National College of Ireland

Optimizing Diabetes Predictive Modeling: A Study of Data Balancing and Advanced Algorithms

MSc Research Project
Data Analytics

Piyush Ingle

Student ID: 22154779

School of Computing

National College of Ireland

Supervisor: Furqan Rustam

National College of

Ireland MSc Project

Submission Sheet

School of Computing

Student Name: .....Piyush Ingle.....

Student ID: .....x22154779.....

Programme: .....Data Analytics..... **Year:** .....2023.....

Module: .....Research Project.....

Supervisor: .....Furqan Rustam.....

Submission Due Date: .....14/12/2023.....

Project Title: Optimizing Diabetes Predictive Modeling: A Study of Data Balancing and Advanced Algorithms

Word Count: .....8335..... **Page Count**.....24.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....**Piyush Ingle**.....

Date: .....14/12/2023.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Diabetes Predictive Modeling: A Study of Data Balancing and Advanced Algorithms

Piyush Ingle

22154779

Abstract

Diabetes is now seen as a chronic illness that poses a global problem since it may affect everyone. Diabetes Mellitus is another name for the disorder that interferes with how our bodies process blood sugar levels. The goal of this study is to apply two data balancing techniques – ADASYN (Adaptive synthetic sampling) and SMOTE (Synthetic Minority Over-sampling) to improve the accuracy Diabetes Prediction Models. In addition to addressing the inherent class imbalance in diabetes datasets, the study looks at how these strategies affect the prediction abilities of five conventional Machine learning algorithms, k-Nearest Neighbors (KNN), AdaBoost, Decision Tree, Logistic Regression, and Gaussian Naïve Bayes. Furthermore, the study digs into the field of deep learning through the utilization of three sophisticated algorithms: Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN). This work attempts to identify the synergies between several machine learning and deep learning algorithms and data balancing strategies for efficient diabetes prediction through a thorough comparison analysis. The findings offer insightful information about how to maximize model performance in healthcare applications and give a detailed grasp of how various predictive modelling techniques interact with data preparation techniques when it comes to diabetes diagnosis. After comparison of all the results we found that SMOTE has given better results with Decision tree as the best performer with accuracy of 82% and F1 score 0.80.

Keywords: Diabetes Prediction, ADASYN, SMOTE, Machine Learning, Deep Learning

1. Introduction

Diabetes is a serious global health issue, with poor and economically developed countries particularly affected. Diabetes is quite common, yet there is currently no proven treatment or cure for it. This condition is indicated by elevated blood sugar levels that last for quite a while. Insufficient insulin release from the pancreas and insufficient insulin action in the body are two significant factors. Insulin is the "key" that opens the metabolic fuel door enabling cells to use glucose as an energy source. Diabetes can have temporary symptoms like dehydration and unconsciousness, but it can also have long-term complications including stroke, blindness, heart attack, foot ulcers, renal failure, etc (Chaturvedi et al., 2023).

Additionally, diabetes raises the risk of developing certain cancers, including endometrial, breast, and colon cancer. Nevertheless, people with diabetes can live long and healthy lives if they receive the proper management and treatment, which usually involves checking blood sugar levels, adopting a healthy lifestyle, and taking medications or insulin as needed. As of 2019, 463 million people worldwide were estimated to have diabetes; this number is expected to rise to 578 million by 2030 and 720 million by 2045. Consequently, the number of diabetic patients is predicted to increase exponentially by 25% in 2030 and 51% in 2045(Patro *et al.*, 2023) . Diabetes has increased in frequency as a chronic condition in recent years. As a result, early detection and diagnosis of diabetes may be challenging. Thus, a precise and practical method for anticipating the development of diabetes is needed. Diabetes is becoming more common now a days, and this can be attributed to a variety of

factors such as poor nutrition and physical inactivity(Krishna et al., 2023). Early diabetes diagnosis is now carried out manually by a medical professional using their knowledge, experience, and observation of the disease. Much data is now collected by the healthcare business, but unlike genetic data, this data may not always reveal inherited hidden patterns. Because some aspects may be missed, which might have a significant impact on the observations and outcomes, these manual assessments are, therefore, very misleading and negative, particularly when it comes to an early diagnosis. It is challenging to forecast with precision when diabetes may manifest. Diabetes cannot be permanently cured, although it may be controlled and treated if a proper diagnosis is obtained early in the course of the illness. Additionally, a timely diagnosis of diabetes helps lessen the chance of complications and major health issues. However, in order to increase accuracy, advanced early and automated diagnostic techniques are desperately needed(Patro et al., 2023).

People can use an automated method for sharing knowledge and problem-spotting to learn how to prevent diabetes and treat it effectively. Furthermore, a great deal of data on the medical histories of people with diabetes is generated, creative methods may be used to gather this vital information for diabetes prediction (Krishna et al., 2023). This study explores the field of diabetes prediction modeling by utilizing the strength of ML and DL algorithms to examine large and varied datasets. The main goal is to create reliable and accurate prediction models that can identify people who are at chance for getting diabetes, allowing for early intervention and individualized treatment plans. The suggested models seek to go beyond conventional diagnostic techniques by utilizing the multitude of data included in genetic information, lifestyle patterns, clinical indicators, and electronic health records. This will provide an active and knowledge-driven approach for diabetes diagnosis(Malini *et al.,* 2021). The increasing global incidence of diabetes and its related consequences highlight the need for efficient prediction models to be developed. Conventional risk analysis techniques frequently fail to capture the complex interrelationships between many factors impacting the start, development, and severity of diabetes(Shrivastava *et al.,* 2022). With their capacity to identify intricate patterns in massive, multidimensional datasets, machine learning (ML) and deep learning (DL) approaches provide a singular chance to decipher the intricate underlying mechanisms of diabetes etiology. The goal of this investigation is to aid in the creation of prediction models that are reliable, readable, and scalable so they can be easily incorporated into clinical procedures(Louisa et al., 2023).

1.1 Research Question:

How data balancing techniques such as SMOTE and ADASYN, improve the performance of a specific set of machine learning and Deep learning algorithms and the comparison of both these data balancing techniques using the results obtained from the algorithms applied?

1.2 Research Objective:

- **Performance Evaluation**: Strict evaluation protocols will be used to gauge how well the developed diabetes prediction models work. Important assessment parameters, such as accuracy, precision, and F1-score, will be highlighted to guarantee an in-depth understanding of the effectiveness of the model.
- **Integration of Clinical Relevance**: The goal of the thesis is to close the gap that exists between prediction accuracy and clinical relevance. The study intends to closely fit with the practical demands of healthcare professionals through thorough analysis and validation methodologies.
- **Analysis of Models**: A crucial part of the study is a careful examination of the models that have been created. This entails analysing how they make decisions, recognizing their advantages and disadvantages, and learning more about their prognostic tendencies. These evaluations will help improve models for real-world implementation.
- **Global Impact**: The main goal is to significantly reduce the prevalence of diabetes globally. The project aims to contribute to a paradigm change in diabetes treatment by applying state-of-the-art computational approaches and making use of copious amounts of publicly available data, therefore opening up new avenues for proactive and successful healthcare interventions.

- **Proactive Medical processes**: By utilizing machine learning (ML) and deep learning (DL) for diabetes prediction, the study's ultimate objective is to change medical processes from reactive to proactive. By empowering medical personnel in early identification and management, this proactive strategy hopes to support the worldwide initiative to lessen the impact of diabetes.

2. Literature Review:

This section features a few researchers that have retrieved publicly available medical data to collect data using machine learning and deep learning techniques and have made some excellent contributions to the field of diabetes-related prediction.

For instance, (Refat *et al.,* 2021) developed a computer-monitored diabetes diagnostic system for that dataset using deep learning and machine learning techniques. Several machine learning models, including Random Forest, XGBoost, Decision Tree, Logistic Regression, K-Nearest Neighbors, and Support Vector Machine, were employed in this experiment. Additionally, they have employed a few Deep Learning foundational techniques, including Artificial Neural Networks, Multilayer Perceptrons, and Long Short-Term Memory. We evaluated the diabetes dataset with each of the aforementioned classification techniques. XGBoost was the top performance in their testing, achieving 100% accuracy, and it was outperforming other deep learning and machine learning methods. Similarly, three distinct machine learning classification techniques were employed in separate research conducted by (Krishna et al., 2023). Cross-validation and hyperparameter tuning were utilized to obtain the best outcomes for the selected data set. As previously indicated, Support Vector Machine and Decision Tree machine learning methods were applied. Several metrics, like the F1 score, recall, precision, and accuracy, are used to evaluate the effectiveness of ML algorithms. Three algorithms were used: SVM, DT, and the suggested classification algorithm. When assessed to each of the two basic classifiers, the suggested classifier provided the best degree of precision out of all three methods. Given that the two studies' findings differ, we are eager to do more research to determine which is most useful in identifying diabetes in its early stages.

