VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

HEALTHCARE ANALYTICS - CSE4068

FINAL REPORT

DIABETES PREDICTION USING ENSEMBLE TECHNIQUES

Team Members

LOKESH KANNA - 19MIA1014

R YUVASHREE - 19MIA1053

K NIHARIKA SAMYUKTHA - 19MIA1083

M Tech CSE with Specialization in Business Analytics

Submitted to

Dr. Raja Sree T

Assistant Professor Senior Grade 2

School of Computer Science and Engineering

ABSTRACT

Diabetes is a chronic metabolic condition brought on by elevated blood glucose levels. Diabetes is considered to be one of the most common, severe, and deadliest diseases as it can lead to various complications and diseases such as renal disease, kidney problems, heart diseases, blindness, and many more. Diabetes is linked to a number of problems as well as a high morbidity and mortality rate. Early detection is extremely important for its timely treatment. Hence, there is a need to design and develop a model that can easily predict diabetes in patients. For the past two decades, there has been an exponential growth of medical data from digital devices, which has helped researchers study the impact of data analysis in medicine. Many previous studies discuss the implementation of machine learning techniques for predicting a person's diabetes considering significant features.

The main aim of this study is to improve the accuracy of diabetes mellitus prediction by utilizing various machine learning techniques, including ensemble methods such as Stacking, Hard Voting, and Soft Voting, with base classifiers like AdaBoost, Logistic Regression, Random Forest, Gradient Boost, Linear Discriminant Analysis, Extra Trees, and Cat Boost. For this experimentation, we will be using the Pima Indians Diabetes dataset, which gathers details on patients with and without diabetes, to construct and evaluate each model before selecting the optimal ensemble model to address this issue. The best performing model was the ensemble model using soft voting. However, the model had a high bias and low variance, which was addressed by calibration. The final model achieved an accuracy of 93.75%, precision of 95.24%, recall of 86.96%, and an F1 score of 90.91%. This study highlights the potential of machine learning techniques for predicting diabetes and the importance of calibration to improve model performance.

INTRODUCTION

Diabetes is a prevalent disease that affects a large number of people worldwide. According to the World Health Organization, approximately 422 million people were living with diabetes in 2014, and this number is projected to increase to 629 million by 2045. This highlights the importance of raising awareness about the disease, as it affects a significant portion of the population. Machine learning has become a powerful tool for predicting and diagnosing various diseases, including diabetes. One of the main advantages of using machine learning in diabetes prediction is that it can take into account a wide range of patient data, including demographic, clinical, and laboratory data, which can be difficult for healthcare professionals to analyze manually. Machine learning models can analyze this data quickly and accurately, providing healthcare professionals with insights that can aid in the early detection and treatment of diabetes.

Machine learning techniques such as logistic regression, decision trees, random forests, gradient boosting, and neural networks have been used extensively in diabetes prediction. These models can identify key features that are associated with diabetes, such as age, BMI, glucose levels, insulin resistance, and family history. Once these features are identified, the model can use them to predict the likelihood of a patient having diabetes.

Ensemble methods such as Stacking, Hard Voting, and Soft Voting methods combine the predictions of multiple machine learning models to produce a more accurate and reliable prediction. By combining the strengths of different models, ensemble methods can overcome the limitations of individual models and provide a more comprehensive prediction. Overall, machine learning has shown great potential in predicting diabetes and can be an effective tool for early detection and treatment. With further research and development, machine learning models can be integrated into clinical practice to improve patient outcomes and reduce the burden of diabetes.

LITERATURE SURVEY

1. Predicting Diabetes Mellitus with Machine Learning Techniques – 2018

Quan Zou, Kaiyang Qu, Yamei Luo, Dehui Yin, Ying Ju, and Hua Tang (2018), have tried building a model for predicting Diabetes Mellitus. They did this study using a decision tree, random forest, and neural network by implementing these on the dataset collected from a hospital in Luzhou, China. It's the hospital Physical Examination data which has 14 attributes in it. Principal component analysis (PCA) and minimum redundancy maximum relevance (mRMR) was used to reduce the dimensionality. By randomly selecting 68994 healthy people and diabetic patients' data, they prepared the training set. Due to the data unbalance, randomly extracted 5 times data also. The results showed that prediction with random forest could reach the highest accuracy (0.8084) when all the attributes were used.

2. Diabetes Prediction Using Ensemble of Different Machine Learning Classifiers - 2020

Md. Kamrul Hasan, Md. Ashraful Alam, Dola Das, Eklas Hossain, (Senior Member, IEEE), and Mahmudul Hasan (2020), had proposed a robust framework for diabetes prediction where the outlier rejection, filling the missing values, data standardization, feature selection, K-fold cross-validation, and different Machine Learning (ML) classifiers like k-nearest Neighbor, Decision Trees, Random Forest, AdaBoost, Naive Bayes, and XGBoost and Multilayer Perceptron (MLP) were employed. The weighted ensemble of different ML models is also proposed, to improve the prediction of diabetes where the weights are estimated from the corresponding Area Under the ROC Curve (AUC) of the ML model. AUC is chosen as the performance metric, which is then maximized during hyperparameter tuning using the grid search technique. All the experiments in this literature were conducted under the same experimental conditions using the Pima Indian Diabetes Dataset. As the result, after all the extensive experiments done, this ensemble classifier is the best-performing classifier with the sensitivity of 0.789, specificity of 0.934, false omission rate of 0.092, a diagnostic odds ratio of 66.234, an AUC of 0.950 which outperforms the state-of-the-art results by 2.00 % in AUC.

3. A review on current advances in machine learning based diabetes prediction - 2021

Varun Jaiswal, AnjliNegi, TarunPal (2021), had worked on with Machine learning algorithms (such as ANN, SVM, Naive Bayes, PLS-DA and deep learning) and data mining techniques are used for detecting interesting patterns for diagnosing and treatment of disease. This paper is an effort to summarize most of the literature concerned with machine learning and data mining techniques applied for the prediction of diabetes and associated challenges. This report would be a helping tool for better prediction of disease, improvement in understanding the pattern of diabetes, and also help for treatment and risk reduction of other complications of diabetes.

4. Predictive Methodology for Diabetic Data Analysis in Big Data - 2015

N.M. Saravanakumar Dr, T.Eswari, P. Sampath, and S.Lavanya.(2015), have started working on this due to their understanding of the need to develop data analytics. Because Diabetic Mellitus (DM) is one of the Non-Communicable Diseases (NCD), which has major health hazards in developing countries such as India. And they have used the predictive analysis algorithm in the Hadoop/Map Reduce environment to predict the diabetes types prevalent, the complications associated with it and the type of treatment to be provided. Based on the analysis, this system has provided an efficient way to cure and care for patients with better outcomes like affordability and availability.

5. A model for early prediction of diabetes - 2019

TalhaMahboob Alam, Muhammad Atif Iqbal, YasirAli, Abdul Wahab, SafdarIjaz, TalhaImtiaz Baig, Ayaz Hussain, Muhammad AwaisMalik, Muhammad MehdiRaza, SalmanIbrar, ZunishAbbas (2019), did diabetes prediction using significant attributes. Thus, the relationship between the differing attributes is also characterized in this study. Various tools were used to determine significant attribute selection, and for clustering, prediction, and association rule mining for diabetes. Significant attribute selection was made via the principal component analysis method. Lately the findings indicate a strong association of diabetes with body mass index (BMI) and with glucose level, which was extracted via the Apriori method. Artificial neural networks (ANN), random forest (RF), and K-means clustering techniques were implemented for the prediction of diabetes. The ANN technique provided the best accuracy of 75.7% and may be useful to assist medical professionals with treatment decisions.

6. Diabetes Prediction using Machine Learning Algorithms - 2019

Diabetes Prediction using Machine Learning Algorithms (2019) by Aishwarya Mujumdar, Dr. Vaidehi Vb applied various machine learning algorithms to a dataset in order to classify individuals as diabetic or non-diabetic. The Logistic Regression algorithm had the highest accuracy at 96%, but the use of a pipeline resulted in the AdaBoost classifier having the highest accuracy at 98.8%. When compared to an existing dataset, the current model demonstrated improved accuracy and precision in predicting diabetes. They also stated that in the future, it may be possible to use this model to predict the likelihood of non-diabetic individuals developing diabetes.

7. Research on Diabetes Prediction Method Based on Machine Learning - 2020

Research on Diabetes Prediction Method Based on Machine Learning (2020) by Jingyu Xue, Fanchao Min Fengying Ma reckoned that although there is no direct relationship between age and diabetes, there is a trend of younger individuals developing diabetes. Early detection of diabetes is crucial for effective treatment, and machine learning has improved the ability to predict diabetes risk. Through the use of data mining methods and various

machine learning techniques, this study found that the support vector machine (SVM) algorithm had the highest accuracy in diagnosing diabetes through a confusion matrix evaluation test. However, it is important to update this research with additional instance datasets regularly. Overall, the application of data mining algorithms and other technologies has made significant contributions to the medical field and disease diagnosis, and it is hoped that it will assist clinicians in making more informed decisions about disease status.

8. Diabetes Prediction Using Machine Learning - 2020

Diabetes Prediction Using Machine Learning (2020) by KM Jyoti Rani focused on developing a system for early detection of diabetes using machine learning classification algorithms. Five algorithms were evaluated using the John Diabetes Database, and the Decision Tree algorithm was found to be the most effective with an accuracy of 99%. The results of this study demonstrate the potential of the designed system for predicting diabetes at an early stage. The work could also be expanded and improved to include additional machine learning algorithms for automating the analysis of diabetes.

9. Diabetes Prediction: A Deep Learning Approach - 2019

Md. Milon Islam and Safial Islam Ayon researched “Diabetes Prediction: A Deep Learning Approach "(2019) and affirmed that diabetes is a serious and potentially life-threatening condition that requires early detection and treatment. They used deep neural network techniques to predict diabetes based on various medical factors to proceed on the same. The accuracy of the model was found to be 98.35% through five-fold cross-validation, which is higher than the accuracy of other methods used to predict diabetes. Their proposed system has the potential to be useful for both medical professionals and the general public in detecting diabetes early on.

10. Deep learning approach for diabetes prediction using PIMA Indian dataset - 2020

Huma Naz and Sachin Ahuja’s “Deep learning approach for diabetes prediction using PIMA Indian dataset” – 2020, aimed to develop a prediction model for assessing the risk of diabetes using the PIMA Indian dataset. The results of this research showed that machine learning algorithms, including decision trees, artificial neural networks, naive Bayes, and deep learning, can be effective in identifying risk factors and improving the accuracy of predicting diabetes. Among these four classifiers, deep learning had the highest accuracy rate at 98.07%. The researchers plan to use this deep learning algorithm to create a tool, such as an app or website, that healthcare professionals can use for early detection of diabetes in the future."

11. Transforming Diabetes Care Through Artificial Intelligence: The Future Is Here - 2019

Irene Dankwa-Mullan, MD, MPH, Marc Rivo, MD, MPH, Marisol Sepulveda, DO, MPH, Yoonyoung Park, ScD, Jane Snowdon, Ph.D., and Kyu Rhee, MD, MPP has conducted a predefined, online PubMed search of publicly available sources of information from 2009 onward using the search terms ‘‘diabetes’’ and ‘‘artificial intelligence.’’. The purpose of this article is to better understand what AI advances may be relevant today to persons with diabetes (PWDs), their clinicians, family, and caregivers. The study included clinically relevant, high-impact articles, and excluded articles whose purpose was technical in nature. A total of 450 published diabetes and AI articles have met the inclusion criteria. The studies represented a diverse and complex set of innovative approaches that aimed to transform diabetes care in 4 main areas: automated retinal screening, clinical decision support, predictive population risk stratification, and patient self-management tools. A review of the high-impact articles has suggested that AI applications are aiming to transform diabetes care in 4 main areas: automated retinal screening, clinical decision support, predictive population risk stratification, and patient self-management tools.

12. Artificial Intelligence: The Future for Diabetes Care - 2020

The discipline of artificial intelligence (AI), which is rapidly expanding, has applications that could revolutionize how this chronic ailment is diagnosed and managed. Diabetes is a global pandemic. Algorithms supporting predictive models for the risk of getting diabetes or its complications have been developed using machine learning principles. Digital treatments have established themselves as lifestyle therapy intervention for the control of diabetes. Clinical decision support is helpful for both patients and healthcare workers as diabetes patients are given more autonomy to self-manage their condition. AI makes it possible to continuously and easily remotely monitor a patient's symptoms and biomarkers. Furthermore, internet forums and social media platforms improve patient involvement in diabetes care. Resource usage in diabetes has been improved thanks to technological advancements. With the use of AI, diabetes management will undergo a paradigm change from traditional management techniques to constructing targeted data-driven precision care.

13. Machine Learning and Data Mining Methods in Diabetes Research - 2017

The aim of the is to conduct a systematic review of the applications of machine learning, data mining techniques, and tools in the field of diabetes research with respect to a) Prediction and Diagnosis, b) Diabetic Complications, c) Genetic Background and Environment, and e) Health Care and Management with the first category appearing to be the most popular. For the analyses, the researchers employed a wide range of ML algorithms for clinical datasets. In general, 85% of those users were characterized by supervised learning approaches and 15% by unsupervised ones, and more specifically, association rules. The most effective and often used algorithm is based on support vector

machines (SVM). The title applications in the chosen papers suggest the value of extracting important knowledge that can lead to new hypotheses aiming for deeper comprehension and additional research in Diabetes Mellitus.

14. Artificial Intelligence for Diabetes Management and Decision Support: Literature Review - 2018

In this paper, the author has reviewed recent efforts to use artificial intelligence techniques to assist in the management of diabetes, along with the associated challenges. Artificial intelligence methods in combination with the latest technologies, including medical devices, mobile computing, and sensor technologies, have the potential to enable the creation and delivery of better management services to deal with chronic diseases like diabetes. They have analyzed papers related to diabetes care from 2010 to 2018 and selected 141 articles for detailed review. The work proposed a functional taxonomy for diabetes management and artificial intelligence. The potential of AI to enable diabetes solutions has been investigated in the context of multiple critical management issues. The results included Blood glucose control strategies, Blood glucose prediction, Detection of adverse glycemic events, Insulin bolus calculators and advisory systems, Risk and patient personalization, Detection of meals, exercise and faults, Lifestyle and daily-life support in diabetes management. The work concluded that artificial intelligence methods are being progressively established as suitable for use in clinical daily practice, as well as for the self-management of diabetes.

15. A data-driven approach to predicting diabetes and cardiovascular disease with machine learning - 2019

Diabetes and cardiovascular disease are two of the main causes of death in the United States. In this work, they evaluated the capabilities of machine learning models in detecting at-risk patients using survey data (and laboratory results), and identified key variables within the data contributing to these diseases among the patients. Using the National Health and Nutrition Examination Survey (NHANES) dataset, the researchers analyzed various supervised machine-learning models to identify patients with such diseases. Multiple machine learning models (logistic regression, support vector machines, random forest, and gradient boosting) were assessed on their classification performance using various time-frames and feature sets for the data. The models were then integrated to create a weighted ensemble model, which may use the strengths of several models to increase the accuracy of detection. For diabetes classification (based on 123 variables), the eXtreme Gradient Boost (XGBoost) model achieved an AUROC score of 86.2% (without laboratory data) and 95.7% (with laboratory data). The results concluded that the top five predictors in diabetes patients were 1) waist size, 2) age, 3) self-reported weight, 4) leg length, and 5) sodium intake.

