İstanbul
Bilgi Üniversitesi

CMPE 407 - Machine Learning Project

by

Onur Çalışkan, 119200059
Emine Esin Yilmaz, 120200059

Supervised by

Özgür Özdemir

Submitted to the
FACULTY OF ENGINEERING AND NATURAL SCIENCES
in partial fulfillment of the requirements for the

Bachelor of Science

in the
COMPUTER ENGINEERING

30.05.2023

Abstract

In this study, the use of machine learning methods in wine quality prediction was investigated. Using a wine dataset, various preliminary steps were performed and quality estimation was made using different machine learning models .

Table of Contents

1 Introduction

1.1 Aim of the Project

In this study, a machine learning project was carried out with the aim of predicting wine quality. Our dataset includes 1599 wine samples containing a combination of various chemical properties. In the first step, the data set was examined in detail and preliminary steps were applied. These steps included ,detecting and correcting outliers, data normalization, and attribute selection. Then, model training was performed using different machine learning algorithms.

These algorithms included classification methods such as support vector machines (SVM), decision trees, and random forests. During the training process, the performance of the models was evaluated using the k-fold cross-validation method and the best performing model was selected. The results showed that the selected model was able to successfully predict wine quality. The accuracy of the obtained model was calculated as 0.74 and stability analysis was performed.

In addition, the impact of attributes on wine quality was also assessed and the most important attributes were identified. This study highlights the availability of machine learning techniques in wine quality prediction and the importance of data analysis and preprocessing steps. The results obtained provide valuable information that can potentially be used in winemakers' quality control and in providing consumers with a better product.

1.2 Dataset Information

In this project, the data set used for wine quality prediction is available in the UCI Machine Learning Repository called the Red-Wine-quality-dataset [1]. This data set includes the chemical properties of the red wines. During our investigations, searches were conducted on Google Scholar to find a suitable data set for our project by referring to relevant research.

As a primary reference, the basic information and characteristics for the data set used in our project are based on information contained in a paper published by the IEEE [2]. This source is research that provides the original source of the data set and is the main basis of our project for making comparisons.

1.3 Methods

The classification method has been a viable option as a model choice used in wine quality estimation. Wine quality estimation is a problem aimed at determining the quality class of a particular wine sample. Therefore, the classification method provides a natural fit to such a forecasting problem. Our dataset includes the chemical properties of wines. These characteristics provide the necessary information for a classification problem, in which wines are correctly assigned to quality categories.

Classification algorithms train on such data sets, analyzing input characteristics and predicting each wine sample into quality classes. Also, the classification method is useful in terms of interpreting and evaluating the results obtained.

Classification algorithms assign each wine sample to a specific quality class, which allows the results to be presented more clearly and unambiguously. Also, the classification method helps us understand the differences between different quality categories and determine which characteristics are more important. For these reasons, the classification method was chosen as a suitable model selection for wine quality estimation.

By using characteristics and labeled quality classes in the dataset, classification algorithms were trained and as a result, quality class estimates were obtained for each wine sample. These estimates provide a valuable source of information for winemakers and consumers, contributing to quality control and product selection. In this direction, various models such as DecisionTreeClassifier, RandomForestClassifier and SVC were used to get the best results and the model that gave the most consistent results was selected according to the results obtained and compared with the referenced research results.

2 Dataset

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Figure 1: Features of the dataset

The table above contains the features of the dataset.There are 12 features in the data set and the 'quality' feature will be used as the target.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

Figure 2: Details of the dataset

The table above contains detailed information about the data set. There are a total of 1599 wine data in the data set. There are no non-null values in the data, and eleven features are of type 'float64'. The target feature 'quality' takes an 'int64' value from 3 to 8.

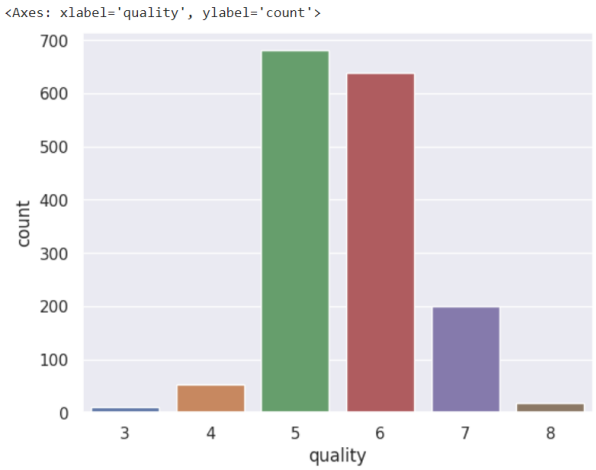

Figure 3: Quality distribution

The plot above gives the 'quality' distribution of 1599 wine data.The majority of wines were rated 5 and 6.