This paper (Okikiola et. al., 2023) developed ontology-based diabetes prediction techniques with Naive Bayes classifiers and decision trees. A model provided in the study served as the basis for this. It is believed that doing this will increase the likelihood that doctors can correctly diagnose diabetes. Using certain input keywords, a Bayesian classifier will categorize test data. A list of user-submitted phrases that represent the signs and symptoms of diabetes is provided by the study's query module algorithm(Rani *et al.,* 2023). Finding documents that connect each user-supplied query term to the relevant keyword category from the domain ontology is the objective of the technique's assessment of the testing documents. When a term in the user's query matches an item in the domain ontology testing document, the computer downloads the document and applies the Naive Bayes classification technique to automatically offer the response. To determine whether the experimental results of the developed diabetes prediction approach are appropriate for diabetes categorization, they will be assessed and tested. Following that, these suggested algorithms will be tested, investigated, and contrasted with the algorithms that are now in use. In a work by (Vijaya, j. et al. 2023), they combined many machine-learning algorithms, including Random Forest, linear regression, Extra tree, KNN, linear SVC, Gaussian NB, SVC, and Decision Tree, with seven different optimization techniques, including PSO, GA, ACO, Cuckoo, Whale, FireFly, and MayFly. They evaluated the performance of optimization algorithms with these traditional machine learning methods in this study, and the findings indicated that optimization techniques outperformed the basic classifier.

In a different research, (Eben *et al.,* 2023) employed a variety of machine learning methods to analyze and categorize the data; they discovered that the AdaBoost and logistic regression classifiers worked well. The best of them was the logistic regression, which yielded an accuracy of 99.80%. It performed well, yielding a 98.50% accuracy. Similar to this, machine learning classifiers were employed by (Gupta *et al.,* 2023) to assess the accuracy of several models. Several classifiers, including K-Nearest Neighbors, Naive Bayes, Logistic Regression, Decision Trees, Random Forests, and Support Vector

Machines, have been used in this experiment to compare and analyze their accuracy. The outcome demonstrates that the Random Forest, with an accuracy of 81.5%, outperformed the others. Despite the aforementioned study, we are still unable to identify a single method that will work best for that particular dataset's early diabetes prediction. In order to determine which has the most to give us, we should do further research.

In a study conducted by (Sivaranjani *et al.,* 2021), the researchers experimented with several algorithms both before and after preprocessing the data. They discovered that Random Forest had the greatest accuracy of 74.44% without preprocessing and 81.4% with preprocessing. Speaking of Random Forest, (Krishna et al., 2023) conducted a research akin to ours in which they used exclusively Random Forest users to categorize diabetes. Additionally, they have placed greater emphasis on data preparation and have looked for findings both before and after data preprocessing.

A novel kind of investigation is conducted by (Rani et al., 2023), in which two tests are conducted to predict diabetes. A balanced dataset was used for the second experiment, whereas an unbalanced dataset was used for the first. While the various classifiers performed differently in the two experiments, Random Forest outperformed the others with an accuracy of 82.70% when the dataset was balanced. In a research that suggested a method for classification and group learning, the classifiers SVM, KNN, Random Forest, Decision Tree, Logistic Regression, and Gradient Boosting are employed (Malini et al., 2021). Furthermore, Random Forest demonstrated the highest accuracy of 78% among these classifiers.

A research has been carried out by (Louisa, O.O., et al., 2023) for developing nations. The way of life of individuals has changed as the nation has developed, and as was previously said, lifestyle plays a significant role in the development of this illness. The fact that diabetes prediction is still an emerging topic in underdeveloped nations has been emphasized more. Even though they focused more on research to enhance the models' performance and increase their accessibility to healthcare professionals in underdeveloped nations, they nevertheless employed SVM and KNN in their analysis.

The paper by (Gurunathan *et al.,* 2023) is the last one we looked at. It suggests developing a machine learning-based web application-based diabetes prediction system by contrasting the KNN with the Random Forest Classifier. Using the Indian Pila Dataset, the study was conducted. The machine learning method K-Nearest Neighbors (KNN) is surpassed by the Random Forest (RF) classifier with Bagging Meta-Estimator. It may be applied by the medical community to precisely identify a range of medical facts. As proposed, an online application for diabetes prediction was created using ML algorithms. In order to accurately anticipate diabetic situations, this suggested effort will concentrate on building a dataset based on location from real data using a deep learning model soon.

2.1 Summary of Literature Review:

The assessment of the literature includes a variety of research that use deep learning (DL) and machine learning (ML) methods to predict diabetes. Numerous models have been investigated, such as Random Forest, XGBoost, Decision Tree, Logistic Regression, and Support Vector Machines. XGBoost has continuously shown good accuracy. Furthermore, in order to improve the efficiency of conventional machine learning algorithms, researchers have combined optimization approaches. AdaBoost, Random Forest, and logistic regression classifiers have demonstrated effectiveness in diabetes prediction with respect to high accuracy rates. Even with these developments, further study is still required to determine the best strategies for early diabetes prediction, taking into account variables like optimization strategies and dataset features. The studies collectively emphasize the potential of ML and DL in transforming diabetes diagnosis from reactive to proactive, contributing to global efforts in early detection and management.

Table 1. Summary of related Work

Sr. No	Author/Year	Approach	Accuracy	Type/ Method
1	Chaturvedi et al. 2023	An Innovative Approach of Early Diabetes Prediction using Combined Approach of DC based Bidirectional GRU and CNN	97.8%	GRU, CNN
2	Patro et al. 2023	An effective correlation-based data modeling framework for automatic diabetes prediction using machine and deep learning techniques	96.13%	CNN
3	Krishna et al. 2023	An Efficient Machine Learning Classification Model for Diabetes Prediction	SVM – 87% DT- 80%	SVM, DT
4	Malini et al. 2021	Diabetic Patient Prediction using Machine Learning Algorithm	LR – 78%	SVM, KNN, LR
5	Shrivastava et al.2022	Early Diabetes Prediction using Random Forest	81%	RF
6	Louisa Osiyi et al. 2023	Prediction of Diabetes Mellitus in Developing Countries: A Systematic Review	99%	SVM, KNN, LR
7	Refat et al. 2021	A Comparative Analysis of Early-Stage Diabetes Prediction using Machine Learning and Deep Learning Approach	XGBOOST-100%	XGBoost, KNN, CNN, LSTM
8	Okikiola et al. 2023	An Ontology-Based Diabetes Prediction Algorithm Using Naïve Bayes Classifier and Decision Tree	-	NB, DT
9	Rani et al. 2023	Diabetes Prediction Using Machine Learning Classification Algorithms	RF – 82.7%	SVM, RF, EGB, DT
10	Vijaya et al. 2023	Diabetes Disease Prediction Using Various Metaheuristic Optimization Algorithms	Whale – 95.05%, RF – 75%	PSO, GA, ACO, Cuckoo, Whale, FireFly, MayFly, RF, Linear regression, GNB
11	Gupta et al. 2023	Diabetes Prediction using different Machine Learning Classifiers	RF – 81%	DT, LR, KNN, SVM, NB, RF
12	Sivaranjani et al. 2021	Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction	RF – 83%	RF, SVM
13	Gurunathan et al. 2023	Web Application-based Diabetes Prediction using Machine Learning	RF – 83.11%	RF, KNN
14	Goel et al. 2013	Evaluation of Sampling Methods for Learning from Imbalanced Data	-	SMOTE, ADASYN
15	Davide et al. 2009	K–Fold Cross Validation for Error Rate Estimate in Support Vector Machine	-	k-fold, SVM

16	Eben et al. 2023	Diabetes Prediction Model for Better Clarification by using Machine Learning	AdaBoost – 99.80%	LR, LDA, ETC, DT, SVC
17	Yahyaoui et al. 2019	A Decision Support System for Diabetes Prediction Using Machine Learning and Deep Learning Techniques	RF- 83%	SVM, RF, CNN
18	Chawla et al. 2002	SMOTE: Synthetic Minority Over-sampling Technique	-	SMOTE
19	Bunkhumpornpat et al. 2009	Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling Technique for Handling the Class Imbalanced Problem	-	SMOTE
20	He et al. 2008	ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning	-	ADASYN

3. Methodology

This section involves various steps such as cleaning, preprocessing, and organizing relevant dataset for analysis. Furthermore, model training employs advanced algorithms, and leveraging computational power to enhance predictive accuracy.

