Red-Wine Quality Analysis

Sreeja Dacharla   
S20200020256  
UG3,ECE,IIITSmail:sreeja.d20@iiits.inDharani Pokuri  
S20200020294  
UG3,ECE,IIITSmail:dharani.p20@iiits.in

***Abstract*—This article discusses linear exploratory data analysis of a given dataset. A brief introduction to each of the topics is given. The dataset is explained thoroughly.The article consults about the models that would be able to fit the dataset we used.**

***Keywords—Logistic Regression,Under sampling, Over sampling,Hybrid sampling, Naive Bayes Classifier, K-Nearest Neighbors classifier, Support Vector Machine, Principal Component Analysis.***

# **1.Introduction**

Red wine is one of the most popular alcoholic beverages in the world, enjoyed by millions of people for its unique taste and health benefits. However, the quality of red wine can vary greatly depending on various factors, such as grape variety, fermentation process, and storage conditions. In order to ensure the best quality of red wine, it is important to have an accurate and reliable method of assessing its quality.

In this project, we will analyze a dataset containing various chemical properties of red wines and their corresponding quality ratings, ranging from 0 to 10. Our goal is to explore the dataset and build classification models that can predict the quality of red wine based on its chemical properties. We will also investigate different sampling techniques to handle imbalanced data and compare the performance of different classification algorithms.

# **2.Dataset Description**

The Red Wine Quality dataset comprises 1599 samples of red wine, each characterized by 12 physicochemical attributes. The dataset contains no missing values or null entries, and all the features are of the same datatype (numerical). The dataset shape is (1599, 12). The 12 attributes are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates, alcohol, and quality.

Fixed acidity measures the amount of non-volatile acids in the wine and is measured in g/dm^3. Volatile acidity measures the amount of acetic acid in the wine and is also measured in g/dm^3. Citric acid measures the amount of citric acid in the wine, and residual sugar measures the amount of sugar left after fermentation, both measured in g/dm^3. Chlorides measure the amount of salt in the wine, and free sulfur dioxide and total sulfur dioxide measure the amount of sulfur dioxide in the wine, both measured in mg/dm^3. Density is the density of the wine, measured in g/cm^3, and pH is the pH value of the wine. Sulfates measure the amount of sulfates in the wine, and alcohol measures the percentage of alcohol in the wine. Finally, quality is the quality rating of the wine, ranging from 0 to 10.

# **3.Data Preprocessing**

## **3.1 Handling Outliers**

Outliers can significantly affect the accuracy of the model by introducing noise and bias. Therefore, we used the Interquartile Range (IQR) method to identify and remove any outliers present in the dataset. We calculated the IQR for each column and used it to define a range of acceptable values. After removing the outliers, we were left with a cleaned dataset(Fig.1) with a reduced number of rows. This preprocessing step helps improve the accuracy of the model by eliminating any noise and bias introduced by outliers in the dataset.

## **3.2 Data normalization**

To ensure that all the features in the dataset have the same scale, we performed data normalization using the min-max scaling technique. This technique scales the data between 0 and 1 by subtracting the minimum value from each value and dividing the result by the range (maximum - minimum) of the feature. By doing this, we ensured that each feature has equal importance during modeling and that the model is not biased towards features with larger values. This preprocessing step helps ensure that the model is not biased towards features with larger values and that each feature has an equal impact on the final results.

# **4.Data Visualization**

## **4.1 Bar plot**

We analyzed the quality of red wine samples in our dataset. To do this, we counted the number of samples for each quality level using the value\_counts() method in pandas. We then created a bar chart to display the distribution of red wine quality levels. The bar chart shows that the majority of samples fall in the middle range of wine quality levels (i.e., 5, 6, and 7), while the number of samples at the lowest and highest quality levels (i.e., 3, 4, 8, and 9) are relatively small. This distribution can provide insights into the characteristics of the red wine samples in our dataset and can be used to guide further analysis and modeling.

