A Comparative Study in Classification of Pumpkin seeds through Multilayer Perceptron and Support Vector Machines

Sahan Chowdhury
Sahan.Chowdhury@city.ac.uk

Abstract

This study investigates two neural computing methods, Multilayer Perceptron (MLP) and Support Vector Machine (SVM), for the classification of pumpkin seed types based on numerical features. The dataset used within this study was pre-processed and model parameters were optimised and validated using grid search. The optimal models were then tested to the test data. ROC curves alongside confusion matrices were used to assess model performance, and results indicated that SVM outperformed MLP in classification accuracy and efficiency.

1. Introduction

The importance of classification task has surged in recent years, particularly in the food and agriculture sector. Accurate classification of products plays a vital role in streamlining the chain from food production to sales chain. Amongst the vast number of agricultural products are pumpkins and pumpkin seeds which belong to the *Cucurbita* family. In fact, global production of pumpkins reached 61 billion pounds across the globe in 2017 [1]. Pumpkin seeds are edible seeds derived from pumpkins, they are known for their health benefits and wide range of nutritional values.

With currently over 45 different varieties of pumpkin seed [1], the need to classify pumpkin seeds accurately is vital. This study focusses on 2 different types of pumpkin seeds originated from Turkey, using numerical features to characterise to each specific seed type. Multilayer perceptron (MLP) and Support vector machines (SVM) were utilised, compared, and critically evaluated. The study of Necati Çetin et al [2] and alongside the study of M. Koklu, S. Sarigil, and O. Ozbek [3] were also used alongside to support findings.

2. Data

The pumpkin seed dataset was obtained from a dataset repository [4], compromising of 2,500 instances distributed over 13 columns. 12 of the columns are features, which were as follows; *'Area', 'Perimeter', 'Major_Axis_Length', 'Minor_Axis_Length', 'Convex_Area', 'Equiv_Diameter', 'Eccentricity', 'Solidity', 'Extent', 'Roundness', 'Aspect_Ration' and 'Compactness'*, while the final column is the class labels. The two distinct classes: *'Çerçevelik'* and *'Ürgüp Sivrisi'* were encoded as '1' and '0' correspondingly. Of the entire dataset there was 1300 instances of '*Çerçevelik'* and 1200 of '*Ürgüp Sivrisi'*, indicating there was a class dominance of 1.08 times by '*Çerçevelik'*.

3. Exploratory data analysis

Prior to model training, exploratory data analysis was conducted to understand the dataset characteristics and ensure overall data quality was maintained before applying it to the

models. Firstly, all the features were converted into *float* data type to ensure consistency of features type and enhance precision. Upon initial inspection of the dataset attributes, it was observed that pumpkin seeds have an average area of 80,658. There is also a lot of variation between seeds shape as the range of '*major axis length'* and '*minor axis length'* varied. The seeds also typically exhibit an elongated shape this is reflected by the average eccentricity equating to 0.86. Where eccentricity closer to 1 indicates the shape is elongated. Descriptive statistics addressed the need to normalise the data, ensuring the effects of scaling issues and feature dominance are mitigated for optimal model performance.

There were no missing values denoted. A histogram plot to visualise the distribution was created for each feature as depicted in *figure (1)*. From the figure it was evident majority of the features had no skew and followed closely to a normal distribution however the features '*Eccentricity*' and '*solidity'* were left skewed, these were addressed through Box-cox transformation. Additionally, some outliers were observed from boxplots. z scores were used to set thresholds and display all values that exceed the threshold for every feature however no outlier was deemed extreme nor necessary to be removed. *Figure (2)* displays the class distribution through a RadViz plot, it was clear that '*Ürgüp Sivrisi'* seed type was closer to feature points. To get a more detailed overview violin subplots were created for every feature, '*Ürgüp Sivrisi'* had the greater variability overall. Lastly, a correlation matrix was computed see *figure (3)*, to explore feature relationships between different features. Notably several features displayed a Pearson correlation coefficient 1, as these were directly proportional i.e. *'Area'* and *'Equiv diameter'*, on the other hand some features showed a negative correlation, but for the most part there was a weak correlation between features.