3.1 Summary of dataset:

The CDC gathers information on health-related telephone surveys called the Behavioral Risk Factor Surveillance System (BRFSS) once a year. Over 400,000 Americans participate in the annual survey, which gathers information on health-related risk behaviors, chronic illnesses, and the use of preventative services. Since 1984, it has been held annually. For this experiment, a CSV file of the 2015 Kaggle dataset was utilized. This original dataset comprises 330 characteristics and answers from 441,455 people. These characteristics are variables that are computed based on replies from specific participants, or they are questions that are posed to participants directly.

3.2 Data Exploration:

The diabetes which we are working on consists of 253680 rows and 22 columns. All the columns contain integer values and hence we are able to see stats of all the columns. In the below dataset, diabetes_012 is the target column which I have renamed later to Diabetes only.

Once we got the data type, we have also tried to pull up unique values in all the columns.

```
df.dtypes
```

```
Diabetes                int64
HighBP                  int64
HighChol                int64
CholCheck               int64
BMI                     int64
Smoker                  int64
Stroke                  int64
HeartDiseaseorAttack    int64
PhysActivity            int64
Fruits                  int64
Veggies                 int64
HvyAlcoholConsump       int64
AnyHealthcare           int64
NoDocbcCost             int64
GenHlth                 int64
MentHlth                int64
PhysHlth                int64
DiffWalk                int64
Sex                     int64
Age                     int64
Education               int64
Income                  int64
dtype: object
```

Fig 1. Datatype of columns.

```
Diabetes: [0 1]
HighBP: [0 1]
HighChol: [0 1]
CholCheck: [0 1]
BMI: [12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
 84 85 86 87 88 89 90 91 92 95 98]
Smoker: [0 1]
Stroke: [0 1]
HeartDiseaseorAttack: [0 1]
PhysActivity: [0 1]
Fruits: [0 1]
Veggies: [0 1]
HvyAlcoholConsump: [0 1]
AnyHealthcare: [0 1]
NoDocbcCost: [0 1]
GenHlth: [1 2 3 4 5]
MentHlth: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30]
PhysHlth: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30]
DiffWalk: [0 1]
Sex: [0 1]
Age: [ 1  2  3  4  5  6  7  8  9 10 11 12 13]
Education: [1 2 3 4 5 6]
Income: [1 2 3 4 5 6 7 8]
```

Fig 2. Unique values in each column

Description of Dataset:

This section offers a variety of details for each column that contains float or integer data. Another name for this is descriptive statistics. The number of values in each row, the mean value for each column, the standard deviation, the min, max, and the quartile data—that is, 25, 50, and 70% of the data in each column—are all displayed in this table.

3.3 Data Preparation:

Data preparation is simply cleaning the data so that it can be used to train a model. It involves a number of stages, which we shall see all of those are carried out on all of the datasets, including eliminating null values if they exist and eliminating undesirable data. Fortunately, when we tried to find null values there were none. All the values in the dataset are integer and float and thus we didn't have to do any encoding.

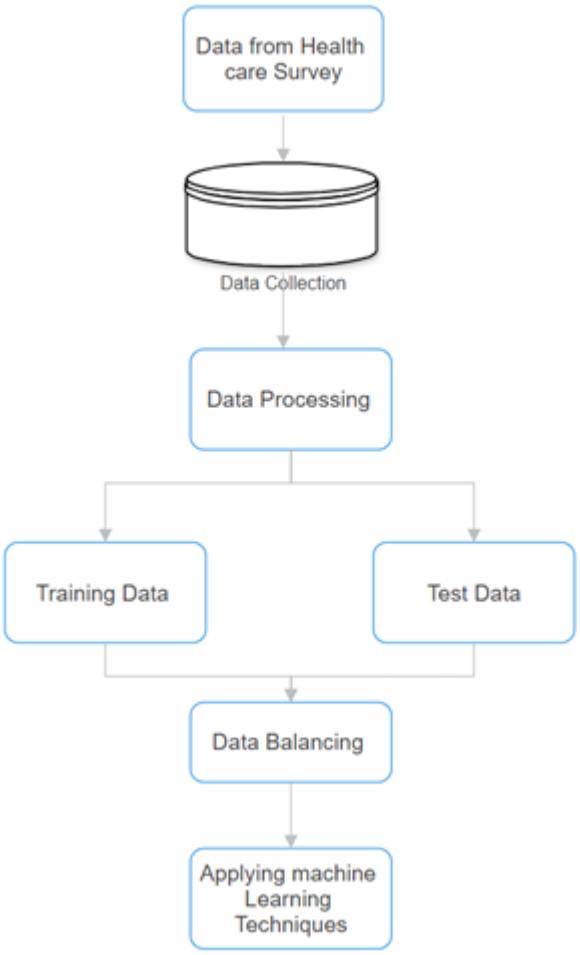

Fig 3. Basic flow of the experiment performed.

3.4 Exploratory data analysis:

Looking at the dataset we tried to find out the relation between different columns of the dataset. For finding the relations and some significance between the columns we have used various histograms, bar plots, scatter plots pie charts, correlation matrix etc. Below are some of the important features that we have tried to explore.

