UNIVERSITY OF HERTFORDSHIRE
School of Physics, Engineering and Computer Science

MSc in Data Science and Analytics with Advanced Research
7COM1039-0109-2023 - Advanced Computer Science Master's Project
19/08/2024

DIABETES RISK PREDICTION USING MACHINE LEARNING MODELS

Name: Preethi Vantala
Student ID: 19049291
Supervisor: Prof. Uchenna Ojiako

Abstract

Diabetes is a chronic condition characterized by high levels of sugar in the blood; if not managed early, then it leads to serious health complications. The aim of this study is to improve the early prediction of diabetes using advanced machine learning models on a large dataset containing 100,000 instances. Algorithms used to evaluate this study included some of the traditional and advanced machine learning algorithms such as Logistic Regression, Random Forest, Decision Tree, Gaussian Naïve Bayes, Gradient Boosting, K-Nearest Neighbours, some Ensemble methods like Ensemble stacking with GridSearchCV. The results raise some pertinent questions regarding diabetes prediction with respect to data quality, feature selection and ethical concerns. In the estimation, the effectiveness of the model is measured with metrics such as accuracy, precision, sensitivity, specificity, F1-Score, ROC-AUC score and ROC-AUC curve. The research was on how generally the Ensemble techniques can turn out with better predictive accuracy and make the results more robust.

Acknowledgement

This research “Diabetes Risk Prediction using Machine learning Models” has been successfully completed. I would like to express my sincere appreciation to my supervisor who gave me the best inputs and the required guidelines during this assignment. Feedback given by my supervisor for the research idea and the interim report were so much helpful to complete this research successfully. I would like to thank my supervisor and everyone who supported me emotionally and contributed this project in one or other way, making it a best experience of my career.

MSc Final Project Declaration

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science and Analytics with Advanced Research at the University of Hertfordshire (UH).

It is my own work except were indicated in the report.

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on the university website provided the source is acknowledged.

Preethi Vantala

19049291.

Table of Contents

Chapter 1: Introduction

Chapter 2: Literature Review

Chapter 3: Methodology

Chapter 4: Results

Chapter 5: Result Analysis

Chapter 1. Introduction

1.1 Diabetes Background:

Diabetes is a debilitating and pervasive chronic disease that affects millions of people globally and poses a considerable challenge to any healthcare system. It is characterized by consistently high blood sugar levels which can result either from poor utilisation of insulin by the body or insufficient insulin production by the pancreas. It is a chronic, metabolic disease that ravages the eyes, kidneys, heart, blood vessels, and nerves due to increase in blood glucose levels. Globally, it has become a growing epidemic of concern. In 2014, 8.5% of adults 18 years and over were affected by it. In 2019, diabetes was responsible for 1.5 million deaths, 48% of these deaths occurred before the age of 70. A target is set for the global community to achieve a relative reduction in the rise of the prevalence of diabetes and obesity by 2025.

There are broadly three types of diabetes: Type 1, Type 2, and gestational diabetes. Type 1 diabetes, otherwise known as autoimmune diabetes, is a chronic condition characterised by insulin deficiency from pancreatic β-cell loss that causes hyperglycaemia (Katsarou & Gudbjornsdottir 2017). Type 2 diabetes is one of the most common emerging global health issues associated with obesity and has been implicated in macrovascular complications, including cardiovascular comorbidities, as well as in microvascular problems, including retinopathy, nephropathy, and neuropathy (DeFronzo & Ferrannini 2015). Gestational diabetes is diagnosed by glucose intolerance during pregnancy, which normally resolves after delivery but predisposes the person to Type 2 diabetes later (Buchanan and Xiang 2015) Change in the fast pace of global diabetes prevalence is due to lifestyle changes, sedentary living, and increased ageing populations, this underlines early detection, prevention, and treatment methods against the disease.

Early detection and timely intervention are very important in the management of diabetes and in preventing complications. Machine learning, with its increasing applications in healthcare, can tremendously improve early diagnosis. Algorithms offered by machine learning, though not standard in clinical practice, have the capability for fast pattern identification and detection of risk factors in a patient's data even more precisely than traditional means. This would enable earlier identification of patients, individual treatment planning, and prognosis. While a good number of studies have been made on machine learning in diabetes diagnosis, many methods still have some limitations. Some produced very positive results with multi-metric evaluation for model effectiveness, while others have exposed their weaknesses.

1.2 Machine learning:

Machine learning (ML) is a field of computer science and a subset of artificial intelligence (AI), which involves computers learning automatically from data. Computers can perform tasks without being explicitly programmed. It is also known as predictive analytics because it is used to make predictions using the given data. In healthcare systems, machine learning is offering advanced techniques to analyse complex datasets and uncover patterns that the traditional ML algorithms might miss (Russell & Norvig, 2016). ML can process vast amounts of data and provide insights into the diagnosis, progression, and treatment of diabetes. Advanced ML detects patients who are suffering from diabetes. This, in turn, helps the health care system take precautions to reduce the risk of diabetes. Machine learning is categorised into three types: 1) supervised learning, 2) unsupervised learning, and 3) reinforcement learning. Each one has its own purpose in data modelling and predictive analysis.

Basically, supervised machine learning is within the domain of machine learning, which uses a training dataset to make predictions. For this, training data processes and builds a function that maps against new data with respect to an expected output value. It's widely used in marketing, finance, health care, and other industries. Some of the supervised machine learning algorithms include random forest, decision trees, and logistic regression. In the diabetes dataset, a decision tree can pinpoint attributes like physical inactivity and family history as significant predictors of diabetes (Breiman et al., 1984). Random forest algorithms work with multiple decision trees, combining their predictions, which increases the accuracy and robustness of the model. It particularly balances the variance and the bias trade-off in the prediction of diabetes (Breiman, 2001). Logistic regression uses a logistic model to predict binary outcome variables and is widely used to estimate the probability of diabetes using risk factors such as BMI, age, and family history (Hosmer, Lemeshow, & Sturdivant, 2013).

Unsupervised machine learning is a type of ML algorithm that examines and clusters unlabelled datasets to find the hidden patterns without human intervention. Since there is no corresponding output data, unlike supervised machine learning, unsupervised learning cannot be directly applied to any classification or regression problem. These algorithms are useful to find patients with similar characteristics and reduce the dimensionality of data. Clustering algorithms like K-means clustering can be utilized to group patients by similar characteristics, which help in formulating personalized treatment plans and providing the corresponding resources (MacQueen, 1967). Principal component analysis is used to reduce the dimensions of large datasets that preserve essential features of datasets. PCA transforms these large complex medical datasets into a simpler version that will be much easier to visualize and interpret data (Jolliffe, 2002).

Reinforcement learning is a type of ML where the agent learns to make decisions by interacting with the environment and receives feedback in the form of penalties and rewards. In dynamic and complex environments, a trial-and-error method is used. Considering diabetes management, reinforcement learning can be used to continuously adjust the dosage based on glucose levels to optimise the insulin therapy. Patient's response to different insulin doses is collected and then used by the algorithm to personalise the treatment and improve the glycaemic control (Sutton & Barto, 2018).

1.3 Current Issues in chosen area:

Diabetes prediction using Machine learning (ML) has made significant strides in recent years, offering new techniques, early diagnosis and treatment. To enhance the reliability and effectiveness of machine learning models in this field there are few current issues and challenges that need to be addressed. Few of the issues and their feasibility are explored and supported by academic sources, as outlined below.

(a) Data Quality and Availability:

Real-world medical data are usually filled with missing values, and these missing values have a strong influence on machine learning models (Jensen et al., 2012). In this respect, handling missing data is of prime importance to come up with robust models. Multiple imputation or other advanced methods like k-nearest neighbours may enhance data quality (Little & Rubin, 2019). Class imbalance is another typical characteristic for diabetes datasets, in which the non-diabetic cases dominate the diabetic ones. This can bias models towards the majority class. In this paper, the SMOTE technique has been used to handle the imbalance of data.

(b) Feature selection and engineering:

Feature selection and engineering identify relevant characteristics from genetics, lifestyle, medical records, etc. Irrelevant or redundant features can harm model performance. Continuous features can be encoded effectively, and data preprocessing libraries in Python (e.g., scikit-learn) and R make standardization straightforward (Pedregosa et al., 2011). Categorical features can be encoded using techniques like one-hot encoding and label encoding.

(c) Ethical and Privacy Concerns:

When handling health data, ethical and privacy concerns should be treated with great magnitude. This checks on the uses of the data for ethics and keeps patient data confidential. Differential privacy and other techniques, such as de-identification, might be used in protecting the patients' data while it's being used for model training (Dwork & Roth, 2014). According to a study by (Vayena et al., 2018) developing frameworks and guidelines for ethical use of ML in healthcare can be helpful to address the concerns that are related to fairness, bias and accountability. The dataset used for this research is also taken from Kaggle which is an open-source platform.

(d) Technical Infrastructure:

It is well known that developing and training machine learning models requires enormous processing power. This may be derived through scalable resources using cloud-based systems such as Microsoft Azure, AWS, or Google Cloud. Machine learning models are then developed by certain Python libraries like TensorFlow, PyTorch, scikit-learn, among many others. Custom prototyping and experimentation are allowed to be performed by tools like Jupyter Notebooks.

(e) Model Performance and validation:

The performance and validation of the model are continuous challenges in machine learning for diabetes prediction. Considering only accuracy for the efficiency of the model can mislead its statement. Applied ML models should ensure good performance across various metrics. The appropriate metrics to be used in evaluating the models include Accuracy, Precision, Recall, and the ROC-AUC curve, showing the performance on the data. (Huang & Ling, 2005).

1.4 Originality of the study:

Recent past studies have been orientated towards prediction of diabetes using machine learning and which model would work better in finding an early diagnosis. Most of the research relies on traditional machine learning models like Random Forest, Support Vector Machines, Decision Trees, and Logistic Regression, comparing the performance metrics for the ideal model. A few others have ventured into deep learning and ensemble methods to improve the models and attain better prediction accuracy from the data. Most of the works in this area, however, have been very narrowed in scope, which focusses on accuracy only, using limited data sets. A variety of metrics in performance are compared in this study using a large dataset from diabetes with key features, the most advanced machine learning techniques, and best data preprocessing to identify the best solution for an early prediction of diabetes.

After going through different previous existing research about diabetes prediction using machine learning algorithms, there remains a significant gap in evaluating and integrating the diverse models and metrics to enhance prediction accuracy and reliability. All these previous studies have primarily focused on comparing a limited set of ML algorithms and performance metrics. They mostly focused on accuracy as the primary metric, which can be misleading, but in medical diagnosis problems like diabetes prediction datasets often have an imbalance. Additionally, most of these studies have implemented these algorithms on datasets that have a limited number of instances.

The objective of this study is to apply and compare a full range of machine learning algorithms, including logistic regression, decision trees, random forests, gradient boosting, Gaussian Naive Bayes, KNN, ensemble stacking, and ensemble stacking with GridSearchCV. This research used data with 100,000 instances to provide higher relevance to the results. Another important aspect of the proposed framework is to analyse and explain overall model performance through accuracy, precision, sensitivity, specificity, F1-score, and the ROC-AUC curve. This paper uses ensemble stacking in predicting accuracy and stability in early diabetes diagnoses by amalgamating the strengths of different algorithms.

The aim is to construct predictive models with the most sophisticated machine learning algorithms, such as Ensemble methods, for superior predictive performance compared to the traditional ones. Incorporating multiple evaluation metrics would detail the performance of a model instead of sticking to a single metric. Large data and a focus on practical implementation ensure findings presented are relevant and would apply to real-world healthcare settings.

This work is important in the progression towards early diagnosis of diabetes in that some of the failures of prior studies have been addressed. By using traditional and advanced machine learning algorithms with a wide range of performance metrics being evaluated on a larger dataset, it increases model robustness and performance. The results, therefore, could be used in developing treatment programs at an individual level, correctly stratifying patients by risk levels. The use of a large dataset proves that these models can scale across different populations and diverse healthcare settings. It thus has the potential to decrease the financial burden on patients and their families by bringing offers of appreciable savings to the management in health systems and insurance companies, hence improving health outcomes and increasing efficiency in healthcare management and public health.

1.5 Research Aim and Objectives:

Aim:

To develop and evaluate advanced machine learning models on a large dataset for early prediction of diabetes, which improves predictive accuracy and facilitates early intervention strategies in healthcare.

Objectives:

- Performance comparison and evaluation of Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Gaussian Naive Bayes, KNN, Ensemble Stacking, and Ensemble Stacking with GridSearchCV in diabetes prediction by a set of comprehensive metrics: accuracy, precision, sensitivity, specificity, F1-score, ROC-AUC score, correlation matrix, and ROC-AUC curve.
- Implementing the ensemble stacking method with GridSearchCV, which combines the strengths of multiple algorithms, in search of higher predictive performance and robustness against traditional individual machine learning models.

1.6 Research Questions:

1. Compared with individual machine learning algorithms, how does the predictive accuracy and robustness increase using ensemble stacking with GridSearchCV in diabetes prediction?
2. How does the use of a broader range of performance metrics contribute to a more nuanced understanding of model performance in diabetes prediction?
3. How do traditional machine learning algorithms compare with advanced machine learning techniques for early detection of diabetes using a large dataset?

The various machine learning algorithms used in this research will be evaluated using a comprehensive set of performance metrics. This study is an attempt at bridging gaps identified in previous studies by offering the best model performance using advanced machine learning techniques. This study does not deal with biological aspects related to diabetes. No experimentation trails on humans are done in this research. It is only concerned about predictive modelling and performance evaluation based on existing data. The technical aspects of the comparison of models and performance metrics are focused, but the implementation costs, the policy implications, or broader economic impacts of diabetes prediction models in healthcare systems are not mentioned in this research.

1.7 Research Approach:

- Choose a dataset containing 100,000 instances related to diabetes prediction.
- Apply all preprocessing techniques to handle missing and null values, normalising features, and treating the imbalance in the class.
- Evaluate traditional and advanced machine learning algorithms, including Logistic Regression (LR), Decision Tree, Random Forest, KNN, Gradient Boost, Gaussian Naive Bayes, Ensemble Stacking, and Ensemble Stacking with GridSearchCV.
- Split the dataset into training and test datasets.
- Train the model on the training dataset and use the testing dataset to predict the values.
- Performance of model evaluation using performance metrics like accuracy, precision, sensitivity, specificity, F1-Score, ROC-AUC score, and ROC-AUC curve.
- Implement the ensemble stacking model that combines predictions obtained from different individual models.
- Check whether the ensemble stacking performances are higher in comparison with the other applied models.
- Ensemble Stacking GridSearchCV: Hyperparameters fine-tuning using GridSearchCV and model power enhancement.
- Compare the performances of each model along with ensemble stacking with GridSearchCV and decide the best model.
- Discussing the limitations of the study, practical implementations, and scope of future research.

