Performance Comparison of Machine Learning Models for Early Diabetes Prediction : Analysis and Web Application Development

Namitha Bhaskar

Chair: Professor Erik Golen

ISTE 793 - Capstone is Information Technology and Analytics

Table of Contents

Table of Figures

Table of Tables

Section 1 – Introduction

Diabetes is one of the most ubiquitous conditions today where more than 830 million people have this condition around the world[2]. According to the World Health Organization (WHO), in 2021, diabetes resulted in 2 million deaths due to kidney failure and was the cause of 11% of all cardiovascular-related deaths[2]. Diabetes is a metabolic condition where the human body is unable to produce sufficient amounts of insulin hormone or where the body develops resistance to the insulin hormone[1][2]. There are primarily two types of diabetes, Type 1 where a person is born with diabetes or gets it in early ages[1] and type 2 which is an acquired diabetes due to several medical and physiological factors.

The International Diabetes Federation (IDF)'s Diabetes Atlas (2025) reports that 11.1% of the adult population (20-79 years) has diabetes, with over 4 in 10 people unaware that they are living with the condition. By 2050, IDF projections show that 1 in 8 adults will be living with diabetes which is approximately an increase of 46%. 4 in 5 adults with diabetes live in low- and middle-income countries[3].

Late detection of these uncontrolled high sugar levels can lead to the narrowing and blockage of blood vessels throughout the body. This can lead to vision loss (diabetic retinopathy), loss of feeling in the feet and subsequent ulcer formation (diabetic neuropathy), kidney damage and failure (diabetic nephropathy), and even strokes and heart attacks[2][4]. The prevalence of this chronic disease is higher in lower and middle income countries and is also one condition that has a high number of leading comorbidities like organ failure and heart attacks[2][3]. Furthermore, diabetes management is also a complex issue with multiple facets based on patient profiles.

Diabetic complications and its associated death rate have been exponentially increasing over the last decade. This condition is also an economic burden since the primary management is with medicines[2]. Since the prevalence of diabetes is higher in lower and middle economic countries, the affordability of the medicine is not widespread. Due to this there is a compliance issue where people often do not take the medication properly leading to an increase in risk of comorbidity cases and even death [2]. As the saying goes, "a stitch in time saves nine", it is essential to identify this condition early and prevent it to better improve the health of the population as a whole.

There are multiple ways in which this condition can be predicted, but this project aims to use machine learning to build a model which will be used to predict diabetes. Today, researchers and tech people have shown great results in predicting the risks associated with this disease. Rapid evolution of technology and techniques like machine learning and deep learning combined with the easy availability of medical data has made the once impossible very much possible today. Diagnosis of diabetes, easy management of blood glucose levels and evaluation of related complications has been made easy today with machine learning models [5].

The focus of this project is to build a predictive model to cater to people of all walks of life. It is not only targeted to individuals who are at an increased risk but also to normal people who are health conscious and who want to keep a check on this chronic disease. This project will help healthcare professionals to rightly identify high-risk patients and better counsel them to start

preventative measures. By integrating data analytics with healthcare, this project aligns with the growing demand for AI-driven medical decision support systems. The ability to predict diabetes risk using non-invasive, easily collected health indicators can benefit both clinicians and individuals, particularly in underserved communities.

Section 2 – Prior Work

Prior work related to diabetes and Machine Learning is very vast and a well known research topic. To start with, authors of paper [5] talk about various machine learning models to detect diabetes at an early stage. The paper focuses on the use of machine learning and deep learning techniques for early-stage diabetes prediction. It uses a benched-mark UCI repository dataset with 16 attributes and 520 data points for training and testing. Performance metrics such as precision, recall, f1-score, execution time, and ROC value are used to evaluate the performance of various algorithms. The XGBoost classifier is found to be the best for predicting diabetes at the initial stage, with about 99.99% training accuracy and 99.0% testing accuracy. Their objectives are mainly to analyze publicly available datasets, perform comprehensive analysis of machine learning and deep learning models and evaluate their performance. They have started with simple logistic regression and then go to discuss Artificial Neural Network and LSTM models.

Research paper [4] offers a different angle for diabetes prediction. The authors have used Hybrid Artificial Neural Network on the Pima Indian diabetes Dataset. They have used a technique called Scaled Conjugate Gradient algorithm which is a supervised learning based feed forward algorithm. They have achieved an accuracy of 100% with a 16 hidden layer system.

The research paper [6] talks about diabetes prediction using ensemble learning. They have used boosting techniques. The paper beautifully explains each boosting model used and also provides clear comparison between each model. They have also illustrated the pre-processing techniques in a very simple and efficient way. The comparison table of all the previous work they researched was an elegant way to show how previous work was done and how they have worked. Data handling and preprocessing done in this paper is the inspiration for this project. The researchers have achieved the highest accuracy of 96% using the Gradient Boosting technique.

Following the previous two papers, paper [7] gives a deep understanding of the dataset they have built using a survey and also the preprocessing techniques and model they have used. They have used techniques such as SMOTE and ADASYN to handle data imbalance. They have also used explainable AI techniques to predict this condition. Finally the researchers have deployed this application as a web application and an Android application. They have used Spyder and Heroku for the web deployment and mobile app deployment respectively.

Furthermore, research paper [8] is a recent and latest paper which talks about prediction of diabetes using medical variables. They have used an open dataset from Kaggle platform to build the model. The focus of this paper and their Literature review thoroughly explains each medical variable and how it has a role in the prediction. The visualizations done by the author are very intuitive and well done. Visualizations provide a very in-depth idea about the correlation between variables like age, gender, BMI and diabetes. Each visualization has been interpreted in simple and clean language which any reader can easily grasp and understand.

Delving more into the machine learning models, the paper [9] aims to evaluate the performance and compare machine learning models for Classification. Their previous work talks about how the PIMA Indian diabetes Dataset and Sylhet Dataset have been used by the majority of researchers in previous years. They have discussed in detail the results obtained by many for the above mentioned datasets. This paper has been written based on the work done on the CDC diabetes Dataset that has 253,680 records and 21 predictor features. They have selected models like Naive Bayes, Support Vector Machine, Bayes Net to build a classification model. They have also used J48 decision trees and Lazy IBK i.e Instance Based KNN algorithm to build the model.

Scholarly articles [10] and [11] focus on Data Visualization. Paper [11] Talks about how a team works to analyze and visualize data. They have used diabetes data from NHANES by CDC. They have given an all-rounder view starting from data cleaning, data preprocessing, handling missing values followed by model building, analyzing data and presenting the findings visually. They have also incorporated a Business Intelligence Architecture using Data Warehouse in their study. They have successfully used ETL (Extract, Transform and Load) to load the data, thoroughly analyze it and create colorful and meaningful visualizations. Similarly, paper [10] talks about visualization using Power BI software. They have used CDC data to show how diabetes has turned out to become a burden for the American people.

Feature selection and Dimensionality reduction are two very essential concepts of Machine Learning. [12] talks about how the authors have performed these techniques on the Pima Indian diabetes Dataset. They have also tried and tested various models and have gotten a validation accuracy result of 81% for their proposed model.

The work done by authors of paper [13] revolves around finding the top 10 predictors that cause Type 2 diabetes. They have trained a model using the XGBoost algorithm and have used SHaply Additive explanation(SHAP) to interpret the model and analyse the importance of each feature. They have visualized their results and found that hBA1C levels (an indicator of amount of glucose in the blood) was the strongest predictor and held #1 position in the Top 10 list.

Authors of papers [14],[15] talk about the significant use of Ensemble Learning methods to predict Diabetes. They have elaborately analysed the effect of imbalanced datasets and have

discussed resampling methods to improve the imbalanced datasets. They have also talked about the traditional models like SVM and Decision Trees and their performance in Diabetes Detection

Thuraka et al. [16] enhance diabetes prediction using hybrid feature selection with AdaBoost, demonstrating improved classification accuracy through optimized feature selection. They have used ANOVA statistical analysis and LASSO regression to select relevant features with which an AdaBoost model has been built. Though they have taken a small sized dataset to do this, their methodology of solving the problem is different from others.

Authors of papers [17] and [18] have used Optuna which is python's hyper parameter tuning package to tune the models they built. They have worked on the Pima Indian Diabetes dataset and have achieved good numbers on all the evaluation metrics. Optuna is a tuning method which is much faster and accurate than conventional tuning methods like GridSearch and RandomSearch. Optuna uses an inbuilt process where it trains and validates the model to give out the best hyperparameters for a given model.

Paper [19] talks about the use of resampling technique SMOTE with Tomek links to predict three common diseases Diabetes, Parkinson's and Vertebral Column Pathologies. They have identified that using this resampling technique has increases model accuracy and evaluation metrics more than using just the individual resampling techniques

Finally coming to a future perspective of handling scenarios using Big Data, this paper [20] talks about how Big Data Analytics can play a role in making far-reaching changes in the management of diabetes and healthcare decision-making. They have touched upon topics related to both Big Data and Machine Learning. The paper talks about growing data size and how big data analytics comes to the rescue. They have used technologies like Apache Spark to handle big data and then performed analysis on the data using several machine learning models like Logistic Regression, Decision Trees and Random Forest. This article gives a refreshing angle of how data is growing today and how people in the tech world can handle it and improve quality of life.

Section 3 – Methodology

The methodology section outlines the step-by-step process followed in building this project from start to finish. The process began with an initial exploration of the CDC's Diabetes Dataset to understand its structure, variables, and class imbalance. Preprocessing and feature engineering steps were then applied to transform the data into a model-ready format, including binning and encoding. A combination of statistical methods and regularization techniques was used for feature selection to retain only the most informative predictors. To address the class imbalance in the dataset, multiple resampling techniques were evaluated, and the most effective approach was applied to the training data. Various ensemble models were then built and compared based on classification performance metrics. Finally, the best-performing models were deployed through a

user-friendly web application built with Streamlit, allowing real-time diabetes risk prediction through a guided input form. Optional interpretability features and user recommendations were also considered to enhance the transparency and usability of the system.

3.1 Loading the Dataset and Basic Data Exploration

As the first step to this project, the dataset was loaded as a pandas dataframe. The dataset used is CDC's Diabetes Dataset. This dataset was a part of CDC's Behavioral Risk Factor Surveillance System (BRFSS 2015). It has a total of 253,680 health-related survey responses with 21 predictors. The predictors are a mix of categorical and continuous variables. The target variable is binary with 0 corresponding to people who are not diabetic and 1 corresponding to people who are pre-diabetic or diabetic. During preliminary exploration, class imbalance was noted, with a significantly higher proportion of non-diabetic cases. Summary statistics, value counts, and variable distributions were examined to understand the data structure and quality.

3.2 Data Cleaning

As part of data cleaning, all the missing and duplicate values present in the dataset were handled. There was no missing data for any of the predictor columns. There were a total of 24,206 duplicate data rows which were dropped. The final cleaned data frame had a total of 229,474 rows and 22 columns.

3.3 Feature Engineering

In addition to data cleaning, several steps of data processing and feature engineering were done to prepare the data for model training. Among the 21 predictors, BMI, Mental Health and Physical Health variables were treated as continuous variables for all further steps of the methodology. Features Age, Education and Income were binned into categorized variables with 3 categories each to reduce dimensionality and improve model interpretability. Binary variables were already encoded as Yes/No (1/0) and above mentioned multi-category features were retained as numerical indicators.

3.4 Frequency Analysis

Frequency Analysis was performed to get a picture at the distribution of each variable. This gave a picture of the patterns present in the data and also the class imbalance for the target variable. Some variables had an even distribution across the two target classes but variables like cholesterol check (CholCheck), difficulty in walking (diffWalk) and physical activity (PhysActivity) had some imbalance between the target classes.

3.5 Correlation Analysis

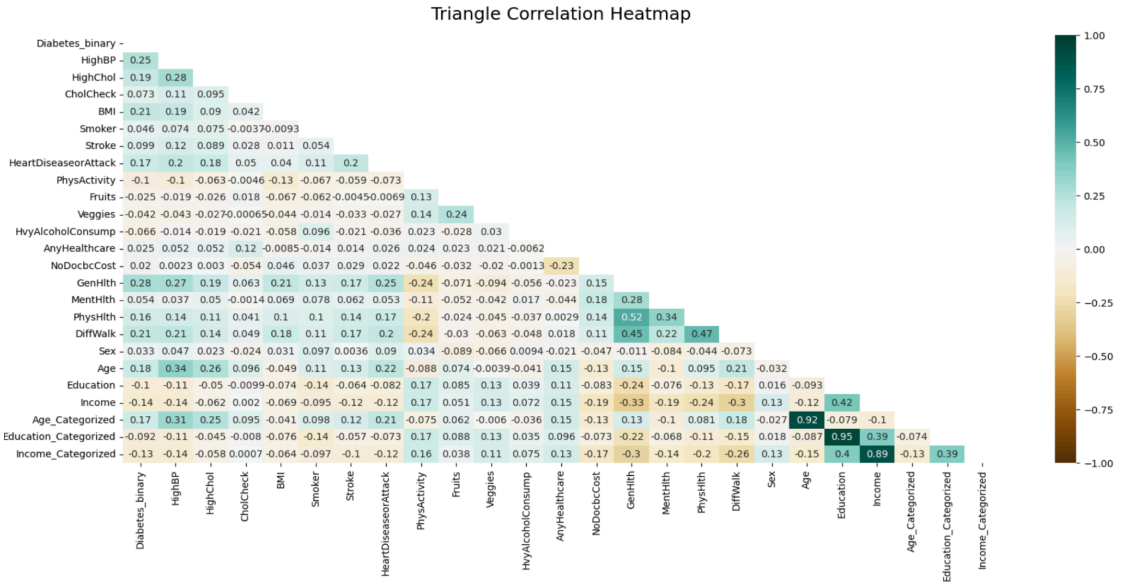


Figure 1 - Triangle Correlation Heatmap

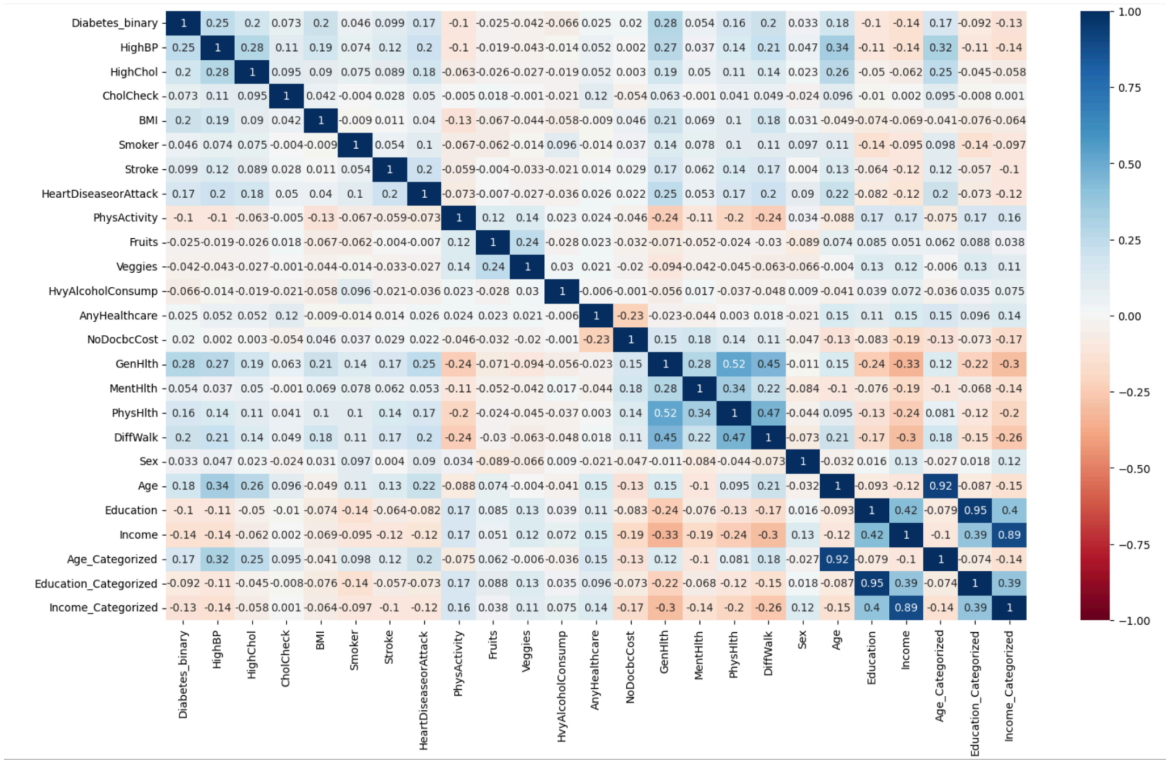


Figure 2 - Full Correlation Heat Map

A full correlation heatmap shown in Fig 2 was generated to examine the relationships between all variables in the dataset. Health-related factors showed moderate associations for instance, highBP correlated with age (0.34), and DiffWalk with PhysHlth(0.45), suggesting that older individuals and those in poorer physical health are more likely to experience mobility

difficulties. Negative correlations were also informative; GenHlth showed a moderate negative correlation with Income (-0.30), and NoDocbcCost with Income (-0.33), indicating that individuals with lower income levels reported worse general health and were more likely to avoid doctor visits due to cost concerns. Most other variables exhibited weak correlations, suggesting minimal direct linear relationships. Additionally, a triangle heatmap shown in Fig 1 was used to simplify the visualization by removing redundant values, enhancing interpretability. Together, these plots provided meaningful insights into how health, demographic, and socioeconomic variables interact within the dataset.