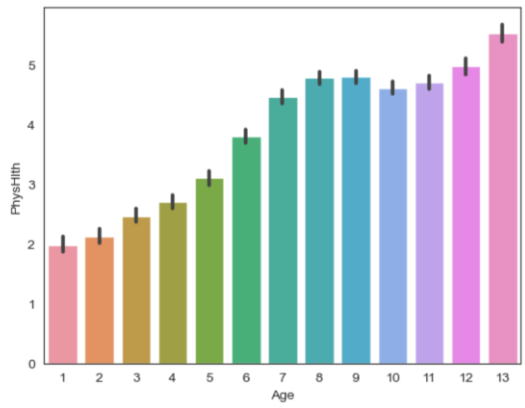

Fig 4. Age Vs Physical Health

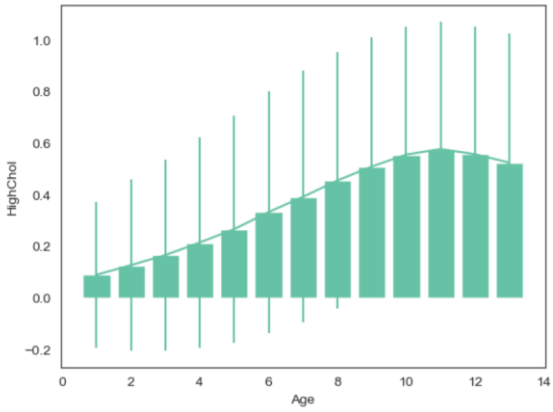

Fig 5. Age Vs High Cholesterol

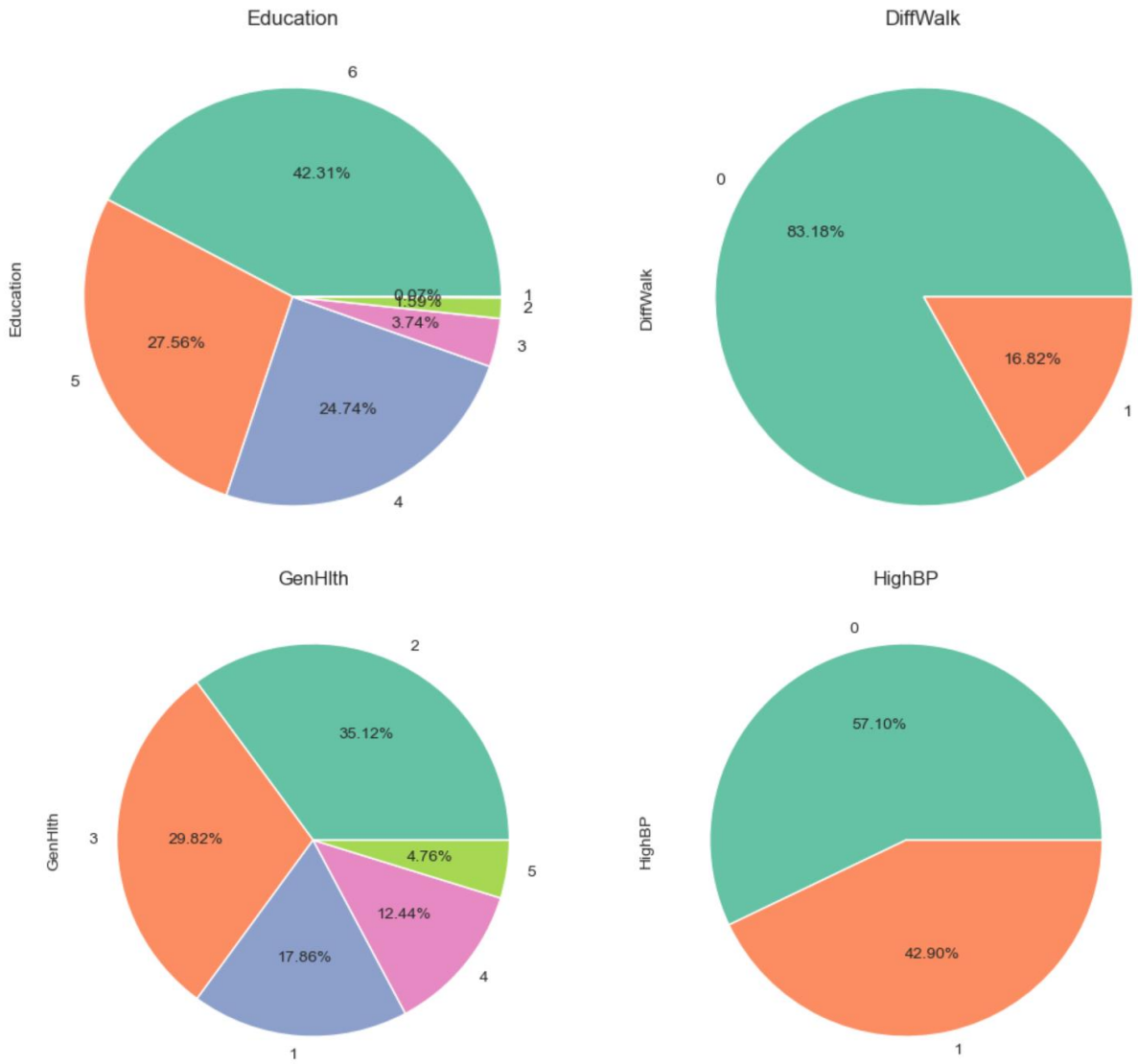

Fig 6. Variables affecting the most.

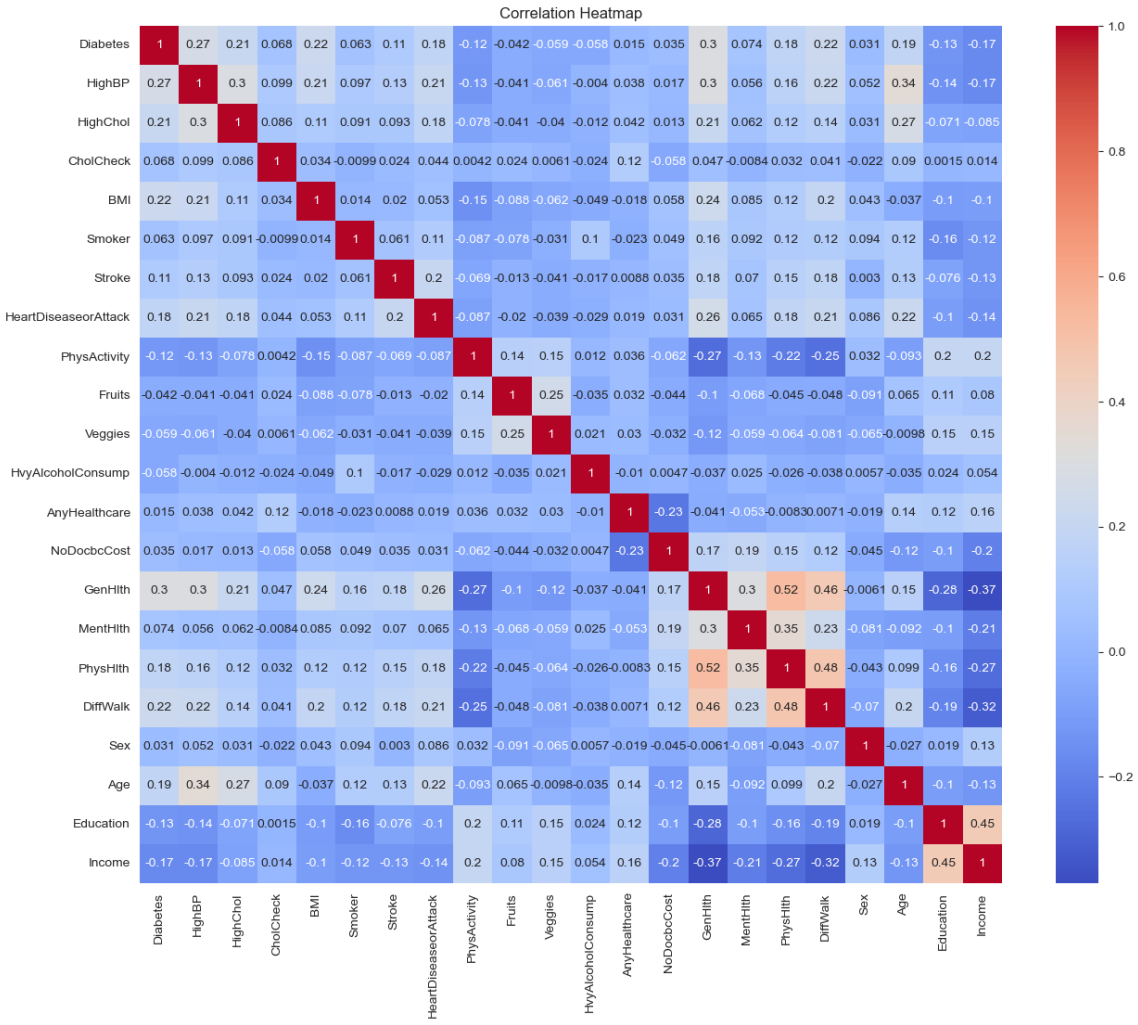

Fig 7. Correlation heat Map

3.5 Feature Selection:

I've just utilized OLS from statsmodels.api in this part to assist us discover the p_value for each column. If the value is less than or equal to 0.05, we've taken that variable into consideration. In this dataset the columns that are taken into consideration are const, HighBP, HighChol, CholCheck, BMI, Smoker, Stroke, HeartDiseaseorAttack, PhysActivity, 'HvyAlcoholConsump', AnyHealthcare, GenHlth, MentHlth, DiffWalk, Sex, Age, Education and Income.

3.6 Data Scaling:

The scaling of features of the dataset is done using the 'Standardscalar' from the Scikit Learn library of python. This is a very useful preprocessing step in machine learning workflows, especially when working with algorithms that are sensitive to the scale of features.

- **The goal of scaling:** In order to avoid some features from predominating over others during model training, equalize feature scales makes sure that all features have comparable scales.
- **Algorithm Sensitivity**: The magnitude of input characteristics affects the performance of some machine learning algorithms, including neural networks, k-nearest neighbors, and support vector machines. Scaling improves the efficiency of these algorithms.

3.7 Data Balancing:

Data balancing is needed in machine learning when there is a significant imbalance in the distribution of classes or outcomes in the training dataset. Class imbalance occurs when one class has a

disproportionately larger or smaller number of instances compared to the other classes. As we have already explored that the columns associated to value 0 in target column are 84%, columns associated to value 1 is 1.82% and for columns associated to 2 are 13.93% which shows a huge imbalance in the date. This imbalance can have several implications for the performance and behavior of machine learning models.

To see which balancing method serves the best we have used two methods those are ‘Smote (Synthetic minority oversampling technique)' and ADASYN (Adaptive synthetic Sampling). The purpose of applying two sampling methods is that we wanted to check the difference between the performance of models that are applied after sampling(Goel *et al.*, 2013).

Cross Validation: K- fold cross validation is a resampling technique commonly used in machine learning to access the performance and generalization ability of a predictive model. It involved partitioning the original dataset into K equal sized folds(subsets) and then performing training and evaluation K times, each time using a different fold as the test set and the remaining folds as the training set. The process is repeated K times, ensuring that each fold is used as the test set exactly once (Davide et al., 2009).

3.8 Models/Techniques

In this experiment we have used five machine learning models and two Deep learning models.

The machine learning models are as follows:

Decision Tree: A decision tree is a popular machine learning algorithm used for both classification and regression tasks. It's a versatile and intuitive model that mimics the human decision-making process. Decision trees are constructed by recursively partitioning the data into subsets based on the values of input features, ultimately leading to a tree-like structure where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents the final predicted outcome (Yahyaoui et al. 2019, and Eben et al., 2023).