## **4.2 Pie Chart**

The pie chart shows the distribution of red wine quality levels in the dataset. The largest proportion of wines in the dataset have a quality rating of 5 or 6, with smaller proportions having ratings of 3, 4, 7, or 8. The chart is useful for quickly visualizing the overall distribution of quality ratings in the dataset. The percentages displayed in the chart provide an easy-to-understand summary of the distribution of quality ratings.

## **4.3 Donut Plot**

The donut plot(Fig.3) displays the distribution of red wine quality ratings in the dataset. The plot shows the percentage of each quality rating as a proportion of the total number of samples. The chart clearly indicates that most of the samples fall into the 5 and 6 quality ratings, accounting for more than 80% of the total samples. The inner white circle adds a unique touch to the plot and makes it more visually appealing. This chart can be useful in identifying the most common quality ratings and their proportion in the dataset.

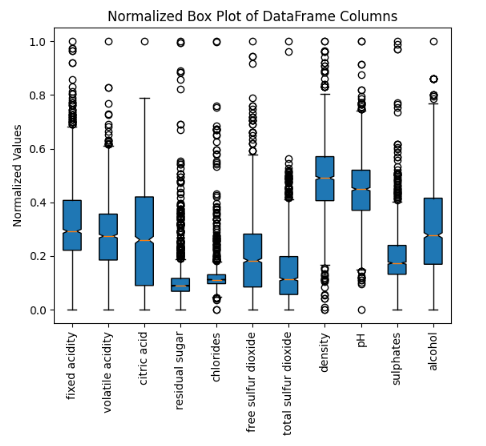


Fig.1.a. Normalized data with outliers

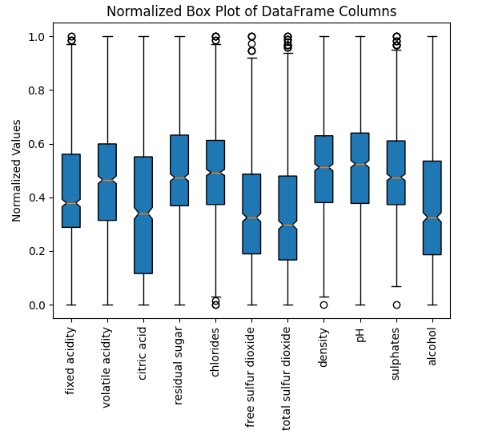


Fig.1.b. Normalized data after outlier removal

## **4.4 Stacked Bar Plot**

The stacked bar chart(Fig.4) shows the distribution of red wine quality ratings in the dataset. The red bars represent the counts for each quality rating, while the blue bars represent the cumulative counts up to that rating. For example, there are more than 1000 wines with a quality rating of 5, and over 2000 wines with a quality rating of 6 or higher. This chart provides a quick overview of the distribution of quality ratings in the dataset, which can be helpful in understanding the quality characteristics of the wines.

## **4.5 Correlation Analysis(Heat Map)**

We conducted a correlation analysis on the dataset and the results are displayed in the following correlation matrix. The values in the matrix represent the correlation coefficients between each pair of variables. The closer the value is to 1 or -1, the stronger the correlation between the two variables. A positive correlation coefficient indicates a positive linear relationship between the variables, while a negative correlation coefficient indicates a negative linear relationship. A correlation coefficient of 0 indicates no linear relationship between the variables.

By examining the correlation matrix, we can see which variables are positively or negatively correlated with each other. This information can help us better understand the relationships between the variables in the dataset and identify potential predictors of wine quality. We also created a heatmap to visualize the correlation matrix using the Seaborn library. The heatmap is a graphical representation of the correlation matrix, where each cell is colored based on the corresponding correlation coefficient. The darker the color, the stronger the correlation between the two variables.

