Comparative Analysis of Multilayer Perceptron and Support Vector Machines for Binary Classification of Banana Quality

Dimitris Nikandrou

Dimitris.Nikandrou@city.ac.uk

Abstract

This coursework showcases a comparison between the Multilayer Perceptron (MLP) and the Support Vector Machine (SVM) in the binary classification of banana quality. The models produced were all analysed extensively, and the findings concluded that the SVM performed slightly better than the MLP.

1. Introduction

As agricultural technology advances, selecting the right tools for quality assessment of fruit and vegetables becomes crucial, and with the major developments in the fields of machine learning and neural computing this task has been made more efficient in recognising patterns in the fruit characteristics that could indicate their quality. The aim of this coursework is to investigate two advanced algorithms, the Multilayer Perceptron (MLP) and the Support Vector Machine (SVM), focusing on their application and performance in the binary classification of banana quality [4] and assessing their performance through various metrics.

Through analysing dataset attributes like weight, acidity and sweetness, the study's goal is to determine the superior algorithm for ensuring the quality of agricultural products. This comparative analysis highlights the role of neural computing in enhancing the precision of agricultural practices.

In this coursework, a brief explanation will be provided on the dataset used, as well as the exploratory data analysis and pre-processing done, prior to the dataset being used to train the two algorithms. Moreover, the methodology used during the implementation phase will be explained and compared. A critical evaluation of the results obtained will also be conducted for the two approaches used and finally a conclusion that determines if the hypothesis statement made before the implementation was indeed correct or wrong.

1.1. Multilayer Perceptron (MLP)

The MLP is an artificial neural network that is used in machine learning for solving a variety of classification tasks, including binary classification. It is a supervised learning classifier that consists of at least 3 layers, the input layer, the hidden layer(s) and the output layer, that make up the network [2]. Each layer is made up of neurons, where each of the neurons in one layer connects to every neuron in the next layer, which are typically associated with weights that can be adjusted during training.

The MLP is very flexible, meaning that it can be utilised for various classification and regression tasks, and it is also a highly customizable algorithm and can be easily adjusted for the needs of a dataset. Additionally, it is also highly robust to noise and external factors. However, MLPs can be vulnerable to overfitting when the network is highly complex, if early stopping is not applied. It can also be considered computationally expensive, especially when the number of neurons is increased and the constant updates on weights [2]. Furthermore, it can be susceptible to feature scaling since it is sensitive to the magnitude of the input data.

1.2. Support Vector Machines (SVM)

The SVM is a supervised learning algorithm widely used for both classification and regression tasks, though it is particularly known for its effectiveness in classification, including binary classification scenarios [3]. The SVM algorithm works by identifying the best dividing line that separates classes in the data with the largest possible space between them, enhancing its ability to classify accurately.

SVMs can be highly versatile, because the hyperplane can be defined in terms of kernel functions (e.g. linear, polynomial etc.), which means that they can be easily adapted to the needs of the dataset. Moreover, SVMs are highly robust against overfitting and are also memory efficient which means that they are computationally efficient. Some drawbacks of the SVMs include the fact that the choice of the kernel trick is crucial, which means that a bad choice might lead to bad results. Moreover, SVMs can be sensitive to noise and outliers in the data, hence a robust data preprocessing phase is essential, to obtain optimal results.

1.3. Hypothesis Statement

The hypothesis that was made before implementing the two networks was that the SVM would perform better than the MLP.

2. Dataset

The dataset that was used for this coursework, was obtained from Kaggle. The dataset consists of 8000 rows and 7 features, plus an additional column which is the target variable column. The distribution of the target variable was examined, to ensure that it was balanced (49.9% Bad and 50.1% Good). The target variable was initially in categorical format, hence encoding was used to assign the values of 1 for ‘Good’ quality and 0 for ‘Bad’ quality. Below, a table with the summary statistics of the dataset.