16. Classification and prediction of diabetes disease using machine learning paradigm - 2020

Md. Maniruzzaman e, Md. Jahanur Rahman, Benojir Ahammed & Md. Menhazul Abedin, (2020) the main objective of this study is to develop a machine learning (ML)- based system for predicting diabetic patients. They have used a diabetes dataset, conducted in 2009–2012, derived from the National Health and Nutrition Examination Survey. The dataset consists of 6561 respondents with 657 diabetic and 5904 controls. Logistic regression (LR) is used to identify the risk factors for diabetes disease based on p-value and odds ratio (OR). They have adopted four classifiers like naïve Bayes (NB), decision tree (DT), Adaboost (AB), and random forest (RF) to predict diabetic patients. Performances of these classifiers are evaluated using accuracy (ACC) and area under the curve (AUC). The overall ACC of the ML-based system is 90.62%. The combination of LR-based feature selection and RF-based classifier performs better with an accuracy of 94.25%

17. Diabetes prediction model based on an enhanced deep neural network - 2020

Huaping Zhou, Raushan Myrzashova & Rui Zheng, (2020) proposed a method that can predict the occurrence of diabetes in the future and also determines the type of the disease that a person experiences. This method will help to provide the right treatment for the patient. By transforming the task into a classification problem, the model is mainly built using the hidden layers of a deep neural network and uses dropout regularization to prevent overfitting. Number of parameters are tuned and the binary cross-entropy loss function is used, which gives a deep neural network prediction model with high accuracy. The experimental results show the effectiveness and adequacy of the proposed DLPD (Deep Learning for Predicting Diabetes) model. The best training accuracy of the diabetes type data set is 94.02174%, and the training accuracy of the Pima Indians diabetes data set is 99.4112%. Extensive experiments have been conducted on the Pima Indians diabetes and diabetic type datasets.

18. Early prediction of diabetes by applying data mining techniques: A retrospective cohort study - 2022

Mohammed Zeyad Al Yousef, Adel Fouad Yasky, Riyad Al Shammari and Mazen S. Ferwana, (2022) have researched to improve healthcare services and assist in building predictive models to estimate the probability of diabetes in patients. A chart review, which was a retrospective cohort study, was conducted at the National Guard Health Affairs in Riyadh, Saudi Arabia. Data were collected from 5 hospitals using National Guard Health Affairs databases. They have used 38 attributes of 21431 patients between 2015 and 2019. The following phases were performed: (1) data collection, (2) data preparation, (3) data mining and model building, and (4) model evaluation and validation. Subsequently, 6 algorithms were compared with and without the synthetic minority oversampling technique. The highest performance was found in the Bayesian network, which had an area

under the curve of 0.75 and 0.71. Although the results were acceptable, the missing data owing to technical issues played a major role in affecting the performance of this model. Nevertheless, the model could be used in prevention, health monitoring programs, and as an automated mass population screening tool without the need for extra costs compared to traditional methods.

19. Diabetes prediction model using data mining techniques - 2023

Rashi Rastogi and Mamta Bansal, (2022) have proposed a diabetes prediction model using data mining techniques. The data mining techniques applied are Random Forest, Support Vector Machine (SVM), Logistic Regression, and Naive Bayes. The proposed mechanism was trained using Python and analyzed with a real dataset from Kaggle. Furthermore, the performance of the proposed mechanism was analyzed using the confusion matrix, sensitivity and accuracy performance metrics. In comparison to other data mining techniques, logistic regression scored higher accuracy of 82.46% whereas in SVM the accuracy is low, i.e.,79.22%.

20. Big data analytics in healthcare by data mining and classification techniques- 2022

Jayasri N.P and R. Aruna, (2021) proposed a healthcare system that aims to evaluate the medical database of diabetes patients by a mixture of innovative hierarchical decision attention network, association rules (AR) and multiclass outlier classification with MapReduce framework. The association rule apriori algorithm in a MapReduce framework considers health data to create regulations. This is employed to discover the association among diseases and their signs. This examination is made by means of UCI machine learning datasets of diabetes containing 50 attributes. The results of the proposed algorithm are offered by parameters for instance precision, accuracy, recall, and F-score. In the future, this algorithm will be allowed to cloud computing structures for improved access and perform in real time.

21. Diabetes Data Prediction in healthcare Using Hadoop over Big Data - 2020

Gajanand Sharma et al, (2020) describes that big data analytics can be applied to a huge amount of data such as Electronic Medical Record (EMR), pharmacy reports, laboratory reports and among other data related to patients, to generate useful patterns and relation between different factors which affect diabetes. The results obtained from this analysis shows relation between different attributes which can be used to improve the healthcare system. In this paper the analysis of the diabetes dataset is done using Hadoop framework, which is a distributive framework and can be used to analyze large amounts of data. The dataset is taken from PIMA Indian Database, which includes different factors that affect diabetes like age, blood pressure, BMI (Body-Mass Index), skin thickness etc. Results produced by the analysis of data are projects on Power BI.

22. Identification of risk factors for patients with diabetes: diabetic polyneuropathy case study - 2020

Case Study Oleg Metsker, Kirill Magoev, Alexey Yakovlev, Stanislav Yanishevskiy, Georgy Kopanitsa, Sergey Kovalchuk and Valeria V. Krzhizhanovskaya has worked on the Identification of risk factors for patients with diabetes: diabetic polyneuropathy case study (2020). The purpose of this study is the implementation of machine learning methods for identifying the risk of diabetes polyneuropathy based on structured electronic medical records collected in databases of medical information systems. It was discovered that inclusion of two expressions, namely "nephropathy" and "retinopathy" allows to increase the performance, achieving up to 79.82% precision, 81.52% recall, 80.64% F1 score, 82.61% accuracy, and 89.88% AUC using the neural network classifier. Additionally, different models showed different results in terms of interpretation significance: random forest confirmed that the most important risk factor for polyneuropathy is the increased neutrophil level, meaning the presence of inflammation in the body. Linear models showed linear dependencies of the presence of polyneuropathy on blood glucose levels, which is confirmed by the clinical interpretation of the importance of blood glucose control.

23. Prediction of Diabetes Using Data Mining Techniques - 2018

Fikirte Girma Woldemichael, Sumitra Menaria has proposed to predict diabetes using data mining techniques (2018). They have used a back propagation algorithm to predict whether the person is diabetic or not. And also, J48, naive bayes and support vector machines were used to predict diabetes. These neural networks were having an input layer with 8 parameters, one hidden layer having 6 neurons and producing one output layer. 5-fold cross-validation technique and a large value learning rate was used to improve the performance of the model. PIMA Indian dataset used to conduct this study. The study was implemented in RStudio using the R programming language. The performance of the Back propagation algorithm to predict diabetic diseases gave 83.11 % accuracy, 86.53% sensitivity and 76% specificity, the result has shown improvement from previous work.

24. Leveraging Pima Dataset to Diabetes Prediction: Case Study of Deep Neural Network - 2022

Pélagie Hounguè, Annie Ghylaine Bigirimana has done a comparative analysis of different works on diabetes prediction using (Deep Neural Network) DNN (2022). The contribution of this paper was given in two-folds: 1) Deep Neural Network (DNN) approach is used on Pima Indian dataset to predict diabetes using 10 k-fold cross validation and 89% accuracy is obtained; 2) comparative analysis of previous work is provided on diabetes prediction using DNN with the tested model. The results show that diabetes detection using PIMA Indian dataset with k-fold cross-validation on pima could decrease the efficiency of the model with respect to using a model.

25. Detection and Prediction of Diabetes Using Data Mining: A Comprehensive Review - 2021

Farrukh Aslam Khan, Khan Zeb, Mabrook Al-Rakhami, Abdelouahid Derhab and Syed Ahmad Chan Bukhari has presented a comprehensive review of the state-of-the-art in the area of diabetes diagnosis and prediction using data mining-based diabetes diagnosis and prediction techniques and their classification based on the underlying models used (2021). Based on the literature review of data mining-based techniques for diabetes detection, classification and prediction, they have provided a comprehensive classification of the commonly used diabetes diagnosis and prediction techniques. They have evaluated different schemes on parameters like, algorithm/model, type of input data (data input), plug-n-play capability, etc. On the basis of this analysis and evaluation, it is concluded that for accurate detection, classification, and prediction of the disease, we need to preprocess the data and use hybrid techniques, which incorporate different models in parallel instead of using an individual model. For preprocessing, we need to use dimensionality reduction, denoising, feature selection, and feature extraction techniques in combination with the classification and prediction schemes for optimal performance and results.

26. Current Techniques for Diabetes Prediction: Review and Case Study - 2019

Souad Larabi-Marie-Sainte, Linah Aburahmah, Rana Almohaini and Tanzila Saba have surveyed all the ML and DL techniques-based diabetes predictions published in the last six years (2019). One study was developed that aimed to implement those rarely and not used ML classifiers on the Pima Indian Dataset to analyze their performance. The decision tree algorithms obtained the highest accuracy and are recommended to be used in the classification and prediction problems. The other algorithms also have competitive accuracy. Hence, I can recommend using these algorithms in the classification and prediction studies to take benefit from their strengths. Moreover, these algorithms can be used in a combined model with other Deep or Machine Learning techniques as well as Artificial Intelligence techniques to boost their accuracy. The classifiers obtained an accuracy of 68%–74%. The recommendation is to use these classifiers in diabetes prediction and enhance them by developing combined models. For the DL algorithms, the highest accuracy achieved by researchers was 95%.

DRAWBACKS (From Existing Systems):

The development of machine learning algorithms for diabetes prediction has shown promise in recent years, but limitations exist in current studies. From our research and literature review we found that these limitations include the use of limited or non-representative datasets, failure to consider other potential risk factors, lack of evaluation on external datasets, and limited consideration of imbalanced data and individual patient data. Additionally, concerns such as accuracy, bias, limited data, and cost are important considerations when developing diabetic prediction models.

However, the proposed system addresses some of these limitations and can provide advantages over traditional approaches. Ensemble techniques, for instance, can help reduce model variance and the risk of overfitting by combining multiple weaker models. They can also improve accuracy by squeezing more information from limited datasets and reducing model bias by combining models trained on different data subsets or with different algorithms. Additionally, ensemble techniques can help simplify model complexity by breaking down complex models into simpler ones that are easier to interpret. Overall, while current studies provide insights into the potential of machine learning algorithms for diabetes prediction, their limitations highlight the need for further research before such algorithms can be applied in clinical practice. The proposed system provides a promising approach to overcome some of these limitations and improve the accuracy and robustness of diabetic prediction models. Future research should continue to explore and develop new techniques and models to further improve the accuracy and effectiveness of diabetes prediction systems.

DATASET

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

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1

This dataset is from the National Institute of Diabetes and Digestive and Kidney Diseases.

Pregnancies: Number of times pregnant
Glucose: Plasma glucose concentration 2 hours in an oral glucose tolerance test
BloodPressure: Diastolic blood pressure (mm Hg)
SkinThickness: Triceps skin fold thickness (mm)
Insulin: 2-Hour serum insulin (mu U/ml)
BMI: Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction: Diabetes pedigree function
Age: Age (years)
Outcome: Class variable (0 or 1) 268 of 768 is 1, the others are 0

PROPOSED SYSTEM ARCHITECTURE

In this project, we use the following three ensemble models. After constructing and evaluating each model, we will select the optimal ensemble model to solve the problem.

- Stacking Model
- Soft Voting Model
- Hard Voting Model

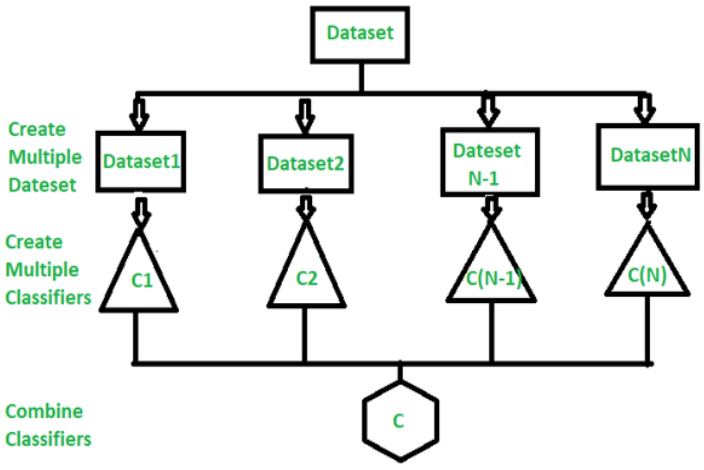
Base Models Used:

- Logistic Regression
- Linear Discriminant Analysis
- CatBoost Classifier
- Extra Trees Classifier
- Gradient Boosting Classifier
- Random Forest Classifier
- AdaBoost Classifier

ENSEMBLE

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote.

- Advantage: Improvement in predictive accuracy.
- Disadvantage: It is difficult to understand an ensemble of classifiers.



STACKING ENSEMBLE

Stacking, also known as a stacked generalization, is one of the most popular ensemble machine learning techniques used to predict multiple nodes to build a new model and improve model performance. Stacking enables us to train multiple models to solve similar problems, and based on their combined output, it builds a new model with improved performance. Multiple models are trained and their predictions are used as input for a higher-level model. This higher-level model then makes the final prediction.

ARCHITECTURE OF STACKING

The architecture of the stacking model is designed in such a way that it consists of two or more base/learner models and a meta-model that combines the predictions of the base models. These base models are called level 0 models, and the meta-model is known as the level 1 model. So, the Stacking ensemble method includes original (training) data, primary level models, primary level prediction, secondary level model, and final prediction. The basic architecture of stacking can be represented as shown below the image.

1. Split training data sets into n-folds using the RepeatedStratifiedKFold as this is the most common approach to preparing training datasets for meta-models.
2. Now the base model is fitted with the first fold, which is n-1, and it will make predictions for the nth folds.
3. The prediction made in the above step is added to the x1_train list.
4. Repeat steps 2 & 3 for remaining n-1folds, so it will give x1_train array of size n,
5. Now, the model is trained on all the n parts, which will make predictions for the sample data.
6. Add this prediction to the y1_test list.
7. In the same way, we can find x2_train, y2_test, x3_train, and y3_test by using Model 2 and 3 for training, respectively, to get Level 2 predictions.
8. Now train the Meta model on level 1 prediction, where these predictions will be used as features for the model.
9. Finally, Meta learners can now be used to make a prediction on test data in the stacking model.

