Diabetes Risk Prediction

Multi Class Classification to identify Diabetes Status

Priyanka Mandloi
Information Technology
Rensselaer Polytechnic Institute
Troy, New York, United State
mandlp@rpi.edu

EXECUTIVE SUMMARY

This project examines the **Diabetes Health Indicators Dataset (link)** from **Kaggle,** sourced from the Behavioral Risk Factor Surveillance System (BRFSS) 2015 to predict the risk of diabetes. The goal is to classify individuals into one of three categories based on their health indicators -

- **Class 0**: No diabetes or diabetes only during pregnancy
- **Class 1**: Prediabetes
- **Class 2**: Diabetes

The predictive task involves building a classification model using a variety of health-related features such as BMI, blood pressure, cholesterol levels, physical health, and more. The objective is to assess the likelihood of developing diabetes based on these indicators. Accurate prediction models can provide insights into key health factors affecting diabetes risk and help prioritize patient care.

The data I am using for this analysis is diabetes_012_health_indicators_BRFSS2015.csv. The dataset consists of 253,680 observations and 21 feature variables and class label (Diabetes_012). Importantly, there are no missing values, which simplifies preprocessing. This large-scale dataset provides an opportunity to understand the underlying patterns that contribute to diabetes onset.

Preprocessing the dataset began with checking for missing values, and since none were found, imputation was unnecessary. Next, numerical features such as 'BMI', 'GenHlth', 'MentHlth', 'PhysHlth', 'Age', and 'Income' require scaling to ensure uniformity. This step ensures these features don't dominate models sensitive to scale. Given the slight imbalance in the class distribution, class imbalance will be handled during model training by applying the SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes effectively.

Additionally, a thorough feature correlation analysis was performed using **Correlation Matrix Heatmap** and a **Chi-Squared** to identify most relevant features. Based on this analysis, the following features emerged as the most important for predicting the target variable:

Feature	Description
Diabetes_012	Class label: 0 = no diabetes, Class label: 1 = prediabetes, Class label: 2 = diabetes (Target variable for prediction)
HighBP	Whether the individual has high blood pressure (1 = Yes, 0 = No)
HighChol	Indicator for high cholesterol levels (1 = Yes, 0 = No)
BMI	Body Mass Index (a continuous measure of body fat based on weight and height)
Stroke	Whether the individual has experienced a stroke (1 = Yes, 0 = No)
GenHlth	Self-reported general health status (scale from 1: Excellent to 5: Poor)
MentHlth	Number of days in the past 30 with poor mental health
PhysHlth	Number of days in the past 30 with poor physical health
DiffWalk	Binary indicator for difficulty walking or climbing stairs (1: Yes, 0: No)
HeartDiseaseorAttack	Binary indicator for history of coronary heart disease or heart attack (1: Yes, 0: No)
Age	Age category (categorical values, higher values indicate older age groups)

Following initial observations can be drawn –

- Most features are binary indicators, making preprocessing straightforward.
- Features such as BMI, HighBP, and Age exhibit a strong relationship with the target variable.
- Class 0 (no diabetes) forms the largest class, necessitating oversampling for Class 1 and Class 2 during training.

BENCHMARKING OF OTHER SOLUTIONS

Notebook Name	Feature Approach	Model Approach	Train/Test Performance
Natecekay's "Diabetes Predictions Using Classification Models" (**link**)	- Utilizes all features, including lifestyle factors and medical indicators - Removal of outliers - Label Encoding to transform non-numerical labels - Feature Scaling	- Train/Test Ratio: 80–20% - **Model**: Random Forest Classifier with parameter tuning - Robust feature engineering and hyperparameter tuning improves results - Feature importance analysis reduces noise	**Accuracy**: 85% **F1 score**: 0.89
Milica Radisavljevic's "Diabetes Indicators Classification" (**link**)	- Feature selection based on correlation score > 0.024 - Handles outliers in MentHlth, PhysHlth, BMI, GenHlth - Feature Scaling	- Train/Test Ratio: 80–20% - **Models:** Decision Tree, Random Forest, Multi-class Logistic Regression with L2 Penalty - Hyperparameter tuning via GridSearchCV improves precision	**Accuracy**: 83% **F1 Score:** 0.91
Muhammad Awwab Khan's "Diabetes Classification" (**link**)	- Feature selection via correlation - Normalization applied to 'BMI', 'GenHlth', 'MentHlth', 'PhysHlth', 'Age', 'Income'	- Train/Test Ratio: 70–30% - **Models**: Decision Tree, Random Forest, Logistic Regression, KNN, Gradient Boosting, Neural Network (1 layer) - SMOTE improves class balance - Gradient Boosting performs best	**Accuracy:** 83.5% **F1 Score:** 0.79

Analysis of Approaches and Kernel Performance:

Natecekay's Kernel:

- **Strength**: Robust feature engineering and hyperparameter tuning for Random Forest Classifier.
- **Success Factor:** Feature importance analysis reduces noise, improving model performance. It balances accuracy and precision effectively, crucial in health-related datasets.

