Classification of Diabetes Mellitus: A Comparative Analysis of various Machine Learning Algorithms

Esha Nadeem
University of Greater Manchester
Deane Road Bolton
BL3 5AB

Rameez Asif, PhD
University of Greater Manchester
Deane Road Bolton
BL3 5AB

Ali Raza
University of Greater Manchester
Dean Road Bolton
BL3 5AB

Adil Mustafa, PhD
ARU University
Bishop Hall Ln,
Chelmsford
CM1 1SQ

ABSTRACT
Diabetes Mellitus, a widespread metabolic disorder characterised by hyperglycaemia, poses a significant global health concern. Unfortunately, it is often diagnosed only after symptoms have developed, thereby reducing opportunities for early intervention. Traditional diagnostic methods primarily identify individuals who are already symptomatic, leaving many at-risk individuals undetected. By integrating machine learning models with Electronic Health Records, it may be possible to identify individuals at high risk during routine health assessments, enabling earlier intervention.

This study aims to compare different machine learning algorithms to determine whether an individual has diabetes. Several performance metrics are used for comparison, including precision, accuracy, F1-score, recall, and computational efficiency. The methodology begins with the pre-processing of the Pima Indians Diabetes Dataset, followed by the training, testing, and validation of four machine learning algorithms: Support Vector Machine, Decision Tree, Artificial Neural Network, and Random Forest, implemented using Python. The process is then repeated in MATLAB using an augmented dataset, and the resulting performances are compared.

The Random Forest algorithm achieved the highest prediction accuracy, with 75% accuracy on the original dataset and 100% accuracy on the augmented dataset. The findings suggest that machine learning approaches could support healthcare professionals in identifying individuals at risk of diabetes at an earlier stage, potentially enabling timely intervention and improving patient outcomes.

General Terms
Medical Diagnosis, Predictive Modelling, Predictive Modelling, Medical Data Mining.

Keywords
Artificial Intelligence; Machine Learning; Diabetes Mellitus; AI in Healthcare; Random Forest; Support Vector Machine; Artificial Neural Network, Decision Tree.

1. INTRODUCTION
Diabetes mellitus, known by the name hyperglycemia (meaning high blood glucose levels), is a metabolic disorder of great complexity that leads to serious health consequences in the affected individuals, and it's when the pancreas in the human body is unable to produce enough insulin, leading to high blood glucose [1]. According to the 11th edition of the International Diabetes Federation (2025), approximately 589 million individuals between the age of 20-70 years have Diabetes Mellitus. This figure is expected to jump around 853 million by 2050 if it's left uncontrolled [2]. This alarming condition underlines the necessity of having more effective diagnostic and predictive methods in place. Diabetes Mellitus exists in numerous types, namely Type 1, Type 2, Gestational, Monogenic, MODI, LADA, and secondary Diabetes. If managed improperly, Diabetes Mellitus descends into devastating conditions such as macrovascular complications that include coronary artery disease and stroke as well as microvascular complications namely diabetic retinopathy, neuropathy, and nephropathy [3,4]. However, catching the disease in the very early stages and combining that with appropriate treatment options against various types of diabetes can pave the way to reduce its risks and complications [5]. But for the fact that preventive measures may only be carried out when a person discovers that they may develop the disease soon, which is not the practical case currently. Most people get to know that they have developed diabetes either due to an indirect cause or after being checked for some direct symptoms or signs of disease. Not only that, but majority of cases also already breach the extreme conditions way well before the patient gets to know about it. The four traditional diagnostic tests for Diabetes recommended by the American Diabetes Association guidelines, are efficient enough in pinpointing disease occurrence however not efficient enough to pick the risk at its early stage. To deal with this condition, science is extending its arms in finding smart yet effective ways to pick diseases including Diabetes Mellitus from scratch where prevention is possible, and this can be achieved with the assistance of Machine learning models.

Machine learning techniques employing Artificial Intelligence can offer healthcare professionals with a predictive approach in assessing individuals who are at risk of Diabetes Mellitus prior to the appearance of signs or/and symptoms. By assembling machine learning models with Electronic Health Records, it can be made possible to pinpoint individuals at high risk via their routine check-ups, offering earlier intervention [6-8]. These models, with the use of patient data, highlight individuals who have elevated risk levels even when they express no noticeable symptoms, offering early medical assistance and intervention as shown in **Figure 1**.

