# White Wine Quality: A Binary Classification Task

Muhammad Faiq Hilman bin Mohd Fauzi
*City University of London*
Muhammad.Mohd-Fauzi.2@city.ac.uk

***Abstract*—**This paper aimed to perform a binary classification task using deep learning neural networks like the Support-Vector Machines and Multi-layered Perceptron models on the white wine quality dataset from the UCI Repository. The purpose of the models is to help winemakers create higher-quality wine to be consumed by their customers and help improve the quality of products in the wine industry by using AI classification models for wine. It was found that the SVM model yielded better results than the MLP.

## I. INTRODUCTION

There are many deep learning neural network models that exist for the purpose of performing various tasks, such as regression, classification, time series and others. However, some perform better than others for a specific task. This paper will be using 2 models for the purpose of a classification task, the models being the Multi-layered Perceptron (MLP) and Support Vector Machines (SVM) on the white wine quality dataset which was acquired from the UCI Repository website. The dataset is linked to red and white variants of the Portuguese "Vinho Verde" wine which was used in Cortez et al (2009), where MLP and SVM were used, albeit on a regression task. In that paper, SVM performed the best. This paper is aimed to discover which model will yield the most optimal performance on a classification task on the white wine dataset by experimenting with different configurations and hyperparameters in each model based on the input of feature variables such as alcohol, citric acid, chlorides, PH values, and others. The importance of building a good model is to help winemakers produce higher-quality wine for their clients and consumers based on the feature variables mentioned previously, thus improving their quality of products, competitiveness, and products in their markets.

### *A.* *Multilayered Perceptron*

Multi-layered perceptrons (MLP) are one of several neural networks that have been widely used for classification tasks in various fields. It consists of multiple layers of interconnected nodes that process input data and produce an output. MLPs have been used for classification tasks such as acute lymphoblastic leukemia detection using microscopic images of blood, polarized signal classification and flow regime classification. MLPs are also usable for regression tasks, and the output of the nodes will be continuous values instead of discrete ones. It is also used for image classification tasks such as content-based image retrieval, land cover product generation, and comparison of conventional and deep learning methods of image classification. MLPs have been shown to be effective for classification tasks when properly designed and trained (He and Chen, 2021).

### *B.* *Support Vector Machines*

Support Vector Machines (SVM) is a type of supervised machine learning algorithm that can be used for classification and regression purposes (Liu et al 2020). SVMs are especially useful for classification tasks, and they function by finding the hyperplane that is most optimal to separate the data points of different classes with the largest margin (Du and Swamy, 2014). SVMs have been used for various classification tasks such as sentiment analysis, social network user labeling, object-based land cover classification, credit risk assessment, and leukemia cancer type dataset classification. SVMs have also been used in combination with other machine learning algorithms such as K-Nearest Neighbors (KNN) to

improve classification accuracy. SVMs perform well in classification tasks when properly designed and trained, as well as when applied to regression tasks.

## II. DATASET

### A. *Data Characteristics*

The white wine dataset has 4898 instances, containing 11 feature variables and 1 target variable which consists of 7 classes. The 7 classes are from 3-9, 3 being very poor and 9 being excellent wine quality. Compared to the red wine dataset, which only had around 1000 instances, the white wine dataset was chosen as there are more instances to work with, this is to avoid any overfitting issues during the training of the models. However, this is not always the case as it depends on the dataset and the task at hand, using more samples may or may not improve the performance of the model (Javaheri, 2021).

### B. *Exploratory Data Analysis*

Several techniques were performed in the exploratory data analysis stage of understanding the data. Figures 1 and 2 are boxplots and count plots which show how the target class of the wine quality is distributed, along with their outliers in the boxplot. It is observed that some outliers exist in classes 4,5 and 8, however, this is negligible, and the outliers were not removed from the dataset and proceeded to be included in the model. As observed in Figure 2, classes 5 and 6 combined for more than 75% of the target class, this is a problem and can lead to a model with misleading results. The measure done to solve this problem was to group the wine into 2 classes, 0 and 1, 0 being the wines with quality less than 5 and 1 if it is more than 5. Despite this transformation, the classes of 0 and 1 were still imbalanced, with 1 almost doubling 0. To prevent any issues in the model and to improve accuracy by reducing the class imbalance, several resampling techniques were considered such as SMOTE (Pradipta et al, 2021). In the end, sklearns resample tool was used to artificially up sample the target class 0 to make it balanced with class 1.

