A Comparative Study of MLP and SVM models

Alisher Amerkhojayev

1. Introduction

Diabetes is becoming a prevalent issue across the world affecting people regardless of their gender, age, or ethnicity. It is estimated that in 2021 there were 529 million people with diabetes worldwide and the number is projected to increase to 1.3 billion people in the next 30 years. Type 2 diabetes cases, which are preventable and could potentially be reversible if identified early, have been increasing in recent years (Ong et.al, 2023). Therefore, early detection would be essential in improving people's lives by reversing chronic disease at an early stage.

Researchers have been implementing different ML frameworks for diabetes prediction such as Naïve Bayes, Decision Trees, Logistic Regression, Support Vector Machines, Multilayer Perceptron, Gaussian Mixture Models, Random Forest, and others, which were all evaluated across different performance metrics (Komi et al., 2017, Hasan et al., 2020, Khanam & Foo, 2021, Sisodia & Sisodia, 2018, Jaiswal et al., 2021). This paper will focus on the critical evaluation of Support Vector Machine (SVM) and Multilayer Perceptron (MLP) models in the determination of diabetes diagnosis.

1.1 Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) is a type of feedforward artificial neural network (ANN) that consists of three layers of nodes: an input layer, a single or multiple hidden layers, and an output layer. MLP is a complex network of multiple layers of neurons which are all connected between the adjacent layers. The input layer is where the signal is received from external sources which then goes through the hidden layer(s) where the computations are carried out via weighted connections that are established from the neurons in the input layer, and finally, go to the output layer where the intended task is computed such as classification or prediction. Due to variability in the number of hidden layers or the number of neurons in each layer, the model can be used for simpler or more complex tasks. Activation functions are essential components of neural networks that allow them to acquire intricate patterns in data. Neural networks utilize signal transformation to convert the input signal of a node into an output signal, which is subsequently transmitted to the next layer (Ali, 2023). This is essential because without these functions neural networks would only be able to model linear relationships. MLPs are trained using the backpropagation algorithm, which calculates the gradients of a loss function in relation to the parameters of the model. These parameters are then updated iteratively to minimize the loss (Jaiswal, 2024).

The MLP's qualities render it an excellent tool for classification prediction; nonetheless, it does have numerous drawbacks. Firstly, there is a risk of encountering local minima, which can prevent the model from discovering the global minimum and so reduce the accuracy of the model. Additionally, when working with a small dataset, there is a high likelihood of overfitting, resulting in poor generalization when applied to other cases (Banerjee, 2024).

1.2 Support Vector Machine

The Support Vector Machine is an algorithm whose objective is to identify a hyperplane in an N-dimensional space that would effectively separate the data points into distinctive discrete classes (Gandhi, 2018). Support Vector Machines (SVMs) are frequently employed in the context of classification tasks. The classification process involves identifying two distinct classes by determining the best hyperplane that maximizes the distance between the nearest data points from different classes. The dimensionality of the input data determines whether the hyperplane is a line in a 2-D space or a plane in an n-dimensional space. The presence of numerous hyperplanes allows for the differentiation of classes. By maximizing the gap between points, the method is able to identify the optimal decision border between classes (IBM, 2023).

The primary benefit of the Support Vector Machine (SVM) technique is in its formulation of the learning problem, which results in a quadratic optimization challenge. It significantly decreases the amount of operations performed during the learning mode (Osowski et al., 2004). Moreover, the use of different kernels is possible in order to find a suitable choice being able to model linear and non-linear relationships. On the other hand, Support Vector Machines may have reduced performance when the number of features is greater than the number of training samples. Furthermore, SVMs do not provide probabilistic justifications for classification judgments (Dhiraj, 2023).

Considering these factors the hypothesis is the following:

In the case where is there is a high dimensionality and a considerably complex relationship between the features, the MLP would perform better than SVM due to the pattern being complex (hidden). However, if there is moderate dimensionality and the features relationship is not as complex, the SVM would perform better than MLP.

