**Project Name:** Classifying Wine Using Logistic Regression and SVM Models

**Course:** PROG39051 Machine Learning Techniques

**Team Members:** Abira Esther Demello, Xiaohong Deng

## Introduction

## Literature Review

## Implementation

### Data Loading and Understanding

The first step in the implementation is to load the dataset and gain a comprehensive understanding of its structure and contents.

A black background with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Data Preprocessing

Data preprocessing is a vital step that ensures the accuracy, completeness, and relevance of data for analysis or machine learning models. These tasks involve several key processes:

#### Data Cleaning:

Data cleaning involves addressing issues like missing data, duplicates, and outliers, which can impact analysis or model training.

##### Handling missing data

A screen shot of a computer

Description automatically generated

The dataset does not have any missing data.

##### Handling duplicated data

A screen shot of a computer

Description automatically generated

The dataset does not have any duplicated data.

##### Handling outliers

A screen shot of a computer code

Description automatically generated

A group of yellow rectangular objects

Description automatically generated with medium confidence

The boxplot displays some unusual data points, marked by the small circles, in 'Ash', 'Alcalinity\_of\_ash', 'Magnesium', 'Proanthocyanins', and 'Color\_intensity'. These points are positioned more than 1.5 times the interquartile range (IQR) away from the quartiles, indicating they might be outliers.

A screenshot of a computer program

Description automatically generated

#### Handling class imbalances

Class imbalances occur when the distribution of values for the class is not uniform, potentially leading to biased model performance. Techniques such as oversampling, undersampling, or adjusting class weights are commonly used to address this imbalance. For this project, the 'oversampling' technique will be employed due to the small dataset size (only 161 rows).

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

#### Feature Selection

Feature selection is essential for reducing model complexity, overfitting, and training time.

When deciding on a feature selection method, we need to consider various factors, including dataset characteristics, computational cost, and modeling objectives.

**Correlation Matrix with Heatmap:** Inappropriate for our categorical target variable.

**RFE (Recursive Feature Elimination):** Not ideal for our two selected machine learning algorithms.

**SelectKBest:** Efficient and straightforward, suitable for quick feature selection.

**ExtraTreesClassifier:** Offers insights into feature importance and handles non-linear relationships well, aligning with our goal of creating logistic regression and SVC classifiers for wine type classification.

Considering our categorical target variable and the requirements of our logistic regression and SVC models, we've chosen ExtraTreesClassifier for feature selection.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

#### Data Transformation - Normalization:

Before building the model, it's essential to normalize the data since the unit of measurement might differ across features. This process ensures that all features contribute equally to the model. Here are the two steps involved:

- Step1: Visualize the distribution of each numerical column using a histogram to assess its normality. Look for a bell curve shape, which indicates a Gaussian distribution.

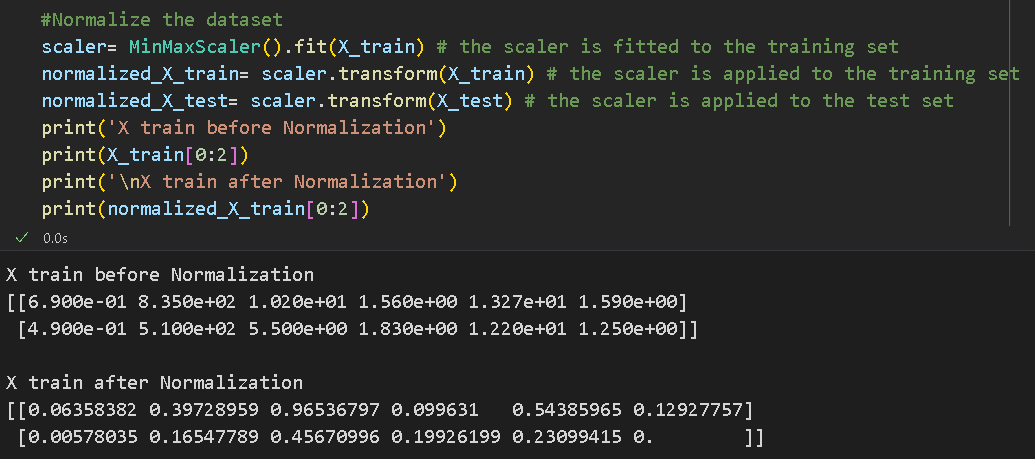
A screen shot of a computer code

Description automatically generated

A graph of different columns

Description automatically generated

- Step2 Normalize the features based on the observed distribution. Since all columes exhibit non-Gaussian distributions, Min-Max Normalization will be applied to these columns.



### Model Building

#### Logistic Regression Model Building

For building the Logistic Regression model, I used the LogisticRegression from sklearn library, initializing with the following hyperparameters:

**max\_iter:** This parameter specifies the maximum number of iterations for the solver to converge. In this project, I chose the default value of 200 for simplicity.