1.8 Overview of Core chapters:

Chapter 1: Introduction

This chapter presents an overview of diabetes, its types, and the role of machine learning in prediction. It also covers the importance of early detection, complications, treatment, and management of diabetes. Traditional and advanced ML models were briefly introduced. The issues at hand are presented, together with their feasibility, along with ethical concerns such as data privacy. This chapter also delineated the purpose, objectives, questions the study sought to answer, and novelty of the study in comparison with existing studies.

Chapter 2: Literature Review:

The second chapter discusses literature related to the use of machine learning techniques for the prediction of diabetes. This review encompasses the ML models used, their results, methodologies, and limitations. Further, it is going to portray how, overtime, the ML techniques have evolved and what new improvements have been made to make such models more predictive and reliable.

Chapter 3: Methodology

It presents the methodology and design of the research, covering data collection, pre-processing, model selection, model training, and performance comparison. This includes models used and implemented and why the selected ML algorithm was employed. Further, the chapter also discusses the benefits associated with the different machine learning models that predict diabetes and the lessons obtained in developing more robust models for future studies.

Chapter 4: Results

This chapter represents the results of the study in a descriptive manner. It includes the summary statistics of the dataset. Visualisation of data distributions, tables and figures summarising the model performance, and results of ML models like accuracy, precision, recall, specificity, F1-Score and ROC-AUC curve are included in this chapter.

Chapter 5: Discussion

The analysis and interpretation of the results are provided in this chapter. Analysis of the performance of ML models, strengths and weaknesses of the models are discussed. How the results of this research are related to the aim, objective, and research questions provided in Chapter 1 are included. The differentiation between your results and existing studies is clearly explained.

Chapter 6: Conclusion and Recommendations

Including the key findings of the research and their implications in this chapter. A summary of the main findings, research objectives, and conclusions drawn from the analysis and results are provided here. The limitations for the models used in the study and the recommendations for future research are included in this chapter.

Chapter 2: Literature Review

2.1 Introduction:

There are many existing studies that are related to diabetes prediction using machine learning algorithms, but still, the new studies are being conducted to detect diabetes in its early stages so that people can take the preventions. Going through different studies about diabetes prediction is much necessary for this research because it allows to find what methods they have used to detect the diabetes and additionally what other methods can be used in this research. Going through existing studies gives an overall idea of what other approaches can be made in this research to predict diabetes at its early stages and how this research can be beneficial to healthcare systems. A literature review is necessary to highlight the research gap and tell exactly what this research would be investigating about and what makes this research unique from the existing studies.

Using different search resources like IEEE, Google Scholar, and many more, it has been found that there are numerous studies related to diabetes prediction using machine learning algorithms. The studies related to this research are increasing rapidly. Multiple studies focused on detecting diabetes with high accuracy by applying different ML algorithms and comparing their performance.

2.2 Studies on Traditional ML Algorithms:

Aishwarya Mujumdar, V Vaidehi Dr. (2019) proposed a diabetes prediction model using 12 individual machine learning algorithms on two different datasets. Comparison among models is done through performance metrics like accuracy, precision, recall, and F1-score. They have boosted the classification accuracy with the new dataset compared with the existing dataset.

KM Jyoti Rani (2020) developed a system for early prediction of diabetes using ML algorithms including K-Nearest Neighbour, Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine (SVM). The performance of the models is measured using only accuracy. Only a few ML models are compared, and the study sticks with only one performance metric.

2.3 Studies on Advanced ML Algorithms:

Unlike the previous studies Mitushi Soni and Dr. Sunita Varma (2020), in their work, used advanced ML techniques like ensemble bagging and gradient boosting ensemble on their dataset to predict the diabetes. Ensemble technique combines the strengths of different ML models. Their research achieved an accuracy of 77%, which is less compared to a few other studies. The dataset used for this study contains only 768 instances, which is comparatively less. Comparing only the accuracy among ML algorithms is not sufficient to effectively predict the disease.

K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan (2020) have proposed a robust framework for predicting diabetes. They implemented all the necessary preprocessing steps, like filling in missing values in the dataset, standardising the data, appropriate feature selection, and applying different ML algorithms along with an ensemble model. The performance metric ROC curve (AUC) is used in this research. Their results state that the ensemble classifier was the best classifier, with other performance metrics like sensitivity and specificity being good. They have done their best by using the advanced ML techniques and considering the different performance metrics other than accuracy, but even they used the dataset, which has only 768 instances. Models performed on datasets that have more data are more likely to achieve high accuracy and precision.

2.4 Studies on Deep Learning Methods:

There are few studies that even used neural networks (NN) methods along with machine learning algorithms for early detection of diabetes. Jobeda Jamal Khanam and Simon Y. Foo (2021) worked on a diabetes dataset, which also has only 768 instances. They have applied seven different machine learning algorithms individually and concluded that logistic regression (LR) and support vector machines (SVM) work well on predicting diabetes. They worked on the NN model, which has different hidden layers, and as a result, they got an accuracy of 88.6% with the neural network layer, which has two hidden layers. Precision, recall, F-measure, and accuracy are the performance metrics that are considered in this research study.

Jobeda Jamal Khanam and Simon Y. Foo (2021) even explained about many studies that have used the Prime Indian Diabetes dataset (PIDD), which has 768 records, and applied machine learning algorithms to predict the diabetes. They argued that there is a problem with such studies in choosing the logical features and the classifier. They proved that using Pearson’s correlation method in their work to find the logical features can exactly predict if a patient has diabetes or not. They applied ML algorithms and used neural networks (NN) with hidden layers to improve the overall performance.

2.5 Consistency of Existing Studies:

A comparative study and review based on popular machine learning algorithms and additionally on ontology-based machine learning classification was done by H. E. Massari, Z. Sabouri, S. Mhammedi, and N. Gherabi (2022). They agreed with all the previous studies, which have compared different ML algorithms to predict diabetes. They even compared SVM, KNN, ANN, Naive Bayes, Logistic Regression, and Decision Tree, and additionally they applied ontology-based machine learning and proved that the best results were achieved by SVM and ontology classifiers.

To predict the diabetes based on several health attributes in the dataset using supervised machine learning, M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda (2023) proposed a model that performs a comparison between K-Nearest Neighbours and Naive Bayes algorithms. They applied these two algorithms to the dataset and concluded that the Naive Bayes algorithm has outperformed KNN. They compared the performance metrics accuracy, precision, and recall between two algorithms. Accuracy of 76.07% was achieved by Naive Bayes. The dataset used in this research is also the Prime Indian Diabetes Database, which has a smaller number of records.

2.6 Limitations of Existing Studies:

The main aim of this research is to apply different machine learning techniques to a large dataset, evaluating different performance metrics, and finding which would be the best technique that can predict diabetes in its early stages. There are several existing studies related to this diabetes prediction, and going through these studies, it was found that they may not have adequately explored a few concepts. Some of the studies, like Aishwarya Mujumdar, V Vaidehi Dr (2019), and KM Jyoti Rani (2020), have perfectly applied traditional machine learning algorithms (SVM, Random Forest, Logistic Regression, etc.) to datasets, compared them, and concluded that the model with the best accuracy is the one that predicts diabetes in its early stages. They applied only the traditional ML algorithms, and KM Jyoti Rani (2020) stuck with only one performance metric called accuracy, which can be misleading. Mitushi Soni and Dr. Sunita Varma (2020) have applied advanced machine learning techniques like ensemble bagging to combine the individual ML models to increase the accuracy. The dataset used could have been larger for more robustness and generalizability. Unlike these few studies that applied neural networks and compared the accuracy, they worked on a smaller dataset. If they have considered a dataset containing more records, they have missed using advanced techniques that predict diabetes at early stages. Few of them have chosen good datasets and advanced techniques but failed to explore a greater number of performance metrics, which was not so efficient.

2.7 Filling the Research Gap:

This research attempts to fill the gaps in existing studies by designing a study that would address the limitations of previous work. This paper uses a large dataset consisting of 100,000 instances and applies comprehensive preprocessing, feature selection, and standardisation techniques to the dataset to make sure the data is optimally prepared. Then, traditional machine learning algorithms and some advanced techniques like ensemble stacking are applied. Performance metrics like accuracy, precision, sensitivity, specificity, F1-score, ROC-AUC score, and the correlation matrix are calculated for each algorithm.

The objective of this study is to demonstrate that better results come out of much more advanced techniques, which provide firm and reliable modelling for the early detection of diabetes. The performance metrics are almost exhaustively considered in this study to deeply explain model performance and its consequences in real clinical practice. It will emphasise the necessity of large datasets for their scalability and generalisability and focus on developing an accurate and reliable model for the early detection of diabetes that could be applied in healthcare settings.

2.8 Methodology of Existing Studies:

Previous studies Aishwarya Mujumdar and V Vaidehi (2019) and KM Jyoti Rani (2020) used traditional ML algorithms to detect diabetes by identifying the risk factors, but they have limited their study to a smaller dataset and evaluated a limited number of performance metrics. Extension to this Mitushi Soni, Dr. Sunita Varma (2020) used advanced ML, which was a great move as they got more accuracy compared to traditional ML, but even it is limited to small datasets. All these previous studies paved the way to this research by incorporating the limitations of the existing studies. Not just using a single method, but the comparison between traditional ML and advanced ML like ensemble stacking has been done in this study. This dual way of approach will address the existing studies and the research gap between this research and the past studies.

Chapter 3. Methodology

3.1 Introduction:

The goal of this research is to apply the different machine learning models to the chosen dataset and predict which one would be the best to predict diabetes in its early stages. This research has experimented with different traditional machine learning algorithms like Random Forest Classifier (RF), Logistic Regression (LR), Decision Tree, Gaussian Naive Bayes, Gradient Boosting, K-Nearest Neighbour (KNN), and Ensemble Stacking algorithms to predict diabetes. Application of each model and their performance on the dataset is discussed below.

Dataset Description:

The dataset is taken from open-source Kaggle, which is named Diabetes dataset Prediction. It has 100,000 patients with 9 attributes.

S.**No**.	**Attributes**
1	Gender
2	Age
3	Hypertension
4	Heart disease
5	Smoking History
6	BMI
7	HbA1c Level
8	Blood Glucose Level
9	Diabetes

The gender attribute indicates the biological sex of the individual patient, which can be male or female in this dataset. Age of everyone is given in this second attribute. Hypertension tells you whether the person has high blood pressure or not; values can be 0 and 1, where 0 indicates no and 1 indicates yes. The individual is diagnosed with heart disease or not, as indicated by the heart disease attribute using values 1 and 0. The smoking history of the individual is recorded using this 5th attribute through the values Never, Former, Current, Not Current, and No Info. BMI indicates the body mass index of the individual, a value that is derived from the height and weight of the individual. The blood sugar levels over the past 2-3 months are indicated by the HbA1c Level attribute. The blood glucose level of the individual is indicated by the 8th attribute in the dataset. The target attribute is the diabetes attribute, which tells whether the individual has diabetes or not through the values 1 and 0.

3.2 Research Approach:

(a) Data Preprocessing:

Data preprocessing is an important technique used before applying the machine learning models to the dataset. Healthcare-related data mostly has missing values and other impurities, which can decrease the effectiveness of the model. To improve the quality of the dataset and effectiveness of the model, preprocessing needs to be done. This dataset undergoes several steps during this data processing.

(b) Loading the dataset: This preprocessing step involves reading the CSV file from your device and importing it to a Pandas data frame before any analysis.

(c) Handling missing values: Any missing values in the dataset are identified and addressed to prevent errors in analysis and modelling. If any missing values are found, they must be removed from the dataset.

(d) Handling duplicate values: Ensuring each record is unique is important for the integrity of the analysis. This step involves finding the duplicate records and dropping the duplicates if any.

(e) Visualising the Data: It gives a better way of visualising and understanding the distribution of variables-categorical and numerical with their anomalies and patterns. For this case, numeric and categorical columns are plotted for the dataset, with age, blood glucose level, and diabetes having a relationship with one another represented by a stacked area chart. These plots provide insight into the data attributes.

(f) Encoding the categorical columns and correlation: The categorical columns are encoded to numerical format using a label encoder, as it is essential to make suitable while applying the machine learning algorithms. A correlation heat map is displayed for the data frame, which identifies the features and talks about the relationship among the attributes.

(g) Class Imbalance and Applying SMOTE: Balancing the dataset is also important before applying the machine learning models because it makes sure that the model does not become biased towards the majority class. In such cases, techniques like SMOTE are used to oversample the minority class. In this dataset, the technique is applied to the target attribute diabetes.

(h) Splitting the Data: The data is split into training and test datasets, which evaluate the model, which is trained and evaluated perfectly, increasing the model performance. In this research, the dataset is divided into 80% of the training dataset and 20% of the test dataset.

(i) Removing the Outliners: This process involves identifying the outliners and removing them using the Interquartile Range (IQR), which effects the model performance and quality of the data. It ensures the training data is clean and free from extreme values.

3.3 Applying Machine Learning Models:

Machine learning uses quite a few tools and methods to transform raw data into meaningful insights. Major algorithm types include the following: Supervised learning deals with regression and classification tasks in creating predictive models from labelled data where outcomes are known. Unsupervised learning deals with unlabelled data to create a descriptive model of pattern recognition, generally focused on clustering. Semi-supervised learning in combination with supervised and unsupervised learning can exploit jointly labelled and unlabelled data. Finally, reinforcement learning learns how to behave optimally through interaction with the environment to achieve rewards or minimise risks.

In this paper, seven machine learning algorithms will be evaluated, including Random Forest Classifier, Logistic Regression, Decision Tree, Gaussian Naive Bayes, Gradient Boosting, K-Nearest Neighbour, and Ensemble Stacking. This will be done to compare traditional algorithms with the advanced Ensemble Stacking technique in a bid to prove that Ensemble Stacking outperformed traditional models on all performance metrics by which it was assessed and was able to predict diabetes with improved speed.

