University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project

7PAM2002-0901-2024

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

LEVERAGING MACHINE LEARNING APPROACHES FOR DIABETES PREDICTION AND ANALYSIS

Student Name and SRN:

FRIDAY ODEH OVBIRORO, 22034665

Supervisor: SOPHIE KOUDMANI

Date Submitted: 3rd JANUARY 2025

Word Count: 7496

GitHub address:

1. Code-Large Dataset:

   https://github.com/fryohatfield/MSc-Project-Code-and-Slide/blob/main/Diabetes1_Predictions_Opt_27_12_24.ipynb

2. Code-Small Dataset:

https://github.com/fryohatfield/MSc-Project-Code-and-Slide/blob/main/Diabetes2_Prediction_optimised_small_2024-12-04.ipynb

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6).

I have not used chatGPT, or any other generative AI tool, to write the report or code (other than where declared or referenced).

I did not use human participants or undertake a survey in my MSc Project.

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

Student SRN number: 22034665

Student Name printed: FRIDAY ODEH OVBIRORO

Student signature: FRIDAY ODEH OVBIRORO

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Abstract

This project investigates the use of three machine learning models, Random Forest, XGBoost and Support Vector Machine (SVM) to predict diabetes progression and analyse related patterns. The study focuses on comparing the performance of these models while identifying significant factors contributing to diabetes progression through data exploration. Two datasets were analysed: a large dataset from the Behavioural Risk Factor Surveillance System (BRFSS) with 253,680 records and a small dataset from the Pima Indians Diabetes dataset, containing 768 records. Data preparation involved handling duplicates, scaling and balancing classes to ensure fair model evaluation. findings Results indicated that Random Forest performed consistently well across datasets. Even with the optimization challenges, it maintained strong performance with the large dataset and achieved 80.1% accuracy on the small dataset. XGBoost demonstrated competitive results, attaining the highest accuracy of 80.5% on the small dataset. SVM struggled with the large dataset but showed moderate improvement on the small dataset. The research concludes that Random Forest is the most reliable model for predicting diabetes progression with XGBoost an alternative. These support practical application in healthcare for early detection and personalized diabetes management. Alongside recommendations for further research to improve model clarity, address class imbalance and incorporate additional features for better predictions.

Contents

1.0. Introduction

1.1 Background

Diabetes is a chronic metabolic disorder characterised by high blood glucose levels due to the inability of the pancreas to produce adequate insulin for the body function. This condition, if left unmanaged, can lead to severe health complications, resulting to heart disease, kidney failure and amputations. The disease's increasing prevalence poses a global health crisis, estimating that diabetes cases will surpass 600 million by 2040 (World Health Organisation, 2023). The alarming projection of diabetes highlights the urgent need for preventive measures, emanating from early detection and effective care management strategies (Houngue & Bigirimana, 2022).

Traditional diagnostic approaches for diabetes, such as fasting plasma glucose (FPG) tests, oral glucose tolerance tests (OGTTs), and glycated haemoglobin (HbA1c) measurements are widely used but often limited in sensitivity or accuracy, particularly for early-stage and prediabetic cases. These methods do not fully capture the complex relationships among multiple risk factors, including genetic predisposition, demographic attributes, lifestyle habits and biochemical markers. Recent advancements in Machine learning (ML) have opened new possibilities for diabetes prediction, allowing researchers to analyse diverse risk factors and identify subtle, complex patterns in data (Khongorzul Dashdondov et al., 2024).

Machine learning approaches are increasingly applied to medical science due to their ability to process vast datasets and uncover valuable insights. By leveraging ML models likes Random Forest, XGBoost and Support Vector (SVM). Researchers and practitioners can develop predictive models that more accurate and reliable than traditional methods. For example, models developed by Kasula (2023) and Rani (2020) achieved high accuracy rate of 85% for Random Forest and 99% for Decision Trees. demonstrating the effectiveness of ML in diabetes prediction. Also, ensemble techniques such as those used by Jain et al. (2024), have proven effective in improving model performance across diverse patient demographics. The innovation and involvement of ML technology for diabetes prediction has led to improved and precise healthcare management interventions.

1.2. Problem Statement

Diabetes is an intricate, compound medical condition that is influenced by a wide range of genetic, environmental and lifestyle factors, making it challenging to detect in its early stages using conventional diagnostic techniques alone. While traditional methods provide basis for diabetes test, they often fail to accurately screen individuals at risk before the disease symptoms manifest, thereby exposing the individual to chances of complications. Additionally, the linear nature of these methods limits their capabilities in capturing the non-linear associations among various indicators of diabetes advancement.

The use of machine learning in diabetes prediction aims to address these challenges by effectively capturing complex patterns related with the disease onset. However, there is a significant gap in choosing the model that could accurately perform best

with robust and reliable predictions for diabetes diagnosis across varied patient groups

1.3. **Justification of the Study**

The growing rate of diabetes and the related health risks emphasize the need for innovative, precise and proactive procedure to its detection and management. Early and accurate diagnosis is essential for effective action, as it enables healthcare professionals to mitigate complications and improve patient outcomes.

1.4. **Research Questions**

i. Model Performance Comparison: How do Random Forest, XGBoost and Support Vector Machines (SVM) compare in their predictive performance for diabetes progression?

ii. Exploratory Data Insights: What insights, trends or patterns can be uncovered through Exploratory Data Analysis (EDA) in relation to diabetes progression?

1.5. **Objectives**

i. To develop machine learning models for predicting early detection of diabetes.

ii. To perform Exploratory Data Analysis for identifying trends and patterns in diabetes progression.

iii. To compare insights from two datasets – large and small

iv. To evaluate and compare the performance of different machine learning models.

v. To provide actionable insights for healthcare providers, to improve health outcomes.

2.0. **Literature Review**

This chapter provides a comprehensive review of existing papers on the use of machine learning approaches in diabetes prediction and analysis. It explores the key studies that have contributed to understanding how machine learning models can enhance the accuracy of diabetes diagnostics and the early identification of risk factors. This literature review plans to put in perspective the research problem, identify gaps in the current knowledge and provide a baseline for the chosen methodologies in this work.

Building upon the growing body of research advocating for machine learning in healthcare, this provides a comparative analysis that highlights the efficacy of advanced machine learning models in diabetes risk prediction (Prasetyo and Izdihar, 2024). Their study addresses the critical global health challenges posed by diabetes and underscores the importance of effective predictive methodologies to mitigate its widespread impact. By analysing three prominent ML models; Gaussian Naïve Bayes, Decision Tree and Artificial Neural Network (ANN). The authors asses each model's performance in diabetes risk prediction. Utilising the Behavioural Risk Factor Surveillance System (BRFSS) dataset, a comprehensive resource for health-related risk behaviours. The paper reveals that the ANN model achieves the highest accuracy with an impressive 84.73%, thereby outperforming both the Gaussian Naïve Bayes and Decision Tree models. This finding stresses the ANN's potentials in advancing diabetes risk prediction. The study's contributions are significant as they pave the way for developing precise diagnostic tools and customised interventions, enhancing diabetes management and addressing its societal burden (Prasetyo, Izdibar & Nabiilah, 2024).

Expanding on the application of machine learning in diabetes prediction, Sivaranjani et al. (2021) focus on using predictive models to prevent diabetes by analysing patterns within a dataset comprising both diabetic and non-diabetic individuals. Leveraging the diabetes 130-US hospitals dataset, which spans from 1999 to 2008. The research employs several machine learning models, including Logistic Regression, K-Nearest Neighbors (KNN), Random Forest and Support Vector machine (SVM). Findings reveal that feature selection plays a pivotal role in optimising model performance, with Rando Forest model standing out by impressive 99.8% accuracy using the raw dataset. Logistic Regression, though initially less accurate, showed potential for enhancement when combined with multiple sampling techniques.

In response to the rising occurrence of diabetes and its acute health implications, Kasturi (2024) investigated a range of machine learning and deep learning models for detection of diabetes. Utilising the Pima Indian Diabetes dataset, this study aims to classify individuals as diabetic and non-diabetic by employing various machine learning algorithms, including Logistic Regression (LR), K-Nearest Neighbours (KNN), Random Forest (RF) and Support Vector Machine (SVM). In addition, a Multi-Layered Feed Forward Neural Network (MLFNN) is implemented from a deep learning perspective. Among these approaches, the MLFNN achieved the highest

accuracy at 92%, suggesting its capabilities for even greater accuracy if applied to larger datasets.

N Nagarjuna and Lakshmi (2024) explore the application of machine learning techniques in diabetes prediction, with a concentration of leveraging various algorithms to enhance predictive accuracy and disease management. Their study employs multiple machine learning models, such as Logistic Regression, Naïve Bayes, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and XGBoost, each of which is evaluated for its effectiveness in classifying diabetic and non-diabetic individuals. Random Forest and XGBoost outshine others with notable performance of 82% and 80% respectively.

Gupta (2024) addresses the challenges of predicting diabetes accurately, especially with issues like outliers and missing data in labelled datasets. To handle these issues, the research introduces a comprehensive prediction framework that incorporates essential data preprocessing techniques, including outlier rejection, missing value imputation, data standardisation, feature selection and K-fold cross-validation. The framework employs various machine learning algorithms, such as K-nearest Neighbour (K-NN), Decision Trees, Random Forest, AdaBoost, Naïve Bayes, XGBoost, and Multi-Layer Perception (MLP). A key innovation in this study is the introduction of a weighted ensembling techniques designed to improve prediction accuracy. To further optimise results, hyperparameter tuning is carried out using grid search. The experiment with the Pima Indian Diabetes dataset, demonstrate the effectiveness of the ensembling classifier, with sensitivity of 0.789, specificity of 0.934, false omission rate of 0.092, a diagnostic odds ratio of 66.234 and an AUC of 0.950, which is a 2% improvement over existing methods.

Gufran et al. (2024) address the growing prevalence of diabetes and the importance of early prediction to prevent its onset. Their research uses lifestyle data from the UCI database and evaluates the effectiveness of six machine learning techniques (MLTs) for diabetes prediction. These techniques include Logistic Regression (LR), Decision Tree Classification (DTC), Random Forest Classification (RFC), Support Vector Classification (SVC), and K-Nearest Classification (KNC). The research shows that Logistic Regression outperforms other algorithms with an accuracy of 93%. By examining patients' lifestyle data and applying techniques like embedding, filter and hybrid feature selection methods, the research demonstrates the advantages of using refined input characteristics.

Elmenshawy et al. (2024) address the global impact of diabetes, which affects 537 million people worldwide and contributes to serious health issues like heart disease, kidney damage and diabetic retinopathy. This study introduces a diabetes prediction framework built using a private Bangladeshi dataset and various machine learning algorithms, such as Decision Tree, SVM, Random Forest, Logistic Regression, K-Nearest Neighbour (KNN) and XGBoost. The XGBoost classifier combined with the ADASYN technique for handling imbalance data, had an accuracy of 80%. Moreover, a stacked ensemble of three classifiers achieved an outstanding accuracy of 99.3%. The framework's adaptability is demonstrated by the use of domain adaptation techniques, enhancing its predictive capability across various settings.

3.0 **Methodology**

This section outlines the research techniques I applied, for predicting diabetes risk and analysing patterns related to the disease. It details the approach used for data collection, ethical considerations, Exploratory Data Analysis (EDA) and Preprocessing.

3.1. **Overview**

In this study my objective is to predict diabetes risk and analyse patterns associated with the disease using health-related survey responses. I utilized two different datasets for comparison: one large dataset from the CDC's Behavioural Risk Factor Surveillance System (BRFSS), and a small dataset sourced from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), the Pima Indians Diabetes dataset.

For this analysis, I used three different models: namely Random Forest, XGBoost and Support Vector Machine (SVM). These models were chosen due to their ability to handle complex, high-dimensional data and their effectiveness in classification tasks. Additionally, I performed EDA to discover meaningful insights and patterns within the data, and also, I preprocessed the data for suitability. After building the models, I evaluated their performance using metrics, such as Accuracy, Precision, Recall and F1 Score.

