A comparative study between Multilayer Perceptron and Support Vector Machine applied to predicting quality of wine.

Dmitry Borisovskiy

1. Brief description of the problem and summary of the used models

1.1. Introduction

The process of wine making and wine drinking has ancient roots and traditions which can be traced up to 8000 years back (Grainger, 2009). However, in recent decades wine became popular as never before and therefore wine quality assessment is currently a topic of interest not just among the ordinary consumers and wine producers but also among scientists from different areas of expertise including data science. There are different studies have been done on application of machine learning to wine classification according to its quality attributes (Hu et al, 2016) and use of data mining techniques in predicting human wine taste preferences (Cortez et al, 2009). This paper aims to study the effectiveness of two of the most popular machine learning algorithms - Multilayer Perceptron (MLP) and a Support Vector Machine (SVM) in distinguishing top quality wine from all other quality levels of wine based on their physiochemical characteristics.

1.2. Support Vector Machine (SVM)

SVM is most commonly referred as a binary classification algorithm that aims to draw a boundary to separate two data classes where support vectors are the closest to the boundary training points of each class (Bahrambeygi and Moeinzadeh, 2017). SVM uses non-linear mapping function to convert data into high dimension space for a better class separation (Batuwita and Palade, 2012). SVM is normally efficient in terms of processing time and computational power requirement. It can also be applied to both linearly and non-linearly separable data which makes it suitable to different kinds of machine learning tasks (Ghorbani et al, 2016). However whenever there is no clear border between the classes and/or there is some noise in data, SVM might be not very accurate. Also SVM does not handle well the class imbalance which negatively affects its accuracy when dealing with imbalanced datasets.

1.3. Multilayer Perceptron (MLP)

MLP is one of the most widely used artificial neural network models which can be applied to both classification and regression tasks (Dash and Behera, 2016). It consists of set of input, hidden and output layers. MLP is a supervised learning algorithm which trained through the application of back-propagation technique where weight of each input is changed to minimize the difference between the predicted values and target values (Bae, Ahn and Lee, 2019). A big advantage of MLP is that it is relatively easy to implement to tasks of different levels of complexity. It is accepted that the difference between predicted and real values goes down (up to a certain point) with the increase of complexity of hidden layers. However it often comes at the cost of higher computational power requirement. Another difficulty associated with using MLP is that the algorithm might stack in a shallow local minimum and never achieve the global minima. One way to help the algorithm to reach the global minimum is to use Gradient Descent with Momentum Backpropagation training function.

2. Dataset and exploratory data analysis

This study uses a well-known wine dataset from UCI Machine Learning Repository. The version of the dataset is slightly different to what is normally used in the studies as it includes both red and white wine types as well as their chemical properties. There are 6497 entries, 12 physiochemical wine quality predictors such as chlorides or pH and 1 target value – the quality of a wine. There is only 1 categorical attribute and the rest are continues values which would need to be normalized since there are some really big numbers.

There are 7 quality classes of wine (from 3 to 9) where 3 is very bed wine and 9 is the top-quality wine. Since the purpose of the analysis is to investigate if the chosen machine learning algorithms can classify a very good wine from all other wines, the quality classes would need to be rearranged in to 2 groups – very good quality wine (quality classes 7, 8 and 9) and average wine (all other classes). So, it would become a binary classification problem.

Figure 1 shows the size of the wine classes after regrouping has been done. There is a big class imbalance with majority of average wine (5192 entries) in comparison to very good wine (1271 entries). As mentioned previously some classification algorithms are not always coping well with data imbalance which might result in overfitting towards majority class. To avoid this, the classes will be rebalanced using Synthetic Minority Oversampling Technique (SMOTE). This technique involves an over-sampling of minority class by creating the synthetic data points, which improves the class recognition (Chawla, 2002). It would be interesting to see how the application of SMOTE for rebalancing the data affects the accuracy of the models.

Figure 1: Size of the quality classes.