A plot was made to see the strength of correlations between the predictor variables and the target variable.

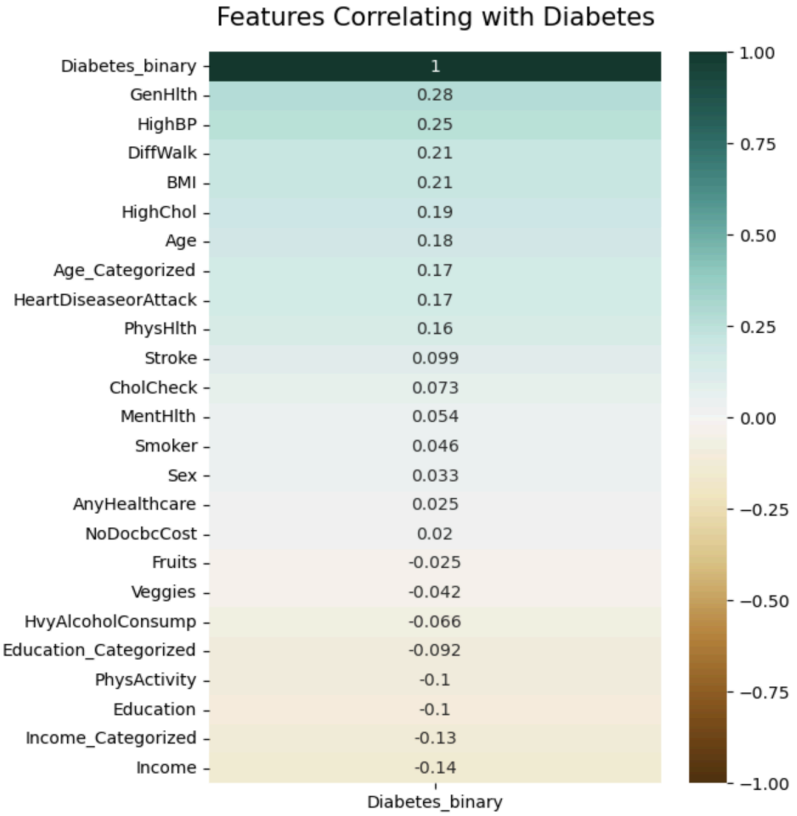


Figure 3 - Strength of Correlation Plot

The strength of correlation plot shown in Fig 3 highlights how individual features relate to the target variable, Diabetes_binary. GenHlth, HighBP, DiffWalk, and BMI show the strongest positive correlations, suggesting a higher likelihood of diabetes among individuals reporting poor general health, high blood pressure, mobility issues, and higher BMI. Negative correlations were observed for Income, Education, and PhysActivity, indicating that healthier lifestyles and better socioeconomic conditions may be associated with a lower risk of diabetes. Overall, most correlations were weak, supporting the need for multivariate modeling.

3.6 Feature Selection using Statistical Analysis

Feature selection was conducted in two stages—bivariate analysis followed by multivariate analysis—to identify the most statistically significant predictors for diabetes classification. Dummy variables were created for all the categorical variables for statistical analysis. In the bivariate analysis, chi-square tests were applied to all categorical variables, while independent t-tests were used for continuous variables like BMI, Mental Health, and Physical Health. As shown in the analysis results shown in table 1, all features exhibited p-values less than 0.05, indicating statistically significant differences in means or proportions between diabetic and non-diabetic groups.

To further confirm the robustness of selected variables, a multivariate analysis using logistic regression with L1 penalty was performed. The resulting coefficients and associated p-values shown in table 2 reaffirmed the significance of all variables. For categorical features with more than two categories (e.g., GenHlth, Age, Education, Income), a Likelihood Ratio Test (LRT) was conducted to determine the global p-values. All LRT p-values were below 0.05, supporting their inclusion. Therefore, based on consistent significance across both bivariate and multivariate tests, all available features were retained for model development.

Bivariate Analysis Results:

	Feature	Test	Statistic	P-value
0	HighBP	Chi-square	14840.421805	0.000000e+00
18	BMI	T-test	-91.959495	0.000000e+00
17	Income_Categorized	Chi-square	4005.884459	0.000000e+00
16	Education_Categorized	Chi-square	1938.056162	0.000000e+00
15	Age_Categorized	Chi-square	6687.971099	0.000000e+00
14	GenHlth	Chi-square	18193.703885	0.000000e+00
12	DiffWalk	Chi-square	9670.630180	0.000000e+00
20	PhysHlth	T-test	-61.969651	0.000000e+00
6	PhysActivity	Chi-square	2312.703694	0.000000e+00
5	HeartDiseaseorAttack	Chi-square	6491.585745	0.000000e+00
4	Stroke	Chi-square	2256.534055	0.000000e+00
1	HighChol	Chi-square	8719.656978	0.000000e+00
2	CholCheck	Chi-square	1205.929127	3.138608e-264
9	HvyAlcoholConsump	Chi-square	997.308681	6.906569e-219
19	MentHlth	T-test	-22.863103	5.033966e-115
3	Smoker	Chi-square	474.898448	2.753229e-105
8	Veggies	Chi-square	399.389542	7.478662e-89
13	Sex	Chi-square	245.553841	2.419784e-55
10	AnyHealthcare	Chi-square	146.936849	8.100957e-34
7	Fruits	Chi-square	141.054995	1.565008e-32
11	NoDocbcCost	Chi-square	92.041500	8.487775e-22

Table 1 - Bivariate Analysis Results

```
Optimization terminated successfully.
         Current function value: 0.372972
         Iterations 7
```

Logit Regression Results

Dep. Variable:	Diabetes_binary	**No. Observations:**	229474
Model:	Logit	**Df Residuals:**	229447
Method:	MLE	**Df Model:**	26
Date:	Wed, 12 Mar 2025	**Pseudo R-squ.:**	0.1281
Time:	14:58:45	**Log-Likelihood:**	-85587.
converged:	True	**LL-Null:**	-98165.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
BMI	0.0100	0.001	12.751	0.000	0.008	0.012
MentHlth	-0.0095	0.001	-11.466	0.000	-0.011	-0.008
PhysHlth	-0.0027	0.001	-3.387	0.001	-0.004	-0.001
HighBP_1	0.8635	0.014	60.260	0.000	0.835	0.892
HighChol_1	0.5707	0.013	42.979	0.000	0.545	0.597
CholCheck_1	-1.2621	0.026	-49.143	0.000	-1.312	-1.212
Smoker_1	-0.1880	0.013	-14.731	0.000	-0.213	-0.163
Stroke_1	0.1255	0.025	5.066	0.000	0.077	0.174
HeartDiseaseorAttack_1	0.3459	0.018	19.642	0.000	0.311	0.380
PhysActivity_1	-0.3197	0.013	-24.080	0.000	-0.346	-0.294
Fruits_1	-0.1503	0.013	-11.556	0.000	-0.176	-0.125
Veggies_1	-0.2720	0.015	-18.756	0.000	-0.300	-0.244
HvyAlcoholConsump_1	-0.9927	0.038	-26.088	0.000	-1.067	-0.918
AnyHealthcare_1	-1.0085	0.023	-43.061	0.000	-1.054	-0.963
NoDocbcCost_1	-0.4431	0.022	-20.009	0.000	-0.487	-0.400
DiffWalk_1	0.2737	0.017	16.552	0.000	0.241	0.306
Sex_1	0.0556	0.013	4.328	0.000	0.030	0.081
Age_Categorized_2	-0.0607	0.022	-2.820	0.005	-0.103	-0.019
Age_Categorized_3	0.2356	0.023	10.269	0.000	0.191	0.281
Education_Categorized_2	-0.1393	0.015	-9.127	0.000	-0.169	-0.109
Education_Categorized_3	-0.1820	0.016	-11.187	0.000	-0.214	-0.150
Income_Categorized_2	-0.2348	0.015	-15.400	0.000	-0.265	-0.205
Income_Categorized_3	-0.3118	0.018	-16.934	0.000	-0.348	-0.276
GenHlth_2	-0.3734	0.023	-16.294	0.000	-0.418	-0.328
GenHlth_3	0.2799	0.022	12.826	0.000	0.237	0.323
GenHlth_4	0.7071	0.025	27.755	0.000	0.657	0.757
GenHlth_5	0.8929	0.034	26.186	0.000	0.826	0.960

Table 2 - Multivariate Analysis Results

3.7 Visualizing Target Variable and Key Features

Based on the statistical analysis, all the key features were visualized and analysed. The target variable was analysed by plotting a pie chart shown in Fig 5 to see the distribution of the target class. A horizontal bar shown in Fig 6 plot was also made for the target variable. The percentage distribution of the target variable was also calculated as part of this step, shown in Fig 4.

```
0 – No Diabetes, 1 – Pre-Diabetic or Diabetic

Diabetes_binary
1.0     35097
0.0    194377
Name: count, dtype: int64

Diabetes_binary
1.0    15.294543%
0.0    84.705457%
```

Figure 4 - Percentage Distribution of the Target Variable

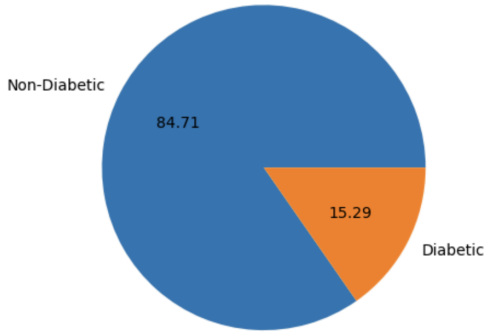


Figure 5 - Pie Chart of Target Variable Diabetes

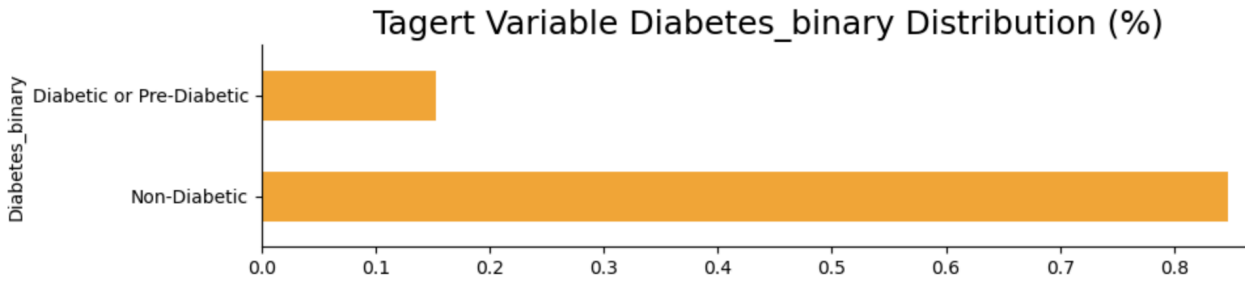


Figure 6 - Bar Chart of Target Variable Diabetes

Following the visualization of the target variable, all the predictor variables were also visualized. Kernel Density Estimation(KDE) Plots were plotted for all continuous variables with the target variable Diabetes.

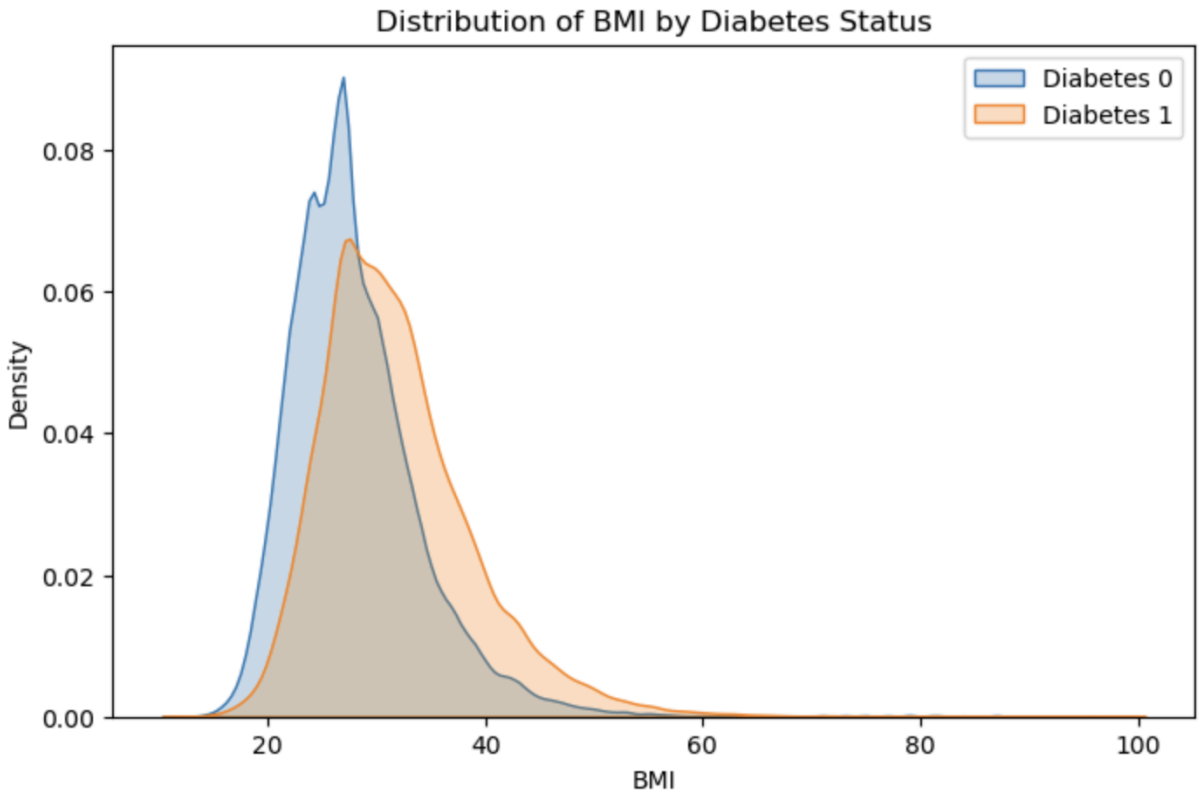


Figure 7 - KDE Plot for BMI vs Diabetes

Distribution of BMI by diabetes status - This plot shown in Fig 7 compares the Body Mass Index (BMI) distribution between individuals with and without diabetes. The orange curve (diabetic individuals) is slightly shifted to the right, indicating that people with diabetes generally have a higher BMI. The peak of the diabetic group is broader and flatter, suggesting more variability in BMI among diabetics. This supports the well-established link between higher BMI and increased risk of diabetes.

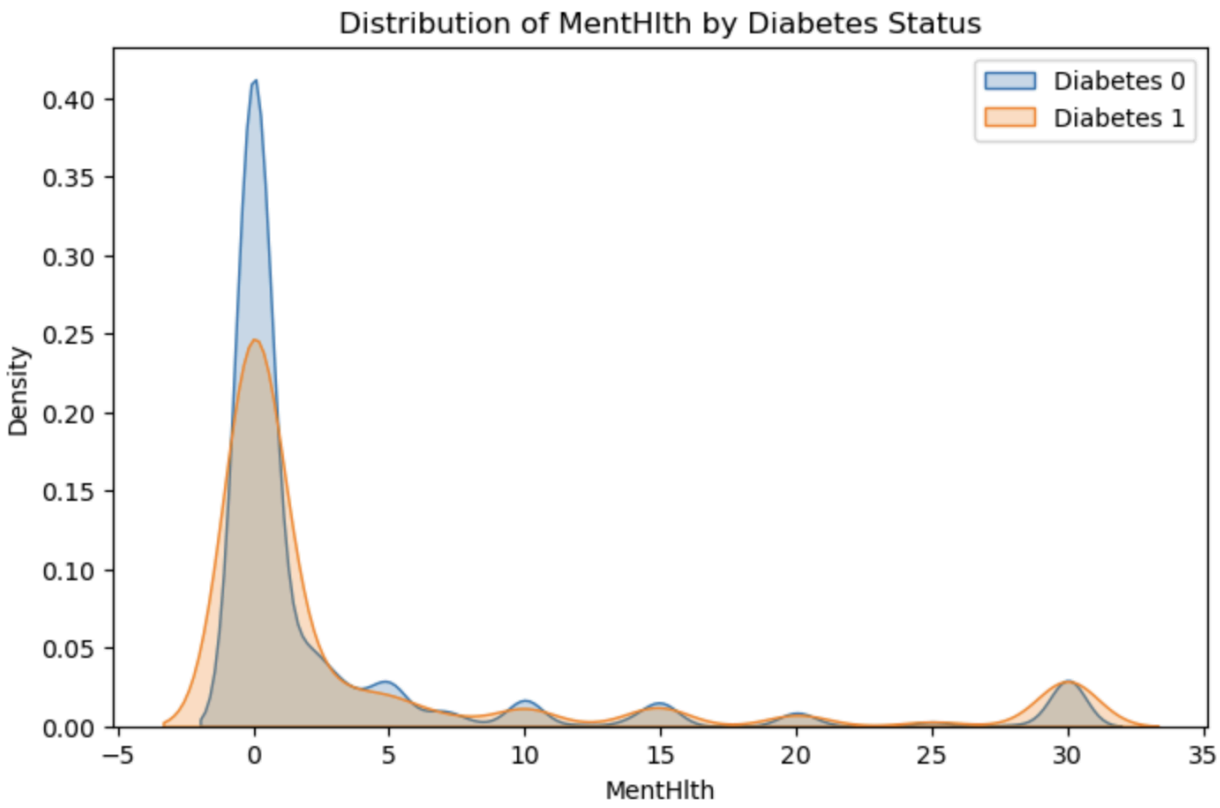


Figure 8 - KDE plot for Mental Health vs Diabetes

Distribution of Mental Health Days by Diabetes Status - The plot shown in Fig 8 shows the number of days in the past month when individuals reported their mental health was not good. Both diabetic and non-diabetic groups show a peak near zero days, meaning most people did not report mental health issues. However, the diabetic group has a slightly heavier tail, indicating a higher frequency of mental health concerns among those with diabetes. This aligns with research that shows diabetes is often associated with increased stress, anxiety, or depression.

