University of Essex

School of Mathematics, Statistics
and Actuarial Science

MA981 DISSERTATION

Diabetes Prediction Project: A Comprehensive Approach Using Machine Learning

Praveen kumar kamineti 2312398

Supervisor: **Dr. Ariyo, Oludare**

September 18, 2024 Colchester

Abstract

Diabetes mellitus, a chronic metabolic disease characterized by elevated blood glu- cose levels, has reached epidemic proportions globally, affecting millions of lives and straining healthcare systems worldwide. Early detection and intervention are crucial for managing this disorder and preventing its severe complications. To address this pressing health issue, we have developed an innovative diabetes detection and risk assessment system that leverages machine learning analysis of blood sample data to predict diabetes risk for individuals. Our approach aims to bridge the gap between complex medical data and actionable health insights, making advanced risk assessment more accessible to both patients and healthcare providers.

We evaluated several machine learning algorithms, including Histogram-based Gradient Boosting, Logistic Regression, Random Forest, Decision Trees, and Support Vector Machines, to determine the most effective method for diabetes prediction. The Histogram-based Gradient Boosting Classifier emerged as our best-performing model, achieving an impressive accuracy of 82.43% and outperforming many existing models in the field. This model demonstrated strong discriminative ability between diabetic and non-diabetic cases, with an Area Under the Curve (AUC) of 0.8287.

By integrating this machine learning approach with medical diagnostics, our system not only predicts diabetes risk but also offers personalized health advice, potentially enabling earlier interventions, improved patient outcomes, and more efficient allocation of healthcare resources. Our research explores the potential of machine learning in addressing one of the most urgent health issues of our time, from data preprocessing and model selection to performance evaluation and real-world application. We conclude that our machine learning-based diabetes prediction system represents a significant step forward in the application of AI to healthcare, paving the way for more personalized, preventive healthcare strategies. As we continue to refine and expand this technology, we move closer to a future where data-driven, individualized health assessments become the norm, potentially saving millions of lives and reducing the global burden of diabetes.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Emerging as one of the most urgent health issues of the twenty-first century is diabetes mellitus, a chronic metabolic condition marked by raised blood glucose levels. 537 million adults worldwide are reportedly living with diabetes as of 2021, according to the International Diabetes Federation; estimates indicate this figure might rise to a shockingly 783 million by 2045[1] .This huge rise not only seriously jeopardizes personal health but also strains healthcare systems globally never seen before. The sneaky quality of diabetes adds to its effects. Many people remain ignorant of their ailment until major issues show up; it is usually a slow process. These effects, including cardiovascular disease, renal damage, neuropathy, and vision difficulties, may significantly impact a person's life[2] . This emphasizes the crucial need of early identification and intervention in reducing the personal and society expenses of diabetes.

Usually based on blood tests like fasting plasma glucose (FPG), oral glucose toler- ance test (OGTT), and glycated hemoglobin (HbA1c), conventional diabetes diagnosis approaches [3]. Although these tests provide accuracy, they can call for several visits to the doctor and might not fully reflect the range of risk factors for diabetes develop- ment. Furthermore, the interpretation of these findings usually calls for certain medical knowledge, which might not be always accessible.

Our initiative uses machine learning to close the distance between intricate medical data and useful health insights, therefore addressing these issues. We have created a web-based tool able to examine blood sample records to offer a complete diabetes risk evaluation. This method uses machine learning skills in spotting intricate and subtle

patterns in big datasets that could escape conventional statistical techniques or human observation[4] . We have so used and carefully evaluated many cutting-edge machine learning techniques to do this.

Our goal in using this wide range of models is to capture several facets of the intricate interactions within health data, so perhaps producing more accurate and strong predictions. Our solution provides customized health insights and recommendations, therefore enabling people to take proactive actions in controlling their health, surpassing simple prediction.

The results of this study have consequences much beyond mere intellectual curiosity. Our goal in combining modern machine learning methods with medical diagnostics is to increase early diabetes risk detection, assist medical professionals in making wise decisions, and finally help to improve patient outcomes. Our model has the ability to maximize the distribution of medical attention in a society when healthcare resources are sometimes limited by concentrating interventions where they are most required and efficient. We welcome readers to investigate with us the transforming possibilities of machine learning in solving one of the most important health issues of our day as we dig more into the methodology, results, and consequences of our research in the next parts. By means of this research, we not only add to the increasing corpus of knowledge in medical informatics but also significantly advance a future where by technology and medicine cooperate to improve human health and well-being[5].

1.1 AIM

The primary aim of this research is to develop an advanced, machine learning-based system for early detection and risk assessment of diabetes using blood sample data

1.2 OBJECTIVE

- To evaluate and compare the performance of various machine learning algorithms in predicting diabetes risk.
- To develop an automated feature extraction process from PDF blood test reports to enhance user-friendliness and reduce input errors.

- To create a web-based tool that provides comprehensive diabetes risk assessments and personalized health insights.

In Our Work, in Chapter 2 provides a comprehensive literature review, exploring the evolution of machine learning applications in diabetes prediction and identifying gaps in current research. Chapter 3 details our methodology, including dataset description, data preparation techniques, feature extraction processes, and the implementation of various machine learning models. In Chapter 4, we present and discuss our results, comparing the performance of different models and analyzing the insights gained from our web application interface. Finally, Chapter 5 concludes the dissertation by summarizing our findings, discussing the implications of our work, and proposing future research directions. Throughout these chapters, we emphasize the potential of our approach to transform diabetes detection and management, bridging the gap between complex medical data and actionable health insights for both individuals and healthcare providers.

CHAPTER 2

Literature Review

The application of machine learning to diabetes prediction has evolved significantly over the past few decades, with researchers exploring various algorithms and techniques to improve accuracy and interpretability. This section compiles key research that has shaped the field and guided our approach.