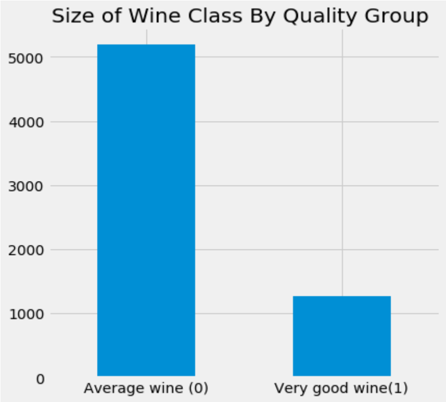
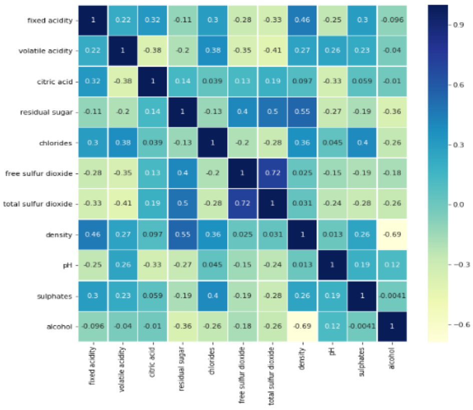


Figure 2: Pearson correlation between the predictors



The initial data analysis has shown that there are 38 missing values which is just over 0.5 % of total entries so rows with missing values can be removed with no major effect on data. Correlation matrix in figure 2 shows the Pearson correlation between the physiochemical characteristics of wine (predictors). A strong correlation between ‘free sulfur dioxide’ and ‘total sulfur dioxide’ is observed, which makes sense as total sulfur dioxide contains free sulfur dioxide. To avoid collinearity free sulfur dioxide will be removed from our study.

3. Methodology, hypothesis statement and networks architecture.

3.1. Method

The data preprocessing will be done in python. This will include removal of missing values, descriptive statistics, class regrouping into binary classification, checking for class imbalance and predictors multicollinearity. The data will be split in to training (70 %) and testing (30%) sets using ‘holdout’ partition. Borderline SMOTE will be applied to training set to resolve the issue with class imbalance. As part of the study this paper will investigate the performance of the models with and without SMOTE applied to data.

The ‘holdout’ approach will be applied to training data to reserve some data (20%) for validation purpose. The training/validation set will be used in the process of model selection to train the models and choose the best hyperparameters. Cross-validation approach will not be applied due to the large size of the dataset. As part of the grid search the best hyperparameters will be chosen according to the highest validation accuracy. After that the best models for each algorithm will be trained again using training set and tested on the unseen data (test set) to compare the performance of the algorithms. The models’ performance will be evaluated by implementing confusion matrix with specific attention to accuracy, F1 Score and area under the ROC curves (AUC).

3.2. Hypothesis statement

It is expected that both algorithms will do well on predicting quality of wine since the research has shown that MLP and SVM are generally very effective in case of binary tasks. MLP should do relatively well on imbalanced data because of its ability to deal with uneven class distribution. It is expected for SVM to perform better on the rebalanced data than on imbalance due to its sensitivity to class imbalance.

3.3. MLP: choice of parameters and hyperparameters

As previously mentioned MLP tends to stack at the local minimum so to avoid this issue 'Gradient descent with momentum backpropagation' network training function has been applied. It updates the weights with respect to gradient descent with momentum, which 'allows a network to respond not only to the local gradient, but also to recent trends in the error surface' (Mathworks). It is known that the initial weights of the network are chosen randomly so it is expected to see a slight variation in final results every time when the network is trained.

The variation in the following hyperparameters has been chosen in order to improve the network performance: hidden layer size, learning rate, momentum. Also maximum numbers of epochs to train were included in the grid search to see how it would influence the network performance. Cross-entropy is used as a cost function to calculate the network performance. However, mean squared error will also be used as an alternative performance function to see if it has any influence on the accuracy of the results. Hyperbolic tangent sigmoid transfer function was chosen for hidden layer and Log-sigmoid transfer function was used for output layer which is optimal in case of single output neuron.