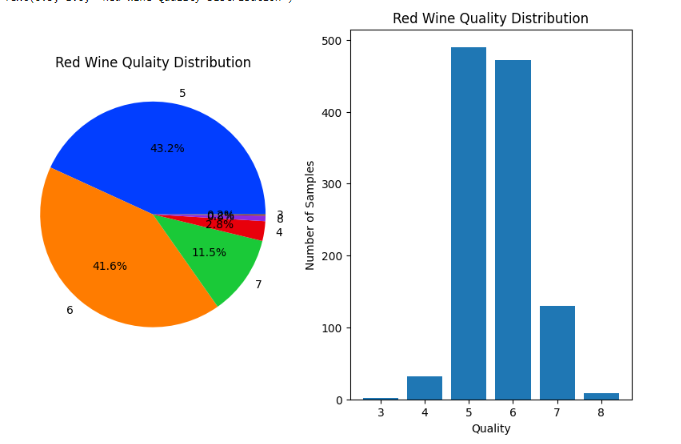


Fig.2. Red Wine Quality Distribution(pie,bar)

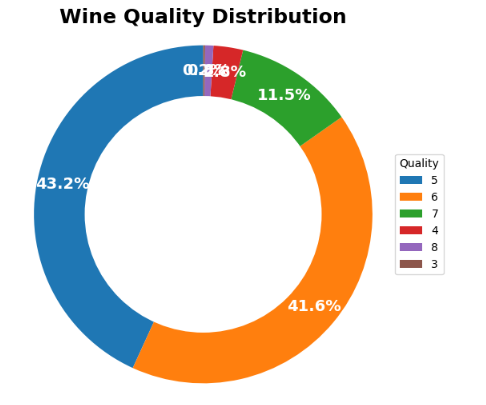


Fig.3. Red Wine Quality Distribution(donut)

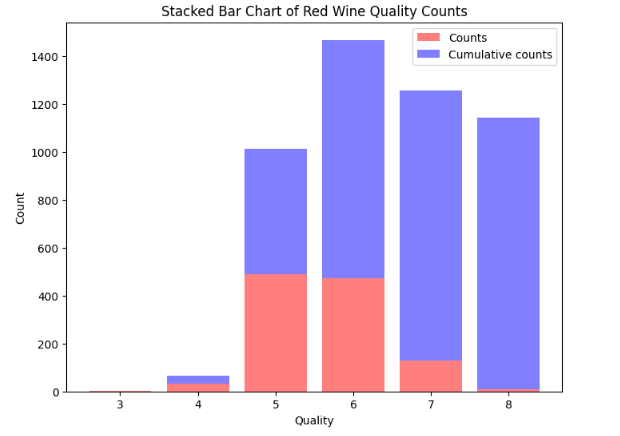


Fig.4. Bar plot for counts and cumulative counts

## **4.6 Pair plot**

We also created a pair plot to visualize the relationships between the different features in our dataset. The pair plot is a matrix of scatter plots, with each feature plotted against every other feature. The diagonal of the matrix shows the distribution of each feature

From the pair plot, we can see that some of the features, such as alcohol and volatile acidity, have a relatively strong linear relationship with quality. We can also see that some of the features have a more complex relationship with quality, such as citric acid and chlorides. Overall, the pair plot is a useful tool for identifying potential relationships between features and gaining a better understanding of the structure of our dataset.

## **4.7 Box Plots after grouping data**

In order to further understand the relationship between chemical properties and wine quality, we grouped the wine samples by their quality ratings and created boxplots for each quality level. The box plots show the distribution of each chemical property for each quality rating.

We observed that the distributions of the chemical properties varied among the different wine quality ratings. For example, the median value of alcohol increased with increasing wine quality, while the median values of volatile acidity and citric acid decreased. We also noticed that some chemical properties had wider ranges of values for certain quality ratings, indicating greater variability among the samples. These findings suggest that certain chemical properties are strong predictors of wine quality, and could be used to guide the production of high-quality wines.

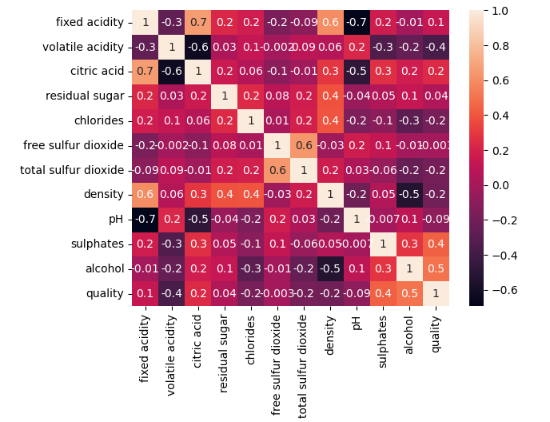


Fig.5. Correlation Heatmap

## **4.8 Bar Plots after grouping data**

By grouping the wine samples according to their quality level, we can observe the differences in the mean values of each chemical property between different quality levels. We can see that the mean value of alcohol increases with increasing wine quality, indicating that alcohol content is an important factor in determining the quality of wine. Similarly, we can see that the mean values of volatile acidity and citric acid generally decrease with increasing wine quality, indicating that these chemical properties are also important predictors of wine quality.z