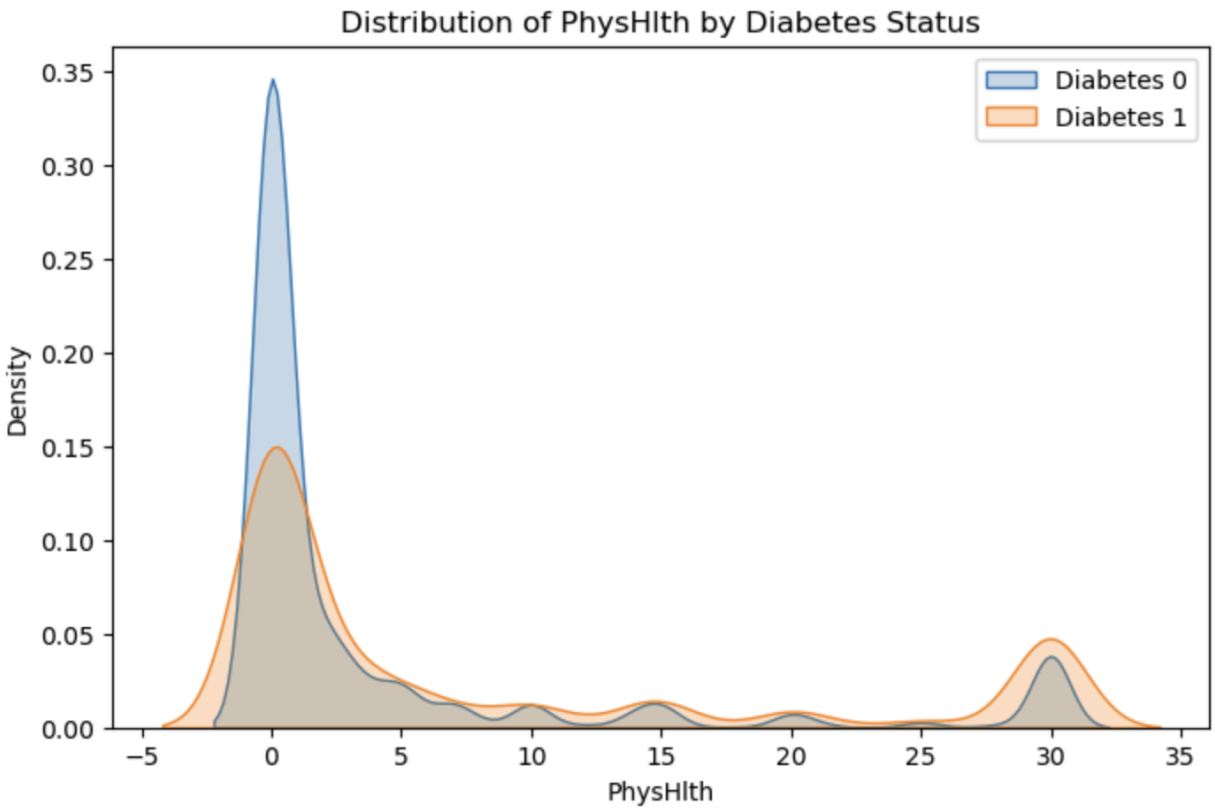
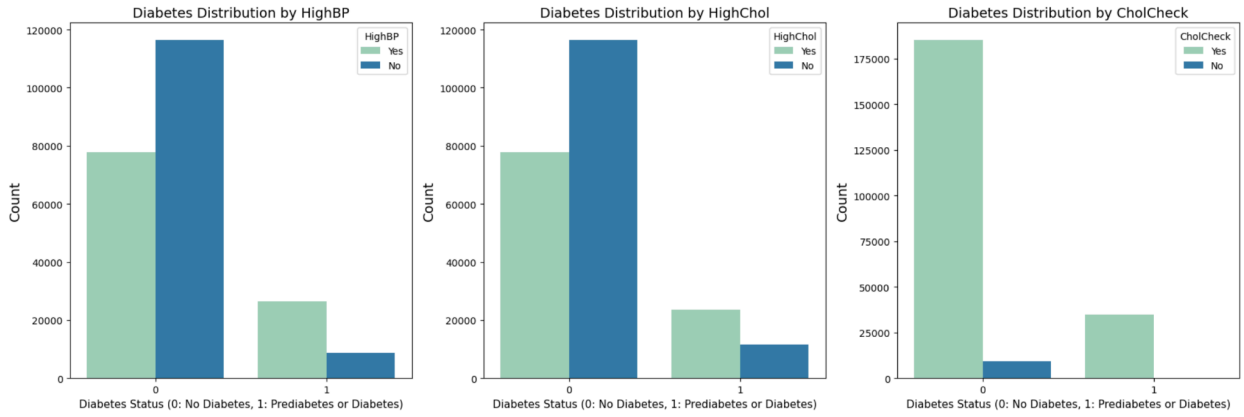


Figure 9 - KDE Plot for Physical Health vs Diabetes

Distribution of Physical Health Days by Diabetes Status - This plot shown in Fig 9 reflects the number of days when individuals experienced poor physical health in the past 30 days. Similar to mental health, both groups peak around zero days, but the diabetic group again has a wider spread. The diabetic curve extends further to the right, suggesting that diabetics report more days of physical discomfort or illness. This highlights the physical burden of managing a chronic condition like diabetes.

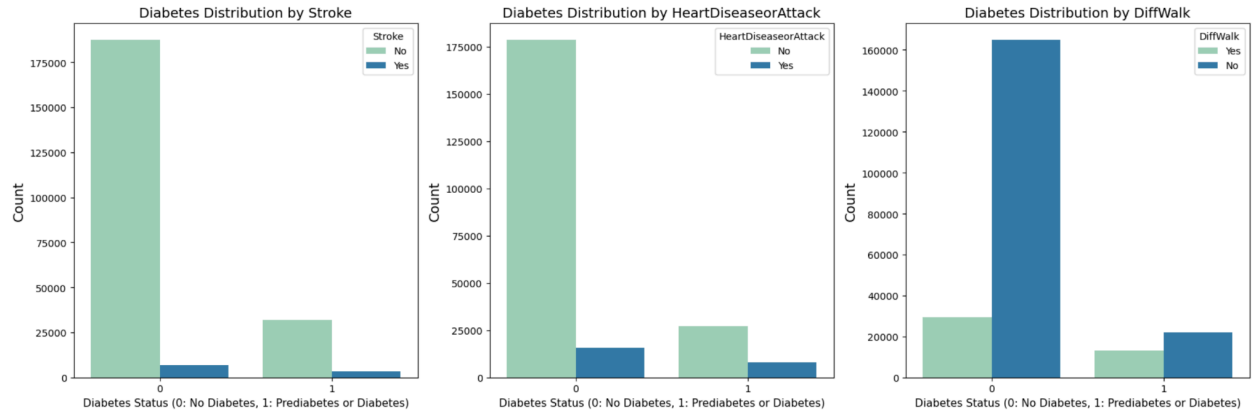
Following the visualizations of continuous variables, all the categorical variables were also visualized with the target variable to analyze patterns and to interpret the diabetes status for each variable. For better interpreting the predictors, the categorical variables were grouped into 3 categories - Health related features, Lifestyle related features and Demographic features.

Health Related Features



A **B** **C**

Figure 10 - Count Plot of Diabetes vs HighBP(A), High Cholesterol(B) and Cholesterol Check(C)



A **B** **C**

Figure 11 - Count Plot of Diabetes vs Stroke(A), Heart Disease or Attack(B) and Difficulty Walking(C)

Distribution of HighBP by Diabetes Status - Count plot 10 A shows that individuals with high blood pressure ("Yes") show a noticeably higher count of diabetes cases compared to those without high BP. This suggests a strong association between hypertension and increased diabetes risk.

Distribution of High Cholesterol by Diabetes Status - A similar trend is observed with high cholesterol levels shown in Fig 10 B. People with high cholesterol are more represented in the diabetic group, indicating its contribution to diabetes development.

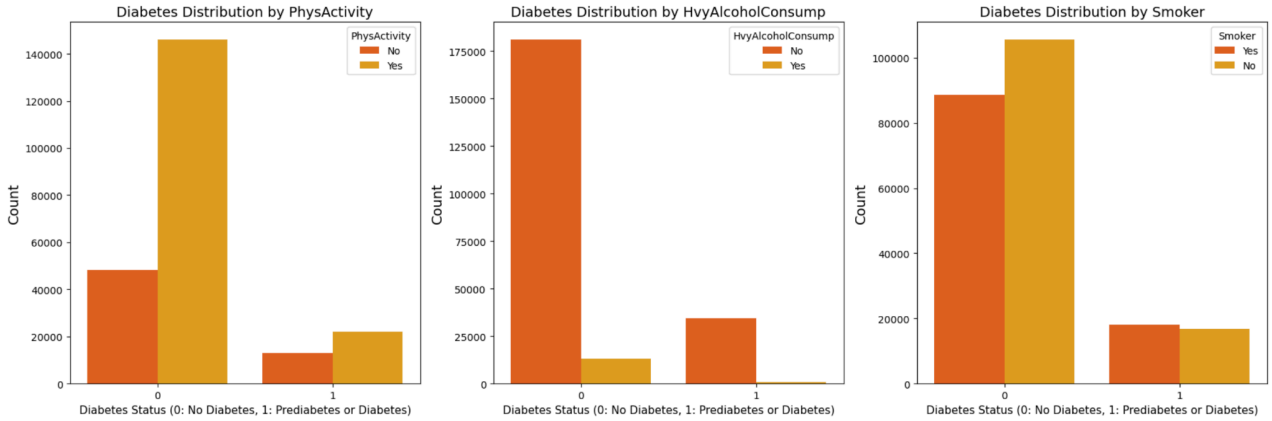
Distribution of people who have had their Cholesterol checked in the past 5 years by Diabetes status - Plot in Fig 10 C shows most individuals, regardless of diabetes status, have had a cholesterol check. However, those who haven't undergone a check seem to have a slightly higher prevalence of diabetes, possibly hinting at delayed detection or lower healthcare engagement.

Distribution of people who have had an incident of Stroke with Diabetes Status - Fig 11 A shows a small proportion of the diabetic group has had a stroke, suggesting that a history of stroke may be associated with diabetes, although the absolute numbers are low.

Distribution of people who have a history of Heart Disease or Heart Attack by Diabetes Status - Fig 11 B shows that those who reported heart disease or heart attacks show higher counts among diabetics, supporting the known link between cardiovascular issues and diabetes.

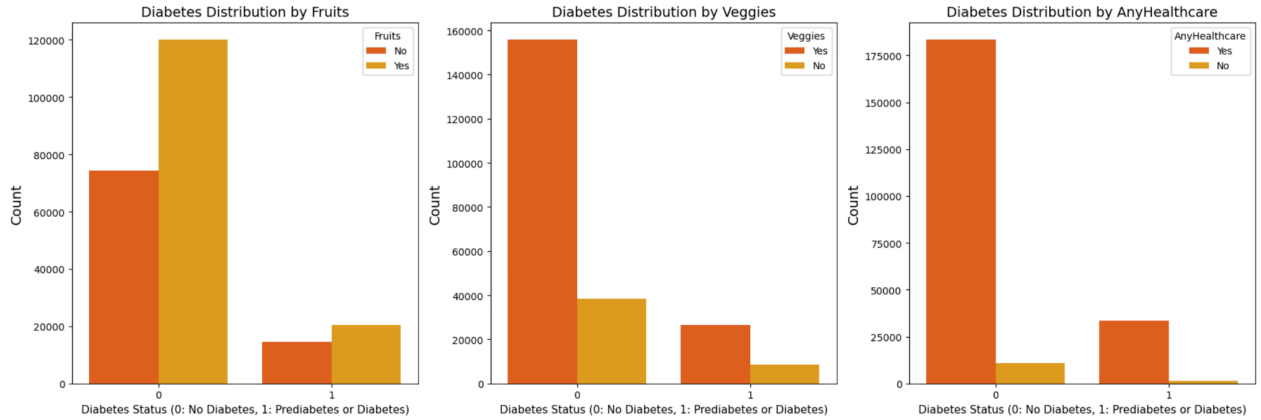
Distribution of people having Difficulty in Walking by Diabetes Status - Fig 11 C shows that difficulty in walking is significantly more common in the diabetic group, likely reflecting complications or comorbidities that arise with the condition.

Lifestyle Related Features



A **B** **C**

Figure 12 - Count Plot of Diabetes vs Physical Activity(A), Heavy Alcohol Consumption(B) and Smoker(C)



A **B** **C**

Figure 13 - Count Plot of Diabetes vs Consumption of Fruits(A), Consumption of Veggies(B) and Access to Any Health Care(C)

Distribution of Physical Activity by Diabetes Status - Fig 12 A shows individuals who reported no physical activity in the past 30 days had a visibly higher prevalence of diabetes compared to those who were physically active. This highlights physical inactivity as a risk factor for diabetes.

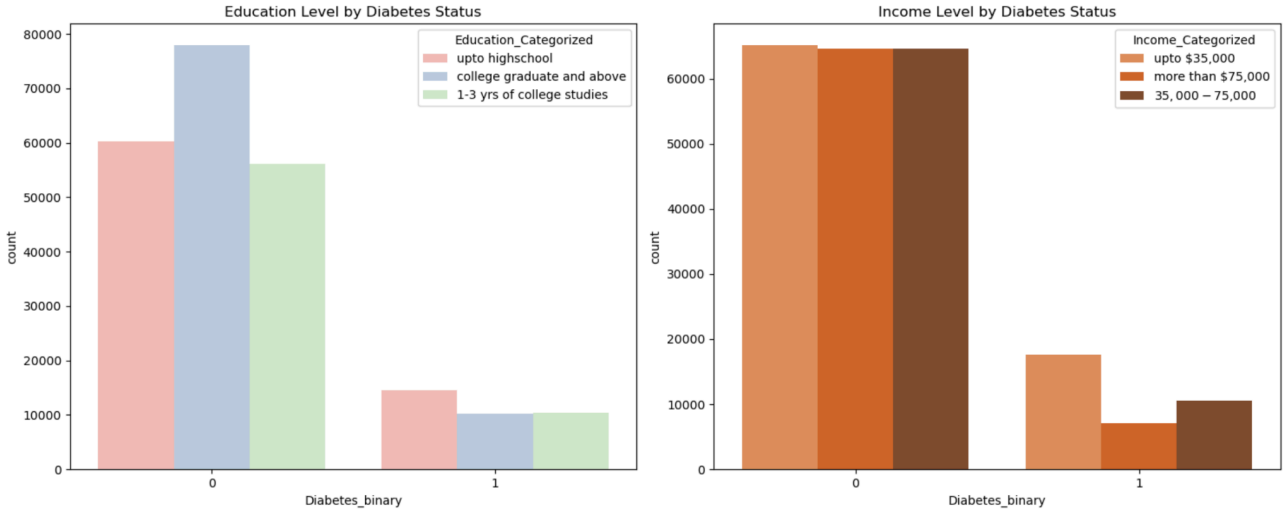
Distribution of people with Heavy Alcohol Consumption habit by Diabetes Status - Surprisingly, a larger proportion of non-heavy drinkers are observed in the diabetic group shown in Fig 12 B. However, this may be due to overall higher representation of non-heavy drinkers in the dataset or lifestyle changes post-diagnosis.

Distribution of people who have smoked at least a 100 cigarettes in their lifetime by Diabetes Status - Fig 12 C shows the difference in diabetes prevalence between smokers and non-smokers is relatively small. Although slightly more non-smokers are diabetic, the difference is not visually significant, indicating a weaker relationship.

Distribution of people who consume fruits and vegetables at least once per day by Diabetes Status - Plots shown in Fig 13 A and 13B shows those who consumed fruits and vegetables at least once per day had a marginally lower incidence of diabetes than those who did not. This suggests that healthy eating habits may have a protective effect.