	count	mean	std	min	25%	50%	75%	max
Size	8000.0	-0.747802	2.136023	-7.998074	-2.277651	-0.897514	0.654216	7.970800
Weight	8000.0	-0.761019	2.015934	-8.283002	-2.223574	-0.868659	0.775491	5.679692
Sweetness	8000.0	-0.770224	1.948455	-6.434022	-2.107329	-1.020673	0.311048	7.539374
Softness	8000.0	-0.014441	2.065216	-6.959320	-1.590458	0.202644	1.547120	8.241555
HarvestTime	8000.0	-0.751288	1.996661	-7.570008	-2.120659	-0.934192	0.507326	6.293280
Ripeness	8000.0	0.781098	2.114289	-7.423155	-0.574226	0.964952	2.261650	7.249034
Acidity	8000.0	0.008725	2.293467	-8.226977	-1.629450	0.098735	1.682063	7.411633

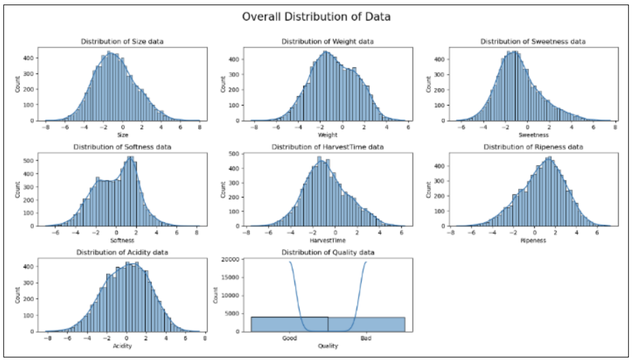
Figure 1: A summary statistics table of the dataset.

2.1. Initial Data Analysis

An exploratory data analysis and data pre-processing steps were conducted.

The exploratory data analysis started with a pie chart to visualize the class distribution within the dataset, clearly indicating the proportion of 'Good' versus 'Bad' quality bananas. A correlation matrix was then computed and visualized to understand the strength between the relationships of the features. Insightful patterns were further explored through pair plots, differentiated by the target class, to look at further insights in the dataset.

Histograms for each variable were generated to examine individual data distributions (see Figure 2), while box plots were created to detect outliers (see Figure 3). These EDA techniques are a stable in every data analysis task and provide the user with a clearer understanding of the dataset that he is working with.



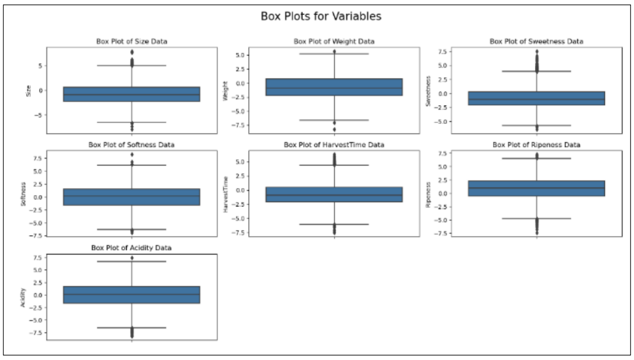


Figure 2: Histograms showing data distribution.

Figure 3: Boxplots showing the spread of the

In the data preprocessing, the initial step was to standardize the numerical features, an essential step for models such as MLP which are sensitive to feature scales. The categorical target, 'Quality', was encoded into a numerical binary format, making the data suitable for subsequent machine learning algorithms. These preprocessing measures are key in limiting potential biases and scaling issues, readying the dataset for accurate model training and evaluation.

3. Methods

In this study, a structured approach was used to train and evaluate two machine learning models, the MLP and the SVM, for classifying banana quality. The data was initially partitioned, allocating 70% for training and 30% for testing, using the holdout method. The testing data was further split to create a validation set for hyperparameter tuning, at a 50:50 ratio.

The MLP network was firstly prepared by converting the sets to tensors and then structured into layers, an input layer, two hidden layers and an output layer, as well as the ReLu activation function. Moreover, for the MLP momentum, patience and weight decay were implemented to improve the robustness of the model. Early stopping was also implemented, meaning that after no improvement in the model’s accuracy was observed, the model would stop running.