Supervised Classification Machine Learning:

A supervised classification machine learning model is used to predict the outcomes based on the input data. The model is trained on a labelled dataset, and the main goal is to accurately predict the output for new, unseen data. The dataset used in this research has a target column named 'diabetes' which indicates a classification problem (0 or 1). To work on such tasks, supervised classification models are well suited. The machine learning models used in this research come under this supervised classification of machine learning models. Each model applied to the dataset is discussed below.

(1) Random Forest Classifier:

Random forest classifier (RF) is a supervised classification algorithm developed by Leo Bremen that makes predictions using multiple decision trees. It evaluates the feature importance, handles missing data, and reduces the variance. Random forest can handle both classification and regression problems. In this study, RF is used to handle both numerical and categorical features. As the dataset is large, an RF classifier can reduce the overfitting compared to a single decision tree. To identify which features are most influential in predicting diabetes is provided by the RF classifier. It also balances bias and variance, which leads to better generalisation on unseen data.

The article by Breiman, L. (2001) emphasises the versatility of RF classifiers in handling different datatypes of the input features effectively. Reduction of overfitting and improving the overall accuracy by using RF is demonstrated by the study Liaw, A., & Wiener, M. (2002). One of the main strengths of RF is to evaluate the importance of features, as provided by Hastie, T., Tibshirani, R., & Friedman, J. (2009). RF is used in this study to acquire high accuracy and reliable performance.

While applying the random forest classifier, the dataset is randomly sampled to create multiple subsets. A decision tree is trained on each subset, but only a random subset of the feature is considered for splitting at each node. Every tree casts a vote for categorisation, and the class with the most votes becomes the final prediction. While applying the random forest classifier, the dataset is randomly sampled to create multiple subsets.

(2) Logistic Regression:

Logistic regression (LR) is a supervised classification model that is used for binary classification tasks. To predict if the given input belongs to certain classes, this statistical method is used. It generates the output with the probability score between 0 and 1, which is useful to make a binary decision. Despite being used for binary classification, the logit(log-odds) transformation of the probability of a binary outcome can be predicted by LR. It is less prone to overfitting and computationally efficient.

This research aim is to predict the absence or presence of diabetes, which is a binary outcome of 0 or 1. According to the study Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013), LR is designed for such binary classification tasks. The insights into the probability of having diabetes and the predictors are given by coefficients in logistic regression (Kleinbaum, D. G., & Klein, M. (2010)). In the study by Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002), compared to complex models like random forest or neural networks, LR is less intensive, which makes it more suitable for large datasets. Menard, S. (2002). Research on applied logistic regression analysis details the LR performance, showing the relationship between features and the log-odds of the outcome.

Applying a logistic regression classifier to the dataset predicts the probability of the target variable that belongs to a specific class. In this model, a logistic (sigmoid) function is used to transform the linear combinations of the input features into a probability. It worked very well when the relationship between the independent variables (example: age, BMI, blood glucose level) and target variable (diabetes) is approximately linear.

(3) Decision Tree:

Decision Tree (DT) is a non-linear machine learning algorithm that is used for both classification and regression tasks. This model involves splitting the data into subsets based on values of input features, which results in a tree-like structure of decisions. A 'test' on each attribute is represented by an internal node, and each branch represents the test outcome. A class label (classification) is represented by each leaf.

A study by Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984) explains how DT can handle the dataset that has mixed data types. To determine which factors are most important in predicting diabetes for this research, DT is used because the research by Quinlan, J. R. (1986) discusses how DT has an ability to provide clear and interpretable results. The nonlinear relationship between diabetes and predictors can be easily caught by DT, and it is given by the study Loh, W.-Y. (2011).

When a decision tree is applied, the dataset is split based on the most significant features, such as BMI and blood glucose level, which creates a tree structure. At each node, the best feature to split is chosen based on the criteria entropy. This model involves a continuous process of creating branches until a stopping criterion is met. When the process meets the maximum samples per leaf, then it stops, and that would be the final prediction for the dataset.

(4) Gradient Boost Classifier (GB):

Gradient boosting is among the strongest machine learning algorithms, used in solving problems of classification and regression. This works by training models sequentially to combine the power of many weak learners for a strong model to predict. Therefore, this study is aimed at seeking a model that can help predict diabetes at an early stage. Friedman, J. H. (2001) evaluated the accuracy of gradient boosting in predictive tasks, and Chen, T., & Guestrin, C. (2016) have explored the flexibility it had in modelling the complicated relations and interactions in data.

Gradient boosting classifier is used to reduce the risk of overfitting and make this model very robust even against noisy datasets by careful tuning. Natekin, A., & Knoll, A. (2013) shed some light on the approaches where this classifier is used to avoid overfitting. Considering how feature contributions are important in predicting diabetes, the gradient boosting classifier helps the researcher to understand the dataset's features just the same way Friedman, J. H. (2002) explains in his book on computational statistics and data analysis.

Gradient boosting initialises with some sort of prediction, usually the mean of a target variable, and then iteratively fits models to residual errors. This algorithm iteratively updates a model with the residual errors of the previous model so that it minimises those residual errors. It sums up all the weighted predictions from the models to give a final output with minimal errors. This makes gradient boosting very effective in the early detection of diabetes.

(5) Gaussian Naïve Bayes Classifier (GNB):

Gaussian Naïve Bayes classifier is a supervised machine learning model based on the Bayes theorem and gaussian distribution. It is a probabilistic classification algorithm that applies the Bayes theorem with the assumption of feature independence and a gaussian (normal) distribution of continuous features. The probability of a class is calculated by the Bayes theorem, and naive Bayes assumes that the features are conditionally independent given the class label. The presence or absence of one feature does not affect the presence or absence of another feature. This model allows the values of each feature to follow a gaussian (normal) distribution.

In this work, the Gaussian Naïve Bayes classifier is used for its simplicity and computational efficiency, which eventually made it very appropriate for large datasets and real-time applications. According to Zhang, H. (2004), GNB is simple and efficient. Since the diabetes dataset in this research has a high dimension, GNB is suitable for such data, as discussed by Rish, I. (2001). Besides, interpretability is important for insights and making informed decisions. In the 2012 study about the practical applications of GNB, Murphy, K. P. (2012) placed a huge amount of attention on ease of understanding and ease in communicating results.

Application of the Gaussian Naïve Bayes classifier to the dataset involves calculating the probability for each class being diabetes or non-diabetes. The posterior probability for each class is calculated by multiplying the prior probability and likelihood of the features. The predictions for each class are noted for the dataset, and the class that has the highest probability is chosen as the final prediction for the data.

(6) K-Nearest Neighbour Classifier (KNN):

The K-Nearest Neighbour classifier is a basic approach but a very effective one in classification and regression. A prime concept driving it is that similar data points should fall under the same class. KNN is a lazy learning algorithm as it doesn't build an explicit model at training time but memorises the dataset at prediction time. It executes computations by calculating distances from new data points.

The KNN model awards class labels based on the majority vote of K-nearest neighbours. One of the reasons for its importance is its simplicity, making sure it's easy to understand and implement, and transparency in research. It was found in the foundational work of Cover, T. M., & Hart, P. E. (1967), which brought about KNN and described the algorithm as simple and effective. Fix, E., & Hodges, J. L. (1951) discussed pros in handling complex and partially unknown datasets like the diabetes dataset used in this research.

This adaptivity of KNN to new data is very good since the prevalence of diabetes may vary with time, a point that was raised by Aha, D. W., Kibler, D., & Albert, M. K. (1991) in their paper on instance-based learning in 1991. The KNN model would calculate the Euclidean distance between features and make a prediction based on the majority class of the nearest neighbours to know whether a patient has diabetes.

The above six traditional machine learning algorithms are applied to the dataset used in this research. All these algorithms were used in many of the studies that are referred to in this research to predict the diabetes as early as possible. Each algorithm works differently with the dataset to predict whether the patient has diabetes or not using unique mechanisms. The objective of this research is to use advanced machine learning algorithms like ensemble stacking to predict diabetes at its early stages and to measure the performance of the model using six metrics that are used in the above algorithms. All these models are trained, evaluated, and compared for the better model performance that results in accurate diabetes prediction.

(7) Ensemble Methods:

The Ensemble methods are performed by integrating a combination of individual base models into one strong predictive model. It includes bagging, boosting, bagged boosting, voting, blending, and stacking techniques. Bagging algorithm, which commences by generating bootstrap samples and several base models are trained. In boosting, the models are trained sequentially. Each newly built model is intended to correct the errors that represent all the previous ones. Voting, in which multiple models are independently trained and their predictions combined under the majority rule. Blending, in which the predictions are made using hold-out validation for training a meta-model, which is contrary to cross-validation. Bagged boosting is a boosted ensemble technique in which the model is trained over different samples of the training data. Stacking is a meta-model, an advanced example of the technique for combining predictions from the base models. Although stacking is much more complicated because the metamodel is trained on the outputs of base models, it quite often does substantially better than the individual base models.

Applying ensemble stacking to the dataset for predicting diabetes is a strategic decision as it leverages the strengths of multiple machine learning models that enhance the accuracy, robustness, and generalizability. 'Stacked generalisation' research by Wolpert, D. H. (1992) demonstrates improving predictive performance by combining multiple models. Ensemble methods like stacking and bagging helps to reduce the overfitting. 'Bagging Predictors' by Breiman, L. (1996) explains how the stacking and bagging methods are useful to reduce the overfitting by combining multiple models. As the ensemble method involves leveraging multiple models, each model has different weaknesses. For example, a logistic regression classifier may underfit and the decision tree may overfit. In such cases, ensemble stacking balances these weaknesses and produces more reliable predictions. A study by Rokach, L. (2010) discusses how ensemble methods, including stacking, help in balancing the weaknesses of models and result in providing robustness and stability in model predictions.

(8) Ensemble Stacking with GridSearchCV:

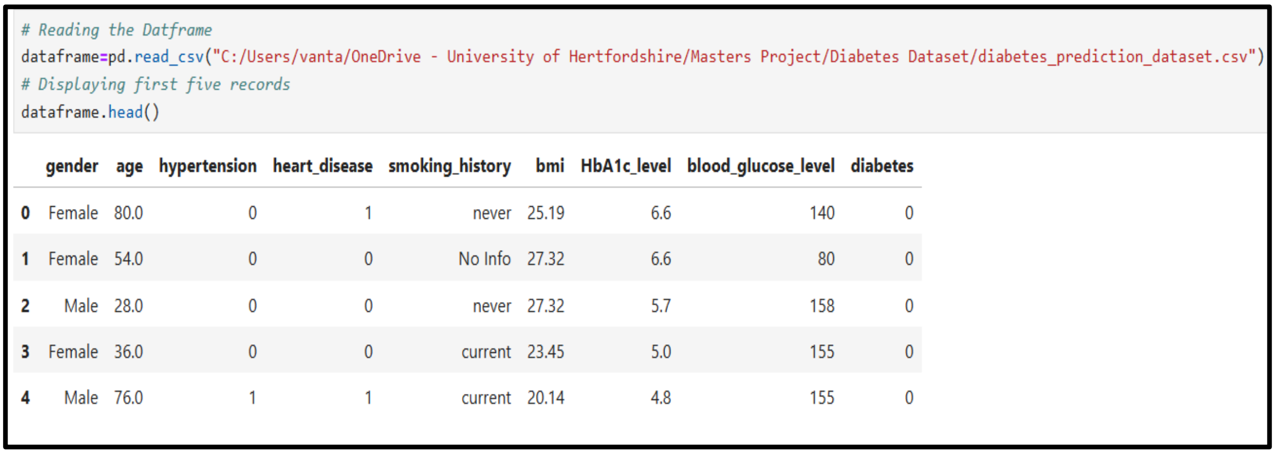
Applying ensemble stacking to the chosen dataset involves training all six base models on the dataset, which include logistic regression, random forest, decision tree, KNN, Gaussian Naive Bayes, and gradient boosting classifiers. The predictions are generated for each model when trained on the training dataset. The predictions from the base models are recorded and are used as input features for the meta-model. The Random Forest classifier is taken as the meta-model in this research, and any model can be applied as the meta-model depending on the problem. The final predictions are made by the meta-model after combining the predictions of the base models. The predictions made by this ensemble stacking have all the performance metrics like accuracy, precision, sensitivity, specificity, F1-score, and ROC-AUC score higher than all the base models, which made this ensemble stacking model the best model to predict diabetes as early as possible. The performance of the ensemble stacking model can be increased more by tuning the hyperparameters. GridSearchCV is used for tuning hyperparameters, which gives the best ensemble stacking model by increasing the accuracy of the model. It is a powerful model that combines the strengths of multiple models and optimises their parameters. Ensemble Stacking with GridSearchCV turns out to be the best performer in predicting the diabetes at early stages in this research.

Chapter 4. Results

This chapter presents the results of the data analysis and machine learning models. It focusses on describing the outcomes of various preprocessing steps, visualisations, transformations, and the performance of the applied machine learning algorithms.

4.1 Data Preprocessing Results:

Data Loading and Initial Inspection: The data is loaded from a CSV file and the first five records of the dataset are displayed.

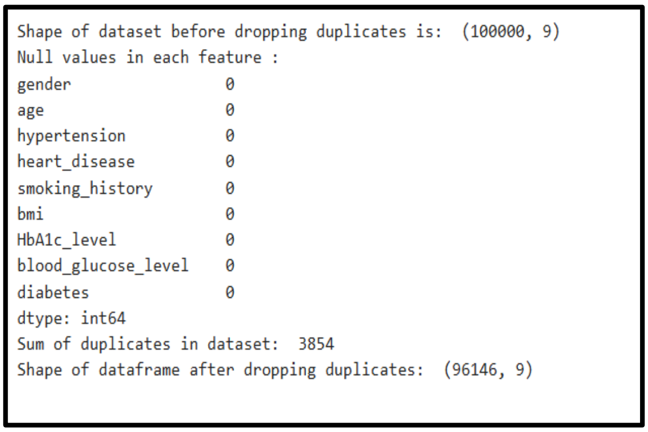


```
# Reading the Datframe
dataframe=pd.read_csv("C:/Users/vanta/OneDrive - University of Hertfordshire/Masters Project/Diabetes Dataset/diabetes_prediction_dataset.csv")
# Displaying first five records
dataframe.head()
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0

Figure 4.1.1: Exploring the Diabetes Dataset

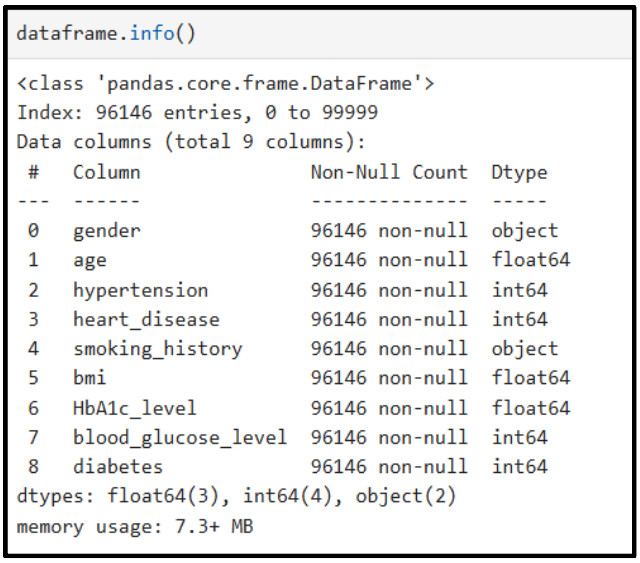
Checking for missing and duplicate values: The dataset is checked for null and duplicate values and dropped the duplicates from the dataset. The shape of dataset is checked before and after dropping the duplicates.