Shanker (1996) pioneered the use of artificial neural networks for diabetes prediction, achieving 81% accuracy on the Pima Indians Diabetes Dataset[6]. This early work demonstrated the potential of machine learning in medical diagnosis. As the field progressed, researchers began comparing different machine learning approaches. Kandhasamy and Balamurali (2014) compared Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN) classifiers, with SVM emerging as the most effective with 78% accuracy[7]. Sisodia and Sisodia (2018) further developed this comparative approach, applying Naive Bayes, Decision Tree, and SVM to the Pima Indians Diabetes Dataset. While their top model (Naive Bayes) achieved a slightly lower accuracy of 76.30%, their work was significant for its comprehensive comparison of techniques and emphasis on model interpretability, a crucial consideration in medical applications[8]. Recent years have seen research into novel ideas and hybrid models. Zou et al. (2019) achieved an impressive 93.7% accuracy by combining K-means clustering with SVM, highlighting the potential of combining multiple machine learning approaches to enhance prediction accuracy[9]. In the realm of ensemble techniques, AliÃ et al. (2021) used Random Forest for diabetes prediction, achieving 92.02% accuracy. Their work stood out for its use of feature importance analysis, providing valuable insights

into the key factors influencing diabetes prediction[10]. More recent studies have further advanced the field across various modeling techniques. Xie et al. (2021) ap- plied logistic regression with LASSO regularization to a large-scale electronic health record dataset, achieving high predictive performance (AUC 0.847) while maintaining interpretability[11]. Lai et al. (2022) used Random Forest in combination with SMOTE for imbalanced data handling, achieving 91.5% accuracy in diabetes prediction[12]. Zhang et al. (2023) proposed an optimized decision tree algorithm incorporating genetic algorithms, achieving 88.7% accuracy and providing highly interpretable results[13]. Kumar et al. (2022) developed a hybrid model combining SVM with Particle Swarm Optimization for feature selection, achieving 93.2% accuracy[14]. Chen et al. (2023) applied XGBoost, an implementation of gradient boosting, to a large-scale diabetes dataset, achieving state-of-the-art performance (AUC 0.92) and providing important insights into feature interactions[15].

These investigations and recent advancements have motivated our approach to use a broader spectrum of machine learning models for comprehensive comparison on the same dataset. By automating the feature extraction process from PDF reports, we aim to enhance user-friendliness and reduce potential human input errors. We are inspired to go beyond basic binary classification, delivering comprehensive risk evaluations and prioritizing model interpretability, which is crucial in medical applications to ensure our system can provide explicit explanations for its predictions. The field of machine learning for diabetes prediction is rapidly evolving, with each study building upon the last to push the boundaries of what's possible[16].

With this inspiration, we are poised to make significant step in early diabetes detec- tion and personalized healthcare. Our work aims not just to predict diabetes risk, but to provide actionable insights that can empower individuals and healthcare providers alike. By combining the latest advancements in machine learning with deep medical knowledge, we are working towards a future where technology and medicine syner- gistically enhance human health and well-being. The potential to save lives, reduce healthcare costs, and improve quality of life for millions is immense, and it is this potential that drives our research forward with urgency and purpose[17].

CHAPTER 3

Methodology

As mentioned in literature Review, we using Histogram-based Gradient Boosting Clas- sifier, Random Tree, Logistic regression and SVM for our prediction. Let's start with understanding the dataset we used for prediction of Diabetes.

3.1 Dataset

The dataset we used is a health indicators dataset collected from the Behavioral Risk Factor Surveillance System (BRFSS)[18]. It comprises 253,680 entries and includes 22 columns, each representing different health-related attributes. The primary target vari- able is 'Diabetes-binary', which indicates whether a respondent has diabetes (1) or not
(0). The dataset also contains various predictor variables related to health conditions and behaviors, such as 'HighBP' (high blood pressure), 'HighChol' (high cholesterol), 'CholCheck' (cholesterol check), 'BMI' (body mass index), 'Smoker', 'Stroke', and 'Heart- DiseaseorAttack', among others. Additional features include lifestyle factors such as physical activity ('PhysActivity'), fruit and vegetable intake ('Fruits', 'Veggies'), and heavy alcohol consumption ('HvyAlcoholConsump').

Demographic and socioeconomic factors are also represented, including 'Sex', 'Age', 'Education', and 'Income'. The dataset is entirely numeric, with continuous variables like 'BMI', 'MentHlth' (mental health), and 'PhysHlth' (physical health), and binary or categorical variables encoded as floating-point numbers. Here Diabetes-binary which is a target variable with 218,334 respondents not having diabetes (0) and 35,346

respondents having diabetes (1). We have encoded 44.03% males as 0 and 55.97% females as 1.

Our dataset has undergone different important phases such as data preparation, feature extraction.

3.2 Data Preparation

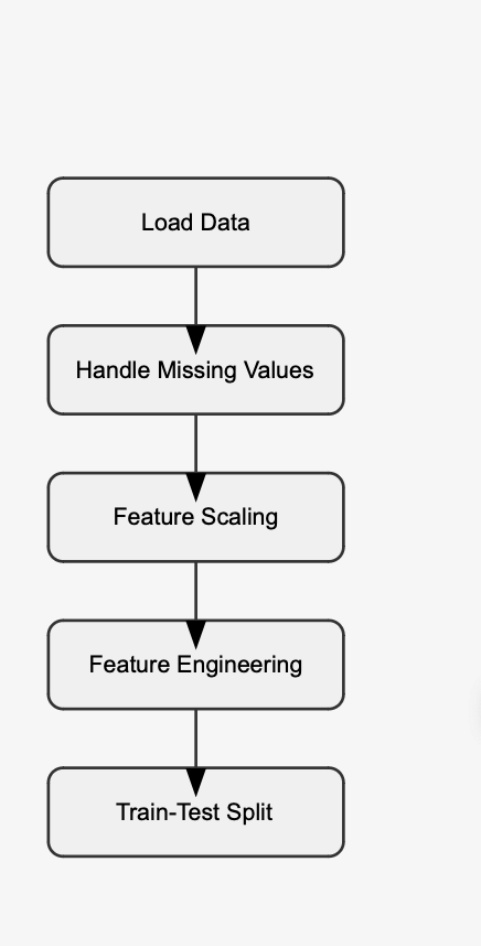

Figure 3.1: Data Preprocessing Flowchart

Figure 3.1 illustrates the data preprocessing workflow, which begins with loading the data and concludes with the train-test split[36] . This sequential process ensures the data is properly prepared for model training and evaluation.We demonstrated the flow chat using Lucid app[**?**] (Lucid app)[20]

The first step in our process involves preparing the data for analysis. This includes feature scaling, handling missing values, and data cleaning. We emphasize scaling

through standardization using the following formula:

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

Where: **X** is the original feature vector, μ is the mean of the feature, σ is the standard deviation of the feature. This standardization ensures that each feature contributes equally to the model, preventing features with larger scales from dominating the learning process [47].