Milica Radisavljevic's Kernel:

- **Strength:** Hyperparameter tuning via GridSearchCV and use of correlation for feature selection.
- **Success Factor:** Optimal tuning enhanced the logistic regression model, leading to the highest F1 score (0.91). It emphasizes interpretability and precision.

Muhammad Awwab Khan's Kernel:

- **Strength:** Use of SMOTE to handle imbalanced classes and Gradient Boosting for performance.
- **Success Factor:** SMOTE ensures better detection of minority classes, a vital aspect in diabetes prediction datasets. Gradient Boosting maintains a good bias-variance tradeoff.

DATA DESCRIPTION AND INITIAL PROCESSING

1 Summary Statistics:

- The dataset has 253,680 observations with no missing data in any of the 22 columns.

- The mean BMI is 28.38, with a standard deviation of 6.61, indicating variation in weight across individuals.
- About 29.7% of the population falls under the category of diabetes or prediabetes (target variable "Diabetes_012").
- Around 42.9% of the individuals have high blood pressure, and 42.4% have high cholesterol.
- Most individuals have had their cholesterol checked (96.3%).
- No missing values were found in the dataset, indicating that the data is complete, and no further imputation or handling of missing values is required.

2 Distribution of Diabetes Status (Target Variable - Diabetes_012):

This count plot shows how much sampled data is distributed between no diabetes (80%), prediabetes (5%) and diabetes (15%) which is our target variable.

This imbalance indicates that further attention required for handling class imbalance during model building.

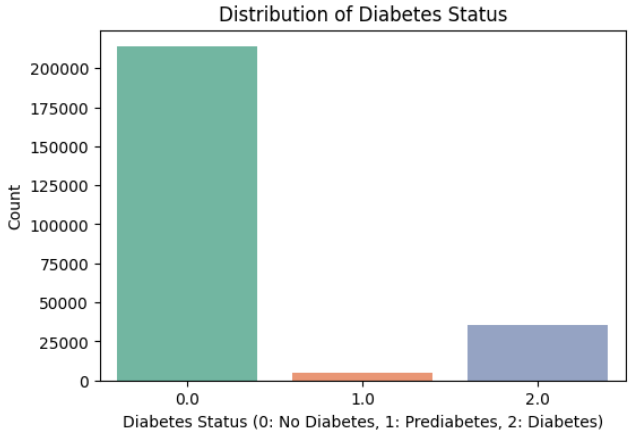

Figure 1: Histogram for Distribution of target class

3 Preprocessing:

3.1 Stratified Sampling

Since this dataset is imbalanced, stratified sampling ensures both classes are adequately represented in training, validation and testing datasets.

3.2 Feature Scaling

As most features are binary in nature, continuous/numeric variables like BMI, Mental Health days, Physical Health days, Age, and Income have varied ranges. To account for the different scales and units of the features, the data was standardized. This involved transforming the features to have a mean of 0 and a standard deviation of 1. All features, except target variable, were standardized to prepare the data for Machine Learning models.

3.3 Correlation and Feature Selection:

The importance of each feature in predicting the target variable was assessed using the Chi-square test and a correlation matrix with a heatmap. This step helps to potentially reduce dimensionality and concentrate on the most relevant features.

