A Comparative Study in diagnosing diabetes through Multilayer Perceptron and Support Vector Machines

Rahem Khan

Rahem.Khan@city.ac.uk

Abstract

This research paper aims to critically evaluate the two best performed supervise learning model on Pima diabetes dataset. The two algorithms being formed and compared against each other are Multilayer Perceptron (MLP) and Support Vector Machines (SVM). Grid search has been performed to determine the best hyperparameters and it's validated via cross validation function. The final test result is then evaluated against confusion matrix, ROC curve and F-1 scores which indicates that SVM is more suitable for binary classification task.

1. Introduction

Diabetes is one of the fastest growing medical condition within UK which can affect up to 5.5 million people across the country by 2030 [1]. Whilst there is risk of greater increase in the future, early detection can play a vital role in treating diabetes.

The main aim of this paper is to compare and critically evaluate two important supervised learning models – Multilayer Perceptron (MLP) and Support Vector Machines (SVM) and determine the diagnosis based on 8 features present in the Pima Indian diabetes dataset [2]. Furthermore, we will explore different configurations and parameters of these models while initiating hyperparameter optimisation via grid search.

Section 2 consists of exploratory data analysis alongside brief description of the dataset used in training and testing for the models. Section 3 draws a comparison of approaches and method used during the implementation stage. Section 4 discusses the results and findings whilst critically evaluate the two presented models. Section 5 concludes the paper.

1.1. Multilayer Perceptron (MLP)

Neural networks, or more precisely artificial neural networks, are a branch of artificial intelligence and it's deemed to be more useful alternatives to traditional statistical modelling in many science disciplines [3]. A Multilayer Perceptron or MLP is a supervised learning algorithm which consist of an input layer connected to a one or more hidden layers and an output layer. The data flows in a forward direction from input layers to output layers. However, it trained via backpropagation algorithm where weights and biases are learned . Weight, bias, and activation function is an integral part of each layer in the MLP except the input layer. The activation function is responsible for mapping the product of the weights alongside their sums with the biases. Training process carries on till we reach the minimum error . The activation and loss function depends on the task we are solving (i.e., classification/regression).

1.2. Support Vector Machines (SVM)

Support Vector Machines is a supervised learning methods used for classification and regression [4]. The main goal of this algorithm is to determine the best hyperplane that distinctly classifies the data points. Kernel-function is to induce such a feature space by mapping the training data into a higher dimensional space where the data is linear separable [5]. A linear kernel is used for linear classification whereas polynomial and gaussian used in non-linear classification problem. SVM provides greater stability and prevents overfitting due to its good generalization capabilities.

2. Dataset

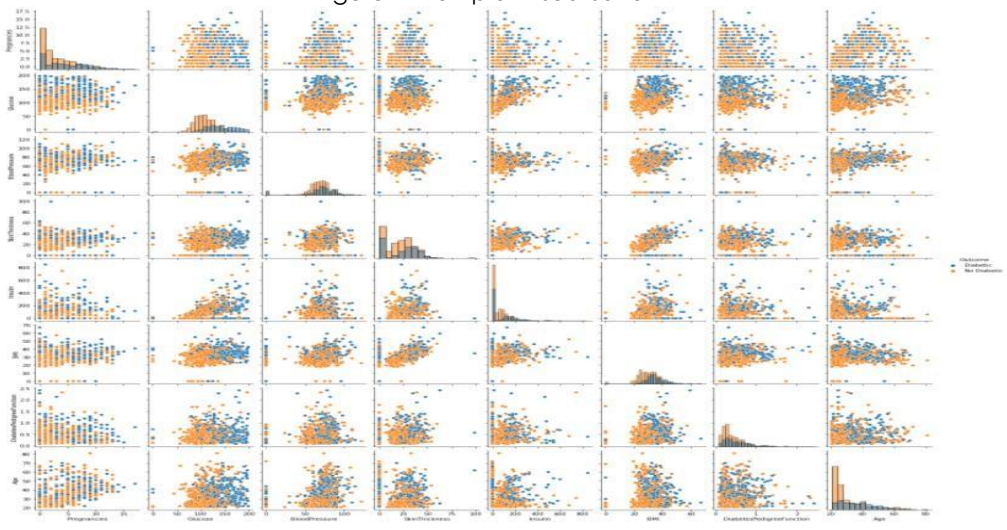
The dataset used in this experimentation is sourced from Kaggle and it is regarding 768 Pima diabetes samples with 8 input features and one target output. Target output has been transformed to binary classification where 1 denotes to diabetic and 0- non-diabetic. This dataset has no missing values and not required any over-sampling/under-sampling technique due to very slight imbalance in the target variable.