3.2. **Data Collection**

The two datasets used in this study are publicly available on Kaggle, ensuring their accessibility for academic and research purposes. These datasets were chosen for their relevance to diabetes prediction and the variety of features they offer for analysis.

i. **Large dataset:**
This dataset is part of the CDC's Behavioural Risk Factor Surveillance System (BRFSS), collected in 2015. It contains 253,680 survey responses from the U.S. residents, with 22 features associated with health indicators that are important in understanding diabetes risk. The dataset has a target variable of **Diabetes_012** with three classes containing **0 for no diabetes**, **1 for prediabetes** and **2 for diabetes.**

See Link: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset

ii. **Small dataset:**
This dataset was collected in 1990 and was sourced from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) and focuses on health indicators related to diabetes among the Pima Indians in Arizona, USA. It consists of 768 records with 8 features. The target variables indicates whether an individual has **diabetes** or **not,** making it suitable for binary classification tasks.

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

3.2.1. Ethical Considerations

i. **GDPR Compliance:** The two datasets used in this study are anonymized, ensuring that personal identifiable information (PII) cannot be traced back to individuals, in line with GDPR principles, such as data anonymization, lawful usage and fairness. Since these datasets are publicly available and do not involve active collection of personal data, they do not fall under stricter GDPR requirements related to personal data collection.

ii. **UH Ethical Policies:** As the datasets are publicly accessible and anonymized, University of Hertfordshire (UH) Ethics Committee approval was not required. This study relies on secondary data from Kaggle, which complies with UH ethical policies for responsible data usage in academic research, ensuring participant privacy and confidentiality.

iii. **Permission to Use the Data:** The datasets are governed by Kaggle's term of use, which allows for academic and non-commercial research purposes. There is no need to obtain explicit permission from individual participants, as the data is openly shared for research under these terms.

iv. **Ethical Data Collection:** The study does not involve active data collection or surveys. The anonymised data ensures that no personal information is used inappropriately. The findings are intended solely for general knowledge and research purposes, not for diagnosing or treating individual cases, thus minimizing the risk of misusing the results.

3.3. Exploratory Data Analysis (EDA)

The exploratory data analysis (EDA) of both diabetes datasets revealed class imbalances in the target variables, which were addressed using Synthetic Minority Over-sampling Technique (SMOTE).

The large dataset offered a comprehensive analysis, identifying key risk factors for diabetes progression, such as high blood pressure, cholesterol and BMI, through count and bar plots. It also highlighted protective factors like physical activities and vegetable consumption, which were more common in non-diabetic individuals. Additionally, mental and physical health declined with diabetes progression. Demographic factors like age and income were linked to higher diabetes prevalence. Correlation analysis confirmed strong relationships between these factors and diabetes.

The small dataset focused on clinical features, including glucose, BMI, insulin levels, pregnancies and genetic predisposition. Bar plots showed that diabetic individuals had higher glucose, insulin and BMI levels, with clear relationships between pregnancies, age and diabetes. Correlation analysis confirmed strong association between glucose, BMI and age with diabetes progression.

The following plots are particularly valuable, as they effectively highlight key insights relevant to the research objectives vis-à-vis the research questions for both diabetes datasets.

3.3.1. **Comparative Analysis of Key Insights using Visualizations from Large and Small Diabetes Datasets**

Class Distribution of Target Variables

Fig 1a. Large dataset

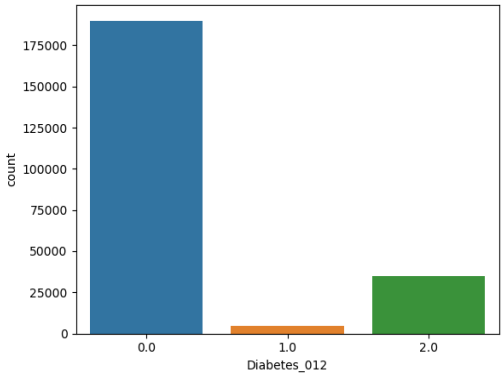
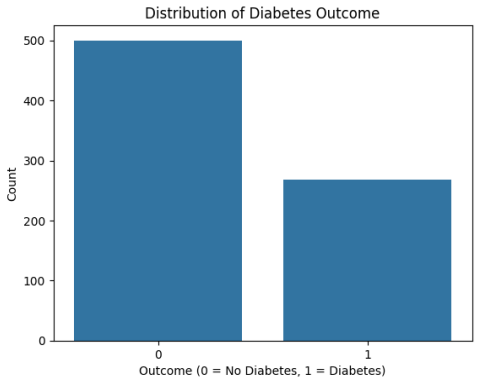


Fig 1b. Small dataset



The large dataset shows a severe class imbalance, with 0.0 for non-diabetic at about 180,000 observations, 2.0 for diabetic at around 30,000 and 1.0 for prediabetic being negligible. The small dataset has a moderate imbalance with 500 non-diabetic and nearly 300 diabetic cases. To address this issue, SMOTE is crucial for balancing the minority classes for fair model performance.

BMI Across Diabetes Categories

Fig 2a. Large dataset

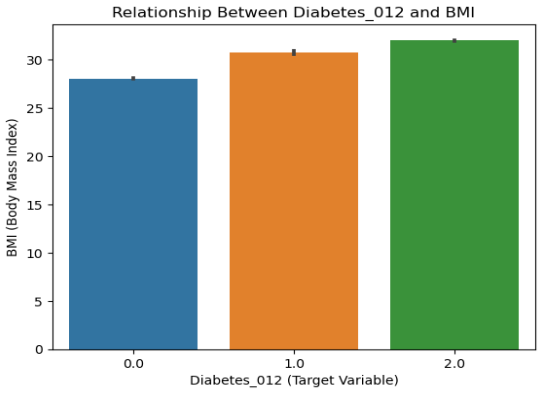
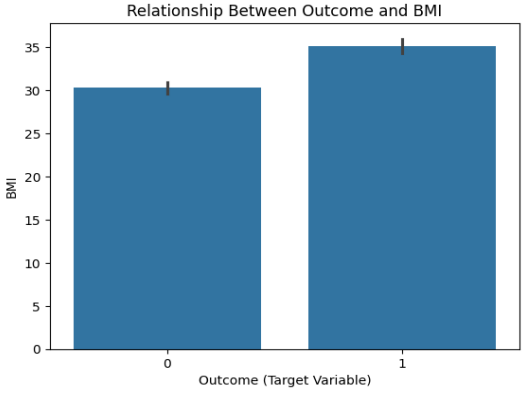


Fig 2b. Small dataset



In the large dataset, the average BMI for non-diabetic individuals (0.0) is approximately 30, for prediabetics (1.0) is around 31 and for diabetics (2.0) it's slightly exceeds 31. This indicates a positive correlation between BMI and diabetes progression. In the small dataset, non-diabetics (0) have an average BMI of 30, while

diabetics (1) have a higher BMI of 35. Both plots show that higher BMI is strongly associated with diabetes, making it a key risk factor.

Age vs. Target Variable (Age distribution Across Diabetes)

Fig 3a. Large dataset

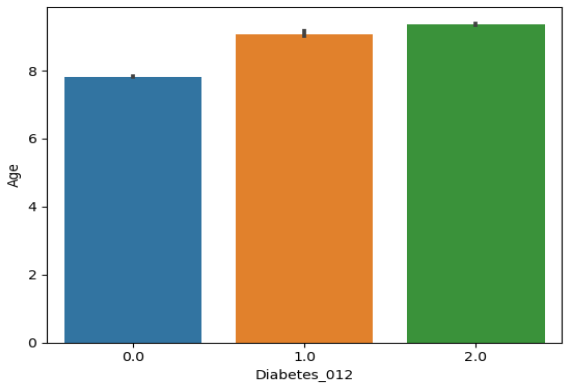
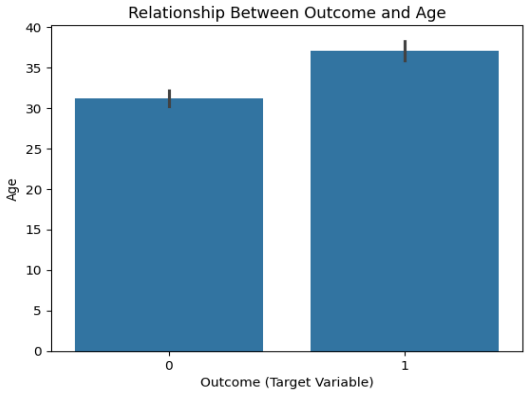


Fig 3b. Small dataset



In the large dataset, the average age is 8 for non-diabetics (0.0), around 9 for pre-diabetics (1.0), and slightly above 9 for diabetics (2.0). The average age in the small dataset is approximately 30 for non-diabetics (0) and 35 for diabetics (1). This demonstrates that older individuals are at a higher risk of diabetes. Both plots confirm that age is strongly associated with diabetes progression, making it a critical feature for predictive modelling.

Correlation Analysis of Features Influencing Diabetes Progression

Fig 4a. Large dataset

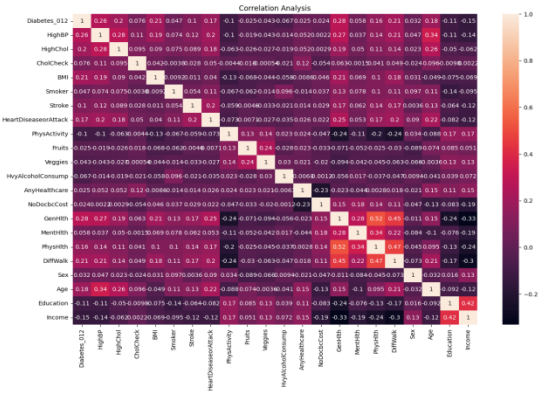
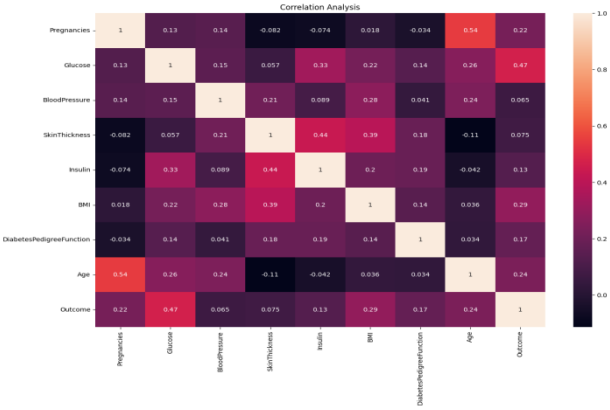


Fig 4b. Small dataset



In the large dataset, the target variable Diabetes_012 is moderately correlated with Age (0.34), High Blood Pressure (0.26) and Cholesterol (0.28), making them key predictors. Socioeconomic factors like Income and Education (0.42) also show a positive correlation with diabetes risk. General health is strongly correlated with

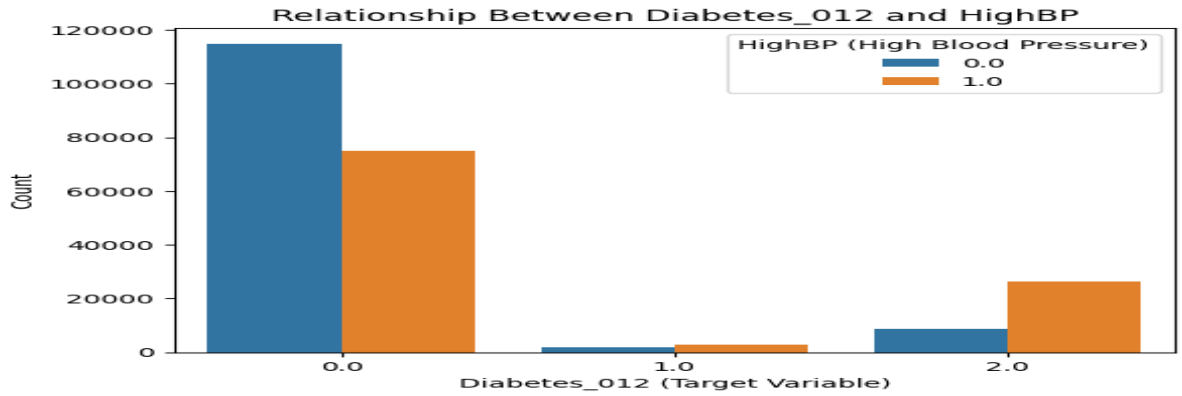
physical health (0.52) and Difficulty Walking has strong positive correlations with both general health (0.45) and physical health (0.47). While Difficulty Walking (0.47) is more strongly correlated with diabetes progression than Age (0.34), it is likely a symptom of advanced diabetes rather than a cause. Note that correlation measures the strength of the relationships between two variables but does not imply causation.