VOTING ENSEMBLES

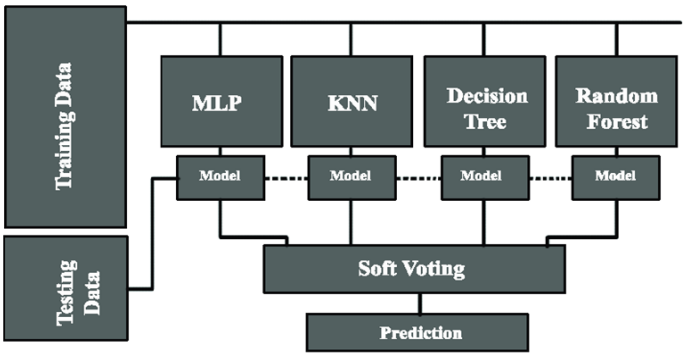
The Voting Classifier is a homogeneous and heterogeneous type of Ensemble Learning, that is, the base classifiers can be of the same or different type. The architecture of a Voting Classifier is made up of a number "n" of ML models, whose predictions are valued in two different ways: hard and soft. In hard mode, the winning prediction is the one with "the most votes".

On the other hand, the Voting Classifier in soft mode considers the probabilities thrown by each ML model, these probabilities will be weighted and averaged, and consequently, the winning class will be the one with the highest weighted and averaged probability.

SOFT VOTING

The soft voting classifier classifies input data based on the probabilities of all the predictions made by different classifiers. It is a type of ensemble learning where multiple models are trained and their predictions are combined by taking the average probability of each class from the individual model predictions. In other words, the output of the ensemble model is the class with the highest average probability from the individual model predictions.

- **ARCHITECTURE OF SOFT VOTING**



- **ALGORITHM**

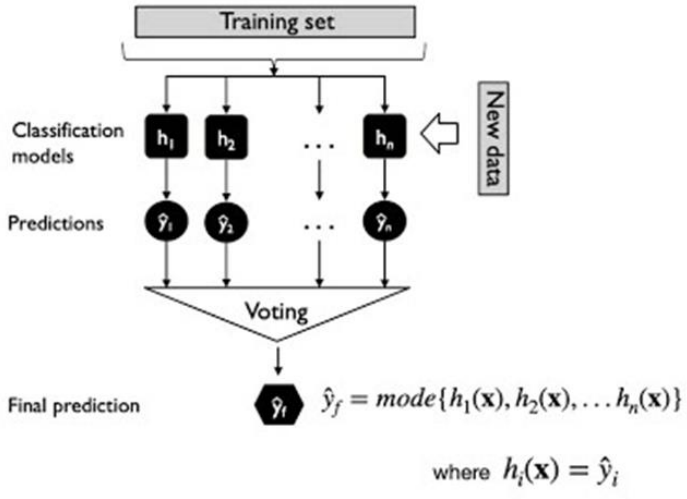
Here is a high-level algorithm for building a soft voting ensemble model:

1. Split the dataset into training and testing sets.
2. Choose a set of base models (e.g. decision trees, logistic regression, k-nearest neighbors) and train each model on the training set.
3. Make predictions on the testing set using each individual model.
4. Calculate the average probability of each class from the individual model predictions.
5. Combine the average probabilities of each class to form the prediction of the ensemble model (i.e., the class with the highest average probability).
6. Evaluate the accuracy of the ensemble model on the testing set.
7. Repeat steps 2-6 with different combinations of base models and select the model with the highest accuracy.

HARD VOTING

In hard voting, the predictions of each algorithm are considered with the ensemble selecting the class with the highest number of votes. The hard-voting ensemble is a type of ensemble learning where multiple models are trained and their predictions are combined by taking a majority vote. In other words, the output of the ensemble model is the mode (most frequently occurring value) of the individual model predictions.

- **ARCHITECTURE OF HARD VOTING**



ALGORITHM

Here is a high-level algorithm for building a hard-voting ensemble model:

1. Split the dataset into training and testing sets.
2. Choose a set of base models (e.g., decision trees, logistic regression, k-nearest neighbors) and train each model on the training set.
3. Make predictions on the testing set using each individual model.
4. Combine the predictions from each model by taking the mode of the predictions. This will be the prediction of the ensemble model.
5. Evaluate the accuracy of the ensemble model on the testing set.
6. Repeat steps 2-5 with different combinations of base models and select the model with the highest accuracy.

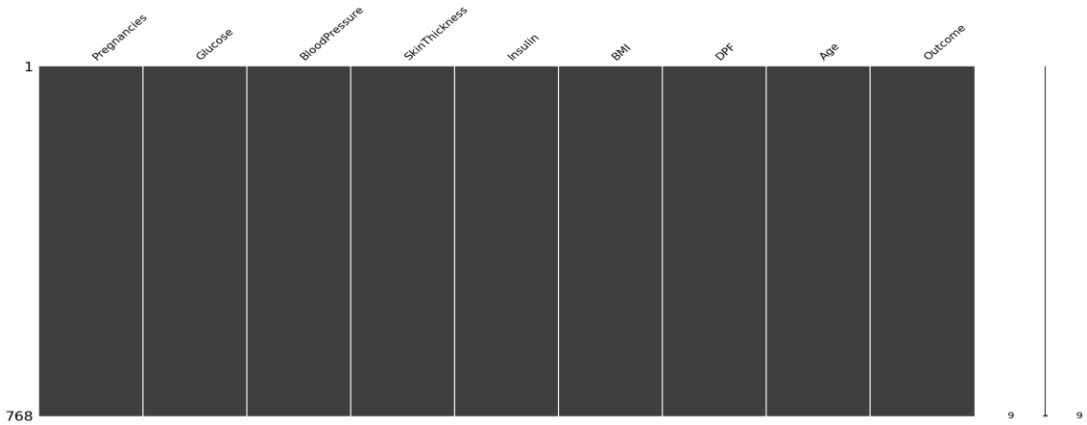
IMPLEMENTATION

The objective of this project is to predict whether a patient has diabetes or not based on certain diagnostic measurements using machine learning techniques. The dataset used in this project is the Pima Indian Diabetes dataset, which contains 768 instances with 8 features.

DATA PREPROCESSING

Firstly, the data was explored by checking for missing values and the distribution of the features. It was observed that the features had varying scales and some of them had a skewed distribution. We decided to perform nonlinear scaling, and decided to use the QuantileTransformer that changes the distribution closest to the normal distribution. Additionally, outliers were removed.

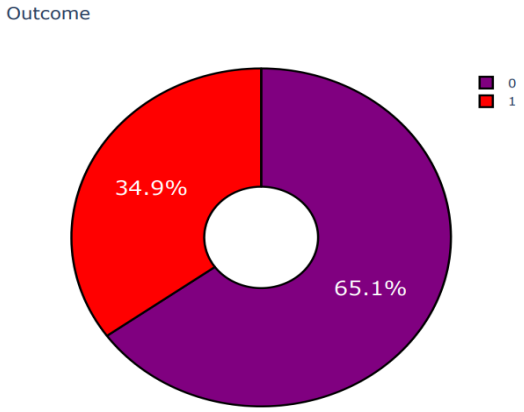
Checking Missing Values and Data Type



The msno.matrix() function is then used to generate a matrix plot that displays the missingness patterns in the diabetes_df DataFrame. The matrix plot shows a grid of white and black cells, where each row corresponds to a variable in the dataset, and each column corresponds to an observation.

A white cell indicates a missing value in the corresponding observation for the corresponding variable, while a black cell indicates the presence of a non-missing value. If there are no missing values in the dataset, the matrix plot will be completely black, indicating that there is no missing data. There is no missing value and all features are numbers. Therefore, there is no need to preprocess for missing values.

Checking Target Imbalance



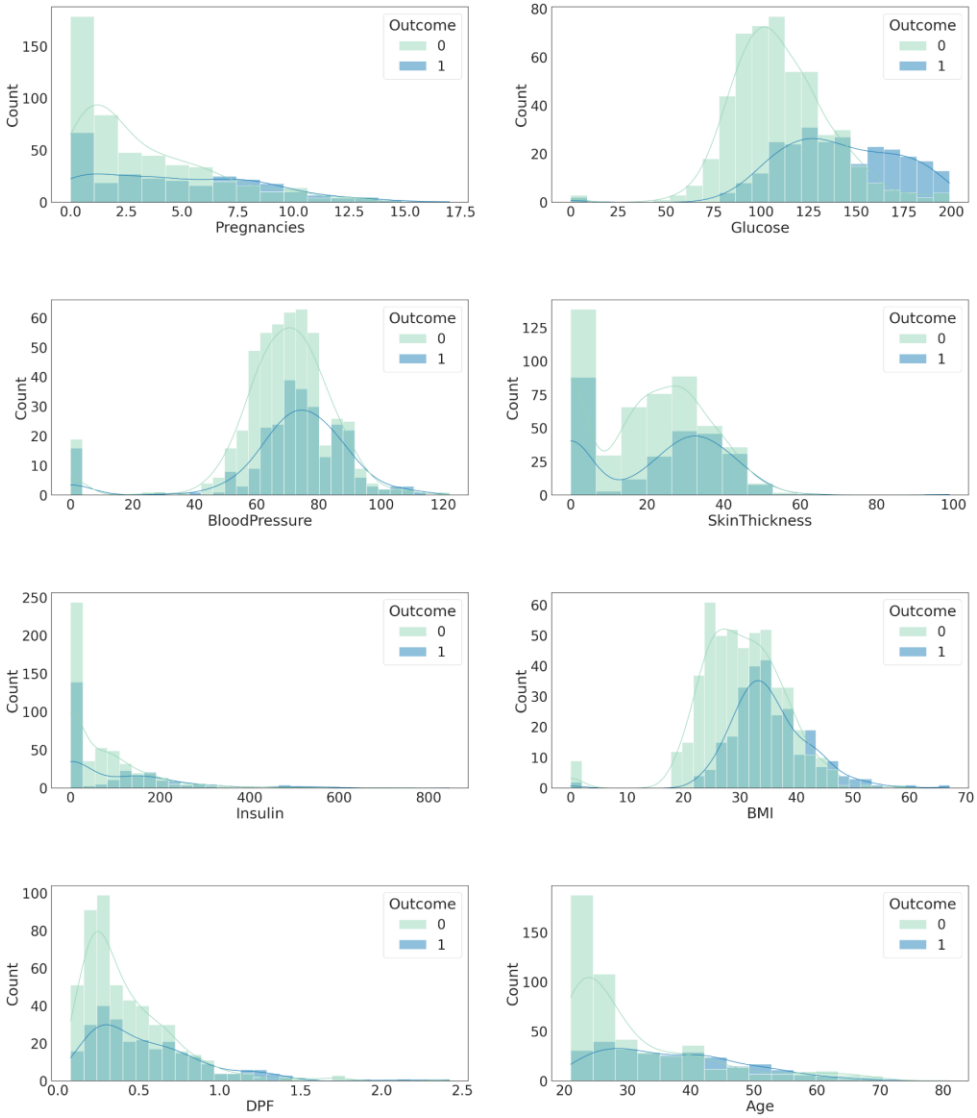
It creates a pie chart that displays the proportion of observations in each class of the 'Outcome' variable. The target is well balanced.

Checking Statistics

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

From the statistics, we can see that some features have minimum values of 0, which does not make sense (e.g., Glucose, BloodPressure, SkinThickness, Insulin, and BMI). These values need to be treated as missing values and imputed accordingly. Also, the standard deviation of some features is quite high, indicating that they may have outliers.

Checking and Removing Outliers



The code generates a set of histograms for each feature in the dataset, with the histograms colored according to the value of the "Outcome" column. This can help identify potential outliers and the distribution of each feature within the dataset.

Proportion of zero values in each feature

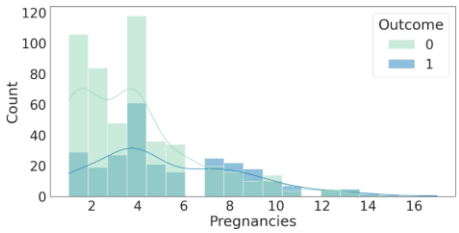
- Pregnancies 0 number of cases 111, percent is 14.45 %
- Glucose 0 number of cases 5, percent is 0.65 %
- BloodPressure 0 number of cases 35, percent is 4.56 %
- SkinThickness 0 number of cases 227, percent is 29.56 %
- Insulin 0 number of cases 374, percent is 48.70 %
- BMI 0 number of cases 11, percent is 1.43 %

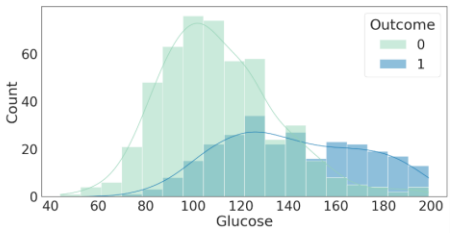
Based on the proportion of zero values in each feature, it seems that SkinThickness and Insulin have a high number of missing values represented as zero. It might not be appropriate to remove all these rows since they represent a significant portion of the dataset. Instead, we can consider imputing these missing values using method mean imputation. If we remove the zero value of each feature, we have a distribution similar to the normal distribution. Therefore, we perform linear scaling and standard scaling.

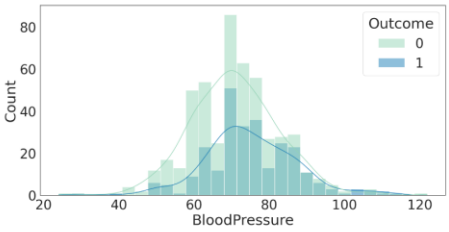
Scaling

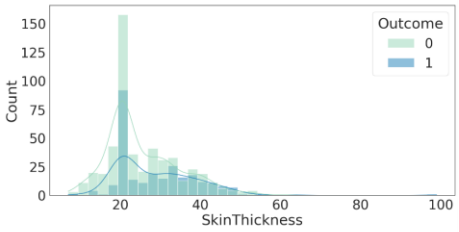
Although the zero values of each feature are converted to mean values, some features have a one-sided shape. Therefore, we decided to perform nonlinear scaling, and decided to use the QuantileTransformer that changes the distribution closest to the normal distribution. The quantile function ranks or smooths out the relationship between observations and can be mapped onto other distributions, such as the uniform or normal distribution.