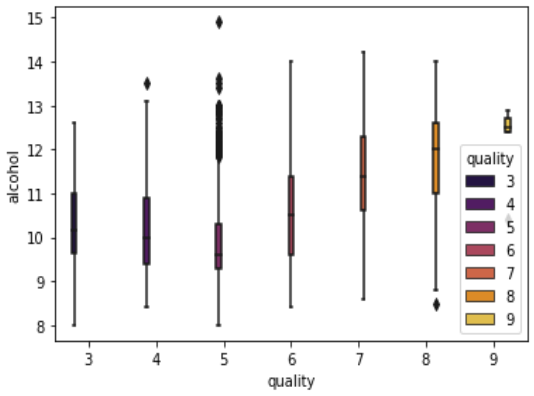


Figure 1: Box plot

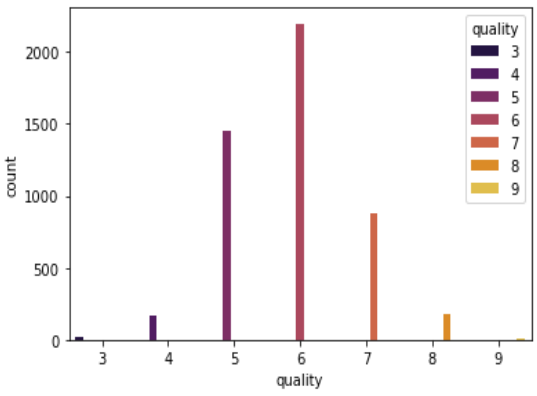


Figure 2: Count plot

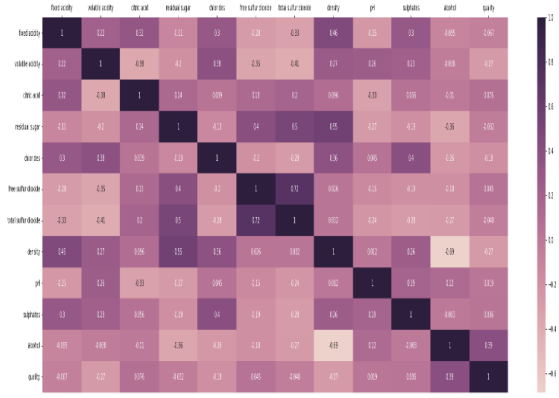

Figure 3: Correlation matrix heatmap

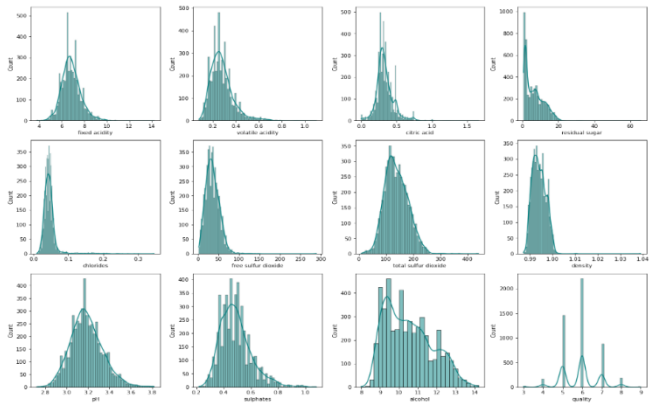

Figure 4: Distribution plots of all variables

To gain a deeper understanding of how the features interact with each other, a correlation matrix heatmap as well as distribution plots of all variables are displayed in Figures 3 and 4. It is observed that most of the features are normally distributed, which is convenient as no transformation is necessary to make the data normally distributed as it already is. In the heatmap, it is also observed that the problem of variables being highly correlated with one another is also absent. Checks were also made if any missing data or values exist in the dataset, and it was found that no missing data existed in the dataset. This is important because the presence of missing data in a dataset will negatively impact the performance of both MLP and SVM models for a classification task (Palanivinayagam and Damaševičius, 2023).

## III. METHODOLOGY

### A. *Hypothesis Statement*

It is hypothesized that the application of machine learning techniques such as MLP and SVM on the dataset will yield high-accuracy binary classification results, where the MLP model will achieve comparable performance to SVM despite its limitations in handling high-dimensional and complex datasets, and the SVM model will outperform MLP due to its robustness in handling such datasets. Additionally, it is expected that feature engineering techniques such as feature scaling and resampling techniques will enhance the performance of both models and that by analyzing the importance of the input features, valuable insights will be gained into the factors that contribute to the quality of white wine and contribute valuable guidance to winemakers in optimizing the wine production process.