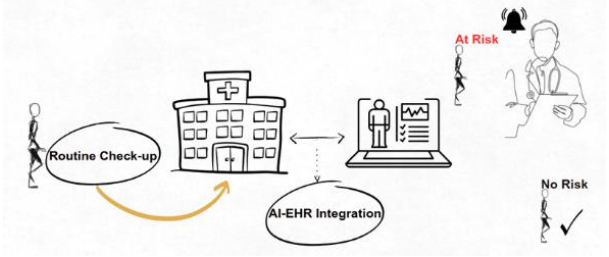


Figure 1. Integration of AI with EHR-Pinpointing individuals at risk of DM

This study involved the use of four well-known machine learning algorithms to compare their predictive analytics for Diabetes detection. Evaluation was conducted using SVM, ANN, DT, and RF. Predictions were first obtained using PIMA INDIANS dataset comprising of 768 records. For training and testing the predictive model, 8 characteristics causing diabetes were picked. Later, with the use of MATLAB and incorporating the Guassian Noice technique, the same dataset was increased to gear up the sample set size to half a million. Based on the results obtained from model performances, RF presented highest accuracy in determining whether a person has Diabetes or not. Random Forest providing 75% and 100% accuracy with original and augmented dataset respectively, gave highest prediction accuracy for the occurrence of Diabetes compared to other three algorithms tested in this study. For the assessment of model performances mainly the robustness, one of the performing models was subjected to two quick and simple tests, named "Flipping test" and "Prediction test". Hence, the outcome of this study can facilitate health care systems to diagnose Diabetes Mellitus at its early stage and to make better and accurate clinical decisions about disease prevention and management, potentially proved to be a lifesaving intervention.

2. MATERIALS AND METHODS

A strategic methodology was implied and carried out in two distinct rounds of analysis, starting out with **Dataset selection**.

2.1 First-round analysis

In first-round analysis, The PIDD extracted as a CSV file from UCI Repository, was used (originally collected by NIDDK, division of the NIH, USA). This dataset represents medical plus demographic characteristics (768 Pima Indian heritage women around the age of 21 years) [9], as described in **Table 1**

Table 1. Feature Description of Dataset

Feature	Description
Pregnancies 6	No. of times an individual has been pregnant.
Glucose	Plasma glucose concentration measured 2 hours post OGTT.
Blood Pressure	Diastolic blood pressure in mm Hg.
Skin Thickness	Triceps-skin thickness in mm.
Insulin	2-hour serum insulin level in mu U/ml.
BMI	Defines Body Mass Index & the unit is kg/m².
Diabetes.Pedigree Function	Likelihood of diabetes with respect to family history.
Age	Patient's age (years).
Outcome	Outcome variable (0 states non-diabetic & 1 states Diabetic).

2.2 Second-round analysis

Class bias was present in this dataset as there were up to 34.9% diabetic and 65.1% non-diabetic subjects. Hence to address this, in second-round analysis, the previous Dataset was augmented to attain a 50-50 class balance using Random Sampling by creating Gaussian Noise [10]. MATLAB was used to scale up the data to half a million samples.

2.3 Data pre-processing

Both datasets underwent 3 data preprocessing stages as described in ***Figure 2***: Outcome Distribution, Handling Missing Data (missing values replaced with 0, np.nan, and imputed via mean, median, or mode), then Outliers identification & visualization.

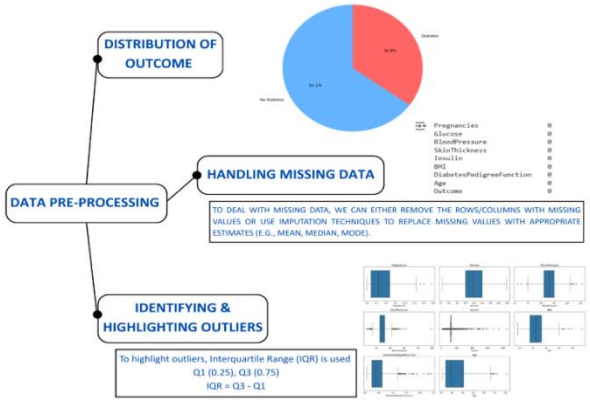


Figure 2. Breakdown of Pre-processing steps

2.4 Distribution into training & testing set

The train_test_split function from the model selection module of sklearn was used for the division of processed data into training and testing sets.

2.5 Model Development

The main objective of preparing the dataset is to implement the machine learning algorithms on that data for making predictions to detect if a person has chances of suffering from Diabetes in future or not. Various well-known Machine learning algorithms, namely DT, RF, ANN, & SVM were implemented in this study.

2.5.1 Decision Tree

This algorithm implied in the study uses a basic tree structure to classify Diabetes status of an individual based on certain variables (BMI, age and glucose levels etc.). The tree structure has 3 levels namely root, child nodes, and leaf nodes. Root (based on their association with target-Diabetic or Non-diabetic) decides the optimal data split. Then child nodes act in determining the best splits using Information Gain (IG) [11]. Finally, the classification for either Diabetic or Non-diabetic is presented by leaf node.