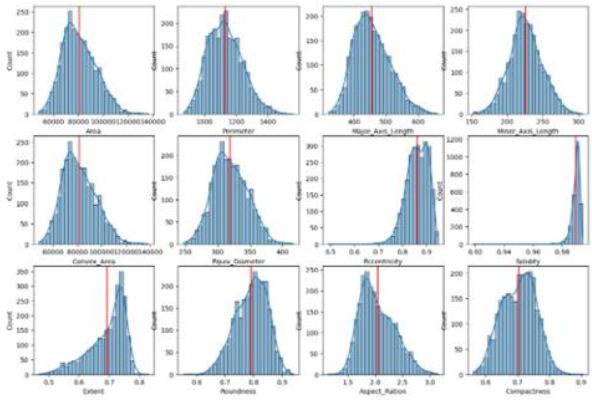

Figure (1): Histogram subplots displaying distribution of features.

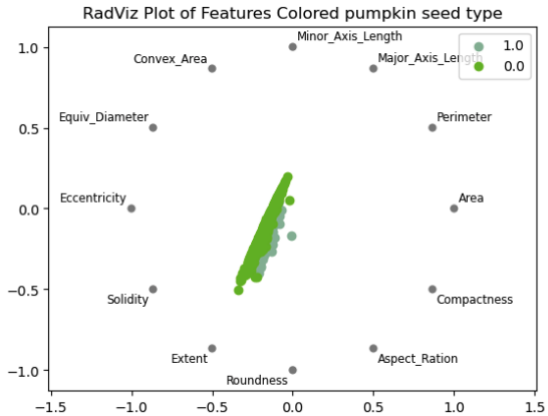

Figure (2): RadViz plot of class distributions

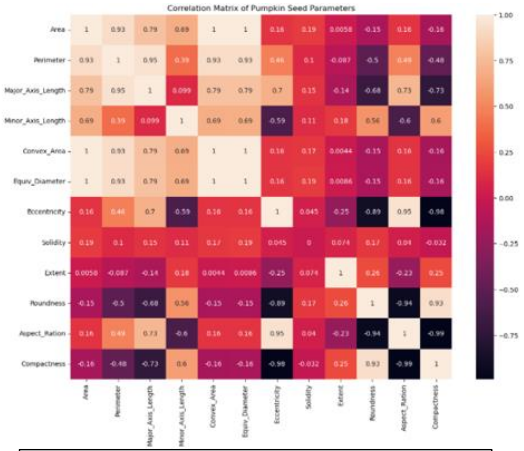

Figure (3): Correlation matrix

4. Summary of Algorithms

Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) is a supervised feedforward artificial neural network. It consists of an input layer, output, and a minimum of one hidden layer. It also contains activation functions and weights to obtain the output. MLP's work only in a forward direction, all nodes are fully connected to the network pass its value to the coming node [5], hence it's a feedforward network.

Advantages (MLP)	Disadvantages (MLP)
- **High accuracy**: The use of backpropagation means it can achieve a high accuracy rate. - **Scalability**: Works well with large datasets. - **Versatility:** Can be applied to both classification and regression problems.	- **Black Box model**: Often referred to as a 'black box model' as its unclear how the network will make its predictions. [6] - **Requires fine-tuning**: For best performance, the model needs to be fine-tuned optimally during training.

Support Vector Machine (SVM)

Support vector machine is a supervised machine learning algorithm used for classification and regression tasks. It identifies the optimal hyperplane that separates the data points into different classes [7]. SVM's are excellent at handling data with a lot of features.

Advantages (SVM)	Disadvantages (SVM)
-**Handling many features:** SVM's handle high dimensional data very well. - **Less prone to overfitting:** SVM's can maximise margin between classes, reducing risk of overfitting.	- **Sensitive to noise:** Does not perform well when there is noise in dataset such as outliers. - **Computationally expensive:** larger datasets require more memory.