For the SVM, a grid search methodology was employed to determine the most effective settings, experimenting with various C and Gamma values to achieve the best validation accuracy.

Training durations were also recorded for each model setup. All the methods listed above were used to support a detailed comparison of how the MLP and SVM perform under different parameters and configurations.

3.1. Architecture and Parameters Used for the MLP

The architecture of the MLP for classifying banana quality starts with an input layer that processes seven features. It includes two hidden layers with varying neuron setups, from 10 and 5 to 128 and 64, to explore different network complexities. The model tests various learning rates from 0.001 to 1 to optimize learning speed and uses a momentum of 0.6 to speed up updates. To prevent overfitting, it employs a small weight decay of 0.0001, penalizing larger weights.

Training uses a patience setting of 10 epochs, stopping early if there’s no improvement, aiming for a maximum of 1000 epochs but stopping sooner if optimal performance is achieved early. This setup ensures the training is both effective and efficient.

3.2. Architecture and Parameters Used for the SVM

The SVM architecture used for classifying banana quality involves systematic testing of different model parameters to find the most effective combination. The SVM model is trained using a range of value configurations for the parameter 'C' and the kernel coefficient 'gamma', with 'C' values set at [0.01, 0.1, 1, 10] and 'Gamma' values at [0.01, 0.1, 1, 10]. The training process assesses both linear and radial basis function (rbf) kernels to determine which combination yields the highest accuracy on the validation set.

This approach not only ensures that the SVM is finely tuned to the dataset but also allows for a clear comparison of how different settings affect performance. The comprehensive testing of parameters is crucial for optimizing the SVM's effectiveness in classifying the quality of agricultural products accurately [3].

4. Analysis and Critical Evaluation of Results

4.1. Model Selection

When comparing the MLP and SVM models, key differences in performance emerged. For MLP, a variety of hidden neuron sizes and learning rates were tested to find the best match for the data. The combination of [64, 32] neurons with a learning rate of 0.01 stopped early at 140 epochs, yielding the highest accuracy of 0.984. This suggests that a moderate amount of complexity in the model, with a moderate learning speed, worked best for MLP.

For SVM, the model's accuracy was sensitive to the right mix of C and Gamma values, especially with the ‘rbf’ kernel. The most accurate SVM model used a C value of 10 and a Gamma value of 0.1, achieving a 0.985 accuracy. As observed from the table below, lower, or higher Gamma values either led to poorer performance or did not change the accuracy by much.

The MLP's performance appeared to be highly sensitive to its initial neuron weights, meaning small changes there could lead to different results. In contrast, the SVM showed a strong dependence on its hyperparameters, with the right combination leading to the most accurate predictions. As observed from the tables below, the SVM performed slightly better than the MLP (see Figures 4 & 5).

Multilayer Perceptron (MLP)			
Hidden Neuron Size	Learning Rate	Early Stopping at Epoch	Accuracy
[10, 5]	0.001	999	0.964
[20, 10]	0.001	999	0.967
[64, 32]	0.001	980	0.983
[128, 64]	0.001	727	0.981
[10, 5]	0.01	261	0.971
[20, 10]	0.01	244	0.97
[64, 32]	0.01	140	0.984
[128, 64	0.01	111	0.978
[10, 5]	0.05	196	0.975
[20, 10]	0.05	76	0.97
[64, 32]	0.05	56	0.98
[128, 64]	0.05	83	0.982
[10, 5]	0.1	85	0.978
[20, 10]	0.1	65	0.974
[64, 32]	0.1	78	0.976
[128, 64]	0.1	66	0.976
[10, 5]	1	15	0.663
[20, 10]	1	28	0.753
[128, 64]	1	11	0.925

Figure 4: Grid Search for MLP

Support Vector Machine (SVM)			
C Value	Gamma Value	Kernel	Accuracy
0.01	N/A	linear	0.881
0.01	0.1	rbf	0.944
0.01	1	rbf	0.846
0.01	10	rbf	0.463
0.1	N/A	linear	0.882
0.1	0.01	rbf	0.898
0.1	0.1	rbf	0.967
0.1	1	rbf	0.973
0.1	10	rbf	0.463
1	N/A	linear	0.881
1	0.01	rbf	0.928
1	0.1	rbf	0.972
1	1	rbf	0.976
1	10	rbf	0.852
10	0.1	rbf	0.985
10	N/A	linear	0.881
10	0.01	rbf	0.963
10	0.1	rbf	0.979
10	1	rbf	0.973