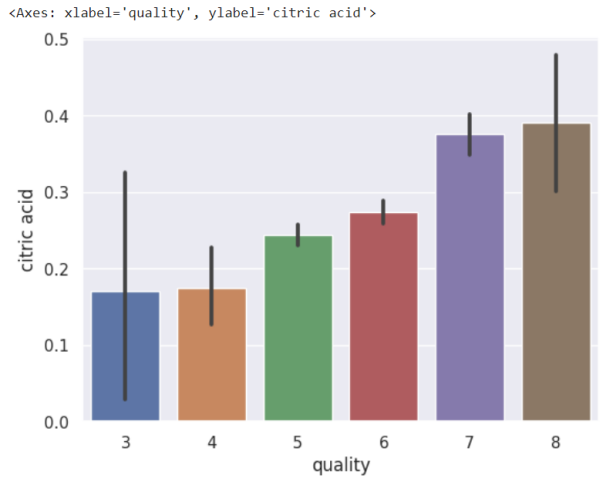

Figure 4: Citric acid vs Quality

As can be understood from the graph above, there is a direct proportion between the amount of citric acid and the increase in wine quality.

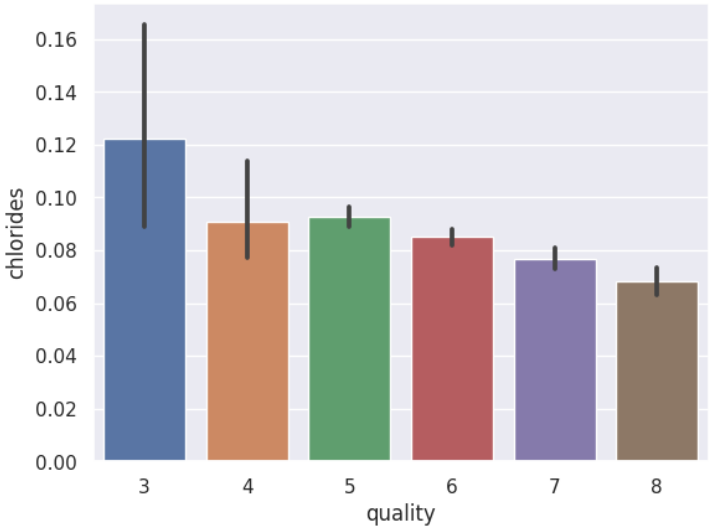

Figure 5: Chlorides vs Quality

In the graph above, it was observed that the ratio of Chlorides quantity to wine quality was inversely proportional.

3 Preproccessing

A preliminary stage is required before the dataset can feed models accurately. As a first step in this direction, after checking whether there is a null value in the data set, the 12'th attribute 'quality' is labeled as the target feature.

3.1 Normalization

Normalization is performed by using the following formula for each sample value in the data set:

X = (X - X.mean()) / X.std()

The purpose of normalization is to make the values in the data set directly comparable to each other. In this way, each trait or variable can contribute in a balanced way in terms of the power of the model to influence their weight during the training process. It can also help train the model faster and better.

3.2 Feature Selection

After the normalization phase, feature selection methods will be applied to select the best features which are the most effective for predicting the target. Firstly, a manual selection method was applied with k=5 (to select the best 5 features).

After that, an automatically selection method was applied using DecisionTreeClassifier model. As result of automatically selecting, there were 4 features selected. 5 out of 4 features were the same as the automatic selection, because of that, the automatic method's results were decided to be used.