IG is given as:

$$IG(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|\,S_v\,|}{|\,S\,|} Entropy(S_v)$$

Where: $IG(S,A)$ information gain of A on dataset S, $Entropy(S)$ of original dataset (S), $Values(A)$ set of possible values of A, S_v subset of S,$|\,S_v\,|/|\,S\,|$ proportion of subset S_v in S, $Entropy(S_v)$ entropy of subset S_v

Entropy is calculated as:

$$Entropy(S) = -\sum_{i=1}^{C} p_{iLog_2(p_i)}$$

Where: S dataset, c no. of classes, p_i proportion of instance that belong to class *i.*

2.5.2 Random Forest

The Random Forest algorithm uses a clustering approach that contributes to an improved classification/prediction accuracy by pairing multi decision trees. The entire process entails taking multiple random subsets from the entire dataset to create a total of k different decision trees. Each tree has a role in deciding a classification prediction (either "Diabetic" or "Non-diabetic"). The final prediction is made by a collective vote from each tree present within the forest [11].

2.5.3 Support Vector Machine

This algorithm makes classification prediction for whether a case is Diabetic or not by pinpointing the most appropriate hyperplane. SVM adjusts the distance of margin between the support vectors which are the points proximal to hyperplane.
The SVM optimization problem can be formulated as:

$$\min(w, b)\ \frac{1}{2} \parallel w \parallel 2$$

Determined by:

$$yi(w \cdot xi + b) \geq 1, \forall i$$

Where:w defines weight vector, b defines bias term, yi defines class tag, xi defines feature vector for *i-th* datapoint SVM is considered ideal for organizing intricate medical data because it is efficient in dealing with linear as well as non-linear boundaries [12].

2.5.4 Artificial Neural Network

The ANN model implied in this study operates with certain data patterns in predicting the risks connected to Diabetes. The architecture of ANN is built on multiple layers interconnecting with different nodes which are also known as neurons. Medical data features e.g. age and blood glucose and BMI measurements are given to input layer which are modified by hidden layers with the use of activation functions and weighted summations, such as the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

where: $f(x)$ defines function output, x defines Input to neuron, $e - x$ defines exponential function [13].

2.6 Analyzing Tool: Python and associated Libraries

This study implied Python as its main programming language for data analysis and machine learning model implementation. Python supports a vast collection of libraries as depicted in ***Table 2*** which can help with data manipulation and machine learning implementations [14,15]. Google Colab and its GPU was the medium used for efficient model training plus straightforward code implementation without incurring costs [16,17].

Table 2. List of Libraries used for Dataset Manipulation

Domain	Liberty used	Purpose
Data Handling & Pre-processing	NumPy	NumPy was used for performing numerical operations e.g., computing the zones in outliers etc [18].
	Pandas	Pandas was used for handling missing values, pointing outliers, mainly cleaning the data [18].
Data Visualization	Matplotlib	Matplotlib was used to design plots to visualize the distribution of data [19].
	Seaborn	Seaborn was used for visualizing data statistics and correlations [20].
Machine Learning Models	Scikit learn	Scikit learn assisted in the implementation of Machine Learning algorithms and their performance evaluation [21].
Deep Learning	Keras/Tensorflow	These libraries were utilized to develop and train ANN, mainly to optimize complexity [22].

2.7 Performance Evaluation

The final step of prediction model was to evaluate the results using different evaluation metrics which were derived from the outcome of confusion matrix. Confusion matrix being the crucial classification prediction tool, provided TP, TN, FP, and FN instances from the performance of all models which became the basis for the evaluation process involving accuracy, precision, F1 score, and Recall [23].

The structure of Confusion matrix is given as:

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

The confusion matrix was used to derive the following performance metrices:

Accuracy: Taking the proportion of correct predictions as true positives and true negatives, from overall predictions made and is given as:

$$Accuracy = \frac{TP + TN}{FN + FP + TN + TP}$$

Precision: Taking the proportion of true positive predictions across all the positive predictions given as:

$$Precision = \frac{TP}{FP + TP}$$

Recall: (which is also known as sensitivity) Measured by taking the proportion of true positive predictions provided by the model given as:

$$Recall = \frac{TP}{FN+TP}$$

F1-score: Calculated to establish a balance between the outcome of recall and precision, particularly with unequal class distribution cases given as:

$$F1(score) = 2 \times \frac{Recall \times Precisison}{Recall + Precision}$$

AUC-ROC: Used to assess model's capacity in distinguishing across the classes, providing an overview of the model's performance over all the potential classification models given as [24,25]:

$$AUC = \int_0^1 True\ Positive\ Rate\ d(False\ Positive\ Rate)$$

3. RESULTS

3.1 Outcome from the Confusion matrix

Outcome from the Confusion matrix of each model was obtained and correlated for PIMA Indians Diabetes Dataset (FRA) as well as augmented dataset (SRA) (*Appendix A1*). Of all other models implied in this study, Random Forest outperformed attaining 23 & 0 false negatives, 16 & 0 false positives, 84 & 50027 true negatives, and 31 & 49973 true positives respectively in FRA and SRA.