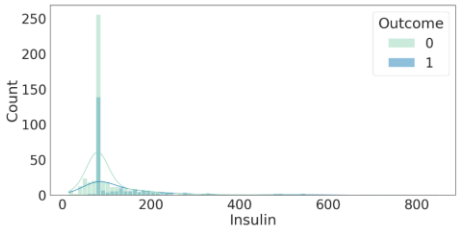
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age
count	576.000000	576.000000	576.000000	576.000000	576.000000	576.000000	576.000000	576.000000
mean	4.315407	121.709154	72.144653	26.341978	118.600098	32.315027	0.468618	32.921875
std	2.916151	30.080570	12.113756	9.158041	94.570101	6.878494	0.339325	11.507539
min	1.000000	44.000000	24.000000	8.000000	14.000000	18.200000	0.078000	21.000000
25%	2.000000	100.000000	64.000000	20.536458	79.799479	27.275000	0.240000	24.000000
50%	3.845052	118.000000	72.000000	23.000000	79.799479	32.000000	0.372500	29.000000
75%	6.000000	138.250000	80.000000	32.000000	127.500000	36.325000	0.612250	40.000000
max	17.000000	199.000000	122.000000	63.000000	846.000000	67.100000	2.420000	81.000000

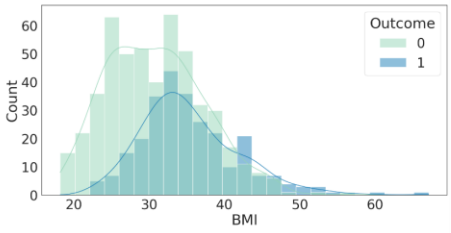

Outcome
0
1
Count
Pregnancies

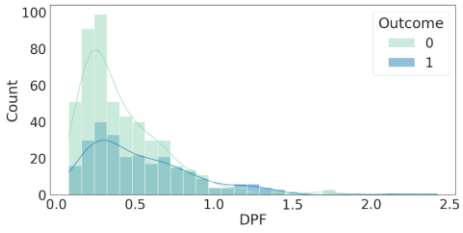

Outcome
0
1
Count
Glucose

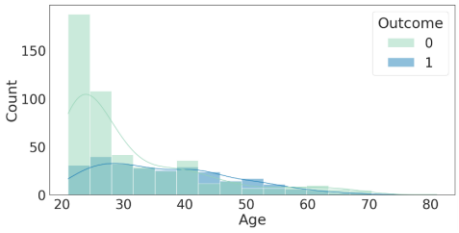

Outcome
0
1
Count
BloodPressure


Outcome
0
1
Count
SkinThickness


Outcome
0
1
Count
Insulin


Outcome
0
1
Count
BMI

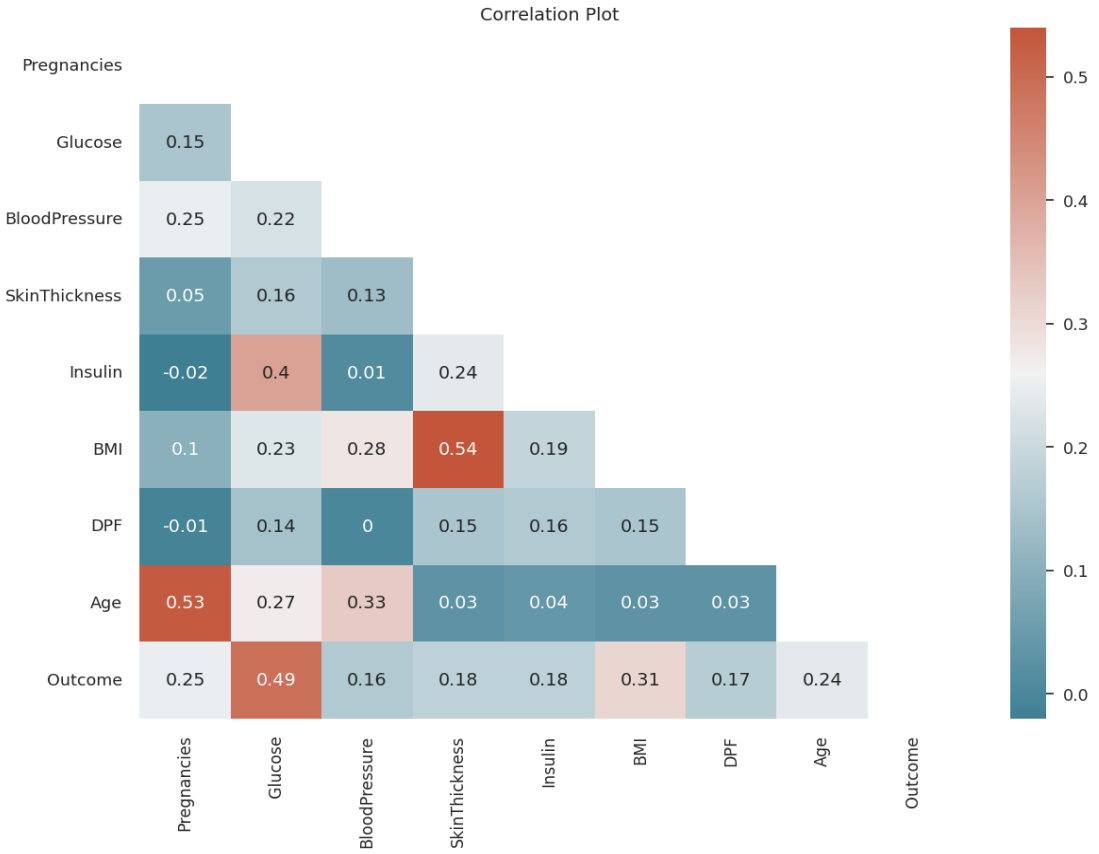

Outcome
0
1
Count
DPF


Outcome
0
1
Count
Age

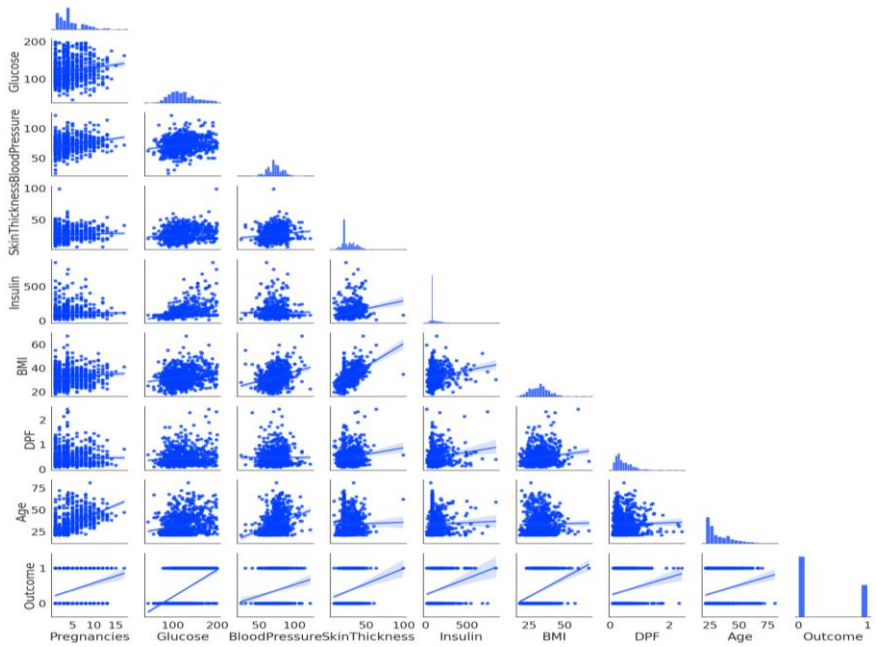
EXPLORATORY DATA ANALYSIS:

To further explore the dataset, various visualizations were created using the seaborn and matplotlib libraries. The visualizations helped in identifying the relationships between the features and the target variable. It was observed that some features had a significant correlation with the target variable.

Checking correlation between features



- Glucose, BMI, Age, and SkinThickness have a moderate positive correlation with the Outcome variable, which indicates that these features may be important predictors of diabetes.
- **BMI has a moderate positive correlation with SkinThickness**, which may indicate that people with a higher BMI tend to have thicker skin.
- The highest correlation values are between **BMI and skin thickness (0.54)**, **age and pregnancies (0.53)**, and **outcome and glucose (0.49).**
- The other correlations are relatively weak or close to zero, such as insulin and pregnancies (-0.02), dpf and pregnancies (-0.01).

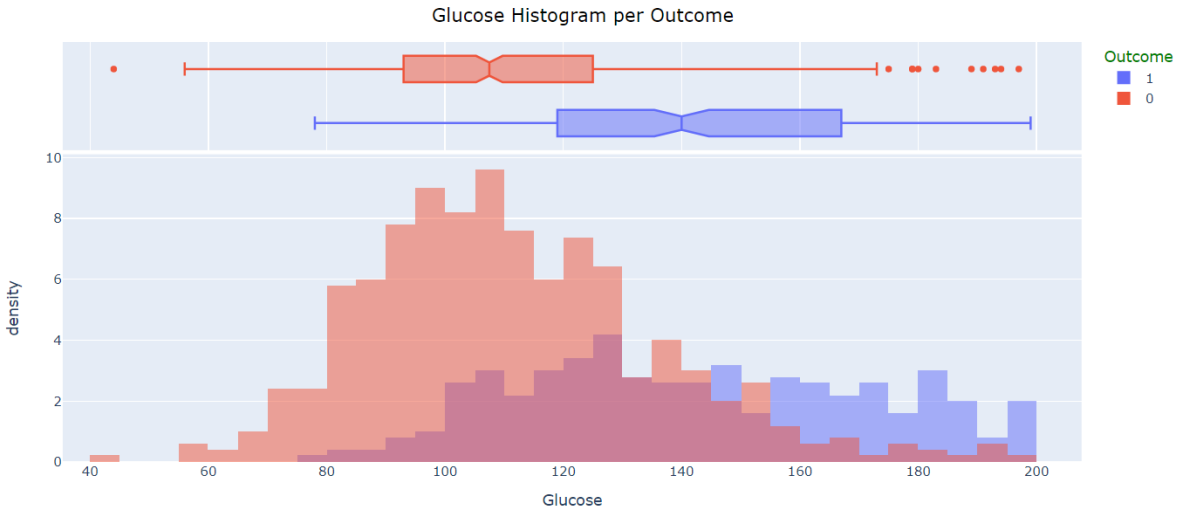


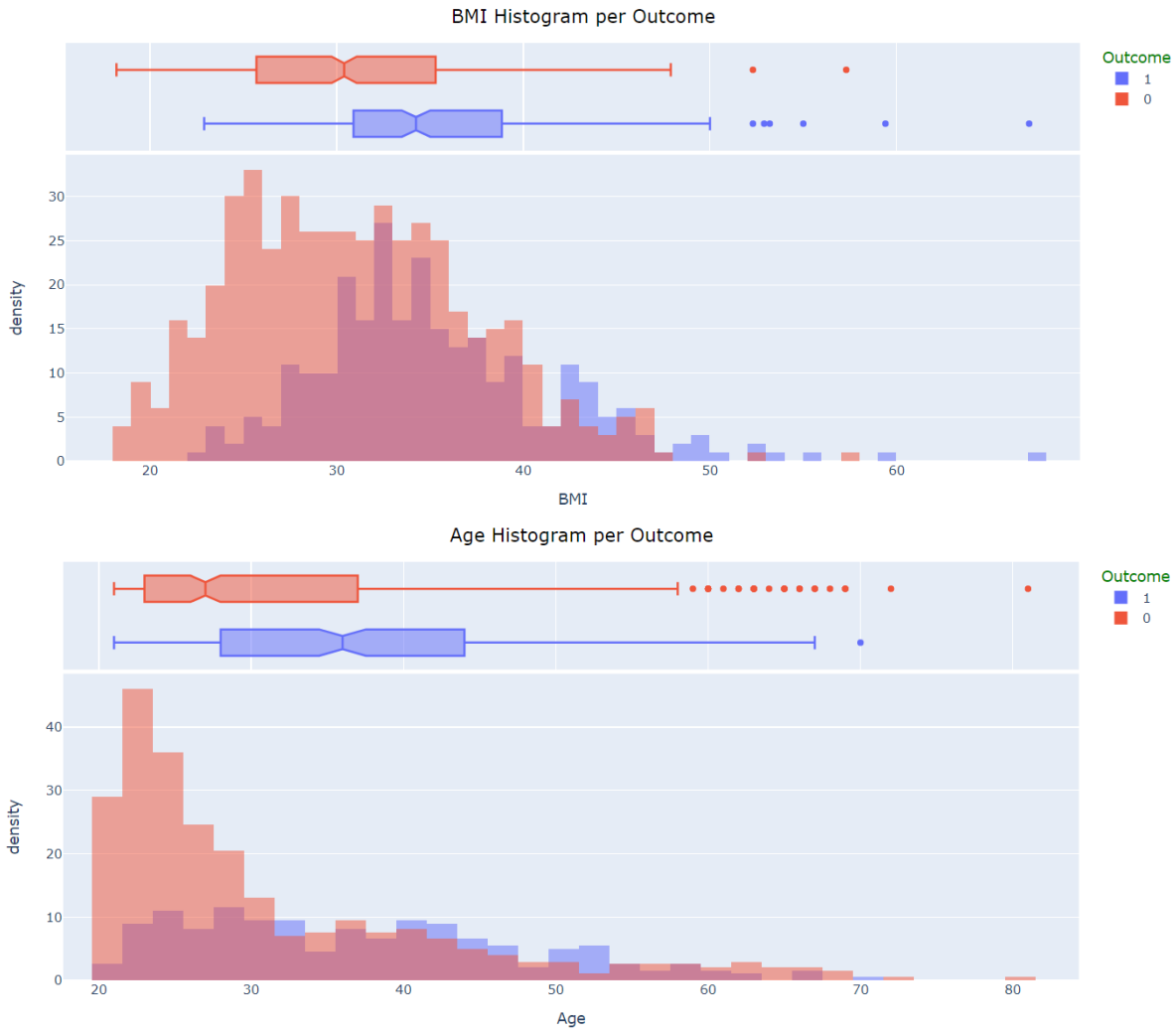
The correlation between Outcome and Glucose is high. Glucose seems to be the most important feature in model training.

BMI, Pregnancies, and Age are also expected to be used as important features in model training.

There is a strong correlation between BMI and SkinThickness, which is not surprising as BMI is calculated based on weight and height, and SkinThickness is a measure of subcutaneous fat

FEATURES:



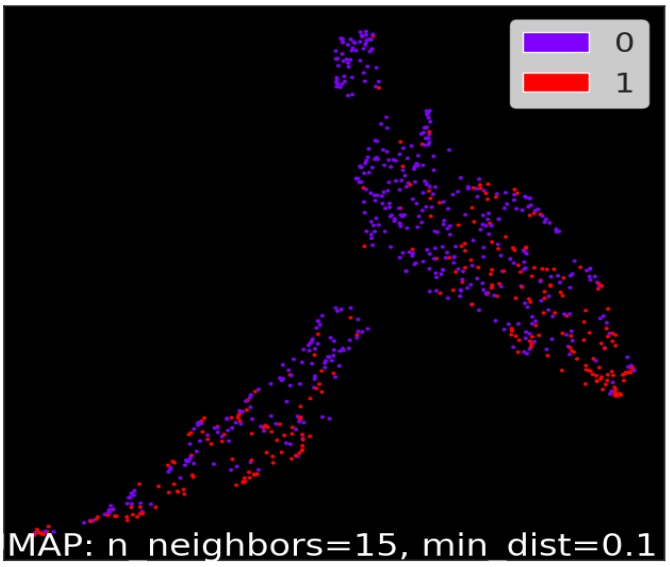


From the above figures, you can see what distribution each feature has for each output. The glucose feature seems to have a clear distinction between the two outcomes, with the distribution for outcome 1 shifted to higher values compared to outcome 0. This indicates that glucose levels could be a strong predictor of diabetes outcome.

Similarly, for BMI, the distribution for outcome 1 is shifted towards higher values compared to outcome 0, indicating that high BMI could also be a potential risk factor for diabetes. Overall, these histograms provide valuable insights into the relationship between features and outcomes in the dataset.

