A Comparative Study on Diabetes Prediction through Multilayer Perceptrons and Support Vector Machines

Ratna Pathak
Ratna.Pathak@city.ac.uk

Abstract

This paper aims to present a critical analysis and evaluation of two algorithm models : Multilayer Perceptron (MLP) and Support Vector Machines (SVM) performed on diabetes prediction. Several combinations of hyperparameters were examined, and the optimal models were assessed on a test dataset using various metrics such as Recall , F score and Receiver Operating Characteristic (ROC) curves, as well as ROC area values (AUC). Among the two models tested, MLP proved to be the most efficient one.

1. Introduction:

Diabetes is a chronic metabolic disorder that impacts a significant number of individuals globally. The condition arises due to insufficient insulin production or insulin resistance, leading to the body's inability to regulate blood sugar levels. If left uncontrolled, diabetes can lead to severe health complications, such as kidney damage, blindness, and heart disease. Early detection and management of diabetes are crucial to prevent these complications. Machine learning algorithms based on predictive models have shown great potential in predicting the onset of diabetes by utilising features like blood pressure, age, BMI, and family history to identify individuals at high risk of developing diabetes. Hence, it is essential to investigate various configurations of machine learning algorithms like MLP and SVM to determine which algorithm is suitable for predicting diabetes.

2. Brief summary of the two neural network models with their pros and cons:

2.1. Multilayer Perceptron (MLP):

Multilayer Perceptron, is the most regularly adopted approach for the purpose of pattern recognition [2]. MLPs are supervised learning classifiers that utilise an input layer, an output layer, and one or more hidden layers to extract crucial information during the learning process, and adjust the weighting coefficients of input components. Based on the amount of error in the output compared to the expected result, the weights in the perceptron are modified through the backpropagation.Compared to other probability-based models, MLPs are independent of any assumptions about the underlying probability density functions or probabilistic informations related to the pattern classes.

There is a need for careful tuning of hyperparameters, without proper tuning, the model's performance may be suboptimal. Finally, MLP is considered a black box model,the interpretability of MLP models is limited, and it can be challenging to understand how the model is making predictions, which may be important for medical applications.It can easily overfit the training data, resulting in poor generalisation performance on new data. This can be mitigated by using regularisation techniques or early stopping. MLP can be computationally expensive to train, especially for larger or more complex problems [2].

2.2. Support Vector Machines (SVM):

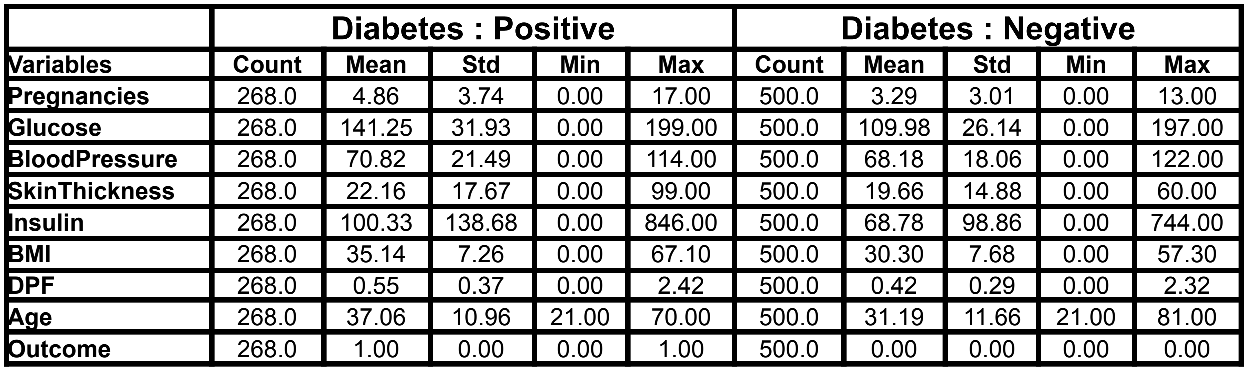
SVMs are a type of supervised learning algorithm that identifies a hyperplane in the feature space for distinguishing data classes. They work by determining the optimal hyperplane that divides the various classes in the data, with the hyperplane serving as the boundary between the classes with the largest margin. SVMs can manage high-dimensional data and nonlinear boundaries using the kernel trick, and are memory-efficient since they only need to store a subset of the training data, known as support vectors, thereby reducing memory consumption.SVMs have the ability to resist overfitting due to the presence of a regularisation parameter that balances the trade-off between achieving a low training error and a low testing error, thereby preventing overfitting.