In the Small dataset, the target variable Outcome is moderately correlated with Age (0.54), Glucose (0.47) and BMI (0.29), indicating these as the strongest predictors. Other variables show weak correlations and there is no significant multicollinearity. Both datasets emphasized that Age and BMI are consistent predators of diabetes.

3.3.2. **Additional Visualizations with Key Insights from Large and Small Diabetes Datasets**

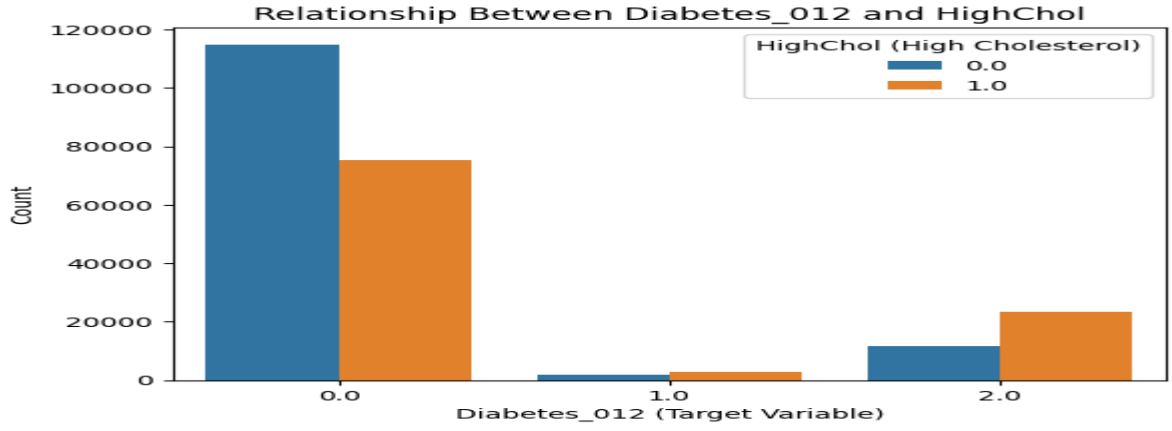
i. **Large dataset** (Blue bar = No, Orange bar = Yes)

Fig 1. Relationship Between Diabetes and High Blood Pressure



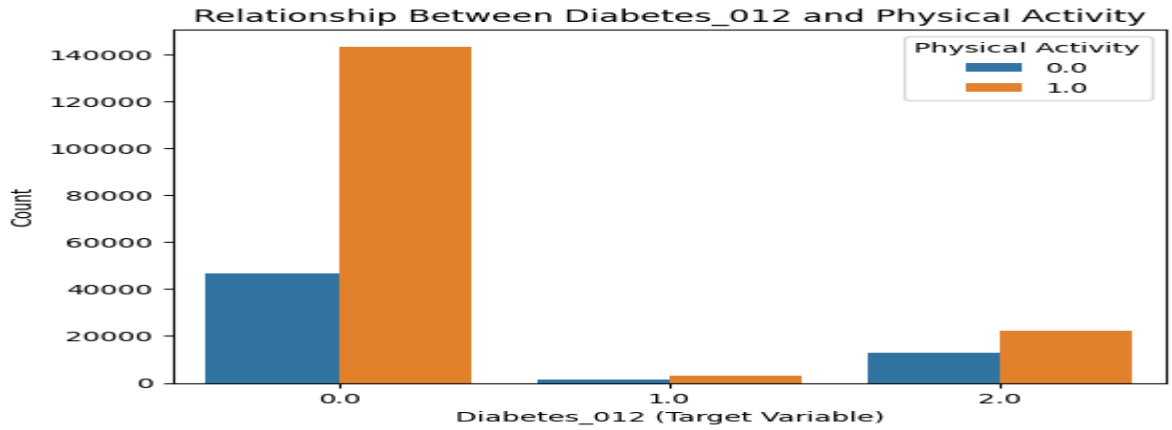
In the above chart, among the non-diabetics, approximately 61% (110,000) do not have high blood pressure, while about 39% (70,000) do. In pre-diabetics, high blood pressure is slightly more common. Among diabetics, nearly 71% (25,000) have high blood pressure and about 29% (1o,000) do not. This trend aligns with a moderate correlation of 0.26 between high blood pressure and diabetes progression, highlighting high blood pressure as a significant risk factor.

Fig 2. Prevalence of High Cholesterol Across Diabetes Progression



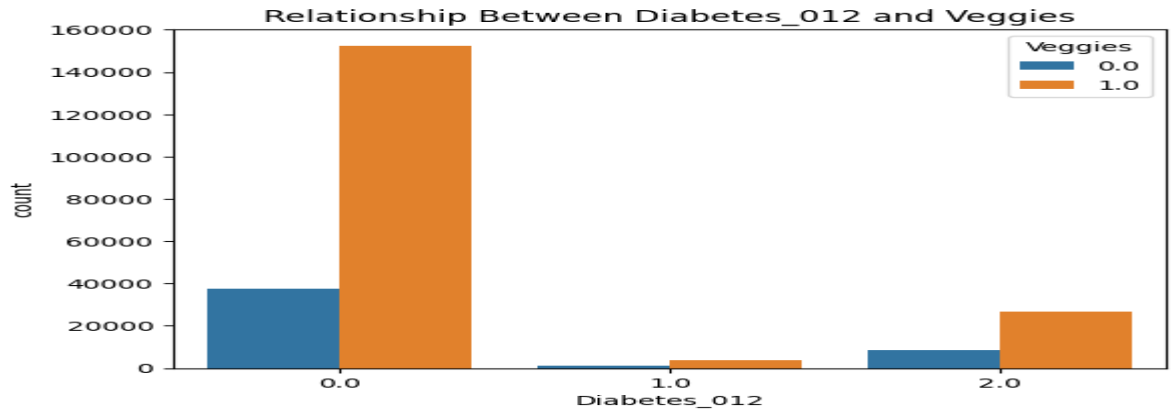
The chart illustrates that, among non-diabetics (0.0), approximately 110,000 individuals (61%) do not have high cholesterol, while about 70,000 (39%) do. In the pre-diabetic group (1.0), high cholesterol affects a slight majority of about 51%. Among diabetics (2.0), nearly 25,000 individuals (71%) have high cholesterol, whereas about 10,000 (29%) do not. This trend indicates high cholesterol as a major risk factor.

Fig 3. Physical Activity and Diabetes Progression



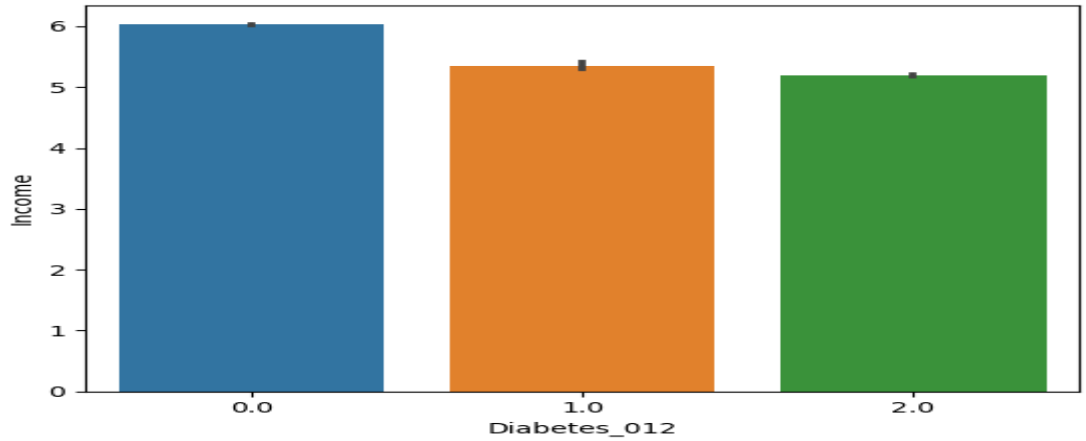
The chart demonstrates that physical activity declines slightly with diabetes progression. Among non-diabetics (0.0), 74% (140,000) are active and 26% (50,000) are not. For pre-diabetics (1.0), 55% are active and 45% are inactive. Among diabetics (2.0), 63% (25,00) remain active and 37% (15,000) are inactive. The weak negative correlation (-0.10) suggests physical activity offers modest protection against diabetes.

Fig 4. Relationship between Vegetable Consumption Across Diabetes Progression



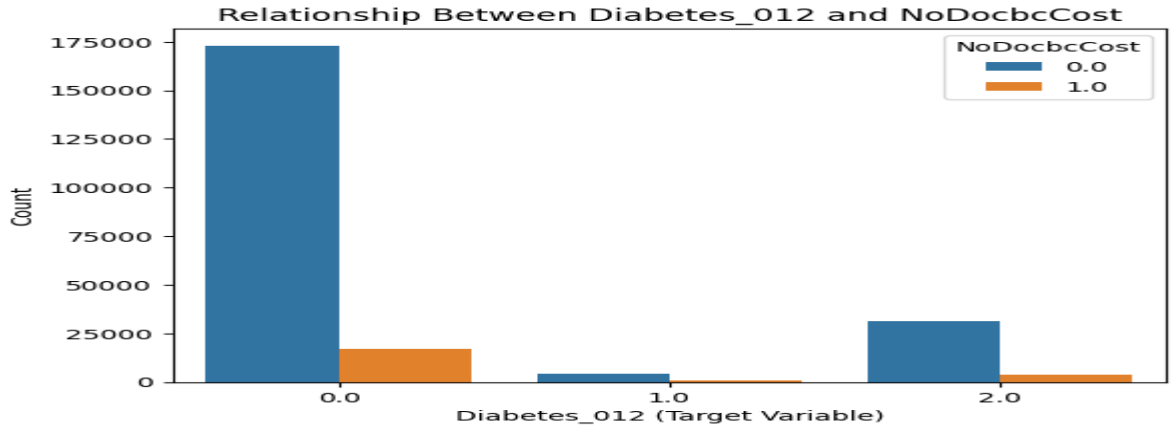
In the plot, vegetable consumption slightly decreases with diabetes progression. Among non-diabetics (0.0), approximately 74% consume vegetables, while about 26% do not. Among diabetics (2.0), about 71% consume vegetables and 29% are not. A weak negative correlation (-0.14) suggests a minor protective effect of vegetable consumption against diabetes.

Fig 5. Income Levels Across Diabetes Progression Categories



The plot shows that income decreases by approximately 8% as diabetes progresses, with non-diabetics averaging about 6 and diabetes about 5.5. A weak negative correlation (-0.15) links lower income to higher diabetes risk, showcasing potential socioeconomic barriers.

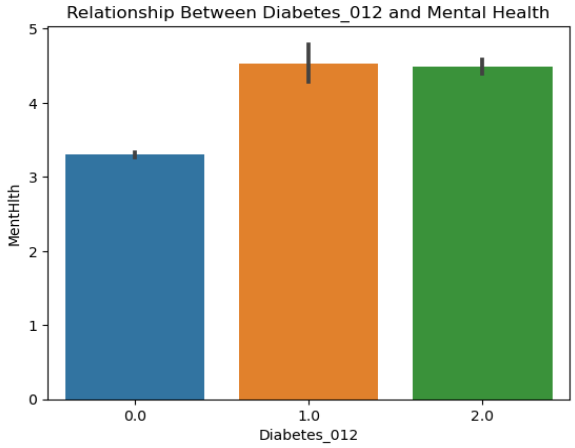
Fig 6. Cost-Related Barriers to Healthcare Across Diabetes Progression



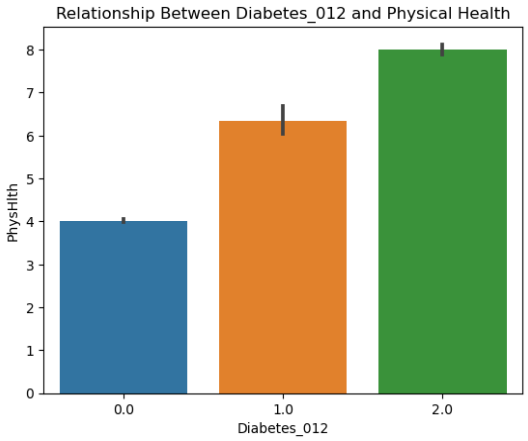
Cost-related barriers slightly increase with diabetes progression. Among non-diabetics, 88% had no barriers, compared to 78% for diabetics. This reflects a weak positive correlation of 0.07.