Logistic Regression: Logistic Regression is a statistical and machine learning algorithm used for binary classification tasks, where the goal is to predict the probability of an instance belonging to a particular class. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm (Louisa, O. et al., 2023).

Adaboost: AdaBoost, short for Adaptive Boosting, is an ensemble learning technique used in machine learning for improving the performance of weak classifiers and creating a strong classifier. The key idea behind AdaBoost is to give more weight to the misclassified instances in the training set so that subsequent weak learners focus more on those instances during their training. The final prediction is then made by combining the predictions of all weak learners, with each learner's contribution weighted based on its accuracy (Vijaya, et al., 2023).

Gaussian Naive Bayes: It is a variant of the Naive Bayes algorithm. Naive Bayes algorithms are a family of probabilistic classifiers based on Bayes' theorem, and they are particularly well-suited for classification tasks.In the case of Gaussian Naive Bayes, the algorithm assumes that the features (input variables) follow a Gaussian (normal) distribution. This means that the likelihood of the features given the class labels is modeled as a Gaussian distribution. It's important to note that the "naive" in Naive Bayes refers to the assumption of independence between features, meaning that the presence or absence of one feature does not affect the presence or absence of another feature (Vijaya, J. et al, 2023).

K-Nearest Neighbors classifier: K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for both classification and regression tasks. It is a simple and intuitive algorithm that makes predictions based on the majority class (for classification) or the average value (for regression) of the K nearest neighbors in the feature space (Gurunathan et al., 2023).

The deep learning Algorithms we used are as follows:

Recurrent Neural Network: In general terms, a Recurrent Neural Network (RNN) is a type of neural network architecture designed for processing sequential data. Unlike traditional feedforward neural networks, which process input data in a single pass, RNNs have connections that form directed cycles, allowing them to maintain a hidden state that captures information from previous time steps.

Long short term Memory: Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to overcome some of the limitations of traditional RNNs in capturing and learning long-term dependencies in sequential data. LSTMs were introduced to address the vanishing gradient problem, which often hinders the ability of standard RNNs to effectively learn from and remember information over long sequences (Refat et al., 2021).

Gated Recurrent Unit: A Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture designed for processing and learning from sequential data. Similar to Long Short-Term Memory (LSTM) networks, GRUs were introduced to address certain challenges faced by traditional RNNs in capturing long-term dependencies in sequential data (Chaturvedi et al., 2023).

4. Design Specification

We have designed the problem of diabetes detection using different sampling techniques and for each sampling technique how the machine learning and deep learning algorithms will perform. Handling imbalanced datasets is a crucial aspect of machine learning and predictive modeling. When the distribution of classes in a dataset is uneven, with one class significantly outnumbering the others, it introduces challenges that can impact the performance and reliability of machine learning models. The significance of handling imbalanced datasets and the motivation for using different sampling techniques can be outlined as follows:

4.1 Motivation for using different sampling techniques:

1) Overcoming class imbalance:
    - Sampling techniques, such as oversampling the minority class or under sampling the majority class, aim to balance the class distribution.
    - This helps the model to learn from a more representative set of examples for each class.
2) Enhancing Model Performance:
    - Imbalance-aware sampling techniques can improve the performance of a model, especially when dealing with rare events or minority classes.
    - By exposing the model to more instances of the minority class, it can better discern patterns and make more accurate predictions.
3) Preventing Overfitting:
    - Imbalanced datasets may lead to overfitting on the majority class, where the model memorizes patterns specific to that class without learning to generalize.
    - Sampling techniques mitigate overfitting by providing a more balanced training set.

Synthetic Minority Oversampling Technique: SMOTE, or Synthetic Minority Over-sampling Technique, is a popular oversampling technique used in machine learning to address the challenge of imbalanced datasets. Imbalanced datasets occur when one class is significantly underrepresented compared to others, leading to biased model performance. SMOTE specifically focuses on the minority class, aiming to balance class distribution by generating synthetic instances(Chawla *et al.,* 2002).

Synthetic Instance Generation:

- The core idea behind SMOTE is to create synthetic instances for the minority class by interpolating between existing instances.
- For each instance in the minority class, SMOTE selects k nearest neighbors and generates synthetic instances along the line segments connecting the instance to its neighbors.

- SMOTE introduces synthetic examples by taking a random selection of features from the minority class instance and its neighbors.
- The synthetic instances are created by combining the features of the selected instance and its neighbors in a controlled manner(Bunkhumpornpat, et al., 2009).

Adaptive Synthetic Sampling (ADASYN): ADASYN is an oversampling technique designed to tackle imbalanced datasets. It adapts its synthetic sampling strategy based on the density distribution of the minority class. It focuses on generating synthetic instances for challenging and under-represented regions within the minority class, addressing the limitations of traditional oversampling methods(He, H. et al,.2008).

- Adaptive Sampling:
  - ADASYN adapts its synthetic sampling approach based on the local density of instances in the minority class.
  - It assigns higher importance to instances in regions with lower density, ensuring a more balanced representation.

4.2 Machine Learning Techniques:

Decision Tree:

Suitability for Diabetes Classification:

- Decision Trees are interpretable and can capture non-linear relationships in data, which can be valuable in understanding factors contributing to diabetes.
- They handle both numerical and categorical data, making them suitable for diverse datasets often encountered in medical research.

Logistic Regression:

Suitability for Diabetes Classification:

- Logistic Regression is simple, interpretable, and well-suited for binary classification tasks.
- It provides probability estimates, allowing for a clear interpretation of the likelihood of diabetes.

Adaboost:

Suitability for Diabetes Classification:

- Adaboost excels when dealing with imbalanced datasets, as it focuses on improving the classification of misclassified instances.
- It can handle complex relationships in the data and adapt to nuances in the diabetes classification task.

Gaussian Naive Bayes (GaussianNB):

Suitability for Diabetes Classification:

- GaussianNB is suitable for continuous features and assumes normal distribution, making it applicable to certain medical datasets.
- It is computationally efficient and can handle a moderate number of features.

K-Nearest Neighbors (KNN):

Suitability for Diabetes Classification:

- KNN is effective when instances of similar classes are close in the feature space.
- It adapts well to local patterns, making it suitable for diabetes classification where instances with similar characteristics may exhibit shared patterns.

Table 2. Algorithms and Hyperparameters

Algorithms	Hyperparameters
Decision Tree (DT)	Here I have used the default parameters. Criterion = 'gini', splitter = 'best', max_depth = None, min_sample_split = 2, min_samples_leaf = 1. (Mayer *et al.,* 2022)
Logistic Regression (LR)	Penalty = 'l2', c=1.0, fit_intercept = True, solver = 'lbfgs'
AdaBoost (Ad)	N_estimators = 100, random_state =0
Gaussian Naive Bayes (GNB)	- No hyperparameters to be tuned explicitly in the standard Gaussian Naive Bayes implementation
k-Nearest Neighbors (KNN)	N_neighbors =5, weights = 'uniform', algorithm = 'auto', leaf_size = 30, p =2

4.3 Deep Learning Techniques:

Recurrent Neural Network (RNN):

Architecture of RNN:

- By progressively scanning the data from left to right and uploading the hidden state at each time step, the RNN accepts an input vector (X) and output is vector (Y).
- All time steps have the same set of parameters.
- This states that the network use same set of parameters denoted by U, V and W.
- W stands for weight connected to the connection between hidden layers, V for the connection from hidden layer H to output layer Y. U for the weight parameter controlling the connection from output layer x to the hidden layer h.

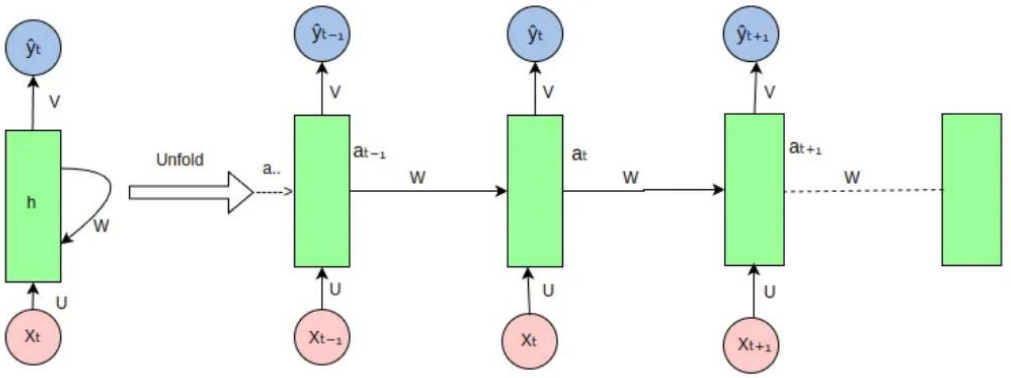

Fig 8. Architecture of RNN

Long Short-Term Memory (LSTM):

Architecture:

- Forget Gate: When an LTM enters this mode, useless data is forgotten.
- Learn Gate: STM and event (current input) are coupled so that the current input can use the essential knowledge that we have recently acquired via STM.