```
Shape of dataset before dropping duplicates is:  (100000, 9)
Null values in each feature :
gender                 0
age                    0
hypertension           0
heart_disease          0
smoking_history        0
bmi                    0
HbA1c_level            0
blood_glucose_level    0
diabetes               0
dtype: int64
Sum of duplicates in dataset:  3854
Shape of dataframe after dropping duplicates:  (96146, 9)
```

Figure 4.1.2: Before and After Data Cleaning

Datatypes and Summary: It is the summary of the dataset after pre-processing.

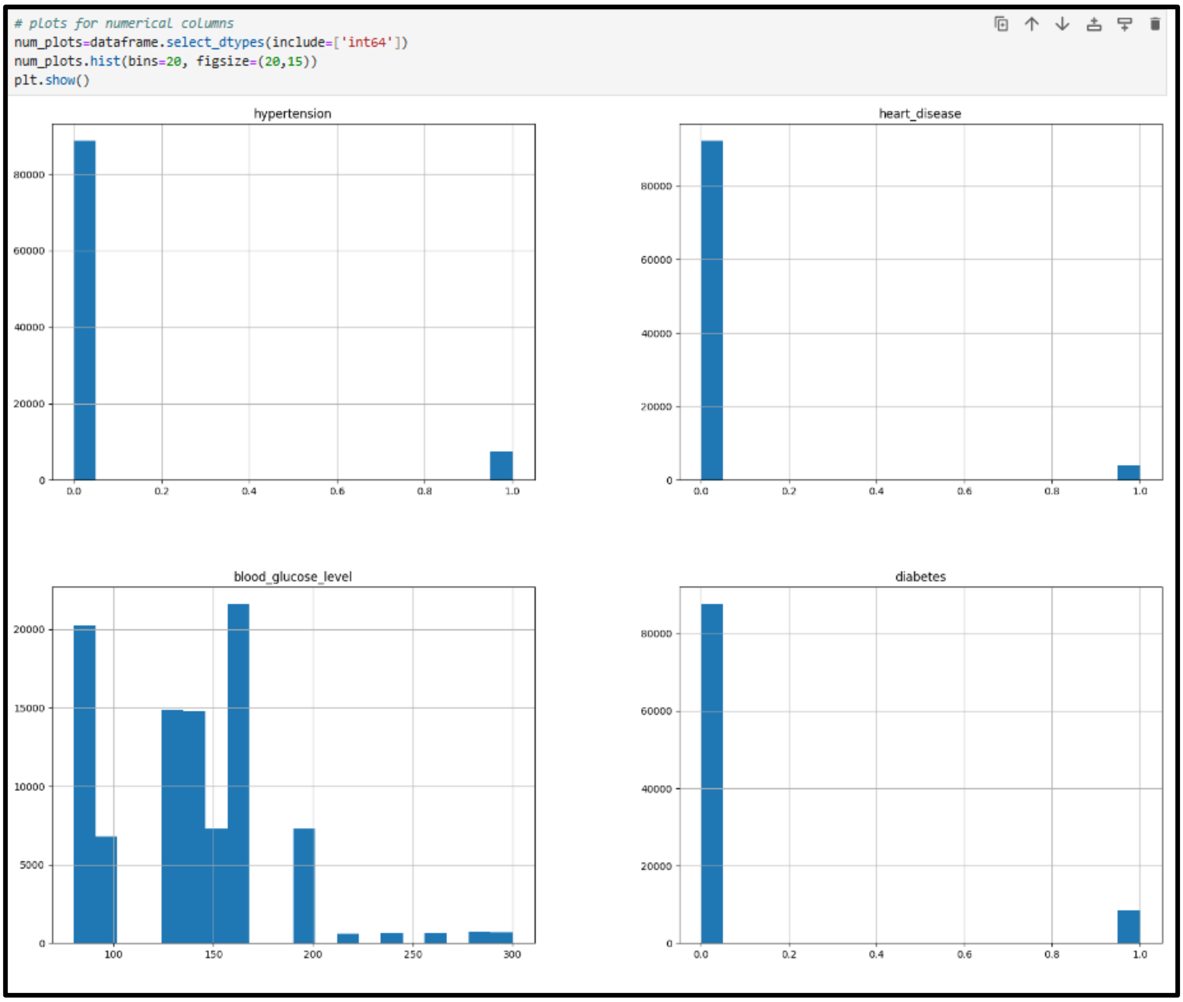


```
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 96146 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   gender               96146 non-null  object
 1   age                  96146 non-null  float64
 2   hypertension         96146 non-null  int64
 3   heart_disease        96146 non-null  int64
 4   smoking_history      96146 non-null  object
 5   bmi                  96146 non-null  float64
 6   HbA1c_level          96146 non-null  float64
 7   blood_glucose_level  96146 non-null  int64
 8   diabetes             96146 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 7.3+ MB
```

Figure 4.1.3: Dataset Features and datatypes

4.2 Data Visualizations:



```
# plots for numerical columns
num_plots=dataframe.select_dtypes(include=['int64'])
num_plots.hist(bins=20, figsize=(20,15))
plt.show()
```

Figure 4.2.1: Histogram visualization for Numerical columns

Categorical Features:

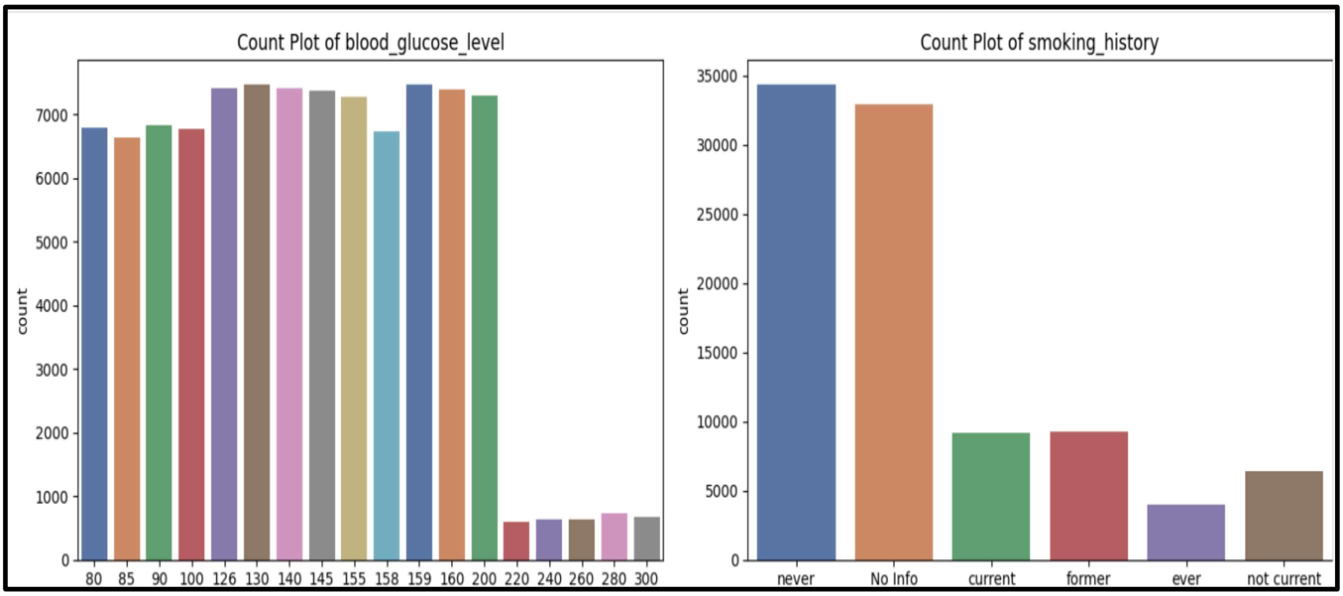


Figure 4.2.2: Count plot for categorical columns

Stacked area chart:

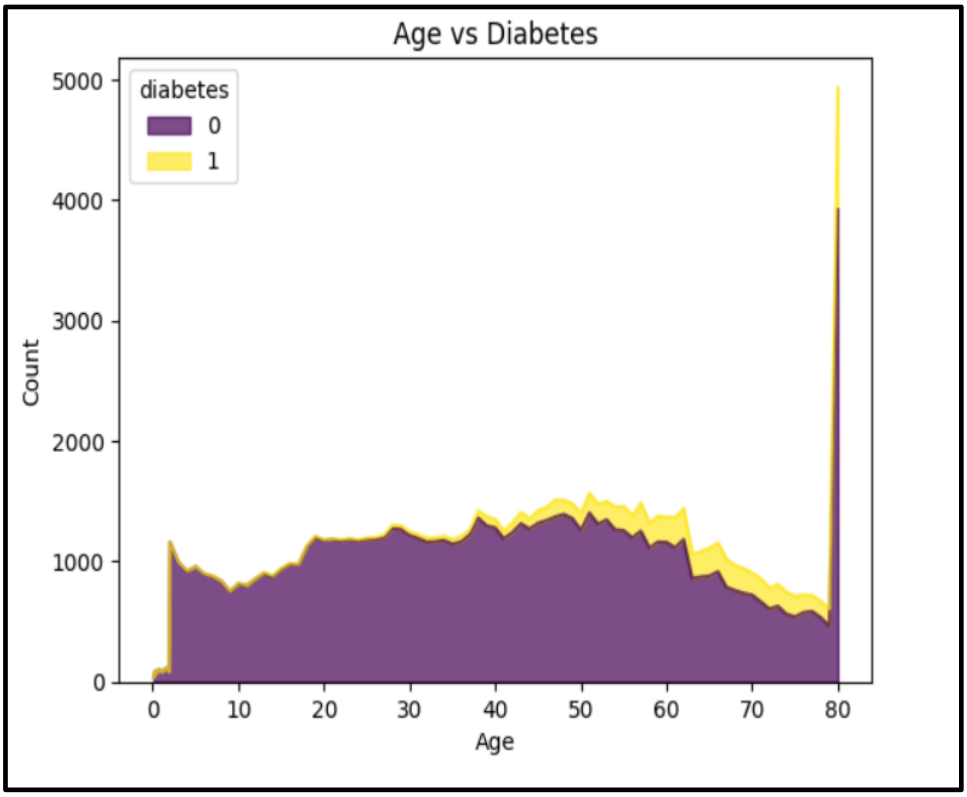
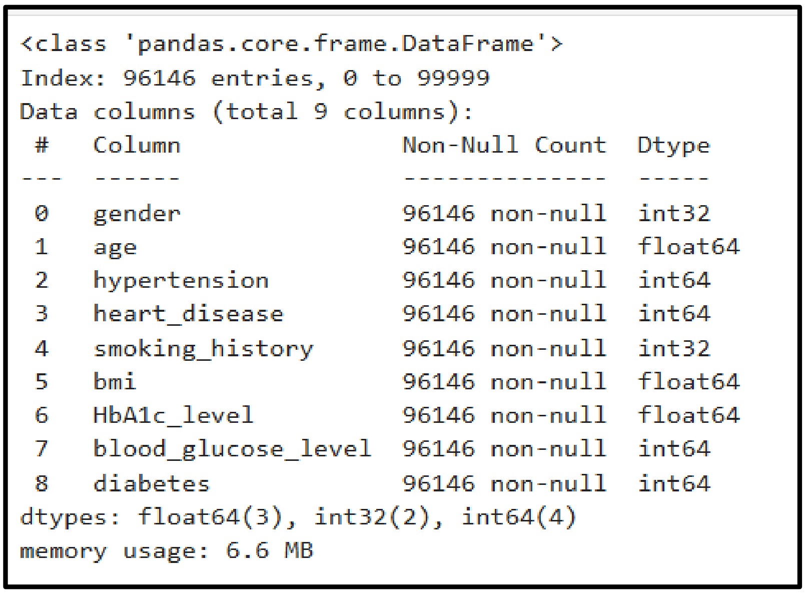


Figure 4.2.3: Stacked area chart between Age and Diabetes

4.3 Data Transformations:

Encoding categorical variables: The categorical variables are encoded using the label encoding technique, and the datatypes of all the features are displayed.