5. Hypothesis

Based on the study of M. Koklu, S. Sarigil, and O. Ozbek [3], which compared models on the same dataset, SVM only performed slightly better with an accuracy of 88.64%, compared to MLP obtaining an accuracy of 88.52%. The models seemed to fair very closely, with SVM taking a slight edge. Furthermore, due to the dataset having a range of features and modest size of the dataset, SVM is expected to perform slightly better. However, the study of Necati Çetin et al [2] which evaluated models across different classes of pumpkin seeds, indicated that MLP generally performed better, this is further supported by the study of M. Koklu, S. Sarigil, and O. Ozbek which found the rate of correctly defined positive patterns to be found in the MLP model. Thus, it is expected for MLP to have higher recall rate and SVM with the higher accuracy rate.

6. Methodology

The dataset was divided into training, validation, and testing, where 60% of the dataset was used in training, 20% for validation and the 20% for testing for comparing both algorithms. Where the validation set, is used to evaluate performance for different combinations of hyperparameters. The base MLP and SVM model were created. the data was scaled using the *min_max scalar*, which was used to improve accuracy. Subsequently grid search was applied to both MLP and SVM individually to find the optimal hyperparameters using a 2-fold cross validation approach, with the required call-back functions. The best models were selected and saved into a pickle file and tested against the test data.

7. Architecture and Parameters used for the MLP and SVM

For the Multilayer Perceptron (MLP) model, the input size was set 12 as there was 12 feature columns in the dataset, hidden layers were set to 50 as this provides a good balance between model complexity and performance. The output size was set to 2 as there is only 2 classes for prediction and dropout was set to 0.2 to reduce to risk of overfitting. *ReLU* activation function was used for non-linearity and *Softmax* activation function was also used in the output layer to compute class probabilities. The model had 3 hidden layers and after each hidden layer dropout was applied. Call back functions including early stopping and epoch stopping were used to halt training process when no significant improvement was observed. Finally, Adam optimizer was used alongside weight decay for regularisation. The model was then fitted using the *NeuralNetClassifier* from Skorch library and relatively low training and validation accuracy was obtained initially. This method was repeated but with scaled data, training and validation accuracy significantly improved. Hyperparameter tuning was conducted using grid search with specified criteria. Runtime analyses were also conducted to assess computational efficiency.

As for the Support vector machine (SVM) model, the support vector classification (SVC) mode from Scikit-learn's SVM module was applied. The scaled data ready was directly used for the model. A *linear* kernel was chosen due to its computational efficiency and random state was set for reproducibility. The model was then optimized with a hyperparameter grid search for '*regularization parameter* (C)', '*gamma'* and '*kernel* type'. The final model was then created with the best parameters obtained from grid search and fitted. The best parameters for both MLP and SVM are shown below highlighted in red.

MLP Hyperparameters	Values
Learning rate	0.0001, 0.01, 0.1
Max Epochs	10, 50, 100
Optimiser	Optim.Adam, Optim.SGD
Batch Size	32, 64

Table (1): MLP Hyperparameters trained over grid search. Values highlighted in red were denoted as the optimal parameters.

SVM Hyperparameters	Values
Regularization parameter. (C)	0.001, 0.1, 1.0, 10
Kernel	Linear, RBF
Gamma	0.001,0.1
Degree	2,3,4

Table (2): SVM Hyperparameters trained over grid search. Values highlighted in red were denoted as the optimal parameters.

8. Results, Findings & Evaluation

The initial MLP model trained with the original data obtained a very low accuracy of 48% on the training set and the validation set, this may be due to the fact the magnitude of the scales amongst the original data can result in a model that learns large weight values [8], this could cause the model to become unstable, potentially a reason for poor performance.

Upon feature scaling, the model's performance significantly increased to 88.8% for the validation data. After optimizing parameters through grid search the validation accuracy increased to 89.4%. Finally, when evaluated on the test data, the accuracy was 86.8%.

Initial validation and training accuracy on the base SVM model with the scaled data was 89.4% and 87.5% respectively. This initial performance already surpassed the accuracy of the MLP model without grid search. After grid search optimization, the SVM and the best parameters were used to create a new architecture the new model had a validation accuracy of 89.6%. Subsequently the test accuracy was 87.2%. This outcome supports the hypothesis that the SVM was expected to attain high accuracy and perform slightly better than MLP. The accuracy score obtained in the final models are also similar of that of the study of M. Koklu, S. Sarigil, and O. Ozbek. The SVM model also was much more efficient taking only around 0.02 seconds to train compared to the MLP model taking 0.25 seconds.