We can also observe that some chemical properties have wider ranges of mean values for certain quality ratings, indicating greater variability among the samples. For example, the mean values of residual sugar and free sulfur dioxide have wider ranges for the lower quality ratings, indicating that there is more variability in these chemical properties among lower-quality wines. Overall, these bar plots provide a valuable summary of the relationship between chemical properties and wine quality and can help guide the production of high-quality wines by highlighting the most important chemical factors to focus on.

# **5.Methodology**

The project will be divided into several steps, starting with data preprocessing to handle missing values and outliers, followed by data visualization to gain insights into the relationships between different variables. We will then create a binary class dataset based on the quality ratings and perform logistic regression using various sampling techniques to handle imbalanced data. Next, we will explore different classification models, including Naive Bayesian, KNN, Decision Tree, SVM, and PCA,SVM, and evaluate their performance using various classification metrics. Finally, we will summarize our findings and provide recommendations for further research.

# **6.Binary Class Analysis**

In order to simplify the analysis, we converted the multi-class classification problem of predicting wine quality ratings into a binary classification problem of predicting whether a wine is of "mid" or "extreme" quality. Wines rated as 5 or 6 were considered as "mid" quality and wines rated 3, 4, 7, 8, or 9 were considered as "extreme" quality.

By converting the dataset into a binary classification problem, we were able to simplify the analysis and focus on identifying the chemical properties that are most important in differentiating between "mid" and "extreme" quality wines. This allowed us to potentially identify key chemical properties that could be targeted in the production of high-quality wines.

## **6.1 Bar plots after grouping with mid/extremes**

After splitting the preprocessed dataset into two binary classes based on the "mid\_or\_extreme" group, box plots (Fig.6)were generated to analyze the distribution of chemical properties for each class. We observed that the mean values of most chemical properties were higher for mid-quality samples than for extreme quality samples.

And also that the distributions of chemical properties were wider and more variable for extreme quality samples than for mid quality samples. This suggests that the chemical properties of mid quality samples are more consistent, while those of extreme quality samples are more diverse.

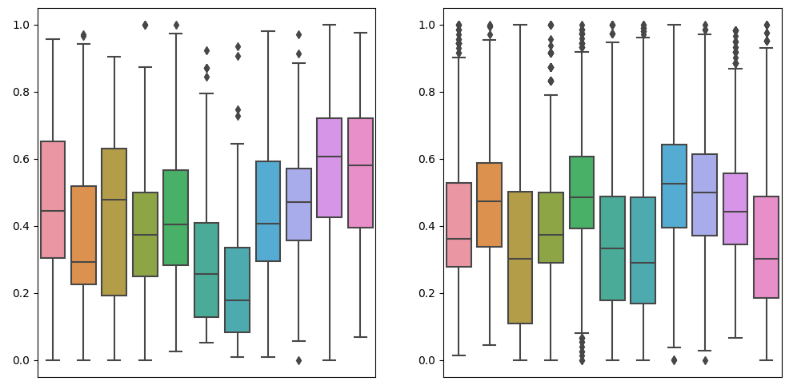


Fig.6. Bar plot after grouping data

Then created a pie chart and a bar plot to visualize the distribution of these groups.The chart shows that the majority of the wines in our dataset (82.5%) are considered mid-quality, while the remaining 17.5% are considered extreme-quality. In addition, we also created a bar plot to visualize the count of each group. The plot shows the same information as the pie chart, but in a different format. The x-axis represents the two groups (0 for extreme quality and 1 for mid-quality), and the y-axis represents the count of each group. The plot shows that there are more than 1200 wines in the mid-quality group and approx of 300 wines in the extreme-quality group.