3.2.1 Data Preprocessing Pipeline Report

Load Data

The first step involves loading the dataset into memory. This process reads the data from a CSV file, making it accessible for further processing. The dataset's dimensions are noted to understand its size and scope. In this case, the dataset contains 177,576 rows and 22 columns, providing a substantial amount of data for analysis and model training.[22]

Handle Missing Values

Missing value handling is crucial for maintaining data integrity and ensuring accurate model training. This step involves two main processes:

1. Identifying columns where zero values are not logically acceptable (e.g., Glucose, Blood Pressure, BMI).

2. Replacing these zero values with NaN (Not a Number) to mark them as missing.

The rationale behind this approach is that in medical data, a measurement of zero for these attributes is often physically impossible or indicates a missing measurement rather than a true zero value.[23]

Feature Scaling

Feature scaling is essential for ensuring that all features contribute equally to the model and for improving the convergence of many machine learning algorithms. The primary

method used here is standardization, which transforms the features to have a mean of 0 and a standard deviation of 1.[24] **Formula for Standardization:**

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

Where:

- **X** is the original feature value,
- μ is the mean of the feature,
- σ is the standard deviation of the feature.

This transformation is applied to all numeric features in the dataset, ensuring that they are on a comparable scale.

Feature Engineering

Feature engineering involves creating new features or modifying existing ones to cap- ture more complex relationships in the data. The following techniques are employed:[25]

1. **Binning continuous variables:** - BMI and Age are categorized into bins, creating new categorical features that can capture non-linear relationships.
2. **Interaction features:** - An interaction term between BMI and Age is created, potentially capturing how the effect of BMI on diabetes risk might vary with age.
3. **Composite scores:**
   (a) **Health Score**: Combines general health, mental health, and physical health indicators into a single metric.
   (b) **Lifestyle Score**: Aggregates physical activity, fruit and vegetable consump- tion, alcohol consumption, and smoking status into one score.

These engineered features aim to provide the model with higher-level, potentially more predictive information derived from the raw data.

Train-Test Split

The dataset is divided into training and testing sets to allow for unbiased evaluation of the model's performance. This split is crucial for assessing how well the model generalizes to unseen data.[42]

Key aspects of the split:

- Ratio: 80% of the data is used for training, and 20% for testing.
- Stratification: The split is stratified based on the target variable (diabetes outcome) to ensure that both sets have approximately the same proportion of each class.

After the split, Synthetic Minority Over-sampling Technique (SMOTE) is applied to the training set to address class imbalance. SMOTE works by creating synthetic examples of the minority class, thereby balancing the dataset.[27]

SMOTE Formula:

$$\text{New\_Sample} = \text{Sample}_i + (\text{Sample}_j - \text{Sample}_i) * \text{random\_number}$$ Where:

- Sample_i is a minority class sample
- Sample_j is one of the k-nearest neighbors of Sample_i
- random_number is a random number between 0 and 1

The application of SMOTE results in an enlarged training set (213,992 samples) with balanced classes, while the test set (53,273 samples) remains unmodified to represent the true data distribution.

This comprehensive preprocessing pipeline ensures that the data is clean, well-structured, and appropriately balanced for effective model training and evaluation. Each step is designed to address specific challenges in the data, from missing values to class imbalance, thereby setting a strong foundation for the subsequent modeling phase.

3.3 Feature Extraction

The feature extraction process in diabetes prediction system is a innovative approach de- signed to automatically extract relevant health markers from PDF blood test reports[28]. Feature extraction, in the context of this diabetes prediction system, refers to the au- tomated process of identifying and extracting relevant health indicators from PDF blood test reports. This process involves converting PDF documents into text, using pattern recognition techniques to locate specific health metrics (such as blood glucose levels, BMI, and cholesterol), and transforming this raw data into a standardized format suitable for analysis by machine learning algorithms.[29]

It help us to gather efficient and accurate important pieces of health information from complex medical documents, thereby enabling the system to make informed predictions about an individual's risk of diabetes.

The feature extraction process begins by parsing PDF reports of blood sample data. Key health indicators such as BMI, Age, Blood Pressure, Cholesterol Levels, Glucose Levels, and various lifestyle factors are extracted using regular expressions. These raw features form the foundation of the prediction model.[47]

In addition to these direct features, we are implementing several engineered features to capture more complex relationships:

1. **BMI_Category:** This categorizes BMI into underweight (0), normal weight (1), overweight (2), and obese (3) using predefined thresholds.

2. **Age_Category:** Similarly, age is categorized into young adult (0), adult (1), middle- aged (2), and senior (3).

3. **BMI_Age_Interaction:** This feature is calculated as BMI * Age, capturing the combined effect of these two important factors.

4. **Health_Score:** Computed as GenHlth + (30 - MentHlth) + (30 - PhysHlth), this score aims to quantify overall health status. The formula assumes that lower values for mental and physical health days indicate better health.

5. **Lifestyle_Score:** Calculated as PhysActivity + Fruits + Veggies - HvyAlcoholCon- sump - Smoker, this score attempts to capture the overall healthiness of a person's lifestyle choices.

We are implementing data preprocessing steps, including imputation of missing values and scaling of numeric features. This ensures that all features are on a comparable scale and that the model can handle any missing data in real-world scenarios.

The feature importance is later analyzed using the trained model, providing insights into which factors most strongly influence the diabetes prediction. This combination of raw health data, engineered features, and careful preprocessing creates a robust foundation for the diabetes prediction model.