Fig 7. Impact of Diabetes Progression on Mental and Physical Health

7a. Mental Health



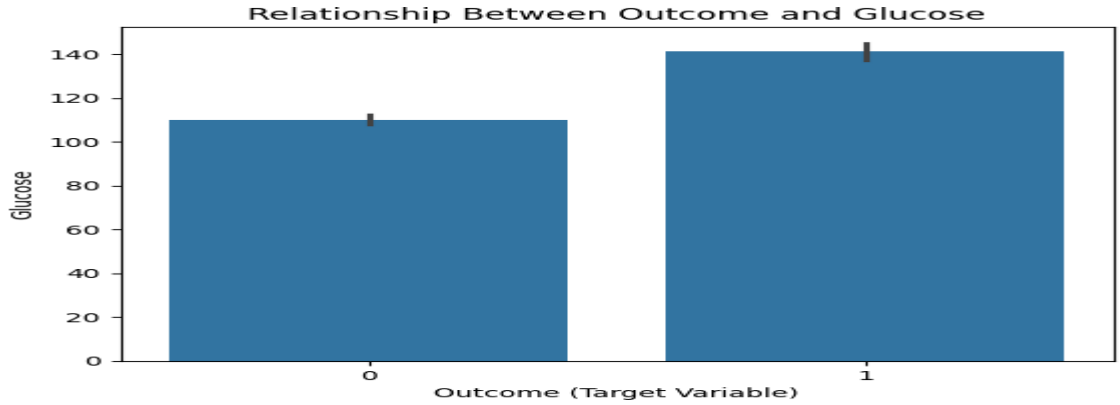
7b. Physical Health



As diabetes progresses, mental health worsens by 50% and physical health by 167%. Physical health shows a stronger decline. Positive correlations stood at 0.16(weak) and 0.27 (moderate) for mental and physical health respectively.

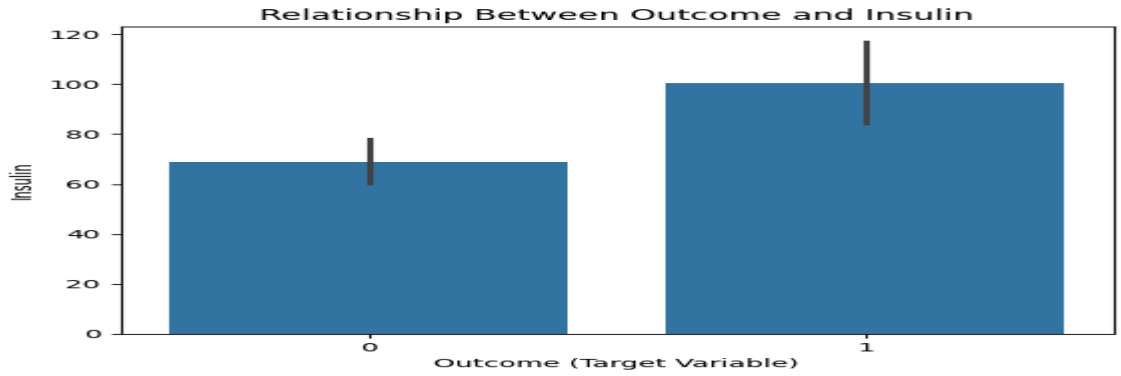
ii. Small Dataset

Fig 1. Relationship between Diabetes Presence and Glucose Levels



Non-diabetics account for about 44% of average glucose levels, diabetics for 56%, with about 27% (110 mg/dL) glucose increase from non-diabetics to diabetics. Its strong positive correlation of 0.47, confirms glucose levels as a key diabetes predictor.

Fig 2. Diabetes Presence and Insulin Levels

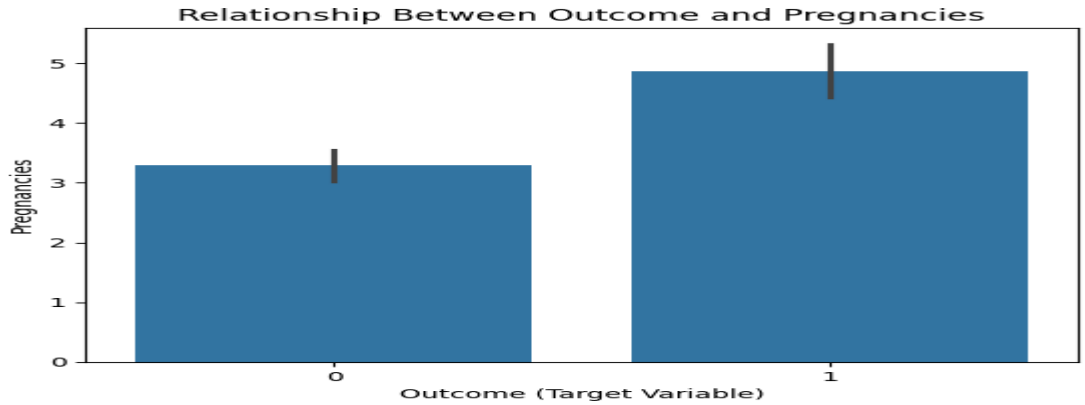


From the plot, diabetic individuals have about 43% higher average insulin level (600 pmol/L) compared to non-diabetics (420 pmol/L). A weak positive correlation (0.13) signifies a modest association between high insulin levels and diabetes.

Fig 3. Diabetes Presence and Genetic Predisposition

Diabetic individuals have a 31% higher average Diabetes Pedigree Function (0.55) compared to non-diabetics (0.42). A moderate positive correlation of 0.17 points out the role of genetic predisposition in diabetes risk.

Fig 4. Diabetes Presence and Number of Pregnancies



The plot reveals that people with diabetes have 67% more pregnancies on average (5) than non-diabetics individuals (3). A moderate positive correlation (0.22) expresses a link between pregnancies and diabetes risk, potentially due to gestational diabetes.

3.4. Data Preprocessing

Preprocessing is a vital step that directly impacts the accuracy, efficiency and interpretability of machine learning models, by ensuring that the data is appropriately prepared and suitable. The following preprocessing steps were carried out.

i. **Data Cleaning:** The large dataset contained duplicate values; this was handled by dropping the duplicates. The small dataset does not contain duplicates.
ii. **Handling Class Imbalance:** Both datasets showed class imbalances, this was addressed by applying Synthetic Minority Over-Sampling Techniques (SMOTE) to balance the classes and ensure that the models could effectively learn from all classes.
iii. **Data Splitting:** I split the dataset into training and testing sets, using an 80-20 split. I used the training data to train the models and the testing data to evaluate their performance.
iv. **Feature Scaling:** I applied StandardScaler to normalize the features so they have a mean of 0 and a standard deviation of 1. The scaler is fitted to the training data and applied to the testing data

Imbalanced Data Handled Using SMOTE

Fig 1a. Large Dataset

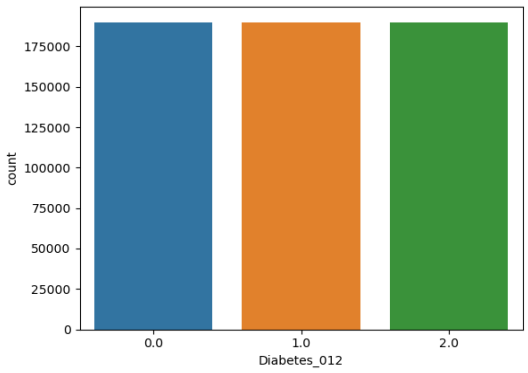
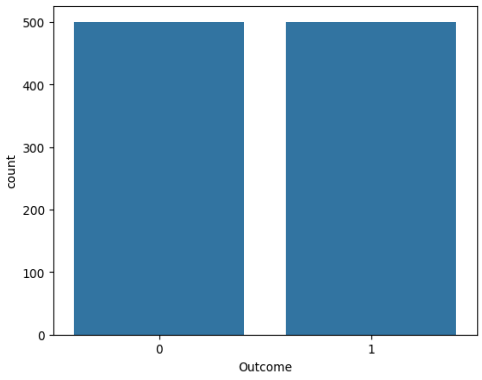


Fig 1b. Small Dataset



The two datasets are now evenly distributed. The application of SMOTE effectively balances the class distributions, ensuring fair training of the models and reducing bias toward majority classes.

3.5. Machine Learning Algorithms

In this project, I used three different machine learning algorithms to predict diabetes status and evaluate their effectiveness, as they are fundamental tools in extracting patterns and making predictions from data. They include Random Forest, XGBoost and Support Vector Machine (SMV). These algorithms stand out for their capabilities to handle classification and regression tasks effectively.

1. **Random Forest:** This is an ensemble learning algorithms that build multiple decision trees during training and merges their outputs (averaging for regression or majority vote for classification). It operates on the principle of bagging (Bootstrap Aggregation), where subsets of the data are created with replacement and used to train individual decision trees. During tree construction, features for splits are randomly selected which introduces diversity and reduces correlation among trees.
   When applied to diabetes large dataset, Random Forest handles the high dimensionality of features effectively, providing robust predictions while mitigating the risk of overfitting. It can effectively manage missing data and noise, making it suitable for real-world healthcare datasets. For a small dataset, Random Forest's ensemble approach stabilizes predictions by reducing variance, though care must be taken to avoid overfitting due to limited data.

2. **XGBoost:** The eXtreme Graadient Boosting (XGBoost) is a high-performance implementation of gradient boosting that focuses on both speed and accuracy. It works by building decision trees subsequentially, where each new tree aims to correct the errors of the previous one. This is achieved by minimizing a loss function, such as mean squared error for regression or log loss for classification, through gradient descent.
   With large diabetes dataset, XGBoost excels in delivering high predictive accuracy by effectively handling missing values and leveraging parallel processing for faster training. The algorithm's regularization techniques ensure robust generalization. This, even with complex feature interactions common in large medical datasets. For small diabetes dataset, XGBoost can still perform well but overfitting is a potential concern. Proper hyperparameter tuning, such as adjusting the learning rate and depth becomes key to prevent overfitting and optimize performance.

3. **Support Vector Machine (SVM):** SVM is a supervised learning algorithm designed to solve classification and regression problems by identifying the optimal hyperplane that separates data into classes. The objective is to maximize the margin which is the distance between the hyperplane and the nearest data point. (Support Vector) from each class. This ensures better generalization and robustness to small changes in the dataset.
   In the case of a large diabetes dataset, SVM's performance can be computationally intensive due to the high dimensionality and size of the data. The use of kernel functions, while powerful may significantly increase computation time, making it less practical for very large dataset without dimensionality reduction techniques. For a small diabetes dataset, SVM has the ability to handle high-dimensional spaces and its reliance on a few support vectors makes it a strong candidate. With appropriate kernel selection and parameter tuning, SVM can achieve high accuracy even with limited data, making it ideal for smaller well-defined datasets.

3.6. Evaluation Metrics

To evaluate the performance of the models, I used the confusion matrix and the classification report as basic tools to calculate the performance of the models.

i. **Confusion Matrix:** This is a table that displays the performance of a classification model by comparing the predicted and the actual values (true) or actual class labels. It has four components. It visualizes the breakdown of correct and incorrect predictions for each class. This helps to identify where the model is performing or where it is making errors.

a. True Positives (TP): Correctly predicted positive instances.
b. True Negatives (TN): Correctly predicted negative instances
c. False Positives (FP): Incorrectly predicted positive instances (Type I error)
d. False Negative FN): Incorrectly predicted negative instances (Type II)

ii. **Classification Report:** This provides metrics for evaluating the performance of a classification model. The key metrics typically include:

a. **Accuracy:** This is the ratio of correctly predicted instances to the total instances.

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

b. **Precision:** This is the proportion of true positive predictions out of all positive predictions made by the model. It evaluates the correctness of positive prediction.

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

c. **Recall (Sensitivity):** This is the proportion of actual positives that the model correctly identified. It is the ratio of true positives to the sum of true positives and false negatives

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

d. **F1-Score:** This is the harmonic mean of precision and recall, used to balance the two metrics.