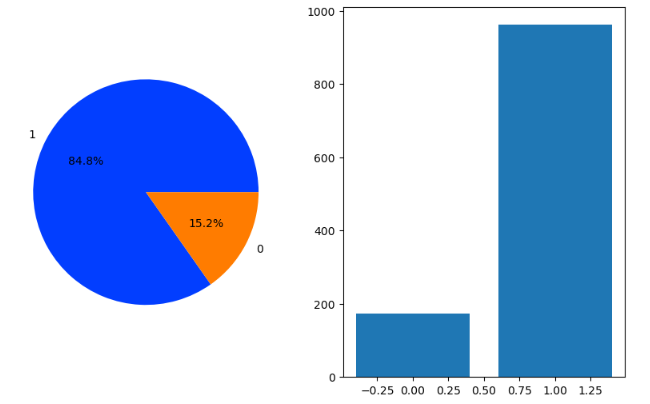


Fig.7. Binary class distribution(pie,bar)

## **6.2 Logistic regression**

After, completing Exploratory data analysis in the Logistic Regression part we first split the data into training and testing sets using “train\_test\_split” function from sklearn.model\_selection module.This helps to ensure that the model is trained on a subset of the data and tested on a completely independent subset, which helps to prevent overfitting and provides a more accurate estimate of the model’s performance on new, unseen data. In this case, we have used test\_size of 0.25 and a random state of 5 to ensure reproducibility.

Then we trained a logistic regression model using the training set and made predictions on the testing set using the predict method. We have evaluated the performance of the model using a confusion matrix, which shows the number of true positive, true negative, false positive, and false negative predictions. We have plotted the confusion matrix as a heatmap(Fig.8) along with other performance metrics including accuracy, f1 score and ROC (Receiver operating characteristics)(Fig.8) plot.The accuracy obtained is 84.85% and f1 score is 0.9155.

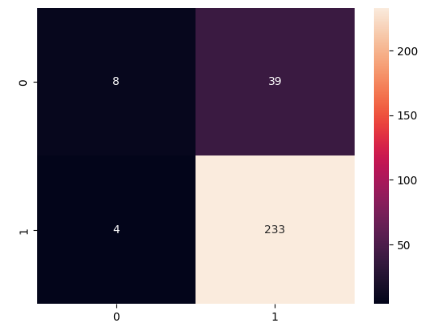


Fig.8.a Confusion matrix for logistic regression

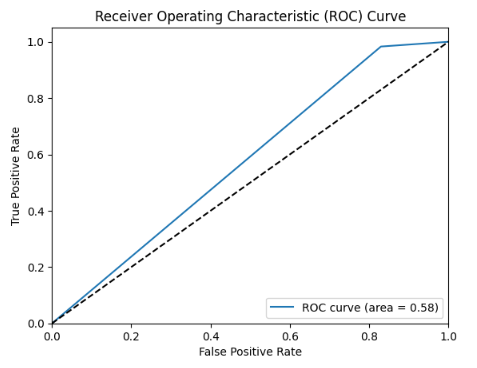


Fig.8.b. ROC curve for logistic regression

## **6.3 Undersampling the data**

To address the class imbalance issue in our dataset, we decided to use undersampling, which involves reducing the size of the majority class (mid-quality wines) to match that of the minority class (extreme-quality wines). Specifically, we randomly selected a subset of mid-quality wines that had the same size as the extreme-quality group and combined them with the latter to form a new dataset with balanced classes. We then shuffled the rows of the resulting dataset to avoid any bias due to the original ordering of the samples.

After this preprocessing step, we trained a logistic regression model on the balanced dataset, using the same features and hyperparameters as before. The purpose of the model was to predict the quality category (extreme or mid) based on the wine's physicochemical properties. We split the data into training and testing sets, with a 75/25 ratio, and standardized the features using the StandardScaler() method. We then fit the logistic regression model on the training set and made predictions on the testing set.

The confusion matrix of the logistic regression model showed(Fig.9) that it had an accuracy of about 80%, which is a decrease over the previous model trained on the imbalanced dataset. The f1 score obtained is 0.80.