One limitation of SVM is its sensitivity to the choice of kernel function, which can heavily impact its performance. Selecting the appropriate kernel function can be challenging and may require expert knowledge. SVMs are deterministic, meaning they do not offer probability estimates for predictions. As the dataset size increases, SVMs become computationally expensive and memory-intensive, potentially requiring significant computational resources. Additionally, SVM are directly suitable for binary classification problems and necessitate the framing of multi-class tasks as a series of binary tasks [2].

3. Dataset:

The data set is taken from kaggle [3], Originally from the National Institute of Diabetes and Digestive and Kidney Diseases. It aims to determine if a patient has diabetes by utilising certain diagnostic measurements. The dataset consists entirely of female patients who are 21 years or older and have Pima Indian heritage. Along with the target variable, which is Outcome, there are several medical predictor variables included in the dataset, such as BMI, insulin level, number of pregnancies, age, and other factors. [3]

Table 1 - Statistic Summary of The Dataset:



	Diabetes : Positive					Diabetes : Negative				
Variables	Count	Mean	Std	Min	Max	Count	Mean	Std	Min	Max
Pregnancies	268.0	4.86	3.74	0.00	17.00	500.0	3.29	3.01	0.00	13.00
Glucose	268.0	141.25	31.93	0.00	199.00	500.0	109.98	26.14	0.00	197.00
BloodPressure	268.0	70.82	21.49	0.00	114.00	500.0	68.18	18.06	0.00	122.00
SkinThickness	268.0	22.16	17.67	0.00	99.00	500.0	19.66	14.88	0.00	60.00
Insulin	268.0	100.33	138.68	0.00	846.00	500.0	68.78	98.86	0.00	744.00
BMI	268.0	35.14	7.26	0.00	67.10	500.0	30.30	7.68	0.00	57.30
DPF	268.0	0.55	0.37	0.00	2.42	500.0	0.42	0.29	0.00	2.32
Age	268.0	37.06	10.96	21.00	70.00	500.0	31.19	11.66	21.00	81.00
Outcome	268.0	1.00	0.00	0.00	1.00	500.0	0.00	0.00	0.00	0.00

4. Initial Data Analysis:

As visualised in the box-plot, Figure 1, there are not many outliers, and it won't be necessary to remove them. Insulin has a higher range which might be the result of data entry errors, measurement errors, or genuine extreme values in the population, this outlier may provide valuable information and should not be removed without careful consideration. Certain features (like, Glucose, BloodPressure, SkinThickness, Insulin, BMI) had values of 0, which may not be appropriate for the given context and could suggest the presence of missing data. The missing data was filled with mean of respective Positive and Negative categories.

In order to consider the possible influence of multicollinearity on machine learning algorithms such as SVM,we used a correlation matrix visualisation to determine if there are any variables with high correlation.As shown in Figure 2, the correlation matrix demonstrates that there are no correlations exceeding 0.7 or -0.7 between any of the variables, also the variables are less in number, suggesting that conducting principal component analysis may not be necessary.

In order to maintain data consistency, we standardized each variable by scaling them to a range between 0 and 1 during the data preparation phase. This was necessary because the variables had different ranges of values.