### B. *Advantages and Disadvantages of MLP and SVM*

| Advantages | Disadvantages |
|---|---|
| Efficient in modeling complex nonlinear systems | Can be slow to converge because of the global nature of the backpropagation causing slow learning |
| Can be used for regression, classification, and time series forecasting | Function as hyperplane classifiers when solving classification problems and can be inefficient if the decision boundary is nonlinear |

Table 1: MLP advantages and disadvantages

| Advantages | Disadvantages |
|---|---|
| Possess unique advantage in solving small samples, nonlinear, and pattern recognition problems involving high dimensions | Selecting parameters heavily impacts classification accuracy and generalization ability |
| Less vulnerable to overfitting problems compared with other techniques | Can be slow to train if dealing with large datasets |

Table 2: SVM advantages and disadvantages

### C. *MLP Architecture*

There are several hyperparameters that will be experimented with, these include the learning rate, momentum, weight decay, number and size of hidden layers, epochs, activation function and optimization function. Cross-entropy loss is used as the loss function as it is most appropriate for a binary classification task and functions by measuring the difference between the predicted output and the actual output (Lan et al, 2020). The final model had 3 layers of varying sizes in each layer.

### D. *SVM Architecture*

The hyperparameters that were used in SVM included C, the kernel function, the degree parameter, which is used if using a polynomial function, gamma, which is used in rbf kernels that determines how smooth the decision boundary is, and coef0 which is the independent term that is only relevant in a polynomial or sigmoid kernel function. During the exploratory data analysis, the data distribution shows that it is not linear, therefore the experimentation of the kernel function will be between rbf, polynomial or the sigmoid. A linear function was not used.

### E. *Training and Evaluation Methodology*

The models were trained in an 80/20 train-test split with a portion of the train split being used as a validation split, the reason for this cross-validation technique is to validate the best-performing model learned from the training loop on a validation set, before applying it to unseen data which is in the test set. Prior to executing the model, GridSearchCV and RandomizedSearchCV were used on the SVM and MLP models to find the best configuration. As MLPs are computationally expensive and more complex, RandomizedSearchCV was used to find the optimal hyperparameters over GridSearchCV, this was to save time and remain computationally efficient in the experimentation process. As SVMs are less complex, GridSearchCV was used to evaluate which hyperparameters are the best. The models are then executed with the suggested configurations.

In the MLP model, the training set is evaluated with the cross-entropy loss function to measure the difference between predicted and actual output. The model is trained using the training set in batches, and after each batch, the training loss is calculated and stored in its list. The test set is evaluated after each epoch of training. The model is put in evaluation mode and the forward pass is done with the test set. The loss and accuracy on the test set are calculated using the cross-entropy loss function and the accuracy_score function, respectively. These values are then appended to their lists. Finally, the best model is evaluated on the test set after the training is complete. The model parameters that achieved the best validation loss are loaded and the model is put in evaluation mode. The forward pass is then done with the test set and the accuracy of the predictions is computed.

For SVM, the data is split into training and validation sets for cross-validation, and then use the training set to train the model, and the validation set is to assess its generalization to new unseen data. We then make predictions on the test set and calculate the test accuracy. Gamma is a hyperparameter that determines the influence of each training example and can affect the flexibility of the model in fitting the data. SVMs are less complex than MLPs and will have fewer hyperparameters to experiment with.

## IV. ANALYSIS AND RESULTS

| SVM | | | |
|---|---|---|---|
| C | Kernel Function | Degree | Test accuracy (%) |
| 0.1 | rbf | 3 | 66.72 |
| 1 | rbf | 3 | 81.37 |
| 10 | rbf | 3 | 85.81 |
| 0.1 | sigmoid | 3 | 48.93 |
| 1 | sigmoid | 3 | 51.07 |
| 10 | sigmoid | 3 | 48.93 |
| 0.1 | linear | 3 | 71.09 |
| 1 | linear | 3 | 71.78 |
| 10 | linear | 3 | 71.7 |

| MLP | | | | |
|---|---|---|---|---|
| Layers | Learning Rate | Weight Decay | Momentum | Test Accuracy |
| 32,63,25 | 0.01 | 0.01 | 0.1 | 69.33 |
| 16,32,63 | 0.1 | 0.1 | 0.5 | 51.69 |
| 8,16,32 | 0.5 | 1 | 0.9 | 49.85 |
| 8,16 | 0.01 | 0.01 | 0.1 | 66.72 |
| 16,32 | 0.1 | 0.1 | 0.5 | 64.49 |
| 32,64 | 0.5 | 1 | 0.9 | 50.92 |
| 8 | 0.01 | 0.01 | 0.1 | 63.11 |
| 16 | 0.1 | 0.1 | 0.5 | 49 |
| 32 | 0.5 | 1 | 0.9 | 49 |