F1-Score = 2 x $\frac{Precision\ X\ Recall}{Precision+Recall}$

- **Support:** In a classification report, support is not a performance metric. It refers to the number of actual occurrences of each class in the dataset.
- **Macro Average:** This calculates the metric (e.g., precision, recall and F1-score) independently for each class and then takes the unweighted mean of these metrics.
- **Weighted Average:** This calculates the metric for each class and then takes the mean, weighted by the support (number of samples) for each class.

4.0 **Results and Analysis**

In this research, I employed multiple models, namely Random Forest, XGBoost and Support Vector Machine, to predict diabetes progression using two datasets, the BRFSS 2015 Diabetes dataset (large) and the Pima Indians Diabetes dataset (small). These models were evaluated using key performance metrics such as accuracy, precision, recall and F1-score for each class. The target class for BRFSS dataset (0 for non-diabetic, 1 for pre-diabetic and 2 for diabetic) and the Pima Indians dataset (0 for non-diabetic and 1 for diabetic). The metrics provide insight into the model's ability to correctly classify the target class for both datasets, which is critical for predicting diabetes progression effectively.

4.1. **Analysis of the Models using Confusion Matrix**

1. **Random Forest**

Fig 1a. Large dataset (BRFSS)

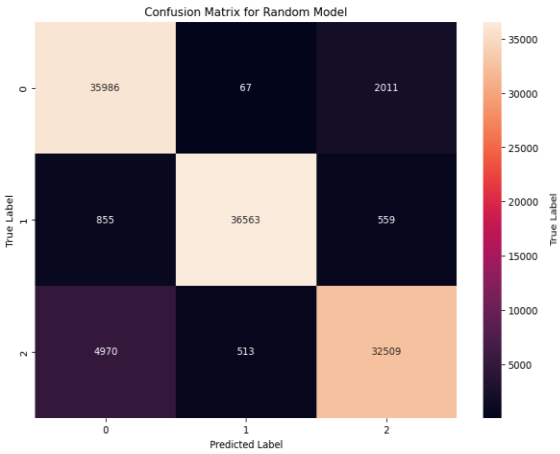
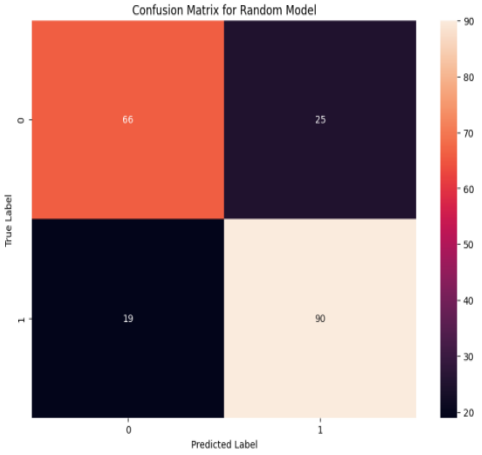


Fig 1b. Small dataset (Pima Indians)



The large dataset being a multi-class classification task, shows strong performance for majority classes (0 and 1), with high true positives (TP) and true negatives (TN). Class 1 achieves 36,563 correctly predicted instances. However, it struggles with the minority class (class 2), leading to significant false negatives FN), such as 4,970 instances of class 2 being misclassified as class 0. Similarly, false positives (FP) are prominent for class 2, where 2,011 instances of class 0 and 559 of class 1, are incorrectly predicted as class 2. These challenges highlight difficulties in handling class imbalance and feature overlap in a large-scale dataset.

In contrast, the small dataset being binary, achieves excellent performance with minimal FN and FP rates. Class 1 is well detected with 90 true positives, while FN (19 instances of class 1 misclassified as class 0) and FP (25 instances of class 0 misclassified as class 1) are low. True negatives (TN) are also high with 66 correctly identified for class 0. The model demonstrates robust accuracy and precision for this simpler classification task, benefiting from the small dataset size and balanced class distribution, unlike the large dataset.

2. XGBoost

Fig 2a. Large dataset

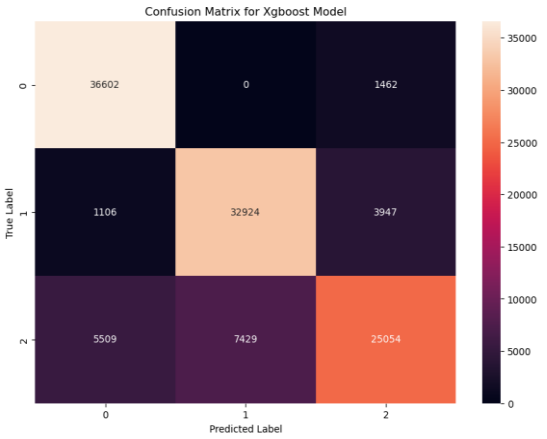
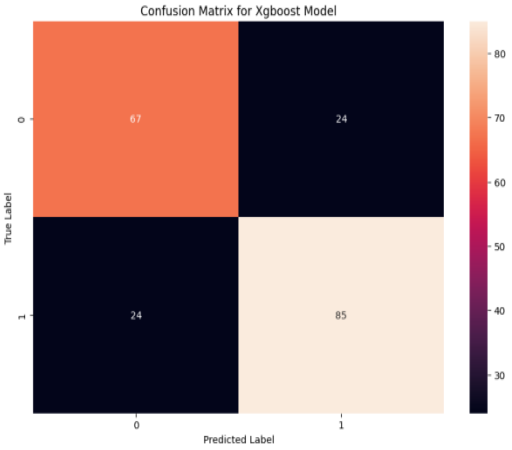


Fig 2b. Small Dataset

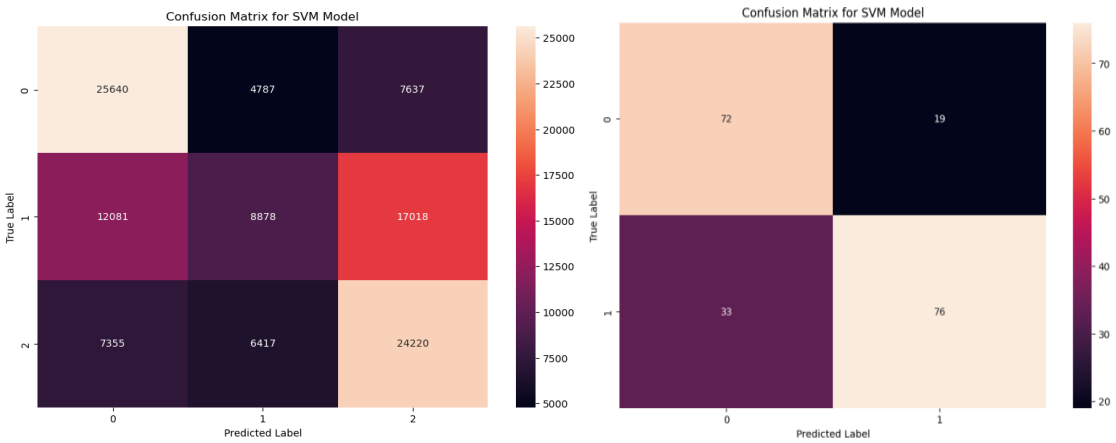


The first confusion matrix represents the performance of an XGBoost model on a large dataset with three classes. (0,1, 2). The model demonstrates good performance with majority of the predictions falling on the diagonal, indicating correct classification. For example, 36,602 instances of class 0, 32,924 of class 1 and 25054 of class 2 were correctly predicted. However, misclassifications are observed, particularly for class 2, which has significant confusion with other classes (7,429 misclassified as class 1). The scale of the dataset with prediction in the thousands highlights the model's capacity to handle complex dataset, but also emphasizes the need to address class specific misclassification issues.

The second confusion matrix is of a small dataset, involving two classes (0 and 1). The model achieves relatively balanced predictions, with 67 instances of class 0 and 85 of class 1 correctly classified. However, misclassifications are notable with 24 instances of each class being incorrectly predicted as the other. The small scale of the dataset means these misclassifications have a large proportional impact on performance evaluation. Overall, while the model performs well for its size, it reveals challenges such as limited data impacting classification reliability.

3. SVM

Fig 3a. Large dataset **Fig 3b. Small dataset**



The first confusion matrix measures the SVM model's performance on a large dataset with three classes (0, 1, 2). The model achieves reasonable accuracy for classes 0 and 2 with 25,640 and 24,220 correct predictions respectively. However, it struggles significantly with class 1, where 12,081 instances are misclassified as class 0 and 17,018 misclassified as class 2. These substantial misclassifications suggest challenges in separating class 1 from the others, possibly due to overlapping features or class imbalance. The large dataset size with predictions in the tens thousands, stresses the need for strategies like better feature selection, class balancing or optimization to improve performance.

The second matrix reflects model's performance on a small dataset, which achieves balanced predictions with 72 and 76 correct classifications for classes 0 and 1 respectively. It represents a notable proportion of misclassifications. 19 instances of class 0 are misclassified as class 1 and 33 of class 1 as class 0. These misclassifications have a large impact due to the dataset's small size, emphasizing the importance of careful tuning and potentially augmenting the data to improve model reliability and accuracy.

4.2 Classification Report Analysis

1. Random Forest

Table 1a. Large Dataset

Class	Precision	Recall	F1-Score	Support
0	0.86	0.95	0.90	38064
1	0.98	0.96	0.97	37977
2	0.93	0.86	0.89	37992
Accuracy			0.92	114033
Macro avg	0.92	0.92	0.92	114033
Weighted avg	0.92	0.92	0.92	114033

Table 1b. Small dataset

Class	Precision	Recall	F1-Score	Support
0	0.78	0.73	0.75	91
1	0.78	0.83	0.80	109
Accuracy			0.78	200
Macro avg	0.78	0.78	0.78	200
Weighted avg	0.78	0.78	0.78	200

The first classification report assesses the performance of a Random Forest model on a large dataset with three classes (0, 1, 2). The model achieves high overall accuracy of 92%, with macro and weighted averages for precision, recall and F1-score all at 0.92. Class 1 performs particularly well, with an F1-score of 0.97, reflecting excellent balance between precision and recall. However, Class 2 has a slight lower recall of 0.86, indicating some challenges in identifying all instances of this class. Overall, the results demonstrate the model's effectiveness in handling large datasets with strong class-level performance.

In contrast, the second table is for model performance on a small dataset with two classes (0 and 1). The overall performance is lower with an accuracy, macro average and weighted average of 0.78 for precision, recall and F1-score. Class 1 performs slightly better with an F1-score of 0.80 compared to 0.75 for class 0. The lower scores highlight the challenges of using a Random Forest model on small datasets, where limited data can restrict the model's ability to generalize effectively. This comparison underscores the model's scalability and the influence of dataset size on its predictive performance with stronger results observed in large datasets.

2. XGBoost

Table 2a. Large dataset

Class	Precision	Recall	F1-Score	Support
0	0.85	0.96	0.90	38064
1	0.82	0.87	0.84	37977
2	0.82	0.66	0.73	37992
Accuracy			0.83	114033
Macro avg	0.83	0.83	0.82	114033
Weighted avg	0.83	0.83	0.82	114033

Table 2b. Small dataset

Class	Precision	Recall	F1-Score	Support
0	0.74	0.74	0.74	91
1	0.78	0.78	0.78	109
Accuracy			0.76	200
Macro avg	0.76	0.76	0.76	200
Weighted avg	0.76	0.76	0.76	200

Table 2a illustrates performance of the XGBoost model on a large dataset with three classes (0, 1, 2). The overall accuracy is 83% with a macro and weighted average F1-score of 0.82, indicating strong but not perfect performance. Class 0 achieves the highest F1-score of 0.90, benefiting from a high recall of 0.96, while class 2 has the lowest F1-score of 0.73, due to a relatively low recall of 0.66. This suggests that the model struggles to correctly classify some instances of class 2, likely due to overlapping features or imbalanced data. These results show the model's capability to handle large datasets effectively, though improvements in class-specific recall especially for class 2, could enhance overall performance.

The second classification report summarises the model's performance on the small dataset with two classes (0 and 1). The overall accuracy, macro average and weighted average F1-score are lower at 76%, reflecting the limitations of working with a small dataset. Class 1 performs slightly better with an F1-score of 0.78 compared to 0.74 for class 0. The balanced but relatively modest precision and recall values for both classes suggest that the model's ability to generalize is constrained by the limited size of the dataset. These results illustrate how dataset size impacts the XGBoost model's performance, with better generalization in the large dataset compared to the small one.