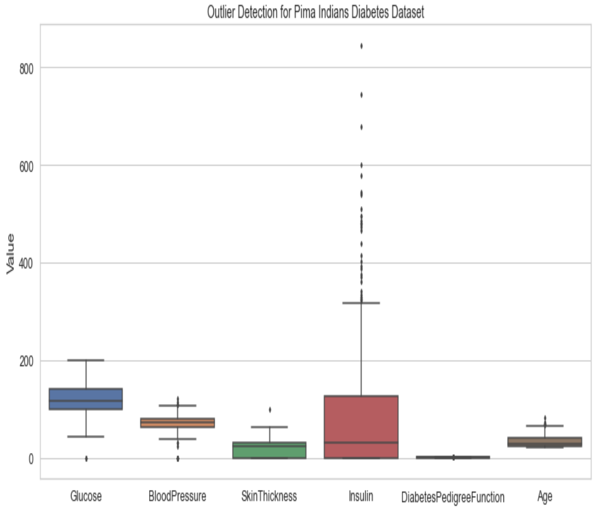


Figure 1. Box plot

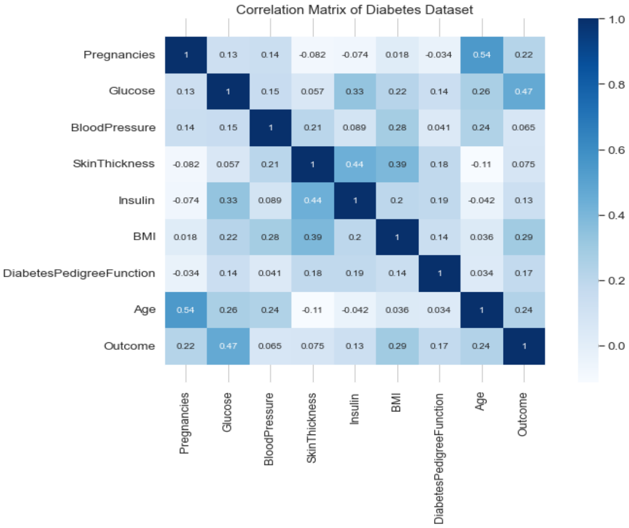


Figure 2. Correlation Matrix

5. Hypothesis Statement:

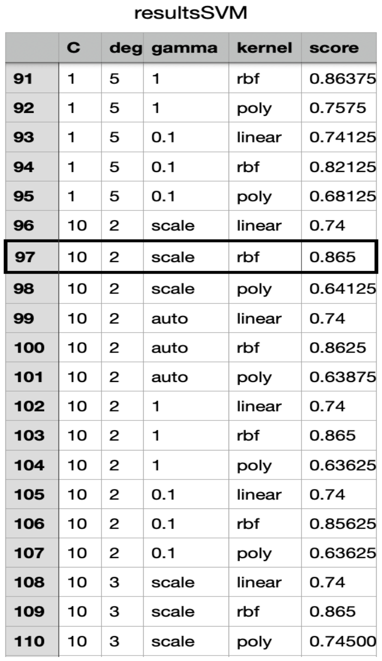
The previous experiments performed using: MLP and SVM have confirmed that both solutions are very well suited for classification, regression and prediction tasks. In classification mode the unbeatable is SVM, while in regression, MLP poses better generalisation ability[4]. Taking into account that we are conducting a binary classification task, we can reasonably expect that SVM will outperform MLP. A study conducted by Kayaer and Yildirim [6] in 2003 was able to achieve a 77.08% result for correct prediction of a test set using the Levenberg-Marquardt training (a variation of backpropagation) algorithm of MLP.A study conducted by Kumari [5] in 2013 utilised SVM with RBF kernel on the PIDD dataset, resulting in an accuracy of 78%, sensitivity of 80%, and specificity of 76.5%.Our approach will ensure that both algorithms have an equal opportunity to optimise, so the grid search optimization will be allocated approximately equal time to train.