The distribution of age for both outcomes look quite similar in the histogram. However, there seems to be a slightly higher density of outcome 1 in the age range of 25-40 compared to outcome 0.

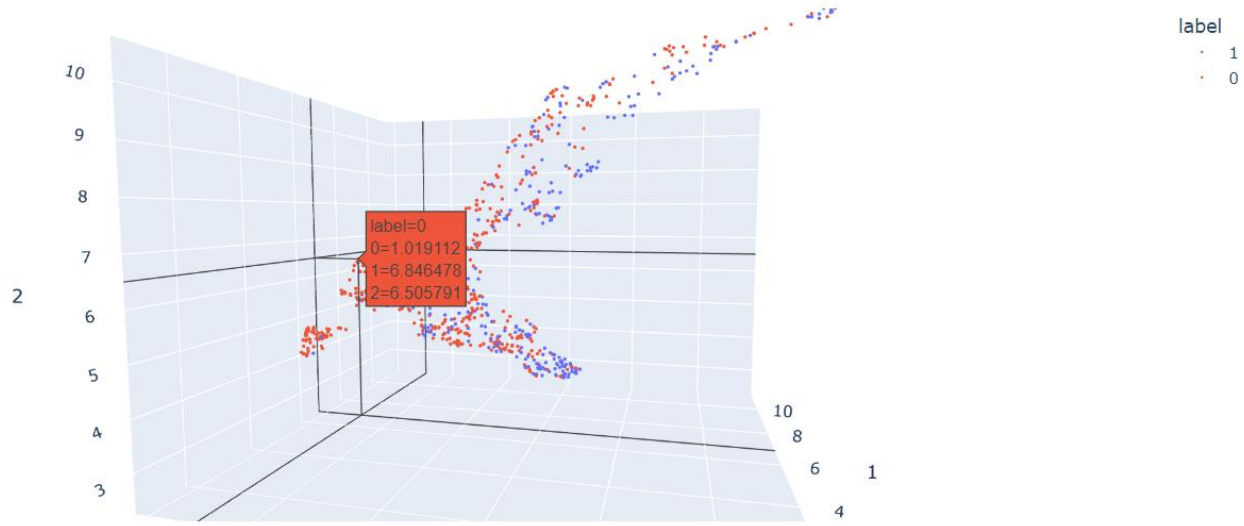
2D plot



The 8-dimensional training dataset shown in the figure above is drawn by reducing the dimensions to 2D. As you can see in the figure, positive and negative are overlapped at the bottom of the figure. Since our models are mainly tree-based models, we will mainly work on determining the boundary, but it seems to be a difficult task in 2D. However, our training dataset is 8-dimensional, just that we can't visualize it. Therefore, it will not be a very difficult task for our models to determine the boundary as shown above.

3D plot

Even if you increase the dimension to 3D, you can see overlapping points.



MODEL BUILDING:

Three ensemble models were built using the Random Forest Classifier, Extra Trees Classifier, and the XGBoost Classifier. The base models were trained using the training dataset and their hyperparameters were tuned using GridSearchCV. The ensemble models were created using both soft and hard voting. The performance of the models was evaluated using the validation dataset.

COMPARING MODELS

	Description	Value
0	Session id	7340
1	Target	Outcome
2	Target type	Binary
3	Original data shape	(768, 9)
4	Transformed data shape	(768, 9)
5	Transformed train set shape	(537, 9)
6	Transformed test set shape	(231, 9)
7	Numeric features	8

Using The Pycaret Library to compare models for the diabetes dataset. The Compare_Models() function allows you to compare various models by training them on the dataset and evaluating their performance using cross-validation.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.7766	0.8383	0.5708	0.7460	0.6373	0.4810	0.4963	0.7550
lda	Linear Discriminant Analysis	0.7766	0.8374	0.5708	0.7441	0.6375	0.4809	0.4955	0.0740
catboost	CatBoost Classifier	0.7617	0.8362	0.6088	0.6833	0.6396	0.4631	0.4680	3.1260
rf	Random Forest Classifier	0.7876	0.8315	0.6295	0.7253	0.6700	0.5153	0.5209	0.7940
et	Extra Trees Classifier	0.7690	0.8276	0.6085	0.6941	0.6450	0.4754	0.4802	1.0240
gbc	Gradient Boosting Classifier	0.7616	0.8228	0.6137	0.6802	0.6408	0.4636	0.4685	0.5980
ada	Ada Boost Classifier	0.7429	0.8065	0.5977	0.6442	0.6165	0.4242	0.4274	0.7460
dummy	Dummy Classifier	0.6518	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0470

```
Processing:   0%|          | 0/41 [00:00<?, ?it/s]
```

Based on the performance metrics above, the top 5 models are:

1. Random Forest Classifier with an AUC score of 0.8315
2. Logistic Regression with an AUC score of 0.8383
3. Linear Discriminant Analysis with an AUC score of 0.8374
4. Extra Trees Classifier with an AUC score of 0.8276
5. Gradient Boosting Classifier with an AUC score of 0.8228

CREATING MODELS

CATBOOST

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7593	0.7985	0.6842	0.6500	0.6667	0.4785	0.4788
1	0.7963	0.8707	0.6842	0.7222	0.7027	0.5479	0.5484
2	0.7593	0.8677	0.5263	0.7143	0.6061	0.4384	0.4490
3	0.7963	0.8737	0.5789	0.7857	0.6667	0.5248	0.5375
4	0.7222	0.8376	0.5789	0.6111	0.5946	0.3836	0.3839
5	0.6852	0.8090	0.6842	0.5417	0.6047	0.3489	0.3555
6	0.7963	0.8617	0.6842	0.7222	0.7027	0.5479	0.5484
7	0.7736	0.7905	0.5556	0.7143	0.6250	0.4664	0.4740
8	0.8302	0.8984	0.6667	0.8000	0.7273	0.6055	0.6108
9	0.6981	0.7540	0.4444	0.5714	0.5000	0.2886	0.2933
Mean	0.7617	0.8362	0.6088	0.6833	0.6396	0.4631	0.4680
Std	0.0446	0.0437	0.0803	0.0824	0.0642	0.0944	0.0943

```
Processing:   0%|          | 0/4 [00:00<?, ?it/s]
```

RANDOM FOREST:

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7222	0.7985	0.6316	0.6000	0.6154	0.3982	0.3985
1	0.8148	0.8429	0.7368	0.7368	0.7368	0.5940	0.5940
2	0.8148	0.8639	0.5789	0.8462	0.6875	0.5624	0.5828
3	0.7963	0.8744	0.6316	0.7500	0.6857	0.5367	0.5410
4	0.8148	0.8398	0.6316	0.8000	0.7059	0.5735	0.5820
5	0.7778	0.7940	0.6842	0.6842	0.6842	0.5128	0.5128
6	0.8333	0.8820	0.7895	0.7500	0.7692	0.6389	0.6394
7	0.8113	0.8302	0.6111	0.7857	0.6875	0.5554	0.5644
8	0.8302	0.8730	0.6667	0.8000	0.7273	0.6055	0.6108
9	0.6604	0.7159	0.3333	0.5000	0.4000	0.1762	0.1832
Mean	0.7876	0.8315	0.6295	0.7253	0.6700	0.5153	0.5209
Std	0.0523	0.0482	0.1149	0.0995	0.0979	0.1288	0.1290

```
Processing:   0%|          | 0/4 [00:00<?, ?it/s]
```

LOGISTIC REGRESSION:

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7222	0.8075	0.5263	0.6250	0.5714	0.3682	0.3711
1	0.7593	0.8361	0.6316	0.6667	0.6486	0.4658	0.4661
2	0.7963	0.8632	0.5263	0.8333	0.6452	0.5123	0.5389
3	0.8148	0.8812	0.6316	0.8000	0.7059	0.5735	0.5820
4	0.7963	0.8195	0.5789	0.7857	0.6667	0.5248	0.5375
5	0.7407	0.8060	0.7368	0.6087	0.6667	0.4577	0.4633
6	0.7778	0.8662	0.6316	0.7059	0.6667	0.5008	0.5025
7	0.7925	0.8365	0.4444	0.8889	0.5926	0.4734	0.5245
8	0.8302	0.8778	0.6111	0.8462	0.7097	0.5940	0.6098
9	0.7358	0.7889	0.3889	0.7000	0.5000	0.3399	0.3670
Mean	0.7766	0.8383	0.5708	0.7460	0.6373	0.4810	0.4963
Std	0.0340	0.0310	0.0966	0.0928	0.0616	0.0762	0.0768

```
Processing:   0%|          | 0/4 [00:00<?, ?it/s]
```

Linear Discriminant Analysis

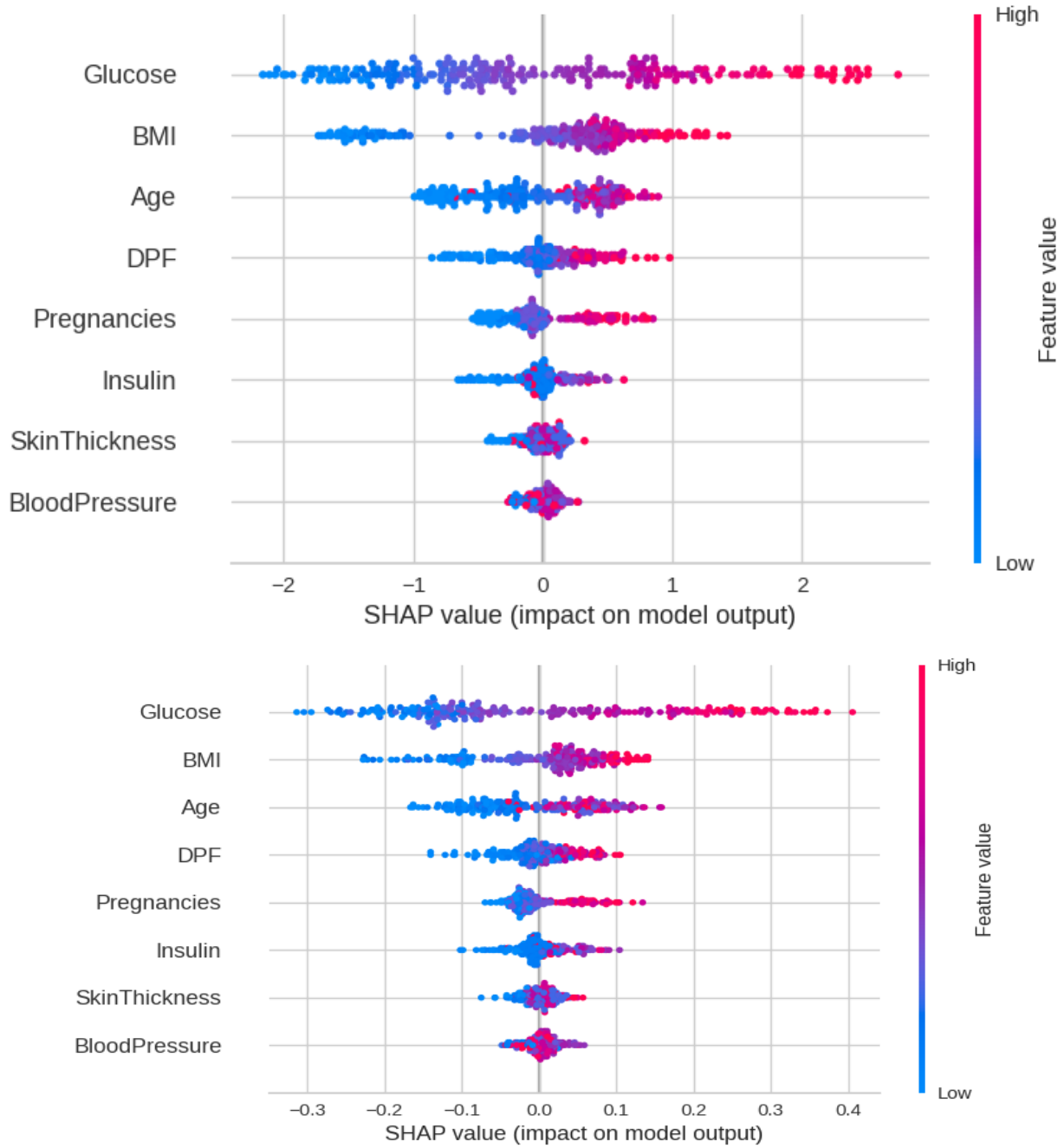
Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7222	0.8075	0.5263	0.6250	0.5714	0.3682	0.3711
1	0.7593	0.8346	0.6316	0.6667	0.6486	0.4658	0.4661
2	0.7963	0.8602	0.5263	0.8333	0.6452	0.5123	0.5389
3	0.8148	0.8842	0.6316	0.8000	0.7059	0.5735	0.5820
4	0.7963	0.8195	0.5789	0.7857	0.6667	0.5248	0.5375
5	0.7407	0.7940	0.7368	0.6087	0.6667	0.4577	0.4633
6	0.7963	0.8662	0.6316	0.7500	0.6857	0.5367	0.5410
7	0.7925	0.8540	0.4444	0.8889	0.5926	0.4734	0.5245
8	0.8302	0.8730	0.6111	0.8462	0.7097	0.5940	0.6098
9	0.7170	0.7810	0.3889	0.6364	0.4828	0.3032	0.3207
Mean	0.7766	0.8374	0.5708	0.7441	0.6375	0.4809	0.4955
Std	0.0372	0.0337	0.0966	0.0971	0.0666	0.0851	0.0866

```
Processing:   0%|          | 0/4 [00:00<?, ?it/s]
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7963	0.7985	0.6842	0.7222	0.7027	0.5479	0.5484
1	0.7593	0.8180	0.6316	0.6667	0.6486	0.4658	0.4661
2	0.7593	0.8376	0.5789	0.6875	0.6286	0.4524	0.4561
3	0.7778	0.8692	0.5789	0.7333	0.6471	0.4882	0.4954
4	0.7593	0.7940	0.5789	0.6875	0.6286	0.4524	0.4561
5	0.7222	0.8045	0.7368	0.5833	0.6512	0.4255	0.4336
6	0.7963	0.8602	0.7368	0.7000	0.7179	0.5587	0.5591
7	0.7170	0.7889	0.5556	0.5882	0.5714	0.3604	0.3607
8	0.8113	0.8762	0.5556	0.8333	0.6667	0.5423	0.5640
9	0.7170	0.7810	0.5000	0.6000	0.5455	0.3424	0.3454
Mean	0.7616	0.8228	0.6137	0.6802	0.6408	0.4636	0.4685
Std	0.0327	0.0336	0.0768	0.0728	0.0499	0.0707	0.0726