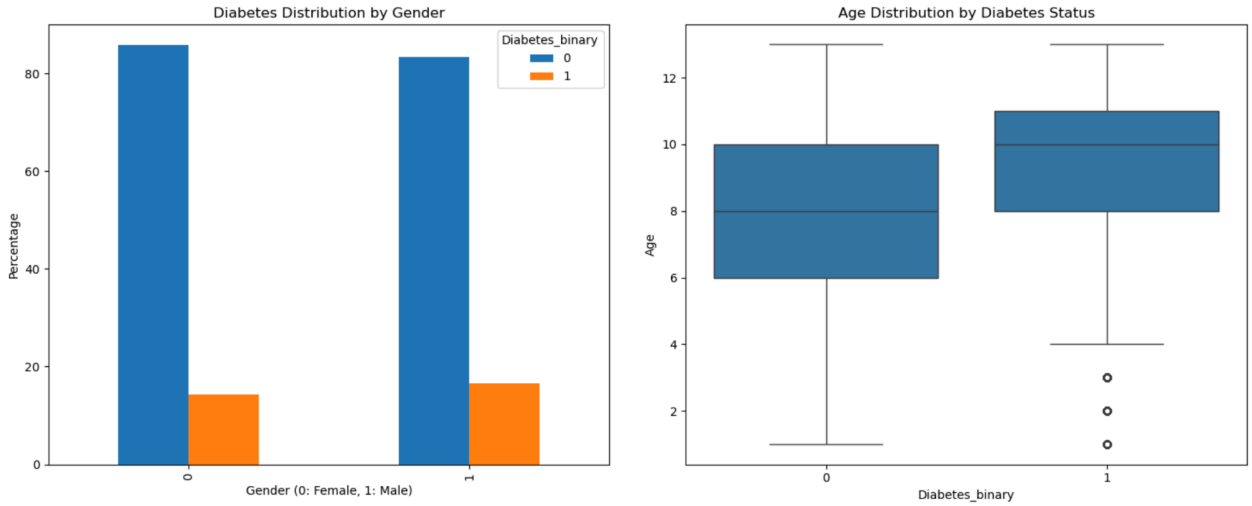
Distribution of people having any kind of healthcare coverage by Diabetes Status - Fig 13 C shows the majority of individuals with access to healthcare reported no diabetes. Those without access showed a slightly higher rate of diabetes, suggesting early detection and management play a role in controlling the disease.

Demographic Features



A **B**

Figure 14 - Count Plot of Diabetes vs Education Level(A), Income Level(B)



A **B**

Figure 15 - Count Plot of Diabetes vs Genger(A), Age(B)

Distribution of people's Education Level by Diabetes Status - Fig 14 A shows individuals with only high school education or less showed a relatively higher number of diabetes cases compared to those with some college or a college degree. This indicates that lower educational attainment may be associated with an increased risk of diabetes.

Distribution of people's Income Level by Diabetes Status - Fig 14 B shows people earning less than $35,000 had the highest number of diabetes cases, while those with income above $75,000 had fewer cases. This trend suggests a possible link between lower income and higher diabetes risk, possibly due to limited access to healthcare or healthier lifestyles.

Distribution of Gender by Diabetes Status - Fig 15 A shows that the diabetes distribution by gender is fairly similar for both males and females, though slightly higher proportions of diabetes were seen among males.

Distribution of Age by Diabetes Status - The boxplot in Fig 15 B shows that individuals with diabetes tend to fall into higher age categories compared to non-diabetics, indicating that diabetes risk increases with age. There is also a wider spread of ages among diabetics.

3.8 Resampling Techniques

Resampling is a process where samples are repeatedly drawn from datasets with or without replacements. This is done to estimate the accuracy of a model or increase the robustness of a model. This technique is a key method for handling class imbalance in datasets. In this project, a combination of 5 resampling methods were tried with 11 machine learning models. These 11 models were a combination of supervised models and ensemble models. The 11 models tried were Logistic Regression, Naive bayes, K-Nearest Neighbour (KNN), Support Vector Machine

(SVM), Decision Tree, Random Forest, Gradient Boosting algorithm, AdaBoost, XGBoost, CatBoost and LightGBM. All the models were run without applying any resampling and assessed to see the performance. Following that, undersampling methods Random Under Sampling (RUS) and Edited Nearest Neighbour (ENN) were tried. Oversampling methods Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic Sampling Approach (ADASYN) were tried. Following both undersampling and oversampling techniques combination resampling was also tried. SMOTE-ENN was the combination technique tried.

Fig 16 shows the distribution of target variables before resampling and Fig 17 shows the distribution of the target variable after applying each resampling technique.

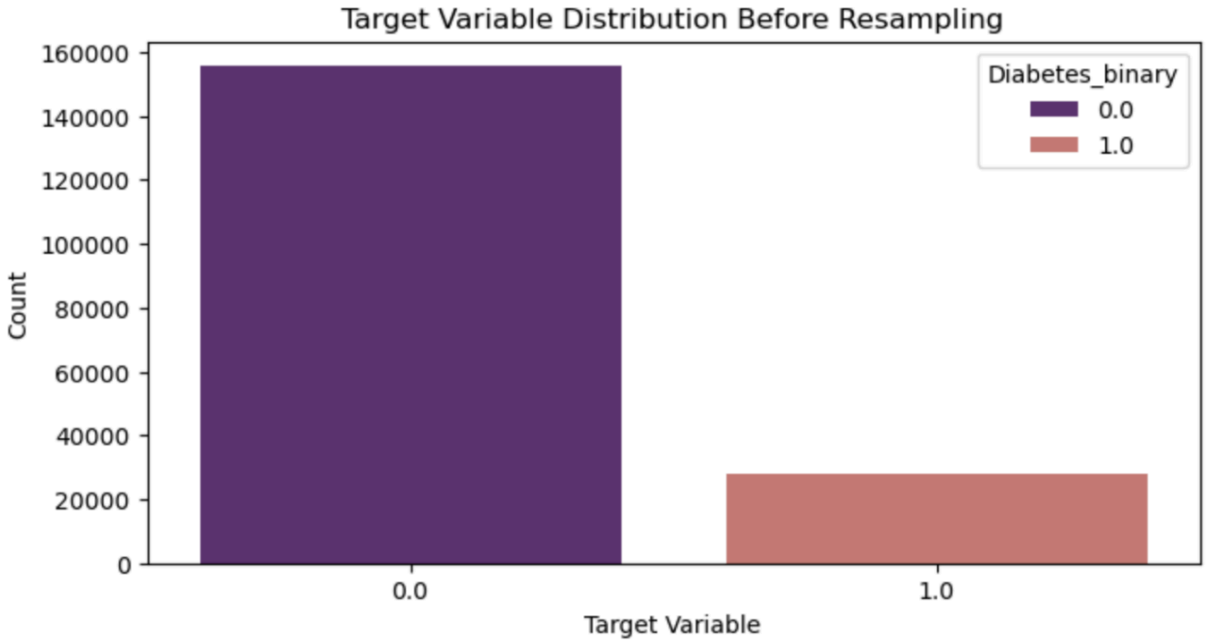


Figure 16 - Distribution of Target Variable before Resampling

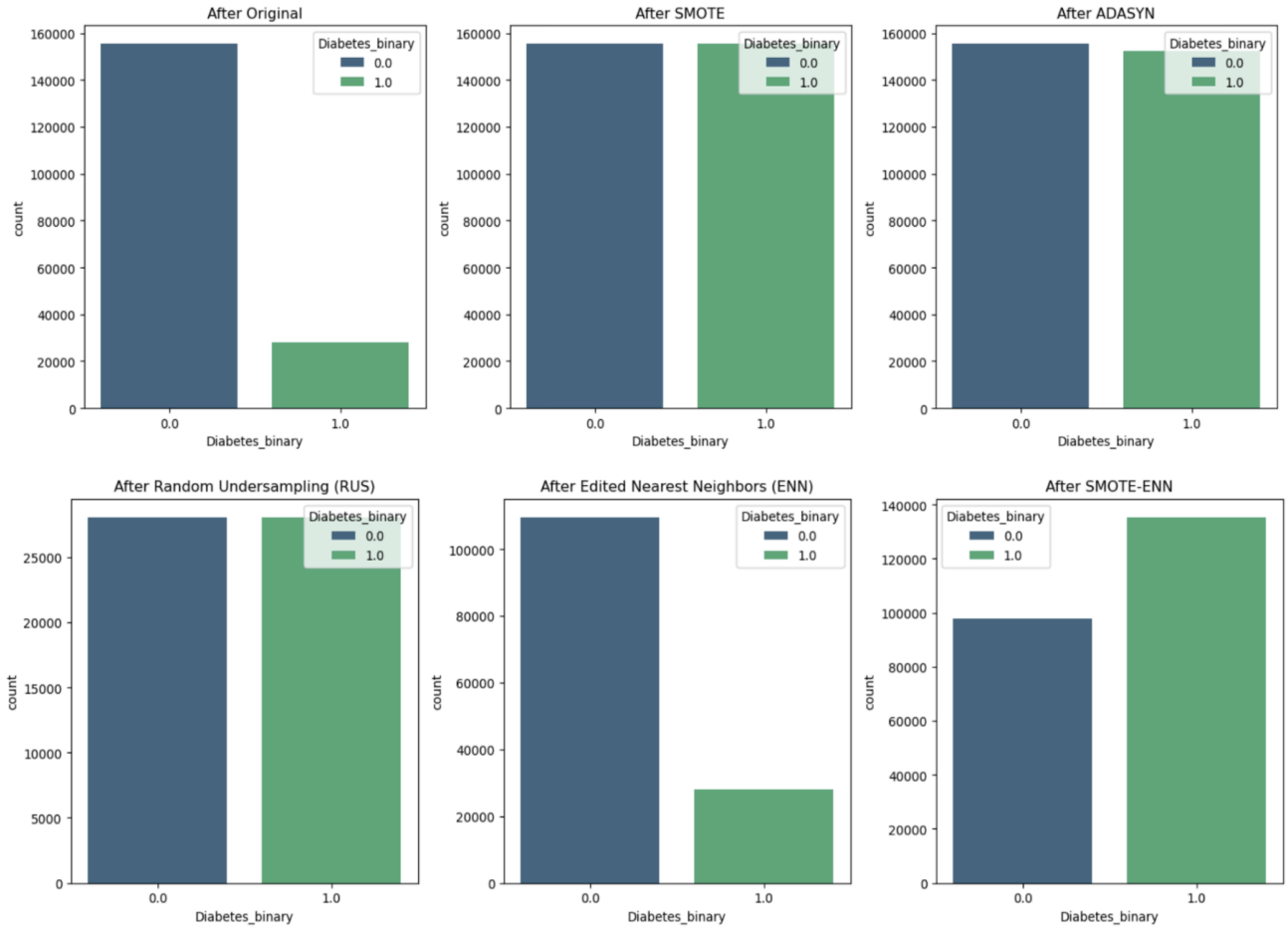


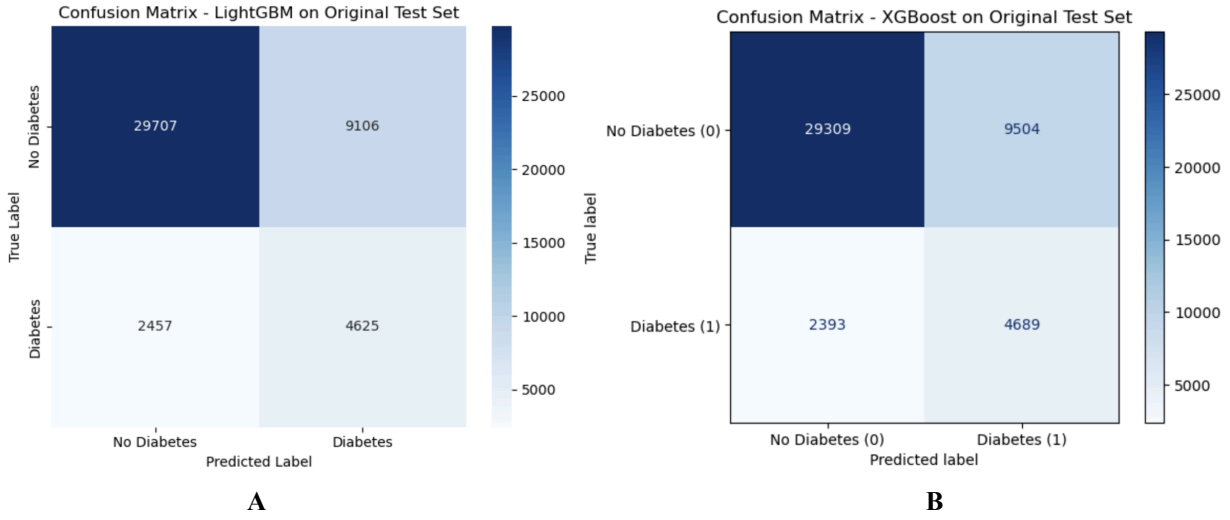
Figure 17 - Distribution of Target Class after applying each Resampling Technique

All these resampling techniques were evaluated using evaluation metrics Recall, Precision, F1 score. Since this is a medical problem, the Recall for the Diabetic class (class 1) was focused along with the model accuracy.

3.9 Model Building and Evaluation

After resampling the training data, model building was the next step. Based on the results of the resampling methods, 8 machine learning models were shortlisted for model building. The 8 models were KNN classifier, Decision Tree, Random Forest, AdaBoost, CatBoost, Gradient Boosting, XGBoost and LightGBM. These models were selected because they had a good balance between evaluation metrics Recall and Accuracy when combined with the finalized resampling methods. All these models had a testing Accuracy between 0.70 and 0.75 whereas the Recall for the Diabetic class ranging from 0.44 to 0.70. A stacking classifier was also built to see if the accuracy and recall would increase but the performance of the stacking classifier was similar to that of the individual Boosting models. Hence the stacking classifier was dropped as it was computationally taxing because of the size of the training set. The time taken to run the

stacking classifier model was very high, hence this was dropped and the focus shifted to working and improving one model to be the best performing model. From the previous resampling step, three techniques were finalised and along with that in this model building step, 4 ensemble learning models were selected. These models were AdaBoost, CatBoost, XGBoost and LightGBM. All these models had an accuracy of approximately 72% and Recall for class 1 around 68%. Fig 18 A and B are confusion plots that were made for the LightGBM and XGBoost models with the original test set.



A **B**

Figure 18 - Confusion Matrix for LightGBM(A) and XGBoost(B) models on the Original Test Set

3.10 Hyperparameter and Threshold Tuning

To enhance model performance, hyperparameter tuning was performed for 3 out of the 4 final models from model building. Hyperparameter tuning was done for AdaBoost, XGBoost and LightGBM models. Tuning was done using Optuna. Optuna is an automated hyperparameter optimization framework. It is designed to find the best hyperparameter settings for a machine learning model by efficiently exploring the search space. Unlike traditional methods like Grid Search or Random Search, Optuna uses a smarter approach, Bayesian Optimization, to choose hyperparameter combinations that are more likely to improve performance. It uses Tree-structured Parzen Estimators (TPE) to predict the best hyperparameter combinations. Along with hyperparameter tuning, threshold tuning was also done to see if the best recall and accuracy was obtained by the usual threshold of 0.5 or something lower. For the AdaBoost model, there are only 2 parameters that can be tuned: n_estimators and learning_rate. To increase the combination of the parameters a base estimator was used. Decision Tree is the default base estimator of the AdaBoost model hence, the parameters of Decision Tree model were added to Optuna and it gave the best parameters. For LightGBM and XGBoost models, parameters like n_estimators, number_of_leaves, learning_rate and max_depth were tuned using Optuna. The threshold was also tuned to see if the all time threshold of 0.5 worked well for a medical setting

like this. Optuna gave the best threshold and best parameters for each model. Based on the best parameters for each model, manual tuning was done and the accuracy and recall for class 1 was increased by 2 to 3% than the metrics obtained from just hyperparameter tuning using Optuna. Based on the final results from Manual Hyperparameter tuning, LightGBm and XGBoost models gave the best results and were chosen for building the Web-Application.

3.11 Testing

A balanced dataset was used to test the models. Testing the models gave a perspective of how the models will work in a real world scenario. AdaBoost, LightGBM and XGBoost were tested on the new balanced dataset and confusion matrices were plotted for all the 3 models. In the testing phase, AdaBoost did not perform well with the best threshold and parameters from manual tuning. LightGBM and XGBoost models gave similar results to the original test results with the best threshold and manual parameters.

3.12 Feature Importance and Model Interpretability

To enhance transparency and understandability of the machine learning models, SHapley Additive Explanations (SHAP) was used to interpret individual predictions. SHAP provides model-agnostic explanations by quantifying the contribution of each feature to a specific prediction, based on cooperative game theory. For this project, SHAP was applied to the final LightGBM and XGBoost models, which supports native SHAP computation through its tree-based structure. The TreeExplainer was used to calculate SHAP values for each input instance, and visualizations such as the waterfall plot and bar plot were generated to highlight the most influential features behind a prediction. These insights allow users to see which factors—such as BMI, General Health, or Physical Activity—contributed most positively or negatively toward a diabetes prediction. By incorporating SHAP, the model becomes more interpretable and trustworthy, especially in a healthcare context where explainability is critical for decision-making and user confidence.