- Remember Gate: This serves as an updated LTM by combining STM and Event data with LTM information that we haven't forgotten.
- Utilize Gate: This gate functions as an updated STM by predicting the output of the current event using LTM, STM, and Event.

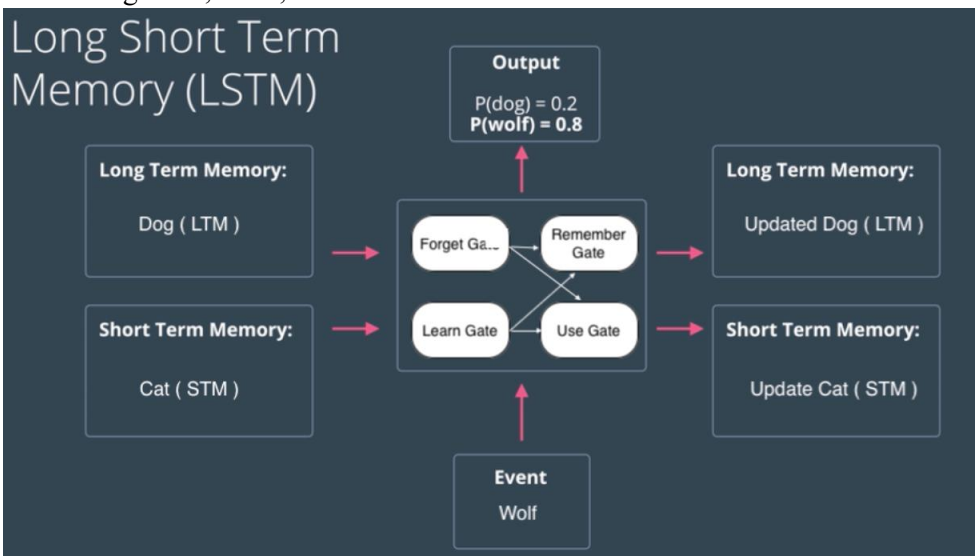

Fig 9. Architecture of LSTM

Gated Recurrent Unit (GRU):

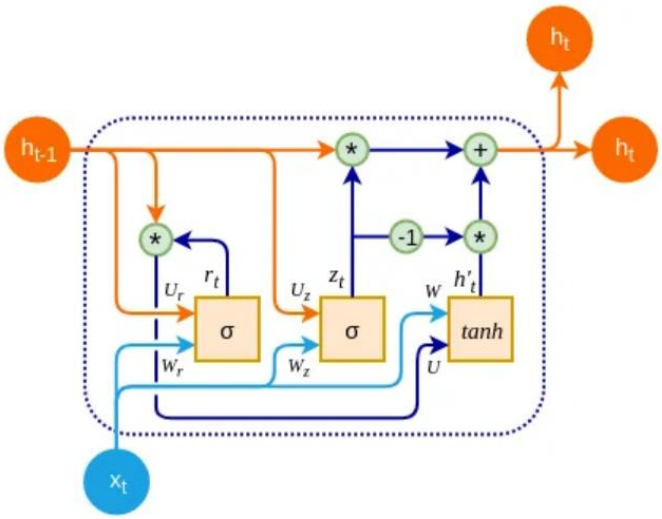

Fig 10. Architecture of GRU

- Input layer: The input layer supplies the GRU with sequential data, such as a word sequence or a time series of numbers.
- Hidden layer: The recurrent computation takes place in the hidden layer. Based on the prior hidden state and the current input, the hidden state is updated at each time step. The network's "memory" of the prior inputs is represented by a vector of integers called the hidden state.
- Reset gate: This gate decides how much of the previously concealed state should be forgotten. It generates a vector of values between 0 and 1 that determines the extent to which the previous hidden state is "reset" at the current time step, given the inputs of the previous hidden state and the current input.
- Update gate: The update gate decides how much of the new hidden state will include the candidate activation vector. It generates a vector of integers between 0 and 1 that determines

the extent to which the candidate activation vector is integrated into the new hidden state, given the inputs of the previous hidden state and the current input.

- Candidate activation vector: The candidate activation vector combines the current input with a modified version of the prior hidden state that has been "reset" by the reset gate. It is calculated by squashing the output between -1 and 1 using the tanh activation function.
- Output layer: The output layer generates the network's output using the final hidden state as an input. Depending on the job at hand, this might be a single number, a series of numbers, or a probability distribution over classes.

4.4 K- Fold Cross Validation

In the context of diabetes prediction, we tried to assess the performance of all the machine learning classifiers that we have implemented using 10-fold cross-validation. The resulting scores provide an estimate of the model's generalization performance across different subsets of the dataset. This approach helps mitigate the risk of overfitting or underfitting that might occur with a single train-test split. The printed scores give you an idea of the classifier's consistency and performance across various folds(Anguita *et al.,* no date).

In summary, cross-validation is a crucial tool in the development and evaluation of diabetes detection models. It ensures that the chosen model is not only accurate on the specific training and test split but also has a consistent and reliable performance across different partitions of the dataset, contributing to the model's overall effectiveness and generalization capabilities.

5. Implementation

5.1 Data Preparation:

While exploring the dataset, tried to find null values, but there are no NaN values in the dataset.

The dataset contains three classes in the target column and those are 0,1 and 2. When we further explored, we found the columns that are related to 0 are 84.24 %, 1 has 1.82% and 2 has 13.93% which demonstrate a case of huge class imbalance. In order to rectify this issue, we have applied two different sampling techniques and those are SMOTE and ADASYN which we will discuss and the result in the sampling section.

While looking for the unique values in the dataset, few columns have a wide range of data and thus there was need of data scaling. Here we have used StandardScalar for the same.

scaler = StandardScaler() creates an instance of the StandardScaler, a preprocessing technique used to standardize (or scale) features by removing the mean and scaling to unit variance.

x_train = scaler.fit_transform(x_train) fits the scaler to the training data (x_train) and transforms it. This ensures that the training set is standardized, with each feature having a mean of 0 and a standard deviation of 1. In the similar fashion we have scaled test data as well.

5.2 Feature Selection:

In this section we have used OLS from statsmodels library in python.

- ‘**lr = sm.OLS(y, X).fit()** fits an ordinary least squares (OLS) linear regression model to the data, where y is the dependent variable, and X is the feature matrix with the added constant term.
- **p_values = lr.pvalues** extracts the p-values associated with each coefficient in the regression model.

- **vars = p_values[p_values <= 0.05].index.tolist()** selects variables (features) with p-values less than or equal to 0.05, indicating statistical significance. These variables are then stored in the vars list.

5.3 Sampling Techniques:

Importing SMOTE:

- **from imblearn.over_sampling import SMOTE** imports the SMOTE class from the imbalanced-learn library, which provides tools for handling imbalanced datasets.

SMOTE Initialization:

- **smote = SMOTE(sampling_strategy='auto', random_state=42)** creates an instance of the SMOTE class.

Applying SMOTE:

- **X, y = smote.fit_resample(X, y)** applies the SMOTE technique to the feature matrix X and the target variable y.
- **fit_resample**: Fits the SMOTE model on the original data and generates synthetic samples to balance the class distribution.

Before sampling the shape of train and test data was (177576, 17) (76104, 17), however after sampling the shape is (448776, 17) (192333, 17)

Importing ADASYN:

- **from imblearn.over_sampling import ADASYN** imports the ADASYN class from the imbalanced-learn library, which provides tools for handling imbalanced datasets.

ADASYN Initialization:

- **adasyn = ADASYN(sampling_strategy='auto', random_state=42)** creates an instance of the ADASYN class.

Applying ADASYN:

- **X, y = adasyn.fit_resample(X, y)** applies the ADASYN technique to the feature matrix X and the target variable y.
- **fit_resample**: Fits the ADASYN model on the original data and generates synthetic samples to balance the class distribution.

5.4 Machine learning techniques implementation:

Importing Libraries:

- **from sklearn.tree import DecisionTreeClassifie**r: Imports the DecisionTreeClassifier from **scikit-learn**.
- **from sklearn.metrics import accuracy_score, classification_report, confusion_matrix:** Imports metrics for evaluating classification models.

Instantiate Decision Tree Classifier:

- **dt = DecisionTreeClassifier()**: Creates an instance of the Decision Tree classifier.