Figure 5: Grid Search for SVM

4.2. Algorithm Comparison

The confusion matrices for the MLP and SVM models both show a high level of accuracy in classifying the quality of bananas, with SVM having a slight edge (see Figures 6 & 7). This small advantage in SVM's performance could be due to its ability to create more clear boundaries between classes, especially when the data has a clear margin of separation. The SVM's slightly lower false positives suggest that it may be better at avoiding overfitting compared to MLP.

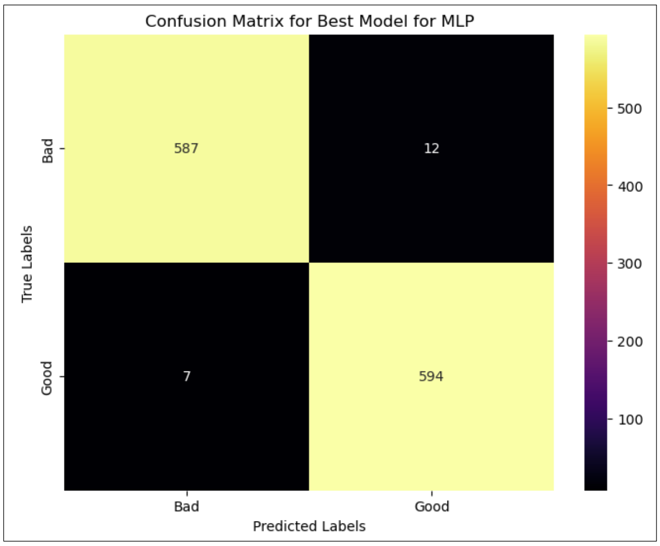


Figure 6: Confusion Matrix for MLP

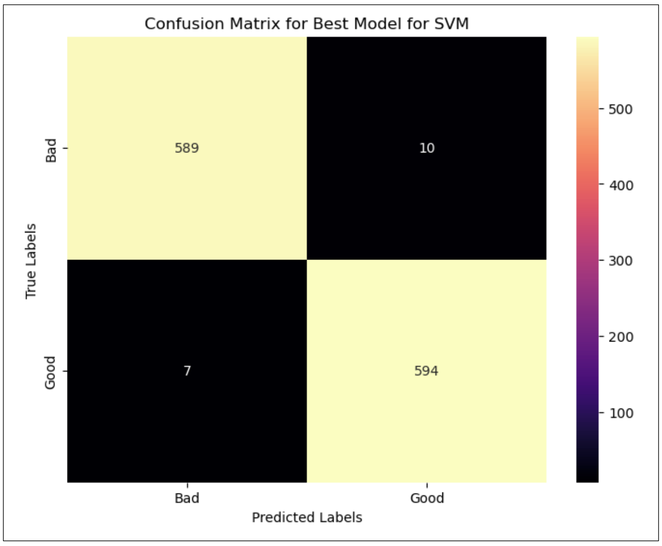


Figure 7: Confusion Matrix for SVM

Analysing the ROC curves, both models score impressively, but SVM's AUC of 0.986 indicates it has a slightly better performance in sensitivity (true positive rate) and specificity (true negative rate) than MLP's 0.984. This suggests the SVM performs slightly better at distinguishing between the classes, which might be due to its effective handling of the kernel trick, which optimizes the decision boundary.