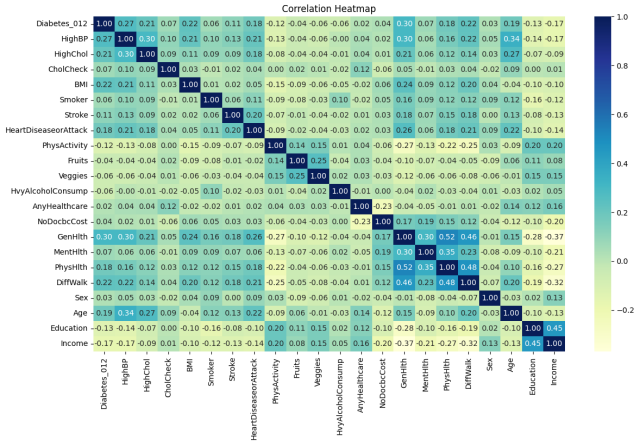

Figure 2: Heatmap to show correlation among features

- **GenHlth** (General Health), **HighBP** (High Blood Pressure), and **BMI** seem to have slightly higher correlations with the target variable (Diabetes_012), indicating their relevance to diabetes status.
- Features like Age, **HighChol** (High Cholesterol), and **DiffWalk** (Difficulty Walking) also show some correlation with diabetes status
- Most features show weak correlations with the target variable (Diabetes_012), indicates that no single feature strongly predicts diabetes on its own.

The below plot (Figure 3) highlights the combined effect of High Blood Pressure (HighBP) and High Cholesterol (HighChol) on the target variable (Diabetes_012). The presence of both HighBP and HighChol appears to amplify the risk of diabetes, making it a critical combination to consider in predictive modeling.

Potential Feature Engineering: Creating a Generalized Health Score that incorporates multiple health conditions (e.g., HighBP, HighChol, and other related features) might help better capture the combined effects.

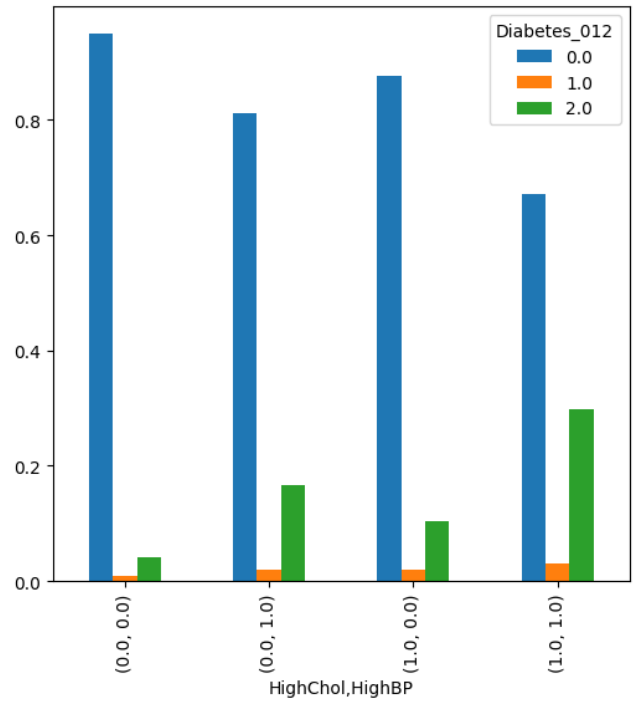

Figure 3: Bar Plot to show combined effect of HighCol and HighBP on target class

3.4 Outlier Detection:

The boxplots (shown below) highlight features with outliers and reveal associations between each feature and diabetes status. Boxplot for 'MentHlth', 'PhysHlth', 'BMI' reflects presence of outlier. I will approach by evaluating model performance with and without outliers to understand the trade-offs. If require will utilized interquartile range (IQR) method for removing outliers.

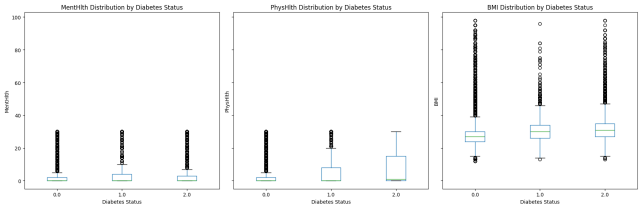

Figure 4: Boxplot to identify outliers

3.5 Addressing Class Imbalance:

Approach 1: The target variable (Diabetes_012) is imbalanced, with most entries in the non-diabetic category. I will use SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic samples for the minority classes (prediabetes and diabetes) and Near Miss (under sampling majority class), balancing the data for improved model performance.

