for the attribute *volatile acidity*.

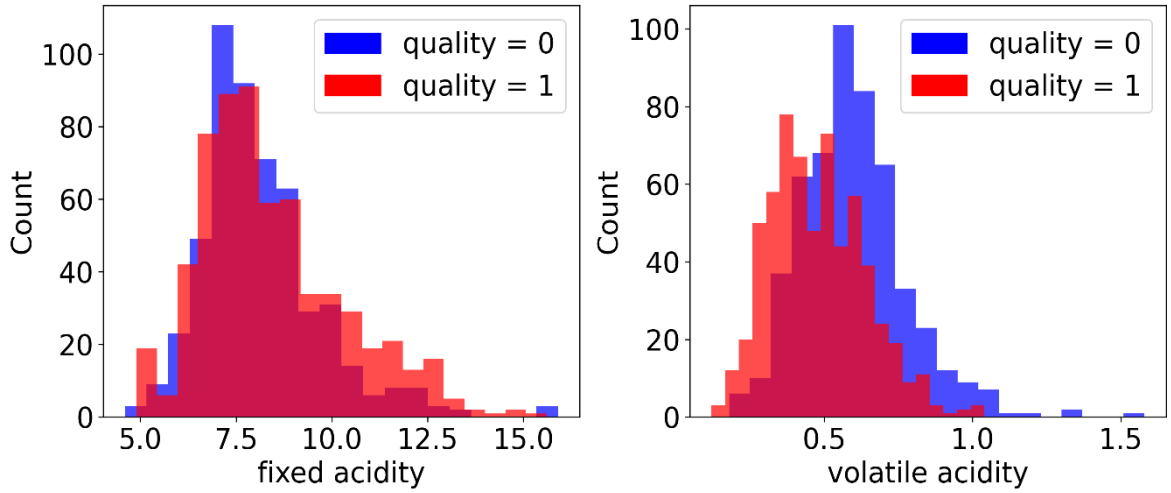


Figure 2: Distribution of Y=0 and Y=1 for the two attributes. The chosen attributes have a large overlap.

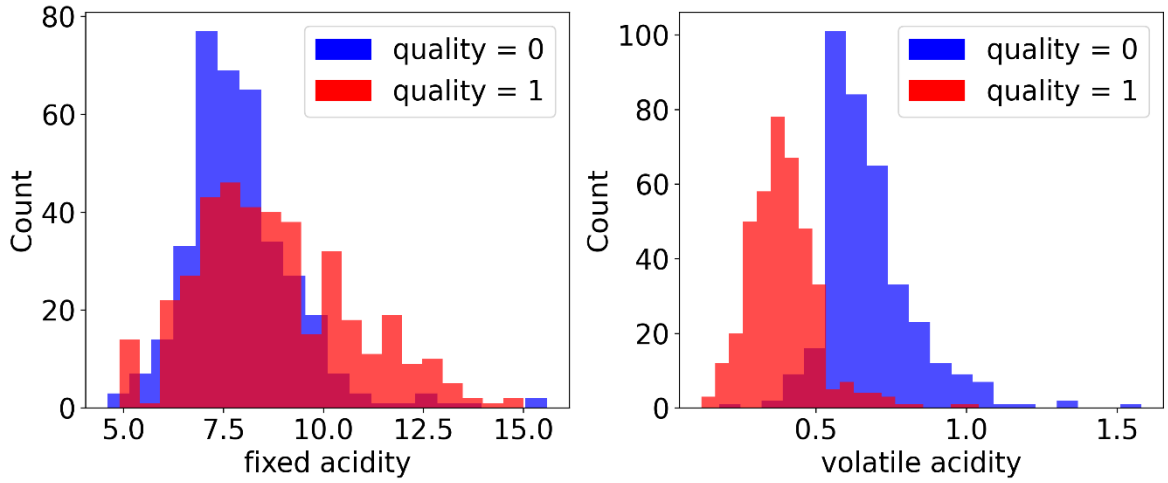


Figure 3: Distribution of Y=0 and Y=1 for the two attributes after data reduction. Only one of the two attributes shows a large overlap.