6. Description of choice of training and evaluation methodology:

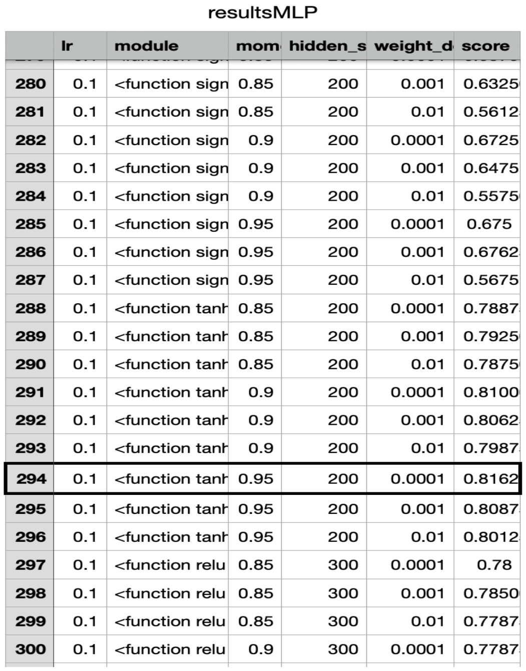
Methodology: To start our analysis, we divided the dataset into a training set and a test set, comprising 80% and 20% of the data, respectively. To account for proportionality in the data subset, we employed the stratify parameter during the splitting process.To handle the imbalanced dataset SMOTE was applied [9] .We also applied feature scaling to the dataset, StandardScaler was the chosen scaling method, which can reduce the impact of outliers and improve the performance of machine learning models.To further ensure the reliability of the results, we employed grid search with 10 fold cross-validation to fine-tune their hyperparameters to reduce overfitting and increase the accuracy.This approach is widely recommended in the machine learning literature[7]. To perform a detailed analysis of the results, we will compute precision, recall, and F-score, and generate a confusion matrix and ROC curve. Additionally, we will calculate the area under the ROC curve (AUC) to evaluate the performance of our two models, so that the performance of the models are compared more comprehensively.

7. Choice of parameters and experimental results:

For the SVM model, we optimised on hyperparameters such as the kernel function (linear, polynomial, or radial basis), regularisation parameter C with different values, degrees of the function, and gamma values. Meanwhile, for the MLP model, we optimised on the number of neurons in the hidden layer, activation function for the hidden layer (logistic, tanh, and relu), learning rate, optimizer momentum, hidden size, and weight decay with various values. We also utilised dropout in the neural network to prevent overfitting, resulting in a more generalised model.[8] To create a starting point for comparison, we developed simple models that were not subjected to any hyperparameter tuning.The Optimised MLP and SVM was able to perform slightly better than the baseline model specially for positive output. Accuracy was used as the scoring method for the grid searches, As shown in table 2. RBF in general performed better in SVM, Tanh and relu performed better in MLP.


resultsSVM

	C	deg	gamma	kernel	score
91	1	5	1	rbf	0.86375
92	1	5	1	poly	0.7575
93	1	5	0.1	linear	0.74125
94	1	5	0.1	rbf	0.82125
95	1	5	0.1	poly	0.68125
96	10	2	scale	linear	0.74
97	10	2	scale	rbf	0.865
98	10	2	scale	poly	0.64125
99	10	2	auto	linear	0.74
100	10	2	auto	rbf	0.8625
101	10	2	auto	poly	0.63875
102	10	2	1	linear	0.74
103	10	2	1	rbf	0.865
104	10	2	1	poly	0.63625
105	10	2	0.1	linear	0.74
106	10	2	0.1	rbf	0.85625
107	10	2	0.1	poly	0.63625
108	10	3	scale	linear	0.74
109	10	3	scale	rbf	0.865
110	10	3	scale	poly	0.74500


resultsMLP

	lr	module	mom	hidden_s	weight_d	score
280	0.1	<function sign	0.85	200	0.001	0.6325
281	0.1	<function sign	0.85	200	0.01	0.5612
282	0.1	<function sign	0.9	200	0.0001	0.6725
283	0.1	<function sign	0.9	200	0.001	0.6475
284	0.1	<function sign	0.9	200	0.01	0.5575
285	0.1	<function sign	0.95	200	0.0001	0.675
286	0.1	<function sign	0.95	200	0.001	0.6762
287	0.1	<function sign	0.95	200	0.01	0.5675
288	0.1	<function tanh	0.85	200	0.0001	0.7887
289	0.1	<function tanh	0.85	200	0.001	0.7925
290	0.1	<function tanh	0.85	200	0.01	0.7875
291	0.1	<function tanh	0.9	200	0.0001	0.8100
292	0.1	<function tanh	0.9	200	0.001	0.8062
293	0.1	<function tanh	0.9	200	0.01	0.7987
294	0.1	<function tanh	0.95	200	0.0001	0.8162
295	0.1	<function tanh	0.95	200	0.001	0.8087
296	0.1	<function tanh	0.95	200	0.01	0.8012
297	0.1	<function relu	0.85	300	0.0001	0.78
298	0.1	<function relu	0.85	300	0.001	0.7850
299	0.1	<function relu	0.85	300	0.01	0.7787
300	0.1	<function relu	0.9	300	0.0001	0.7787