```
<class 'pandas.core.frame.DataFrame'>
Index: 96146 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   gender               96146 non-null  int32
 1   age                  96146 non-null  float64
 2   hypertension         96146 non-null  int64
 3   heart_disease        96146 non-null  int64
 4   smoking_history      96146 non-null  int32
 5   bmi                  96146 non-null  float64
 6   HbA1c_level          96146 non-null  float64
 7   blood_glucose_level  96146 non-null  int64
 8   diabetes             96146 non-null  int64
dtypes: float64(3), int32(2), int64(4)
memory usage: 6.6 MB
```

Figure 4.3.1: Datatypes of Columns after Encoding

Correlation matrix: A heatmap is generated to visualize the correlation between the features.

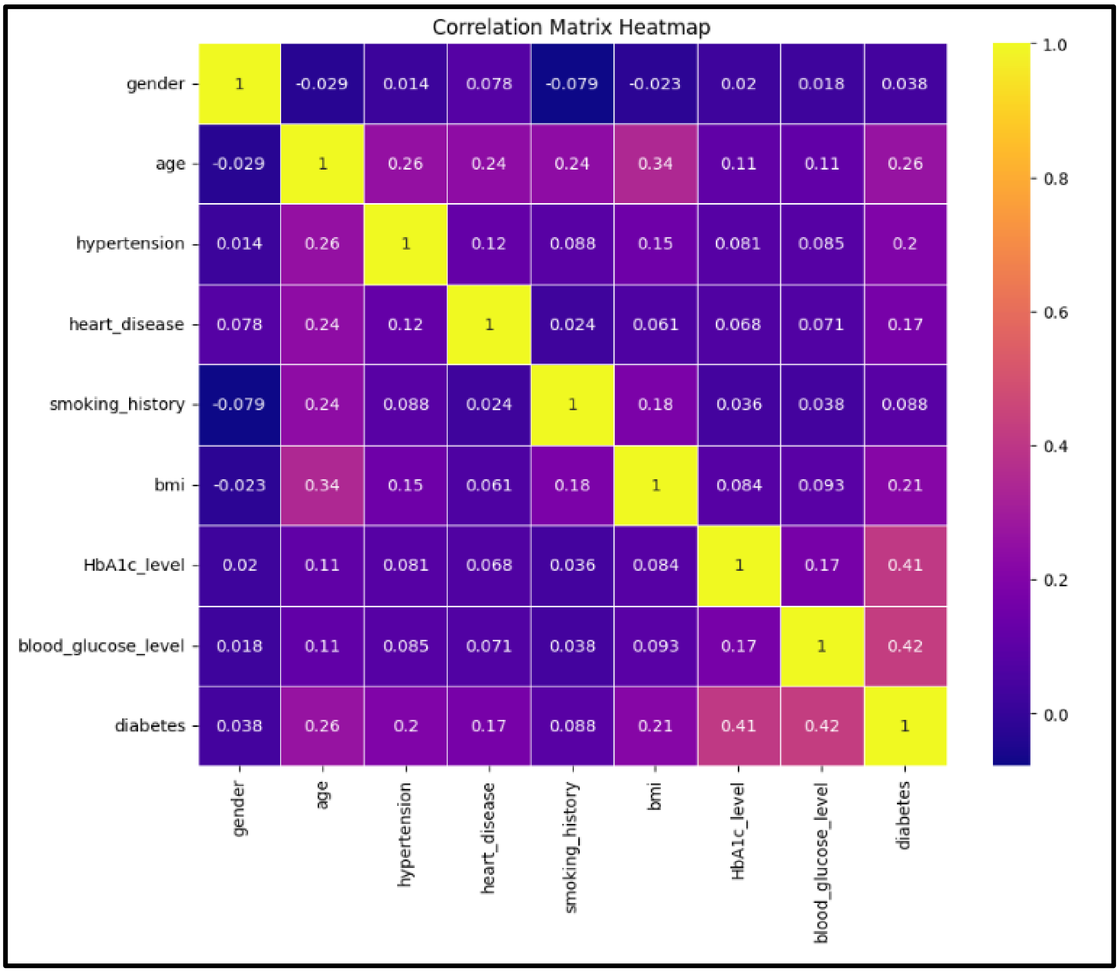
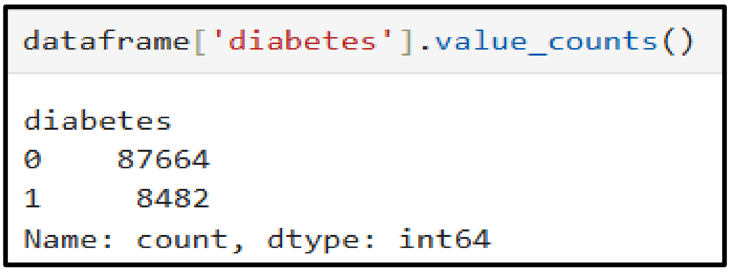


Figure 4.3.2: Correlation Heatmap

4.4 Addressing class Imbalance:

Checking for class imbalance: Class counts for 0 and 1 in diabetes are displayed here.

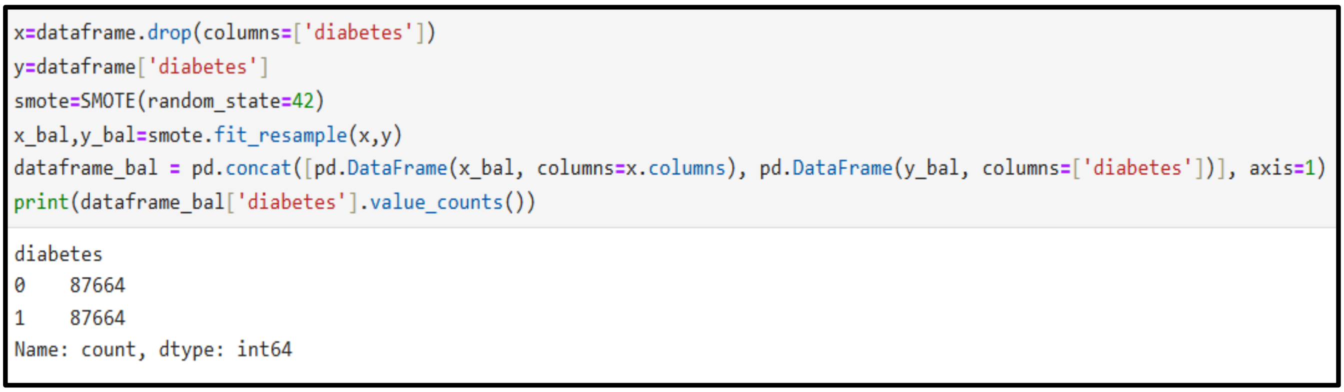


```
dataframe['diabetes'].value_counts()
```

```
diabetes
0    87664
1     8482
Name: count, dtype: int64
```

Figure 4.4.1: Class Counts for diabetes Column

Apply SMOTE:_SMOTE technique is applied to handle the class imbalance and the counts are checked after applying this technique.



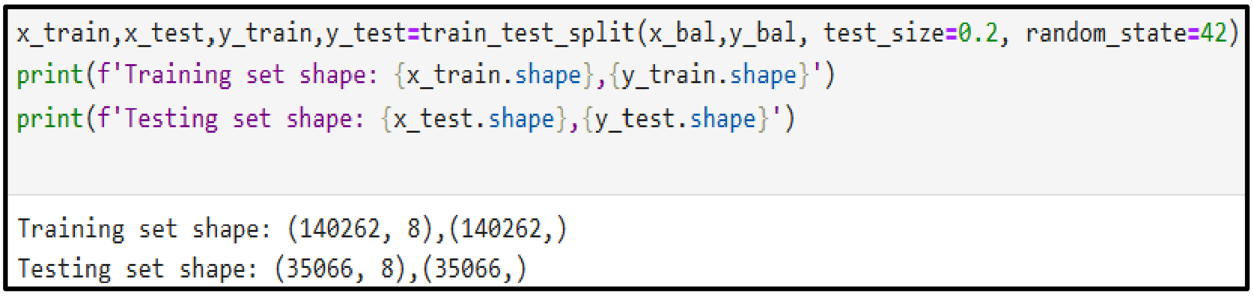
```
x=dataframe.drop(columns=['diabetes'])
y=dataframe['diabetes']
smote=SMOTE(random_state=42)
x_bal,y_bal=smote.fit_resample(x,y)
dataframe_bal = pd.concat([pd.DataFrame(x_bal, columns=x.columns), pd.DataFrame(y_bal, columns=['diabetes'])], axis=1)
print(dataframe_bal['diabetes'].value_counts())
```

```
diabetes
0    87664
1    87664
Name: count, dtype: int64
```

Figure 4.4.2: Class Counts after applying SMOTE

4.5 Splitting data and removing outliners:

Splitting the data:

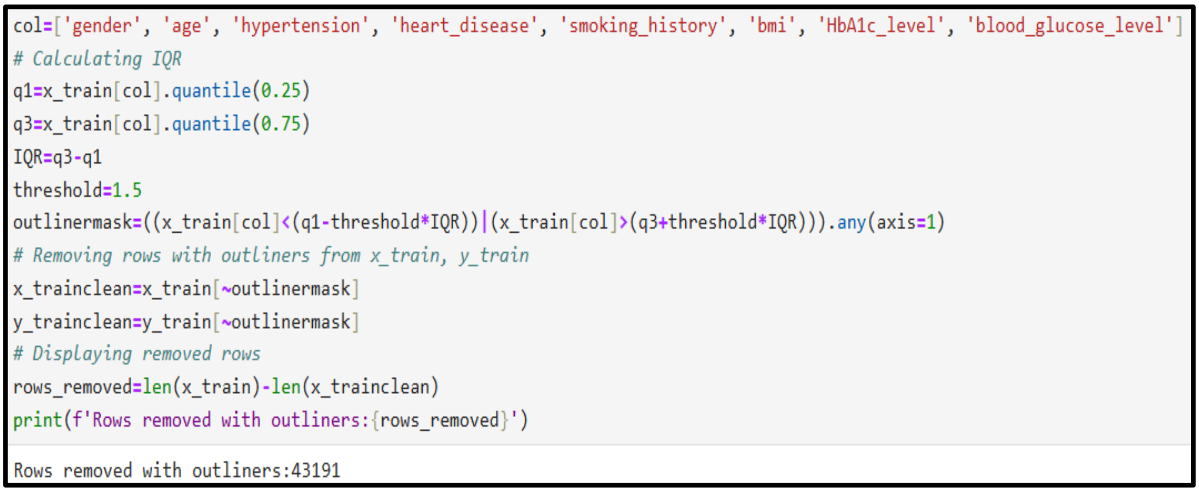


```
x_train,x_test,y_train,y_test=train_test_split(x_bal,y_bal, test_size=0.2, random_state=42)
print(f'Training set shape: {x_train.shape},{y_train.shape}')
print(f'Testing set shape: {x_test.shape},{y_test.shape}')
```

```
Training set shape: (140262, 8),(140262,)
Testing set shape: (35066, 8),(35066,)
```

Figure 4.5.1: Training and test data sets shape

Removing outliners: The Interquartile range IQR) is calculated and used to remove outliners from the training data.



```
col=['gender', 'age', 'hypertension', 'heart_disease', 'smoking_history', 'bmi', 'HbA1c_level', 'blood_glucose_level']
# Calculating IQR
q1=x_train[col].quantile(0.25)
q3=x_train[col].quantile(0.75)
IQR=q3-q1
threshold=1.5
outlinermask=((x_train[col]<(q1-threshold*IQR))|(x_train[col]>(q3+threshold*IQR))).any(axis=1)
# Removing rows with outliners from x_train, y_train
x_trainclean=x_train[~outlinermask]
y_trainclean=y_train[~outlinermask]
# Displaying removed rows
rows_removed=len(x_train)-len(x_trainclean)
print(f'Rows removed with outliners:{rows_removed}')
```

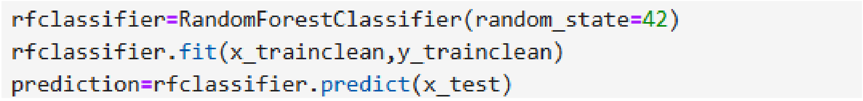
```
Rows removed with outliners:43191
```

Figure 4.5.2: Row count in training set after removing outliners

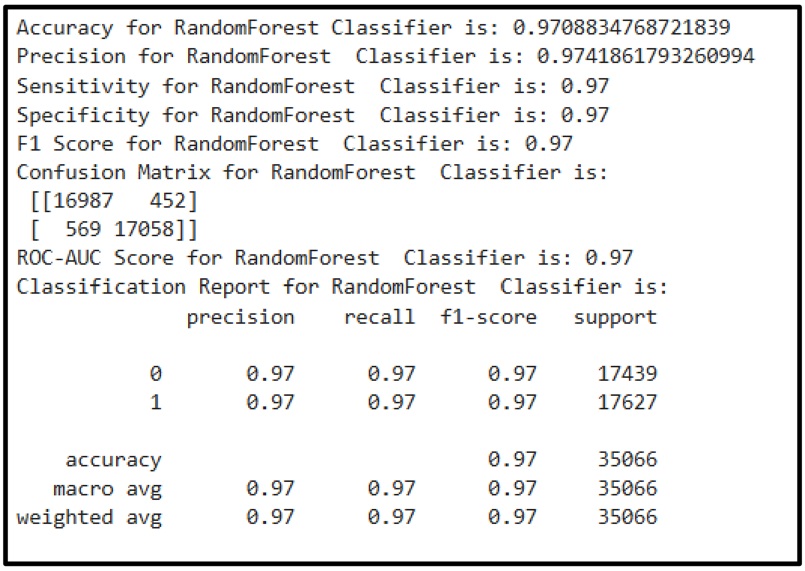
4.6 Applying Machine learning algorithms:

The Machine learning models are trained on training data and the test data is used for prediction. The Performance metrics like accuracy, precision, sensitivity, specificity, F1-score, confusion matrix, ROC-AUC score and classification report are displayed for all the models applied.

Random forest Classifier:



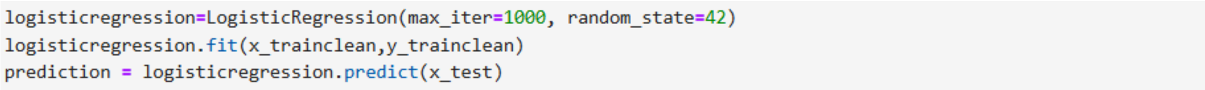
```
rfclassifier=RandomForestClassifier(random_state=42)
rfclassifier.fit(x_trainclean,y_trainclean)
prediction=rfclassifier.predict(x_test)
```



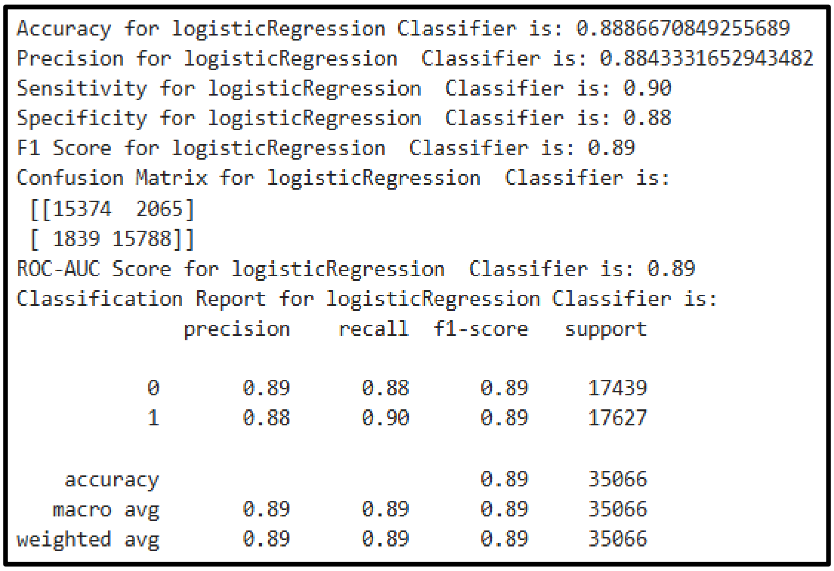
```
Accuracy for RandomForest Classifier is: 0.9708834768721839
Precision for RandomForest  Classifier is: 0.9741861793260994
Sensitivity for RandomForest  Classifier is: 0.97
Specificity for RandomForest  Classifier is: 0.97
F1 Score for RandomForest  Classifier is: 0.97
Confusion Matrix for RandomForest  Classifier is:
 [[16987   452]
 [  569 17058]]
ROC-AUC Score for RandomForest  Classifier is: 0.97
Classification Report for RandomForest  Classifier is:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     17439
           1       0.97      0.97      0.97     17627