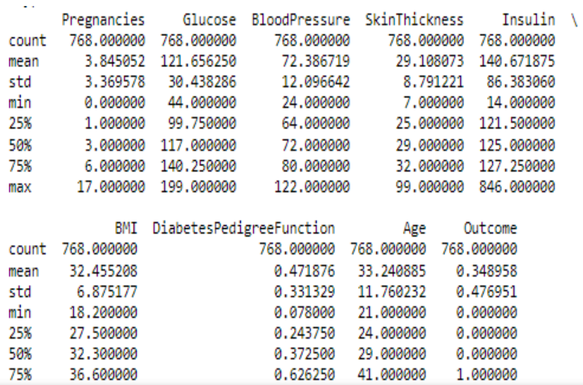
2. Data

The diabetes dataset is taken from Kaggle which was originally taken from the National Institute of Diabetes and Digestive and Kidney Diseases (Kaggle). The dataset is comprised of 768 rows of patients of Pima Indians from Arizona who are all female above the age of 21 of which 268 are diabetic and 500 are not. In order to tackle this, we implement different methods for both MLP and SVM, Smote for MLP and setting the balanced class weight for SVM. It contains 9 features 1 of them being the classification ‘Outcome’ which is the indication whether the patient has Diabetes Type 2 (1) or not (0), and which 8 are attributes: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age. The distribution of patients with and without diabetes shown in Figure 1

The dataset, contains a considerable amount of missing values in 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI' is labelled as 0, which is classified as such due to 0 being impossible in these attributes, whereas the 0 for ‘Pregnancies’ is normal. In order to tackle this problem, we employ the method of substituting missing values with the median values in the corresponding columns. We do so by setting the 0 as the NaN values and later substituting them with the median values. This has been put in place in order to avoid unintentional misplacement of the numbers from different columns. The problem is there is 374 missing values for Insulin and 227 for SkinThickness which could skew our results. After that we can perform the initial data analysis.

2.1 Initial Data Analysis

After implementing the aforementioned modifications, generate a Table 1 that displays the numerical values of each attribute and their corresponding distribution. This is specifically done to observe any latent distribution characteristics of the data. We utilise histograms to illustrate the distribution of each attribute shown in Figure 1 The Glucose, BloodPressure, and BMI are exibitng normal distrubtion while others are positively skewed. The Insulin and SkinThickness have a mode for the median values due to the high number of missing values. Moreover, we would not be removing the outliers due to the nature of the dataset as people are distributed unevenly and it would be essential to include these statistics to try to classify even for those who are on the far spectrum of the distributions. This could introduce noise and deviations but these patient are still important in the classification of diabetes.



```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  121.656250      72.386719      29.108073  140.671875
std       3.369578   30.438286      12.096642       8.791221   86.383060
min       0.000000   44.000000      24.000000       7.000000   14.000000
25%       1.000000   99.750000      64.000000      25.000000  121.500000
50%       3.000000  117.000000      72.000000      29.000000  125.000000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    32.455208                  0.471876   33.240885    0.348958
std      6.875177                  0.331329   11.760232    0.476951
min     18.200000                  0.078000   21.000000    0.000000
25%     27.500000                  0.243750   24.000000    0.000000
50%     32.300000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
```