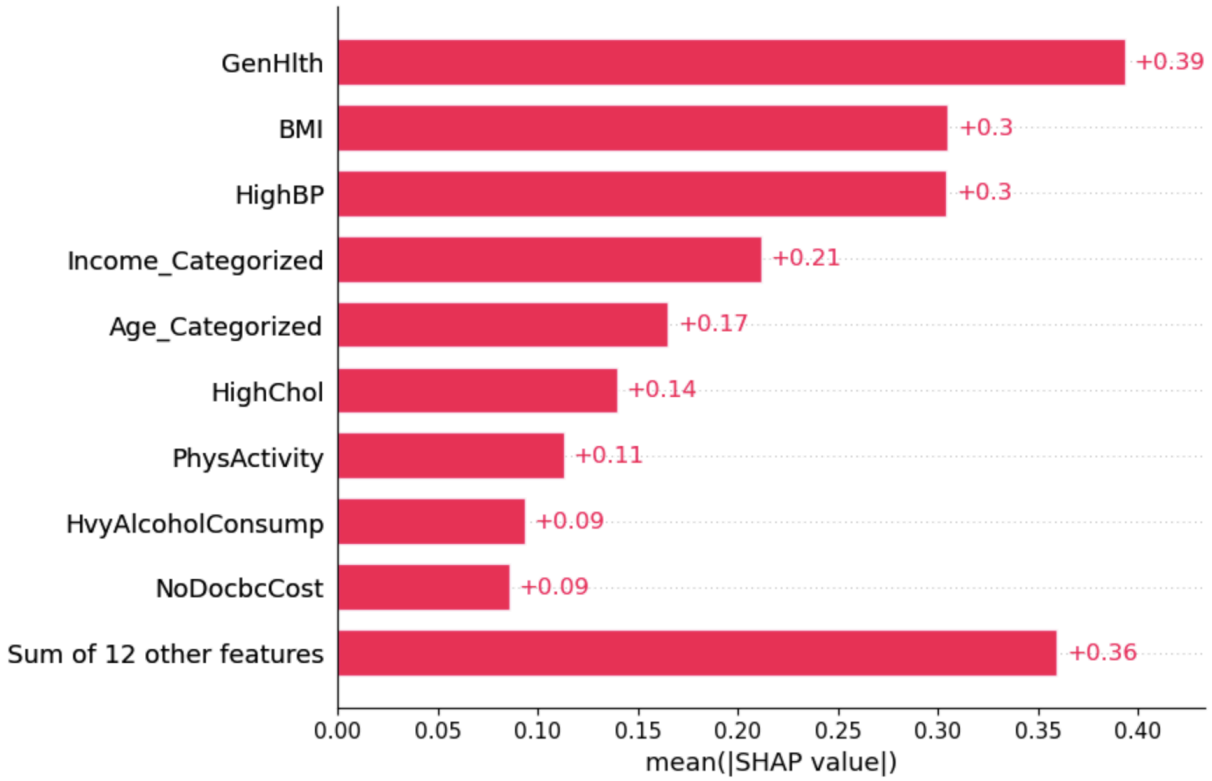


Figure 19 - SHAP Bar Plot for LightGBM model

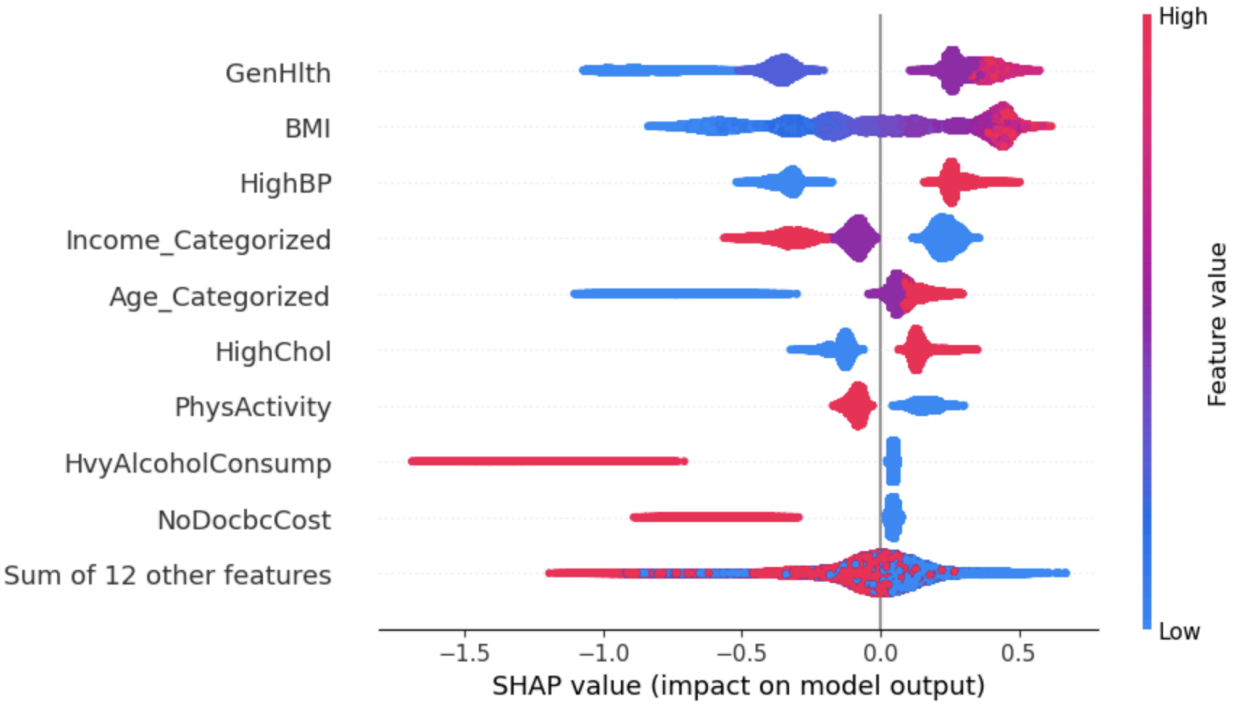


Figure 20 - Beeswarm Plot for LightGBM Model

The SHAP bar plot shown in Fig 19 highlighted the most impactful features on average, with GenHlth (General Health) emerging as the top contributor—indicating that poor self-reported health strongly increases the likelihood of diabetes. Other influential features included HighBP, BMI, Income_Categorized, and Age_Categorized, showing that both health-related and socio-economic variables played a significant role in prediction. The beeswarm plot shown in Fig 20 further revealed how high or low values of individual features impacted predictions. For example, higher values of GenHlth, HighBP, and BMI (in red) pushed predictions toward the diabetic class, while lower values (in blue) reduced risk. Similarly, lower income and older age were associated with higher SHAP values, indicating increased diabetes risk. These visualizations confirmed that the model's learned behavior aligns with real-world medical understanding, improving its interpretability and trustworthiness.

3.13 Web Application Development

To make the diabetes prediction system accessible and interactive for end users, a web application was developed using the Streamlit framework. The app was structured into multiple pages to enhance usability and organization: a landing page, an About Diabetes page, an About the Project section, a Predict Diabetes form, and an optional Insights section. The trained LightGBM and XGBoost models were saved as .pkl files and seamlessly integrated into the app to allow real-time predictions based on user input. The prediction form was carefully designed to be user-friendly, accepting input through intuitive widgets such as dropdowns, sliders, and radio buttons. Features were labeled clearly using Yes/No and descriptive category names, while internally they were converted into model-ready formats. Upon submission, the model processes the input, applies the appropriate threshold (0.563 for LightGBM or 0.555 for XGBoost), and returns the prediction result along with the probability. Additional functionality such as a Clear Form button, Back to Home navigation, and session management using st.session_state was implemented to improve the user experience. The modular, visually guided design ensures that the application is not only functional and accurate but also interpretable and accessible for non-technical users.

Section 4 – Experiments and Results

This section presents the experiments conducted and the results obtained during the model development and evaluation process. Various classification models were trained and tested on the preprocessed dataset, both with and without resampling techniques, to address the underlying class imbalance. Performance metrics such as accuracy, precision, recall, and F1-score were used to assess each model's ability to detect diabetic cases accurately. The impact of different resampling strategies, hyperparameter optimization, and threshold tuning is discussed in detail. The results highlight the trade-offs between metrics and provide insights into which model configurations were most effective for early diabetes prediction.

4.1 Resampling Methods

As mentioned in the methodology section, various resampling methods were tried with a set of 11 machine learning models. The results of each resampling technique is given as a table below.

🔹 Baseline Results:

Resampling	Model	Train Accuracy	Test Accuracy	Test F1	Test Recall (Class 0)	Test Recall (Class 1)
Baseline	Logistic Regression	0.850397	0.850528	0.239299	0.976335	0.153726
Baseline	Naive Bayes	0.755500	0.756727	0.423146	0.788018	0.583416
Baseline	KNN	0.994591	0.823009	0.271937	0.932581	0.216128
Baseline	Decision Tree	0.851176	0.852773	0.225559	0.981428	0.140191
Baseline	SVM	0.777714	0.778952	0.277370	0.869508	0.277390
Baseline	Random Forest	0.994553	0.843861	0.247743	0.965866	0.168115
Baseline	Gradient Boosting	0.853774	0.855213	0.263059	0.979113	0.168970
Baseline	AdaBoost	0.850043	0.852577	0.304911	0.968335	0.211426
Baseline	XGBoost	0.863508	0.853361	0.269431	0.975512	0.176806
Baseline	LightGBM	0.856460	0.856128	0.263798	0.980271	0.168543
Baseline	CatBoost	0.867458	0.853971	0.268979	0.976438	0.175666

Table 3 - Testing dataset results for models without any Resampling

The above table 3 shows the evaluation metrics results for all the models when no resampling has been done. As seen in the above table, test accuracy is very good and high but the recall for Class 1 i.e. the Diabetic class is very low especially in models like Logistic Regression, Decision Tree, and Gradient Boosting. While Naive Bayes and KNN offered more balanced results, they had lower overall accuracy.

Following this Undersampling techniques RUS and ENN were tried and the results for each are given in table 4 and table 5.

🔹 Random Undersampling (RUS) Results:

Resampling	Model	Train Accuracy	Test Accuracy	Test F1	Test Recall (Class 0)	Test Recall (Class 1)
Random Undersampling (RUS)	Logistic Regression	0.730519	0.714522	0.447872	0.706837	0.757088
Random Undersampling (RUS)	Naive Bayes	0.704751	0.693823	0.425793	0.685076	0.742271
Random Undersampling (RUS)	KNN	0.997062	0.663907	0.395074	0.654208	0.717624
Random Undersampling (RUS)	Decision Tree	0.716023	0.688485	0.428417	0.674967	0.763357
Random Undersampling (RUS)	SVM	0.641356	0.646541	0.357494	0.647186	0.642969
Random Undersampling (RUS)	Random Forest	0.997044	0.694738	0.433207	0.682452	0.762787
Random Undersampling (RUS)	Gradient Boosting	0.740063	0.707092	0.452446	0.691892	0.791281
Random Undersampling (RUS)	AdaBoost	0.731658	0.712387	0.449174	0.702567	0.766776
Random Undersampling (RUS)	XGBoost	0.775073	0.701972	0.446907	0.686568	0.787292
Random Undersampling (RUS)	LightGBM	0.750641	0.704085	0.450718	0.687879	0.793845
Random Undersampling (RUS)	CatBoost	0.767950	0.706025	0.451857	0.690452	0.792278

Table 4 - Testing dataset results for models with RUS resampling technique

🔹 Edited Nearest Neighbors (ENN) Results:

Resampling	Model	Train Accuracy	Test Accuracy	Test F1	Test Recall (Class 0)	Test Recall (Class 1)
Edited Nearest Neighbors (ENN)	Logistic Regression	0.857641	0.809282	0.457985	0.860274	0.526856
Edited Nearest Neighbors (ENN)	Naive Bayes	0.804033	0.724371	0.420734	0.736984	0.654509
Edited Nearest Neighbors (ENN)	KNN	0.997472	0.767055	0.414929	0.808031	0.540105
Edited Nearest Neighbors (ENN)	Decision Tree	0.849643	0.821985	0.428031	0.891758	0.435532
Edited Nearest Neighbors (ENN)	SVM	0.784246	0.738098	0.358386	0.785009	0.478273
Edited Nearest Neighbors (ENN)	Random Forest	0.997443	0.797255	0.454220	0.841599	0.551646
Edited Nearest Neighbors (ENN)	Gradient Boosting	0.865915	0.813160	0.471364	0.861637	0.544664
Edited Nearest Neighbors (ENN)	AdaBoost	0.860024	0.810524	0.461815	0.860891	0.531557
Edited Nearest Neighbors (ENN)	XGBoost	0.881220	0.806450	0.469703	0.850859	0.560479
Edited Nearest Neighbors (ENN)	LightGBM	0.870505	0.808716	0.472574	0.853560	0.560336
Edited Nearest Neighbors (ENN)	CatBoost	0.884474	0.810088	0.472078	0.856107	0.555207

Table 5 - Testing dataset results for models with ENN resampling techniques

Random Undersampling (RUS) improved class balance but often led to information loss and reduced accuracy. ENN helped refine decision boundaries by removing noise, improving precision and F1-score, particularly in models like Logistic Regression and XGBoost.

Oversampling methods SMOTE and ADASYN were tried and the results can be found in table 6 and table 7.

🔹 SMOTE Results:

Resampling	Model	Train Accuracy	Test Accuracy	Test F1	Test Recall (Class 0)	Test Recall (Class 1)
SMOTE	Logistic Regression	0.740243	0.715350	0.448497	0.707866	0.756803
SMOTE	Naive Bayes	0.713545	0.695087	0.428256	0.685770	0.746688
SMOTE	KNN	0.996807	0.733871	0.367609	0.775054	0.505770
SMOTE	Decision Tree	0.756725	0.756292	0.435072	0.782051	0.613620
SMOTE	SVM	0.644018	0.645038	0.358799	0.644254	0.649380
SMOTE	Random Forest	0.996804	0.834775	0.338018	0.935693	0.275823
SMOTE	Gradient Boosting	0.879737	0.828304	0.446162	0.896208	0.452201
SMOTE	AdaBoost	0.791988	0.749690	0.455235	0.761575	0.683858
SMOTE	XGBoost	0.914148	0.853971	0.292441	0.972528	0.197322
SMOTE	LightGBM	0.911316	0.855039	0.313345	0.970367	0.216270
SMOTE	CatBoost	0.921766	0.853971	0.282595	0.974200	0.188061

Table 6 - Testing dataset results for models with SMOTE resampling techniques

🔹 ADASYN Results:

Resampling	Model	Train Accuracy	Test Accuracy	Test F1	Test Recall (Class 0)	Test Recall (Class 1)
ADASYN	Logistic Regression	0.721564	0.705480	0.445320	0.693281	0.773045
ADASYN	Naive Bayes	0.695340	0.682013	0.425252	0.666272	0.769198
ADASYN	KNN	0.996776	0.725678	0.364911	0.763659	0.515316
ADASYN	Decision Tree	0.744872	0.685456	0.417668	0.676047	0.737569
ADASYN	SVM	0.624707	0.630112	0.343339	0.629720	0.632284
ADASYN	Random Forest	0.996773	0.834187	0.326429	0.937365	0.262715
ADASYN	Gradient Boosting	0.877773	0.830766	0.443983	0.900993	0.441801
ADASYN	AdaBoost	0.779744	0.745245	0.450667	0.756431	0.683288
ADASYN	XGBoost	0.912178	0.854646	0.290546	0.973814	0.194615
ADASYN	LightGBM	0.910405	0.854821	0.303543	0.971808	0.206867
ADASYN	CatBoost	0.921004	0.853688	0.281588	0.973969	0.187491

Table 7 - Testing dataset results for models with ADASYN resampling techniques

Among oversampling techniques, SMOTE significantly improved Class 1 recall across models such as AdaBoost, Gradient Boosting, and Random Forest, though at the cost of some precision and accuracy. ADASYN, a variation of SMOTE that targets harder examples, also enhanced recall but introduced some noise, slightly impacting accuracy.

◆ SMOTE-ENN Results:

Resampling	Model	Train Accuracy	Test Accuracy	Test F1	Test Recall (Class 0)	Test Recall (Class 1)
SMOTE-ENN	Logistic Regression	0.849561	0.649831	0.425687	0.613952	0.848554
SMOTE-ENN	Naive Bayes	0.800043	0.686633	0.422270	0.675404	0.748825
SMOTE-ENN	KNN	1.000000	0.672470	0.403587	0.663057	0.724605
SMOTE-ENN	Decision Tree	0.857968	0.677198	0.426242	0.657912	0.784015
SMOTE-ENN	SVM	0.781717	0.616603	0.382076	0.587998	0.775039
SMOTE-ENN	Random Forest	0.999996	0.769561	0.460628	0.792340	0.643396
SMOTE-ENN	Gradient Boosting	0.917153	0.733609	0.460507	0.731840	0.743411
SMOTE-ENN	AdaBoost	0.877091	0.684236	0.437116	0.663031	0.801681
SMOTE-ENN	XGBoost	0.948436	0.791415	0.471368	0.824519	0.608064
SMOTE-ENN	LightGBM	0.943113	0.790282	0.472747	0.821972	0.614760
SMOTE-ENN	CatBoost	0.954877	0.796165	0.472393	0.832184	0.596666

Table 8 - Testing dataset results for models with SMOTE-ENN resampling techniques

Table 8 shows results of combining SMOTE and ENN which offered a balanced improvement in recall and robustness, benefiting models such as XGBoost and LightGBM. Overall, tree-based models handled resampling more effectively than simpler models, and SMOTE, SMOTE-ENN emerged as the two best choices when high recall for the minority class is a priority.