Figure 5: Experimentation results of several configurations

Figure 5 shows the best-performing models in several configurations. The best model for SVM is using an RBF kernel function, with C=10 and degree=3. As SVMs seek to separate the classes in a hyperplane, the way the data is distributed is important in determining which function is appropriate. A polynomial function is computationally expensive and takes a long time to run as found in Mojtaba (2023), so it was excluded from the experimentation due to the interest of time The worst model is one that uses a sigmoid function and a C of 0.1, with an accuracy of 48.93%. It appears that SVM is particularly sensitive to which kernel function is used.

In the case of MLP, it is more sensitive to the learning rates, weight decay, and momentum. This is because it uses Stochastic Gradient Descent as the optimization function, and it seeks to find the local minima of the model by adjusting the weights in the model. Setting values that are too high or too low for the learning rate, weight decay and momentum is bad for the model because a high learning rate can cause the model to overshoot optimal weights and converge slowly or not at all in some cases and the opposite is true. A high weight decay causes models to overfit and the opposite is the case. A high momentum leads to slower convergence and poor generalization whereas a low momentum also causes the model to get stuck in the local minima (Zou et al, 2020). The best model had an accuracy of 69.33% while the worst models had an accuracy of 49%, which is expected because the momentum, learning rate, and weight decay values were very high. The number and size of hidden layers did not affect the model as much as the other hyperparameters.

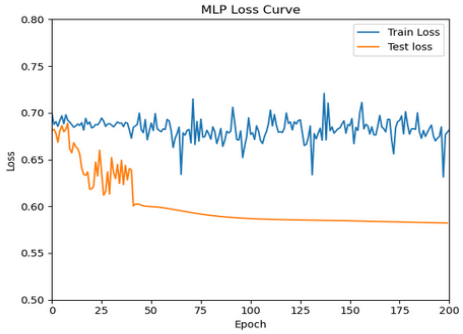


Figure 6: MLP Training and Test Loss Curves

Figure 6 shows how as the epochs progress, the training and test losses decline as the MLP model starts to learn the data and the loss begins to plateau after the 50th epoch. 50th epoch and onwards, the model did not reduce the losses any further and the model included early stopping, so it plateaued because further training did not improve the performance.

Confusion Matrix: MLP

| | |
|---|---|
| [545 | 119] |
| [285 | 355]] |

Classification Report: MLP

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.66 | 0.82 | 0.73 | 664 |
| 1 | 0.75 | 0.55 | 0.64 | 640 |
| accuracy | | | 0.69 | 1304 |
| macro avg | 0.70 | 0.69 | 0.68 | 1304 |
| weighted avg | 0.70 | 0.69 | 0.68 | 1304 |

Figure 7: MLP Results

Confusion Matrix: SVM

| | |
|---|---|
| [[590 | 74] |
| [ 97 | 543]] |

Classification Report: SVM

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.86 | 0.89 | 0.87 | 664 |
| 1 | 0.88 | 0.85 | 0.86 | 640 |
| accuracy | | | 0.87 | 1304 |
| macro avg | 0.87 | 0.87 | 0.87 | 1304 |
| weighted avg | 0.87 | 0.87 | 0.87 | 1304 |

Figure 8: SVM Results

Figures 7 and 8 show the classification report which has results obtained from the best-performing MLP and SVM models. In comparison to MLP, SVM performs significantly better in all important metrics such as precision, recall, f-1 score and accuracy. In the case of our goal, which is to correctly predict wine with good quality, it is important to pay attention to the recall metric. Recall is where it measures the model's ability to identify all positive instances, including those that are misclassified as negative. It is especially useful when the cost of false negatives is high, such as detecting wine of good quality. Winemakers will miss out on profits if they discard good quality wine that has been incorrectly classified as poor quality. SVM's recall of 0.85 significantly outperforms MLP recall of 0.55. MLP performs reasonably decent, with an accuracy of almost 70% and a recall of 0.82 for the class of 0. This implies that the model correctly identified 82% of all instances that belong to class 0 out of all instances that actually belong to class 0. In other words, the model correctly predicted 82% of all negative instances in the dataset. Its accuracy also means that MLP can predict with 69% probability which wines are of good quality and which are of bad quality. Accuracy may be misleading as a metric if the target classes were heavily imbalanced, but it was already accounted for in the data pre-processing when it was artificially upsampled to balance the classes. It is important to compare the models using all metrics, and SVM clearly outperforms in all metrics, making it the better model.