Derived from the results of confusion matrix, Random Forest gained 75% accuracy in FRA and 100% ideal accuracy in SRA, reflecting highest accuracy throughout both the rounds of analysis, followed by the accuracy of DT as 66% in FRA & 95% in SRA, SVM's accuracy as 70% in FRA & 75% in SRA, and then ANN's accuracy as 73% in FRA & 65% in SRA respectively as depicted in ***Table 3***.

Table 3. Comparative Analysis of Model Performance (Before & After Data Expansion)

Model Used	Accuracy (Before/After)	Precision (Before/After) Class 0/ Class 1	Recall (Before/After) Class 0/ Class 1
Random Forest	0.75 1.0	0.79/0.67 0.1.0/1.0	0.84/0.58 1.0/1.0
Support Vector Machine	0.70 0.75	0.75/0.59 0.74/0.77	0.82/0.48 0.79/0.72
Decision Tree	0.66 0.95	0.72/0.52 0.98/0.93	0.78/0.44 0.93/0.98
Artificial Neural Network	0.73 0.65	N/A 0.82/0.61	N/A 0.41/0.91

3.2 Comparison of accuracy

The following ***Figure 3*** of the bar chart illustrates accuracy performance of each model, with Random Forest attaining the highest accuracy (*Appendix B*).

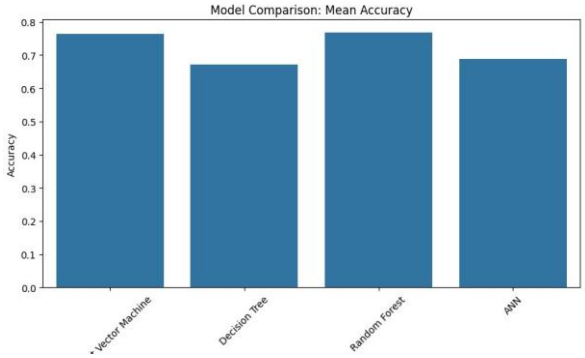


Figure 3. Performance comparison of each Model against Accuracy

3.3 Random Forest-Optimization + Validation

To optimize and validate the outcome of RF, a **fivefold cross validation** was implied [26], and the average score turned out to be an approximate 0.7618. Later after that, **hyperparameter tuning** involving GridSearchCV was used [27] (*Appendix A1, A2*). With an optimal cross-validation score 0.91 RF indicated strong accuracy levels in terms of its performance. Moreover, RF with a **ROC-AUC score** 0.81 reflected reasonably good classification prediction ability as shown in ***Figure 4***.

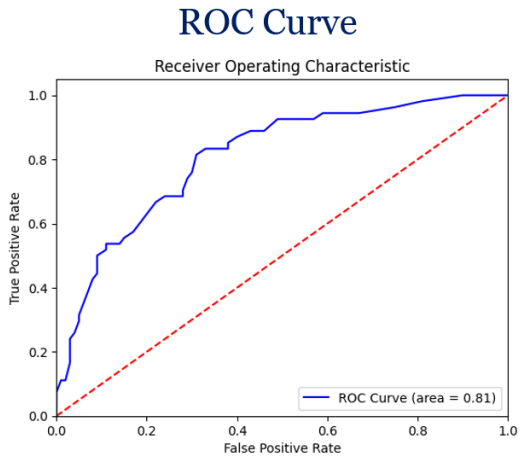


Figure 4. ROC-AUC Grapgh

3.3.1 Flipping Test

Apart from optimization & validation, performance (mainly robustness and consistency) of one of the models was assessed using the **Flipping Test.** This was done by choosing two random diabetic entities from the dataset and flipping their feature readings. The model identified both the actual and the flipped entities as diabetic as depicted in ***Figure 5***.

```
Original diabetic participant's features:
 Pregnancies                 6.000
Glucose                    148.000
BloodPressure               72.000
SkinThickness               35.000
Insulin                      0.000
BMI                         33.600
DiabetesPedigreeFunction     0.627
Age                         50.000
Name: 0, dtype: float64
Features to flip with:
 Pregnancies                 8.000
Glucose                    183.000
BloodPressure               64.000
SkinThickness                0.000
Insulin                      0.000
BMI                         23.300
DiabetesPedigreeFunction     0.672
Age                         32.000
Name: 2, dtype: float64
Original prediction for known diabetic: Diabetic
Flipped prediction for similar features: Diabetic
Model is consistent in predictions after flipping.
```

Figure 5. Results from Flipping Test

3.3.2 Prediction Test

For further validation of RF performance, **Prediction Test** was carried out where two presumed diabetic and non-diabetic cases with their feature values were given to the random forest without exposing the outcome of both cases. Random Forest model accurately predicted the outcome of both cases, depicting its efficacy in predicting Diabetes cases as depicted in ***Figure 6***.

```
Case 1 Prediction: Diabetic
Case 2 Prediction: Non-Diabetic
/usr/local/lib/python3.11/dist-packages/skle
```