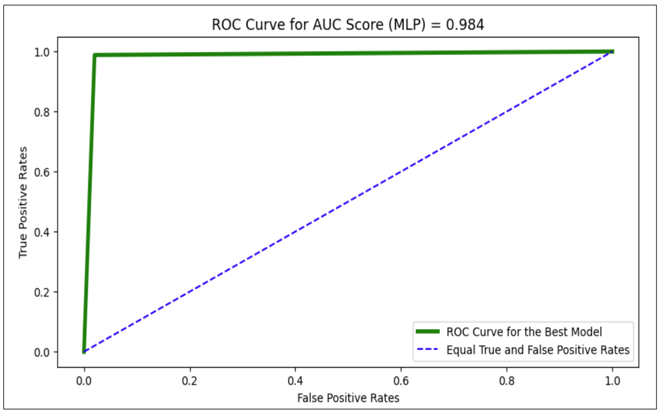


Figure 8: ROC Curve for MLP

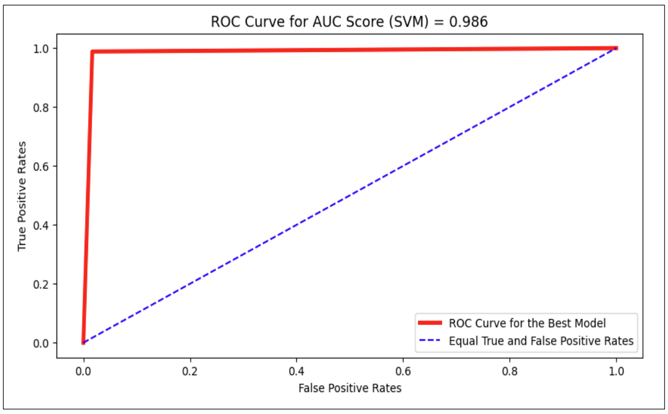


Figure 9: ROC Curve for SVM

It was concluded that the SVM model is slightly more effective for this particular classification task. This efficiency is typically seen in SVMs when they are applied to smaller datasets, such as this one. SVMs are excellent at drawing a clear boundary (hyperplane), between the classes, which is crucial for making accurate predictions. The kernel trick is also able to handle more complex data patterns without increasing computational complexity, which was demonstrated during the experiments, as the training time required for the SVM was less than that of the MLP.

On the other hand, MLPs require larger datasets to demonstrate their full potential, due to their reliance on initial weight settings and the learning rate [1]. These factors are critical in helping the MLP converge faster.

	MLP	SVM
True Negatives	587	589
False Positives	12	10
False Negatives	7	7
True Positives	594	594
Accuracy	0.984	0.985
Precision	0.98	0.98
Recall	0.988	0.988
F1 Score	0.984	0.984

Figure 10: Results Summary Table

In real world applications, the selection between SVM and MLP models relies on the characteristics of the dataset, the computational resources that are available, and the implications of potential errors that may appear during the implementation phase. If the dataset is extensive and somewhat unstructured, an MLP may be more appropriate, provided that fine hyperparameter tuning has been completed. However, for smaller and cleaner datasets, an SVM could be preferable. A balance of factors should be considered carefully before deploying any two of the algorithms, in real life applications.

5. Conclusion, Lessons Learned & Future Work

In conclusion, this coursework dived deep into the implementation and understanding of two popular Neural Computing algorithms. Each model's performance was analysed within the context of binary classification for banana quality. The SVM model demonstrated a slight superiority in accuracy, due to its robustness and efficiency in handling smaller and cleaner datasets. Its kernel trick and hyperparameter sensitivity allowed for a close higher performance, compared to the MLP. It can be said that both algorithms can be used in this specific case and produce excellent results, however due to the nature of this dataset, the SVM was found to be more appropriate.

This coursework highlighted the importance of precisely tuning the hyperparameters of both algorithms used, to achieve the most optimal results, as well as the importance of using any ML or Neural Computing algorithm in context with the dataset. In future work, several other algorithms like Boltzmann Machines or RNNs could be deployed on the dataset. Moreover, k-fold data partitioning could be used, as well as experimentation on larger datasets and increased complexity in the algorithms deployed.

6.0 References

[1] Zanaty, E.A., 2012. Support vector machines (SVMs) versus multilayer perception (MLP) in data classification. *Egyptian Informatics Journal*, *13*(3), pp.177-183.

[2] Gardner, M.W. and Dorling, S.R., 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, *32*(14-15), pp.2627-2636.