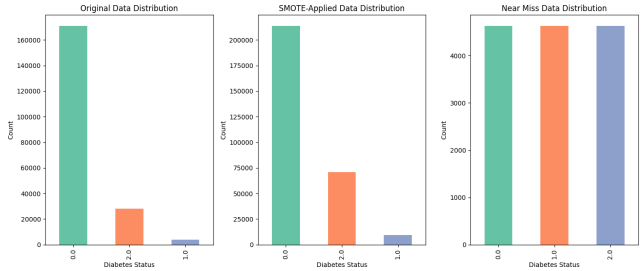

Figure 5: Resampling Distribution

Approach 2: Using algorithms, such as random forests, and logistic regression, which allow adjusting class weights to assign more importance to the minority classes. E.g. RandomForestClassifier(class_weight="balanced")

I will use Evaluating Metrics as F1-Score. Accuracy is not the best metric in imbalanced datasets because it may favor the majority class. Due to this reason, I will Focus on F1-Score as this metric give a better understanding of how well the model performs on the minority classes.

MODELING

The modeling phase aims to evaluate the relevance of independent variables and compare the performance of different machine learning algorithms to predict diabetes status (Diabetes_012). The dataset contains a mix of categorical and continuous features, representing health conditions, lifestyle factors, and demographics. Due to the imbalanced nature of the target variable, particular care was taken to evaluate performance metrics beyond accuracy, such as **precision**, **recall**, and **F1-score**, to assess the models' ability to predict minority classes effectively.

1 Analysis and Relevance of Independent Variables

The relevance of independent variables was analyzed using correlation heatmaps, distribution plots, feature importance from tree-based models. For instance, BMI showed a positive correlation with diabetes status, as Obesity is a well-established risk factor for diabetes. Health Indicators like High Blood Pressure and High Cholesterol, these metabolic risk factors show moderate correlation with the target variable and strong relevance based on domain knowledge. The presence of both High Blood Pressure and High Cholesterol appears to amplify the risk of diabetes, making it a critical combination to consider in predictive modeling.

The self-reported health indicators highlight physical and general health issues often associated with diabetes. Also, age distribution indicates older age groups are at higher risk for diabetes. Smoking and cardiovascular conditions are risk factors for diabetes, could be critical in distinguishing between target class.

2 Models

I have worked on 3 different models: Logistic Regression, Random Forest Classification, and XGBoost (eXtreme Gradient Boosting) Classification.

Firstly, I employed resampling techniques, including Near Miss and SMOTE, to address class imbalance. For SMOTE, I synthetically doubled the minority class rather than matching its size to that of the majority class. This approach was strategically chosen to balance the dataset while mitigating the risks of overfitting and unnecessary duplication of synthetic samples. By limiting oversampling to twice the size of the minority class, the model benefits from a more diverse and representative dataset distribution, reducing the likelihood of overfitting.

Before training the model, I standardized the data, as the features were numerical. Standardization ensures that all features are on the same scale, preventing the model from being biased toward variables with larger magnitudes. Additionally, I identified and removed irrelevant features using a combination of correlation heatmaps and Chi-squared (Chi2) scores to focus on the most significant predictors and improve model efficiency.

2.1 Logistic Regression:

Initially trained a Logistic Regression model on the imbalanced dataset without addressing the class imbalance and with assigning class_weight="balanced". Observed poor performance for minority classes (1: Prediabetes and 2: Diabetes) due to their underrepresentation in the dataset.

Metrics for the baseline model:

- High accuracy driven by the dominant majority class (0: No Diabetes).
- Low precision, recall, and F1-scores for classes 1 and 2.
- ROC AUC Score: 0.81
- Accuracy: 84.58

To improve performance for minority classes, I applied **Near Miss** and **SMOTE** resampling techniques.

2.1.1 Impact on Performance:

- Overall reduction in accuracy, resulting from reduce biasness of majority class.
- Using Near Miss, Increased recall for minority classes (1 and 2) as the model learned to predict them better. Overall increased ROC-AUC score and significant improvement in F1-score for minority classes due to better recall (ROC – AUC – 0.89).
- Using SMOTE, Improves the ROC-AUC score by effectively identifying the minority class (ROC – AUC – 0.82).