Based on all the above resampling techniques, SMOTE and SMOTE-ENN were selected for further experiment. Based on the evaluation metrics of the 11 models, 7 models were chosen for the next level of experimentation with the selected resampling techniques.

4.2 Final Resampling and Model Building

The 7 models chosen were Decision Tree, Random Forest, AdaBoost, CatBoost, Gradient Boost, XGBoost and LightGBM. Each resampling technique was applied for the training dataset and each model was run individually. The results of each resampling method is tabulated below.

SMOTE Resampling Technique

Model	Accuracy	Recall 0	Recall 1	F1 score	Precision
Decision Tree	0.7129	0.7620	0.4435	0.2538	0.3228
Random Forest	0.7574	0.8070	0.4855	0.3817	0.3145
AdaBoost	0.7068	0.7054	0.7143	0.4292	0.3067
CatBoost	0.7262	0.7518	0.5857	0.3976	0.3010
GradientBoost	0.7216	0.7375	0.6344	0.4121	0.3060
XGBoost	0.7070	0.7124	0.6774	0.4164	0.3005
LightGBM	0.7073	0.7137	0.6723	0.4248	0.2999

Table 9 - Results for SMOTE resampling method

Table 9 shows that using the SMOTE resampling method for the selected models, AdaBoost, CatBoost, XGBoost and LightGBM had a fair enough Accuracy and Recall for class 1

SMOTE-ENN Resampling Technique

Model	Accuracy	Recall 0	Recall 1	F1 score	Precision
Decision Tree	0.66	0.67	0.66	0.38	0.26
Random Forest	0.67	0.66	0.75	0.42	0.29
AdaBoost	0.60	0.56	0.86	0.40	0.26
CatBoost	0.65	0.63	0.76	0.40	0.27
GradientBoost	0.64	0.62	0.74	0.41	0.27
XGBoost	0.62	0.59	0.82	0.40	0.26
LightGBM	0.63	0.59	0.81	0.40	0.27

Table 10 - Results for SMOTE-ENN resampling method

After resampling with SMOTE-ENN, the count of the majority class (class 0) became less than class 1(can be seen in Fig 17). The technique works that way where it adds synthetic points for the minority class and reduces the count of the majority class. That is why the recall of 1 is higher than recall of 0 in SMOTE-ENN which can be seen from tables 9 and 10. But the accuracy and precision was lower than that of SMOTE and therefore, more searching for resampling methods gave way to SMOTE-TOMEK. With further research on combined resampling techniques, SMOTE with TOMEK links was found. It worked well and gave slightly better results than SMOTE-ENN. It balances the disadvantage of SMOTE-ENN which adds noise to the training data. In SMOTE-TOMEK, SMOTE algorithm adds the synthetic data points whereas the TOMEK links balance out the noise and eliminates them.

SMOTE-TOMEK Resampling Technique

Model	**Accuracy**	**Recall 0**	**Recall 1**	**F1 score**	**Precision**
Decision Tree	**0.7107**	**0.7568**	**0.4581**	**0.3281**	**0.2558**
Random Forest	**0.7555**	**0.8019**	**0.5010**	**0.3874**	**0.3157**
AdaBoost	**0.7055**	**0.7022**	**0.7235**	**0.4312**	**0.3071**
CatBoost	**0.7219**	**0.7440**	**0.6011**	**0.4002**	**0.2999**
GradientBoost	**0.7161**	**0.7280**	**0.6509**	**0.4144**	**0.3040**

Model	Accuracy	Recall 0	Recall 1	F1 score	Precision
XGBoost	**0.7058**	**0.7100**	**0.6829**	**0.4173**	**0.3005**
LightGBM	**0.7029**	**0.7076**	**0.6772**	**0.4130**	**0.2971**

Table 11 - Results for SMOTE-TOMEK resampling method

SMOTE-Tomek was selected as the resampling strategy because it offers a good balance between oversampling and noise reduction. Unlike SMOTE alone, it helps clean overlapping regions between classes, and compared to SMOTE-ENN, it is less aggressive in removing borderline data. During testing, SMOTE-TOMEK yielded slightly better performance metrics (shown in table 11), especially in terms of recall and generalization on the test set, making it a more stable and effective choice for this problem.
So based on the above results, **AdaBoost, LightGBM and XGBoost** had a good enough accuracy along with a good balance between recall 0 and recall 1. These 3 models were selected for further hyperparameter tuning and web-application development.

4.3 Hyperparameter Tuning

Hyperparameter tuning was done to further improve the accuracy and recall. Optuna was used to find the best threshold and best parameters for each model. Using the best parameters as a baseline manual hyperparameter tuning was done for each model.

4.3.1 Tuning using Optuna

AdaBoost Model - with custom Decision Tree as Base Estimator

Metric	Value
Accuracy	0.7524
Precision	0.7600
Recall	0.7429
F1 Score	0.7513

Table 12 - Training Performance of AdaBoost

Metric	Value
Accuracy	0.7307

Precision	0.3109
Recall_0	0.7522
Recall_1	0.6128
F1 Score	0.4125

Table 13 - Test Performance of AdaBoost

```
Classification Report for AdaBoost:
               precision    recall  f1-score   support

           0       0.91      0.75      0.83     38813
           1       0.31      0.61      0.41      7082

    accuracy                           0.73     45895
   macro avg       0.61      0.68      0.62     45895
weighted avg       0.82      0.73      0.76     45895
```

Tables 12 and 13 show the training performance and test performance of the AdaBoost model. After tuning with Optuna, the Adaboost model had a training accuracy of 75% and test accuracy of 73%. Recall for class 1 was 61%.

LightGBM model

Metric	**Value**
Accuracy	0.7130
Precision	0.7621
Recall	0.6254
F1 Score	0.6870

Table 14 - Training Performance of LightGBM

Metric	**Value**
Accuracy	0.7697
Precision	0.3556

Recall_0	0.7995
Recall_1	0.6062
F1 Score	0.4482

Table 15 - Test Performance of LightGBM

```
Classification Report for LightGBM:
              precision    recall  f1-score   support

           0       0.92      0.80      0.85     38813
           1       0.36      0.61      0.45      7082

    accuracy                           0.77     45895
   macro avg       0.64      0.70      0.65     45895
weighted avg       0.83      0.77      0.79     45895
```

Tables 14 and 15 show the training performance and test performance of the LightGBM model. After tuning with Optuna, the LightGBM model had a training accuracy of 71% and test accuracy of 76% but the Recall for class 1 was 60%.

XGBoost Model

Metric	**Value**
Accuracy	0.7222
Precision	0.7634
Recall	0.7496
F1 Score	0.7019

Table 16 - Training Performance of XGBoost

Metric	**Value**
Accuracy	0.7645
Precision	0.3498
Recall_0	0.7921
Recall_1	0.6131

F1 Score	0.4455

Table 17 - Test Performance of XGBoost

Classification Report for XGBoost:

	precision	recall	f1-score	support
0	0.92	0.79	0.85	38813
1	0.35	0.61	0.45	7082
accuracy			0.76	45895
macro avg	0.63	0.70	0.65	45895
weighted avg	0.83	0.76	0.79	45895

Tables 16 and 17 show the training performance and test performance of the XGBoost model. After tuning with Optuna, the XGBoost model had a training accuracy of 72% and test accuracy of 76%. Recall for class 1 was 61%.

4.3.2 Manual Tuning Based on Best Parameters from Optuna

AdaBoost

Run	Accuracy	Recall 0	Recall 1	F1 Score	Precision	Best Threshold
Run 1	0.7139	0.7192	0.6853	0.4250	0.3081	0.503
Run 2	0.7314	0.7497	0.6310	0.4203	0.3151	0.529
Run 3	0.7120	0.7169	0.6851	0.4233	0.3063	0.504
Run 4	0.7351	0.7552	0.6247	0.4212	0.3177	0.512
Run 5	0.7217	0.7315	0.6685	0.4257	0.3123	0.506

Table 18 - Manual parameter tuning results for AdaBoost Test data

From the above runs shown in table 18, the best parameters gave the following results -

Threshold: 0.506 | F1 Score: 0.4257 | Recall (1): 0.6685 | Precision: 0.3123 | Accuracy : 0.7217

LightGBM

Run	Accuracy	Recall 0	Recall 1	F1 Score	Precision	Best Threshold
Run 1	0.7691	0.7989	0.6058	0.4474	0.3546	0.583
Run 2	0.7710	0.8034	0.5936	0.4445	0.3552	0.591

Run 3	0.7499	0.7732	0.6226	0.4345	0.3337	0.571
Run 4	0.7351	0.7523	0.6409	0.4275	0.3207	0.538
Run 5	0.7481	0.7654	0.6531	0.4444	0.3368	0.563

Table 19 - Manual parameter tuning results for LightGBM Test Data

From the above runs shown in table 19, the best parameters gave the following results -

Threshold: 0.563 | F1 Score: 0.4444 | Recall (1): 0.6531 | Precision: 0.3368 | Accuracy : 0.7481

XGBoost

Run	Accuracy	Recall 0	Recall 1	F1 Score	Precision	Best Threshold
Run 1	0.7566	0.7804	0.6262	0.4426	0.3423	0.579
Run 2	0.7354	0.7534	0.6370	0.4262	0.3203	0.542
Run 3	0.7426	0.7687	0.6005	0.4184	0.3212	0.542
Run 4	0.7186	0.7507	0.5426	0.3731	0.2842	0.445
Run 5	0.7408	0.7551	0.6621	0.4408	0.3304	0.555

Table 20 - Manual parameter tuning results for XGBoost Test Data

From the above runs shown in table 20, the best parameters gave the following results -

Threshold: 0.555 | F1 Score: 0.4408 | Recall (1): 0.6621 | Precision: 0.3304 | Accuracy : 0.7408

Consolidated results

Model	Accuracy	Recall 0	Recall 1	F1 Score	Precision
AdaBoost	0.7137	0.7189	0.6854	0.4249	0.3079
LightGBM	0.7481	0.7654	0.6531	0.4444	0.3368
XGBoost	0.7408	0.7551	0.6621	0.4408	0.3304

Table 21 - Final best evaluation metrics obtained for each model

The consolidated results in table 21 show the final results of both manual parameter tuning and tuning using Optuna. For each model the accuracy was maintained and the recall for class 1 was increased to a maximum of 66% from 60%.

4.4 Testing

To ensure fair model evaluation, testing was conducted using a **balanced test dataset**, where both classes—diabetic and non-diabetic—were equally represented. This allowed for a more unbiased assessment of the model's capability to distinguish between the two outcomes without being influenced by class imbalance. Additionally, the decision threshold was fine-tuned using F1-score optimization to improve sensitivity to both classes. Testing on a balanced dataset enabled a clearer interpretation of model performance, particularly for recall of the diabetic class, and ensured that evaluation results reflected true predictive power rather than skewed class proportions.

During Training, the best threshold obtained was 0.506, but with this threshold, the testing dataset performed poorly and the threshold had to be reduced to 0.414 to obtain the following results. Even then the accuracy remained low. For the LightGBM model, the best threshold during Training was 0.563 and the Test threshold was also 0.56. For the XGBoost model, Training Threshold - 0.575 and Testing Threshold was also 0.57.

Model	Accuracy	Recall 1	F1 score
AdaBoost	0.5507	0.6831	0.6071
LightGBM	0.6677	0.6929	0.6794
XGBoost	0.6601	0.7242	0.6841

Table 22 - Testing metrics for the new Balanced Dataset

Table 22 shows the results of model performance on the balanced test set. Thus, based on all the experimentation and tuning, the models chosen for the final web-application development were **LightGBM** and **XGBoost** as they provided the best recall for class 1 and also had a good accuracy during training and testing with the original dataset and the new balanced dataset.

4.5 Web Application Development

The web application developed for this project consists of 4 elements. Firstly the landing page is the first page a user is navigated to (Fig 21). On this page there are navigation tiles at the bottom to guide the user to subsequent pages (Fig 22). The first tile navigates the user to an About Diabetes page (Fig 23, 24, 25). This page gives the user information on what Diabetes is and some current demographics and statistics. The main aim of this page is to bring the user up to speed on Diabetes and it's impact on the world population today. Users can either navigate back to the Home page or go on to the prediction page (Fig 28, 29, 30) where they can fill out a form and predict their Diabetes status. Additionally, a project description page (Fig 26, 27) shows the

details of this project. All the following are the images of each web page and finally sample results and recommendations have also been shown in Fig 31, 32, 33, 34, 35.