## **6.4 Oversampling the data**

To further analyze the performance of the logistic regression model and explore different techniques to handle class imbalance, we decided to try oversampling, which involves increasing the size of the minority class (extreme-quality wines) by duplicating some of its samples until its size matches that of the majority class (mid-quality wines). Specifically, we used the resample() method to randomly select and duplicate some of the extreme-quality wines until their count matched that of the mid-quality wines. We then combined the resulting oversampled class with the original mid-quality class to form a new dataset with balanced classes.

After creating the new dataset, we followed the same steps as before to train and evaluate a logistic regression model on it. We split the data into training and testing sets, with a 75/25 ratio, and standardized the features using the StandardScaler() method. We then fit the logistic regression model on the training set and made predictions on the testing set. The confusion matrix of the logistic regression model(Fig.10) showed that it had an accuracy of about 74%, which is lesser than the accuracy of the model trained on the undersampled dataset. The f1 score obtained is also 0.74 which is less compared to under sampling.

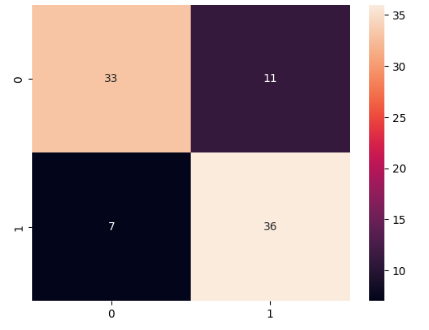


Fig.9.a Confusion matrix for undersampled data

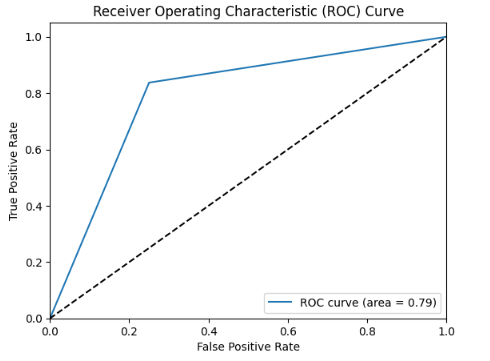


Fig.9.b ROC curve for undersampled data

## **6.5 Hybrid Sampling the data**

After performing both undersampling and oversampling techniques, we decided to explore hybrid sampling techniques to further improve the performance of our model. We used the SMOTEENN technique, which combines both oversampling and undersampling techniques. The SMOTEENN technique oversamples the minority class and then applies the edited nearest neighbor rule to clean the resulting dataset. We also applied RandomUnderSampler to balance the class distribution further.

However, after training the logistic regression model on the hybrid resampled dataset, we found that the accuracy was only 70%, which is less than the accuracy obtained by both undersampling and oversampling. The f1 score obtained for the hybrid sampling was 0.79, which is slightly better than the oversampled dataset, but still lower than the undersampled dataset(Fig.11).

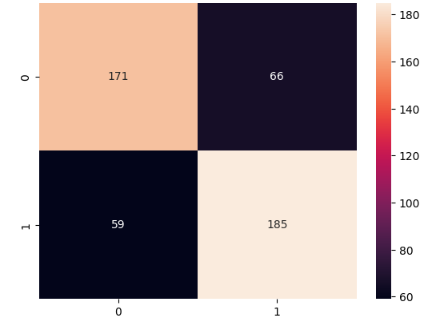


Fig.10.a Confusion matrix for oversampled data

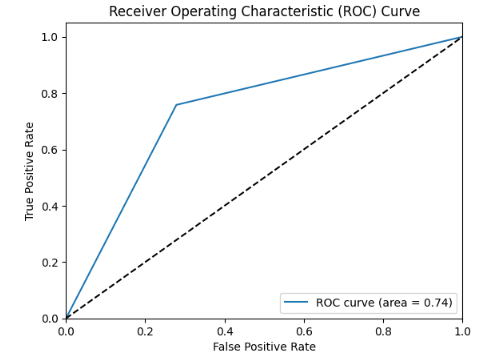


Fig.10.b ROC curve for oversampled data

# **7.Multiclass analysis**

## **7.1 Multinomial Naive Bayesian**

In this section, we used the Multinomial Naive Bayes Classifier to predict the quality of wines based on their chemical properties. First, we created a new column in our dataset called "mid\_or\_extreme" to classify wines as either midquality or extreme-quality based on their quality score. We then split our dataset into training and testing sets and used the Multinomial Naive Bayes Classifier to fit and predict the wine quality.