8. Analysis and critical evaluation of results

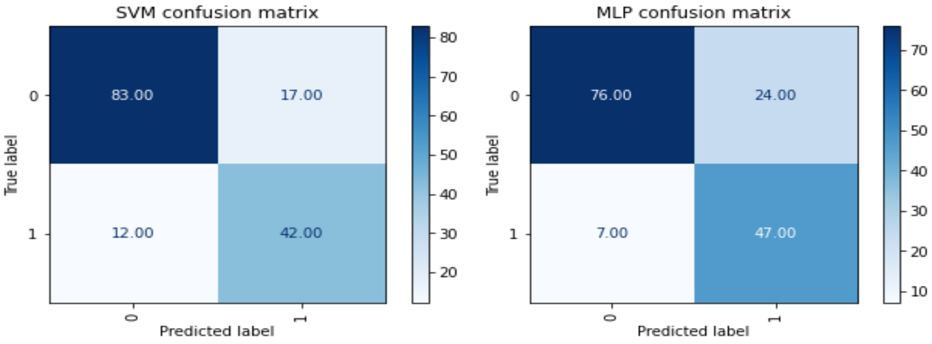
Selecting best model:

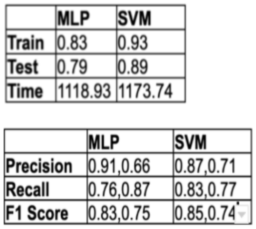
For the SVM model, the best hyperparameters were a regularisation parameter of 10, a polynomial degree of 2, a gamma value of scale, and an RBF kernel type, which resulted in an accuracy score of 0.865. For the MLP model, the optimal hyperparameters were a learning rate of 0.1, a hidden size of 200, a tanh activation function,an optimizer with 0.95 momentum, and a weight decay of 0.0001, which produced the highest classification accuracy of 0.8162.

Overall, these metrics suggest that the SVM model with the identified hyperparameters performed reasonably well considering approximately the same amount of time allotted for optimising the hyperparameter, Although MLP has higher accuracy in identifying positive instances compared to negative instances.

Table 2 - A section of Results of both grid searches for the models:

Analysis of results:





	MLP	SVM
Train	0.83	0.93
Test	0.79	0.89
Time	1118.93	1173.74

	MLP	SVM
Precision	0.91,0.66	0.87,0.71
Recall	0.76,0.87	0.83,0.77
F1 Score	0.83,0.75	0.85,0.74

Figure 3. Confusion Matrix

Table 4. Results

Figure 4 displays the ROC curves and AUC for each model. MLP have higher AUC than SVM. For our diabetes detection analysis, accurately classifying class 1 (presence of diabetes) is more important than class 0, as diabetes is a life-threatening condition. Table 4 shows that MLP had a higher recall of 0.87 for class 1 compared to SVM's recall of 0.77, indicating that MLP is more likely to detect diabetes than SVM, even if the classification is incorrect. Additionally, MLP achieved a slightly higher F-score of 0.75 for class 1 compared to MLP's F-score of 0.74.The conclusion is that MLP and SVM are commensurable,Both models had similar results. Therefore, despite SVM having an overall higher accuracy in hyperparameter tuning, we conclude that MLP is the preferred model for our diabetes detection task. This paper assumes that detecting class 1 is more important, although it is worth noting that stakeholders may have varying priorities and could seek to minimise false negatives. In such cases, precision could serve as a more suitable performance measure.