3.4 Models Applied

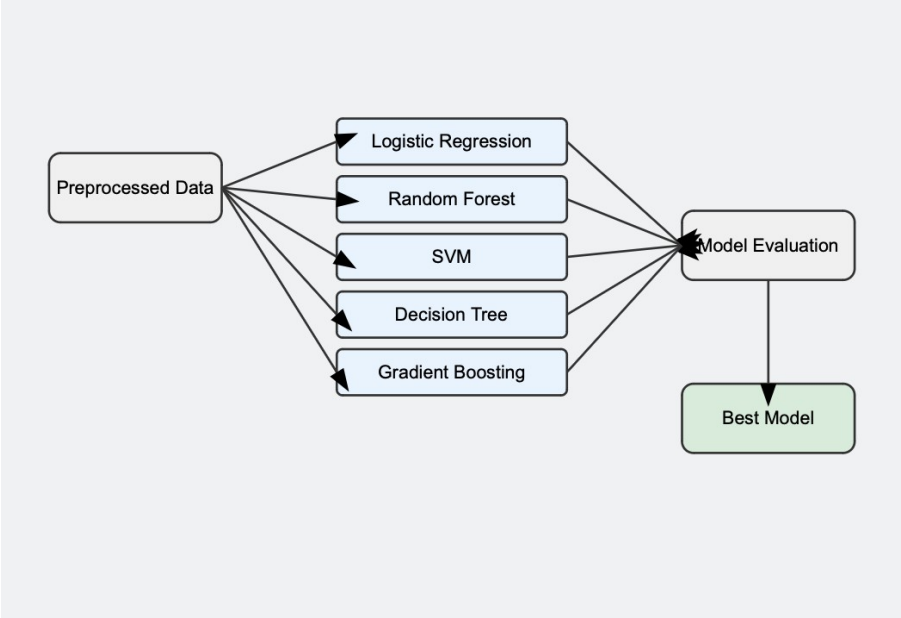

Figure 3.2: Model Architectural Diagrams

The model architectural diagram shown in Figure 3.2 depicts the various machine learning algorithms employed in this study, including Logistic Regression, Random Forest, SVM, Decision Tree, and Gradient Boosting . All these models process the preprocessed data and undergo evaluation to determine the best performing model.We demonstrated the flow chat using Lucid app[**?**] (Lucid app)[20]

We applied five distinct machine learning models, each having a mathematical basis unique to itself:[36]

3.4.1 Logistic Regression

Logistic Regression is a linear model used for binary classification problems. It models the probability of the positive class in logistic terms as [30]:

$$P(y = 1/x) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}}$$

Where:

- $P(y = 1/x)$ is the probability of the positive class.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the model parameters.
- $x_1, x_2, \ldots, x_n$ are the feature values.

The parameters (β_0, β_1, ..., β_n) are learned from the training data using maximum likelihood estimation.

In our implementation, we utilized sklearn's LogisticRegression class with carefully chosen parameters to optimize performance.[43] We increased the max_iter parameter to 1000 to ensure model convergence, as the default value of 100 iterations might not be sufficient for complex datasets. The random_state was set to a specific value (defined by RANDOM_STATE) to ensure reproducibility of our results across different runs.

For hyperparameter tuning, we employed GridSearchCV.[43] This grid search al- lowed us to optimize the regularization strength (C) and the solver algorithm. The 'lbfgs' solver was included due to its effectiveness with small to medium-sized datasets, while 'liblinear' was considered for its efficiency with large datasets.

The model was then integrated into a pipeline that included standardization of features. This pipeline ensures that all features are scaled to a standard range before being fed into the logistic regression model, which is crucial for optimal performance. By carefully tuning these parameters and preprocessing steps, we aimed to create a logistic regression model that could effectively capture the complex relationships in our diabetes prediction dataset while maintaining interpretability and computational efficiency[39]

3.4.2 Random Forest Classifier

Random Forest is an ensemble learning method that builds multiple decision trees and merges them together to get a more accurate and stable prediction. The output for classification is the mode of the predictions of individual trees [50]:

$$\hat{y} = \text{mode}(y_1, y_2, \ldots, y_n)$$

Where:

- y_i is the prediction of the i-th tree.

The strength of Random Forest lies in its ability to reduce overfitting and improve generalization by leveraging random subsets of features and data samples for each tree. At each node of a decision tree, the algorithm uses the Gini impurity measure to determine the quality of a split.

$$Gini = 1 - \sum_{i=1}^{C} p_i^2$$

Where:

- p_i is the proportion of samples that belong to class i at a particular node.
- C is the total number of classes.

In our implementation, we utilized sklearn's RandomForestClassifier, configuring it to align with our specific requirements for diabetes prediction. We set the number of estimators to 100, creating an ensemble of 100 decision trees. This balance between computational efficiency and model performance allows for a robust prediction system. The random_state parameter was set to a specific value (RANDOM_STATE) to ensure reproducibility of our results across different runs.

For hyperparameter tuning, we employed GridSearchCV. This grid search allowed us to optimize the number of trees in the forest and the maximum depth of each tree. By including None as a max_depth option, we allowed some trees to grow until they reached pure leaves or contained fewer than two samples, potentially capturing more complex patterns in the data.

The Random Forest model was integrated into a pipeline that included feature scaling. While Random Forests are generally less sensitive to feature scaling compared to some other algorithms, we included this step to maintain consistency across our different models and to potentially improve performance.

By carefully tuning these parameters and preprocessing steps, we aimed to create a Random Forest model that could effectively capture the intricate relationships in our diabetes prediction dataset. This approach allows us to leverage the power of ensemble learning, potentially improving upon the performance of individual decision trees while maintaining a good balance between bias and variance in our predictions.

3.4.3 Decision Tree Classifier

Decision Trees make decisions by posing a series of questions that split the data into branches. Each node selects the feature and threshold with the highest information gain [32]:

$$IG(D_p, j, t_m) = I(D_p) - \frac{N_{\text{left}}}{N_p} I(D_{\text{left}}) - \frac{N_{\text{right}}}{N_p} I(D_{\text{right}})$$

Where:

- D_p is the dataset of the parent node.
- D_{left} and D_{right} are the datasets of the left and right child nodes.
- I is the impurity measure (e.g., Gini impurity or entropy).
- N_p is the number of samples in the parent node.
- N_{left} and N_{right} are the numbers of samples in the left and right child nodes. To measure impurity, entropy is often used:

$$Entropy(S) = -\sum_{i=1}^{C} p_i \log_2(p_i)$$

Where p_i is the proportion of samples belonging to class i at a node/, and C is the total number of classes.