Table 1

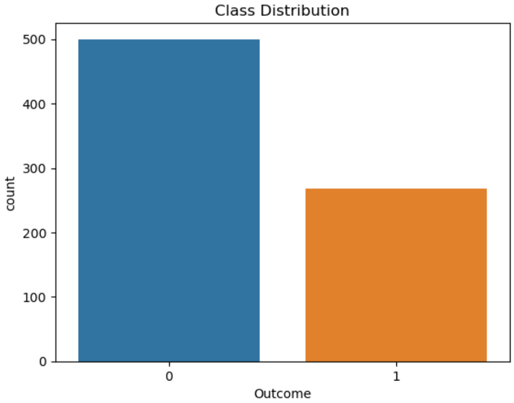


Figure 1

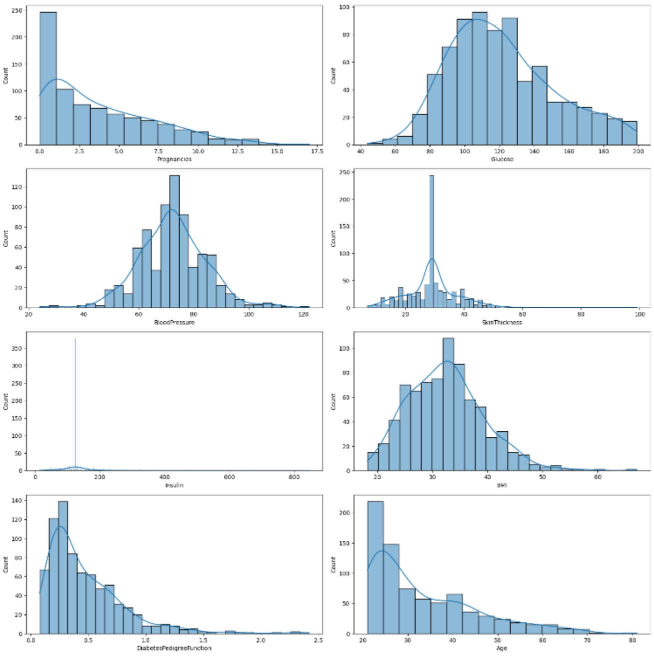


Figure 2

We generate a correlation matrix to identify any redundant or highly associated qualities among the features. Our analysis reveals a positive association between all qualities and the Outcome. This suggests that variables with values greater than the median could potentially serve as risk factors. Furthermore, we observe that most of the attributes exhibit a weak correlation with each other, suggesting their potential independence. However, there are a few exceptions, such as Age and Pregnancies, and Skin Thickness and BMI, which show a stronger correlation. These correlations can be explained by the fact that as Age increases, there tends to be more offspring, and as BMI increases, the skin thickness also tends to increase, indicating that the person's weight exceeds the healthy range leading to thicker skin due to excess fat.

3. Methodology

We normalize all the attributes is an important factor as it would improve the accuracy and the speed up the calculations in our model. Due to the difference in the magnitude of our attributes it is absolutely essential to normalize them as they would skew our results due to some of the attributes being disproportionately large, influencing the models in the process (Sola & Sevilla, 1997). Moreover, we split the dataset into training and testing data, 80% and 20% respectively.

3.1 MLP

For the MLP model, we initialize the maximum number of epochs as 300 for the convergence of our model, where would change the number in case the convergence happens faster. The higher number will allow our model to learn more from the data and thus set at this number. Since the data is unbalanced, rather than using the data as it is, we employ Smote in order to tackle the said unbalance for generalization. In the parameter grid, we set 1 and 2 hidden layers as our dataset is relatively small and employs 5,10,15 neurons due to the fact that we only have 8 attributes.

For the activation function, we use the ‘relu’ and the ‘tanh’ to account for linear and non-linear gradients. Relu is expected to perform better due to the disregard of the vanishing gradient problem (Kumar, 2023). In order to tackle the problem of overfitting, we introduce the regularization term alpha. The parameter alpha determines the magnitude of the step taken at each iteration of the gradient descent method. It determines the rate at which the algorithm approaches the best possible result. The alpha value is essential since it can significantly influence the optimization process. An elevated alpha value can result in the algorithm surpassing the minimum point, resulting in oscillations or potentially even divergence. Conversely, a small alpha value might lead to sluggish convergence, necessitating a greater number of iterations to achieve the ideal solution (Coursetech) Thus it is important to set the value of alpha at the optimal and we do so by the method of iterations. Next, they introduce two ‘solver’ which update the weights and the biases of the model: ‘adam’ and ‘sgd’.

Additionally, we set the initial learning rate at different points for a better understanding of which initial learning rate is optimal. Next, we introduce GridSearchCV for optimization to find the optimal hyperparameters for our model. We set the cross-validation at 5. After the best parameters are found, we use them for our test data.

3.2 SVM

For the SVM model, since the dataset is unbalanced, we set the class weight to be balanced to adjust the weights inversely proportional to class frequencies in the data. Next, we define the parameter grid. Firstly, we employ the regularization parameter C to avoid the overfitting problem and thus set at 0.1, 1, and 10. It regulates the balance between optimizing the margin and reducing the training error (IBM). We set the kernel to three different types: rbf, poly, and linear. The gamma term is set to ‘auto’ and ‘scale’ from the scikit-learn to find out which term best suits our model. The degree we look at for the ‘poly’ kernel is 3,4. We use the GridSearchCV for the best hyperparameters and set our cross-validation at 5. After the best parameters are found, we use them for our test data.

4. Results, Findings, and Evaluation

Figures 3 and 4 show the heatmap of the best models chosen for the MLP and SVM models and the accuracies of the hyperparameters set at the grid search. For the MLP the best model performance was shown at {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (15, 15) meaning 2 hidden layers and 15 neurons, 'learning_rate_init': 0.01, 'solver': 'adam'} which scored a validation accuracy of 0.843. Meanwhile, the best SVM model performance was shown at { 'C': 1, 'degree': 3, 'gamma': 'scale', 'kernel': 'rbf'} with a validation accuracy of 0.769. Moreover, we can see from the heatmap that the accuracy within the models varies depending on the hyperparameters. We can see that MLP had a smaller absolute difference in the validation scores of 0.041 of different parameters, while the SVM model had an absolute discrepancy of 0.108 which indicates that the SVM model is more sensitive to the changes in the hyperparameters than MLP.