```
Processing:   0%|          | 0/4 [00:00<?, ?it/s]
```

The interpret_model function in PyCaret allows for interpretation of the trained model using SHAP (SHapley Additive exPlanations) plots. These plots help to understand the contribution of each variable in the model to the final prediction.



Observation: As expected, glucose is used as the most important feature. SkinTickness and BloodPressure have the low importance.

Tuning Hyperparameters

This function tunes the hyperparameters of a given estimator. The output of this function is a score grid with CV scores by fold of the best selected model based on optimize parameter.

Top 1, Top 2, Top 3, Top 4, and Top 5 models were tuned with different feature importance and decision boundary. Also, there is a big difference in feature importance from the catboost classifier.

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Fold							
0	0.7778	0.7699	0.6842	0.6842	0.6842	0.5128	0.5128
1	0.7963	0.8541	0.6842	0.7222	0.7027	0.5479	0.5484
2	0.7963	0.8782	0.5789	0.7857	0.6667	0.5248	0.5375
3	0.7963	0.8857	0.6316	0.7500	0.6857	0.5367	0.5410
4	0.7778	0.8271	0.5789	0.7333	0.6471	0.4882	0.4954
5	0.7222	0.8030	0.7368	0.5833	0.6512	0.4255	0.4336
6	0.8333	0.8782	0.7895	0.7500	0.7692	0.6389	0.6394
7	0.8302	0.8270	0.6667	0.8000	0.7273	0.6055	0.6108
8	0.8491	0.9095	0.7222	0.8125	0.7647	0.6542	0.6566
9	0.6604	0.7429	0.3333	0.5000	0.4000	0.1762	0.1832
Mean	0.7840	0.8376	0.6406	0.7121	0.6699	0.5111	0.5159
Std	0.0533	0.0510	0.1203	0.0943	0.0987	0.1296	0.1281

```
Processing:   0%|          | 0/7 [00:00<?, ?it/s]
Fitting 10 folds for each of 10 candidates, totalling 100 fits
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7407	0.8015	0.7368	0.6087	0.6667	0.4577	0.4633
1	0.8148	0.8722	0.8421	0.6957	0.7619	0.6126	0.6201
2	0.7963	0.8647	0.7895	0.6818	0.7317	0.5689	0.5729
3	0.8148	0.8842	0.7895	0.7143	0.7500	0.6035	0.6054
4	0.8333	0.8316	0.8421	0.7273	0.7805	0.6473	0.6518
5	0.7407	0.8075	0.8421	0.5926	0.6957	0.4815	0.5041
6	0.8519	0.9008	0.8421	0.7619	0.8000	0.6828	0.6850
7	0.7547	0.8286	0.6667	0.6316	0.6486	0.4605	0.4609
8	0.8302	0.9079	0.8333	0.7143	0.7692	0.6362	0.6409
9	0.6981	0.7238	0.6111	0.5500	0.5789	0.3447	0.3458
Mean	0.7876	0.8423	0.7795	0.6678	0.7183	0.5496	0.5550
Std	0.0480	0.0530	0.0785	0.0648	0.0661	0.1025	0.1022

```
Processing:   0%|          | 0/7 [00:00<?, ?it/s]
Fitting 10 folds for each of 10 candidates, totalling 100 fits
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7222	0.8075	0.5263	0.6250	0.5714	0.3682	0.3711
1	0.7593	0.8361	0.6316	0.6667	0.6486	0.4658	0.4661
2	0.7963	0.8632	0.5263	0.8333	0.6452	0.5123	0.5389
3	0.8148	0.8827	0.6316	0.8000	0.7059	0.5735	0.5820
4	0.7963	0.8195	0.5789	0.7857	0.6667	0.5248	0.5375
5	0.7407	0.8090	0.7368	0.6087	0.6667	0.4577	0.4633
6	0.7778	0.8662	0.6316	0.7059	0.6667	0.5008	0.5025
7	0.7925	0.8397	0.4444	0.8889	0.5926	0.4734	0.5245
8	0.8302	0.8794	0.6111	0.8462	0.7097	0.5940	0.6098
9	0.7358	0.7889	0.3889	0.7000	0.5000	0.3399	0.3670
Mean	0.7766	0.8392	0.5708	0.7460	0.6373	0.4810	0.4963
Std	0.0340	0.0311	0.0966	0.0928	0.0616	0.0762	0.0768

```
Processing:   0%|          | 0/7 [00:00<?, ?it/s]
Fitting 10 folds for each of 10 candidates, totalling 100 fits
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7037	0.7970	0.5263	0.5882	0.5556	0.3344	0.3355
1	0.7407	0.8286	0.5789	0.6471	0.6111	0.4176	0.4190
2	0.7778	0.8632	0.4737	0.8182	0.6000	0.4609	0.4939
3	0.8333	0.8737	0.6316	0.8571	0.7273	0.6112	0.6260
4	0.7222	0.8090	0.5263	0.6250	0.5714	0.3682	0.3711
5	0.7222	0.8060	0.6842	0.5909	0.6341	0.4122	0.4151
6	0.8148	0.8632	0.6316	0.8000	0.7059	0.5735	0.5820
7	0.7925	0.8444	0.4444	0.8889	0.5926	0.4734	0.5245
8	0.8302	0.8968	0.6111	0.8462	0.7097	0.5940	0.6098
9	0.7358	0.7667	0.3889	0.7000	0.5000	0.3399	0.3670
Mean	0.7673	0.8349	0.5497	0.7362	0.6208	0.4585	0.4744
Std	0.0460	0.0383	0.0894	0.1120	0.0702	0.0983	0.1019

```
Processing:   0%|          | 0/7 [00:00<?, ?it/s]
Fitting 10 folds for each of 10 candidates, totalling 100 fits

Original model was better than the tuned model, hence it will be returned.
NOTE: The display metrics are for the tuned model (not the original one).
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.6481	0.7684	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.6481	0.8617	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.6481	0.8737	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.6481	0.8812	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.6481	0.8391	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.6481	0.7985	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.6481	0.9038	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.6604	0.8397	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.6604	0.8984	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.6604	0.7667	0.0000	0.0000	0.0000	0.0000	0.0000
Mean	0.6518	0.8431	0.0000	0.0000	0.0000	0.0000	0.0000
Std	0.0056	0.0479	0.0000	0.0000	0.0000	0.0000	0.0000

```
Processing:   0%|          | 0/7 [00:00<?, ?it/s]
Fitting 10 folds for each of 10 candidates, totalling 100 fits
```

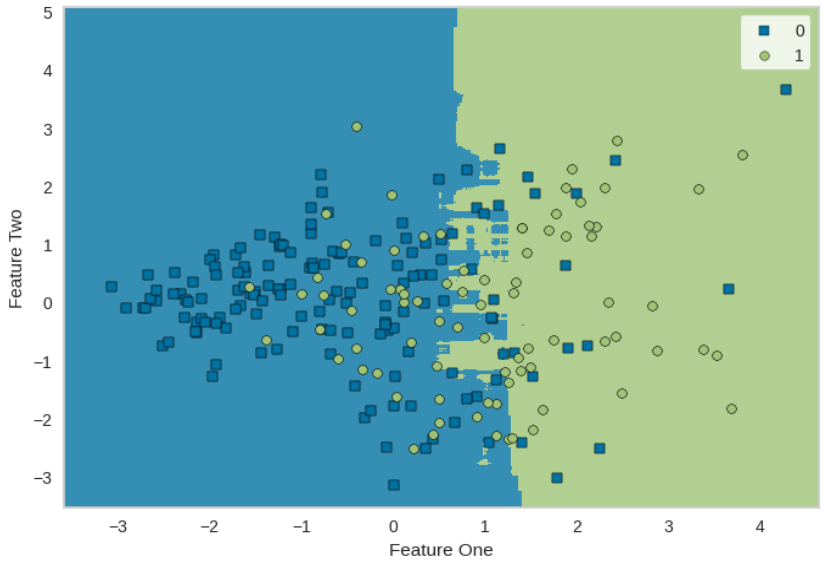
STACKING

```
stack_model = stack_models(estimator_list = top5, meta_model = top5[0],optimize = 'AUC')
```

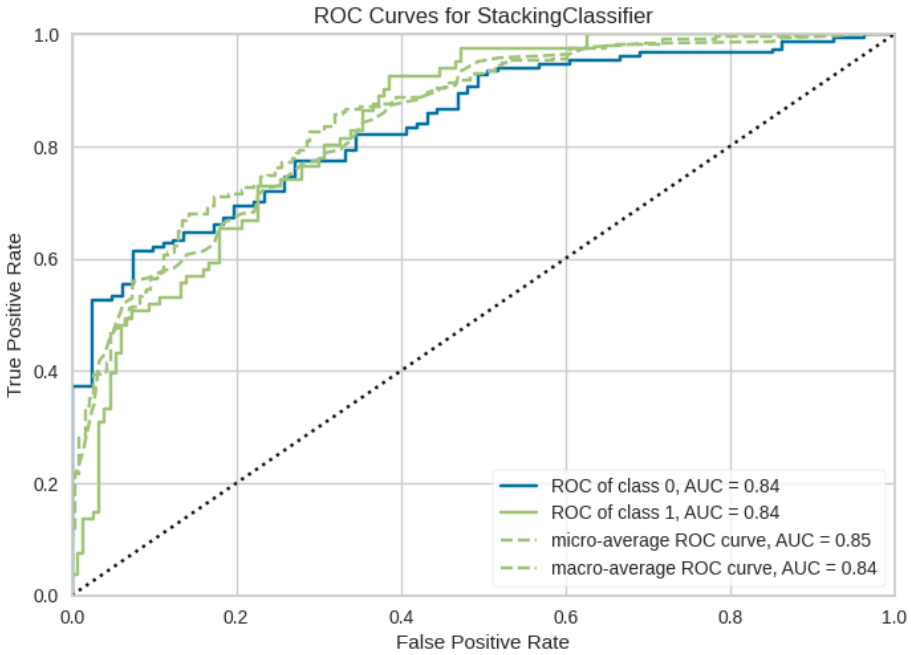
Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7407	0.8120	0.5789	0.6471	0.6111	0.4176	0.4190
1	0.7593	0.8496	0.6316	0.6667	0.6486	0.4658	0.4661
2	0.7778	0.8617	0.5263	0.7692	0.6250	0.4749	0.4921
3	0.7963	0.8842	0.5789	0.7857	0.6667	0.5248	0.5375
4	0.7963	0.8301	0.5789	0.7857	0.6667	0.5248	0.5375
5	0.7222	0.8075	0.6842	0.5909	0.6341	0.4122	0.4151
6	0.7778	0.8722	0.6316	0.7059	0.6667	0.5008	0.5025
7	0.7925	0.8397	0.4444	0.8889	0.5926	0.4734	0.5245
8	0.8302	0.8746	0.6111	0.8462	0.7097	0.5940	0.6098
9	0.6792	0.7889	0.3889	0.5385	0.4516	0.2332	0.2394
Mean	0.7672	0.8420	0.5655	0.7225	0.6273	0.4621	0.4744
Std	0.0413	0.0304	0.0853	0.1065	0.0665	0.0916	0.0958

```
Processing:   0%|          | 0/6 [00:00<?, ?it/s]
```

The top 5 models, you can stack them using the stack_models() function. This will create a meta-model that takes the outputs of the top 5 models as inputs and makes the final prediction. The optimize parameter is set to 'AUC' to optimize the stacked model using AUC.



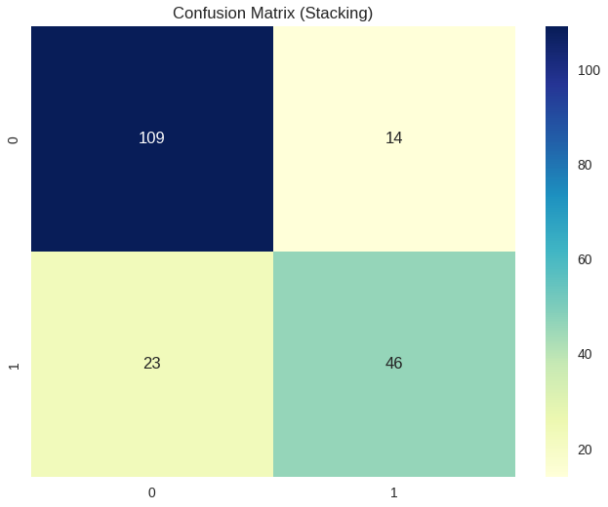
In the case of a stacking model, in some cases overfitting and in some cases underfitting. Overfitting can occur when the model is too complex and fits the training data too closely, leading to poor performance on new, unseen data. Underfitting can occur when the model is too simple and cannot capture the underlying patterns in the data, leading to poor performance on both the training and test data.



It looks like the AUC score of the stacking model is pretty good for both classes, with an AUC of 0.84 for both class 0 and class 1. This indicates that the model has good predictive performance.

```
#prediction
pred = stack_model.predict(X_test)
pred_proba = stack_model.predict_proba(X_test)[:,1]
#Accuracy
confusion_stack = get_clf_eval(y_test,pred,pred_proba)
```

```
accuracy: 0.8073, precision: 0.7667, recall: 0.6667,    F1: 0.7132, AUC:0.8
908
```



Precision and recall have a trade-off relationship.

SOFT VOTING

Soft voting is a technique in which the predicted probabilities of each model are averaged to make the final prediction. In this way, the output of each model is taken into account and the final prediction is made based on a weighted average of the predictions of the models.

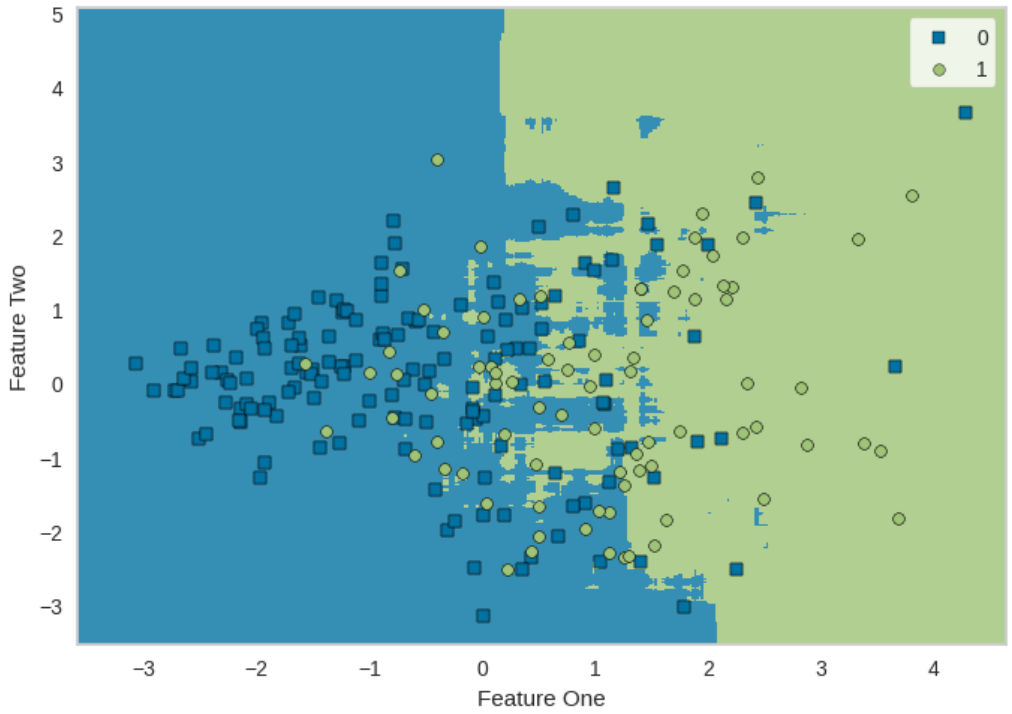
```
blend_soft = blend_models(estimator_list = top5, optimize = 'AUC',method = 'soft')
```

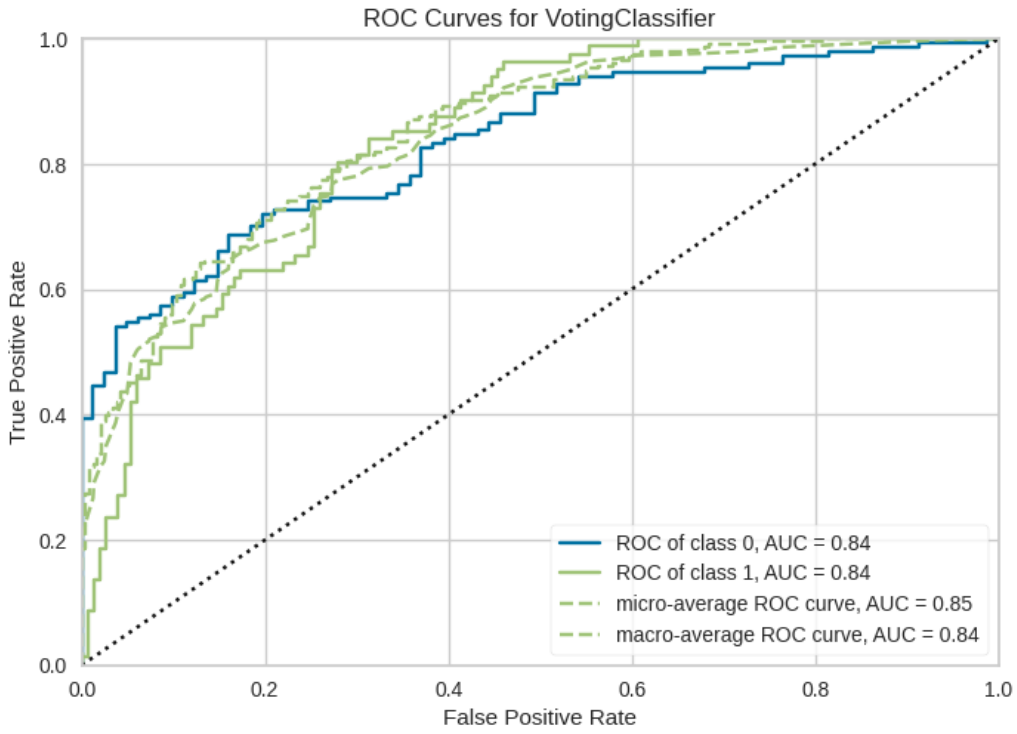
Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7037	0.8135	0.5789	0.5789	0.5789	0.3504	0.3504
1	0.7963	0.8571	0.7368	0.7000	0.7179	0.5587	0.5591
2	0.7778	0.8677	0.5263	0.7692	0.6250	0.4749	0.4921
3	0.7778	0.8917	0.5789	0.7333	0.6471	0.4882	0.4954
4	0.7593	0.8391	0.5263	0.7143	0.6061	0.4384	0.4490
5	0.7222	0.8120	0.6842	0.5909	0.6341	0.4122	0.4151
6	0.7778	0.8842	0.6842	0.6842	0.6842	0.5128	0.5128
7	0.8113	0.8556	0.5000	0.9000	0.6429	0.5285	0.5706
8	0.8491	0.8714	0.6667	0.8571	0.7500	0.6443	0.6547
9	0.6604	0.7921	0.3333	0.5000	0.4000	0.1762	0.1832
Mean	0.7636	0.8484	0.5816	0.7028	0.6286	0.4584	0.4682
Std	0.0522	0.0316	0.1126	0.1173	0.0904	0.1214	0.1246

```
Processing:   0%|          | 0/6 [00:00<?, ?it/s]
```

The output shows the performance metrics of the 10-fold cross-validation for the soft voting ensemble method.

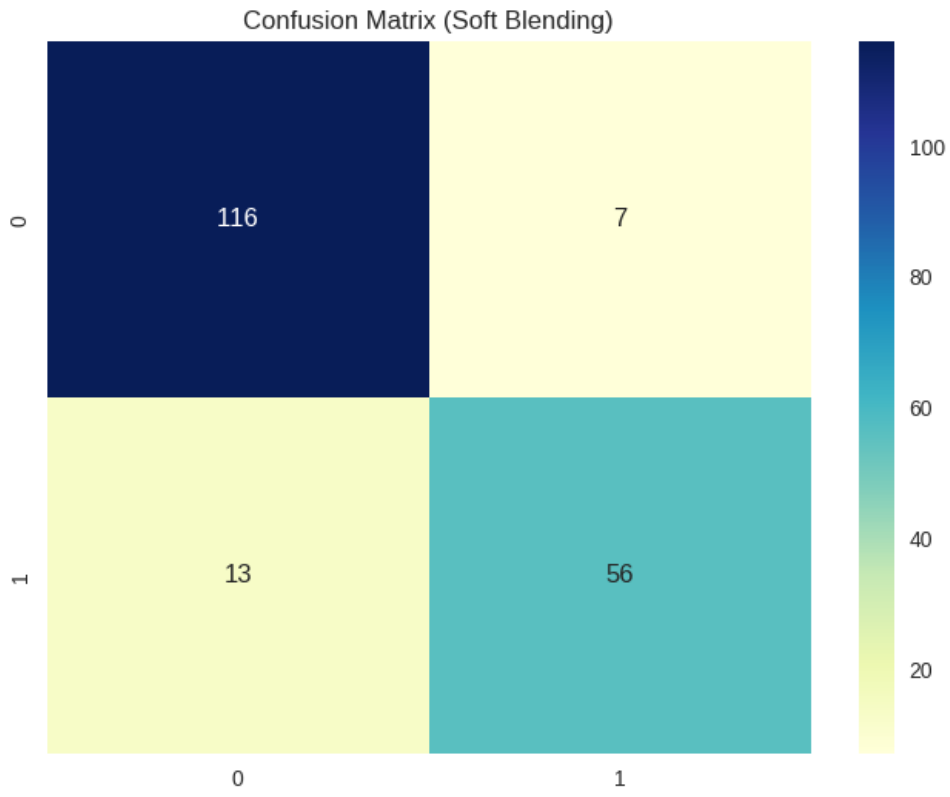
- Accuracy: The average accuracy of the model is 76.36%.
- AUC: The average AUC score of the model is 0.8484.
- Recall: The average recall score of the model is 0.5816.
- Precision: The average precision score of the model is 0.7028.
- F1: The average F1 score of the model is 0.6286.
- Kappa: The average kappa score of the model is 0.4584.
- MCC: The average Matthews correlation coefficient (MCC) score of the model is 0.4682.


0
1
Feature Two
Feature One
5
4
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
4


ROC Curves for VotingClassifier
True Positive Rate
False Positive Rate
ROC of class 0, AUC = 0.84
ROC of class 1, AUC = 0.84
micro-average ROC curve, AUC = 0.85
macro-average ROC curve, AUC = 0.84
0.0
0.2
0.4
0.6
0.8
1.0

```
#prediction
pred = blend_soft.predict(X_test)
pred_proba = blend_soft.predict_proba(X_test)[:,1]
#Accuracy
confusion_soft = get_clf_eval(y_test,pred,pred_proba)
```

```
accuracy: 0.8958, precision: 0.8889, recall: 0.8116,    F1: 0.8485, AUC:0.9
584
```



The confusion matrix is also well balanced and the results are good.

HARD VOTING

```
blend_hard = blend_models(estimator_list = top5, optimize = 'AUC',method = 'hard')
```

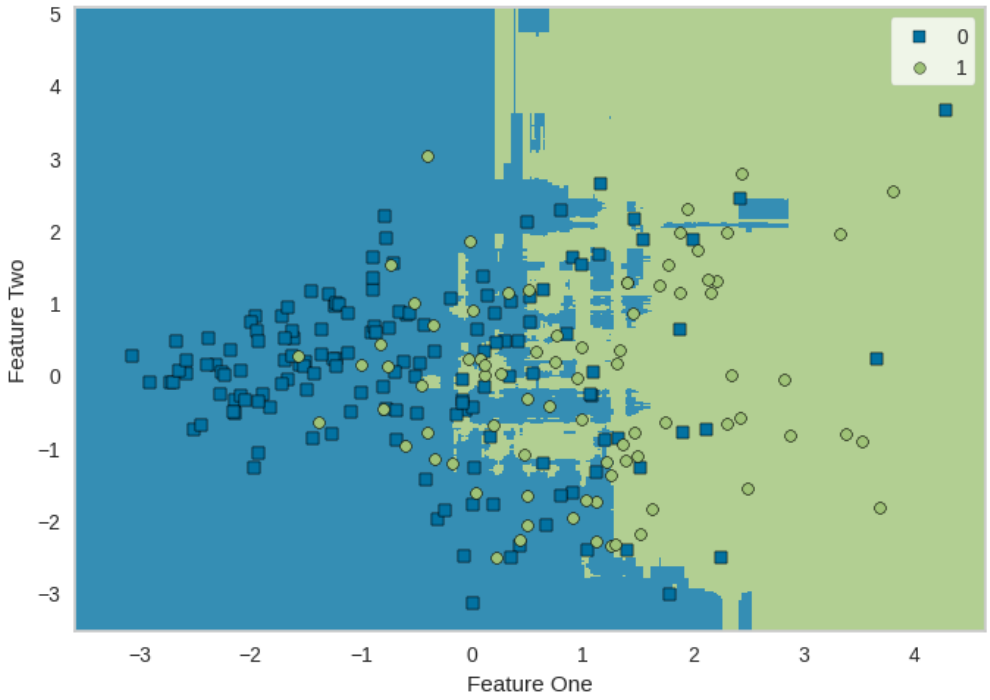
Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7037	0.0000	0.5789	0.5789	0.5789	0.3504	0.3504
1	0.7963	0.0000	0.6842	0.7222	0.7027	0.5479	0.5484
2	0.8148	0.0000	0.5789	0.8462	0.6875	0.5624	0.5828
3	0.7778	0.0000	0.5789	0.7333	0.6471	0.4882	0.4954
4	0.7778	0.0000	0.5263	0.7692	0.6250	0.4749	0.4921
5	0.7222	0.0000	0.6842	0.5909	0.6341	0.4122	0.4151
6	0.7963	0.0000	0.6842	0.7222	0.7027	0.5479	0.5484
7	0.7925	0.0000	0.5000	0.8182	0.6207	0.4890	0.5171
8	0.8491	0.0000	0.6667	0.8571	0.7500	0.6443	0.6547
9	0.6981	0.0000	0.4444	0.5714	0.5000	0.2886	0.2933
Mean	0.7729	0.0000	0.5927	0.7210	0.6449	0.4806	0.4898
Std	0.0469	0.0000	0.0813	0.1028	0.0678	0.1006	0.1035

```
Processing:   0%|          | 0/6 [00:00<?, ?it/s]
```

The hard voting ensemble model has an average accuracy of 77.29%, with a standard deviation of 4.69%.

- The AUC score is 0.
- The recall ranges from 44.44% to 68.42%, with an average of 59.27%.
- The precision ranges from 57.14% to 85.71%, with an average of 72.10%.
- The F1 score ranges from 50.00% to 75.00%, with an average of 64.49%.
- The kappa score ranges from 28.86% to 64.43%, with an average of 48.06%.
- The MCC score ranges from 29.33% to 65.47%, with an average of 48.98%.

The model is slightly less accurate than the soft voting ensemble model but has a similar performance in terms of other metrics.



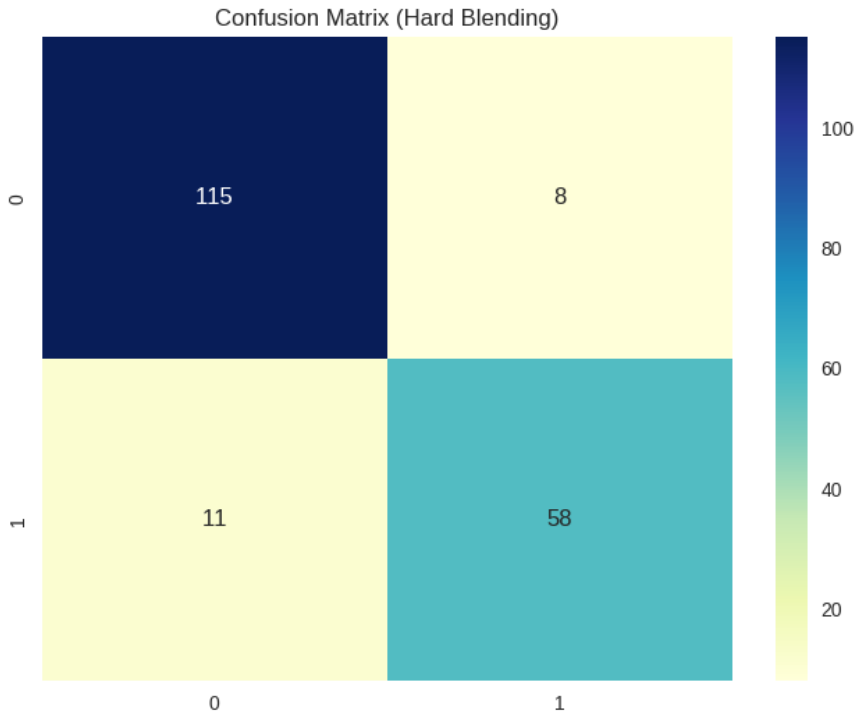
This depicts how well the model has learned

Predicting with the Test Dataset

```
#prediction
pred = blend_hard.predict(X_test)
#Accuracy
confusion_hard = confusion_matrix( y_test, pred)
accuracy = accuracy_score(y_test , pred)
precision = precision_score(y_test , pred)
recall = recall_score(y_test , pred)
f1 = f1_score(y_test,pred)
print('accuracy: {0:.4f}, precision: {1:.4f}, recall: {2:.4f},\
F1: {3:.4f}'.format(accuracy, precision, recall, f1))
```