Training and Prediction:

- **dtPre = dt.fit(x_train, y_train).predict(x_test)**: Fits the Decision Tree model on the training set (x_train, y_train) and makes predictions on the test set (x_test).

This demonstrates how we have implemented Decision tree. In the similar manner we have implemented **Adaboost, GaussianNB, KNN, and Logistic Regression**.

In case of Logistic regression, we have passed some parameters which are as follows.

LR = LogisticRegression(solver='sag', C=3.0, multi_class='multinomial'): Creates an instance of the Logistic Regression model with the following parameters:

- solver='sag': Specifies the optimization algorithm to use. 'sag' stands for Stochastic Average Gradient, a variant of the gradient descent optimization algorithm.
- C=3.0: The inverse of regularization strength. Smaller values of C indicate stronger regularization.
- multi_class='multinomial': Indicates that the logistic regression model should be used for a multi-class classification problem, and the 'multinomial' option specifies that the cross-entropy loss should be used.

5.5 Deep learning Technique Implementation:

Data Preparation for RNN:

- X_train_reshaped and X_test_reshaped reshape the input data to a 3D array, which is necessary for inputting sequential data into the SimpleRNN layer.
- y_train_categorical and y_test_categorical perform one-hot encoding on the target labels for categorical crossentropy loss.

Model Definition:

- **model = Sequential()** initializes a sequential model.
- model.add(SimpleRNN(units=50, activation='relu', input_shape=(X_train_reshaped.shape[1], 1))) adds a SimpleRNN layer with 50 units, ReLU activation, and input shape defined by the reshaped data.
- **model.add(Dense(units=3, activation='softmax'))** adds a Dense layer with softmax activation for multi-class classification (assuming 3 classes).

Model Compilation:

- **model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])** compiles the model with the Adam optimizer, categorical crossentropy loss, and accuracy as the metric.

Model Training:

- We have trained the model using the training data for 100 epochs with a batch size of 128.

Implementation of LSTM and GRU is similar to RNN and all the algorithms are imported through tensorflow.keras library.

6. Evaluation

6.1 Results Without Data Balancing

Table 3. Results Without Data Balancing

Model name	Accuracy	Class	Precision	Recall	F1- score
Decision Tree	0.98	0	0.87	0.86	0.87
		1	0.03	0.03	0.03
		2	0.30	0.32	0.31
		MacroAvg	0.40	0.40	0.40
Logistic regression	0.84	0	0.86	0.98	0.92
		1	0.00	0.00	0.00
		2	0.55	0.18	0.27
		MacroAvg	0.47	0.38	0.39
AdaBoost	0.84	0	0.86	0.98	0.92
		1	0.00	0.00	0.00
		2	0.57	0.20	0.30
		MacroAvg	0.48	0.39	0.41
Gaussian NB	0.76	0	0.91	0.81	0.86
		1	0.03	0.01	0.01
		2	0.34	0.58	0.43
		MacroAvg	0.42	0.47	0.43
KNN	0.831	0	0.86	0.95	0.91
		1	0.08	0.00	0.01
		2	0.42	0.22	0.29
		MacroAvg	0.46	0.39	0.40
RNN	0.84	0	0.86	0.97	0.91
		1	0.00	0.00	0.00
		2	0.55	0.20	0.29
		MacroAvg	0.47	0.39	0.40
LSTM	0.840	0	0.86	0.97	0.91
		1	0.00	0.00	0.00
		2	0.55	0.20	0.29
		MacroAvg	0.47	0.39	0.40
GRU	0.848	0	0.86	0.97	0.91
		1	0.00	0.00	0.00
		2	0.55	0.20	0.29
		MacroAvg	0.47	0.39	0.40

6.2 Results using SMOTE

Table 4. Results using SMOTE

Model name	Accuracy	Class	Precision	Recall	F1- score
Decision Tree	0.82	0	0.81	0.70	0.75
		1	0.89	0.97	0.93
		2	0.77	0.81	0.79
		MacroAvg	0.82	0.83	0.82
Logistic regression	0.55	0	0.61	0.66	0.64
		1	0.52	0.48	0.50
		2	0.52	0.53	0.53
		MacroAvg	0.55	0.56	0.55
AdaBoost	0.56	0	0.64	0.65	0.64
		1	0.53	0.50	0.51
		2	0.53	0.54	0.53
		MacroAvg	0.56	0.57	0.56
Gaussian NB	0.49	0	0.71	0.40	0.51
		1	0.42	0.73	0.53
		2	0.52	0.36	0.43
		MacroAvg	0.55	0.50	0.49
KNN	0.76	0	0.79	0.62	0.69
		1	0.78	0.96	0.86
		2	0.74	0.72	0.73
		MacroAvg	0.77	0.77	0.76
RNN	0.60	0	0.69	0.62	0.65

		1	0.59	0.61	0.60
		2	0.56	0.60	0.58
		MacroAvg	061	0.61	0.61
LSTM	0.63	0	0.69	0.63	0.66
		1	0.61	0.69	0.65
		2	0.60	0.57	0.59
		MacroAvg	0.63	0.63	0.63
GRU	0.63	0	0.70	0.63	0.66
		1	0.61	0.71	0.65
		2	0.60	0.57	0.59
		MacroAvg	0.64	0.63	0.63

6.3 Results using ADASYN

Table 5. Results using ADASYN

Model name	**Accuracy**	**Class**	**Precision**	**Recall**	**F1- score**
Decision Tree	0.81	0	0.80	0.69	0.74
		1	0.89	0.97	0.93
		2	0.76	0.80	0.78
		MacroAvg	0.82	0.82	0.82
Logistic regression	0.54	0	0.60	0.66	0.63
		1	0.51	0.49	0.50
		2	0.50	0.48	0.49
		MacroAvg	0.54	0.54	0.54
AdaBoost	0.54	0	0.63	0.65	0.64
		1	0.52	0.51	0.51
		2	0.50	0.49	0.50
		MacroAvg	0.55	0.55	0.55
Gaussian NB	0.48	0	0.70	0.41	0.51
		1	0.42	0.74	0.54
		2	0.49	0.32	0.38
		MacroAvg	0.54	0.49	0.48
KNN	0.75	0	0.78	0.61	0.68
		1	0.77	0.96	0.85
		2	0.72	0.70	0.71
		MacroAvg	0.76	0.75	0.75
RNN	0.59	0	0.66	0.63	0.64
		1	0.57	0.65	0.61
		2	0.55	0.50	0.52
		MacroAvg	0.59	0.59	0.59
LSTM	0.61	0	0.68	0.62	0.65
		1	0.61	0.67	0.64
		2	0.57	0.56	0.57
		MacroAvg	0.62	0.62	0.62
GRU	0.61	0	0.70	0.60	0.65
		1	0.61	0.68	0.64
		2	0.57	0.58	0.57
		MacroAvg	0.62	0.62	0.62

6.4 Results with K-Fold cross validation

Table 6. Results using K-Fold

Model name	**Accuracy**	**Class**	**Precision**	**Recall**	**F1- score**
Decision Tree	0.86	0	0.85	0.74	0.79
		1	0.92	0.98	0.95
		2	0.80	0.86	0.83
		MacroAvg	0.86	0.86	0.86
Logistic regression	0.54	0	0.59	0.66	0.63
		1	0.51	0.47	0.49
		2	0.51	0.49	0.50
		MacroAvg	0.54	0.54	0.54
AdaBoost	0.56	0	0.63	0.65	0.64
		1	0.52	0.51	0.51

		2	0.53	0.53	0.53
		MacroAvg	0.56	0.56	0.56
Gaussian NB	0.50	0	0.70	0.40	0.51
		1	0.42	0.72	0.53
		2	0.52	0.37	0.43
		MacroAvg	0.55	0.50	0.49
KNN	0.85	0	0.94	0.62	0.74
		1	0.85	0.99	0.92
		2	0.79	0.93	0.85
		MacroAvg	0.86	0.85	0.84

The evaluation of diabetes prediction models reveals a substantial improvement in performance when employing data balancing techniques, particularly SMOTE and ADASYN, in comparison to results without sampling. The initial models without data balancing exhibited notable imbalances in class predictions affecting accuracy, precision, recall, and F1 scores across multiple machine learning and deep learning Algorithms.

Every time SMOTE and ADASYN are compared, SMOTE turns out to be the most efficient sampling technique. When compared to ADASYN and the unsampled models, the models improved using SMOTE consistently showed higher F1-scores. With SMOTE, the F1-score—a critical data that finds a balance between accuracy and recall—improved significantly, suggesting a better way to find a balance between retrieving positive instances accurately and preventing false positives (precision).