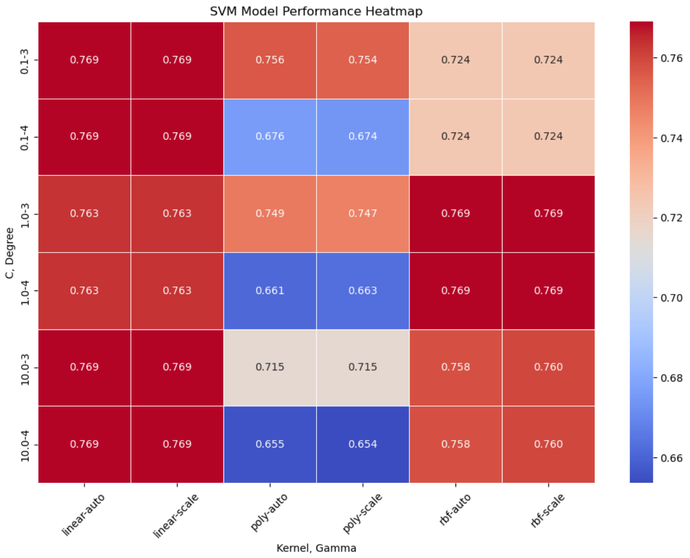


Figure 3

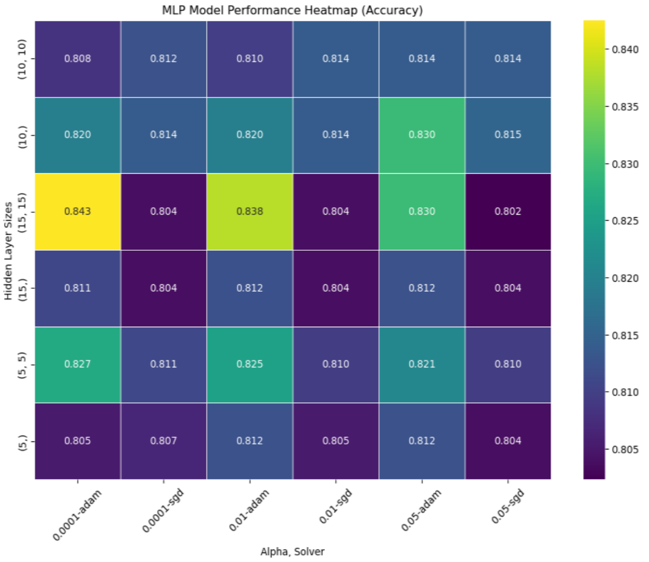


Figure 4

However, on the other hand, when we look at the accuracy scores of the test sample, the SVM has a higher score of 0.723 while MLP had 0.693. This indicates that the SVM model performed better with the best hyperparameters than MLP. This could be explained by the different methods of tackling the problem of the unbalanced dataset. The use of SMOTE had improved the validation accuracy of the MLP and so did the setting of balanced weight classes for SVM, however, the unseen test data was unbalanced and so the SVM model performed better than the MLP model. As discussed before, there is a chance of overfitting in the smaller dataset for the MLP model which brought the SVM model on top.

Furthermore, when we look at the confusion matrices, we see that SVM had TN=105, FN= 18, FP= 46, and TP=62, while the MLP model had TN=111, FN= 25, FP= 42, and TP=51. In the diagnosis of diabetes type 2, it is important to indicate the number of TP and FN where the SVM model had indicated a higher number of cases in TP and a lower number of FN. This framework shows that SVM performed better at indicating the TP which are the cases where the diagnosis has been given right and lower number of FN where the models have incorrectly misclassified the patients with diabetes type 2. This is further solidified, in a Receiver Operation Curve (ROC) which was plotted in Figure 5 to check for the Area Under the Curve (AUC) which has shown that the SVM had a higher AUC of 0.79 while MLP had AUC of 0.72. The AUC curve is important as it is not influenced by the number of samples in each class (Obuchowski, 2003).