    accuracy                           0.97     35066
   macro avg       0.97      0.97      0.97     35066
weighted avg       0.97      0.97      0.97     35066
```

Figure 4.6.1: Performance Metrics for Random Forest Classifier

Logistic Regression Classifier:



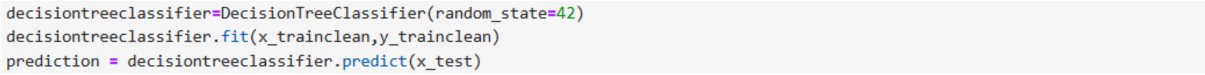
```
logisticregression=LogisticRegression(max_iter=1000, random_state=42)
logisticregression.fit(x_trainclean,y_trainclean)
prediction = logisticregression.predict(x_test)
```



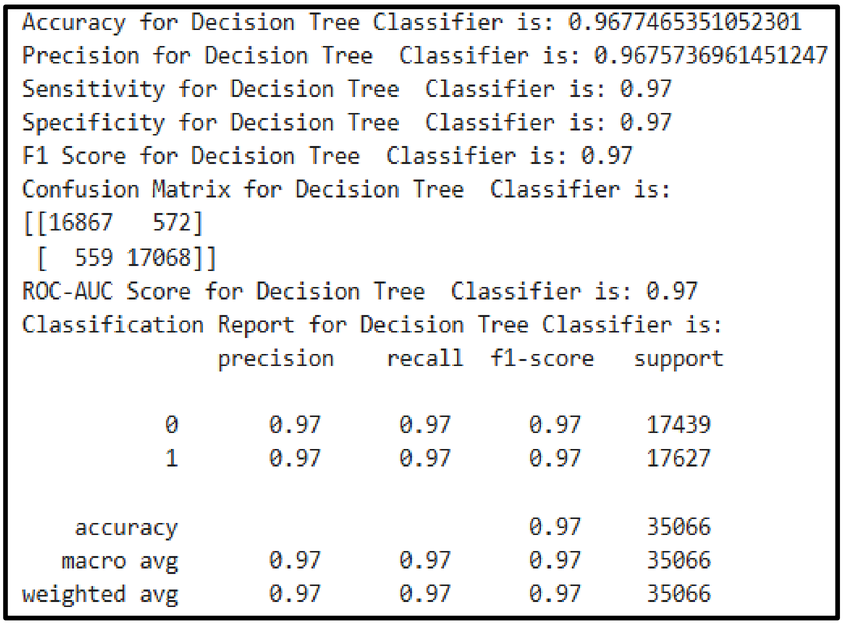
```
Accuracy for logisticRegression Classifier is: 0.8886670849255689
Precision for logisticRegression  Classifier is: 0.8843331652943482
Sensitivity for logisticRegression  Classifier is: 0.90
Specificity for logisticRegression  Classifier is: 0.88
F1 Score for logisticRegression  Classifier is: 0.89
Confusion Matrix for logisticRegression  Classifier is:
 [[15374  2065]
 [ 1839 15788]]
ROC-AUC Score for logisticRegression  Classifier is: 0.89
Classification Report for logisticRegression Classifier is:
              precision    recall  f1-score   support

           0       0.89      0.88      0.89     17439
           1       0.88      0.90      0.89     17627

    accuracy                           0.89     35066
   macro avg       0.89      0.89      0.89     35066
weighted avg       0.89      0.89      0.89     35066
```

Figure 4.6.2: Performance Metrics for Logistic Regression Classifier

Decision Tree Classifier:



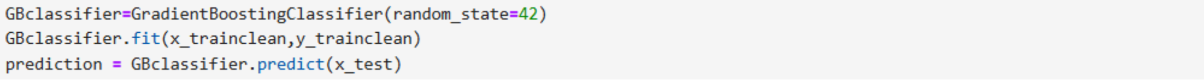
```
decisiontreeclassifier=DecisionTreeClassifier(random_state=42)
decisiontreeclassifier.fit(x_trainclean,y_trainclean)
prediction = decisiontreeclassifier.predict(x_test)
```



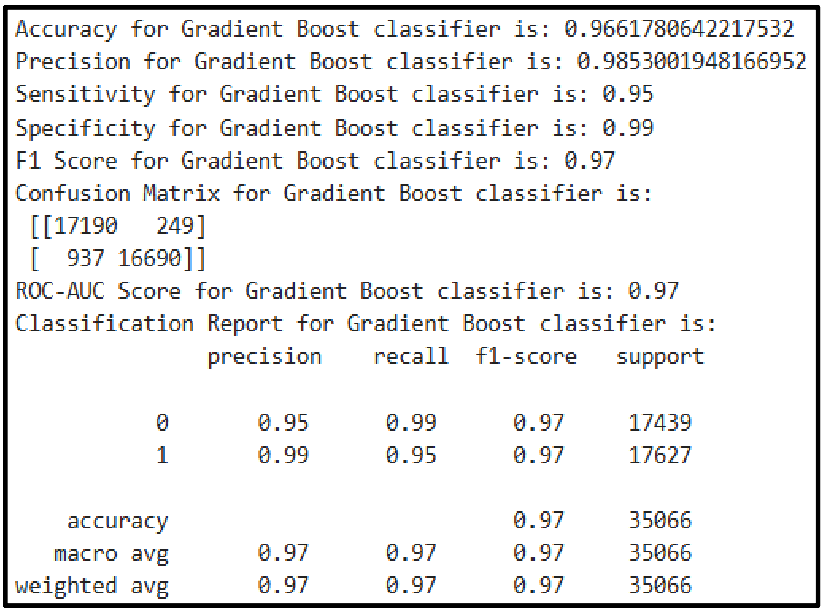
```
Accuracy for Decision Tree Classifier is: 0.9677465351052301
Precision for Decision Tree  Classifier is: 0.9675736961451247
Sensitivity for Decision Tree  Classifier is: 0.97
Specificity for Decision Tree  Classifier is: 0.97
F1 Score for Decision Tree  Classifier is: 0.97
Confusion Matrix for Decision Tree  Classifier is:
[[16867   572]
 [  559 17068]]
ROC-AUC Score for Decision Tree  Classifier is: 0.97
Classification Report for Decision Tree Classifier is:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     17439
           1       0.97      0.97      0.97     17627

    accuracy                           0.97     35066
   macro avg       0.97      0.97      0.97     35066
weighted avg       0.97      0.97      0.97     35066
```

Figure 4.6.3: Performance Metrics for Decision Tree Classifier

Gradient Boosting Classifier:



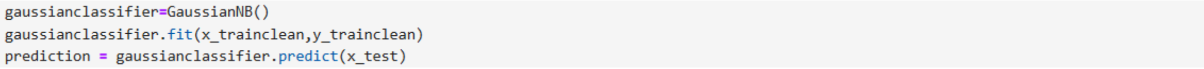
```
GBclassifier=GradientBoostingClassifier(random_state=42)
GBclassifier.fit(x_trainclean,y_trainclean)
prediction = GBclassifier.predict(x_test)
```



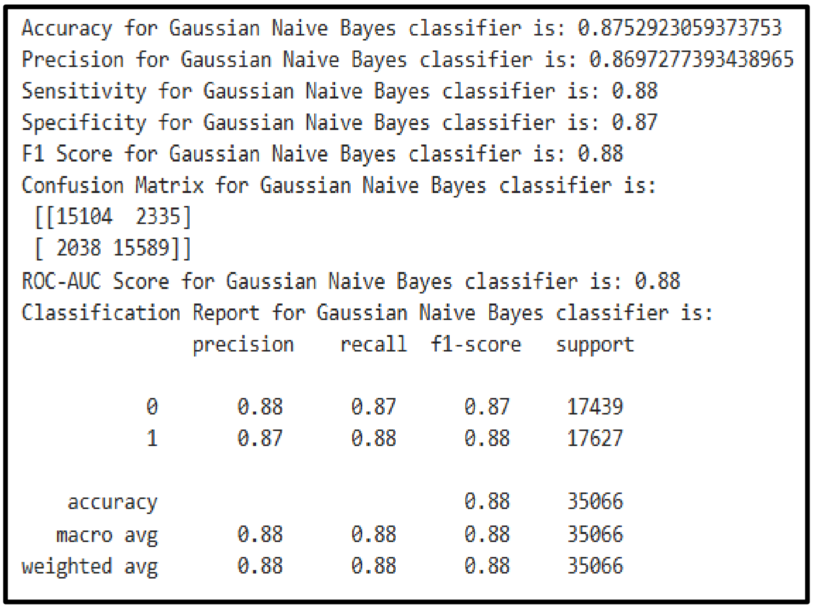
```
Accuracy for Gradient Boost classifier is: 0.9661780642217532
Precision for Gradient Boost classifier is: 0.9853001948166952
Sensitivity for Gradient Boost classifier is: 0.95
Specificity for Gradient Boost classifier is: 0.99
F1 Score for Gradient Boost classifier is: 0.97
Confusion Matrix for Gradient Boost classifier is:
 [[17190   249]
 [  937 16690]]
ROC-AUC Score for Gradient Boost classifier is: 0.97
Classification Report for Gradient Boost classifier is:
               precision    recall  f1-score   support

           0       0.95      0.99      0.97     17439
           1       0.99      0.95      0.97     17627

    accuracy                           0.97     35066
   macro avg       0.97      0.97      0.97     35066
weighted avg       0.97      0.97      0.97     35066
```

Figure 4.6.4: Performance Metrics for Gradient Boosting Classifier

Gaussian Naïve Bayes Classifier:



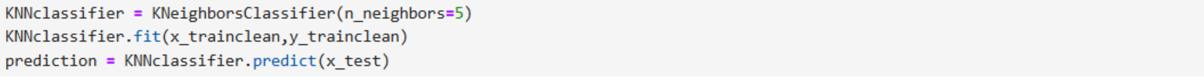
```
gaussianclassifier=GaussianNB()
gaussianclassifier.fit(x_trainclean,y_trainclean)
prediction = gaussianclassifier.predict(x_test)
```



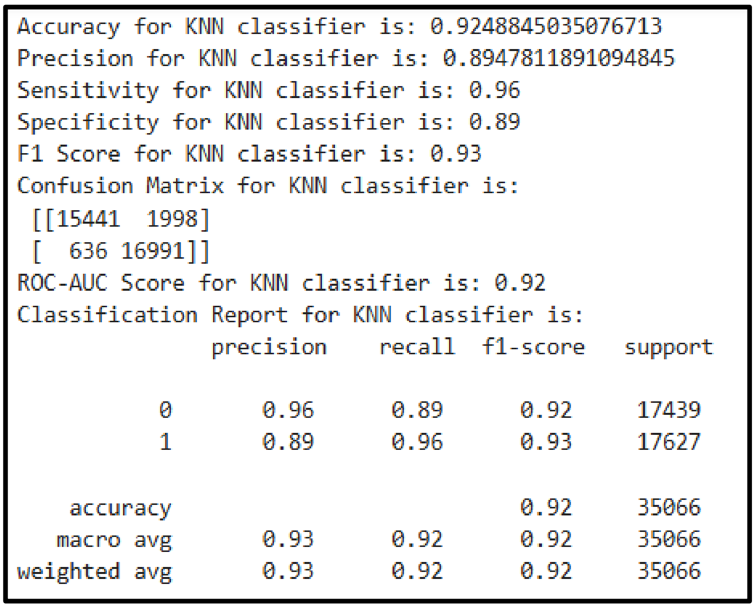
```
Accuracy for Gaussian Naive Bayes classifier is: 0.8752923059373753
Precision for Gaussian Naive Bayes classifier is: 0.8697277393438965
Sensitivity for Gaussian Naive Bayes classifier is: 0.88
Specificity for Gaussian Naive Bayes classifier is: 0.87
F1 Score for Gaussian Naive Bayes classifier is: 0.88
Confusion Matrix for Gaussian Naive Bayes classifier is:
 [[15104  2335]
 [ 2038 15589]]
ROC-AUC Score for Gaussian Naive Bayes classifier is: 0.88
Classification Report for Gaussian Naive Bayes classifier is:
               precision    recall  f1-score   support

           0       0.88      0.87      0.87     17439
           1       0.87      0.88      0.88     17627

    accuracy                           0.88     35066
   macro avg       0.88      0.88      0.88     35066
weighted avg       0.88      0.88      0.88     35066
```

Figure 4.6.5: Performance Metrics for Gaussian Naïve Bayes Classifier

K-Nearest Neighbour Classifier:



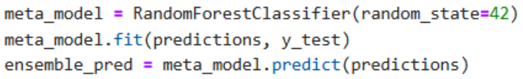
```
KNNclassifier = KNeighborsClassifier(n_neighbors=5)
KNNclassifier.fit(x_trainclean,y_trainclean)
prediction = KNNclassifier.predict(x_test)
```



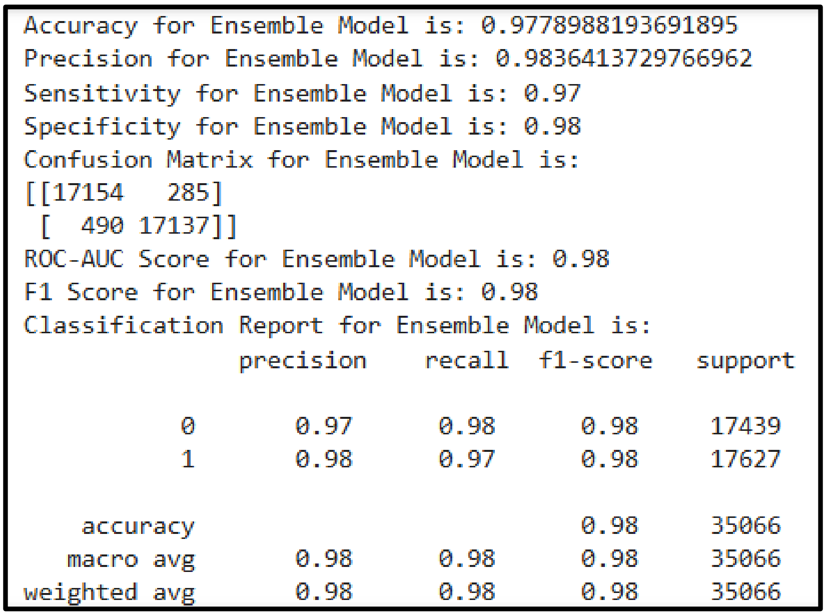
```
Accuracy for KNN classifier is: 0.9248845035076713
Precision for KNN classifier is: 0.8947811891094845
Sensitivity for KNN classifier is: 0.96
Specificity for KNN classifier is: 0.89
F1 Score for KNN classifier is: 0.93
Confusion Matrix for KNN classifier is:
 [[15441  1998]
 [  636 16991]]
ROC-AUC Score for KNN classifier is: 0.92
Classification Report for KNN classifier is:
              precision    recall  f1-score   support