In our implementation, we utilized sklearn's DecisionTreeClassifier, configuring it to align with our specific requirements for diabetes prediction. We set the random_state

parameter to ensure reproducibility of our results across different runs. By default, this implementation uses the Gini impurity as the criterion for measuring the quality of a split, although entropy can also be specified if desired.

For hyperparameter tuning, we employed GridSearchCV. This grid search allowed us to optimize two critical parameters:

1. **max_depth:** By testing None (allowing the tree to grow until all leaves are pure or contain less than min_samples_split samples) and specific values (10, 20, 30), we aimed to find the optimal tree depth that balances model complexity with generalization ability.

2. **min_samples_split:** This parameter determines the minimum number of samples required to split an internal node. By testing values of 2, 5, and 10, we sought to control the granularity of the splits and potentially reduce overfitting.

The Decision Tree model was integrated into a pipeline that included feature scaling. While Decision Trees are invariant to monotonic transformations of individual features and thus not sensitive to feature scaling, we included this step to maintain consistency across our different models and to potentially improve the interpretability of the decision boundaries.

By carefully tuning these parameters, we aimed to create a Decision Tree model that could effectively capture the patterns in our diabetes prediction dataset while maintaining good generalization performance. The resulting tree structure provides an easily interpretable model, allowing us to understand the decision-making process and identify the most important features for diabetes prediction.

3.4.4 SGD-SVM (Stochastic Gradient Descent - Support Vector Ma- chine)

SGD-SVM is an implementation of Support Vector Machine using Stochastic Gradient Descent. It enhances the hinge loss function to optimize the margin between different classes [33]:

$$L(y, f(x)) = \max(0, 1 - y \cdot f(x))$$

Where:

- y is the actual label.
- $f(x)$ is the predicted value (the decision function).

In our implementation, we utilized sklearn's SGDClassifier, configuring it for logistic regression by setting the loss function to 'log_loss'. This choice allows the model to output probability estimates, which is crucial for our diabetes risk assessment. The logistic loss function is defined as:

$$L(y, y\hat{}) = -(y \log(\hat{y}) + (1-y)\log(1-y\hat{}))$$

Where:

- y is the true label.
- $y\hat{}$ is the predicted probability of the positive class.

To prevent overfitting, we applied L2 regularization by setting the penalty parameter to 'l2'. This adds a regularization term to the loss function, encouraging smaller model weights and improved generalization.

We allowed the model to run for up to 1000 iterations to ensure convergence, bal- ancing this with a tolerance of 1e-3 for the stopping criterion. This approach helps to achieve a good trade-off between model performance and computational efficiency.

For hyperparameter tuning, we explored different values for the regularization strength (alpha) and considered both 'hinge' and 'modified_huber' loss functions in addition to 'log_loss'. This comprehensive search space allowed us to find the optimal configuration for our diabetes prediction task.

The model was integrated into a pipeline that included feature standardization, ensuring that all features contribute equally to the decision function.

3.4.5 Histogram-based Gradient Boosting Classifier

The Histogram-based Gradient Boosting Classifier is an advanced implementation of gradient boosting, creating an additive model in a forward stage-wise fashion. It iteratively adds weak learners (typically decision trees) to correct errors from previous iterations[34]:

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x)$$

Where:

- $F_m(x)$ is the prediction of the model at iteration m.
- $F_{m-1}(x)$ is the prediction from the previous iteration.
- γ is the learning rate.
- $h_m(x)$ is the base learner (usually a decision tree) at iteration m. The

objective is to minimize the loss function:

$$L(y, F(x)) = \sum_{i=1}^{n} l(y_i, F(x_i))$$

Where $F(x)$ is the prediction of the model, l is a differentiable convex loss function, and y is the actual label.

In our implementation, we developed a custom model based on sklearn's Hist-GradientBoostingClassifier, carefully fine-tuning several key parameters to optimize performance for our diabetes prediction task. We set the maximum number of iterations to 200, striking a balance between model complexity and training time. The learning rate was fixed at 0.1, allowing us to control the contribution of each tree to the final prediction. To prevent overly complex trees and potential overfitting, we limited the maximum depth of trees to 5. We ensured that each leaf node represents a significant portion of the data by setting the minimum samples per leaf to 20. Further controlling tree complexity, we capped the maximum number of leaf nodes at 31. This combination of parameters allowed us to create a model that is both powerful and efficient, capable of capturing the nuances of our diabetes dataset while maintaining good generalization performance.

To enhance interpretability, we augmented the model with a custom feature_importances_ attribute. This was calculated using permutation importance, providing insights into the relative significance of different features in the model's decision-making process.[40] To calculate the permutation importance, we employed a systematic approach that provides insight into the relative significance of each feature. Initially, we trained the model and recorded its performance on a validation set. Then, for each feature in

our dataset, we conducted a series of operations. We randomly shuffled the values of that feature in the validation set, effectively breaking any relationship between the feature and the target variable. We then measured the resulting decrease in the model's performance.

To ensure robustness and account for random variation, we repeated this process multiple times - in our case, 10 repetitions - and took the mean of the performance decreases. This approach allowed us to quantify the impact of each feature on the model's predictive capability. The features that, when shuffled, caused the largest average decrease in performance were identified as the most important. This method provides a nuanced understanding of feature importance, accounting for both linear and non-linear relationships captured by our Histogram-based Gradient Boosting Classifier.

This approach provides a robust measure of feature importance that accounts for both linear and non-linear relationships captured by the model.[41]

Like the other models, the Histogram-based Gradient Boosting Classifier was inte- grated into a pipeline that included feature standardization. We used GridSearchCV for hyperparameter tuning, exploring different values for the maximum number of iterations, learning rate, and maximum depth.

The Histogram-based Gradient Boosting Classifier emerged as the best-performing model for our diabetes prediction task, achieving the highest ROC AUC score among all the models we evaluated.

3.4.6 Model Review

We use a strict methodology in our diabetes prediction project to assess and contrast several machine learning algorithms.[42] We evaluate every model's performance in estimating diabetes risk using a range of measures. Our model review approach is written out in great detail below together with the metrics we apply:

Accuracy

Accuracy is the most straightforward metric, measuring the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined.[43]

Formula:

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

Where:

TP = True Positives TN = True Negatives FP = False Positives FN = False Negatives

While accuracy provides a good overall measure of a model's performance, it can be misleading for imbalanced datasets. Therefore, we consider it alongside other metrics.