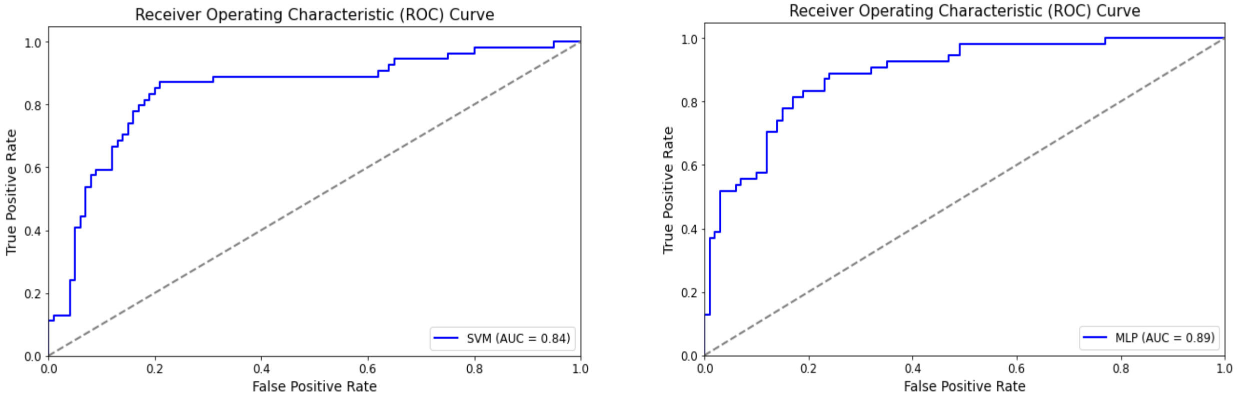


Figure 4. ROC curve

9. Lessons learned and future work:

In our findings, we discovered that the performance of MLP is greatly influenced by the number of neurons and size of the hidden layers. Nonetheless, having larger hidden layers leads to increased computing time required for hyperparameter tuning and training. Consequently, we suggest that while MLP has superior computational training capabilities compared to SVM, however, it's worth noting that the cost of optimization might outweigh the benefits. In our study, we discovered that the tanh activation function was the most effective in the MLP model and RBF kernel was most efficient in the SVM model and had a significant impact on the training scores. The ROC curve and AUC values are valuable metrics for assessing models. To improve future research models, we recommend trying different training techniques like boosting, and using feature extraction methods like PCA on the dataset for better performance [8]. It would be intriguing to investigate the possibility of incorporating penalties into the loss function during model training to achieve improved performance by allocating greater weight to correctly predicted positive classes. Additionally, introducing noise to the data before model training may enhance generalisation, ultimately resulting in improved performance on unseen data during testing.

8. Reference

[1] M. Jahangir, H. Afzal, M. Ahmed, K. Khurshid and R. Nawaz, "An expert system for diabetes prediction using auto tuned multi-layer perceptron," 2017 Intelligent Systems Conference (IntelliSys), London, UK, 2017, pp. 722-728, doi: 10.1109/IntelliSys.2017.8324209.

[2] Bishop, C.M., 1995. Neural networks for pattern recognition. Oxford university press.

[3] Dataset: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[4] L. Ljung and G. Karlsson, "MLP and SVM networks - a comparative study," Signal Processing Symposium, 2004. NORSIG 2004. Proceedings of the 6th Nordic, 2004, pp. 153-156, doi: 10.1109/NORSIG.2004.250120.

[5] A. Kumari, and R. Chitra, “Classification of diabetes disease using support vector machine,” International Journal of Engineering Research and Applications, vol. 3, no. 2, pp. 1797-1801, 2013.

[6] K. Kayaer and T. Yildirim, “Medical Diagnosis on Pima Indian Diabetes Using General Regression Neural Networks,” International Conf. Artif. Neural Networks Neural Inf. Process., 2003, pp. 181–184.

[7] Bhoi, S.K., Panda, S.K., Jena, K.K., Abhisekha, P.A., Sahoo, K.S., Samee, N.U., Pradhan, S.S., & Sahoo, R.R. (2021). Prediction of Diabetes in Females of Pima Indian Heritage: A Complete Supervised Learning Approach. Journal of Medical Systems, 45(7), 81. https://doi.org/10.1007/s10916-021-01780-6

[8] Anwar, F., Qurat-Ul-Ain, Yasir Ejaz, M., & Moazzam, M. (2018). A comparative analysis on diagnosis of diabetes mellitus using different approaches – A survey. Journal of King Saud University-Computer and Information Sciences, 30(3), 330-345. https://doi.org/10.1016/j.jksuci.2017.06.006

[9] ElSeddawy AI, Karim FK, Hussein AM, Khafaga DS. Predictive Analysis of Diabetes-Risk with Class Imbalance. Comput Intell Neurosci. 2022 Oct 11;2022:3078025. doi: 10.1155/2022/3078025. PMID: 36268149; PMCID: PMC9578843.