[3] Patle, A. and Chouhan, D.S., 2013, January. SVM kernel functions for classification. In *2013 International conference on advances in technology and engineering (ICATE)* (pp. 1-9). IEEE.

[4] Dataset: https://www.kaggle.com/datasets/l3llff/banana

Appendix 1-Glossary

Multilayer Perceptron (MLP): A type of artificial neural network that consists of multiple layers of neurons, each connected to subsequent layers, commonly used for complex pattern recognition.

Support Vector Machine: A supervised learning algorithm used for classification and regression tasks, which finds the optimal boundary (hyperplane) that maximizes the margin between different classes.

Momentum: A technique that helps accelerate gradients vectors in the right directions, hence leading to faster converging.

Early Stopping: A method used to stop the training of a machine learning model if there is no significant improvement in performance over a given number of epochs to prevent overfitting.

C Values: In SVM, the regularization parameter that determines the trade-off between achieving a low error on the training data and minimizing the model complexity for better generalization.

Gamma Values: A parameter in kernel-based techniques like SVM, that influences the decision boundary; a low value creates a loose fit, while a high value creates a tight fit.

Kernel Trick: A method used in SVM that transforms data into a higher-dimensional space to make it possible to perform linear separation when data is not linearly separable in the original space.

Linear Function: A function that models a direct proportional relationship between input and output, represented by a straight line in algebra.

Radial Basis Function: A function used in various types of machine learning algorithms, typically as a kernel for SVM, which measures similarity based on distance from a central point.

Grid Search: A search method used to find the optimal hyperparameters of a model by systematically varying each hyperparameter over a specified range of values and evaluating the model's performance.

Appendix 2-Implementation Details & Figures

In this coursework, the implementation process began with a detailed exploratory data analysis to understand the underlying patterns and characteristics of the dataset, which consisted of various attributes related to banana quality. This phase involved visualizing the distribution of the classes and key features using histograms, pie charts, and correlation matrices to assess relationships.

Following the EDA, data preprocessing was conducted to prepare the dataset for modelling. This included encoding categorical variables into a numerical binary format and standardizing the numerical features.

The core of the implementation involved setting up, training, and evaluating the two different machine learning models. Each model was carefully constructed using appropriate hyperparameters. The MLP was tested across various combinations of neuron configurations and learning rates, while the SVM was adjusted through different values of C and Gamma in a grid search approach to identify the optimal hyperparameters for maximum classification accuracy.

The training process for both models included a mechanism for early stopping to prevent overfitting. Performance metrics such as accuracy, precision, recall, and F1 score were calculated to evaluate and compare the performance of each model in classifying the quality of bananas.