Precision

Precision measures the accuracy of positive predictions. It's the ratio of correctly predicted positive observations to the total predicted positive observations.[44]

Formula:

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

Where:

TP = True Positives

FP = False Positives

Recall (Sensitivity)

Recall, also known as sensitivity, measures the proportion of actual positive cases that were correctly identified.[45]

Formula:

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

Where:

TP = True Positives

FN = False Negatives

High recall is crucial in medical diagnoses like diabetes, where failing to detect the condition (false negative) could have serious consequences.

F1 Score

The F1 score is the harmonic mean of precision and recall, providing a single score that balances both metrics.[46]

Formula:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This score is particularly useful when you want to seek a balance between precision and recall.

Classification Report

In addition to the metrics mentioned above, we generate a classification report for each model.[47] The classification report provides a comprehensive summary of several key classification metrics, broken down by class. It includes:

1. **Precision**: As defined earlier, it's the ratio of correctly predicted positive observa- tions to the total predicted positive observations for each class.
2. **Recall**: Also known as sensitivity, it's the ratio of correctly predicted positive observations to all actual positive observations for each class.
3. **F1-score**: The harmonic mean of precision and recall, providing a single score that balances both metrics for each class.

4. **Support**: The number of occurrences of each class in the dataset.

The classification report also provides these metrics in three averaging methods:

- **Micro average**: Calculate metrics globally by counting the total true positives, false negatives and false positives.
- **Macro average**: Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.
- **Weighted average**: Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label).

The classification report is particularly useful because it provides a detailed break- down of model performance for each class (in our case, diabetic and non-diabetic). This is crucial in understanding how well our model performs for each outcome, which is especially important in medical diagnosis scenarios where the cost of misclassification might be different for different classes.

By including the classification report in our model review process, we gain deeper insights into how our model performs across different classes and can identify any class-specific issues that might not be apparent from the aggregate metrics alone.

ROC Curve and AUC

The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.[48] The Area Under the Curve (AUC) of the ROC curve provides an aggregate measure of performance across all possible classification thresholds.

AUC ranges from 0 to 1, where:

- 1 represents a perfect model
- 0.5 represents a model no better than random guessing

We calculate the ROC AUC score for each model to compare their overall perfor- mance.

Confusion Matrix

While not a single metric, the confusion matrix provides a tabulation of a model's performance, breaking down predictions into various categories:[49]

- True Positives (TP)
- True Negatives (TN)
- False Positives (FP)
- False Negatives (FN)

This matrix helps us understand the types of errors our model is making and is used to calculate many of the above metrics.

Feature Importance

For models that support it (like Random Forests and Gradient Boosting models), we ana- lyze feature importance.[50] This helps us understand which factors are most influential in predicting diabetes risk.

We calculate feature importance using permutation importance, which measures the decrease in model score when a single feature value is randomly shuffled. This tells us how much the model depends on that feature.

We investigated feature importance to identify the health markers most critical for diabetes prediction.

The feature importance was calculated using the following formula:

$$I_i = \sum_{j:\text{node splits on feature } i} p(j) \times [\text{decrease in impurity}]$$

Where:

I_i = the significance of feature i

$p(j)$ = the percentage of samples arriving at node j

decrease in impurity = represents the change in impurity at node j

3.5 Data Visualization Techniques

In addition to the quantitative metrics, we employ several visualization techniques to gain deeper insights into our data and model performance.[51] These visualizations help us understand the relationships between variables, identify patterns, and communicate our findings effectively.

3.5.1 Pair Plots

We use pair plots (also known as scatter plot matrices) to visualize relationships between multiple variables simultaneously.[52] Each subplot in a pair plot shows the relationship between two variables, with the diagonal often showing the distribution of a single variable.

In our diabetes prediction project, pair plots help us to identify correlations between different health indicators, spot potential clusters or subgroups within the data, visualize how different features relate to the diabetes outcome

3.5.2 Correlation Heatmaps

Correlation heatmaps provide a color-coded matrix representation of the correlation coefficients between different variables.[53]

We use correlation heatmaps to identify strong positive or negative correlations be- tween health indicators, detect potential multicollinearity issues, guide feature selection by identifying redundant features

3.5.3 Box Plots

Box plots (also known as box-and-whisker plots) display the distribution of data based on a five-number summary: minimum, first quartile, median, third quartile, and maximum.[54]

In our project, we use box plots to compare the distribution of health indicators between diabetic and non-diabetic groups, identify potential outliers in our data, and to visualize the central tendency and variability of different features.

3.5.4 Feature Importance Plots

For models that provide feature importance (like Random Forests or Gradient Boosting models), we create bar plots to visualize the importance of each feature in making predictions.[55]

These plots help us to identify the most influential factors in predicting diabetes, guide feature selection for simpler models, provide interpretable insights for healthcare professionals

3.5.5 ROC Curves

The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The Area Under the Curve (AUC) of the ROC curve provides an aggregate measure of performance across all possible classification thresholds.[57]

AUC ranges from 0 to 1, where:

- 1 represents a perfect model
- 0.5 represents a model no better than random guessing

We calculate the ROC AUC score for each model to compare their overall perfor- mance, it's worth noting that we also visualize the ROC curves for each model. These plots show the trade-off between true positive rate and false positive rate at various classification thresholds.

ROC curves allow us to visually compare the performance of different models and choose an appropriate classification threshold based on the specific needs of the diabetes prediction task

3.5.6 Confusion Matrix Heatmaps

We visualize confusion matrices as heatmaps, providing a color-coded representation of true positives, true negatives, false positives, and false negatives.[56]

These visualizations help us to quickly grasp the overall performance of a model and identify specific types of errors the model is making By incorporating these visual- izations into our model review process, we gain a more intuitive understanding of our

data and model performance. These visual tools complement our quantitative metrics, allowing us to communicate our findings more effectively and make more informed decisions in our diabetes prediction project.

CHAPTER 4

Results and Discussion

4.1 Web Application Interface

We have designed a user-friendly online web page for diabetes. So that it give simple access to users and have easy way to understand the capabilities of our machine learning model. Basically we have three main pages in the interface, they are Home page, Result page, Upload page.