Metric	Baseline Logistic Regression	Near Miss	SMOTE
Accuracy	High (biased toward class 0)	Moderate	Moderate
Precision (Class 1)	high	High	High
Recall (Class 1)	Very Low	High	Low
F1-Score (Class 1)	Very Low	High	Low
Precision (Class 2)	Moderate	High	Moderate
Recall (Class 2)	Low	High	Moderate
F1-Score (Class 2)	Low	High	Moderate

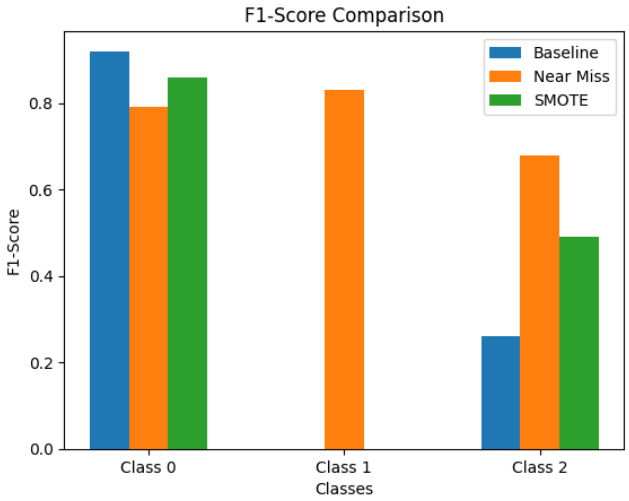

Figure 6: Bar Chart to show Comparison on F1-Score for logistic Regression

Additionally, I combined HighBP and HighChol by summing their values, as this combination demonstrated potential for identifying more positive cases. Given the slight right skew in the distribution of BMI, I applied a log transformation to normalize it. Through these feature engineering techniques and the application of Near Miss sampling, I successfully improved the model's efficiency in predicting all three class labels.

```
Logistic Regression with under sampling ssing Near Miss:
Accuracy: 0.79273119827276
f1_score: 0.7938127949190231
roc_auc: 0.9153718703312166
Classification Report:
              precision    recall  f1-score   support

         0.0       0.75      0.90      0.82       926
         1.0       0.95      0.77      0.85       926
         2.0       0.71      0.71      0.71       927
```

2.2 Random Forest Classification:

Next, I utilized the Random Forest model, taking advantage of its capability to effectively manage feature interactions and handle imbalanced datasets. Initially, I trained the Random Forest model using 100 estimators, a random state of 42, and balanced class weights. Compared to Logistic Regression, the Random Forest model demonstrated superior performance with both class weighting and SMOTE resampling, achieving the following metrics -

- **Accuracy**: ~85%
- **Precision and Recall:** Improved for class 2 but still moderate for class 1.
- **F1-Score:** Balanced performance across classes.
- **ROC Curve:** AUC ~0.90, outperforming Logistic Regression.

```
Random Forest Classifier:
Accuracy: 0.8465231900837703
f1_score: 0.8304093937633233
roc_auc: 0.8911852676149217
Classification Report:
              precision    recall  f1-score   support

         0.0       0.87      0.96      0.91     42741
         1.0       0.62      0.07      0.13      1852
         2.0       0.76      0.62      0.68     14139

    accuracy                           0.85     58732
   macro avg       0.75      0.55      0.57     58732
weighted avg       0.83      0.85      0.83     58732
```

Now to improve the accuracy, I plotted a feature importance plot for the random forest classifier and got my top features.

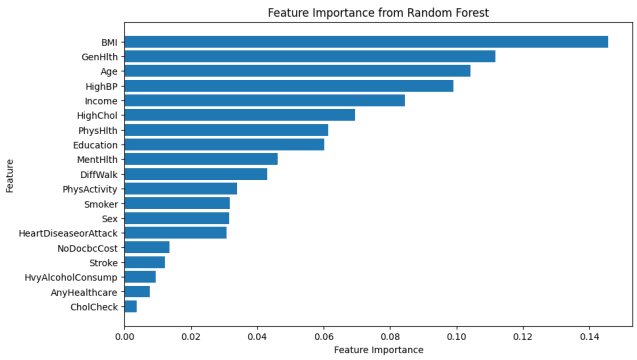

Figure 7: Feature Importance using Random Forest

I selected the top 13 features to train another Random Forest model and evaluated its accuracy. However, the ROC-AUC score was lower than the previous model (ROC-AUC: 0.88).

2.3 XGBoost Classification:

With the XGBoost model, multiple weak predictors are combined to create a strong and robust predictor. This approach leverages the power of gradient boosting, enabling the model to capture complex patterns effectively.

Once trained, XGBoost is highly efficient in both prediction and computation. I achieved exceptional results with this model, and what sets my approach apart is the extensive hyperparameter tuning I performed, which significantly enhanced its performance. Additionally, XGBoost's ability to handle imbalanced datasets, its support for regularization to reduce overfitting, and its scalability make it a powerful choice for classification tasks like this.