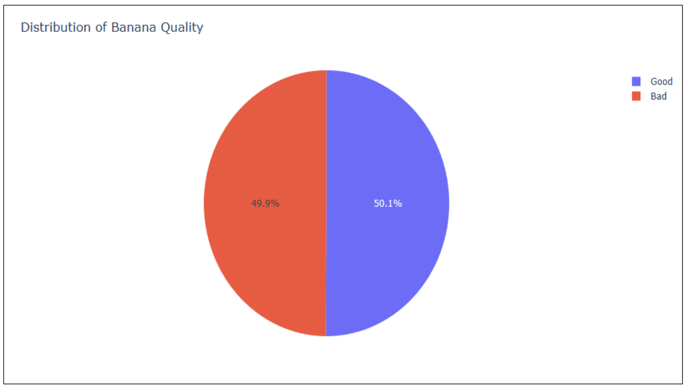


Figure 11: Distribution of Target Variable

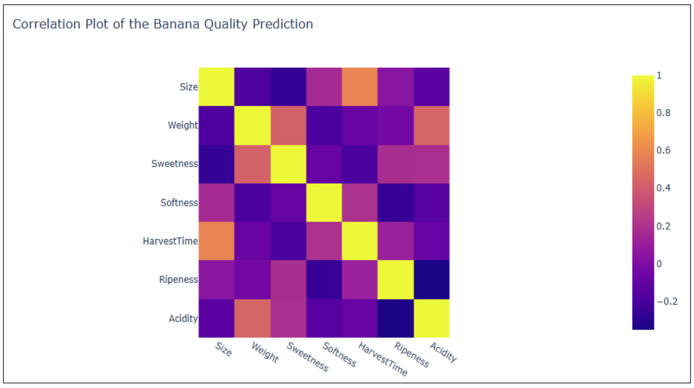


Figure 12: Correlation Heatmap

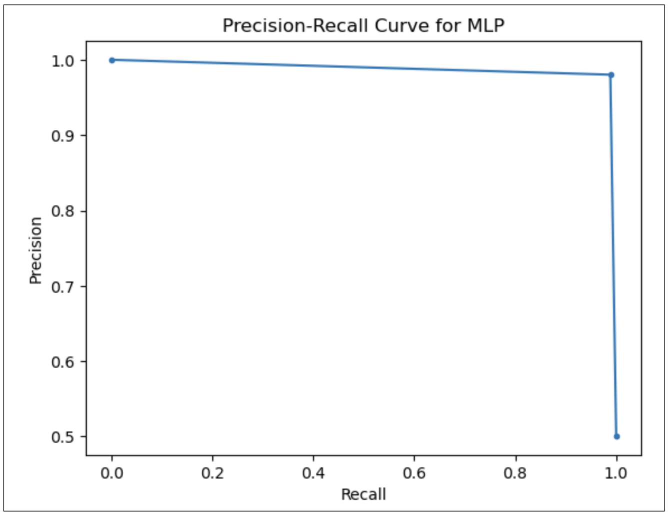


Figure 13: Precision vs Recall Curve for MLP

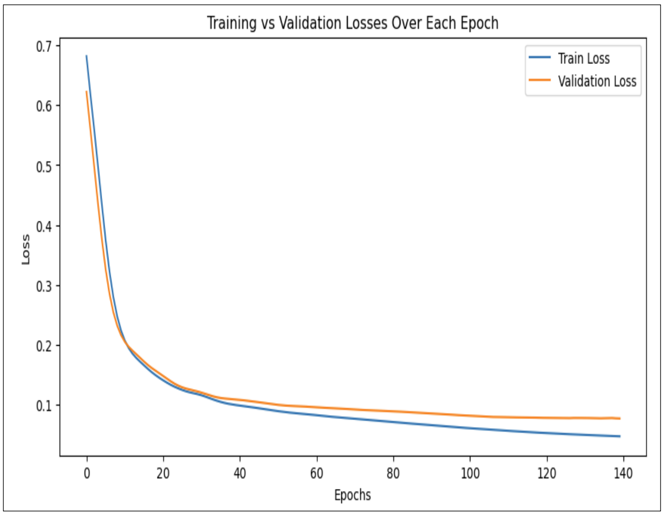


Figure 14: Training vs Validation Losses for MLP

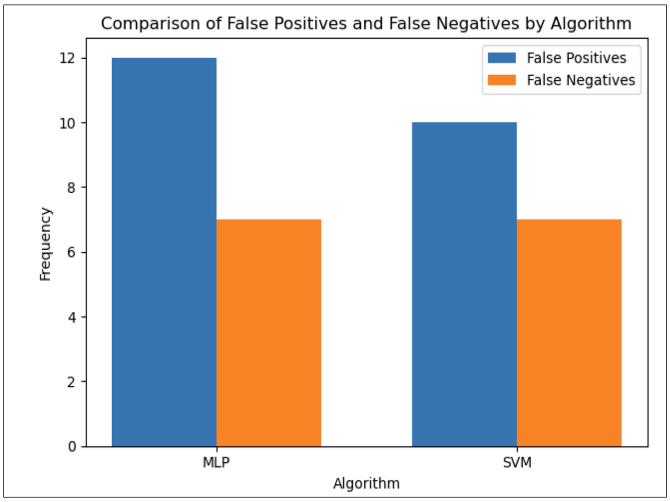


Figure 15: False Positives vs False Negatives for each algorithm