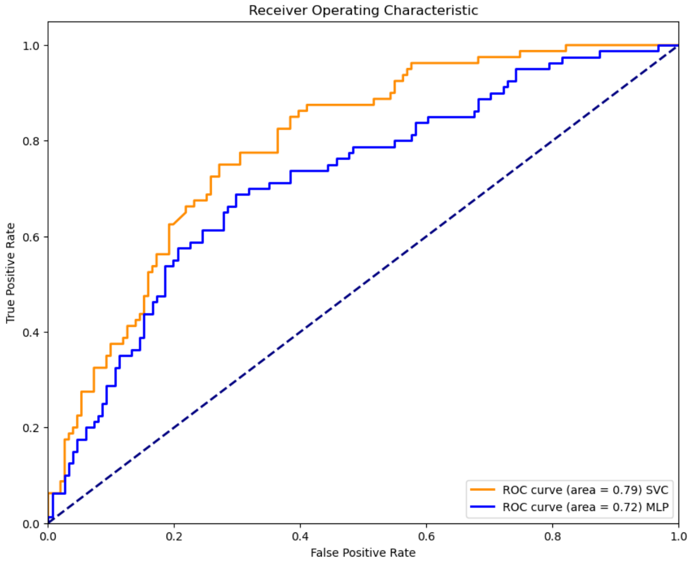


Figure 5

5. Conclusion

In our comparative study, we employed a Multilayer Perceptron and Support Vector Machine to classify patients with diabetes type 2. We evaluated the performance of both models in predicting outcomes using unseen data.

In conclusion, both models have performed similarly in this task. Both models have been constructed to the best knowledge and accuracy possible. Since each model has its own setting, it is important to understand that each of the models can be used differently for the same task. Our study has shown that the SVM model is slightly better at classifying the diabetes type 2 patients both in terms of accuracy on the unseen data and the AUC. However, the MLP model had performed better at the training data which could be influenced by the use of SMOTE or explained by the possible overfitting even the steps taken to avoid it.

The takeaway from performing these tasks is that the models are applicable and can be used for classification on this dataset and could be done differently which would yield different results. It is important to consider specific nuances of models, regardless of whether it would be MLP, SVM, or any other. Moreover, we learned how unbalanced data could influence the model results and it is always crucial to examine the data first. I recognize that in this study, there could be a more diverse set of hyperparameters that I have not tested, which could be a potential improvement for the future.

References

[1] K. L. Ong et al., “Global, regional, and national burden of diabetes from 1990 to 2021, with projections of prevalence to 2050: A systematic analysis for the global burden of disease study 2021,” The Lancet, vol. 402, no. 10397, pp. 203–234, Jul. 2023. doi:10.1016/s0140-6736(23)01301-6

[2] M. Komi, J. Li, Y. Zhai, and X. Zhang, “Application of data mining methods in diabetes prediction,” *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, Jun. 2017. doi:10.1109/icivc.2017.7984706

[3] Md. K. Hasan, Md. A. Alam, D. Das, E. Hossain, and M. Hasan, “Diabetes prediction using ensembling of different machine learning classifiers,” IEEE Access, vol. 8, pp. 76516–76531, 2020. doi:10.1109/access.2020.2989857

[4] J. J. Khanam and S. Y. Foo, “A comparison of machine learning algorithms for diabetes prediction,” ICT Express, vol. 7, no. 4, pp. 432–439, Dec. 2021. doi:10.1016/j.icte.2021.02.004

[5] D. Sisodia and D. S. Sisodia, “Prediction of diabetes using classification algorithms,” Procedia Computer Science, vol. 132, pp. 1578–1585, 2018. doi:10.1016/j.procs.2018.05.122

[6] V. Jaiswal, A. Negi, and T. Pal, “A review on current advances in machine learning based diabetes prediction,” Primary Care Diabetes, vol. 15, no. 3, pp. 435–443, Jun. 2021. doi:10.1016/j.pcd.2021.02.005

[7] M. Ali, “Introduction to activation functions in neural networks,” DataCamp, https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks.

[8] S. Jaiswal, “Multilayer perceptrons in Machine Learning: A comprehensive guide,” DataCamp, https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning.

[9] S. Banerjee, “Exploring the power and limitations of Multi-Layer Perceptron (MLP) in Machine Learning,” Medium, https://shekhar-banerjee96.medium.com/exploring-the-power-and-limitations-of-multi-layer-perceptron-mlp-in-machine-learning-d97a3f84f9f4 (accessed Apr. 19, 2024).

[10] R. Gandhi, “Support Vector Machine - introduction to machine learning algorithms,” Medium, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (accessed Apr. 19, 2024).

[11] “What is support vector machine?,” IBM, https://www.ibm.com/topics/support-vector-machine (accessed Apr. 19, 2024).

[12] S. Osowski, K. Siwek, and T. Markiewicz, Proceedings of the 6th Nordic Signal Processing Symposium - Norsig 2004, Jun. 2004.

[13] K. Dhiraj, “Top 4 advantages and disadvantages of support vector machine or SVM,” Medium, https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107.

[14] A. D. Khare, “Diabetes dataset,” Kagg4le, https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset/data.