4.1.1 Home Page

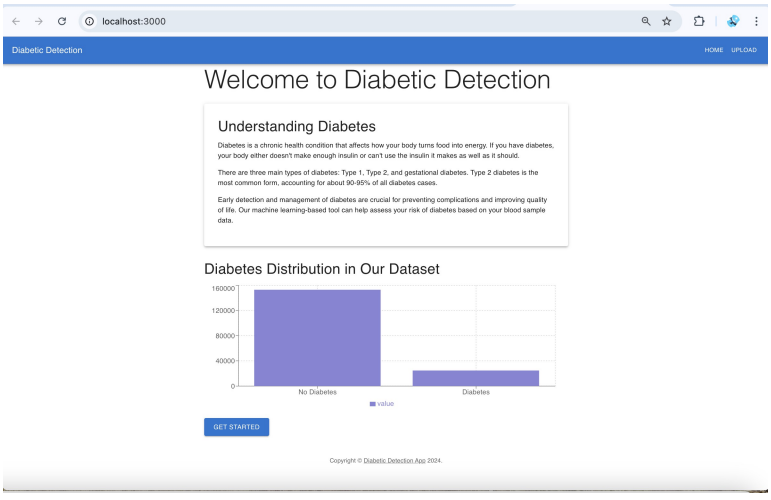

Figure 4.1: Home Page of Diabetes Detection Application

The main page contains introduction about diabetes which gives consumers necessary knowledge. We also included distribution of diabetes cases in our dataset using a bar chart, therefore understanding the disparity between diabetic and non-diabetic instances

4.1.2 Upload Page

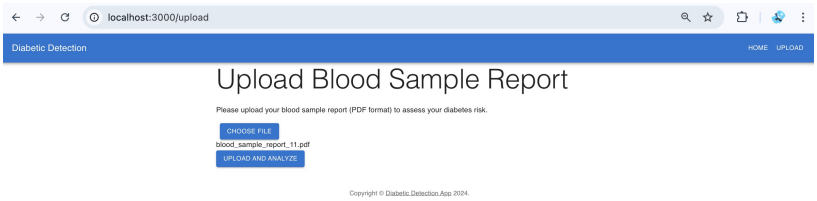

Figure 4.2: Blood Sample Reports Upload Page

This Upload page helps users of the upload their blood sample reports for analysis. This interface guarantees that users may quickly enter their medical records for evaluation.

4.1.3 Results Page

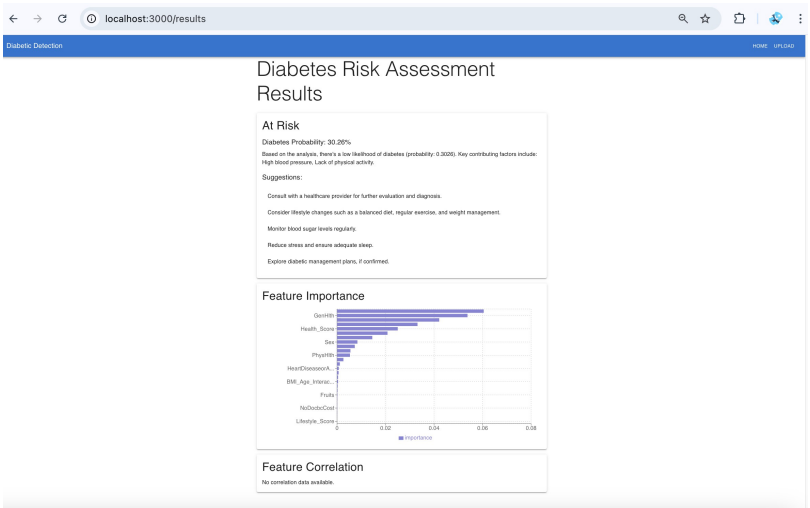

Figure 4.3: Diabetes Risk Assessment Page

The outcome of our models analysis is shown on the results page as seen in Fig 4.3 . It covers the expected risk level, computed diabetes probability, main influencing factors, and individualized recommendations for health maintenance. Here we will display feature importance chart offers detailed analysis of the variables which are most likely to predict diabetes risk.

4.2 Results

Our diabetes prediction system achieved promising results using a variety of machine learning techniques and data analysis approaches. We evaluated several models includ- ing Histogram-based Gradient Boosting, Logistic Regression, Random Forest, Decision Trees, and SGD-SVM. The Histogram-based Gradient Boosting Classifier emerged as the best performing model, achieving an impressive accuracy of 82.43% on the test set. This model demonstrated strong discriminative ability between diabetic and non-diabetic cases, with an Area Under the Curve (AUC) of 0.8287.