3. SVM

Table 3a. Large dataset

Class	Precision	Recall	F1-Score	Support
0	0.57	0.67	0.62	38064
1	0.44	0.23	0.31	37977
2	0.50	0.64	0.56	37992
Accuracy			0.52	114033
Macro avg	0.50	0.51	0.49	114033
Weighted avg	0.50	0.52	0.49	114033

Table 3b. Small dataset

Class	Precision	Recall	F1-Score	Support
0	0.69	0.79	0.73	91
1	0.80	0.70	0.75	109
Accuracy			0.74	200
Macro avg	0.74	0.74	0.74	200
Weighted avg	0.75	0.74	0.74	200

Table 3a evaluates the performance of an SVM model on the large dataset, having overall accuracy of 52%, with macro and weighted averages for precision, recall and F-score around 0.50, indicating poor overall performance. Class 0 achieves the highest F1-score of 0.62, due to relatively better recall of 0.67. But classes 1 and 2 perform poorly, particularly class 1, which has a recall of just 0.23. These results suggest significant difficulty in distinguishing between the classes, likely due to the complexity of the dataset or limitations in the SVM model's ability to handle complex multi-class problems effectively.

The second report explains the model's performance on the small dataset, having overall accuracy of 74%, with macro and weighted averages for precision, recall and F1-score also at 0.74, showing a moderate improvement compared to the larger dataset. Class 1 performs slightly better, achieving an F1-score of 0.75 compared to 0.73 for class 0. These results express that while the SVM model struggles with larger and more complex datasets, it can perform adequately on small, simpler tasks with fewer classes. This comparison highlights the importance of dataset size and complexity in evaluating SVM performance and suggests the need for alternative approaches or model optimizations for large datasets.

4.3. Optimization of Models for both Datasets Using Hyperparameter Tuning

Model optimization refers to the process of improving model's performance by fun-tuning its parameters, hyperparameters and structure to better fit the underlying data. Hyperparameter tuning is a crucial step in enhancing the performance of models. The table below shows the results summary of the models' optimization using hyperparameter tuning.

Table 1. **Results Summary of Models' Optimization for the Two Datasets**

Models	Datasets	Stratified Sampling (Yes or No)	Accuracy (%)
Random Forest	Large	Yes (5% subset)	76.6
	Small	No	80.1
XGBoost	Large	Yes (5% subset)	76.3
	Small	No	80.5
SVM	Large	Yes (5% subset)	47.8
	Small	No	76.0

In this project, I performed optimization using hyperparameter tuning. Random Forest, XGBoost and SVM were evaluated on large and small diabetes datasets. On the large dataset, hyperparameter tuning could not execute on the full dataset until a 5% stratified subset, due to computational constraints. Random Forest achieved the highest accuracy of 76.6%, followed by XGBoost with 76.3%, while SVM struggled with an accuracy of 47.8%, reflecting its limitations with large-scale data.

On the small dataset, which did not require stratified sampling, XGBoost performed best with 80.5%, followed by Random Forest and SVM with 80.1% and 76.0% respectively. These results highlight XGBoost's efficiency in small datasets, while Random Forest demonstrated consistent reliability across dataset sizes. SVM performed well on small dataset but struggled significantly with the large one.

4.4. Model Performance Overview

The table below shows the results summary before and after mode's optimization for the two diabetes datasets.

Table 1. **Results Summary before and after Model's Optimization for the two Datasets**

Models	Datasets	Stratified Sampling (Yes or No)	Accuracy before Optimization (%)	Accuracy after Optimization (%)
Random Forest	Large	Yes (5% subset)	92.0	76.6
	Small	No	78.0	80.1
XGBoost	Large	Yes (5% subset)	83.0	76.3
	Small	No	76.0	80.5
SVM	Large	Yes (5% subset)	52.0	47.8
	Small	No	74.0	76.0

The performance summary illustrates that Random Forest and XGBoost consistently outperformed SVM. For the large dataset, a 5% stratified subset of the data was evaluated. After optimization, Random Forest and XGBoost's accuracies dropped from 92.0% to 76.6% and 83.0% to 76.3% respectively, likely due to challenges with tuning on a subset. SVM lagged significantly with accuracy declining from 52.0% to 47.8%, indicating its limitations in handling complex, non-linear relationships even after optimization.

For the small dataset, Random Forest improved from 78.0% to 80.1%, and XGBoost achieved the highest accuracy, increasing from 76.0% to 80.5% after optimization. SVM also improved moderately from 74.0% to 76.0%. The results emphasize that Random Forest and XGBoost are more robust across dataset sizes. Particularly effective in capturing intricate patterns of non-linear relationships, with XGBoost excelling after tuning on small dataset, while SVM remains more suitable for small tasks.

5.0 Analysis and Discussion

5.1. Discussion

What do the Results Mean: The results from this study provide comprehensive understanding into the performance of three Machine Learning models, such as Random Forest, XGBoost and SVM, on two datasets of varying size and complexity. Random Forest and XGBoost outperformed SVM across both datasets. On the large dataset, performance declined after optimization due to tuning on a 5% stratified subset, with Random Forest dropping from 92.0% to 76.6% and XGBoost from 83.0% to 76.3%. This reflects challenges in capturing the full dataset's complexity. SVM performed poorly with accuracy falling from 52.0% to 47.8%, showcasing its inability to handle complex, non-linear relationships in large tasks.

On the small dataset, Random Forest improved from 78.0% to 80.1%, and XGBoost achieved the highest accuracy, increasing from 76.0% to 80.5%. SVM also improved moderately from 74.0% to 76.0%, demonstrating better suitability for smaller, simpler tasks.

What Model Works the best and why: From the results, Random Forest emerged as the overall best-performing model due to its consistent performance across both datasets. Its ensemble approach, which combines multiple decision trees, allows it to reduce overfitting and improve generalization. Why Random Forest emerged best also include its robustness to noise, ability to handle both categorical and continuous data and resilience to class imbalance when paired with strategies like SMOTE. These characteristics makes it versatile and effective for diverse datasets.

5.2. Comparison with the Literature

The results align with prior research emphasizing the superior performance of the ensemble methods like Random Forest and XGBoost for diabetes prediction. Studies by Kasula (2023) and Rani (2020) reported high accuracies of 85% and 99% respectively, for Random Forest and Decision Trees, supporting this study's findings. However, the challenges with XGBoost's recall for the diabetic class and SVM's overall poor performance are consistent with findings from Kasturi (2024), which highlighted the difficulty of handling imbalanced datasets. The use of SMOTE helped mitigate these issues but did not completely resolve them. Compared to the literature, the results for SVM were lower, reflecting the model's sensitivity to dataset size and complexity.

5.3. Limitations of the Results

One major limitation is the persistent class imbalance, particularly with the large dataset. Despite the application of SMOTE, this imbalance negatively affected XGBoost's recall for the diabetic class. The second challenge is the tuning on a 5% subset for the large dataset, which negatively impacted the optimization process for Random Forest and XGBoost, leading to reduced accuracy. Thirdly, the small size of the Pima Indians Diabetes dataset also restricted the model's ability to generalize effectively. Moreover, the interpretability of Random Forest and XGBoost remains a

challenge as these models provide high accuracy but lack transparency compared to simpler models like Logistics Regression.

5.4. **Relation to Project Objectives and Answering the Research Question**

This investigation met its objectives by evaluating model performance and uncovering key predators like BMI, glucose levels, and age through EDA. Random Forest was identified as the most accurate and reliable model, with XGBoost performing well on small dataset. The Research question was addressed by showcasing Random Forest's superiority and providing insights into feature relationships with diabetes progression.

5.5. **Relation to Project Application and Practical Use of the Models**

Random Forest and XGBoost demonstrated strong practical applicability in healthcare for diabetes prediction. Random Forest high accuracy, robustness and ability to handle complex data make it ideal for early detection, risk identification and management strategies. XGBoost, with further tuning is also effective particularly for smaller datasets. Insights from EDA, such as BMI's correlation with diabetes progression, contribute actionable knowledge. However, SMV's poor performance makes it unsuitable for practical deployment in this context.

5.6. **Conclusion**

Summary of Key Results: Random Forest and XGBoost consistently outperformed SVM in predicting diabetes progression. On the large dataset, Random Forest and XGBoost accuracy decreased after optimization on a 5% subset of the data from 92.0% to 76.6% and 83.0% to 76.3% respectively. On the small dataset, Random Forest's accuracy increased from 78.0% to 80.1%, and XGBoost achieved the highest accuracy of 80.5%. SVM performed poorly on the large dataset (47.8%) but showed moderate improvement on the small dataset (76.0%)

Justifiable Conclusions: Random Forest proved to be the most reliable and effective model, with XGBoost excelling on small dataset. SVM struggled with complex and imbalanced data making it less suitable for this task.

Application and Real-World Use: Random Forest and XGBoost can be used in healthcare systems for early diabetes detection, risk assessment and tailored interventions. Insights from EDA can guide prevention programs and public health initiatives.

Future Work: Future research should focus on improving model interpretability, addressing class imbalance with advanced techniques and exploring Neural Networks or hybrid models for enhanced performance. Expanding datasets to include lifestyle and genetic factors can further refine predictions.

6.0. References

Elmenshawy, K., Wael, N., Ahmed, R. and El-Douh, A.A. (2024). Diabetes Prediction using Machine Learning and Explainable Artificial Intelligence Techniques. *SciNexuses*, 1, pp.28–43. doi:https://doi.org/10.61356/j.scin.2024.1306.

Gufran Ahmad Ansari, Salliah Shafi Bhat and Mohd Dilshad Ansari (2024). Machine Learning Techniques for Diabetes Mellitus Based on Lifestyle Predictors. *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, 17. doi:https://doi.org/10.2174/0123520965291435240508111712.

Gupta, N.P. (2024). Diabetes Prediction Using Machine Learning. *Deleted Journal*, 20(7s), pp.2244–2257. doi:https://doi.org/10.52783/jes.3960.

Hounguè, P. and Bigirimana, A.G. (2022) Leveraging pima dataset to diabetes prediction: Case study of deep neural network', *Journal of Computer and Communications*, 10(11), pp. 15–28. doi:10.4236/jcc.2022.1011002.

Jain, R., Nitin Kumar Tripathi, Pant, M., Chutiporn Anutariya and Chaklam Silpasuwanchai (2024). Investigating Gender and Age Variability in Diabetes Prediction: A Multi-Model Ensemble Learning Approach. IEEE Access, [online] pp.1-1. doi:https://doi:.org/10.1109/access.2024.3402350.

Khongorzul Dashdondov, Lee, S. and Munkh-Uchral Erdenebat (2024). Enhancing Diabetes Prediction and Prevention through Mahalanobis Distance and Machine Learning Integration. *Applied Sciences*, 14(17), pp.7480–7480. doi:https://doi.org/10.3390/app14177480.

Kasturi, K. (2024). Comparison of Machine Learning Models for Diabetes Prediction. *International Journal of Advanced Research in Science Communication and Technology*, pp.531–536. doi:https://doi.org/10.48175/ijarsct-19072.

Kasula, B.Y. (2023). Machine Learning Applications in Diabetic Healthcare: A Comprehensive Analysis and Predictive Modeling. *International Numeric Journal of Machine Learning and Robots,* [online] 7(7). Available at: https://injmr.com/index.php/fewfewf/article/view/19.

Mishra, H.V.K., Singh, A.P., Sharma, V. and Khanna, E. (2024). Exploring the Effectiveness of Various Machine Learning Algorithms in Detecting Alzheimer's Disease: A Comparative Analysis. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.4776479.

N Nagarjuna and Dr. Lakshmi HN (2024). Predictive Modeling of Diabetes Mellitus Utilizing Machine Learning Techniques. *CVR Journal of Science & Technology*, 26(1), pp.112–117. doi:https://doi.org/10.32377/cvrjst2618.

Rani, K.J. (2020). Diabetes Prediction Using Machine Learning. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 6(4), pp.294-305. doi:https://doi.org/10.32628/cseit206463.