           0       0.96      0.89      0.92     17439
           1       0.89      0.96      0.93     17627

    accuracy                           0.92     35066
   macro avg       0.93      0.92      0.92     35066
weighted avg       0.93      0.92      0.92     35066
```

Figure 4.6.6: Performance Metrics for K-Nearest Neighbour Classifier

Ensemble Stacking:



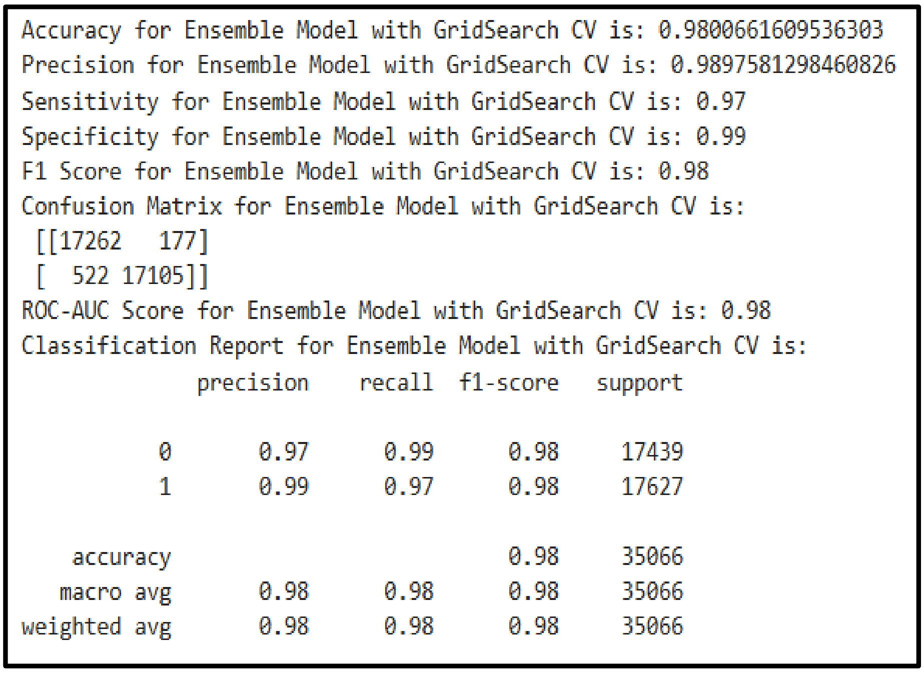
```
meta_model = RandomForestClassifier(random_state=42)
meta_model.fit(predictions, y_test)
ensemble_pred = meta_model.predict(predictions)
```



```
Accuracy for Ensemble Model is: 0.9778988193691895
Precision for Ensemble Model is: 0.9836413729766962
Sensitivity for Ensemble Model is: 0.97
Specificity for Ensemble Model is: 0.98
Confusion Matrix for Ensemble Model is:
[[17154   285]
 [  490 17137]]
ROC-AUC Score for Ensemble Model is: 0.98
F1 Score for Ensemble Model is: 0.98
Classification Report for Ensemble Model is:
              precision    recall  f1-score   support

           0       0.97      0.98      0.98     17439
           1       0.98      0.97      0.98     17627

    accuracy                           0.98     35066
   macro avg       0.98      0.98      0.98     35066
weighted avg       0.98      0.98      0.98     35066
```

Figure 4.6.7: Performance Metrics for Ensemble Stacking

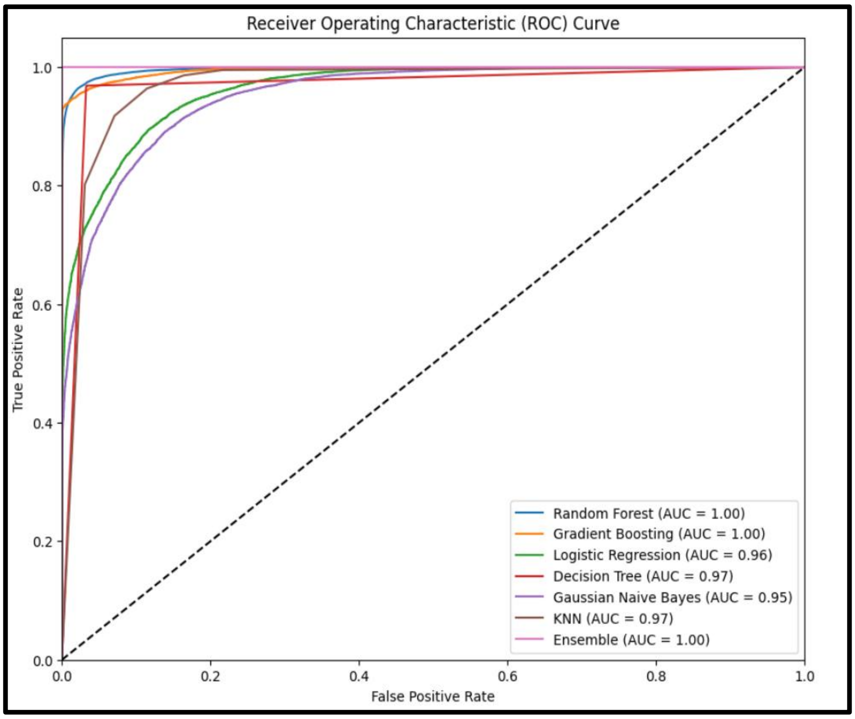
Ensemble Stacking with GridSearchCV: This involves tuning of hyperparameters using GridSearchCV and the performance metrics of Ensemble stacking are increased.



```
Accuracy for Ensemble Model with GridSearch CV is: 0.9800661609536303
Precision for Ensemble Model with GridSearch CV is: 0.9897581298460826
Sensitivity for Ensemble Model with GridSearch CV is: 0.97
Specificity for Ensemble Model with GridSearch CV is: 0.99
F1 Score for Ensemble Model with GridSearch CV is: 0.98
Confusion Matrix for Ensemble Model with GridSearch CV is:
 [[17262   177]
 [  522 17105]]
ROC-AUC Score for Ensemble Model with GridSearch CV is: 0.98
Classification Report for Ensemble Model with GridSearch CV is:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     17439
           1       0.99      0.97      0.98     17627