4.5.1 Landing Page

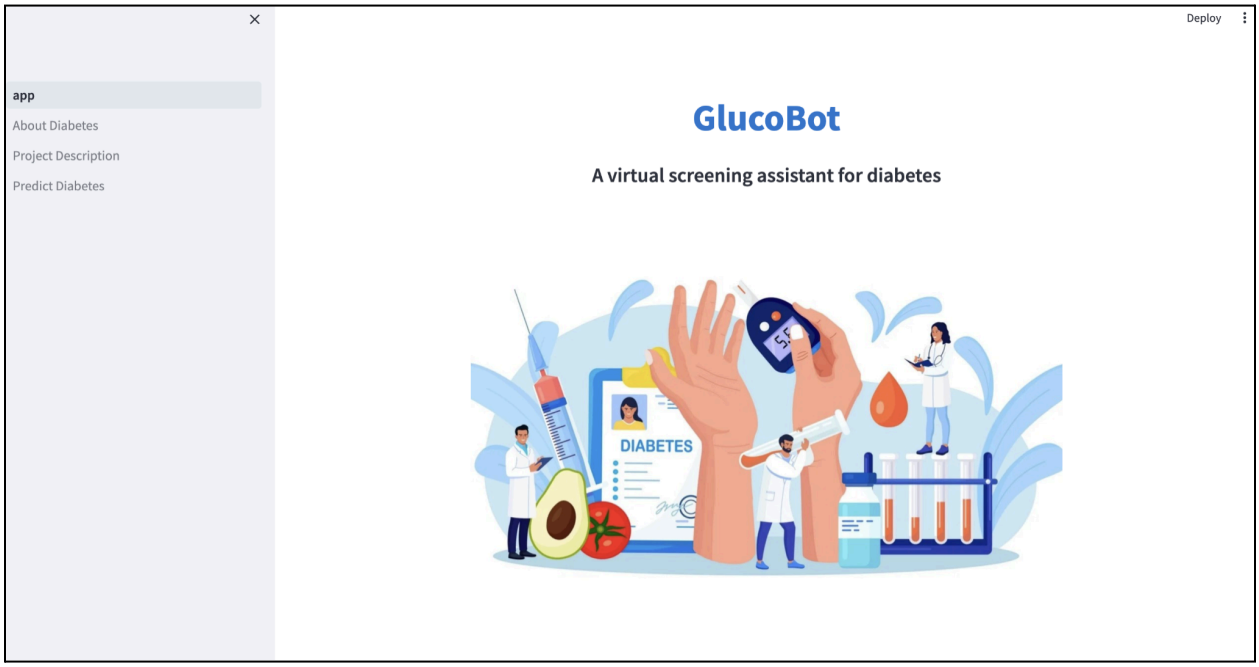


Figure 21 - Screenshot of Landing Page

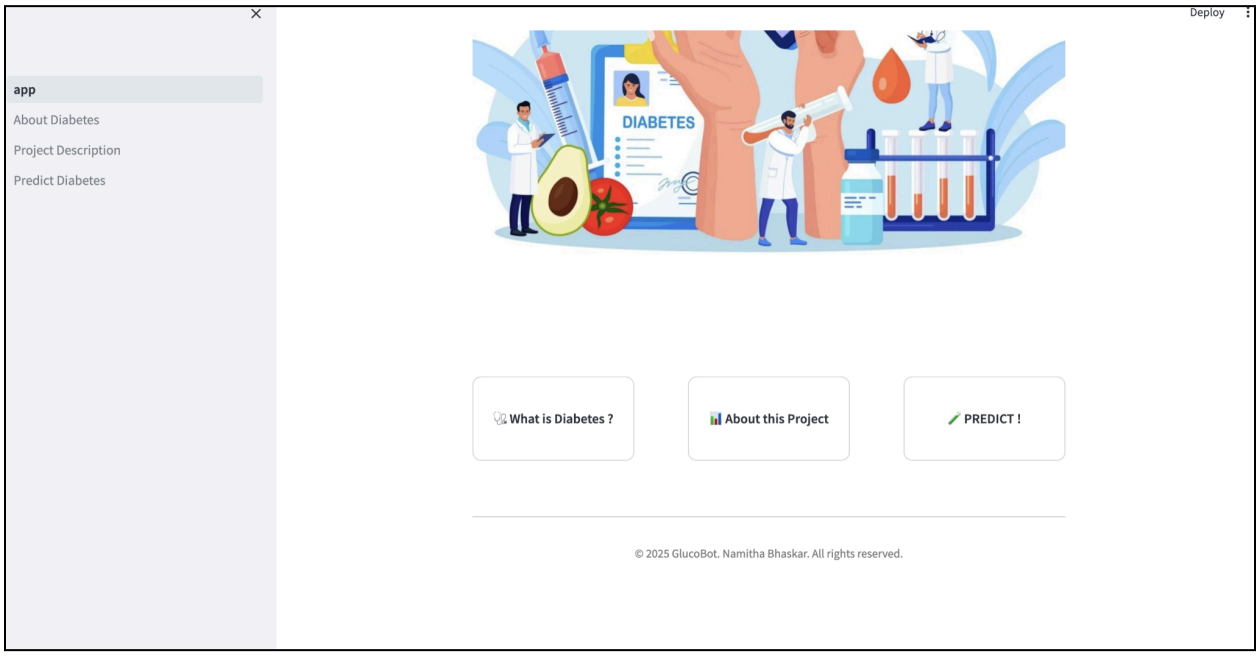


Figure 22 - Screenshot of navigation tiles in the Landing Page

4.5.2 About Diabetes Page

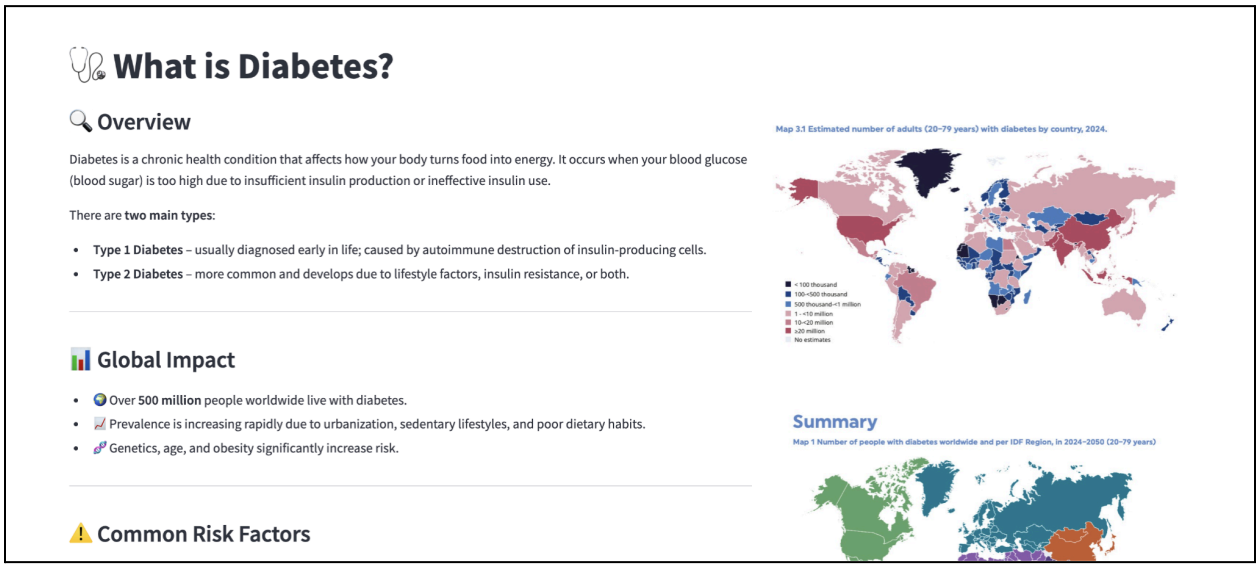


Figure 23 - Screenshot of About Diabetes Page

Figure 24 - Continuation of the About Diabetes Page

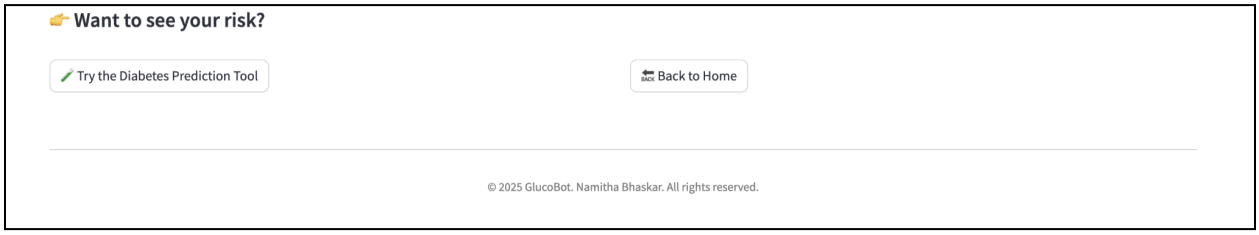


Figure 25 - Navigation buttons in the About Diabetes Page

4.5.3 Project Description

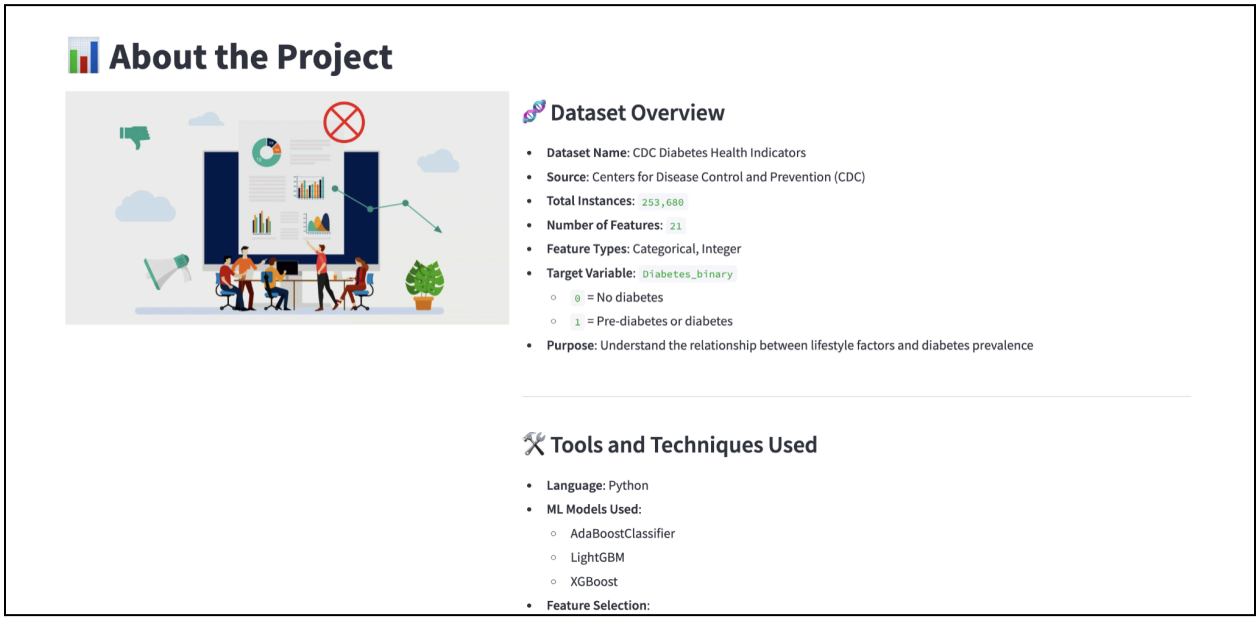


Figure 26 - Screenshot of Project Description Page

Figure 27 - Continuation of the Project Description Page

4.5.4 Predict Diabetes Page

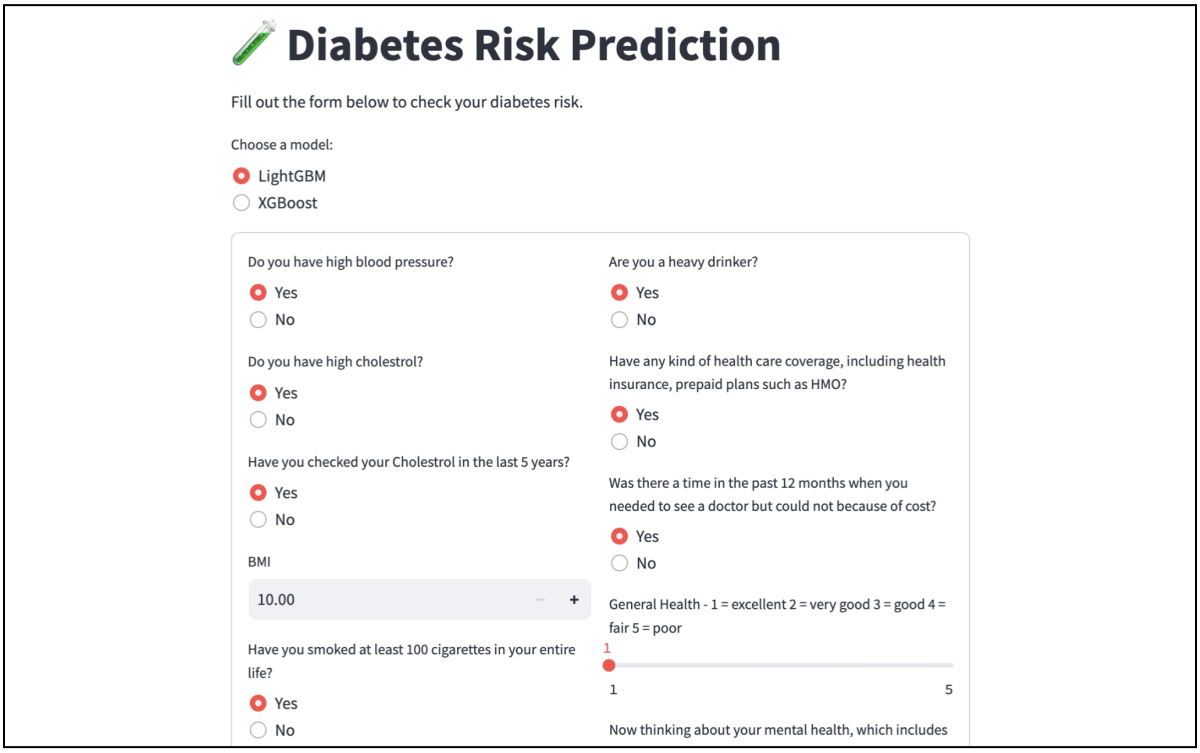


Figure 28 - Screenshot of Diabetes Prediction Form

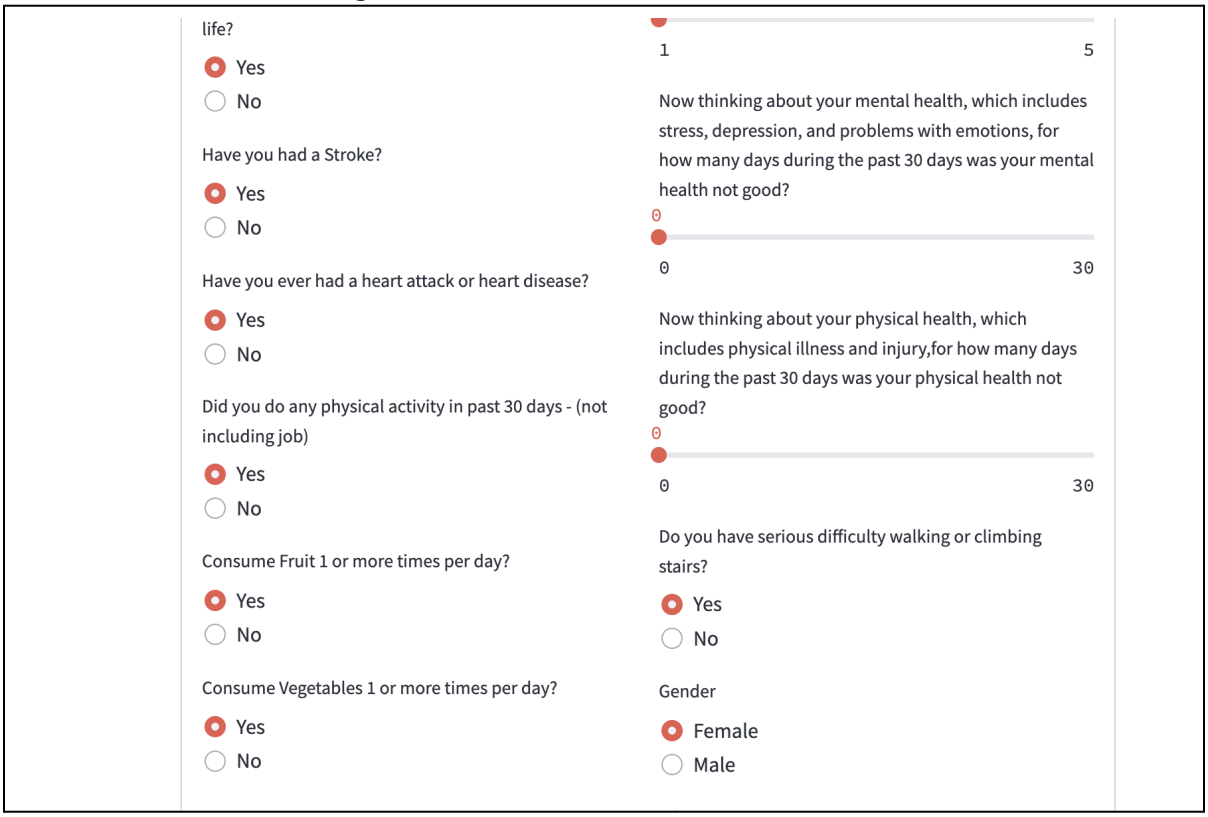


Figure 29 - Continuation of the Diabetes Prediction Form

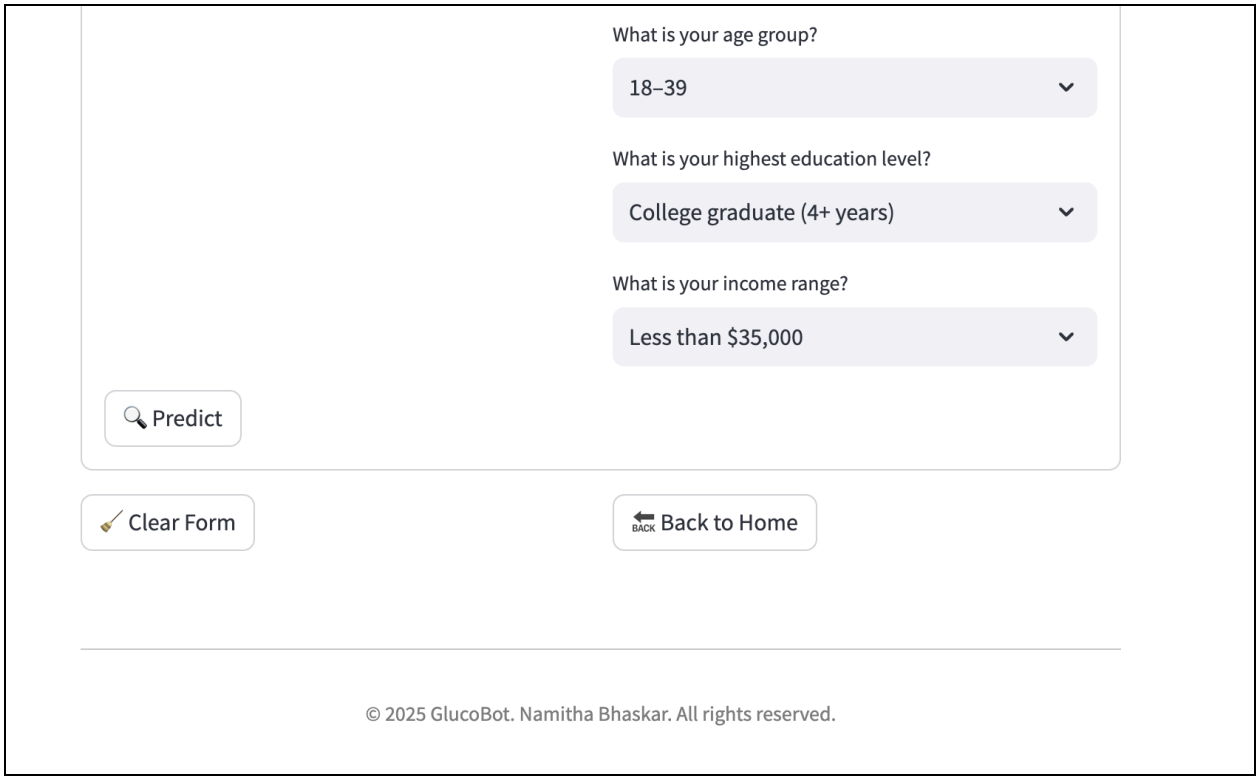


Figure 30 - Diabetes Prediction Form with Predict and Navigation Buttons

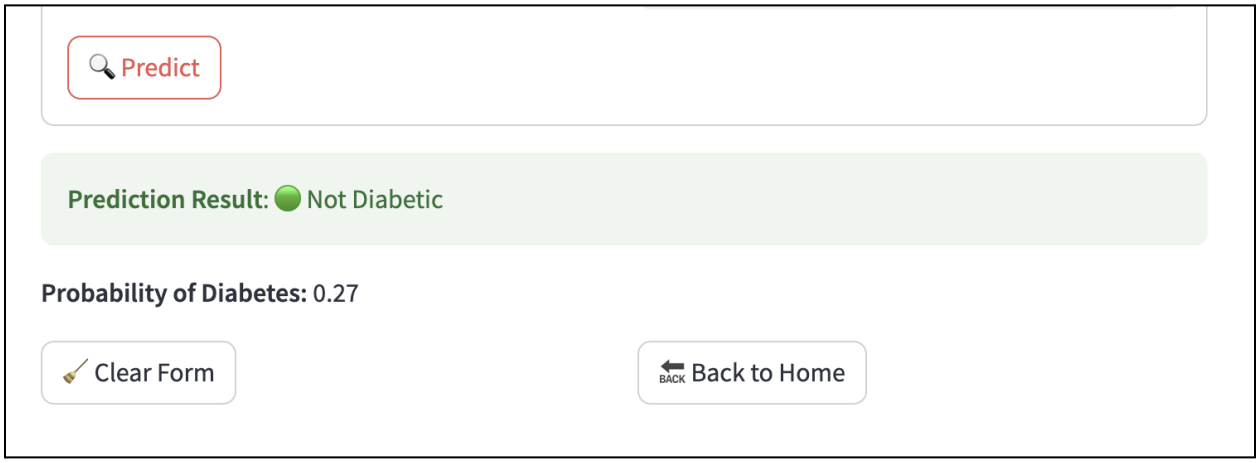


Figure 31 - Example of a Prediction showing Not Diabetic result and Probability

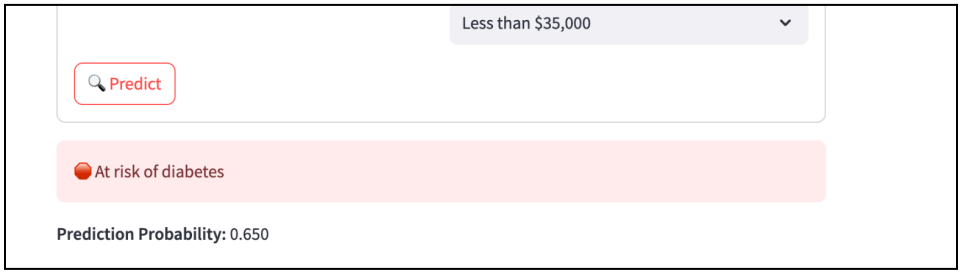


Figure 32 - Example of a Prediction showing Diabetic result and Probability

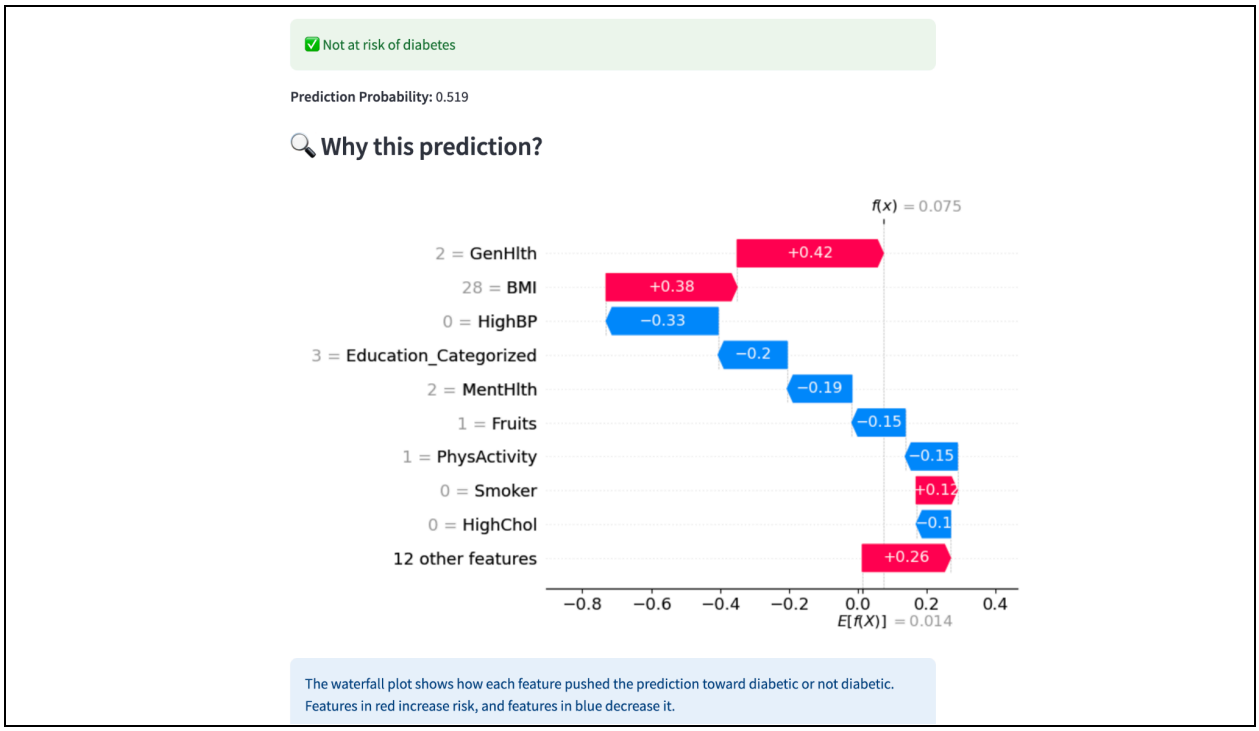


Figure 33 - SHAP Waterfall Plot for user given data

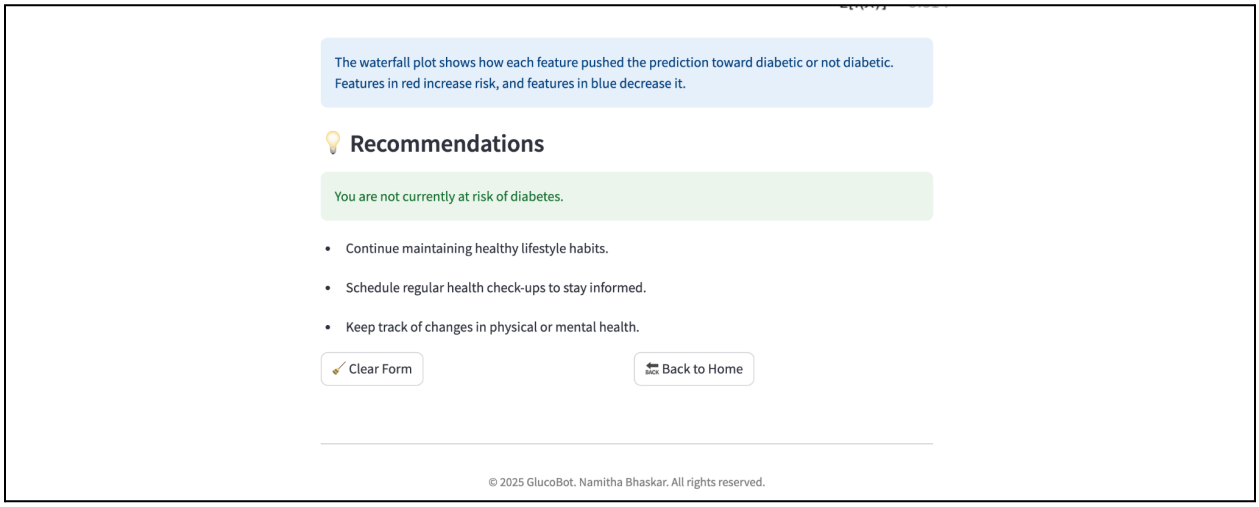


Figure 34 - Recommendations for Not Diabetic Prediction

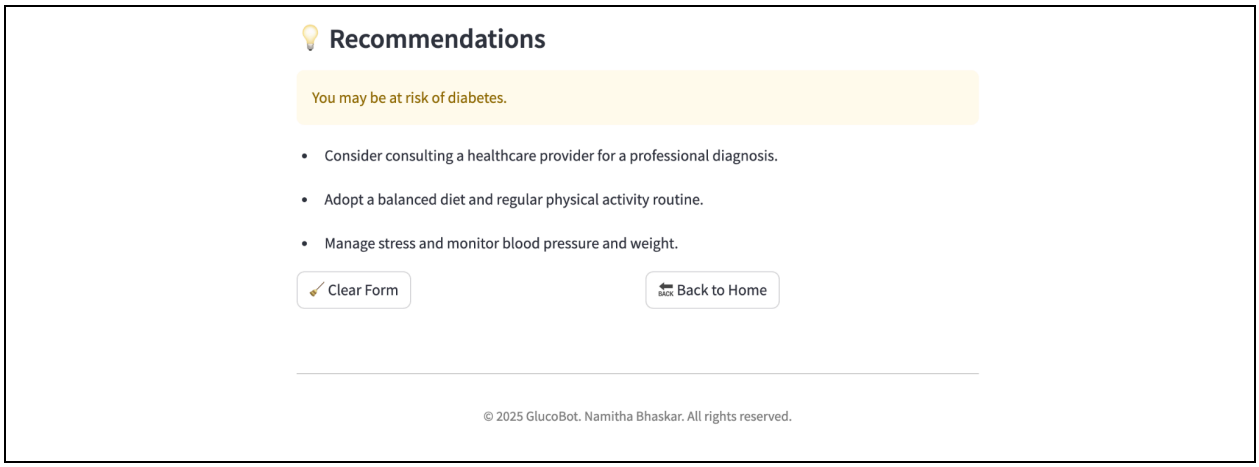


Figure 35 - Recommendations for Diabetic Prediction

Section 5 – Conclusions and Future Work

This project set out to identify the most effective machine learning model for early diabetes prediction among adults, with the aim of supporting timely interventions and improved healthcare decision-making. Through extensive experimentation with various classifiers and resampling techniques, tree-based ensemble models, particularly XGBoost and LightGBM, delivered strong performance when evaluated on a balanced test set. Key performance metrics such as recall, F1-score, and class-wise precision highlighted these models' ability to accurately identify both diabetic and non-diabetic individuals. Additionally, interpretability was addressed through SHAP analysis, which confirmed that features like general health, high blood pressure, BMI, age, and income had the highest predictive impact—aligning with known medical insights.

A major highlight of this work was the development of GlucoBot, a user-friendly Streamlit web application that allows users to input health and lifestyle data and receive a real-time diabetes risk prediction. The app incorporates best-performing models and includes a clean, guided form interface, making it accessible to non-technical users.

While the project successfully demonstrated the potential of machine learning models for diabetes prediction, there remains room for improvement. One key limitation was the quality and scope of the dataset used; it was based on self-reported survey data, which may contain inaccuracies or lack clinical depth. Future work could benefit from a more comprehensive dataset, ideally one that includes clinical test results (e.g., blood glucose levels, HbA1c), genetic factors, medication history, and longitudinal health records. Incorporating additional predictors such as dietary patterns, stress levels, family history, and healthcare access over time could also enhance model performance.

Despite extensive tuning and resampling, the best models achieved a test accuracy of approximately 74% and recall for the diabetic class in the range of 65–68%, suggesting that

further refinement is possible. In future iterations, advanced evaluation metrics such as AUC-ROC, precision-recall curves, and calibration plots could provide deeper insights into model performance, particularly in imbalanced or clinical settings. Exploring deep learning approaches, or neural-symbolic methods could also help boost predictive performance. Lastly, conducting real-world testing with user feedback and integrating the system into clinical workflows or mobile health platforms would make the tool more actionable and impactful in practice.

References

[1] A. T. Kharroubi, “Diabetes mellitus: The epidemic of the century,” *WJD*, vol. 6, no. 6, p. 850, 2015, doi: 10.4239/wjd.v6.i6.850.

[2] World Health Organization . 2024. A.Loke . Newsroom Blog on Diabetes (Nov, 2024) https://www.who.int/news-room/fact-sheets/detail/diabetes

[3] International Diabetes Federation . 2025 . Facts and Figures https://idf.org/about-diabetes/diabetes-facts-figures/