The classification report shows that the accuracy of the model is 56%. We also plotted a confusion matrix(Fig.12) to visualize the performance of the classifier. The confusion matrix shows that the model performed well in classifying mid quality wines, but struggled with extreme-quality wines. The precision and recall for extreme-quality wines were relatively low compared to mid-quality wines. Overall, the Multinomial Naive Bayes Classifier did not perform well in predicting the quality of wines based on their chemical properties

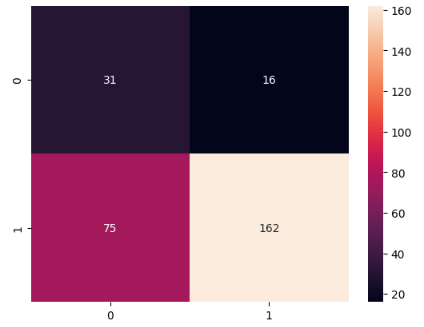


Fig.11.a Confusion matrix for hybrid sampled data

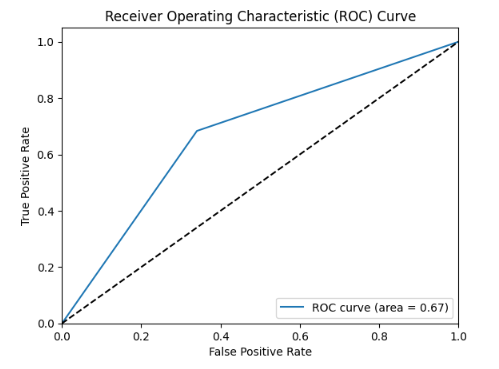


Fig.9.b ROC curve for undersampled data

## **7.2 K-Nearest Neighbours Classifier**

We applied the K-Nearest Neighbors (KNN) model to the preprocessed data using the same train-test split as before. The KNN algorithm is a type of instance-based learning or lazy learning, where the model predicts the class of a new data point based on the classes of its k nearest neighbors in the training data. For our implementation, we used the KNeighborsClassifier function from the scikit-learn library. After fitting the model to the training data, we predicted the classes of the test data and obtained an accuracy of 71%. We visualized the results using a confusion matrix(Fig.13) heatmap and obtained the classification report.

## 

Fig.12 Confusion matrix for Naive Bayesian Classifier

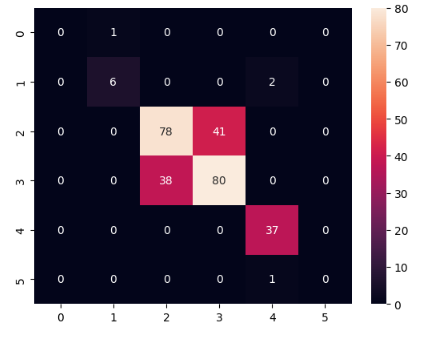


Fig.13 Confusion matrix for K-Nearest Neighbors Classifier

## **7.3 Decision Tree Classifier**

We applied the Decision Tree Classifier (DTC) model to the preprocessed data using the same train-test split as before. The classifier was initialized with a criterion of ‘entropy’ and a random state of 5. The entropy criterion measures the amount of uncertainty and randomness in our data, and the random state ensures that our results are reproducible.For our implementation, we used the DecisionTreeClassifier function from the scikit-learn library. After fitting the model to the training data, we predicted the classes of the test data and obtained an accuracy of 70.42%. We visualized the results using a confusion matrix(Fig.14) heatmap and obtained the classification report.