    accuracy                           0.98     35066
   macro avg       0.98      0.98      0.98     35066
weighted avg       0.98      0.98      0.98     35066
```

Figure 4.6.8: Performance Metrics for Ensemble Stacking with GridSearchCV

4.7 ROC-AUC Curve for all the models applied:



4.8 Performance Metrics Table:

Model	Accuracy	Precision	Sensitivity	Specificity	F1-Score	ROC-AUC Score
Random Forest	97.08	97.41	97	97	97	97
Logistic Regression	88.86	88.43	90	88	89	89
Decision Tree	96.77	96.75	97	97	97	97
Gradient Boost	96.61	98.53	95	99	97	97
Gaussian NB	87.52	86.97	88	87	88	88
K-Nearest Neighbour	92.48	89.47	96	89	93	92
Ensemble Stacking	97.78	98.36	97	98	98	98
Ensemble Stacking with GridSearch CV	98	**98.97**	**97**	**99**	**98**	**98**

Chapter 5. Result Analysis

The results of the data analysis and the machine learning models in relation to the research objectives and aims are discussed in this chapter. It also compares the findings with existing literature, which provides a critical understanding of the outcomes. This chapter includes the model performances and their implications in healthcare.

5.1 Interpretation of Results:

(a) Data Preprocessing:

Reading the dataset and displaying the first five records of the dataset is given by Fig 4.1.1. The features of the dataset in the result are gender, age, hypertension, heart disease, smoking history, BMI, HbA1c_level, blood glucose level, and diabetes. In fig. 4.1.2, the shape of the dataset is displayed as (100000, 9) where 100000 are the number of patient records and 9 are the columns of the dataset. There are no null values in any of the features. The number of duplicates in the dataset is found as 3854, so the duplicates are dropped, and now the shape of the date frame is (96146,9). Fig 4.1.3 shows there are no null values in the dataset and the datatypes of each feature in the dataset.

(b) Data Visualizations:

Fig 4.2.1: Histograms for the numerical columns: Hypertension, Heart Disease, Blood Glucose Level, and Diabetes. The records corresponding to the value of 0 in these plots to the absence of the disease and 1 to the presence of it. Most of the people are not having hypertension, heart disease, or diabetes, and only a small proportion have it. In the case of blood glucose levels, all values range between 50 and 300, making a wide variation in glucose levels a common feature among the patients in this dataset

Fig 4.2.2: Count plots of Blood Glucose Level and Smoking History: The count plot of Blood Glucose Level includes things like 80, 85, 90, etc. The y-axis is the number of the level of individuals. There is an even distribution from 80 to 160, having the count most of the time between 6000 and 8000. The count significantly drops for levels above 160 but only has a few at levels above 200. The count plot for Smoking History has the following categories on the x-axis: never, no info, current, former, ever, and not current. The largest groups are those who never smoked or have no smoking history information; all other categories were very small in number.

Humans are the main species affected by diabetes, and therefore have been extensively studied concerning the impact of age. Fig 4.2.3 Uses a Stacked Area chart to illustrate the relationship between age and diabetes. In this plot, 0 = no diabetes and 1 = has diabetes. What appears from this graph is that very few people who have diabetes are under 40, but there are a significant number of people between the ages of 40 and 80 who have diabetes.

(c) Data Transformations:

The categorical columns are encoded using a label encoder, and after encoding, the datatypes of each feature are given by Fig. 4.3.1. The result shows the data types of all the features, which are int or float. The data types of smoking history and gender are encoded to int before they used to be objects.

The correlation matrix heatmap is shown in Fig. 4.3.2, which displays the correlation coefficients between the features of the dataset. The correlation between two variables is represented by each cell, with colours indicating the strength and direction of the relationship. Yellow indicates positive correlation; purple indicates negative correlation; and black represents little or no correlation. A strong positive correlation (0.42) is seen between blood glucose level and diabetes; a moderate positive correlation (0.24) between smoking history and heart disease; age and hypertension have a weak positive correlation (0.26).

(d) Addressing class Imbalance:

Fig 4.4.1 shows the class counts for diabetes, where it shows 87664 for the 0 class and 8482 for the 1. The classes need to be balanced using the SMOTE technique. After applying SMOTE, fig. 4.4.2 shows the class counts after applying SMOTE, with both classes having 87664 counts.

(e) Splitting data and Removing Outliners:

The dataset is split into training and test data, showing the data sets shape in fig. 4.5.1, where the training set shape is (140262, 8), (140262,) and the test set shape is (35066, 8),(35066,).The training data is used for application of the machine learning models, and they need the data without outliners. Fig. 4.5.2 shows the number of rows removed from the training dataset as they have outliners. The interquartile range is calculated to remove the rows with outliners. The result shows that 43191 rows with outliners are removed.

5.2 Results in relation to Aim and Objectives:

The aim of the research is to develop and evaluate advanced machine learning models for early prediction of diabetes using a comprehensive dataset, aiming to provide predictive accuracy and facilitating early intervention strategies in healthcare. The dataset used in this research is bigger data, which has 100,000 records (patients) and 9 features (attributes). Six traditional machine learning (ML) algorithms, one advanced ML (Ensemble Stacking), and ensemble stacking with GridSearchCV are applied to the dataset, and the performance metrics for all the algorithms are noted. The ensemble stacking with GridSearchCV gave the best performance compared to other algorithms.

Objective 1: Performance comparison and evaluation of Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Gaussian Naive Bayes, KNN, Ensemble Stacking, and Ensemble Stacking with GridSearchCV in diabetes prediction by a set of comprehensive metrics: accuracy, precision, sensitivity, specificity, F1-score, ROC-AUC score, correlation matrix, and ROC-AUC curve.

In this research, the mentioned performance metrics in Objective 1 are calculated for the nine models for diabetes prediction. However, the ensemble stacking with the GridSearchCV model has shown improved performance metrics when compared with the other models. The performance metrics are explained below:

Accuracy: The proportion of correctly classified instances among the total instances. It is calculated as:

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

Here TP is True Positives, TN is True Negatives, FP is False Positives, and FN is False Negatives.

Precision: The proportion of true positive instances among the instances classified as positive. It is calculated as:

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

Sensitivity (Recall): The proportion of true positive instances among the actual positive instances is called sensitivity. It is calculated as:

$$\text{Sensitivity} = \frac{TP}{(TP+FN)}$$

Specificity: The proportion of true negative instances among the actual negative instances. It is calculated as:

$$\text{Specificity} = \frac{TN}{(TN+FP)}$$

F1-Score: The harmonic mean of precision and recall is called F1-Score. It is calculated as:

$$\text{F1-Score} = \frac{(Precision*Recall)}{(Precision+Recall)}$$

ROC-AUC Score: ROC is receiver operating characteristic curve and AUC is Area under the curve. ROC-AUC Score shows how well the classifier distinguishes between positive and negative classes.

Fig. 4.7: ROC-AUC curve of all models against the binary classification task. The x-axis is the false positive rate, and the y-axis represents the true positive rate. The higher and more to the left the curve is located, the better the performance of the model. AUC scores are in the bottom right corner, colouring each model's performance. The Random Forest, Gradient Boosting, and Ensemble models resulted in perfect classification with an AUC of 1.00. Logistic Regression, Decision Tree, KNN, and Gaussian Naive Bayes did almost as well as them; their AUC scores are very close to 1.00.

Table 4.8: Performance metrics of all models: accuracy, precision, sensitivity, specificity, F1-score, and ROC-AUC score. Obviously, ensemble stacking with GridSearchCV outperformed all other models in these metrics.

Objective 2: Implementing the ensemble stacking method with GridSearchCV, which combines the strengths of multiple algorithms, in search of higher predictive performance and robustness against traditional individual machine learning models.

Implementation of ensemble stacking methods with GridSearchCV and evaluating their performance metrics can be viewed in Fig. 4.6.8. The ensemble model is the one that combines the strengths of multiple algorithms. In this research paper, it combines the strengths of all the individual models used on the dataset. The result shows this model acquired an accuracy of 98.06%, which is the highest performance compared to the individual models like Random Forest (97.08%), Logistic Regression (80.86%), Decision Tree (96.77%), Gradient Boost (96.61%), Gaussian NB (87.52%), K-Nearest Neighbour (92.48%), and Ensemble Stacking (97.78%).

Not only the accuracy, but the ensemble stacking with GridSearchCV has the highest precision, sensitivity, specificity, F1-Score, and ROC-AUC score compared with the individual machine learning models and the advanced ML model ensemble stacking.

5.3 Analysis of Research Questions:

Research Question 1: Compared with individual machine learning algorithms, how does the predictive accuracy and robustness increase using ensemble stacking with GridSearchCV in diabetes prediction?

Ensemble stacking with GridSearchCV performed best among all individual models in every metric: accuracy (98%), precision (98.97%), and the F1-Score (98%). This improved performance is a consequence of the fact that ensemble stacking amalgamates a variety of advantages of different models, hence decreasing bias and variance. GridSearchCV has further optimised the model by tuning hyperparameters that improved the robustness and predictive accuracy of the model.

Research Question 2: How does the use of a broader range of performance metrics contribute to a more nuanced understanding of model performance in diabetes prediction?

This study considers a holistic view of metrics that includes accuracy, precision, sensitivity, specificity, F1-Score, and ROC-AUC to interpret model performance in detail.

Accuracy does not indicate type of errors but shows overall correctness. Precision and sensitivity measure how good the model is at predicting actual positives that is, focussing on positive prediction performance. Specificity tells how well the model can predict true negatives; a feature highly relevant in a medical diagnosis for preventing false positives. The F1-Score is a balanced measure between precision and recall, providing a single metric that accounts for both false negatives and positives. ROC-AUC thus quantifies the sensitivity-specificity trade-off across all thresholds, thereby grasping the discriminating ability of the model. These metrics, beyond accuracy, guarantee the reliability and robustness of the model. This is particularly true in high-stakes fields like medicine, where misclassification can be quite expensive.

Research Question 3: How do traditional machine learning algorithms compare with advanced machine learning techniques for early detection of diabetes using a large dataset?

Logistic Regression: This is the most straightforward and interpretable algorithm, but with poor performance when compared to others.

Decision Tree and Random Forest: Both can grant more accuracy and robustness since they handle nonlinear relationships and feature interactions.

Gradient Boosting: Good precision and F1-Score, performing well in imbalanced data.

Gaussian Naïve Bayes: Performed little poor because the algorithm assumed independence between features.

KNN: It is well-balanced in its performance but computationally intensive for large datasets.

Advanced techniques like ensemble stacking and ensemble stacking using GridSearchCV outperformed traditional models and merged their strengths with a view to optimising hyperparameters, reducing overfitting, and improving generalisation for the detection of early diabetes.

5.4 Relation with Existing Literature:

The findings in this research agree with a few past studies, and this research has extended to use some advanced techniques, which is different from few studies those got stuck with only traditional methods.

Some models like SVM, Random Forest, and logistic regression were found effective by Aishwarya Mujumdar and KM Jyoti Rani, but the ensemble methods had superior performance compared to the traditional methods, which were given by studies by Mujumdar & Vaidehi (2019) and Rani (2020). Similarly, Sunita Varma and Mitushi Soni (2020) also worked on the benefits of ensemble techniques to improve the predictive accuracy. This research states the importance of choosing the large dataset and wide range of performance metrics to provide an accurate and robust model to detect diabetes by addressing the gaps in the past studies.

5.5 Critical Analysis:

The main aim of this study is to use a large dataset and comprehensive set of performance metrics to provide a robust evaluation of machine learning models for early diabetes detection. This research succeeded in using a wide range of performance metrics, which provided a thorough assessment of the strengths and weaknesses of each model used on the dataset. It successfully demonstrated the use of advanced techniques like ensemble stacking with GridSearchCV in improving the predictive accuracy for diabetes. The model performance is enhanced by tuning the hyperparameters using GridSearchCV. However, it is very important to consider the time and complexity while using these advanced techniques. As ensemble stacking combined multiple strengths of traditional models, it takes little extra time to show the output compared with the regular models.

Chapter 6. Conclusion and Recommendations

6.1 Summary:

This chapter summarises the findings of research, evaluates the study, and discusses the application of the model in real-world and future research. The limitations of the study are mentioned in this chapter, along with a few directions for further investigations.

Summary of Achievements:

According to the aim of this study, it successfully implemented different machine learning models, including traditional and advanced, evaluating their performances. Each model's performance is measured using multiple metrics like accuracy, precision, sensitivity, specificity, F1-score, and ROC-AUC score. By implementing ensemble stacking with GridSearchCV, a robust model with enhanced predictive accuracy is achieved.

Critical Evaluation of Study:

The study demonstrated the predictive performance of the advanced machine learning techniques like ensemble stacking with GridSearchCV to detect diabetes. It achieved the highest accuracy, which is aligned with few existing literatures that supported the use of ensemble methods, which improved the accuracy. The high understanding of the models is provided in the research paper using comprehensive evaluation metrics, which avoids the limitations of using single metrics like accuracy. This approach ensured to provide the best model to detect diabetes at early stages.

6.2 Implications:

Healthcare: The incidence of diabetes-related complications can be reduced using these improved predictive models in healthcare. The precautions can be taken at early stages, which reduces the number of patients who would suffer with diabetes after their 50’s. Healthcare providers can use this study to reduce the risk of diabetes for each individual and can provide customised treatment effectively.

Academia: This research contributes to the knowledge on the use of advanced machine learning techniques in the medical field. It highlights how important it is to use a proper dataset, a selective model, and comprehensive evaluation metrics to develop the best model that detects the disease.

6.3 Future Research:

Using ensemble methods in this research and the successful results suggests future research should use more advanced techniques like combining the strengths of different algorithms. This study has provided a framework for evaluating the predictive models, which can be applied for other medical conditions other than diabetes. Using different data preprocessing techniques and feature engineering can also improve the model's performance. Additionally, deep learning models like long short-term memory (LSTM) and convolutional neural networks (CNNs) can be implemented for performance enhancement.

Practical Applications: The insights generated by this research give several practical applications that can be considered.

- The models that are developed in this research can be applied in clinical healthcare systems to assist the professionals in identifying the patients who are at risk of diabetes early.
- These models can also be used in public health campaigns to screen populations and identify the individuals who might benefit from preventive interventions.

- Based on the individual profile, the predictive models can be employed to tailor personalised healthcare plans, which can improve patient outcomes.

6.4 Limitations and Recommendations:

Limitations:

Even though there are good contributions from this research, it even has a few limitations, which are as follows:

- The dataset used in this research is quite large, which is a positive point, but it may not capture all possible risk factors for diabetes. More diverse datasets can be incorporated by future studies.
- Advanced techniques like ensemble stacking and ensemble stacking with GridSearchCV are used, which improves the performance but has few limitations. Ensemble stacking, which combines multiple strengths, would be computationally intensive as it involves training multiple models instead of one. Ensemble with GridSearchCV involves tuning the hyperparameters, which multiples the computational load. So, the combined impact would be time-consuming, especially with the large datasets.
- A specific dataset is used in this research, and the models are developed on that dataset. The results of this research may not generalise across all diabetes-related datasets. Their generalisability for other populations and settings needs to be validated through future research.

Recommendations for Future Research:

Based on the results of this research and the limitations of this study, future recommendations are below:

- To validate and enhance the generalisability of the predictive models, larger and diverse datasets must be used.
- Explore other advanced machine learning algorithms, like combinations of algorithms, to increase the performance of the models.
- Focussing more on the practical implementations of these models in clinical settings and exploring how these practices have an impact on patient outcomes.

6.5 Conclusion:

This research successfully developed and evaluated the advanced machine learning techniques for predicting diabetes in its early stages. It demonstrated the impact of ensemble stacking with GridSearchCV on the chosen dataset. The study has its impact on healthcare, academia, and future research, providing a foundation for future advancements in predictive modelling for diabetes and other medical conditions. By addressing the limitations of the research and following the contributions, future research can continue to improve the accuracy using the predictive models, ultimately enhancing patient care and outcomes.

References:

1. Mujumdar, A. and Vaidehi, V., 2020. Diabetes Prediction using Machine Learning Algorithms. *Procedia Computer Science*, 165, pp.292-299.
2. Rani, K.J., 2020. Diabetes Prediction Using Machine Learning. *International Journal of Computer Sciences and Engineering*, 8(4), pp.272-276.
3. Soni, M. and Varma, S., 2020. Diabetes Prediction using Machine Learning Techniques. *International Journal of Engineering Research & Technology (IJERT)*, 9(9), pp.189-192.
4. Sonar, P. and JayaMalini, K., 2019. Diabetes Prediction Using Different Machine Learning Approaches. *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, pp.367-372.
5. Yahyaoui, A., Jamil, A. and Rasheed, J., 2020. A Decision Support System for Diabetes Prediction Using Machine Learning and Deep Learning Techniques. *IEEE Access*, 8, pp.107225-107233.
6. Khanam, J.J. and Foo, S.Y., 2021. A comparison of machine learning algorithms for diabetes prediction. *Heliyon*, 7(3), e05847.
7. Sarwar, M.A., Kamal, N., Hamid, W. and Shah, M.A., 2019. Prediction of Diabetes Using Machine Learning Algorithms in Healthcare. *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp.1-6.
8. Saru, S. and Subasree, S., 2019. Analysis and prediction of diabetes using machine learning. *SSRN Electronic Journal*.
9. Sisodia, D. and Sisodia, D.S., 2018. Prediction of Diabetes using Classification Algorithms. *Procedia Computer Science*, 132, pp.1578-1585.
10. Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y. and Tang, H., 2018. Predicting Diabetes Mellitus with Machine Learning Techniques. *Frontiers in Genetics*, 9, p.515.
11. Breiman, L., 2001. Random forests. Machine Learning, 45, pp.5-32.
12. Liaw, A. and Wiener, M., 2002. Classification and regression by random Forest. R News, 2(3), pp.18-22.
13. Friedman, J., 2009. The elements of statistical learning: Data mining, inference, and prediction. 2nd ed. New York: Springer.
14. Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5), pp.1189-1232.
15. Chen, T. and Guestrin, C., 2016. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.785-794.
16. Natekin, A. and Knoll, A., 2013. Gradient boosting machines, a tutorial. Frontiers in Neurorobotics, 7, p.21.
17. Friedman, J.H., 2002. Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4), pp.367-378.
18. Zhang, H., 2004. The optimality of naive Bayes. Aa, 1(2), p.3.
19. Rish, I., 2001. An empirical study of the naive Bayes classifier. In IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 3(22), pp.41-46.

20. Murphy, K.P., 2012. Machine learning: a probabilistic perspective. Cambridge, MA: MIT Press.

21. Cover, T. and Hart, P., 1967. Nearest neighbour pattern classification. IEEE Transactions on Information Theory, 13(1), pp.21-27.

22. Fix, E., 1985. Discriminatory analysis: nonparametric discrimination, consistency properties. Vol. 1. USAF School of Aviation Medicine.

23. Aha, D.W., Kibler, D. and Albert, M.K., 1991. Instance-based learning algorithms. Machine Learning, 6, pp.37-66.

24. Rokach, L., 2010. Ensemble-based classifiers. Artificial Intelligence Review, 33, pp.1-39.

25. Breiman, L., 1996. Bagging predictors. Machine Learning, 24, pp.123-140.

26. Wolpert, D.H., 1992. Stacked generalization. Neural Networks, 5(2), pp.241-259.

27. Katsarou, A., Gudbjörnsdottir, S., Rawshani, A., Dabelea, D., Bonifacio, E., Anderson, B.J., Jacobsen, L.M., Schatz, D.A. and Lernmark, Å., 2017. Type 1 diabetes mellitus. Nature Reviews Disease Primers, 3(1), pp.1-17.

28. DeFronzo, R.A., Ferrannini, E., Groop, L., Henry, R.R., Herman, W.H., Holst, J.J., Hu, F.B., Kahn, C.R., Raz, I., Shulman, G.I. and Simonson, D.C., 2015. Type 2 diabetes mellitus. Nature Reviews Disease Primers, 1(1), pp.1-22.

29. Buchanan, T.A. and Xiang, A.H., 2005. Gestational diabetes mellitus. The Journal of Clinical Investigation, 115(3), pp.485-491.

30. Russell, S.J. and Norvig, P., 2016. Artificial Intelligence: A Modern Approach. 3rd ed. Upper Saddle River, NJ: Pearson.

31. Gordon, A.D., Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J., 1984. Classification and Regression Trees. Biometrics, 40(3), p.874.

32. Hosmer Jr, D.W., Lemeshow, S. and Sturdivant, R.X., 2013. Applied Logistic Regression. 3rd ed. New York: John Wiley & Sons.

33. MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1(14), pp.281-297.

34. Jolliffe, I.T., 2002. Principal component analysis for special types of data. In Principal Component Analysis (pp.338-372). New York: Springer.

35. Sutton, R.S. and Barto, A.G., 2018. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA: MIT Press.

36. Little, R.J. and Rubin, D.B., 2019. Statistical Analysis with Missing Data. 3rd ed. Hoboken, NJ: John Wiley & Sons.

37. Jensen, P.B., Jensen, L.J. and Brunak, S., 2012. Mining electronic health records: towards better research applications and clinical care. Nature Reviews Genetics, 13(6), pp.395-405.

38. Dwork, C. and Roth, A., 2014. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 9(3–4), pp.211-407.

39. Huang, J. and Ling, C.X., 2005. Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 17(3), pp.299-310.