## Tasks

a) Choose **one attribute** which has a significant distribution overlap of Y = 0 and Y = 1.
b) Delete data samples from the data set to separate the histograms of the two classes for one attribute (see Fig. 2).
c) With the modified data set, redo Task 1 c) and comment on the new performance.

# Task 3: Principal Component Analysis

The goal for this task is to study the Principal Component Analysis (PCA) technique to be used for a binary classification problem. Apply the PCA to the full data from the file "Wine_Test_01.csv". With the attributes of the new space, apply then the same task as stated in Task 1 c).

## Basic knowledge and approach

PCA is an unsupervised technique for analyzing the linear combination or relationship of data which are in our case the attributes. It provides a useful method for reducing the attribute space when the attributes are highly correlated. The goal of PCA is to eliminate the correlated attributes. The dimensionality is reduced by transforming the attributes into new attributes (features) called Principal Components (PC). The main aim of PCA is to represent the information in the data with as few attributes as possible[2,3]. The correlation with the target is not considered. It is therefore possible that an attribute with a high correlation to the target value is sorted out with the impact that the classification quality gets worse. As the PCA process contains numerically vector/matrix operations, scaling/standardization of the row data set must be done beforehand.

## Tasks

The PCA technique to identify the PCs should be applied on the **train data** set only. After deciding about the number components (PCs) transform the train data and the test data into the new PC Space.

a) Apply a PCA and display "number of components" vs "cumulative variance", s. Lecture example[4,]. Using the proper classes from the sklearn Libs[5,]. From the curve "number of components" vs "cumulative variance" choose the number of components to be used.

[2] https://www.machinelearningplus.com/machine-learning/principal-components-analysis-pca-better-explained/
[3]https://en.wikipedia.org/wiki/Principal_component_analysis
[4] jaeger-recomender-final-student-2025.pdf
[5] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

b) With the found number of PCs obtained from a) transform the test data into the new PC space, i.e. apply PCA and then perform modeling by employing SVM with a grid search.
   - Perform a grid search with 10 different sets of hyperparameters, employing the Support Vector Machine (SVM).
   - Display the best hyperparameter.
   - Calculate and display the accuracy obtained with the transformed test data.
   - Compare the results to the result of task 1.

# Task 4: Recursive Feature Elimination

The goal for this task is to study and assess one recursive feature elimination (RFE) technique to be used for a binary classification problem. After applying the RFE technique, study the obtained performance of the classifier. Again, the full data from the file "Wine_Test_01.csv" should be used.

## Basic knowledge and approach

RFE is a technique to analyze the importance of attributes (features) and delete them for the following classification. Thus, it might be a useful task for e.g., reducing the feature space thus saving CPU power. Additionally, the reduction of the feature space may improve the performance of the model.

The RFE technique should be applied to the **train data** set only. After finding the most important attribute by applying the RFE you must make sure that your **test data** are in the **same attribute space** as the **training data**.

RFE is an iterative process with the goal of systematically removing the least important attribute[6]. It **uses the "Y"** and therefore the process requires a classifier, i.e., e.g., SVM. This is a significant difference to the PCA technique, where the "Y" is not used. Make sure to use the same classifier chosen for the RFE for the following classification task. Use the proper classes from the sklearn Libs[7].

## Tasks 4.1

a) Choose an RFE estimator and its parameter,
   e.g., "estimator = svm.SVC(kernel='linear' , C = 1000, gamma=0.1)"
b) Select features, e.g. "selector = RFE(estimator, n_features_to_select=8)"
c) Apply a RFE and by using the proper classes from sklearn get the names of the attributes. Display them.
d) With the reduced number of attributes obtained from c) perform modeling with the same estimator chosen for RFE.
   - Perform a grid search with 10 different sets of hyperparameters, employing the Support Vector Machine (SVM).

[6] https://en.wikipedia.org/wiki/Feature_selection
[7] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

- Display the best hyperparameter.
- Calculate and display the accuracy obtained with the test data.
- Compare the results to the result of task 1.

## Task 4.2 (Optional)

Perform a RFE technique where the “most important attributes” are calculated. Try to find a RFE solution where the optimal number of attributes are calculated. Use any help of the internet, but you need to cite the source. Again, repeat the steps laid out in Task 1c and display the achieved accuracy. Compare these results to the ones obtained in Task 1 and comment on them.

RJ/02/2025