## **7.4 Support vector machine**

We trained a Support Vector Machine (SVM) model to predict the quality of wine using the preprocessed dataset. SVM is a powerful and widely used classification algorithm that can be applied to a wide range of datasets. After splitting the dataset into training and testing sets using the same method as before, we used the SVC class from the svm module in scikit-learn to fit the model to the training data with a linear kernel. We then made predictions on the test set and evaluated the model's performance using several metrics.

The SVM model achieved an accuracy of 72% on the test set, which is slightly more than the accuracy obtained by the KNN model. Overall, while SVM is a powerful and flexible algorithm, it appears to be more effective than KNN in predicting the quality of wine in this particular dataset. Finally SVM is giving more accuracy among KNN and MNB(Fig.15)

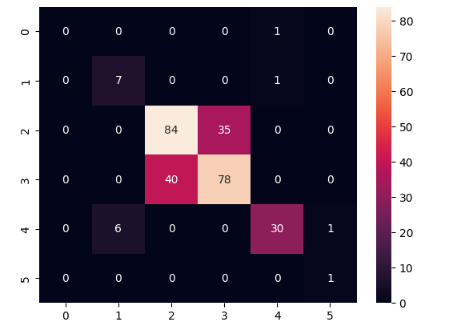


Fig.14 Confusion matrix for Decision Tree

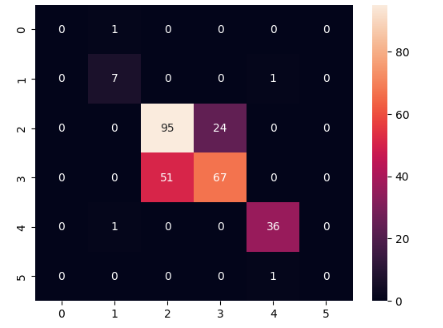


Fig.15 Confusion matrix for SVM

## **7.5 Dimensionality Reduction with PCA , SVM Classification**

Dimensionality reduction is a common technique used in machine learning to reduce the number of features in a dataset while retaining as much of the original information as possible. One popular method for dimensionality reduction is Principal Component Analysis (PCA), which transforms high-dimensional data into a lower-dimensional space while preserving the maximum amount of variance.

In this analysis, we apply PCA to the wine quality dataset and then use Support Vector Machine (SVM) classification to predict the quality of the wine. First, we split the data into training and testing sets using a 75:25 ratio. Next, we apply PCA with 3 components to the training data using the PCA function from the sklearn.decomposition module, and then transform the testing data using the transform method of the PCA object.

We then fit a SVM model with a linear kernel to the transformed training data using the SVC function from the sklearn.svm module. Finally, we make predictions on the transformed testing data using the predict method of the SVM model.

The accuracy of the SVM classifier on the transformed testing data was measured to be 71%. Which is almost equal to the accuracy obtained with SVM model. But, even if the data is reduced the accuracy is decreased by only 1%. ,Thus The results obtained suggest that this approach may be a useful technique for reducing the dimensionality of high-dimensional data and improving the accuracy of classification models(Fig.16).

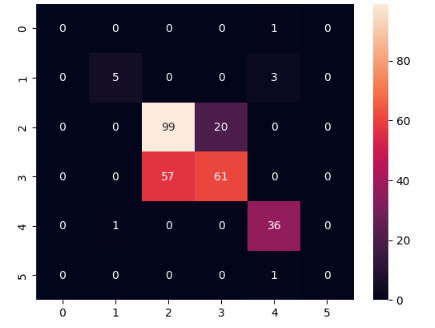


Fig.16 Confusion matrix for SVM with PCA

# **8.Conclusion**

This project has involved several key steps to analyze the wine dataset. First, we have done data preprocessing and removed outliers.Next, we have done data visualization to gain insights into the relationships between different variables.

To perform binary classification based on the wine quality ratings, we have used logistic regression with various sampling techniques to handle imbalanced data. Subsequently, we made an analysis of different classification models, including Naive Bayesian, KNN, Decision Tree, SVM, and PCA-SVM, and evaluated their performance using various classification metrics.

In our analysis, we found that SVM had the highest accuracy among all the classification models tested, achieving an accuracy of 72%. The SVM model was particularly effective at correctly classifying both good and bad quality wines, with a high recall value.