Figure (5) and *(6)* displays the confusion matrix of the unseen test data for both MLP and SVM models, the false negatives and true negatives are identical this could be due to the class imbalance, as identified class encoded as '1' – '*Çerçevelik'*, had a class dominance, and therefore there may have been model bias. SVM only has 2 greater true positives.

Recall and F1 scores were identical too, except for precision where SVM had a higher precision (0.85) for identifying class '1', this could again be down to the class imbalance. The study of M. Koklu, S. Sarigil, and O. Ozbek also noted very similar precision, recall and f1 scores between MLP and SVM.

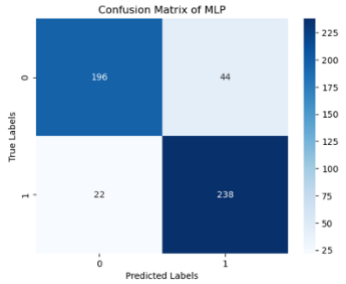

Figure (5): Confusion matrix for MLP

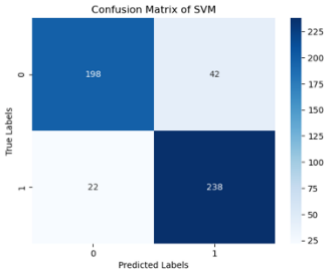

Figure (6): Confusion matrix for SVM

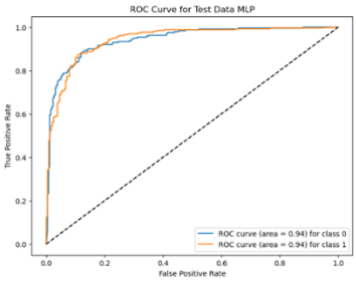

Figure (7): ROC curve for MLP

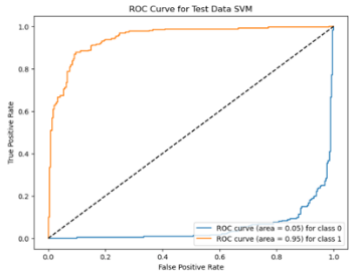

Figure (8): ROC curve for SVM

Receiver Operation Curve (ROC) was also plotted as shown *figure (7)* and *(8)*. The MLP model demonstrated high AUC of 0.94 for both classes. Likewise, the SVM model obtained an AUC of 0.95 for class '1' but a significantly lower AUC of 0.05 for class '0', which surprisingly doesn't align with the high accuracy the SVM model obtained. This low AUC for class '0' could potentially highlight the challenges in differentiating class '0' instances, possibly due to being the minority class. SVMs can be sensitive to class

imbalances, where the model might prioritize the majority class (class 1) at the expense of the minority class (class 0).

9. Conclusion

This study evaluated the use of Multilayer perceptron (MLP) and Support Vector Machine (SVM) for classifying pumpkin seed types based on numerical features. Both models demonstrated high classification accuracies, with SVM slightly outperforming MLP. The SVM model was also much more efficient and faster, highlighting the superiority of the SVM model in this instance. The initial shortcoming of the MLP model highlighted the importance of pre-processing steps, like feature scaling, for enhancing model performance. SVM's lower AUC for class '0' indicated challenges in distinguishing this class, possibly due to class imbalance. Future work to further enhance this study could be using the *SMOTE* technique to address of class imbalances, secondly as the study of M. Koklu, S. Sarigil, and O. Ozbek mentioned having more features such as colour and texture incorporated to the features could potentially improve the performance of the model.

10. Reference

[1] "Pumpkin Seeds," 88 Acres. https://88acres.com/pages/pumpkin-seeds (accessed Apr. 19, 2024).

[2] Necati Çetin et al., "Binary classification of pumpkin (Cucurbita pepo L.) seeds based on quality features using machine learning algorithms," European food research & technology, Nov. 2023, doi: https://doi.org/10.1007/s00217-023-04392-w.

[3] M. Koklu, S. Sarigil, and O. Ozbek, "The use of machine learning methods in classification of pumpkin seeds (Cucurbita pepo L.)," Genetic Resources and Crop Evolution, Jun. 2021, doi: https://doi.org/10.1007/s10722-021-01226-0.