AdaBoost, Gaussian NB, KNN, decision trees, logistic regression, and deep learning algorithms (RNN, LSTM, GRU) all showed consistently higher F1-scores when using SMOTE, demonstrating its effectiveness in reducing class imbalance and improving overall prediction accuracy. It also means that the production of synthetic samples by SMOTE successfully closes the gap between the majority and minority classes, resulting in more robust and balanced models.

The comparison concludes with a significant improvement in F1-scores, demonstrating SMOTE's improved performance over ADASYN and unsampled models. This emphasizes how important it is to choose the right data balancing method, with SMOTE appearing as a useful tool for strengthening diabetes prediction models by resolving class imbalance and boosting generalization in general.

7. Conclusion and Future Work

To sum up, the assessment of diabetes prediction models emphasizes how important data balancing strategies are for resolving class imbalance problems and enhancing the overall performance of the models. Key measures including precision, recall, and F1-score were affected by the early models' failure to accurately recognize the minority class due to sampling limitations. SMOTE is effective at creating synthetic samples that bridge the gap between minority and majority classes; of the sampling methods evaluated, it consistently performed better than ADASYN. The F1-score, in particular, showed a notable improvement with SMOTE, demonstrating its improved capacity to balance recall and precision.

Studying novel data balancing methods like Conditional Tabular GANs (CTGAN) may be useful for future research. Using Generative Adversarial Networks (GANs), CTGAN creates artificial samples while maintaining the complex correlations that comprise the dataset. The addition of CTGAN to the diabetes prediction framework may result in synthetic samples that are more realistic, which would improve the model's capacity to generalize to earlier unknown data.

Furthermore, adding strong algorithms to the prediction models, like Random Forest and Extra Tree Classifier, can be advantageous. These ensemble approaches are renowned for their capacity to manage complicated and varied datasets, offering increased precision and resistance against overfitting. Furthermore, the utilization of deep learning algorithms, such as Convolutional Neural Networks (CNNs), enables the capture of complex spatial dependencies present in the data. This is especially important for cases where spatial relationships are critical, like the analysis of medical images for the purpose of predicting diabetes.

In conclusion, in order to further improve the predictive power of diabetes prediction models, future research should concentrate on investigating sophisticated data balancing strategies like CTGAN and combining ensemble approaches (Extra Tree Classifier, Random Forest) with deep learning algorithms (CNNs). This multidisciplinary strategy could lead to the development of more reliable and accurate models, improving healthcare analytics and diabetes diagnosis.

References:

Chaturvedi, A., Mohapatra, L., Jain, A., Sharmila Emn, D. Suganthi and Romala Vijaya Srinivas (2023) 'An Innovative Approach of Early Diabetes Prediction using Combined Approach of DC based Bidirectional GRU and CNN'. doi:https://doi.org/10.1109/icesc57686.2023.10193133.

Patro, K.K., Allam, J.P., Sanapala, U., Marpu1, C.K., Samee,N.A., Alabdulhafith, M., and Plawiak, P., (2023) 'An effective correlation-based data modeling framework for automatic diabetes prediction using machine and deep learning techniques', *BMC Bioinformatics*, 24(1). Available at: https://doi.org/10.1186/s12859-023-05488-6.

T.V. Sai Krishna, Siva Kumar Pathuri, Kasturi Sai Sandeep, Manoj Kumar Padhi, Aswani, I. and D. Haritha (2023) 'An Efficient Machine Learning Classification Model for Diabetes Prediction'doi:https://doi.org/10.1109/csnt57126.2023.10134615.

M Malini, Gopalakrishnan, B., Dhivya, K.T. and S Naveena. (2021) 'Diabetic Patient Prediction using Machine Learning Algorithm', *2021 Smart Technologies, Communication and Robotics (STCR)*. doi:https://doi.org/10.1109/stcr51658.2021.9588925.

Shrivastava, A.K., Karthikeyan, V. Kaushik, S., Sudagar, M. (2022) 'Early Diabetes Prediction using Random Forest', in *3rd International Conference on Electronics and Sustainable Communication Systems, ICESC 2022 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp. 1154–1159. Available at: https://doi.org/10.1109/ICESC54411.2022.9885683.

Louisa Osiyi, O., Adebiyi Ayodele, A. and Igbekele Emmanuel, O. (2023) 'Prediction of Diabetes Mellitus in Developing Countries: A Systematic Review', in *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals, SEB-SDG 2023*. Institute of Electrical and Electronics Engineers Inc. Available at: https://doi.org/10.1109/SEB-SDG57117.2023.10124482.

Refat, M.A.R., Amin, M.A., Kushal, C., Yeasmin, M.N., Islam, M.K., (2021) 'A Comparative Analysis of Early Stage Diabetes Prediction using Machine Learning and Deep Learning Approach', in *Proceedings of IEEE International Conference on Signal Processing,Computing and Control*. Institute of Electrical and Electronics Engineers Inc., pp. 654–659. Available at: https://doi.org/10.1109/ISPCC53510.2021.9609364.

Okikiola, F.M., Adewale, O.S. and Obe, O.O. (2023) 'An Ontology-Based Diabetes Prediction Algorithm Using Naïve Bayes Classifier and Decision Tree', in *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals, SEB-SDG 2023*. Institute of Electrical and Electronics Engineers Inc. Available at: https://doi.org/10.1109/SEB-SDG57117.2023.10124491.

Rani, P., Lamba, R., Ravi Kumar Sachdeva, Priyanka Bathla and Aledaily, A.N. (2023) 'Diabetes Prediction Using Machine Learning Classification Algorithms', doi:https://doi.org/10.1109/icsca57840.2023.10087827.

Vijaya, J., Chandra, S. and Baghel, P. (2023) 'Diabetes Disease Prediction Using Various Metaheuristic Optimization Algorithms', in *2023 IEEE 8th International Conference for Convergence in Technology, I2CT 2023*. Institute of Electrical and Electronics Engineers Inc. Available at: https://doi.org/10.1109/I2CT57861.2023.10126222.

Gupta, T., Manoj Prasath T, Rani, C. and Rajesh Kumar M (2023) 'Diabetes Prediction using different Machine Learning Classifiers', in *ViTECoN 2023 - 2nd IEEE International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies, Proceedings*. Institute of Electrical and Electronics Engineers Inc. Available at: https://doi.org/10.1109/ViTECoN58111.2023.10157531.

Sivaranjani, S., Ananya, S., Aravinth, J. and Karthika, R. (2021) 'Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction', in *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*. Institute of Electrical and Electronics Engineers Inc., pp. 141–146. Available at: https://doi.org/10.1109/ICACCS51430.2021.9441935.

Gurunathan, P. T. S., Raghuraj, S., Roahit, S., and Nithishkumar, S., (2023) 'Web Application-based Diabetes Prediction using Machine Learning', *7th International Conference on Computing Methodologies and Communication, ICCMC 2023*. Institute of Electrical and Electronics Engineers Inc., pp. 296–302. Available at: https://doi.org/10.1109/ICCMC56507.2023.10083583.

Goel, G., Maguire, L., Li, Y., and McLoone, S. (2013) 'Evaluation of Sampling Methods for Learning from Imbalanced Data', *LNCS 7995*, pp. 392–401.

Davide Anguita, Ghio, A., Sandro Ridella and Sterpi, D. (2009) 'K-Fold Cross Validation for Error Rate Estimate in Support Vector Machines', *DMIN*, pp.291–297.

Lysa Eben, J., Jayasudha, R., Ramya, S., Kaliappan, S., Shobha Aswal and Khalid (2023) 'Diabetes Prediction Model for Better Clarification by using Machine Learning', in *6th International Conference on Inventive Computation Technologies, ICICT 2023 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp. 126–129. Available at: https://doi.org/10.1109/ICICT57646.2023.10134235.

Yahyaoui, A., Jamil, A., Rasheed, J., Yesiltepe, M. (2019) 'A Decision Support System for Diabetes Prediction Using Machine Learning and Deep Learning Techniques', *IEEE,* 978-1-7281-3992-0/19/.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002) 'SMOTE: Synthetic Minority Over-sampling Technique', *Journal of Artificial Intelligence Research*, [online] 16(16), pp.321–357. doi:https://doi.org/10.1613/jair.953.

Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2009) 'Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling Technique for Handling the Class Imbalanced Problem', *LNAI 5476*, pp. 475–482, 2009.

He, H., Bai, Y., Garcia, E. A., and Li, S.,(2008) 'ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning', *IEEE,* 978-1-4244-1821-3/08.