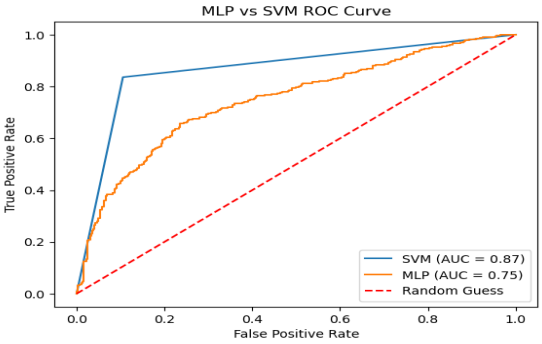


Figure 9: ROC comparison of MLP and SVM

The last key metric that was used to evaluate the models were ROC and AUC, known as Receiver Operating Characteristic and Area Under the Curve. ROC plots the true positive rate against the false positive rate, the purpose of which is to show how good the model is at telling the difference between good and bad wine. AUC measures the overall performance of the model across all possible classification thresholds. An AUC above 0.5 is decent. SVM's AUC of 0.87 over MLP's AUC of 0.75 further solidifies SVM as the superior model in this study. However, ROC and AUC do have disadvantages because it does not consider the cost of false positives and false negatives (Chicco and Jurman, 2023). This is why it is important to evaluate the models with as many metrics as possible to get a clearer understanding and more reliable estimate on which is the better performer.

## V. CONCLUSIONS, RECOMMENDATIONS & FUTURE WORK

After an iterative process of experimenting with the many different hyperparameters for both MLP and SVM models, the SVM model performed the best for the binary classification task of predicting the quality of the wine based on input from the data. Using ROC, confusion matrices and classification reports is also extremely helpful in evaluating the performances of these models when applied to the dataset.

There are 11 feature variables in the dataset, in the future any new work should consider some feature selection to exclude those that aren't relevant and include those that are relevant for predicting the quality of the wine. Besides SVM and MLP, there are also other models which can be used such as Random Forest, Convolutional Neural Networks and Naïve Bayes.

Extensive experimentation with different models can help identify which works best to achieve the goal of this study. Another area of improvement for the modeling would be to use more advanced cross-validation techniques such as k-fold cross-validation and use an average score of the cross-validation results to get a more reliable estimate of the model's performance.

## REFERENCES


[1] Camelo, PHC and de Carvalho, RL (2020). "Multilayer Perceptron Optimization Through Simulated Annealing and Fast Simulated Annealing." *Academic Journal on Computing, Engineering and Applied Mathematics*. 1(2), pp. 28–31. Available at: https://doi.org/10.20873/ajceam.v1i2.9474. (Accessed: 6th April 2022)

[2] Chicco, D., Jurman, G. (2023) 'The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification.' *BioData Mining* 16, pp.4. Available at: https://doi.org/10.1186/s13040-023-00322-4 (Accessed: 3rd April 2022)

[3] Cortez P., Cerdeira A., Almeida F., Matos T., Reis J. (2009) 'Modelling wine preferences by data mining from physicochemical properties', *Decision Support Systems*, 47(4), pp. 547-553.

[4] Du, KL., Swamy, M.N.S. (2014). 'Support Vector Machines. In: Neural Networks and Statistical Learning.' *Springer, London*. Available at: https://doi.org/10.1007/978-1-4471-5571-3_16 (Accessed: 12th April 2022)

[5] G. A. Pradipta, R. Wardoyo, A. Musdholifah and I. N. H. Sanjaya (2021), "Radius-SMOTE: A New Oversampling Technique of Minority Samples Based on Radius Distance for Learning From Imbalanced Data," *IEEE Access*, vol. 9, pp. 74763-74777, Available at: 10.1109/ACCESS.2021.3080316. (Accessed: 3rd April 2022)