GLOSSARY

- ***C:*** A hyperparameter in the SVM algorithm that controls the trade-off between maximizing the margin and minimizing the classification error.
- ***Confusion Matrix:*** A table used to evaluate the performance of a classification model, which shows the number of true positive, true negative, false positive, and false negative predictions.
- ***Cross-Validation:*** A technique used to assess the performance and generalization of a model by partitioning the data into training and validation sets, and repeatedly training and testing the model on different partitions.
- ***Generalisability:*** The ability of a model to perform well on unseen data from the same distribution as the training data.
- **Hyperparameter**: A parameter in a machine learning algorithm that is set before the training process, and controls the behavior of the algorithm.
- ***Learning Rate:*** A hyperparameter that controls the step size at each iteration during the optimization process in a neural network.
- ***MLP:*** A type of artificial neural network that is commonly used for classification and regression tasks, consisting of an input layer, one or more hidden layers, and an output layer.
- ***PCA:*** A dimensionality reduction technique that transforms a high-dimensional dataset into a lower-dimensional space while retaining most of the information.
- ***Precision:*** A performance metric that measures the proportion of true positive predictions over the total number of positive predictions made by a model.
- ***RBF:*** Radial Basis Function, a kernel function commonly used in SVMs for mapping the input data to a higher-dimensional feature space.
- ***Recall:*** A performance metric that measures the proportion of true positive predictions over the total number of actual positive instances in the dataset.
- ***ReLu:*** Rectified Linear Unit, a non-linear activation function commonly used in neural networks.
- ***SMOTE:*** Synthetic Minority Over-sampling Technique, a method for balancing imbalanced datasets by creating synthetic samples of the minority class.
- ***SoftMax:*** An activation function commonly used in the output layer of a neural network for multiclass classification tasks, which converts the output values into probabilities.
- ***SVM:*** Support Vector Machines, a type of machine learning algorithm commonly used for classification and regression tasks, which identifies a hyperplane in the feature space that maximally separates the data into different classes.
- ***Stratify:*** A sampling method that ensures the same proportion of each class in the dataset is present in the training and validation sets.

References:
[1] Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4). springer.
[2] Alpaydin, E. (2010). Introduction to machine learning (2nd ed.). Cambridge, MA: MIT Press.
[3] Chollet, F. (2018). Deep learning with Python. Manning Publications.
[4] Kelleher, J. D., & Tierney, B. (2018). Data science: An introduction. CRC press.

IMPLEMENTATION DETAILS:

- The functions used for training the models were as follows:
  - torch.nn
  - sklearn.svm.SVC
  - sklearn.model_selection.GridSearchCV
- The SMOTE technique was applied to the data using the imblearn sampling library.
- The SVM and MLP models, respectively, identified as optimal for use in testing : SVM_Best.joblib and MLP_Best.joblib
- The test sets to be used when evaluating the models: X_test.csv and y_test.csv
- The code has been tested in an environment running Python 3 (3.9.12) with the following packages installed: pandas, numpy, matplotlib, sklearn, seaborn, imblearn, joblib, time, skorch and torch.
- All data pre-processing was done separately : Data Preprocessing Final
- For MLP models, the dataset was converted from pandas DataFrame to tensor (torch.tensor) which is the only acceptable format of data in PyTorch.
- The optimiser used for MLP models was :stochastic gradient descent (torch.optim.SGD).
- Adam optimiser (torch.optim.Adam) was also tried, but did not give better results.
- To maintain the time, the hyperparameter was tuned in a way to maintain the same time for both the models.
- Values of number of Epochs and Batch size were also initially added in parameters of MLP, but it exceeded the time exponentially.
  But if time permits these can be added to find a more optimized model.