3.4. SVM: choice of hyperparameters

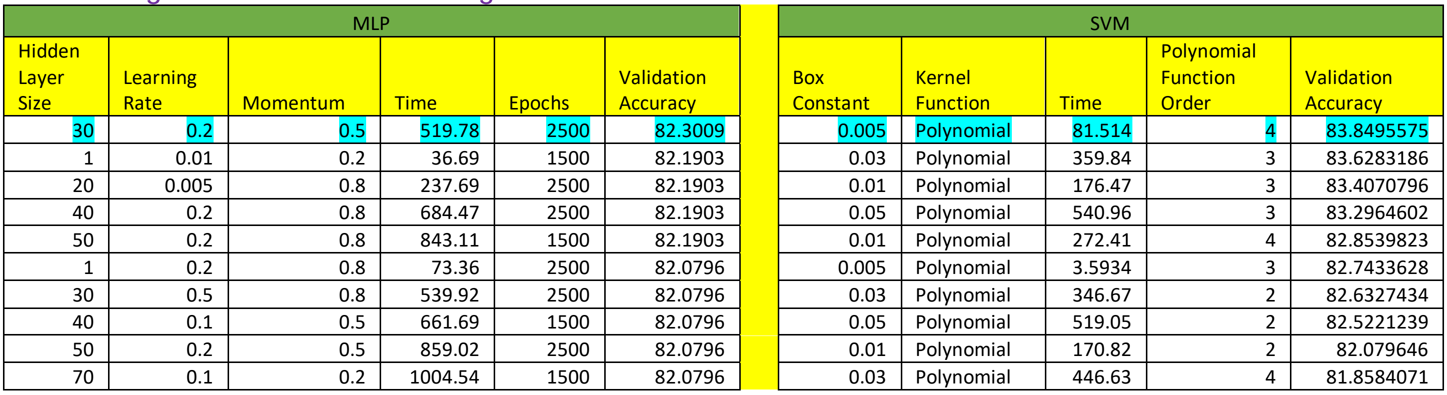
SVM does not require such an extensive tuning of input and output parameters as needed for MLP. However the choice of right hyperparameters is very important for a proper functioning of SVM algorithm as not all hyperparameters work with each other. Box constraint, kernel function and polynomial kernel function order are used as hyperparameters. Box constraint helps to prevent overfitting through the control of 'maximum penalty imposed on margin-violating observations' (MathWorks). The number of support vectors assigned by SVM classifier is tuned by changing the box constraint. Kernel function is a hyperparameter that responsible for processing the inputs in to the appropriate format. The introduction of polynomial kernel function order hyperparameter can only be done if Kernel function is set to polynomial otherwise there will be an error in the algorithm performance.