I believe my model is well-optimized, as the hyperparameter tuning process allowed me to identify the ideal parameters, resulting in improved accuracy and overall performance.

I started with creating baseline XGBoost Model using initial configuration as defaulted hyperparameters with 100 estimators, a learning rate of 0.1, a maximum depth of 6 and ***scale_pos_weight*** defined as ration of class label 0 count to class label 2 count. Trained on the raw, imbalanced dataset. This model heavily biased toward the majority class (0: No Diabetes).

I performed hyperparameter tuning to determine the optimal parameters for this model on SMOTE resample data to enhance performance across all class labels. Parameters that I have used –

```
param_grid = {
  'max_depth': [3, 5, 7],
  'n_estimators': [50, 100, 200],
  'learning_rate': [0.01, 0.1, 0.2],
  'subsample': [0.8, 1.0],
  'colsample_bytree': [0.8, 1.0]
}
```

And got my best parameters as:
Best Parameters: {'colsample_bytree': 1.0, 'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200, 'subsample': 1.0}

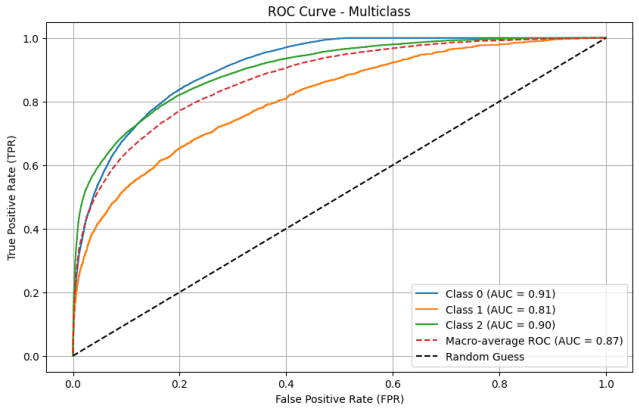

Figure 8: ROC Curve for Best Model

- The model performs exceptionally well for Class 0 and Class 2 (AUC > 0.90), showing its strength in distinguishing individuals with no diabetes and those with diabetes.

- The macro-average AUC of 0.87 highlights that, while the model is effective overall, but has some room for improvement, especially for minority class (Class 1).

This model gave me the best accuracy, f1_score and ROC AUC score. Balanced performance across all classes due to the incorporation of SMOTE and hyperparameter tuning.

Results:

```
XGB Classifier
Best Parameters: {'colsample_bytree': 1.0, 'learning_r
Accuracy: 0.8534189198392699
f1_score: 0.8344572782131469
roc_auc: 0.9045161942874265
Classification Report:
              precision    recall  f1-score   support

         0.0       0.86      0.97      0.92     42741
         1.0       0.75      0.06      0.12      1852
         2.0       0.81      0.59      0.68     14139
```

Metrics Summary Table –

Balancing Technique	Accuracy	F1-Score	ROC AUC Score
Logistic			
Class Weight	64%	0.72	0.81
Near Miss	76%	0.77	0.89
SMOTE	76%	0.74	0.81
Random Forest			
Class Weight	83%	0.79	0.77
Near Miss	80%	0.80	0.91
SMOTE	84%	0.83	0.89
XGBoost			
Class Weight	84%	0.81	0.82
Near Miss	79%	0.80	0.92
SMOTE	86%	~0.84	~0.91

APPENDIX

Dataset link:

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

GitHub link:

https://github.com/pmandloi28/Machine-Learning-Project

Earlier, I analyzed the relationship between dependent and independent features by plotting correlation heatmaps, distribution and boxplots to assess their relevance. Based on these insights, I identified the important features. Subsequently, I resampled and standardized the data to ensure uniform scaling and proceeded to implement the models.

Logistic Regression faces challenges in effectively handling imbalanced datasets, even when using class weighting and resampling techniques

I opted for Random Forest instead, as it excels in classification tasks and is well-suited for imbalanced datasets. Random Forest outperformed Logistic Regression by distinguishing class labels more effectively.

I also explored removing BMI outliers, but this unexpectedly reduced the model's performance. I believe this occurred because it made the feature less informative, as obesity is a crucial factor in identifying diabetes.

The proposed solution leverages XGBoost, combined with hyperparameter tuning, and resampling techniques like SMOTE, to address the limitations of alternative models such as Logistic Regression and Random Forest.