[4] "DATASETS," www.muratkoklu.com. https://www.muratkoklu.com/datasets/

[5] Shiksha.com, 2020. https://www.shiksha.com/online-courses/articles/understanding-multilayer-perceptron-mlp-neural-networks/

[6] "Multilayer Perceptron," Deepchecks. https://deepchecks.com/glossary/multilayer-perceptron/

[7] V. Kanade, "What Is a Support Vector Machine? Working, Types, and Examples," Spiceworks, Sep. 29, 2022. https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector-machine/

[8] J. Brownlee, "How to use Data Scaling Improve Deep Learning Model Stability and Performance," Machine Learning Mastery, Feb. 03, 2019. https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

Appendix

Glossary

Float – Data type that converts numbers into decimal points.

ReLU- Short for rectified linear unit, is an activation function. It outputs the inputs if its positive else it outputs 0.

Softmax - An activation function which turns outputs into probability functions.

Normalisation – Feature scaling where numerical features are transformed into the same scale.

SMOTE – Synthetic minority over sampling technique. Used to address class imbalance, by creating synthetic instances of the class that is less represented (minority class).

Kernel- Used commonly with support vector machines (SVM). Transforms inputs into the required form.

Precision- Classification metric used evaluate accuracy of positive predictions made by a certain model.

Recall – Classification metric that identifies how many instances the specific model has correctly identified the positive instance (true positive) over the actual positive instances.

F1 – Classification metric that combines the mean of precision and recall. Gives an indication of how well the specific model has performed overall.

Hyperparameter – Parameters that are fine-tuned that occur while training a model, i.e. learning rate, batch size.

Grid search – Hyperparameter tuning technique, which exhaustively check every possible combination of hyper parameters to find the best parameters for the model.

Cross validation – a performance measure to evaluate how well a model trains on unseen data by creating multiple folds and using one-fold for validation.

Implementation

Initial comparison displayed the model performed poorly with the original data for MLP in fact the validation and training data both attained the same level of accuracy of 48%. As

a result of the poor accuracy the data was scaled and this instantly improved performance. The scaled data was then used for SVM, and accuracy was already high on validation sets. The use of hyperparameters slightly improved both models' performances. The figures below display the Precision - Recall curves for both models (see figure (9) and (10)). As mentioned, the precision and recall were almost identical for both models, with SVM only having 0.01 higher average precision. Nonetheless both models had almost perfect modelling and performed very well.

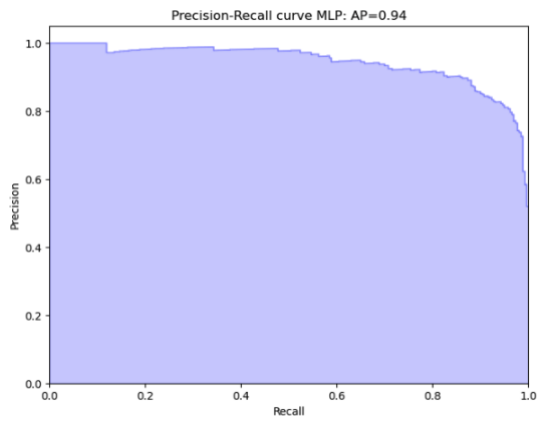

Figure (9): Precision- Recall curve for MLP

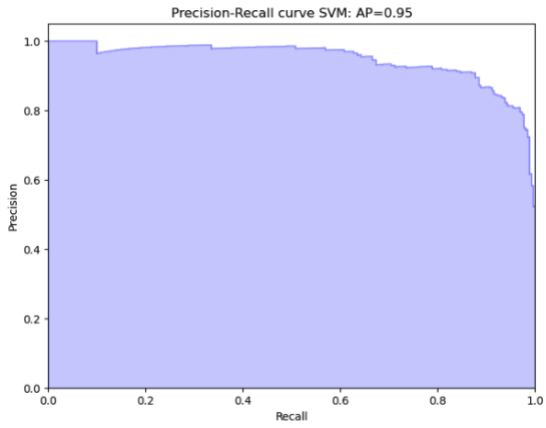

Figure (10): Precision- Recall curve for SVM