CSIT456_01

Final Project Report

Project Title: Quality of Wines exported from Portugal

Team Members: William Bautista and Teresa Miller

---

1. **Description**
    a. **Basic Information**
        i. As avid wine enjoyers, we wanted to find out if it was possible to determine the quality of wine based on a certain chemical makeup -- meaning can you predict if a certain wine is good quality without ever even tasting it? To answer our question, we used machine learning to find out.
    b. **Project Objectives**
        i. Our project's objective was verifying if it was possible to determine wine quality through physicochemical tests based on the certain chemical makeup of the wines by applying machine learning algorithms to a public wine dataset.
    c. **Description of the Data Set**
        i. Through the UCI ML Repository,* we selected a dataset detailing vinho (wine) samples from the north of Portugal. The dataset consists of two CSV files for two different types of wines. One CSV file contains data about red wine while the other contains white wine. Both of the files contain information regarding the quality of the red and white archetypes.

These twelve archetypes included fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, Density, pH, Sulphates, alcohol level, and quality. Although the dataset we selected included both white and red wine data, we decided to test only the red wine for a more focused analysis.

2. Data Analysis using Machine Learning
    a. Data Preparation
        i. To start, we split and loaded the data for red wine by parsing through the CSV file. Luckily, the dataset we found had been cleaned up prior to loading it so it did not require any additional cleaning or fixing such as finding missing data in any columns, odd inputs, or strange/unneeded data. However, after studying and running a more descriptive and explorative analysis of the dataset, we noticed that the dataset contained several outliers which we knew were important to remove since they would most likely affect the performance of our machine learning algorithms. We used the sklearn preprocessing module to normalize our data in order to have a more defined accuracy in our predictions by standardizing the datasets features through removing the mean and scaling to unit variance. Finally, we checked the correlation between the datasets columns to better understand the relationship between the eleven features and quality. As part of our data preprocessing step, we began by setting our target variable, Y, to “quality” and then splitting the dataset into training and test

sets (80% and 20% respectively).

b. Model Selection

   i. For our project, we decided to use classification to analyze our dataset with our two specific models being Logistic Regression and K-Nearest Neighbors. We decided to use these two supervised learning algorithms as it best suited our dataset due to the fact we desired to analyze the predictions of discrete target variables as our most important feature, quality, was an ordered, categorical, and discrete variable. We chose Logistic Regression as it does not have a hyperparameter to tune and that it is an effective classification algorithm for binary classification tasks. As for the reason we selected K-Nearest Neighbors, KNN's was a nonparametric model so we know that it wouldn't make any unnecessary assumptions and only deal with the data we gave it. Additionally, KNN's hyperparameter, n_neighbors, which needed to be tuned for more accuracy was done with ease.

c. Model Implementation

   i. Although Logistic regression is a binary classification algorithm, which can be used to answer questions such as Yes/No, True/False, etc, we were able to use and implement the model in this case through multi-class classification because in our dataset there are 5 classes for quality to be predicted as. We set the regressions parameters for multi_class to

’multinomial’, solver to ’newton-cg’. Then predicted the class labels for samples in X. We then applied the classification report metrics to evaluate. After applying and getting a classification report for our dataset, we observed that the accuracy score was .625, which is a bit low so we went ahead and computed the Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores resulting in a higher score of .73. As for tuning our hyperparameters, Logistic Regression did not really have any critical hyperparameters to tune so there wasn't much else to do other than what we already implemented. As for the implementation for K-Nearest Neighbors, we decided to use it because it was simple to apply and easy to tune its hyperparameter as well. To start off we imported the KNeighborsClassifier to select n_neighbors. We then use the train set to fit it into KNN. We received an accuracy of .498 or 49%. This accuracy was about 50% which was alright. Since we wanted to improve the accuracy, we had to tune the hyperparameter n_neighbors. Using a StratifiedKFold technique, we received a cross validation score of .625 and a n_neighbors of 1. Using the new value of n_neighbors in the KNN model, we received an improved accuracy of .611. This is a 61% accuray with about a 11% increase in accuracy from the original model.

3. Evaluation

    a. Although building our machine learning models were successful, we needed to evaluate our models as well as it was an important step in the analysis process. We were able to evaluate our two classification models by applying and analyzing evaluation metrics in order to see the accuracy of how many of our predictions were correct. We applied precision metrics to our Logistic Regression model in order to see how good our model is when the prediction is positive as well as recall metrics to measure how good our model was at correctly predicting positive classes. Our results indicated that our model was able to indeed correctly predict positive classes as it was closer to 1 than 0. In order to take into account false positive and false negative classes, we also applied F1 score metrics which is the weighted average of precision and recall. Additionally, the ROC curve/AUC metrics were applied to summarize the performance of the model at different threshold values by combining confusion matrices at all threshold values producing a score of .731. Although the best value for AUC is 1, our results indicated that our classifier was at least good. As for KNN, accuracy was somewhat low for the model at 50%. After tuning, we received a cross validation score of .625 which suggests that our model would have an accuracy similar to that of the value produced. Indeed we received an accuracy score of .611 which was a 10% increase from the untuned model. It should be noted that if the test size were to increase to .3, the tuned accuracy of the model would be in fact lower then the accuracy of a test set with .2.

4. Conclusion

a. We applied classification to answer how several variables are related -- specifically how the 11 categorized features of our data set impacts quality of wine. By comparing the two models, we were able to find that they are more or less equally accurate in predicting new data although we would consider Logistic Regression as the better model due to the higher AOC score it produces. In the end, the accuracy that is produced by our two models is an indicator that it is indeed possible to determine the quality of wine without tasting it by testing the 11 features of our dataset and analyzing it through machine learning.