4. Experimental results and evaluation

4.1. Grid search and model selection

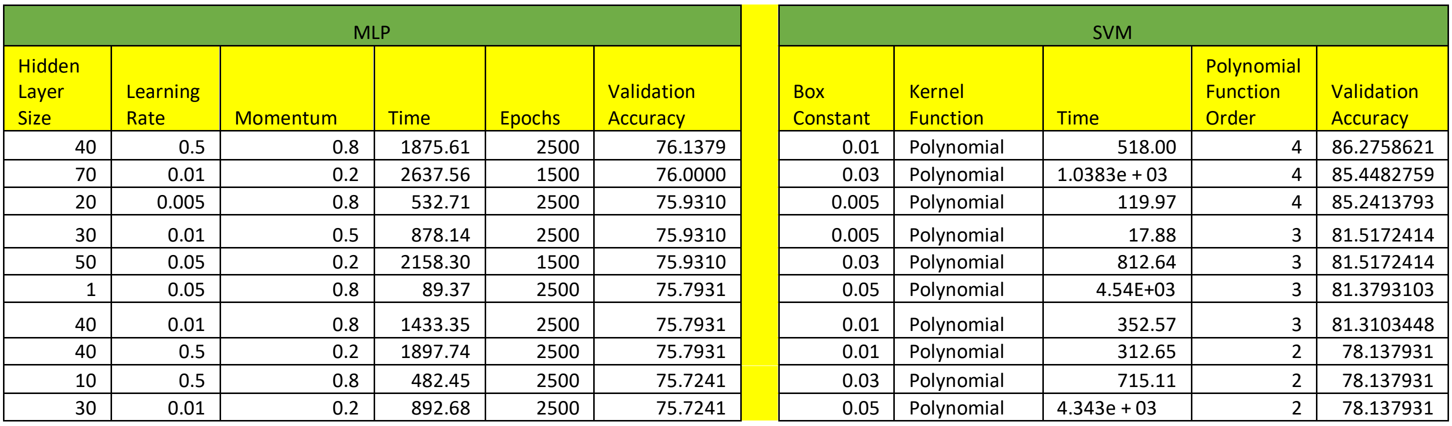
As part of the experiment the grid search has been done with original class imbalance data as well as with SMOTE rebalanced data. Figures 3 and 4 demonstrate the top results sorted by the validation accuracy for each case. For SVM the highest accuracy results were achieved when kernel function was polynomial but only in a right combination with other hyperparameters, otherwise the accuracy could go below 55 %. Also the accuracy was much better when the classes were balanced which goes in line with the hypothesis. This phenomenon is well explained by Batuwita and Palade (2012) who argue that training SVM models on an imbalanced data might lead to skewness towards minority class and as the result poor performance on minorities. Also the scientists suggest that the presence of class imbalance leads to the imbalance in ratio between positive and negative support vectors which might result in higher chance for points at the border to be classified as negative.

Figure 3: Grid search on original imbalance data.



MLP							SVM				
Hidden Layer Size	Learning Rate	Momentum	Time	Epochs	Validation Accuracy		Box Constant	Kernel Function	Time	Polynomial Function Order	Validation Accuracy
30	0.2	0.5	519.78	2500	82.3009		0.005	Polynomial	81.514	4	83.8495575
1	0.01	0.2	36.69	1500	82.1903		0.03	Polynomial	359.84	3	83.6283186
20	0.005	0.8	237.69	2500	82.1903		0.01	Polynomial	176.47	3	83.4070796
40	0.2	0.8	684.47	2500	82.1903		0.05	Polynomial	540.96	3	83.2964602
50	0.2	0.8	843.11	1500	82.1903		0.01	Polynomial	272.41	4	82.8539823
1	0.2	0.8	73.36	2500	82.0796		0.005	Polynomial	3.5934	3	82.7433628
30	0.5	0.8	539.92	2500	82.0796		0.03	Polynomial	346.67	2	82.6327434
40	0.1	0.5	661.69	1500	82.0796		0.05	Polynomial	519.05	2	82.5221239
50	0.2	0.5	859.02	2500	82.0796		0.01	Polynomial	170.82	2	82.079646
70	0.1	0.2	1004.54	1500	82.0796		0.03	Polynomial	446.63	4	81.8584071

Figure 4: Grid search on SMOTE rebalance data.



MLP							SVM				
Hidden Layer Size	Learning Rate	Momentum	Time	Epochs	Validation Accuracy		Box Constant	Kernel Function	Time	Polynomial Function Order	Validation Accuracy
40	0.5	0.8	1875.61	2500	76.1379		0.01	Polynomial	518.00	4	86.2758621
70	0.01	0.2	2637.56	1500	76.0000		0.03	Polynomial	1.0383e + 03	4	85.4482759
20	0.005	0.8	532.71	2500	75.9310		0.005	Polynomial	119.97	4	85.2413793
30	0.01	0.5	878.14	2500	75.9310		0.005	Polynomial	17.88	3	81.5172414
50	0.05	0.2	2158.30	1500	75.9310		0.03	Polynomial	812.64	3	81.5172414
1	0.05	0.8	89.37	2500	75.7931		0.05	Polynomial	4.54E+03	3	81.3793103
40	0.01	0.8	1433.35	2500	75.7931		0.01	Polynomial	352.57	3	81.3103448
40	0.5	0.2	1897.74	2500	75.7931		0.01	Polynomial	312.65	2	78.137931
10	0.5	0.8	482.45	2500	75.7241		0.03	Polynomial	715.11	2	78.137931
30	0.01	0.2	892.68	2500	75.7241		0.05	Polynomial	4.343e + 03	2	78.137931