Table 1 – Data Summary

Variable	Type
Pregnancies	Continuous
Glucose	Continuous
Blood Pressure	Continuous
Skin Thickness	Continuous
Insulin	Continuous
BMI	Continuous
Diabetes Pedigree Function	Continuous
Age	Continuous
Outcome	Binary (0 or 1)

2.1. Exploratory Data Analysis

Figure 1- Pair-plot visualisation



Pair plot visualisation has been implemented to determine the correlation between key input features and the target variable which is in our case (1-diabetic, 0-non-diabetic). This visualisation can help in uncovering some interesting insights regarding the data which otherwise difficult to interpretate while looking at the raw data.

As expected, individual with glucose level higher than 125 diagnosed with diabetes. It is also noted that high blood pressure is very much correlated with diabetes. BMI features suggest that individuals with high BMI more likely to be diagnosed with the disease.

3. Methods

This section of the report consists of all the training, validation, and testing steps alongside model architecture formation and hyperparameters used in building both models.

3.1. Methodology

We have used (train_test_split) to split our data keeping 80% for training and validation set, 20% for test set and to draw comparison between MLP and SVM models. The test set is unseen by the model.

The model selection process based on grid search where hyperparameters tuning takes place on both MLP and SVM. We have implemented 10-fold cross validation due to its reliability in accessing model performance and reduce bias. The models will be compared on precision, recall, F-score, and AUC metrics alongside visualisation of confusion matrix and ROC curve which will gives us a good indication of the best performing models on the given dataset.

3.2. Architecture and Parameters used for the MLP

Model tuning will take place of number of hidden layers, nodes per hidden layer, momentum and learning rate. The number of hidden layers and the number of neurons in each layer of a deep learning network are two key parameters, which have main influence on the performance of the algorithm [6]. Our MLP network consist of a input layer, hidden layer and output layer. The ReLU activation function was applied on the hidden layer and SoftMax function was applied on the outer layer due to its suitability for binary classification problem. The negative log likelihood loss function has been implemented as a loss function.

3.3. Architecture and Parameters used for the SVM

In this section we will set out the kernel function- Linear, Polynomial, RBF and Sigmoid as the starting point. The box constraint is a regularisation parameter which plays it’s part in avoiding misclassifying training examples. Grid search has been applied to tune the hyperparameters in order to select the best parameters such as C, degree of polynomial and kernel function for SVM.

4. Results, Findings & Evaluation

4.1.Model Selection

Grid search was performed on SVM with the range of C, kernel, and degree parameter. The process took longer than I expected. However, it did manage to configure the optimum hyperparameter for this analysis. Our grid search returned a C value of 0.1, degree of 2 and a polynomial kernel with an accuracy score of 0.76. Whereas MLP performed best with a learning rate of 0.1, momentum of 0.9 with 200 hidden layers gives us an accuracy score 0.74.

Table 1 – Grid Search Results

SVM				
	C	**degree**	**kernel**	**score**
0	0.1	2	linear	0.649965
1	0.1	2	rbf	0.729839
2	0.1	2	poly	0.767296
3	0.1	3	linear	0.649965
4	0.1	3	rbf	0.729839
5	0.1	3	poly	0.759748
6	0.1	4	linear	0.649965
7	0.1	4	rbf	0.729839
8	0.1	4	poly	0.742977
9	1	2	linear	0.761635
10	1	2	rbf	0.754228
11	1	2	poly	0.759853
12	1	3	linear	0.761635
13	1	3	rbf	0.754228
14	1	3	poly	0.737491
15	1	4	linear	0.761635
16	1	4	rbf	0.754228
17	1	4	poly	0.757862
18	10	2	linear	0.757862
19	10	2	rbf	0.746646
20	10	2	poly	0.733788
21	10	3	linear	0.757862
22	10	3	rbf	0.746646
23	10	3	poly	0.756045
24	10	4	linear	0.757862
25	10	4	rbf	0.746646
26	10	4	poly	0.7413