[15] J. Sola and J. Sevilla, “Importance of input data normalization for the application of neural networks to complex industrial problems,” IEEE Transactions on Nuclear Science, vol. 44, no. 3, pp. 1464–1468, Jun. 1997. doi:10.1109/23.589532

[16] S. Kumar, “Comparison of sigmoid, Tanh and Relu activation functions,” AITUDE, https://www.aitude.com/comparison-of-sigmoid-tanh-and-relu-activation-functions/.

[17] Coursesteach, “Machine learning (part 8),” Medium, https://medium.com/@Coursesteach/machine-learning-part-8-2e5e4c92de4b (accessed Apr. 19, 2024).

[18] IBM, “SVM Node Expert Options,” SVM node expert options, https://www.ibm.com/docs/en/spss-modeler/saas?topic=node-svm-expert-options (accessed Apr. 19, 2024).

[19] N. A. Obuchowski, “Receiver operating characteristic curves and their use in Radiology,” Radiology, vol. 229, no. 1, pp. 3–8, Oct. 2003. doi:10.1148/radiol.2291010898.

Appendix A

Pregnancies: To express the Number of pregnancies

Glucose: To express the Glucose level in blood

BloodPressure: To express the Blood pressure measurement

SkinThickness: To express the thickness of the skin

Insulin: To express the Insulin level in blood

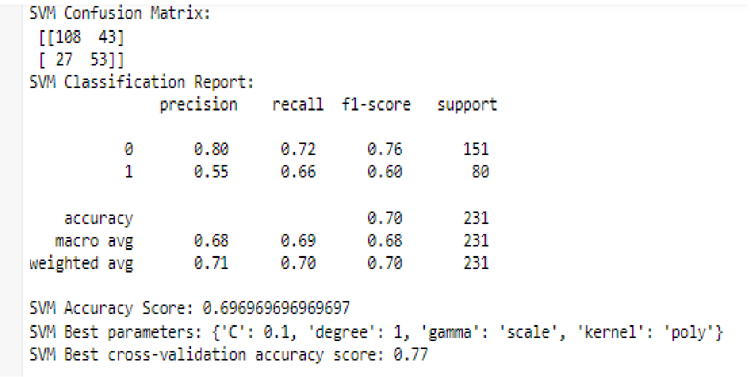
BMI: To express the Body mass index

DiabetesPedigreeFunction: To express the Diabetes percentage

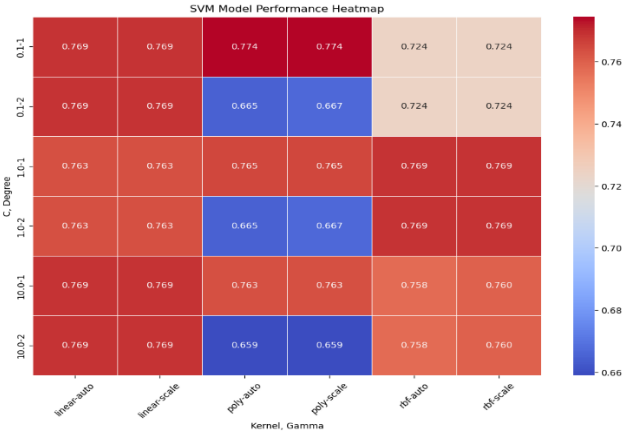
Age: To express the age

Outcome: To express the final result 1 is Yes and 0 is No

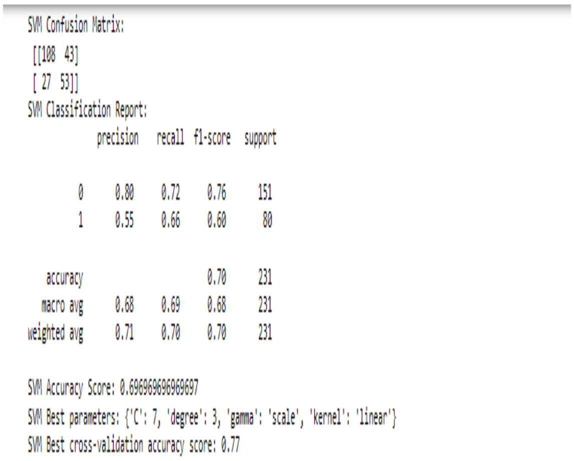
SVM Degree 1,2, C 0.1, 5, 7.