Simeon Yuda Prasetyo and Zahra Nabila Izdihar (2024). Multi-layer Perceptron Approach for Diabetes Risk Prediction using BRFSS Data. pp.303–308. doi:https://doi.org/10.1109/icsima62563.2024.10675535.

Simeon Yuda Prasetyo, Zahra Nabila Izdihar and Ghinaa Zain Nabiilah (2024). Analyzing Machine Learning Approaches for Diabetes Risk Prediction: Comparative Performance Assessment Using BRFSS Data. pp.324–329. doi:https://doi.org/10.1109/icicos62600.2024.10636871.

Sivaranjani, S., Ananya, S., Aravinth, J. and Karthika, R. (2021). *Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICACCS51430.2021.9441935.

World Health Organization (2023). *DIABETES*. [online] World Health Organisation. Available at: https://www.who.int/news-room/fact-sheets/detail/diabetes.

7.0. Appendix

```
# -*- coding: utf-8 -*-
"""Diabetes1_Predictions_Opt_27_12_24.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1cis36u3ujGtuPx3-hOy-
fN6xBSn-x3dm

## Leveraging Machine Learning Approaches for Diabetes Prediction and
Analysis

This project aims to provide answers to the below research questions

i. Model Performance Comparison: How do Random Forest, XGBoost, and
Support Vector Machines (SVM) compare in their predictive performance
for diabetes progression?

ii. Exploratory Data Insights: What insights, trends, or patterns can
be uncovered through Exploratory Data Analysis (EDA) in relation to
diabetes progression?

## About the Dataset
The dataset consists of 22 variables

1.BMI: Continuous variable reflecting the individual's body mass index.

2.MentHlth: Number of days in the past 30 days when mental health was
not good.

3.PhysHlth: Number of days in the past 30 days when physical health was
not good.

4.Age: Age of the individual in years.

5.Education: Ordinal variable representing education level (1 = Never
attended school, increasing levels of education).

6.Income: Ordinal variable representing income levels (1 = Less than
$10,000, increasing levels of income).

7.GenHlth: Self-reported general health status (1 = Excellent, 2 = Very
good, 3 = Good, 4 = Fair, 5 = Poor).

8.NoDocbcCost: Frequency-based numerical indication of cost-related
healthcare inaccessibility.

9.Diabetes_binary: Binary variable (1 for Diabetes, 0 for No Diabetes)

10.HighBP: Binary variable indicating high blood pressure (1 for Yes, 0
for No)

11.HighChol: Binary variable indicating high cholesterol (1 for Yes, 0
for No)
```

```
12.CholCheck: Binary variable indicating cholesterol check in the past
five years (1 for Yes, 0 for No).

13.Smoker: Binary variable indicating smoking history (1 for Smoker, 0
for Non-Smoker).

14.Stroke: Binary variable indicating history of stroke (1 for Yes, 0
for No).

15.HeartDiseaseorAttack: Binary variable indicating heart disease or
myocardial infarction (1 for Yes, 0 for No).

16.PhysActivity: Binary variable indicating physical activity in the
past 30 days (1 for Yes, 0 for No).

17.Fruits: Binary variable indicating daily fruit consumption (1 for
Yes, 0 for No).

18.Veggies: Binary variable indicating daily vegetable consumption (1
for Yes, 0 for No).

19.HvyAlcoholConsump: Binary variable indicating heavy alcohol
consumption (1 for Yes, 0 for No).

20.AnyHealthcare: Binary variable indicating healthcare coverage (1 for
Yes, 0 for No).

21.DiffWalk: Binary variable indicating difficulty walking or climbing
stairs (1 for Yes, 0 for No).

22.Sex: Binary variable indicating biological sex (1 for Male, 0 for
Female).
"""

# Importing the required libraries for this project

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

!pip install scikit-learn-intelex
from sklearnex import patch_sklearn
patch_sklearn()

from sklearn.model_selection import train_test_split,GridSearchCV
from imblearn import over_sampling
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import RFE
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
!pip install xgboost
from xgboost import XGBClassifier
from sklearn.model_selection import KFold,
StratifiedKFold,cross_val_score
from sklearn.model_selection import RandomizedSearchCV
```

```
from sklearn.svm import SVC
import scipy.stats as stats

!pip install -U scikit-learn xgboost
import sklearn
import xgboost

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

print(sklearn.__version__)
print(xgboost.__version__)

# Reading in the dataset

db=
pd.read_csv(r"C:\Users\USER\Desktop\diabetes_012_health_indicators_BRFS
S2015.csv")

# Checking the top 5 rows of the data
db.head()

"""## Exploring and understanding the data

In the project code, I will begin by understanding the dataset

"""

# Understanding basic information about the data

db.info() # This data contains 253680 rows and there are all in
numerical formats,
          # which is as a result of the values been encoded

# Checking if the data contains duplicate values
db.duplicated().sum()

# We have 23,899 duplicate values in the data

# Checking if the dataset contains any missing values
db.isnull().sum()
# The dataset contains no missing value so i wont have to worry about
dealing with missing values

"""## Data Cleaning

I will move on to cleaning the data by removing the duplicate in the
dataset
"""

# Dropping the duplicates in the datasets

db.drop_duplicates(inplace=True)

"""## Exploratory Data Analysis (EDA)
```

```
After understanding and cleaning the data, i will move on to extra
insights from the data and understand trends and  patterns in relation
to diabetes progression
"""

# Insight on the distribution of the dependent variable

sns.countplot(x='Diabetes_012', data=db)

"""The dataset shows a significant class imbalance in the target
varaible, with most individuals classified as non-diabetic(0), a small
minority as prediabetic(1), and a moderate number as diabetic(2). This
imbalance which impact model performance will be be treated using the
SMOTE oversampling technique


"""

# Diabetes_012 vs HighBP
sns.countplot(x='Diabetes_012', hue='HighBP', data=db)
plt.title("Relationship Between Diabetes_012 and HighBP")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="HighBP (High Blood Pressure)", loc='upper right')
plt.show()

"""High blood pressure is prevalent across all groups but is more
common among diabetic (2.0) and prediabetic (1.0) individuals than
among non-diabetic (0.0) individuals, suggesting that hypertension is a
significant risk factor contributing to the development and progression
of diabetes"""

# Bivariate Analysis: Diabetes_012 vs HighChol
sns.countplot(x='Diabetes_012', hue='HighChol', data=db)
plt.title("Relationship Between Diabetes_012 and HighChol")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="HighChol (High Cholesterol)", loc='upper right')
plt.show()

"""High cholesterol is more common in diabetic (2.0) and prediabetic
(1.0) individuals than in non-diabetic (0.0) individuals.
This indicates that high cholesterol may be a contributing factor to or
a consequence of diabetes
"""

# Diabetes_012 vs CholCheck
sns.countplot(x='Diabetes_012', hue='CholCheck', data=db)
plt.title("Relationship Between Diabetes_012 and CholCheck")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="CholCheck (Cholesterol Check)", loc='upper right')
plt.show()

"""Cholesterol checks are highly prevalent across all groups, with the
majority of individuals, including non-diabetic (0.0), prediabetic
(1.0), and diabetic (2.0) individuals, undergoing regular monitoring
```

```
(CholCheck = 1). This suggests widespread awareness of the importance
of cholesterol monitoring"""

# Diabetes_012 vs BMI
sns.barplot(x='Diabetes_012', y='BMI', data=db)
plt.title("Relationship Between Diabetes_012 and BMI")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("BMI (Body Mass Index)")
plt.show()

"""The chart shows a positive correlation between BMI and diabetes
progression. Individuals with diabetes (category 2) have the highest
BMI, followed by those who are pre-diabetic (category 1), while non-
diabetics (category 0) have the lowest BMI. This highlights the
importance of weight management in preventing or mitigating the
progression of diabetes"""

#  Diabetes_012 vs Smoker
sns.countplot(x='Diabetes_012', hue='Smoker', data=db)
plt.title("Relationship Between Diabetes_012 and Smoking")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="Smoker", loc='upper right')
plt.show()

"""The chart reveals that in the non-diabetics group the non-smokers
slightly outnumber the  smokers. In the pre-diabetic and diabetic
groups, smokers and non-smokers are nearly equally represented,
suggesting that smoking might play a role in diabetes risk but is not
the sole factor"""

# Diabetes_012 vs Stroke
sns.countplot(x='Diabetes_012', hue='Stroke', data=db)
plt.title("Relationship Between Diabetes_012 and Stroke")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="Stroke", loc='upper right')
plt.show()

"""The chart shows that strokes are rare across all groups but are
slightly more common among diabetics (category 2) compared to non-
diabetics (category 0) and pre-diabetics (category 1). This suggests a
potential link between diabetes progression and increased stroke risk,
though strokes remain relatively uncommon overall."""

# Bivariate vs HeartDiseaseorAttack
sns.countplot(x='Diabetes_012', hue='HeartDiseaseorAttack', data=db)
plt.title("Relationship Between Diabetes_012 and Heart Disease or
Attack")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="Heart Disease or Attack", loc='upper right')
plt.show()

"""The chart shows that heart disease or attacks are relatively rare
across all groups but slightly more common in diabetics (category 2)
compared to non-diabetics (category 0) and pre-diabetics (category 1).
Most individuals, regardless of diabetes progression, do not have heart
```

```
disease, though the data suggests an increasing risk of heart issues as
diabetes progresses"""

# Diabetes_012 vs PhysActivity
sns.countplot(x='Diabetes_012', hue='PhysActivity', data=db)
plt.title("Relationship Between Diabetes_012 and Physical Activity")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="Physical Activity", loc='upper right')
plt.show()

"""The chart shows that most individuals, particularly non-diabetics
(category 0), engage in physical activity, as indicated by the large
count for active individuals (1.0). However, among diabetics (category
2), the proportion of physically active individuals decreases,
suggesting a potential link between lower physical activity levels and
diabetes progression"""

# Diabetes_012 vs Fruits
sns.countplot(x='Diabetes_012', hue='Fruits', data=db)
plt.title("Relationship Between Diabetes_012 and Fruits")
plt.show()

"""The chart illustrates the relationship between fruit and
consumption and diabetes progression. It shows that non-diabetics
(category 0) are more likely to consume fruits (1.0), while diabetics
(category 2) have a lower proportion of regular fruit  consumers. This
suggests that a diet rich in fruits  may be associated with a lower
risk of diabetes or its progression."""

# Diabetes_012 vs Veggies
sns.countplot(x='Diabetes_012', hue='Veggies', data=db)
plt.title("Relationship Between Diabetes_012 and Veggies")
plt.show()

"""The chart shows that individuals who consume vegetables regularly
(1.0) are predominantly non-diabetic (category 0), while diabetics
(category 2) have a noticeably lower proportion of vegetable
consumption. This suggests that regular vegetable intake may play a
role in reducing the risk of diabetes or slowing its progression,
highlighting the importance of a healthy diet in diabetes
management."""

# Diabetes_012 vs Heavy Alcohol Consumption
sns.countplot(x='Diabetes_012', hue='HvyAlcoholConsump', data=db)
plt.title("Relationship Between Diabetes_012 and Heavy Alcohol
Consumption")
plt.show()

"""The chart shows that heavy alcohol consumption (1.0) is relatively
rare across all diabetes categories, with the majority of individuals,
particularly non-diabetics (category 0), not engaging in heavy
drinking. Among diabetics (category 2), heavy alcohol consumption is
even less common. This suggests that heavy alcohol consumption is not a
prevalent factor among diabetics"""

# Diabetes_012 vs General Health
sns.barplot(x='Diabetes_012', y='GenHlth', data=db)
```

```
plt.title("Relationship Between Diabetes_012 and General Health")
plt.show()

"""The chart shows a positive relationship between diabetes progression
and general health ratings (GenHlth), with diabetics (category 2)
reporting the poorest general health compared to pre-diabetics
(category 1) and non-diabetics (category 0). This indicates that
diabetes is associated with a decline in overall health."""

# Diabetes_012 vs Mental Health
sns.barplot(x='Diabetes_012', y='MentHlth', data=db)
plt.title("Relationship Between Diabetes_012 and Mental Health")
plt.show()

"""The chart shows that individuals with diabetes (categories 1 and 2)
report worse mental health (MentHlth) compared to non-diabetics
(category 0). The trend suggests a potential connection between
diabetes progression and declining mental health, highlighting the
importance of mental health support for individuals managing
diabetes."""

# Diabetes_012 vs Physical Health
sns.barplot(x='Diabetes_012', y='PhysHlth', data=db)
plt.title("Relationship Between Diabetes_012 and Physical Health")
plt.show()

"""The chart illustrates that individuals with diabetes (categories 1
and 2) report progressively poorer physical health (PhysHlth) compared
to non-diabetics (category 0). Diabetics (category 2) experience the
worst physical health, highlighting a clear association between
diabetes progression and declining physical well-being."""

# Diabetes_012 vs Difficulty Walking
sns.countplot(x='Diabetes_012', hue='DiffWalk', data=db)
plt.title("Relationship Between Diabetes_012 and Difficulty Walking")
plt.show()

"""The chart indicates that individuals with diabetes (categories 1 and
2) are more likely to report difficulty walking (DiffWalk = 1) compared
to non-diabetics (category 0). This association strengthens as diabetes
progresses, with diabetics (category 2) showing a higher proportion of
difficulty walking"""

# Diabetes_012 vs  Age
sns.barplot(x='Diabetes_012', y='Age', data=db)
plt.show()

"""The chart shows that as diabetes progresses (from category 0 to 2),
the average age of individuals increases. Diabetics (category 2) are
older on average than non-diabetics (category 0) and pre-diabetics
(category 1). This suggests that diabetes is more prevalent among older
individuals, highlighting age as a potential risk factor for diabetes
progression."""

# Diabetes_012 vs Education
sns.barplot(x='Diabetes_012', y='Education', data=db)
plt.show()
```

```
"""The chart indicates that education levels are similar across all
diabetes categories (0, 1, and 2), with no significant variation
observed. This suggests that education level does not appear to have a
strong direct association with diabetes progression in this dataset."""

# Diabetes_012 vs Income
sns.barplot(x='Diabetes_012', y='Income', data=db)
plt.show()

"""The chart shows that income levels slightly decrease as diabetes
progresses from non-diabetics (category 0) to diabetics (category 2).
Individuals in the diabetic category have lower average income compared
to non-diabetics. This suggests a potential association between lower
income and higher diabetes prevalence."""

# Diabetes_012 vs NoDocbcCost
sns.countplot(x='Diabetes_012', hue='NoDocbcCost', data=db)
plt.title("Relationship Between Diabetes_012 and NoDocbcCost")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="NoDocbcCost", loc='upper right')
plt.show()

"""The chart shows that individuals in the diabetic categories (1 and
2) are slightly more likely to report "NoDocbcCost" (an inability to
see a doctor due to cost) compared to non-diabetics (category 0).
However, the overall prevalence of this issue remains low across all
groups. This suggests that financial barriers to healthcare might
slightly impact those with diabetes more,"""

# Diabetes_012 vs AnyHealthcare
sns.countplot(x='Diabetes_012', hue='AnyHealthcare', data=db)
plt.title("Relationship Between Diabetes_012 and AnyHealthcare")
plt.xlabel("Diabetes_012 (Target Variable)")
plt.ylabel("Count")
plt.legend(title="AnyHealthcare", loc='upper right')
plt.show()

"""The chart shows that the majority of individuals across all diabetes
categories (0, 1, and 2) report having access to healthcare
(AnyHealthcare = 1). However, diabetics (category 2) have a slightly
higher proportion of individuals without healthcare access
(AnyHealthcare = 0) compared to non-diabetics (category 0)."""

# Visualizing the correlation analysis to investigate relationship
between the variables
plt.figure(figsize=(15,10))
sns.heatmap(db.corr(), annot=True)
plt.title("Correlation Analysis")

"""The correlation analysis reveals that diabetes progression
(Diabetes_012) is strongly linked to several health conditions and
demographic factors. It is positively correlated with high blood
pressure (0.26) and high cholesterol (0.28), indicating that
individuals with diabetes are more likely to experience these
conditions. Age also shows a strong positive correlation (0.34) with
diabetes, suggesting that older individuals are more prone to the
disease. Conversely, negative correlations with income (-0.15) and
```