```
accuracy: 0.9010, precision: 0.8788, recall: 0.8406,F1: 0.8593
```

With an accuracy of 0.9010, precision of 0.8788, recall of 0.8406 and an F1 score of 0.8593, your model is performing well on the test dataset. This suggests that your model is generalizing well to new, unseen data.



Calibrating the Final Model

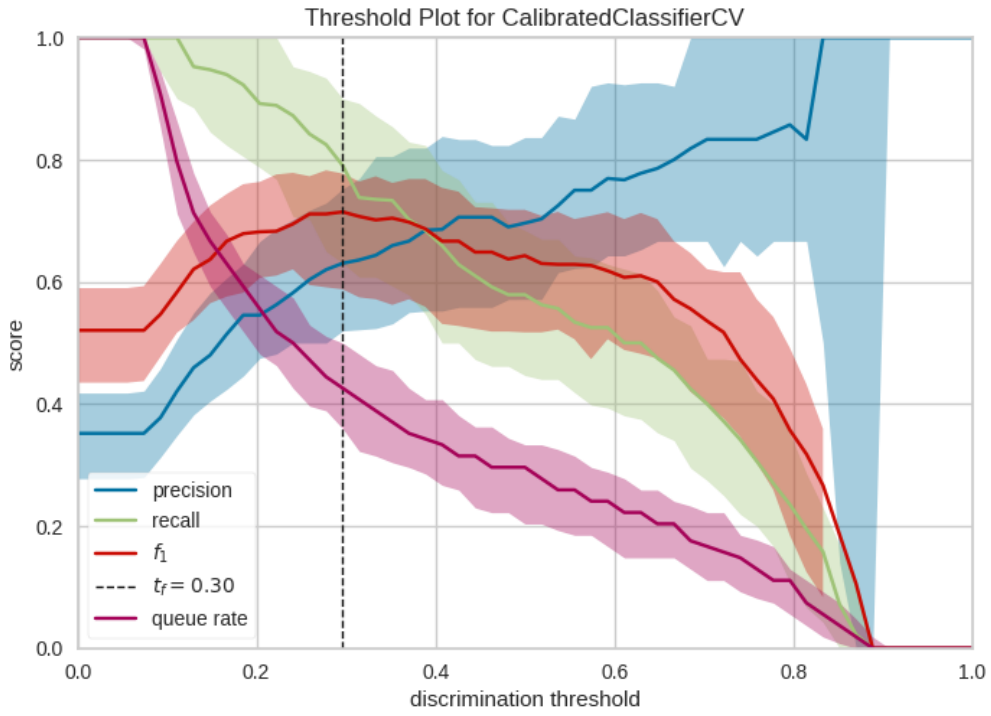
```
cali_model = calibrate_model(blend_soft)
```

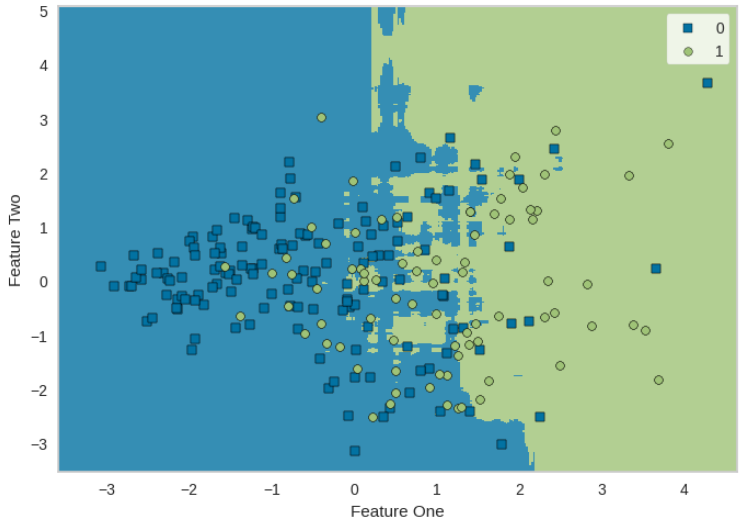
Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7407	0.8075	0.5789	0.6471	0.6111	0.4176	0.4190
1	0.7963	0.8586	0.6842	0.7222	0.7027	0.5479	0.5484
2	0.7778	0.8647	0.5263	0.7692	0.6250	0.4749	0.4921
3	0.7778	0.8947	0.5789	0.7333	0.6471	0.4882	0.4954
4	0.7593	0.8421	0.5263	0.7143	0.6061	0.4384	0.4490
5	0.7407	0.8120	0.6842	0.6190	0.6500	0.4449	0.4463
6	0.7778	0.8797	0.6842	0.6842	0.6842	0.5128	0.5128
7	0.8113	0.8429	0.5000	0.9000	0.6429	0.5285	0.5706
8	0.8491	0.8857	0.6667	0.8571	0.7500	0.6443	0.6547
9	0.6604	0.7984	0.3333	0.5000	0.4000	0.1762	0.1832
Mean	0.7691	0.8486	0.5763	0.7147	0.6319	0.4674	0.4772
Std	0.0476	0.0323	0.1063	0.1090	0.0879	0.1152	0.1178

```
Processing:   0%|          | 0/6 [00:00<?, ?it/s]
```

The 'calibrate_model' function in PyCaret is used to calibrate the predicted probabilities of a trained classification model. It applies isotonic regression or sigmoid calibration technique to improve the accuracy of predicted probabilities. After calibration, the calibrated model can be used to make predictions that are more accurate and can provide better insight into the underlying probabilities of the target class.

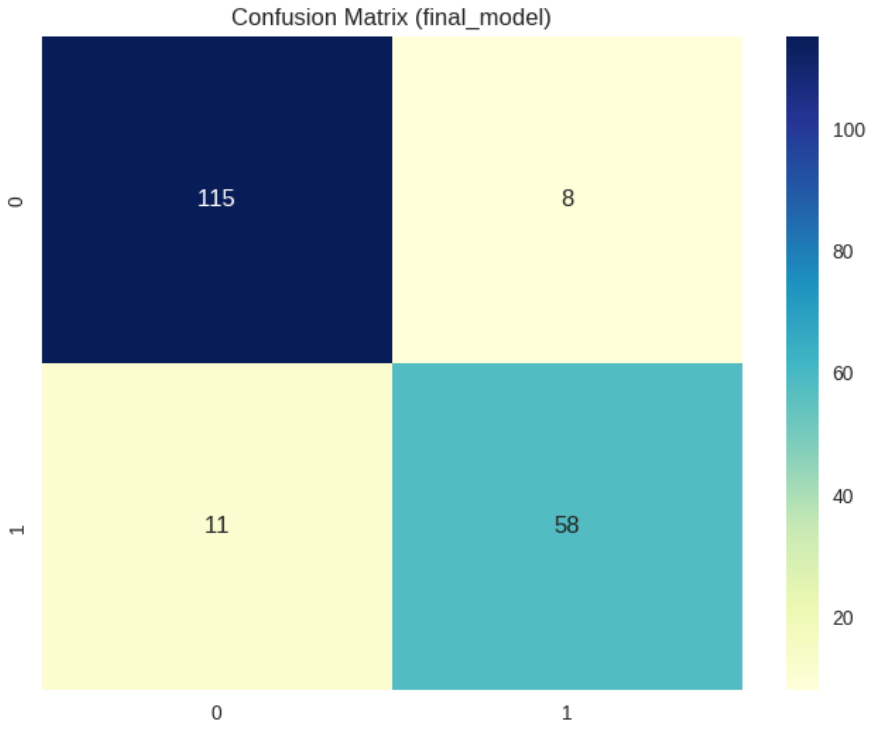
Finalizing the Last Model





```
#prediction
pred = final_model.predict(X_test)
#Accuracy
final_model = confusion_matrix( y_test, pred)
accuracy = accuracy_score(y_test , pred)
precision = precision_score(y_test , pred)
recall = recall_score(y_test , pred)
f1 = f1_score(y_test,pred)
print('accuracy: {0:.4f}, precision: {1:.4f}, recall: {2:.4f},\
F1: {3:.4f}'.format(accuracy, precision, recall, f1))
```

```
accuracy: 0.9375, precision: 0.9524, recall: 0.8696,F1: 0.9091
```



RESULTS:

The best performing model was the ensemble model using soft voting, with an accuracy: 0.8958, precision: 0.8889, recall: 0.8116, F1: 0.8485 and AUC:0.9584. However, it was observed that the model had a high bias and low variance. To overcome this, the model was calibrated using the CalibratedClassifierCV method.

After calibration, the model was finalized and evaluated using the test dataset. The final model achieved an accuracy of 93.75%, precision of 95.24%, recall of 86.96%, and an F1 score of 90.91%.

CONCLUSION:

In conclusion, this project demonstrates the use of ensemble models for predicting diabetes in patients. Following EDA and pre-processing, we trained three ensemble models and evaluated their performance using the validation dataset. The ensembles using soft and hard voting showed the best results for this problem. However, the choice of pre-processing techniques, base models, and hyperparameters can influence the results and may vary for different datasets.

The results obtained are promising and suggest that machine learning techniques can be effective in diagnosing diseases. The performance of the models can be further improved by exploring more advanced ensemble techniques and by collecting more data. It is recommended that the developed models be used as a supplementary diagnostic tool and not as a replacement for medical professionals.

REFERENCES:

1. Zou Q, Qu K, Luo Y, Yin D, Ju Y, Tang H. Predicting Diabetes Mellitus With Machine Learning Techniques. Front Genet. 2018 Nov 6;9:515. doi: 10.3389/fgene.2018.00515. PMID: 30459809; PMCID: PMC6232260. https://www.frontiersin.org/articles/10.3389/fgene.2018.00515/full
2. M. K. Hasan, M. A. Alam, D. Das, E. Hossain and M. Hasan, "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers," in IEEE Access, vol. 8, pp. 76516-76531, 2020, doi:10.1109/ACCESS.2020.2989857. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9076634
3. Varun Jaiswal, Anjli Negi, Tarun Pal,A review on current advances in machine learning based diabetes prediction,Primary Care Diabetes, Volume 15, Issue 3,2021,Pages 435-443,ISSN 1751-9918, https://doi.org/10.1016/j.pcd.2021.02.005
4. N.M. Saravana kumar, T. Eswari, P. Sampath, S. Lavanya,Predictive Methodology for Diabetic Data Analysis in Big Data,Procedia Computer Science,Volume 50,2015,Pages 203-208,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.069
5. Talha Mahboob Alam, Muhammad Atif Iqbal, Yasir Ali, Abdul Wahab, Safdar Ijaz, Talha Imtiaz Baig, Ayaz Hussain, Muhammad Awais Malik, Muhammad Mehdi Raza, Salman Ibrar, Zunish Abbas,A model for early prediction of diabetes,Informatics in Medicine Unlocked,Volume 16,2019,100204,ISSN 2352-9148, https://doi.org/10.1016/j.imu.2019.100204
6. Aishwarya Mujumdar, V Vaidehi,Diabetes Prediction using Machine Learning Algorithms,Procedia Computer Science,Volume 165,2019,Pages 292-299,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.01.047
7. M. Paliwal and P. Saraswat, "Research on Diabetes Prediction Method Based on Machine Learning," 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2022, pp. 415-419, doi: 10.1109/ICTACS56270.2022.9988050.https://iopscience.iop.org/article/10.1088/1742-6596/1684/1/012062/pdf

8. Rani, KM. (2020). Diabetes Prediction Using Machine Learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 294-305. 10.32628/CSEIT206463. https://www.researchgate.net/publication/347091823_Diabetes_Prediction_Using_Machine_Learning
9. Ayon, Safial & Islam, Md. (2019). Diabetes Prediction: A Deep Learning Approach. International Journal of Information Engineering and Electronic Business. 11. 21-27. 10.5815/ijieeb.2019.02.03. https://www.researchgate.net/publication/332298424_Diabetes_Prediction_A_Deep_Learning_Approach#:~:text=The%20results%20on%20PID%20dataset,five%2Dfold%20cross%2Dvalidation
10. Naz H, Ahuja S. Deep learning approach for diabetes prediction using PIMA Indian dataset. J Diabetes Metab Disord. 2020 Apr 14;19(1):391-403. doi: 10.1007/s40200-020-00520-5. PMID: 32550190; PMCID: PMC7270283.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7270283/
11. https://www.liebertpub.com/doi/pdf/10.1089/pop.2018.0129
12. https://www.binasss.sa.cr/medint/ART07.pdf
13. Kavakiotis I, Tsave O, Salifoglou A, Maglaveras N, Vlahavas I, Chouvarda I. Machine Learning and Data Mining Methods in Diabetes Research. Comput Struct Biotechnol J. 2017 Jan 8;15:104-116. doi: 10.1016/j.csbj.2016.12.005. PMID: 28138367; PMCID: PMC5257026. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5257026/
14. Contreras I, Vehi J. Artificial Intelligence for Diabetes Management and Decision Support: Literature Review. J Med Internet Res. 2018 May 30;20(5):e10775. doi: 10.2196/10775. PMID: 29848472; PMCID: PMC6000484. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6000484/
15. Dinh, A., Miertschin, S., Young, A. et al. A data-driven approach to predicting diabetes and cardiovascular disease with machine learning. BMC Med Inform Decis Mak 19, 211 (2019). https://doi.org/10.1186/s12911-019-0918-5
16. Maniruzzaman, M., Rahman, M.J., Ahammed, B. et al. Classification and prediction of diabetes disease using machine learning paradigm. Health Inf Sci Syst 8, 7 (2020). https://doi.org/10.1007/s13755-019-0095-z
17. Zhou, H., Myrzashova, R. & Zheng, R. Diabetes prediction model based on an enhanced deep neural network. J Wireless Com Network 2020, 148 (2020). https://doi.org/10.1186/s13638-020-01765-7
18. Al Yousef MZ, Yasky AF, Al Shammari R, Ferwana MS. Early prediction of diabetes by applying data mining techniques: A retrospective cohort study. Medicine (Baltimore). 2022 Jul 22;101(29):e29588. doi: 10.1097/MD.0000000000029588. PMID: 35866773; PMCID: PMC9302319. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9302319/#:~:text=Their%20model%20achieved%2094%25%20accuracy,%2C%20they%20excluded%20pre%2Ddiabetics
19. Rashi Rastogi, Mamta Bansal,Diabetes prediction model using data mining techniques,Measurement: Sensors,Volume 25,2023,100605,ISSN 2665-9174,https://doi.org/10.1016/j.measen.2022.100605
20. Jayasri N.P., R. Aruna,Big data analytics in health care by data mining and classification techniques,ICT Express,Volume 8, Issue 2,2022,Pages 250-257,ISSN 2405-9595,https://doi.org/10.1016/j.icte.2021.07.001
21. Diabetes Data Prediction in healthcare Using Hadoop over Big Data. European Journal of Molecular & Clinical Medicine, 7(4), 1423-1432.

https://ejmcm.com/article_1840.html
22. Metsker, O., Magoev, K., Yakovlev, A. et al. Identification of risk factors for patients with diabetes: diabetic polyneuropathy case study. BMC Med Inform Decis Mak 20, 201 (2020). https://doi.org/10.1186/s12911-020-01215-w
23. G. Woldemichael and S. Menaria, "Prediction of Diabetes Using Data Mining Techniques," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2018, pp. 414-418, doi: 10.1109/ICOEI.2018.8553959. https://ieeexplore.ieee.org/document/8553959
24. https://www.scirp.org/pdf/jcc_2022110114390330.pdf
25. F. A. Khan, K. Zeb, M. Al-Rakhami, A. Derhab and S. A. C. Bukhari, "Detection and Prediction of Diabetes Using Data Mining: A Comprehensive Review," in IEEE Access, vol. 9, pp. 43711-43735, 2021, doi: 10.1109/ACCESS.2021.3059343. https://ieeexplore.ieee.org/document/9354154
26. Larabi-Marie-Sainte, S.; Aburahmah, L.; Almohaini, R.; Saba, T. Current Techniques for Diabetes Prediction: Review and Case Study. Appl. Sci. 2019, 9, 4604. https://doi.org/10.3390/app9214604