**penalty:** 'l2' is a common choice for Logistic Regression and is the default in scikit-learn. It helps with regularization.

**C:** The default value of 1.0 is chosen for C, controlling the regularization strength. Since the dataset is not extremely large or small, the default is suitable.

**solver:** 'lbfgs' is suitable for small datasets like this project.

**random\_state:** I set the random\_state to a fixed seed (42) for reproducibility and to ensure consistent results across runs.

A screen shot of a computer

Description automatically generated

#### SVM Model Building

For building the SVM model, I used the SVC Classifier from sklearn library, initializing with the following hyperparameters:

**kernel:** 'rbf' is a good default choice for non-linear classification problems. It can handle complex relationships between features.

**C:** The default value of 1.0 is chosen for C, controlling the regularization strength. Since the dataset is not extremely large or small, the default is suitable.

**random\_state:** I set the random\_state to a fixed seed (42) for reproducibility and to ensure consistent results across runs.

A screenshot of a computer program

Description automatically generated

## Evaluation

#### Logistic Regression Model Building

##### Evaluate the model using a confusion matrix.

A confusion matrix is a matrix that breaks down correctly and incorrectly classified into: True positive (TP): Correctly predicting the positive class. True Negative (TN): Correctly predicting the negative class. False Positive (FP): Incorrectly predicting the positive class.

A diagram of positive and negative

Description automatically generated A screenshot of a computer

Description automatically generated

The confusion matrix shows the following:

* Rows correspond to the actual classes.
* Columns correspond to the predicted classes.
* The diagonal elements (top-left to bottom-right) represent the number of correct predictions for each class.
* Off-diagonal elements represent incorrect predictions.

A screen shot of a computer code

Description automatically generated

**Confusion Matrix:** both the train and test confusion matrices show high accuracy in classifying samples as there is only two misclassification on the training data and test data.

##### Evaluate the model using an accuracy score

The accuracy score calculates the accuracy by comparing the predicted labels to the true labels and returns a value between 0 and 1, where 1 indicates perfect accuracy. The model has achieved high accuracy on both the training and test data, which suggests that it is performing well and generalizing effectively to unseen data.

A screen shot of a computer

Description automatically generated

**Accuracy:** The accuracy scores of 98% on the training set and 96% on the test set indicate that the Logistic Regression model performs exceptionally well in both training and unseen data.

##### Evaluate the model using a classification\_report

The classification\_report function provides a detailed report with various metrics such as precision, recall, and F1-score for each class in the classification. These metrics are commonly used to evaluate the model's performance.

A screen shot of a computer program

Description automatically generated

**Classification Report:** The precision, recall, and F1-score metrics provide a detailed view of the model's performance for each class. The weighted average F1-score of 0.96 for the test set demonstrates a strong overall performance.

#### SVM Model Building

##### Evaluate the model using a confusion matrix.

A screen shot of a computer

Description automatically generated

**Confusion Matrix:** both the train and test confusion matrices show high accuracy in classifying samples as there is only one misclassification on the training data.

##### Evaluate the model using a accuracy score

A screenshot of a computer program

Description automatically generated

**Accuracy:** The accuracy scores of 0.99 on the training set and 1.0 on the test set indicate that the SVM model performs exceptionally well in both training and unseen data.

##### Evaluate the model using a classification\_report

A screen shot of a computer

Description automatically generated

**Classification Report:** The precision, recall, and F1-score metrics provide a detailed view of the model's performance for each class. The weighted average F1-score of 1.0 for the test set demonstrates a strong overall performance.

#### Compare Logistic Regression Model and SVM Model

Based on the evaluation metrics for both the Logistic Regression and SVM models:

* Confusion Matrix:
* Logistic Regression: The confusion matrix for both the training and test sets shows high accuracy, with only two misclassification in the training data and the test data.
* SVM: The confusion matrix for both the training and test sets also shows high accuracy, with only one misclassification in the training data.
* Accuracy Score:
* Logistic Regression: Achieved an accuracy score of 0.99 on the training set and 0.96 on the test set. This indicates that the Logistic Regression model performs exceptionally well on both the training and unseen data.
* SVM: Achieved an accuracy score of 0.99 on the training set and 1.0 on the test set. Similar to the Logistic Regression model, the SVM model performs exceptionally well on both the training and unseen data.
* Classification Report:
* Logistic Regression: The precision, recall, and F1-score metrics from the classification report show a strong overall performance. The weighted average F1-score of 0.96 for the test set indicates high precision and recall.
* SVM: The precision, recall, and F1-score metrics from the classification report also show a strong overall performance. The weighted average F1-score of 1.0 for the test set indicates high precision and recall.

In summary, both the Logistic Regression and SVM models demonstrate excellent performance in classifying wine types based on their chemical attributes. They have high accuracy scores, minimal misclassifications in the confusion matrices, and strong precision, recall, and F1-scores according to the classification reports. However, the SVM model shows slightly better performance compared to the Logistic Regression model.

## Conclusion