Figure 6. Results from Prediction test

4. DISCUSSION

Early detection of Diabetes Mellitus is currently one of the most crucial, real-world concerns. According to the latest IDF Diabetes Atlas (2025), 1 in 9 adults of the 20-79 years age group are suffering from Diabetes Mellitus which is unfortunately representation of about 11.1% of the total adult population. Not only that, 4 in 10 adults out of this figure, suffering from DM are not aware of the condition [2]. This growing figure is a matter of serious concern for healthcare professionals. However, an early diagnosis strategy along with appropriate preventive measures and treatment regime against various types of DM can help minimize its risks and complications [5]. The aim of this study was to do a comparative analysis of the performance of four well known ML algorithms (RF, SVM, DT, and ANN) in accurately classifying Diabetes Mellitus at an early stage. Various performance metrics (accuracy, precision, recall, F1-score) along with computational efficiency assessed the models over the actual and the expanded PIMA Indians Diabetes Dataset.
The results obtained from confusion matrix of the models in FRA depicted that Random Forest having accuracy score of 75% outperformed the remaining algorithms, followed by Artificial Neural Network (73%), Support Vector Machine (70%), and Decision Tree (66%). Random Forest proved attained highest accuracy for predictive analysis in classifying Diabetes cases, possibly due to the ability to reduce overfitting as well as the ensemble behavior of model [28]. Previous investigations show that class distribution and size of dataset used in training and testing the models significantly impact model performance and adaptability [29]. To cater for these limitations, the next stage of this study (second round-analysis) was to augment the dataset by adding Gaussian Noice, which considerably improved model performances, specifically improving the performance of DT and RF. Random Forest showed perfect classification results (100% accuracy, F1-scoreprecision, and recall), depicting its adaptability when trained and tested on a diverse dataset. This result aligns with the study [28], which mentioned the same results for Random Forest's excellent performance in disease diagnosis involving complicated feature interactions. SVM showed better performance post-expansion, attaining 75% accuracy, though it had a poor score in recall for Class 1 cases (Diabetic cases). This result is consistent with the study [29]. After dataset expansion (in SRA) ANN experienced a fall in accuracy from 73% to 65% [30].

For optimization and further validation of RF model, 5-fold cross validation was implied giving an average score of 0.7618. For hyperparameter optimization, GridSearchCV was implied, which elevated Random Forest's accuracy to 0.91, which is also a highlight of how important model tuning is, in enhancing model's performance [26]. An AUC above 0.8 is considered excellent [31]. Random Forest's ROC score of 0.81 assessed in this study demonstrated its exceptional ability in discriminating to declare a certain case as diabetic or non-diabetic. Towards the end of this study, flipping and prediction tests was carried out to test model consistency. In the flipping test, feature values were swapped between two diabetic cases. Whereas in the prediction test, features of two assumed cases (one Diabetic and other non-Diabetic) were given to one of the models to make a prediction. Models passed both tests. As stated in a study [32], interpretability and consistency hold a great value in clinical settings but must be paired with strong predictive accuracy.

Note that RF can produce highly accurate outcomes however, it can end up with targets that are either skewed or poorly calibrated/distributed. RF being an ensemble model of DT,

which means it makes classification predictions because of tree fractions picking from every class. This functions well for classification however, each tree separately voting in "all or none" manner can sometimes result in probabilities that are not properly calibrated. Which can be portrayed as a classification prediction to be around 0 or 1 while the actual outcome might be around 0.7, 0.8 etc., leading to overconfidence in class predictions. To cater for the situation, this study suggests post-processing approaches which can be implied, such as **Isotonic Regression** or **Plot Scaling** which establish` a connection with the original predictions to more calibrated ones via held up validations.

5. CONCLUSIONS

Predictive analytics with the assistance of Machine Learning can alter the ways for researchers and medical practitioners in healthcare to gain insights from medical records to make accurate decisions [33]. This study utilized four well known ML algorithms (SVM, ANN, DT, and RF) and compared the outcomes for predictive analysis of DM. these algorithms were implied first PIMA Indian dataset which is comprised of 768 records with 8 Diabetes causing factors to train and test the predictive models. In the second phase of study, this dataset was expanded to raise the sample size to half a million by implying Guassian Noice in MATLAB. Results showed that RF had the highest accuracy in precisely classifying whether a person has diabetes or not. RF had 75% and 100% accuracy respectively with original PIMA Indian Diabetes dataset and the augmented data. This was ranked with the highest score compared to the remaining three algorithms analyzed in this study. Hence this study came to a conclusion that RF model is accurate in predicting the early risk of Diabetes. So, the findings of this study could contribute to supporting healthcare by pinpointing Diabetic cases at an early stage to make informed and accurate preventive and managing decisions, certainly proved to be a lifesaving tool.

6. REFERENCES

[1] World Health Organization (WHO), 2024. Diabetes: Fact Sheet. Available at: https://www.who.int/news-room/fact-sheets/detail/diabetes [Accessed 12 January 2025].