Interestingly for MLP the best validation accuracy was demonstrated by the models trained on original imbalanced data. A possible explanation for this might be the presence of both white and red types of wine in the dataset which according to Cortez et al (2009) might influence the performance of the models. They argue that red and white wines have completely different taste and different chemical properties and therefore it is better to separate them from each other before applying machine learning algorithms. Since in our study both types of wine were kept together, application of SMOTE could possibly increase a noise in the data which had some negative effect on the prediction accuracy. A relatively good performance of MLP on imbalanced dataset scientists explain by some specific features of MLP network architecture and particularly by its error function that can give priority to strengthening minority class weight-updating over weight-updating of majority class (Oh, 2011).

In the process of grid search for MLP there were two cost functions tested: cross-entropy and mean squared error. It was found that generally the validation accuracy was higher when cross-entropy cost function was used. It can be explained by the fact that cross-entropy is more suitable for classification tasks and mean squared error is known to be effective with regression. Our findings are also supported by Golik, Doetsch and Ney (2014) who argue that cross-entropy is better in finding local optimum and leads to faster convergence whereas squared error tends to stack in a ‘bad’ local optimum. In terms of algorithm training time for SVM there is a clear link between increasing the box constrain and longer running time. Increase in box constrain decreases the number of support vectors and increases training time. For MLP we can observe that the training time increases with the increase in hidden neurons. As discussed earlier, increase in hidden neurons makes the network more complex which leads to a longer training time and higher computational power requirement.

4.2. Comparison and evaluation of the best models

In the testing process it has been decided to keep the original imbalance dataset to train the best MLP and SVM models using hyperparameters highlighted in figure 3. The performance of the models on test data is demonstrated by confusion matrices in figure 5, which is a very useful way to display the predicted values with true values. Confusion matrices have been also used to calculate the performance measures for our models which are summarized in figure 6. Precision and recall are very important measures in some cases. For example, in heart disease prediction classifying a person with heart disease as healthy (false negative) might have dramatic consequences. In wine quality detection, missing a very good quality wine or classifying a bad wine as top quality is not as crucial and therefore this paper concentrates on evaluation of accuracy and F1 score results. In addition ROC curves have been plotted (figure 5) to visualize the comparison of the models in terms of their ability to discriminate between the classes.

Figure 5: Confusion matrices and test ROC cures for MLP and SVM.

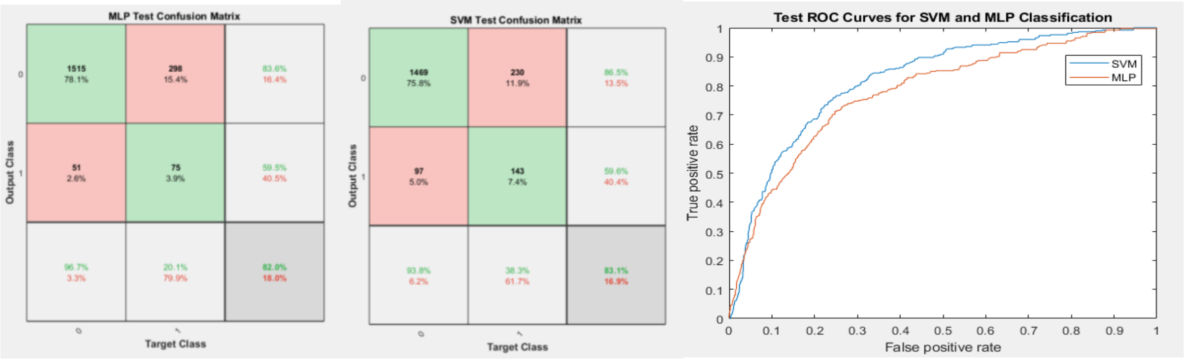


Figure 6: Performance measures

MLP		SVM
82.0	Accuracy	83.15
83.6	Precision	93.8
96.7	Recall	86.5
0.89	F1 Score	0.90
0.78	AUC	0.82