education (-0.11) point to a possible socioeconomic connection, where lower income and education levels may contribute to higher diabetes prevalence.

Health and lifestyle factors also play a significant role. Higher BMI is moderately associated with high blood pressure (0.19) and high cholesterol (0.18), emphasizing the impact of obesity on these conditions. Physical activity, on the other hand, negatively correlates with diabetes progression (-0.10), suggesting that an active lifestyle reduces diabetes risk. Furthermore, fruit and vegetable consumption is positively linked to physical activity (0.13 and 0.14), indicating that individuals with healthier lifestyles tend to have better diets.

Mental and physical health indicators reveal critical insights into overall well-being. General health is strongly correlated with physical health (0.52), and difficulty walking (DiffWalk) has strong positive correlations with both general health (0.45) and physical health (0.47). These findings highlight how poor physical and general health are linked to mobility issues. Socioeconomic factors also show a moderate positive relationship between income and education (0.42), while both are weakly negatively correlated with diabetes, high blood pressure, and high cholesterol, underscoring their protective influence on health outcomes.

Preparing and Preprocessing the Data
"""

```
# Splitting  the data into target and independent variable
X=db.drop(['Diabetes_012'], axis=1)
y= db['Diabetes_012']

# Ensuring the data is balanced using smote technique so there is no
bias in the model

# Initializing the sampling object
sm=over_sampling.SMOTE()

# Applying the sampling to the dataset
X,y=sm.fit_resample(X,y)

# Checking the data after applying the SMOTE

sns.countplot(x=y, data=db)

# Checking the total number of rows  after applying the SMOTE
X.shape

# Train-test split using the 80:20 ratio
x_train, x_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=101)

# Normalizing the features by scaling them to be within the same scale
scaler=StandardScaler()

x_train_scaled=scaler.fit_transform(x_train)
x_test_scaled=scaler.transform(x_test)

"""### Model Building
```

```
Next step, I will now be employing the 3 algorithims to build the
models as well as evaluating the performance of the model built

#### Building the Random Forest Model
"""

# Objectifying the Random Forest Model
random=RandomForestClassifier(random_state=42)

# Fitting the Random Forest Model  to learn  the training data
random.fit(x_train_scaled,y_train)

# How well did the Random model learn the data
train_random_accuracy =random.score(x_train_scaled,y_train)

# Making prediction on the test data
random_prediction = random.predict(x_test_scaled)

train_random_accuracy

"""#### Evaluating the performnace of the Random Forest model  on test
data"""

# Evaluating the Random model prediction on test data
test_random_accuracy=metrics.accuracy_score(y_test, random_prediction)

classification_random_report = metrics.classification_report(y_test,
random_prediction)

random_confusion_matrix = metrics.confusion_matrix(y_test,
random_prediction)

test_random_accuracy

print(classification_random_report)

# Visualizing  the confusion matrix
plt.figure(figsize=(10,7))
sns.heatmap(random_confusion_matrix, annot=True, fmt="d")
plt.title('Confusion Matrix for Random Model')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

"""#### Building the XgBoost Model"""

# Objectifying the Xgboost Model

xg=XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42)

# Fitting the Xgboost Model  to learn  the training data
xg.fit(x_train_scaled,y_train)

# How well did the Xgboost Model learn the data
train_xg_accuracy = xg.score(x_train_scaled,y_train)
```

```
# Making prediction on the test data
xg_prediction = xg.predict(x_test_scaled)

print(train_xg_accuracy)

"""#### Evaluating the performnace of the Xgboost model  on test
data"""

# Evaluating the Xgboost model prediction on test data
test_xg_accuracy=metrics.accuracy_score(y_test, xg_prediction)

classification_xg_report = metrics.classification_report(y_test,
xg_prediction)

xg_confusion_matrix = metrics.confusion_matrix(y_test, xg_prediction)

print(test_xg_accuracy)

print(classification_xg_report)

# Visualizing  the confusion matrix
plt.figure(figsize=(10,7))
sns.heatmap(xg_confusion_matrix, annot=True, fmt="d")
plt.title('Confusion Matrix for Xgboost Model')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

"""Building the Support Vector Model"""

# Objectifying the Support Vector Model
support=LinearSVC(random_state=42)

# Fitting the Model  to learn  the training data
support.fit(x_train_scaled,y_train)

# How well did the model learn the data
train_support_accuracy =support.score(x_train_scaled,y_train)

# Making prediction on the test data
support_prediction =support.predict(x_test_scaled)

print(train_support_accuracy)

"""#### Evaluating the performnace of the Support model  on test data

"""

# Evaluating the model prediction on test data
test_support_accuracy=metrics.accuracy_score(y_test,
support_prediction)

classification_support_report = metrics.classification_report(y_test,
support_prediction)

support_confusion_matrix = metrics.confusion_matrix(y_test,
support_prediction)
```

```
print(test_support_accuracy)

print(classification_support_report)

# Visualizing  the confusion matrix
plt.figure(figsize=(10,7))
sns.heatmap(support_confusion_matrix, annot=True, fmt="d")
plt.title('Confusion Matrix for SVM Model')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

"""# **Optimizing the Model Using Hyperparameter Tuning**

To improve the performance of my models in order to maximize their
accuracy, I will be applying grid search hyperparameter tuning to our
models.

This will be applied on a subset of the data to reduce computational
load,
"""

# Performing Stratified Sampling to retain 5% of the entire dataset
X_sample, _, y_sample, _ = train_test_split(
    X, y,
    test_size=0.95,  # Retain only 5% of the data
    stratify=y,  # Ensure class proportions are preserved
    random_state=101)

# Train-test split within the 5% sample
x_train, x_test, y_train, y_test = train_test_split(
    X_sample, y_sample,
    test_size=0.2,
    stratify=y_sample,
    random_state=101)

print(f"5% Sample Size: {X_sample.shape}, Train Size: {x_train.shape},
Test Size: {x_test.shape}")

"""Random Forest Hyperparameter Tuning"""

rf_param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [10, 20, 30],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2],
    'bootstrap': [True, False],
    'criterion': ['gini', 'entropy'],
    'max_features': ['sqrt', 'log2', None]
}

# GridSearchCV for Random Forest
rf = RandomForestClassifier(random_state=101)
grid_search_rf = GridSearchCV(
    estimator=rf,
    param_grid=rf_param_grid,
    cv=3,
    scoring='accuracy',
```

```
    verbose=2,
    n_jobs=-1
)

grid_search_rf.fit(x_train, y_train)
best_rf = grid_search_rf.best_estimator_

# Evaluate Random Forest
y_pred_rf = best_rf.predict(x_test)
print("Random Forest Best Parameters:", grid_search_rf.best_params_)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred_rf))

"""# **XgBoost Hyperparameter**"""

# Define hyperparameter grid for XGBoost
param_grid_xgb = {
    'learning_rate': [0.01, 0.1],
    'max_depth': [3, 6],
    'n_estimators': [100, 200],
    'subsample': [0.8, 1.0],
    'colsample_bytree': [0.8, 1.0]
}

# GridSearchCV for XGBoost
xgb = XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=101)
grid_search_xgb = GridSearchCV(
    estimator=xgb,
    param_grid=param_grid_xgb,
    cv=3,
    scoring='accuracy',
    verbose=2,
    n_jobs=-1
)

grid_search_xgb.fit(x_train, y_train)
best_xgb = grid_search_xgb.best_estimator_

# Evaluate XGBoost
y_pred_xgb = best_xgb.predict(x_test)
print("XGBoost Best Parameters:", grid_search_xgb.best_params_)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred_xgb))

"""# **Supoort Vector Hyperparameter**"""

#Support Vector Mode
svm_param_grid = {
    'C': [0.1, 1, 10],
    'loss': ['hinge', 'squared_hinge'],
    'max_iter': [1000, 2000,5000, 10000]
}

# GridSearchCV
grid_search_lsvc = GridSearchCV(
    estimator=support,
    param_grid=svm_param_grid,
    cv=3,  # 3-fold cross-validation
    scoring='accuracy',
```

```
    verbose=2,
    n_jobs=-1
)

# Fit the GridSearchCV on training data
grid_search_lsvc.fit(x_train, y_train)

# Get the best estimator and parameters
best_lsvc = grid_search_lsvc.best_estimator_
print("Best Parameters for LinearSVC:", grid_search_lsvc.best_params_)

# Evaluate the best model
y_pred_lsvc = best_lsvc.predict(x_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred_lsvc))
```