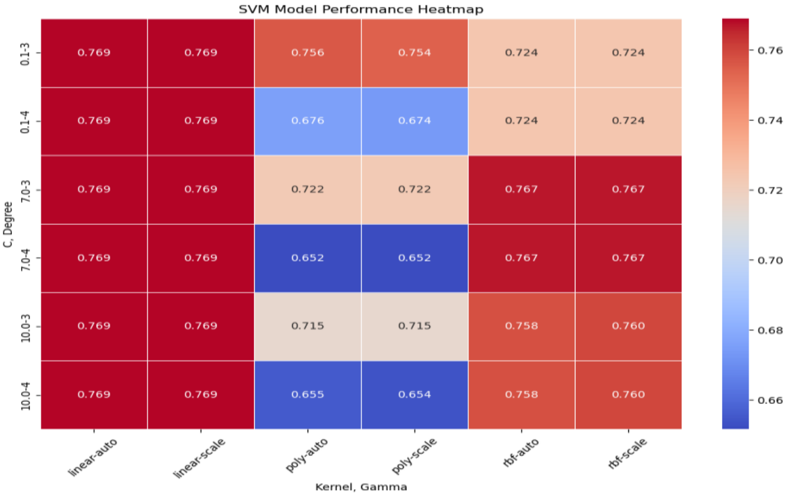
```
SVM Confusion Matrix:
 [[108  43]
 [ 27  53]]
SVM Classification Report:
               precision    recall  f1-score   support

           0       0.80      0.72      0.76       151
           1       0.55      0.66      0.60        80

    accuracy                           0.70       231
   macro avg       0.68      0.69      0.68       231
weighted avg       0.71      0.70      0.70       231

SVM Accuracy Score: 0.696969696969697
SVM Best parameters: {'C': 0.1, 'degree': 1, 'gamma': 'scale', 'kernel': 'poly'}
SVM Best cross-validation accuracy score: 0.77
```



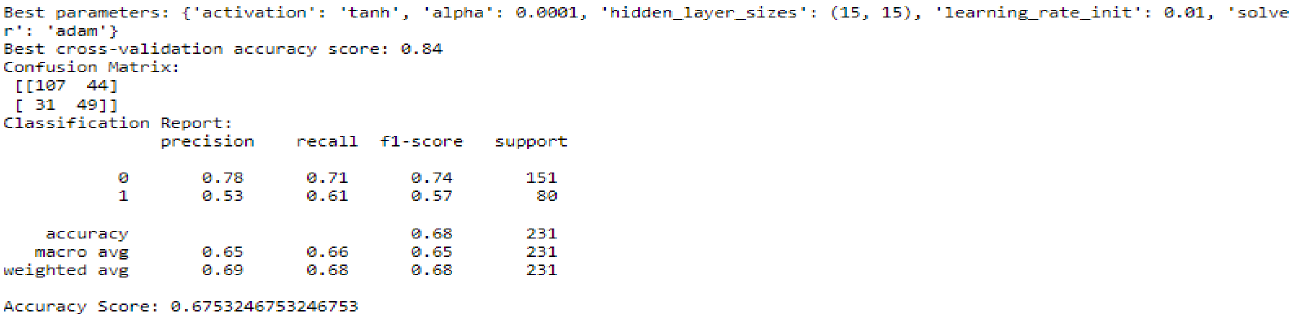
SVM Degree 3,4 C 0.1, 7, 10



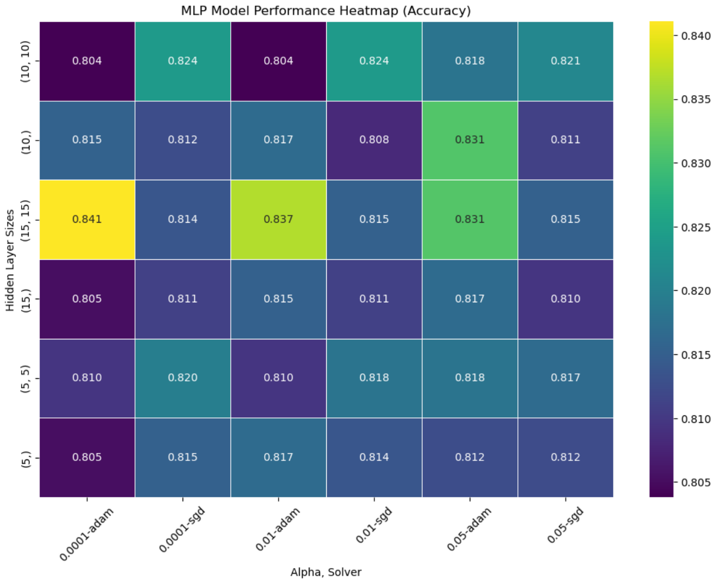
```
SVM Confusion Matrix:
 [[108  43]
 [ 27  53]]
SVM Classification Report:
               precision    recall  f1-score   support

           0       0.80      0.72      0.76       151
           1       0.55      0.66      0.60        80

    accuracy                           0.70       231
   macro avg       0.68      0.69      0.68       231
weighted avg       0.71      0.70      0.70       231

SVM Accuracy Score: 0.696969696969697
SVM Best parameters: {'C': 7, 'degree': 3, 'gamma': 'scale', 'kernel': 'linear'}
SVM Best cross-validation accuracy score: 0.77
```