MLP				
	lr	**momentum**	**hidden size**	**score**
0	0.05	0.85	50	0.638749
1	0.05	0.9	50	0.638749
2	0.05	0.95	50	0.638749
3	0.05	0.85	100	0.638749
4	0.05	0.9	100	0.640636
5	0.05	0.95	100	0.644444
6	0.05	0.85	200	0.640636
7	0.05	0.9	200	0.655486
8	0.05	0.95	200	0.688924
9	0.1	0.85	50	0.642523
10	0.1	0.9	50	0.636897
11	0.1	0.95	50	0.651852
12	0.1	0.85	100	0.646331
13	0.1	0.9	100	0.698323
14	0.1	0.95	100	0.696541
15	0.1	0.85	200	0.731761
16	0.1	0.9	200	0.744689
17	0.1	0.95	200	0.744654
18	0.2	0.85	50	0.704018
19	0.2	0.9	50	0.705765
20	0.2	0.95	50	0.724458
21	0.2	0.85	100	0.724389
22	0.2	0.9	100	0.694654
23	0.2	0.95	100	0.681447
24	0.2	0.85	200	0.726345
25	0.2	0.9	200	0.665024
26	0.2	0.95	200	0.678057

4.2. Algorithm Comparison

Model evaluation is performed via classification metrics such as Precision, Recall, F1- Score, ROC visualisation and confusion matrix. Both model is suitable for the classification task. Both models tend to predict class 0 more frequently with a recall rate of 0.85 and 0.92 respectively, however, SVM achieved a higher recall rate of 0.51 for predicting class 1 compared to MLP's 0.48, which means that SVM performed better when it comes to predicting the presence of diabetes, which is very crucial in disease diagnosis.

Figure 2- Confusion Matrix

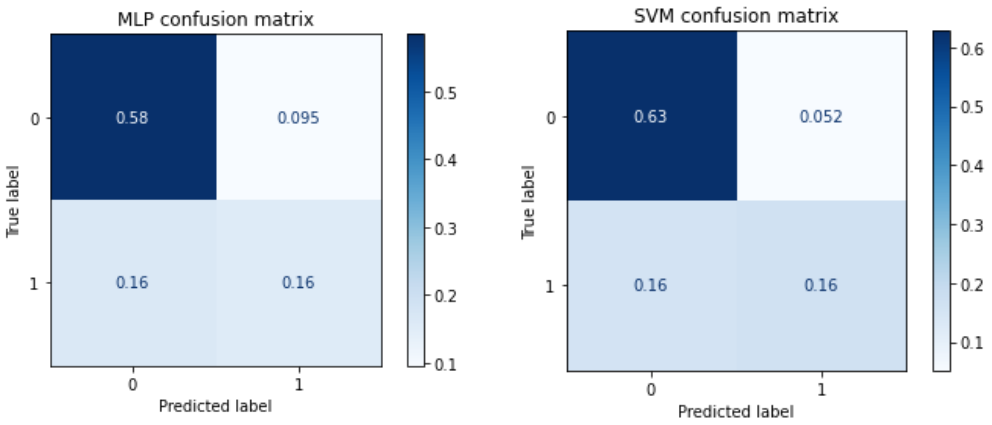


Table 2- Performance Matrix

	MLP	**SVM**
Precision	0.78 | 0.62	0.80 | 0.76
Recall	0.85 | 0.48	0.92 | 0.51
F1	0.81 | 0.54	0.85 | 0.61