[2] International Diabetes Federation (IDF), 2024. IDF WHA 2024 Event. Available at: https://idf.org/media/uploads/2023/04/IDF-WHA-2024-event.pdf [Accessed 12 January 2025].

[3] American Diabetes Association (ADA), 2024. Classification and diagnosis of diabetes: Standards of medical care in diabetes—2024. Diabetes Care, 47(Supplement 1), pp. S12–S19. Available at: https://diabetes.org [Accessed 12 January 2025].

[4] International Diabetes Federation (IDF), 2024. IDF WHA 2024 Event. Available at: https://idf.org/media/uploads/2023/04/IDF-WHA-2024-event.pdf [Accessed 12 January 2025].

[5] Feleke, S.F., Dessie, A.M., Getachew, Z.D., Bizuneh, F.K., Kidie, A.A., Yayeh, B.M., Ayal, B.G. and Tesfa, N.A., 2025. Burden of diabetic ketoacidosis and its predictors among diabetic patients in Ethiopia: Systematic review and meta-analysis. PloS one, 20(1), p.e0309097.

[6] Assadi, A., Laussen, P.C., Goodwin, A.J., Goodfellow, S., Dixon, W., Greer, R.W., Jegatheeswaran, A., Singh, D., McCradden, M., Gallant, S.N. and Goldenberg, A., 2022. An integration engineering framework for machine learning in healthcare. Frontiers in Digital Health, 4, p.932411.

[7] Aliferis, C. and Simon, G., 2024. Artificial Intelligence (AI) and Machine Learning (ML) for healthcare and health sciences: The need for best practices enabling trust in AI and ML. In: Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls. Cham: Springer International Publishing, pp.1–31.

[8] Nichols, J.A., Chan, H.W.H. & Baker, M.A.B., 2018. Machine learning: applications of artificial intelligence to imaging and diagnosis. Biophysical Reviews, 11(1), pp.111-118. Available at: https://doi.org/10.1007/s12551-018-0449-9

[9] UCI Machine Learning Repository, 2017. Pima Indians Diabetes Database. [online] Kaggle. Available at: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[10] García-Ordás, M.T., Benavides, C., Benítez-Andrades, J.A., Alaiz-Moretón, H. and García-Rodríguez, I., 2021. Diabetes detection using deep learning techniques with oversampling and feature augmentation. Computer Methods and Programs in Biomedicine, 202, p.105968.

[11] Nai-aruna, N. and Moungmaia, R., 2015. Comparison of Classifiers for the Risk of Diabetes Prediction. Procedia Computer Science, 69, pp. 132–142. Available at: http://www.sciencedirect.com [Accessed 13 March 2025]. https://doi.org/10.1016/j.procs.2015.10.014

[12] Yu, W., Liu, T., Valdez, R., Gwinn, M. and Khoury, M.J., 2010. Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. BMC Medical Informatics and Decision Making, 10, p. 16. Available at: http://www.biomedcentral.com/1472-6947/10/16

[13] Sisodia, D. and Sisodia, D.S., 2018. Prediction of Diabetes using Classification Algorithms. Procedia Computer Science, 132, pp. 1578–1585. Available at: https://www.sciencedirect.com [Accessed 13 March 2025]. https://doi.org/10.1016/j.procs.2018.05.122

[14] Badawy, M., Ramadan, N. and Hefny, H.A., 2023. Healthcare predictive analytics using machine learning and deep learning techniques: a survey. Journal of Electrical Systems and Information Technology, 10(1), p.40.

[15] Mahabub, A., 2019. A robust voting approach for diabetes prediction using traditional machine learning techniques. SN Applied Sciences, 1(12), p.1667.

[16] Carneiro, T., Da Nóbrega, R.V.M., Nepomuceno, T., Bian, G.B., De Albuquerque, V.H.C. and Reboucas Filho, P.P., 2018. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. Ieee Access, 6, pp.61677-61685.

[17] Fermo, I.R., Flores, F.C. and Andrade, C.M.G., 2021. Data Augmentation Technique in Neural Network Training. Brazilian Journal of Experimental Design, Data Analysis and Inferential Statistics, 1(1), pp.69-76.

[18] Faraz, S. and Singh, P., 2022. Diabetes Prediction using Machine Learning. Journal of Applied Science and Education, 2(02), pp.1-12.

[19] Tosi, S., 2009. Matplotlib for Python developers. Packt Publishing Ltd.

[20] Waskom, M.L., 2021. Seaborn: statistical data visualization. Journal of Open Source Software, 6(60), p.3021.

[21] Sai Madhuri, K., Prathusha, B., Manish Reddy, G., Deshmukh, G., Bhuvan Sai Kumar, K. and Markos, A., 2023, June. Performance Analysis of Machine Learning Algorithms by Using WEKA and Scikit-Learn. In International Conference on Data Science and Big Data Analysis (pp. 633-644). Singapore: Springer Nature Singapore.