These features are in the figure below

```
Selected features:  [False  True False False False False  True False False  True  True]
Selected features:  ['volatile acidity' 'total sulfur dioxide' 'sulphates' 'alcohol']
```

Figure 6: Most effective features

4 Experiments

4.1 Classification Algorithms used in Project

As mentioned in the Introdution section, it was decided to use the classification method for wine quality estimation. Accordingly, it was decided to use DecisionTreeClassifier, SVC, RandomForestClassifier and Naive Bayes models to make classifications.

1) Naive Bayes: Naive Bayes is often the preferred algorithm in classification problems. This model works based on Bayes' Theorem and uses a probability-based approach to make classifications. Naive Bayes can be simply and quickly trained, and generally perform well with small to medium-sized datasets. Also, the computational cost of the model is low, which is advantageous when working with large data sets. For a classification problem such as wine quality estimation, the Naive Bayes model can produce effective results, especially where the data set is linearly separable

2) SVC (Support Vector Classifier): SVC is a classification algorithm belonging to the family of support vector machines (SVM). This model aims to find the hyperplane that will best distinguish between classes. SVC can handle high-dimensional datasets well, which is effective in classification problems. SVM can also be used when the data set is not linearly separable thanks to kernel functions. The SVC model for a classification problem, such as wine quality estimation, can also produce effective results in cases where the data set cannot be separated linearly or where there is a complex decision boundary.

3)DecisionTreeClassifier:

DecisionTreeClassifier is a classification algorithm based on decision trees. This model classifies the data set by dividing it and creating decision nodes. Decision trees are easy to understand and interpret, and can provide an order of importance for features added to the model. For a classification problem, such as wine quality estimation, the DecisionTreeClassifier is especially useful in situations where we want to identify and understand the relationships between traits in the data set. In addition, this model can also be resistant to outliers or missing data in the data set.

4) RandomForestClassifier:

RandomForestClassifier is an ensemble learning algorithm created by combining many decision trees. This model creates multiple decision trees with random sampling and classifies the results by combining them. RandomForestClassifier can be resistant to overfitting and generally performs well. For a classification problem, such as wine quality estimation, RandomForestClassifier can be used to rank the importance of traits in the data set, increase the diversity of decision trees, and make a more accurate and stable classification.

4.2 Cross Validation

K-Fold Cross Validation is a method used to accurately evaluate and generalize the performance of a model. It allows us to obtain more reliable results than the methods of dividing the data set as training and test data.

This method divides the data set into K equal parts (folds) and uses one shard as test data for each fold and the rest of the pieces as training data. This process is repeated K times and a different part is selected as the test data for each iteration. As a result, K-Fold Cross Validation provides a cross-validation process in which all samples in the data set are used as both training and testing data.

To use this method in the data set, a cv was determined as:

cv = KFold(nsplits=10, shuffle=True, randomstate=7) before training the model.

This code uses the K-Fold method to perform the Cross Validation operation. "cv = KFold(nsplits=10, shuffle=True, randomstate=7)"

creates a KFold object for the K-Fold Cross Validation operation.

The parameter 'nsplits=10' indicates that we will divide the data set into 10 parts. In this case, the Cross Validation process will consist of ten folds.

The shuffle=True parameter causes the data set to be shuffled in each fold. This ensures that different samples are selected in each fold if the samples in the data set come in sequential order. This prevents the model from

becoming dependent on sequential data and can make better generalizations.

The parameter 'randomstate=7' ensures that the mixing process is repeatable. When the same 'randomstate' value is used, the same mixing process is performed on each run, thus ensuring that the results are reproducible. This code is used to start the Cross Validation process.

4.3 Training

Cross Validation technique was applied with the selected best features and 4 different models were trained. As a result of these train stages, the average accuracy values of the models are given in the figure below.

```
Mean of experiment scores of Decisiontree: 0.6447798742138365
Mean of experiment scores of GaussianNB: 0.5778498427672956
Mean of experiment scores of BernoulliNb: 0.4315369496855347
Mean of experiment scores of RandomForest: 0.7117020440251572
Mean of experiment scores of SVC: 0.49405660377358496
```

Figure 7: Training results

To increase the accuracy of the models and to find the best parameters for the models, Hyper parameter Tuning technique was used.

Hyperparameter Tuning is a process used to improve the performance of machine learning models. Hyperparameters are parameters that control the structure and behavior of the model, which remain constant during training. Examples might include hyperparameters such as the maximum depth in a Decision Tree model, parameter C in a Support Vector Machine model, or the number of trees in a Random Forest model.