[6] He X, Chen Y.(2021) 'Modifications of the Multi-Layer Perceptron for Hyperspectral Image Classification.' *Remote Sensing*. 13(17), pp. 3547. Available at: https://doi.org/10.3390/rs13173547 (Accessed: 6th April 2022)

[7] Javaheri, B. (2021) 'Speech & Song Emotion Recognition Using Multilayer Perceptron and Standard Vector Machine.' *Preprints.org.* Available at: https://doi.org/10.20944/preprints202105.0441.v1 (Accessed: 6th April 2022)

[8] Liu, S., Du, H., Feng, M. (2020). Robust Predictive Models in Clinical Data—Random Forest and Support Vector Machines. In: Celi, L., Majumder, M., Ordóñez, P., Osorio, J., Paik, K., Somai, M. (eds) *Leveraging Data Science for Global Health. Springer, Cham*. Available at: https://doi.org/10.1007/978-3-030-47994-7_13 (Accessed: 6th April 2022)

[9] Mojtaba, F. (2023) "A kernel-based method for solving the time-fractional diffusion equation". *Numerical methods for partial differential* equations (0749-159X), 39 (3), pp. 2719.

[10] Palanivinayagam A, Damaševičius R. (2023) 'Effective Handling of Missing Values in Datasets for Classification Using Machine Learning Methods.' *Information*. 14(2), pp. 92. Available at: https://doi.org/10.3390/info14020092 (Accessed: 6th April 2022)

[11] Zou, D., Cao, Y., Zhou, D. (2020) 'Gradient descent optimizes over-parameterized deep ReLU networks.' *Mach Learn*. 109, pp. 467–492. Available at: https://doi.org/10.1007/s10994-019-05839-6 (Accessed: 13th April 2022)


## Appendix 1 – Glossary

| Term | Definition |
|---|---|
| Perceptron | A simple binary classification algorithm that is used to classify input data into one of two classes based on a linear decision boundary. |
| Softmax | A function that converts a vector of real numbers into a probability distribution over several classes. |
| Neural network | A type of machine learning algorithm that consists of layers of interconnected nodes, each of which performs a nonlinear transformation on its input. |

| | |
|---|---|
| Kernel Function | A function that measures the similarity between two inputs in a high-dimensional feature space, often used in machine learning algorithms such as support vector machines. |
| Activation Function | A function that determines the output of a neuron in a neural network based on its input, often used to introduce nonlinearity into the model. |
| Optimization Function | A function used to optimize the parameters of a machine learning model by minimizing a loss function, often using methods such as gradient descent. |
| Momentum | A technique used in optimization algorithms that takes into account the previous updates to the model's parameters when computing the current update. |
| Learning Rates | A hyperparameter that determines the step size used in optimization algorithms such as gradient descent. |
| Receiver Operation Curve | A graphical plot that illustrates the performance of a binary classification algorithm by showing the trade-off between true positive rate and false positive rate. |
| F-1 Score | A measure of a classification algorithm's accuracy that combines precision and recall. |
| Precision | The ratio of true positive predictions to the total number of positive predictions made by a binary classification algorithm. |
| Recall | The ratio of true positive predictions to the total number of actual positive instances in the data. |

**Appendix 2 – Implementation details**

The target class in the original dataset contained 7 classes, the first few iterations of the models were designed to perform multi-class classification tasks, however, the 7 classes were heavily imbalanced and it became too complex to artificially populate each class. The models also had low accuracy because of this imbalance and the multi-class nature of the target class. It was decided to group the target classes into 1 and 0 and changed the task of the project into a binary classification instead. As a result, the models greatly improved in accuracy.

The first few test runs of the models were used on data that were not standardized, and with an unbalanced dataset. Accuracy got better after the data points were standardized prior to being fed into the model. Before standardization, the accuracies of the SVM and MLP models were 79.18% and 77.45%. After standardization, SVM and MLP went to 81.06% and 87.04%. Accuracy also improved further after implementing scikit learn’s resampling tools to balance the target class of 1 and 0.

GridSearchCV and RandomizedSearchCV were used to find the combination of hyperparameters that yielded the best results in the model. As it was computationally expensive and time-consuming, the final notebook did not contain the implementation of those techniques, the final notebook only contains the hyperparameter configurations that are the best from those techniques.

Although GridSearchCV recommended using a constant learning rate, in the final model an adaptive learning rate yielded better results than the configurations that were recommended from grid search. The results in grid search had an accuracy of 85.81% but the same accuracy could not be replicated using the hyperparameters that were suggested.