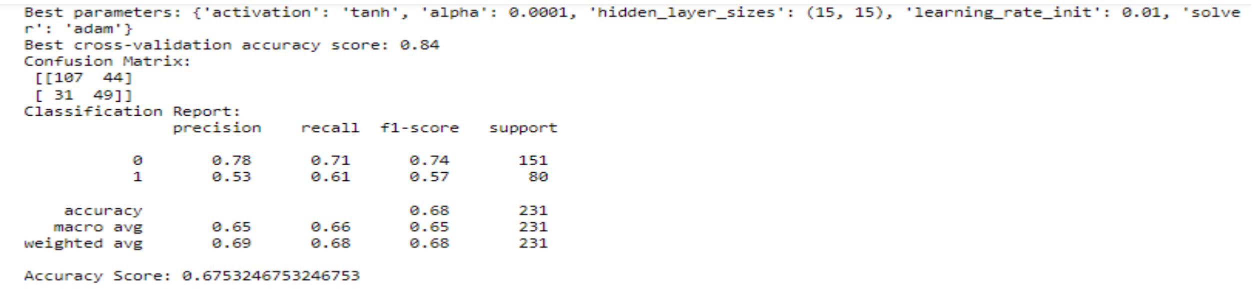
MLP max iterations 500



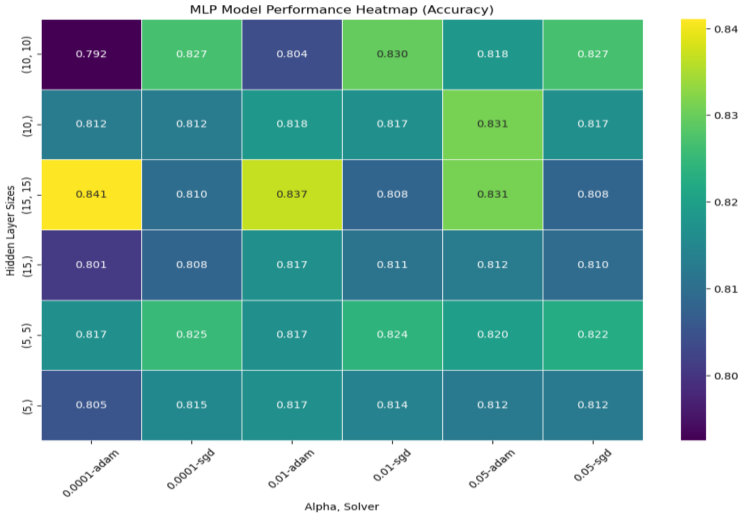
```
Best parameters: {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (15, 15), 'learning_rate_init': 0.01, 'solve
r': 'adam'}
Best cross-validation accuracy score: 0.84
Confusion Matrix:
 [[107  44]
 [ 31  49]]
Classification Report:
               precision    recall  f1-score   support

           0       0.78      0.71      0.74       151
           1       0.53      0.61      0.57        80

    accuracy                           0.68       231
   macro avg       0.65      0.66      0.65       231
weighted avg       0.69      0.68      0.68       231

Accuracy Score: 0.6753246753246753
```



MLP max iterations 1000



```
Best parameters: {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (15, 15), 'learning_rate_init': 0.01, 'solve
r': 'adam'}
Best cross-validation accuracy score: 0.84
Confusion Matrix:
 [[107  44]
 [ 31  49]]
Classification Report:
               precision    recall  f1-score   support

           0       0.78      0.71      0.74       151
           1       0.53      0.61      0.57        80

    accuracy                           0.68       231
   macro avg       0.65      0.66      0.65       231
weighted avg       0.69      0.68      0.68       231

Accuracy Score: 0.6753246753246753
```


MLP Model Performance Heatmap (Accuracy)
Hidden Layer Sizes
(10, 10)
(10,)
(15, 15)
(15,)
(5, 5)
(5,)
0.0001-adam
0.0001-sgd
0.01-adam
0.01-sgd
0.05-adam
0.05-sgd
Alpha, Solver