[4] B. Paul and B. Karn, “Diabetes Mellitus Prediction using Hybrid Artificial Neural Network,” in *2021 IEEE Bombay Section Signature Conference (IBSSC)*, Gwalior, India: IEEE, Nov. 2021, pp. 1–5. doi: 10.1109/IBSSC53889.2021.9673397.

[5] M. A. R. Refat, Md. A. Amin, C. Kaushal, M. N. Yeasmin, and M. K. Islam, “A Comparative Analysis of Early Stage Diabetes Prediction using Machine Learning and Deep Learning Approach,” in *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, Solan, India: IEEE, Oct. 2021, pp. 654–659. doi: 10.1109/ISPCC53510.2021.9609364.

[6] S. M. Ganie, P. K. D. Pramanik, M. Bashir Malik, S. Mallik, and H. Qin, “An ensemble learning approach for diabetes prediction using boosting techniques,” *Front. Genet.*, vol. 14, p. 1252159, Oct. 2023, doi: 10.3389/fgene.2023.1252159.

[7] I. Tasin, T. U. Nabil, S. Islam, and R. Khan, “Diabetes prediction using machine learning and explainable AI techniques,” *Healthcare Tech Letters*, vol. 10, no. 1–2, pp. 1–10, Feb. 2023, doi: 10.1049/htl2.12039.

[8] A. R. Thezo, “Diabetes Prediction Using Medical Variables: Analysis & Data Visualization,” *ESL*, vol. 3, no. 01, pp. 24–28, Feb. 2024, doi: 10.56741/esl.v3i01.472.

[9] K. Alpan, B. Arman, and K. Dimililer, “Performance Evaluation and Comparison of Machine Learning Algorithms in Classification of CDC Diabetes Health Indicators Dataset by WEKA,” in *2024 8th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkiye: IEEE, Nov. 2024, pp. 1–5. doi: 10.1109/ISMSIT63511.2024.10757280.

[10] O. C. Ekwebene *et al.*, “The burden of diabetes in America: a data-driven analysis using power BI,” *Int J Res Med Sci*, vol. 12, no. 2, pp. 392–396, Jan. 2024, doi: 10.18203/2320-6012.ijrms20240203.

[11] S. S. J. Qi and S. Nagalingham, “Business Intelligence Data Visualization for Diabetes Health Prediction,” *IJACSA*, vol. 14, no. 1, 2023, doi: 10.14569/IJACSA.2023.0140190.

[12] S. Sivaranjani, S. Ananya, J. Aravinth, and R. Karthika, “Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction,” in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India: IEEE, Mar. 2021, pp. 141–146. doi: 10.1109/ICACCS51430.2021.9441935.

[13] M. Lugner, A. Rawshani, E. Helleryd, and B. Eliasson, “Identifying top ten predictors of type 2 diabetes through machine learning analysis of UK Biobank data,” *Sci Rep*, vol. 14, no. 1, p. 2102, Jan. 2024, doi: 10.1038/s41598-024-52023-5.

[14] S. Mahajan, P. K. Sarangi, A. K. Sahoo, and M. Rohra, “Diabetes Mellitus Prediction using Supervised Machine Learning Techniques,” in *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, Gharuan, India: IEEE, May 2023, pp. 587–592. doi: 10.1109/InCACCT57535.2023.10141734.

[15] F. I. E. Sari, F. W. Edlim, F. A. Ramadhan, Muhtadin, and D. A. Navastara, “Performance Analysis of Resampling and Ensemble Learning Methods on Diabetes Detection as Imbalanced Dataset,” in *2022 Fifth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, Surabaya, Indonesia: IEEE, Sep. 2022, pp. 1–5. doi: 10.1109/ICVEE57061.2022.9930467.

[16] B. Thuraka, V. Pasupuleti, C. S. Kodete, R. S. Chigurupati, N. S. K. M. K. Tirumanadham, and V. Shariff, “Enhancing Diabetes Prediction using Hybrid Feature Selection and Ensemble Learning with AdaBoost,” in *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Kirtipur, Nepal: IEEE, Oct. 2024, pp. 1132–1139. doi: 10.1109/I-SMAC61858.2024.10714776.

[17] S. Kumari, Ranganayaki. V. C, S. K, K. Selvi, T. A. Mohanaprakash, and C. Tamilselvi, “Optuna-Optimized Machine Learning Technique for Accurate Diabetes Prediction and Classification,” in *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, Kaski, Nepal: IEEE, Oct. 2024, pp. 1478–1485. doi: 10.1109/ICSES63445.2024.10763036.

[18] S. D. Kurniawan, P. Purwono, A. Ma’Arif, and I. Suwarno, “Diabetes Classification Problem with CatBoost Method and Optuna Gradient Boosting Optimization,” in *2023 6th International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia: IEEE, Nov. 2023, pp. 361–366. doi: 10.1109/ICOIACT59844.2023.10455940.

[19] M. Zeng, B. Zou, F. Wei, X. Liu, and L. Wang, “Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data,” in *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, Chongqing, China: IEEE, May 2016, pp. 225–228. doi: 10.1109/ICOACS.2016.7563084.

[20] M. Nauman, A. S. Almadhor, M. Albekairi, A. R. Ansari, M. A. B. Fayyaz, and R. Nawaz, “The Role of Big Data Analytics in Revolutionizing Diabetes Management and Healthcare Decision-Making,” *IEEE Access*, vol. 13, pp. 10767–10785, 2025, doi: 10.1109/ACCESS.2025.3526456.