Hyperparameter Tuning uses trial and error to find the best values of hyperparameters. In this process, models are created by experimenting with possible combinations from a specific range of hyperparameters or sets of values, and their performance is evaluated. The goal is to find the best combination of hyperparameters based on a specific performance metric (e.g., accuracy, precision, or F1 score).

Hyperparameter Tuning has several different approaches. Grid Search tries to get the best results by trying all combinations in a given hyperparameter range or set of values. Random Search evaluates performance by selecting random combinations within a given range of hyperparameters or

set of values. Bayesian Optimization uses a probabilistic-based approach to explore the hyperparameter lookup field more efficiently compared to trial and error methods.

The goal of Hyperparameter Tuning is to improve the performance of the model and improve its generalization capability. Incorrect hyperparameter values can cause the model to encounter generalization problems such as overfitting or underfitting. Therefore, finding the optimal values of the hyperparameters is critical to achieving the best performance of the model.

4.4 After Hyperparameter Tuning Results

```
Accuracy (tuned): 0.7421383647798742
Precision (tuned): 0.7035878458013068
Recall (tuned): 0.7421383647798742
F1 score (tuned): 0.7164401016646565
```

Figure 8: Tuned Accuracy of RandomForestClassifier

```
Best Estimator: RandomForestClassifier(max_depth=10, max_features='log2', min_samples_split=5,
                       n_estimators=200)
Best Parameter: {'n_estimators': 200, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'log2', 'max_depth': 10}
```

Figure 9: Best parameters for RandomForestClassifier

5 Results

Model	Best Parameters	Accuracy Before tuning	Accuracy After tuning
SVC	C': 10, 'gamma': 'scale', 'kernel': 'linear'	0.49	0.64
RandomForestClassifier	n_estimators': 200, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'log2', 'max_depth': 10	0.71	0.74
DecisionTreeClassifier	max_depth': 4, 'min_samples_leaf': 7, 'min_samples_split': 2	0.64	0.61
NaiveBayes(Gaussian)	var_smoothing': 1e-05	0.58	0.64

Figure 10: Accuracy and Parameter values of the models

As seen in the table above, the model that made the most consistent prediction was the RandomForestClassifier model. The precision, recall and f1-score values of this model were also compared both after tuning and tuning

The reasons why the RandomForestClassifier model gives the most effective results are as follows:

1) Random Forest is created by combining multiple decision trees. This method allows each tree to make its own prediction, and then combine these estimates to form the final estimate.

2) Random Forest allows each decision tree to operate independently within itself. Decision trees have the ability to learn complex relationships and interactions.

3) Random Forest is also effective at managing outliers. Because they are created by combining multiple trees, outliers will not have a large impact on a single tree.

6 Comparison (Bonus)

Model	Accuracy in this study(tuned)	Accuracy in IEEE research paper
SVC	0.64	0.67
RandomForestClassifier	0.74	0.65
NaiveBayes(Gaussian)	0.64	0.559

Figure 11: Accuracy and Parameter values of the models

In the table above, the accuracy values of the feature selection, cross validation and hyperparameter tuning methods in the study were compared with the results of the research [2] based on the study.

It was concluded that the methods applied made a positive contribution to the consistency of the estimation result.

7 Conclusion

In this study, a red wine dataset was used consisting of 12 features and 1599 instances. The aim was to predict the quality of the red wines. Since the quality values are discrete and labeled, classification methods and models(SVC,RandomForestClassifier,DecisionTreeClassifier,Naive Bayes) were used.

Preprocessing methods were applied in order to achieve the best results and the less processing time. Hyperparameter Tuning technique was used to increase the performances of the models. Before and after the tuning, the accuracy values of all models were examined.

The best accuracy was achived with the RandomForestClassifier model. Being the best model, the precision, recall and f1-scores of the RandomForestClassifier model were also examined. After applying parameter tuning, all of the scores of the model have been increased.

In order to improve results, collecting more labeled data can be applied to the study. While increasing the size of the set, the models can capture more patterns and generalize better.

8 References

Sources and data set used during the research

[1]UCI Machine Learning Repository, Wine quality data set, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

[2]https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=9104095

[3]https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[4]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifi