A comparative study in Diabetes Diagnosis through Multilayer Perceptron and Support Vector Machine

Bona Chow

Abstract

Multilayer Perceptron (MLP) and Support Vector Machine (SVM) algorithms yielded similar test AUCs and accuracies on the Pima Indian Diabetes dataset. MLP's higher specificity of 89% compared to SVM (78%) makes MLP more suitable for practical use, as there is a high cost associated with the incorrect classification of a large number of non-diabetes patients.

1. Introduction

The global prevalence of diabetes among adults over 18 years old increased from 4.7% in 1980 to 8.5% in 2014 according to World Health Organization[(1)]. Early diagnosis and treatment are essential to avoid complications such as blindness, kidney failure, heart attacks, stroke and lower limb amputation. Machine learning algorithms may aid the diagnosis of diabetes.

The purpose of this study is to compare, contrast and critically evaluate two algorithms, namely Multilayer Perceptron (MLP) and Support Vector Machine (SVM), applied in the diagnosis of diabetes based on the Pima Indian Diabetes database.

1.1 Multilayer Perceptron (MLP)

MLP is a class of feed-forward artificial neural network consisting of fully connected neurons in an input layer, one or more hidden layers and an output layer[(2)]. During training, data are repeatedly passed into MLP and each neuron passes to the next layer a signal that is a function of sum of the inputs. An error signal which is calculated by the difference between the actual and the predicted output is back propagated through the network such that the weight of each neuron can be adjusted to minimize the error (loss).

Advantages of MLP are that it works on non-linearly separable data and does not require prior knowledge or assumption on the data distribution. A disadvantage of MLP is that it is prone to overfitting. Nonetheless, this may be reduced by the application of dropout, batch normalization and early stopping[(3)(4)]. Another disadvantage is that it is technically challenging, computationally expensive and time-consuming to build and train MLP models. Variations in architecture, activation function, learning rate, dropout rate, batch size and other parameters can produce significant difference in predictive power. Furthermore, MLP may yield a solution that is a local minimum instead of the global minimum[(5)], depending on the choice of initial weights, learning rate and momentum.

1.2 Support Vector Machine Classifier (SVM)

In the context of binary classification, SVM is an algorithm that separates two classes by maximizing the margin between data points and the optimal decision boundary in a feature space[(6)]. A kernel can be applied to transform input data into a higher dimensional feature space to enable computation of a decision boundary.

SVM has the advantage of yielding global and unique solution. In addition, the computation time and cost are relatively low, as only the examples closest to the decision boundary (support vectors) determine the optimal decision boundary. A disadvantage of SVM is that it assumes the data is linearly separable, which is not always appropriate. Furthermore, the selection of kernel requires either prior knowledge on the distribution of the data or trial-and-error. Underfitting occurs when the decision boundary does not effectively separate the data points into two classes in the selected transformed space. In addition, any noise near the decision boundary would have a significant impact to the model. Nonetheless, regularization can be incorporated into SVM by separating the two classes with a large margin, enhancing its immunity to noise.

1.3 Hypothesis

The first hypothesis is that the MLP yields a higher Area Under the Receiver Operation Curve (AUC) score than SVM. SVM assumes that the data is linearly separable, but this is not the case for our dataset. MLP does not have such an assumption. This is supported by the literature – MLP achieved an accuracy of 0.82[7], whereas SVM attained a lower accuracy of 0.78[8]. The second hypothesis is that randomized search for MLP requires a longer time than for SVM. SVM only considers a small number of data points closest to the decision boundary, whereas training MLP involves numerous iterations of complex computation.

2. Dataset

The Pima Indian Diabetes database was obtained from Kaggle[9] (originally from UCI). The dataset originally contained 8 numeric (continuous) features of 768 Pima Indian females aged 21 or above. A total of 268 females were diagnosed with diabetes and 500 females were not. Six features, namely 'Pregnancies' (number of previous pregnancies), 'Glucose' (2-hour plasma glucose concentration in an oral glucose tolerance test), 'BloodPressure' (diastolic blood pressure), 'BMI' (body mass index), 'DiabetesPedigreeFunction' (diabetes pedigree function) and 'Age' (age in years) were used for analysis. Missing values were encoded as zero and distributed randomly in the columns 'SkinThickness' (tricep skinfold thickness), 'Insulin' (2-hour serum insulin), 'Glucose', 'BloodPressure' and 'BMI'. Removing the columns 'SkinThickness' and 'Insulin' and imputing the remainder with mean yielded highest accuracy in our experiment and were implemented (see Appendix).

2.1 Data Exploration

The scatterplot matrix revealed the data were not linearly separable with significant overlapping between the two target classes across all pairs of features (Figure 1). The distribution of the two classes appeared most different across 'Glucose', where the mean for non-diabetes females was significantly lower than for diabetes patients (Figure 2). Correlation analysis revealed 'Glucose' showed moderate Pearson correlation of 0.49 with the target 'Outcome'. Normalized bar plots of each feature with 'Outcome' overlay revealed the increasing likelihood of diabetes with 'Pregnancies', 'Glucose', 'BloodPressure','BMI' and 'DiabetesPedigreeFunction' (Figure 3). Multicollinearity was not detected.

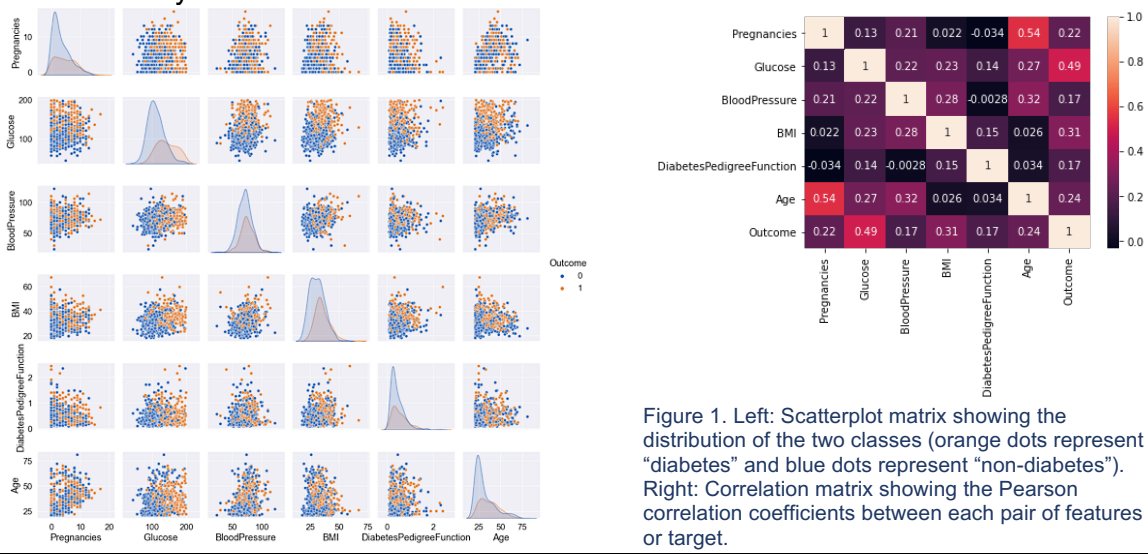

Figure 1. Left: Scatterplot matrix showing the distribution of the two classes (orange dots represent "diabetes" and blue dots represent "non-diabetes"). Right: Correlation matrix showing the Pearson correlation coefficients between each pair of features or target.

Feature	Mean for diabetes patients	Mean for non-diabetes patients	Standard deviation for diabetes patients	Standard deviation for non-diabetes patients
Pregnancies	4.866	3.298	3.741	3.017
Glucose	142.166	110.710	29.542	24.717
BloodPressure	75.147	70.935	11.946	11.935
BMI	35.385	30.888	6.595	6.505
DiabetesPedigreeFunction	0.551	0.430	0.372	0.299
Age	37.067	31.190	10.968	11.668

Figure 2. Mean and standard deviation of each feature for diabetes and non-diabetes patients.

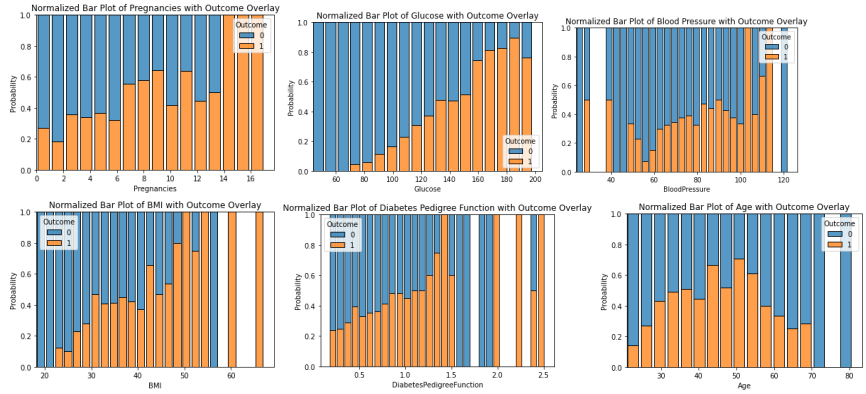

Figure 3. Normalized bar plots of each feature with 'Outcome' overlay.

3. Methodology

The following training and evaluation methodology were adopted.

1. Divided data into training and test sets in 80:20 ratio using stratified sampling such that equal proportions of diabetes patients and non-diabetes patients were in both sets.
2. Applied Synthetic Minority Over-sampling Technique (SMOTE) on the minority non-diabetes class in the training set such that both classes became equal in size and the bias towards the majority class was eliminated during training[10].
3. Standardized the features in both training and test sets to zero mean and unit variance.
4. Selected the best model for each algorithm based on AUC using randomized search on the hyperparameters on the training data. Stratified five-fold cross-validation was used to reduce bias and increase generalizability of the models. Using AUC is more appropriate than accuracy, because AUC is less sensitive to differences in target class proportions in the training and test sets. For MLP, Early Stopping was applied to reduce overfitting such that the training stopped when there was no improvement in validation accuracy (determined using a random selection of 20% of the training data that was unseen during training) in the last 5 epochs or when a maximum of 100 epochs were completed.
5. Measured the time required to perform randomized search for each algorithm across five scoring functions (accuracy, precision, recall, F1 score and AUC) on a Macbook Pro equipped with a 2.3GHz Intel Core i5 processor with Python's Time module.
6. Evaluated and compared the algorithms based on AUC on the test data and the time required for the randomized search.

3.1 Architecture and Parameters for MLP

The MLP contained 6 input neurons, one or two hidden layers and two output neurons. Rectified linear unit (RELu) activation function was applied in the input and hidden layers. Softmax was used in the output layer to give predicted probabilities for each class. Stochastic gradient decent algorithm was used to update the weights of neurons. The optimal combination of the number of neurons, optimizer momentum, learning rate, dropout rate for input and hidden layers, and batch size in MLP with one or two hidden layers were selected using stratified 5-fold cross validated Randomized Search based on AUC score.

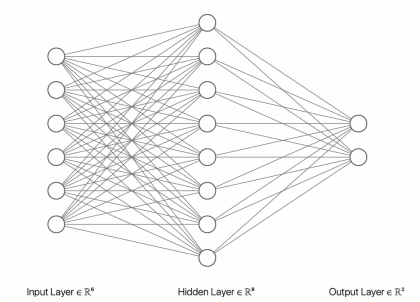

Figure 4: The architecture of the best MLP model.

The best model used 9 neurons in one hidden layer (Figure 4), optimizer momentum of 0.3, learning rate of 1, dropout rate of 0.2 and batch size of 64 (Figure 5).
The computation time required for the Randomized Search across five performance metrics (accuracy, precision, recall, F1 score and AUC) was shorter for MLP with 1 hidden layer (332.91 seconds) than for MLP with 2 hidden layers (427.95 seconds), as the search for MLP with 2 hidden

layers included an additional parameter that specifies the number of neurons in the second hidden layer and the number of possible combinations of number of neurons increased from 6 to 16.

3.2 Architecture and Parameters for SVM

The optimal kernel, regularization parameter C, and for polynomial and RBF kernels, the kernel coefficient gamma, were determined using stratified 5-fold cross validated Randomized Search based on AUC score.

RBF kernel, C of 100 and gamma of 0.02 (Figure 5) were used to build the optimal SVM model. The number of support vectors for the two classes were 195 and 203. The computation time for the Randomized Search across five performance metrics (accuracy, precision, recall, F1 score and AUC) only required 4.98 seconds.

Kernel	C	Gamma	Validation AUC
Linear	0.1	-	0.834 ± 0.055
Third degree polynomial	0.1	0.03	0.814 ± 0.052
Third degree polynomial	1	0.03	0.818 ± 0.046
Third degree polynomial	10	0.02	0.828 ± 0.041
Third degree polynomial	10	0.03	0.829 ± 0.042
Third degree polynomial	100	0.01	0.828 ± 0.043
Third degree polynomial	100	0.02	0.825 ± 0.053
RBF	1	0.01	0.841 ± 0.052
RBF	1	0.02	0.849 ± 0.054
RBF	100	0.02	0.851 ± 0.044

Number of hidden neurons	Optimizer momentum	Learning rate	Dropout rate	Batch size	Validation AUC
3	0.3	0.1	0.2	64	0.832 ± 0.068
4	0.1	1	0.0	32	0.832 ± 0.062
4	0.3	1	0.2	16	0.816 ± 0.068
5	0.1	0.01	0.5	64	0.819 ± 0.066
5	0.9	0.01	0.0	64	0.842 ± 0.079
6	0.9	0.1	0.5	32	0.836 ± 0.056
7	0.1	0.01	0.2	32	0.841 ± 0.050
8	0.3	0.01	0.2	64	0.830 ± 0.053
8	0.3	1	0.2	64	0.844 ± 0.071
8	0.3	1	0.5	16	0.828 ± 0.056
3, 6	0.1	1	0.2	32	0.805 ± 0.060
3, 9	0.3	0.01	0.0	32	0.833 ± 0.046
3, 9	0.6	0.01	0.5	64	0.817 ± 0.060
6, 3	0.9	1	0.2	64	0.500 ± 0.000
6, 6	0.3	0.01	0.2	16	0.833 ± 0.076
6, 6	0.9	0.01	0.0	16	0.838 ± 0.058
6, 12	0.9	0.1	0.0	32	0.790 ± 0.183
9, 6	0.3	0.01	0.5	16	0.835 ± 0.060
12, 3	0.1	0.01	0.0	16	0.840 ± 0.060
12, 3	0.9	0.1	0.2	16	0.500 ± 0.000

Figure 5. Left: Randomized Search results for SVM. Right: Randomized Search results for MLP.

4 Results, Findings and Evaluation

4.1 Model Selection

For SVM, applying RBF kernel, relatively large C of 100 and moderate gamma of 0.02 yielded the highest AUC. The choice of kernel appeared to have strongest impact on AUC, as all RBF kernel SVMs showed higher AUCs than linear or third degree polynomial kernel SVMs. Increasing C slightly increased the AUC for RBF and polynomial SVMs, as this decreased the margin and increased the complexity of the decision boundary, which resulted in higher variance and lower bias[11]. Increasing gamma, which determines how far the influence of each training example, showed mixed impact on AUCs for third degree polynomial and RBF SVMs.

For MLP, the combination of 8 neurons in a single hidden layer, relatively large batch size of 64, high learning rate of 1, small dropout rate of 0.2 and small momentum of 0.3 yielded the highest AUC. The architecture appeared to show the highest impact on AUC. Increasing the number of hidden neurons from 3 to 5 appeared to correlate with increase in AUC. As the model complexity increased, the variance increased and the bias reduced. Nonetheless, adding more hidden neurons or a second hidden layer resulted in little or no increase in AUCs, as the models started to become too complex and overfitted. Further investigation is needed to determine the influence of each of the parameters optimizer momentum, learning rate, dropout rate and batch size on AUC. Two models with two hidden layers and optimizer momentum of 0.9 lacked discriminative power between the two target classes (AUC of 0.5), which reflected the steps taken by stochastic gradient descend algorithm were so large that the minimum loss could not be found.

4.2 Algorithm comparison

When constructing the final model, all training data were used for SVM, whereas for MLP, only 80% of the training data were used, as 20% of the data was withheld for validation in Early Stopping. Both final MLP and SVM models showed relatively high AUC scores of 0.878 and 0.889 respectively on training data and slightly lower AUCs on the test data (Figure 6). The stratified 5-fold cross validation AUCs for SVM (0.851) and MLP (0.844) obtained previously (Figure 5) were between the AUCs on training and test data for the final models. MLP achieved a marginally higher AUC of 0.831 than SVM (0.820) on the test data, whereas SVM showed slightly better performance (0.740) than MLP (0.734). These accuracies were lower than the values of 0.78 and 0.82 reported previously for SVM[8] and MLP[7] respectively.

	AUC on training data	AUC on test data	Accuracy on test data	Computation time in seconds
SVM	0.889	0.820	0.740	4.98
MLP	0.878	0.831	0.734	332.91 for MLP with 1 hidden layer; 427.95 for MLP with 2 hidden layers
Dummy classifier	0.533	0.434	0.442	-

Figure 6. Table showing AUC score on training and test data, accuracy on test data and training time for SVM and MLP models.

The higher AUC scores on training data than on the test data for both algorithms indicated both MLP and SVM were slightly overfitted with relatively high variance and low bias. When compared to MLP, SVM showed more prominent overfitting, higher variance and lower bias. The generalizability of these models can be increased by reducing the model complexity or by providing more data for training. Model complexity can be reduced by using a simpler kernel or a smaller regularization parameter C for SVM, and using fewer hidden neurons, increasing the dropout rate and increasing the batch size for MLP[12].

SVM required significantly less computation time for Randomized Search. Two-layer and three-layer MLP required 66.8 times and 85.9 times longer computation time than SVM respectively. As we expected, the repeated, iterative search for minimal loss in MLP required significantly longer time than SVM's approach in maximizing the margin of the decision boundary.

The ROC and confusion matrices revealed SVM displayed a slightly higher sensitivity (also known as true positive rate or recall) of 72% than MLP's 65% (Figure 7). This indicated SVM exhibited slightly better capability in making positive diagnosis for diabetes, particularly for this case where incorrectly classifying diabetes patients as non-diabetic may cause serious complications. Nonetheless, the specificity (=1- true negative rate) for MLP 89% was significantly higher than SVM's 78%. Considering diabetes occurs in a relatively small proportion of population (8.5% in 2014[1]), in countries where medical resources are inadequate, MLP would be more suitable than SVM, as there is a high cost associated with incorrect classification of a large number of non-diabetes patients.

The precision-recall curves revealed both SVM and MLP attained precision of 60% and 58%, indicating they were capable of making correct positive diagnosis out of all positive predictions made respectively, at the imbalanced diabetes to non-diabetes class ratio of 268:500[13].

The use of SMOTE introduced bias towards the minority diabetic class during training. This may have a particularly significant impact on SVM when the synthesized data were close to the decision boundary. Such impact can be lessened by combining SMOTE with some undersampling of the majority non-diabetic class instead of solely applying SMOTE on the minority class, so that fewer synthetic data would be required[10]. The bias would have less influence on MLP as the training data was passed into the neural network in batches and for multiple epochs.

5. Conclusion

Both MLP and SVM exhibited high capability in diagnosing diabetes. The differences between the performance of the two algorithms were subtle, as reflected by the similarity in AUCs and accuracies on test or training data. MLP's higher specificity compared to SVM makes MLP more

suitable for real-life situation, where diabetes only occurs in a small proportion of population and there is a high cost associated with the incorrect classification of a large number of non-diabetes patients. There is, however, small trade-offs in sensitivity. Furthermore, significantly longer computation time and larger efforts in terms of architecture and hyperparameter optimization would be required for training MLP when compared to SVM. Further investigation should be made into developing ensemble models such as bagging, which is expected to have lower variance and higher generalizability[14].

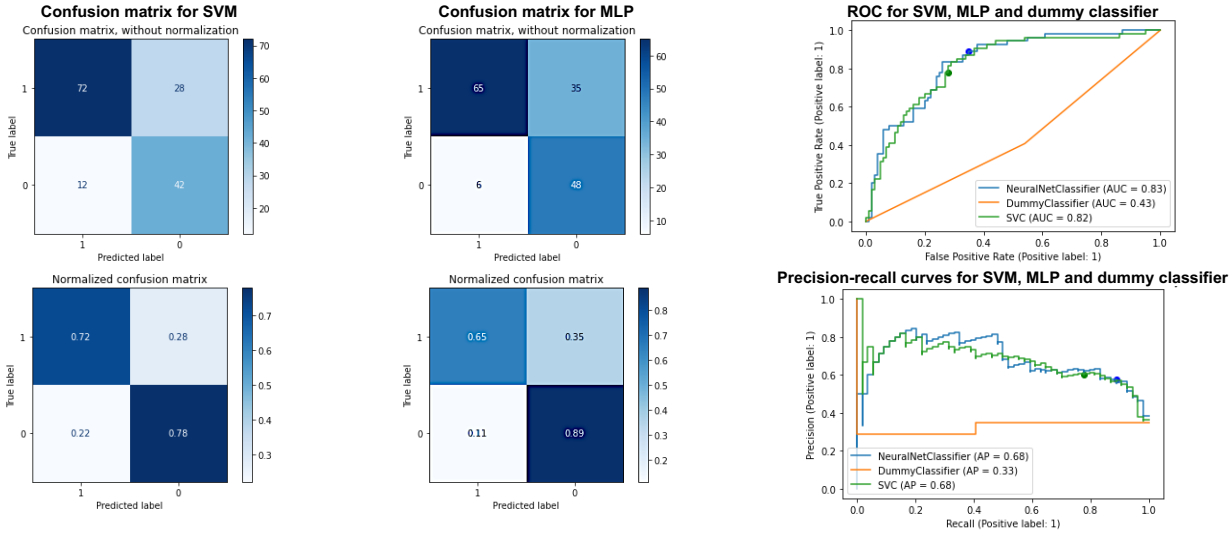

Figure 7. Left: Confusion matrix for the final SVM model. Middle: Confusion matrix for the final MLP model. Right Top: ROC for final SVM, final MLP and a dummy classifier that makes stratified random class predictions. Optimal false positive rate and true positive rate for SVM (28.0%, 77.8%) and MLP (35.0%, 88.9%) are indicated. Right Bottom: Precision-recall curves for final SVM, final MLP and a dummy classifier that makes stratified random class predictions. Optimal recall and precision for SVM (77.8%, 60.0%) and MLP (88.9%, 57.8%) are indicated.

References

1. Who.int. 2021. *Diabetes*. [online] Available at: <https://www.who.int/news-room/fact-sheets/detail/diabetes/> [Accessed 12 April 2021].
2. Gardner, M. and Dorling, S., 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14-15), pp.2627-2636.
3. Garbin, C., Zhu, X. and Marques, O., 2020. Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79(19-20), pp.12777-12815.
4. Prechelt, L., 1998. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4), pp.761-767.
5. Fukumizu, K. and Amari, S., 2000. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 13(3), pp.317-327.
6. CS229 Lecture notes by Andrew Ng. 2021. [online] Available at: <https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf> [Accessed 12 April 2021].
7. Ebenezer Obaloluwa Olaniyi, K., 2021. *Onset Diabetes Diagnosis Using Artificial Neural Network*. [online] Ijser.org. Available at: <https://www.ijser.org/paper/Onset-Diabetes-Diagnosis-Using-Artificial-Neural-Network.html> [Accessed 12 April 2021].
8. van Gestel, T., Suykens, J., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., de Moor, B. and Vandewalle, J., 2004. Benchmarking Least Squares Support Vector Machine Classifiers. *Machine Learning*, 54(1), pp.5-32.
9. Kaggle.com. 2021. *Pima Indians Diabetes Database*. [online] Available at: <https://www.kaggle.com/uciml/pima-indians-diabetes-database> [Accessed 12 April 2021].
10. Chawla, N., Bowyer, K., Hall, L. and Kegelmeyer, W., 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, pp.321-357.
11. Valentini, G. and Dietterich, T., 2002. Bias—Variance Analysis and Ensembles of SVM. *Multiple Classifier Systems*, pp.222-231.
12. Garbin, C., Zhu, X. and Marques, O., 2020. Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79(19-20), pp.12777-12815.
13. Saito, T. and Rehmsmeier, M., 2015. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3), p.e0118432.
14. Wang, H., Zheng, B., Yoon, S. and Ko, H., 2018. A support vector machine-based ensemble algorithm for breast cancer diagnosis. *European Journal of Operational Research*, 267(2), pp.687-699.

Appendix – Implementation details

1. An experiment was conducted to determine the best method of handling zero values (Table A1). Removing the columns 'BloodPressure' and 'SkinThickness' and imputing zero values in remaining columns with mean yielded highest accuracy.

Method	Mean accuracy	Standard deviation of accuracy
No imputation	0.755241	0.022017
Drop the columns 'BloodPressure' and 'SkinThickness' and mean imputation in other columns	0.764358	0.019465
Mean imputation	0.759189	0.027278
K nearest neighbour imputation	0.752670	0.031348
Multivariate imputation	0.751371	0.027420

Table A1. Mean and standard deviation of accuracies of stratified 5-fold cross-validated SVMs with radial basis function kernel.

2. Stratified 5-fold and 10-fold cross-validation were compared (Table A3). Both showed similar AUCs but stratified 5-fold cross-validation yielded smaller standard deviations than 10-fold. The validation data probably became too small in size when 10 folds were used and a larger standard deviation resulted from the inclusion of any extreme value for validation. Stratified 5-fold cross-validation was used in Randomized Search.

Number of folds	Mean AUC	Standard deviation of AUC
5 folds	0.833656	0.028039
10 folds	0.832812	0.046983

Table A2. Mean and standard deviation of AUC of stratified 5-fold and 10-fold cross-validated SVMs with linear kernel.

3. SVM with different kernels were compared (Table A3). Radial basis function (RBF) kernel SVM yielded highest AUC. Linear and third degree polynomial kernel were the next highest.

Kernel	Mean AUC	Standard deviation of AUC
Linear	0.833594	0.027981
Second degree polynomial	0.636563	0.032193
Third degree polynomial	0.816875	0.023330
Radial basis function	0.859781	0.025351
Sigmoid	0.743875	0.030180

Table A3. Mean and standard deviation of AUC of stratified 5-fold cross-validated SVMs with different kernels.

4. The optimal combination of kernel (RBF, linear or third degree polynomial), kernel parameter C (0.1, 1, 10 or 100), and, for RBF and polynomial kernel, regularization parameter gamma (0.01, 0.02 or 0.03) was determined using stratified 5-fold cross-validated Randomized Search.

5. Three MLP architectures were evaluated (Table A4). MLP with one hidden layer and 9 neurons yielded highest AUC, followed by MLP with two hidden layers each with 6 neurons.

MLP architecture	Mean AUC	Standard deviation of AUC
1 hidden layer with 9 neurons	0.841719	0.036300
2 hidden layers each with 6 neurons	0.835797	0.028925
4 hidden layers each with 4 neurons	0.688109	0.138627

Table A4. Mean and standard deviation of AUC of stratified 5-fold cross-validated MLPs with different architectures.

6. MLPs with one or two hidden layers each containing either 3, 6, 9 or 12 neurons were compared. The AUCs were similar and Randomized Search was subsequently performed for MLPs with one hidden layer and with two hidden layers.

MLP architecture	Mean AUC	Standard deviation of AUC
1 hidden layer with 3 neurons	0.844094	0.037250
1 hidden layer with 6 neurons	0.839922	0.032704
1 hidden layer with 9 neurons	0.846203	0.021688
1 hidden layer with 12 neurons	0.846500	0.033317
2 hidden layers each with 3 neurons	0.844844	0.031171
2 hidden layers each with 6 neurons	0.845062	0.031665
2 hidden layers each with 9 neurons	0.835625	0.027757
2 hidden layers each with 12 neurons	0.845438	0.027029

Table A5. Mean and standard deviation of AUC of stratified 5-fold cross-validated MLPs with 3, 6, 9 or 12 hidden layer neurons.

7. The optimal combination of number of neurons (3 to 7 in a single hidden layer or either 3, 6, 9 or 12 in each of the two hidden layers), optimizer momentum (0.1, 0.3, 0.6 or 0.9), learning rate (0.01, 0.1 or 1), dropout rate (0.0, 0.2 or 0.5) and batch size (16, 32 or 64) was determined using stratified 5-fold cross-validated Randomized Search.