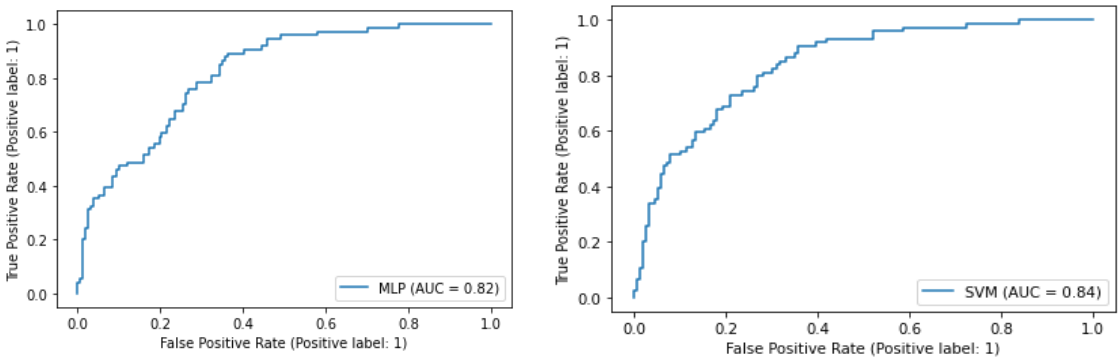
Table 3 – Accuracies of models with computational time

	MLP	**SVM**
Train	0.76	0.78
Validation	0.74	0.77
Test	0.74	0.79
Time (sec)	61.4	6.31

Table 3 presented the classification accuracies of both models on the training set, validation, testing set alongside the computation time it took for hyperparameter tuning via grid search. According to the results, SVM achieved higher accuracy than MLP and its computationally simple, whereas MLP took longer in grid search when number of hidden layers and epochs increases.

Both the models consist of moderate bias and variance as indicated by the small fluctuations between the training and test set accuracies. However, MLP seen a slightly bigger drop on the test set accuracy which shows our MLP model did not perform greatly when it comes to reducing bias and preventing overfitting as compared to SVM.

Figure 3- ROC curve and AUC for MLP and SVM



In our ROC and AUC curves we do not see a big difference in the ROC and AUC of each model, despite of the differences in their classification accuracies. As its more important to detect diabetes in our analysis which is class 1 rather than class 0. Therefore, based on that principle SVM has a slightly higher recall score of 0.51 whereas MLP scored 0.48, which shows that SVM is more likely to identify the presence of disease as compared to MLP. Thus, we concluded that the preferred model for our disease detection task is SVM.

5. Conclusion, lessons learned and future work

In this study we critically evaluated and compared two trained models in predicting diabetes based on its features. Furthermore, we concluded that MLP and SVM are comparable and able to achieve higher accuracy by tuning its hyperparameters. However, SVM performed better than MLP in this classification problem.

We also understand that MLP performance relies upon its size of hidden layers and number of neurons in order to achieve higher accuracy. However, higher number of hidden layers require more computational time when it comes to grid search for hyperparameter tuning and training times. Whereas SVM is computationally faster than SVM and does not require much hyperparameter tuning. Thus, it is suggested that MLP is more capable of computational training than SVM and has a potential to outperform SVM when dealing with large number of hidden layers.

Future work can include a SMOTE algorithm in balancing the data and optimise the model again to see any visible improvements, however, the oversampling method tend to cause overfitting which we need to be mindful about.

5. References

1- https://www.diabetes.org.uk/professionals/position-statements-reports/statistics

2- https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

3- *PII: S1352-2310(97)00447-0 | Elsevier Enhanced Reader* (no date). doi:10.1016/S13522310(97)00447-0.

4- Jakkula, V. (no date) 'Tutorial on Support Vector Machine (SVM)', p. 13.

5- Hofmann, M. (no date) 'Support Vector Machines — Kernels and the Kernel Trick', p. 16.

6- Qolomany, B. *et al.* (2017) 'Parameters optimization of deep learning models using Particle swarm optimization', in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1285–1290. doi:10.1109/IWCMC.2017.7986470.

Appendix 1- glossary

F-1 score: The F-1 score is weighted average of the precision and recall.

Kernel Trick: Its transforms data into n-dimensional places using the hyperplane .

Stochastic gradient descent: SGD calculates the derivatives from training data and update the calculation accordingly.

Activation: It is a function which maps the weighted inputs to the output of the neuron.

Bias: An intercept which added to the weight's prior activation.

Perceptron: A supervised learning algorithm which consist of 4 segments including input, weight and bias, activation function and output.

Appendix 2-Implementation details