[22] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. {TensorFlow}: a system for {Large-Scale} machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).

[23] Mujumdar, A. and Vaidehi, V., 2019. Diabetes prediction using machine learning algorithms. Procedia Computer Science, 165, pp.292-299.

[24] Sathyanarayanan, S. and Tantri, B.R., 2024. Confusion matrix-based performance evaluation metrics. African Journal of Biomedical Research, pp.4023-4031.

[25] Jaskowiak, P.A. and Costa, I.G., 2023. Clustering validation with the area under precision-recall curves. arXiv preprint arXiv:2304.01450.

[26] Nti, I.K., Nyarko-Boateng, O. and Aning, J., 2021. Performance of machine learning algorithms with different K values in K-fold cross-validation. International Journal of Information Technology and Computer Science, 13(6), pp.61-71.

[27] Hossain, M.R. and Timmer, D., 2021. Machine learning model optimization with hyper parameter tuning approach. Glob. J. Comput. Sci. Technol. D Neural Artif. Intell, 21(2), p.31.

[28] Noviyanti, C.N. and Alamsyah, A., 2024. Early detection of diabetes using random forest algorithm. Journal of Information System Exploration and Research, 2(1).

[29] Chowdhury, M.M., Ayon, R.S. and Hossain, M.S., 2024. An investigation of machine learning algorithms and data augmentation techniques for diabetes diagnosis using class imbalanced BRFSS dataset. Healthcare Analytics, 5, p.100297.

[30] Ekong, A., Attih, I., James, G. and Edet, U., 2024. Effective classification of diabetes mellitus using support vector machine algorithm. Researchers Journal of Science and Technology, 4(2), pp.18-34.

[31] Shwartz-Ziv, R. and Armon, A., 2022. Tabular data: Deep learning is not all you need. Information Fusion, 81, pp.84-90.

[32] Li, J., 2024. Area under the ROC Curve has the most consistent evaluation for binary classification. PLoS One, 19(12), p.e0316019.

[33] Shickel, B., Tighe, P.J., Bihorac, A. and Rashidi, P., 2017. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE journal of biomedical and health informatics, 22(5), pp.1589-1604.

[34] Sarwar, M.A., Kamal, N., Hamid, W. and Shah, M.A., 2018, September. Prediction of diabetes using machine learning algorithms in healthcare. In 2018 24th international conference on automation and computing (ICAC) (pp. 1-6). IEEE

Data Availability Statement: The PIMA Indians Diabetes Dataset used in this study was extracted as a CSV file from UCI Repository which is available online Pima Indians Diabetes Database. The expanded PIDD used in this study is available on MATLAB.

Acknowledgments: The author would like to express sincere gratitude to everyone who provided support throughout the course of this study. The author is grateful to the Creator for bestowing the wisdom, strength, and perseverance required to complete this project. Heartfelt thanks are also extended to the author's parents and friends for their continued trust and support. Special thanks are extended to the supervisors, **Dr Rameez Asif and Dr Adil Mustafa**, for their continuous guidance, cooperation, and support throughout the development and completion of this project. The author would also like to thank the **University of Greater Manchester** for providing the necessary resources and facilities to support this work.

Conflicts of Interest: The authors declare no conflicts of interest.

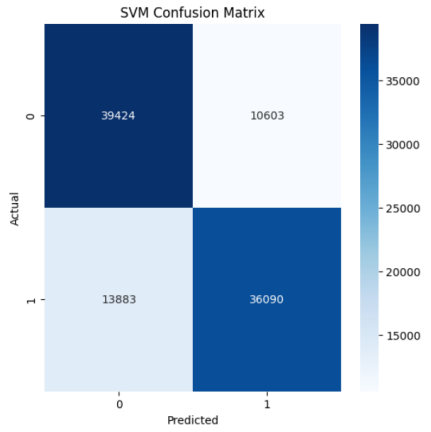
Abbreviations

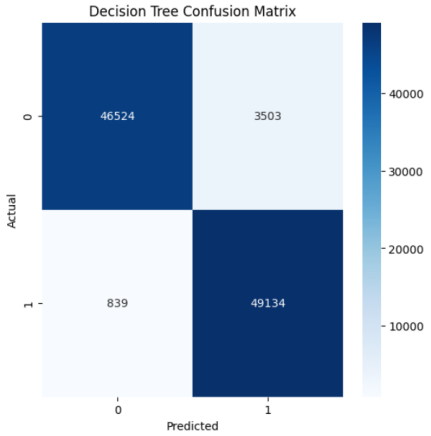
The following abbreviations are used in this manuscript:

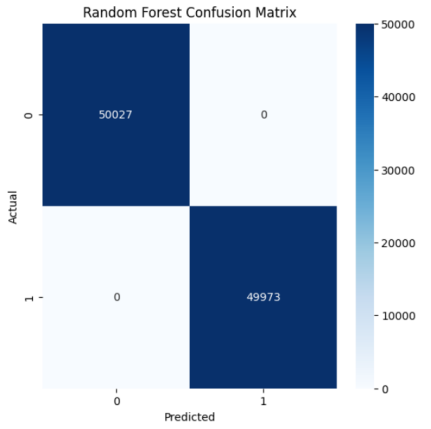
IDF	International Diabetes Federation
DM	Diabetes Mellitus
ML	Machine Learning
AI	Artificial Intelligence
ROC-AUC	Receiver Operating Characteristic-Area Under the Curve
FRA	First Round Analysis
SRA	Second Round Analysis
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
GPU	Graphics Processing Unit
IG	Information Gain
ANN	Artificial Neural Network
SVM	Support Vector Machine
RF	Random Forest
DT	Decision Tree
OGTT	Oral Glucose Tolerance Test
NIDDK	National Institute of Diabetes and Digestive and Kideny
CSV	Comma Separated Values
PIDD	PIMA Indians Diabetes Dataset
MATLAB	Matrix Laboratory
LADA	Latent Autoimmune Diabetes in Adults
MODY	Maturity Onset of Diabetes in the Young

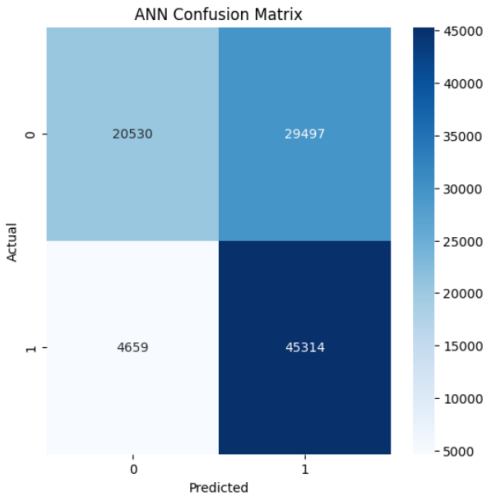
Appendix A

Results from Confusion Matrix post Dataset Expansion









Appendix B

Results from Model Performances

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.84      0.81       100
           1       0.66      0.57      0.61        54

    accuracy                           0.75       154
   macro avg       0.72      0.71      0.71       154
weighted avg       0.74      0.75      0.74       154

Confusion Matrix:
 [[84 16]
 [23 31]]
```

```
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.78      0.75       100
           1       0.52      0.44      0.48        54

    accuracy                           0.66       154
   macro avg       0.62      0.61      0.61       154
weighted avg       0.65      0.66      0.66       154

Confusion Matrix:
 [[78 22]
 [30 24]]
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.6917 - loss: 0.7183
ANN Model Accuracy: 0.69
```

```
ANN Accuracy: 0.66885
ANN Classification Report:
              precision    recall  f1-score   support

         0.0       0.63      0.80      0.71     50027
         1.0       0.73      0.54      0.62     49973

    accuracy                           0.67    100000
   macro avg       0.68      0.67      0.66    100000
weighted avg       0.68      0.67      0.66    100000
```

```
SVM Accuracy: 0.75516
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.79      0.76     50027
           1       0.77      0.72      0.75     49973

    accuracy                           0.76    100000
   macro avg       0.76      0.76      0.75    100000
weighted avg       0.76      0.76      0.75    100000
```

```
Decision Tree Accuracy: 0.95658
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.96     50027
           1       0.93      0.98      0.96     49973

    accuracy                           0.96    100000
   macro avg       0.96      0.96      0.96    100000
weighted avg       0.96      0.96      0.96    100000
```

```
Random Forest Accuracy: 1.0
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     50027
           1       1.00      1.00      1.00     49973

    accuracy                           1.00    100000
   macro avg       1.00      1.00      1.00    100000
weighted avg       1.00      1.00      1.00    100000
```

Appendix C

Appendix A.1 Cross Validation of RF

```
[ ] from sklearn.model_selection import cross_val_score

    # Example: Cross-validation for Random Forest
    rf_model = RandomForestClassifier(random_state=42)
    cv_scores = cross_val_score(rf_model, X, y, cv=5)  # 5-fold cross-validation
    print(f"Random Forest CV Scores: {cv_scores}")
    print(f"Mean CV Score: {cv_scores.mean()}")
```

```
Random Forest CV Scores: [0.74025974 0.74675325 0.75974026 0.83660131 0.75163399]
Mean CV Score: 0.7669977081741788
```

Appendix A.2 Hyperparameter Tuning

```
from sklearn.model_selection import GridSearchCV

# Set up the parameter grid for Random Forest
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10]
}

# Initialize GridSearchCV
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5, n_jobs=-1, scoring='accuracy')
grid_search.fit(X_train, y_train)

# Best parameters and best score
print("Best Parameters:", grid_search.best_params_)
print("Best CV Score:", grid_search.best_score_)
```

```
Best Parameters: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}
Best CV Score: 0.7768892443022791
```