As expected the models did well on predicting quality of wine as both of them showed accuracy higher than 80 %. The accuracy for SVM model is slightly higher than for MLP, which means that the ratio of correctly classified observations to all observations was better for SVM. However according to Chicco and Jurman (2020) in case of class imbalance accuracy is not always a fair measure and might be misleading as it puts majority class in advantage over minority class when it comes to classifier performance. It can be argued that in the situation of uneven class distribution F1 score might be a more reliable measure as it considers both false negative and false positive results and finds the harmonic mean between precision and recall (Chicco and Jurman, 2020). As can be seen in figure 6 the F1 score is almost the same for both algorithms with a minor advantage of SVM. However that tiny difference might be disregarded since due to the random choice of the initial weights of MLP algorithm, every time when you rerun the experiment the results change and sometimes the F1 score was exactly the same for both models. AUC – area under the ROC curves is another useful measure that can help us to understand which model is better in distinguishing between average wine and very good wine classes. As can be seen from the visualization of ROC curves and in figure 6 area under the curve is bigger for SVM than for MLP which means that SVM is better in classifying between average wine class and very good wine class.

5. Conclusion and future work

This paper has been investigating the effectiveness of Multilayer Perceptron and Support Vector Machine in predicting quality of wine (average wine or very good wine) according to the wine physiochemical characteristics. Considering a large dataset and presence of a class imbalance, both algorithms have done well in predicting quality of wine with a minor advantage of SVM. Overall, the findings are in line with what was expected and not contradicting with the results of the similar research papers. It was showed that for good performance of MLP it is crucial to choose right hyperparameters such as hidden layer size or learning rate and appropriate parameters such as training function or cost function. For SVM it was easier to tune the parameters however SVM is more sensitive to class imbalance so rebalancing the data leads to a significant improvement in its performance.

It was surprising to see that MLP accuracy became worse when the class imbalance was corrected by application of SMOTE. One possible reason for this might be that MLP network architecture requires further tuning to cope with changes in the dataset structure. So, in future studies it would be interesting to experiment with different parameters of the network to see how algorithm can cope with SMOTE rebalanced data. For example, we can try to use MLP softmax output transfer function which is known to be effective for classification tasks. Also, it would be interesting to see how the performance of MLP would be affected if we run the experiment for each type of wine (red and white) separately.

6. References

Bae, J., Ahn., J. and Lee, S.J.(2019) 'Comparison of Multilayer Perceptron and Long Short-Term Memory for Plant Parameter Trend Prediction', Nuclear Technology.

Bahrambeugi, B. and Moeinzadeh, H. (2017) 'Comparison of support vector machine and neural network classification method in hyperspectral mapping of orhiolite melanges – A case study of east Iran', The Egyptian jurnal of remote sensing and Space Sciences.

Batuwita, R. and Palade, V. (2012) 'Class imbalance learning methods for support vector machines', Singapore-MIT Alliance for Research and Technology Centre.

Chicco, D. and Jurman, G. (2020) 'The advantages of the Matthews correlation coefficient (MCC) overF1 score and accuracy in binary classification evaluation', BMC Genomics 21:6.

Chawla, N., Bowyer, K., Hall., L. and Kegelmeyer, W.P. (2012) 'SMOTE: Synthetic Minority Over-sampling Technique', Journal of Artificial Intelligence Research 16, 321–357.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T. and Reis, J. (2009) 'Modeling wine preferences by data mining from physicochemical properties', Decision Support Systems 47, 547-533.

Dash, T. and Bahera, H.S. (2018)'A comprehensive study on evolutionary algorithm-based multilayer perceptron for real-world data classification under uncertainty' Willey Expert System

Golik, P. and Ney, H. (2013) ' Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison', Conference Paper.

Grainger, K. (2009) 'Wine quality: tasting and selection', A John Wiley & Sons, Ltd., Publication

Hu, G., Xi, T. and Mohammed, F. (2016) 'Classification of Wine Quality with Imbalanced Data', 978-1-4673-8075-1/16

Mathworks at https://uk.mathworks.com

Oh, S. (2011) 'Error back-propagation algorithm for classification of imbalanced data', Neurocomputing 74, 1058-1061.

Appendix 1 – glossary

TERM	DEFENITION
Perceptron	Perceptron is simple single-layer binary classifier which divides the input space with a linear decision boundary.
Gradient descent with momentum backpropagation	is a network training function that updates weight and bias values according to gradient descent with momentum.
Hyperbolic tangent sigmoid transfer function	Transfer functions calculate a layer's output from its net input. A = tansig (N,FP) takes N and optional function parameters and returns A, the S-by-Q matrix of N's elements squashed into [-1 1].
Log-sigmoid transfer function	Transfer functions calculate a layer's output from its net input. A = logsig (N,FP) takes N and optional function parameters, and returns A, the S-by-Q matrix of N's elements squashed into [0, 1].
Soft max transfer function	Transfer functions calculate a layer's output from its net input. A = softmax (N,FP) takes N and optional function parameters, and returns A, the S-by-Q matrix of the softmax competitive function applied to each column of N.
Local minimum	A local minimum of a function is a point where the function value is smaller than at nearby points, but possibly greater than at a distant point.
Global minimum	A global minimum is a point where the function value is smaller than at all other feasible points.
Normalized Data	Rescaling data to have values between 0 and 1
Synthetic Minority Oversampling Technique (SMOTE)	This technique involves an over-sampling of minority class by creating the synthetic data points, which improves the class recognition
Pearson correlation	A number between -1 and 1 that indicates the extent to which two variables are linearly related
Collinearity	Correlation between predictor variables
Holdout Partition	Random nonstratified partition for holdout validation on n observations. This partition divides the observations into a training set and a test (or *holdout*) set.
Accuracy	Number of correct predictions over total number of predictions
Precision	True Positive / (True Positive + False Positive). Fraction of relevant instances among the retrieved instances
Recall	True Positive / (True Positive + False Negative). Fraction of the total amount of relevant instances that were retrieved
F1 Score	Harmonic mean between precision and recall
Receiver Operating Characteristic (ROC)	It is a plot of the true positive rate against the false positive rate
Cross-entropy performance function	Calculates a network performance given targets and outputs, with optional performance weights and other parameters. The function returns a result that heavily penalizes outputs that are extremely inaccurate (y near 1-t), with very little penalty for fairly correct classifications (y near t).
Mean squared error performance function	It measures the network's performance according to the mean of squared errors.
Learning Rate	Parameter that scales the magnitude of the weight updates in order to minimize the network's loss function.
Box constraint	Helps to prevent overfitting through the control of maximum penalty imposed on margin-violating observations
Kernel function	Hyperparameter that responsible for processing the inputs in to the appropriate format
Momentum	Parameter that helps to avoid the algorithm getting stuck in a local minimum

Appendix 2 – Implementation details

The submission would contain the following files:

- Report – PDF File
- BEST_MODELS_FINAL.m – Matlab Code: contains best SVM and MLP models trained with the best hyperparameters on original (no SMOTE) data and tested on test data.

- MLP_NoSmote.m - Matlab Code: Grid Search for MLP on original data
- MLP_SMOTE.m - Matlab Code: Grid Search for MLP on SMOTE rebalanced data.
- SVM_NoSmote.m - Matlab Code Grid Search for SVM on original data.
- SVM_Smote.m - Matlab Code: Grid Search for SVM on SMOTE rebalanced data.
- Data_Preprocessing .ipynb - ipynb File: Data preprocessing which include removal of missing values, descriptive statistics, class regrouping into binary classification, checking for class imbalance and predictors multicollinearity. The data is split in to training (70 %) and testing (30%) sets using 'holdout' method. Borderline SMOTE is applied to training set to resolve the issue with class imbalance .

- wine_train.csv – csv file: train data with no SMOTE applied
- wine_train_smote.csv – csv file: train data with SMOTE applied
- wine_test – csv file: test data to be used for testing best models