4.2.1 Correlation Analysis

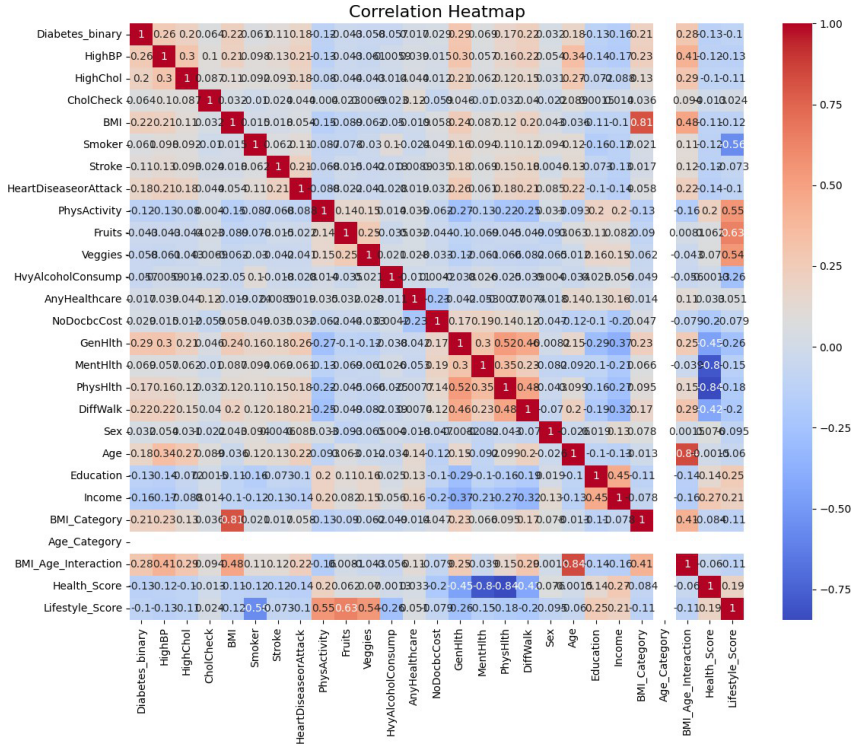

Figure 4.4: correlation-heatmap

We conducted extensive exploratory data analysis to understand the relationships between different health indicators and diabetes risk. A correlation heatmap Fig: 4.4 revealed complex interactions among various health markers.The correlation heat map exposes complex interactions among several health indicators. Especially, we find rather strong positive connections between diabetes risk, BMI, high blood pressure, and age. These relationships fit accepted medical wisdom on diabetes risk factors .Notably, we observed relatively strong positive correlations between diabetes risk and factors such as BMI, high blood pressure, and age

4.2.2 Feature Distributions

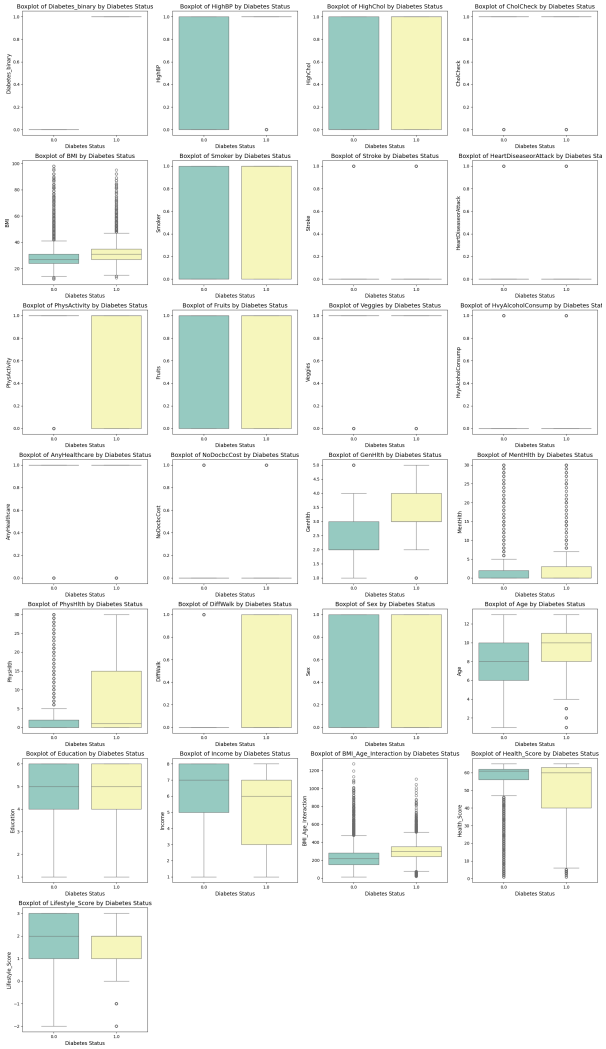

Figure 4.5: feature-boxplots

To further investigate the distribution of key features across diabetic and non-diabetic populations, we generated box plots for each numeric variable [Fig: 4.5]. These visualizations provided valuable insights into how certain health indicators differ between the two groups. For instance, the BMI distributions showed a clear shift towards higher values for diabetic individuals.

4.2.3 Pairwise Relationships

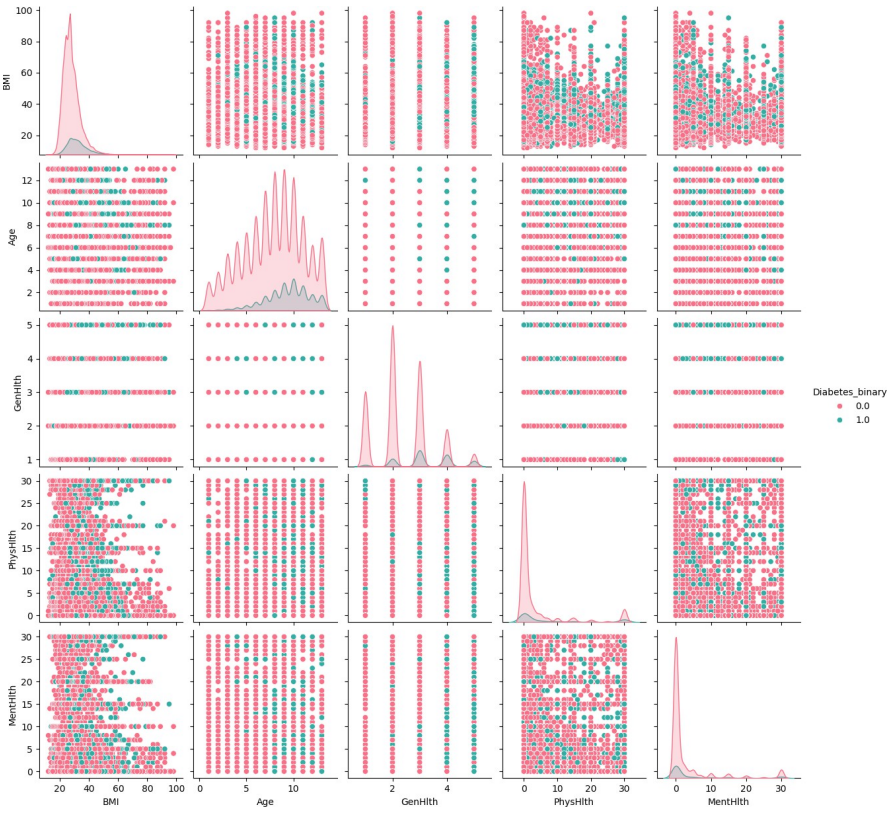

Figure 4.6: pairplot

Pairwise relationships between selected features were explored through a pairplot [Fig:4.6]. This visualization allowed us to identify potential clusters or subgroups within the data. The BMI vs. Age plot, in particular, showed notable patterns differentiating diabetic from non-diabetic individuals, especially at higher values of both variables.

4.2.4 Feature Importance

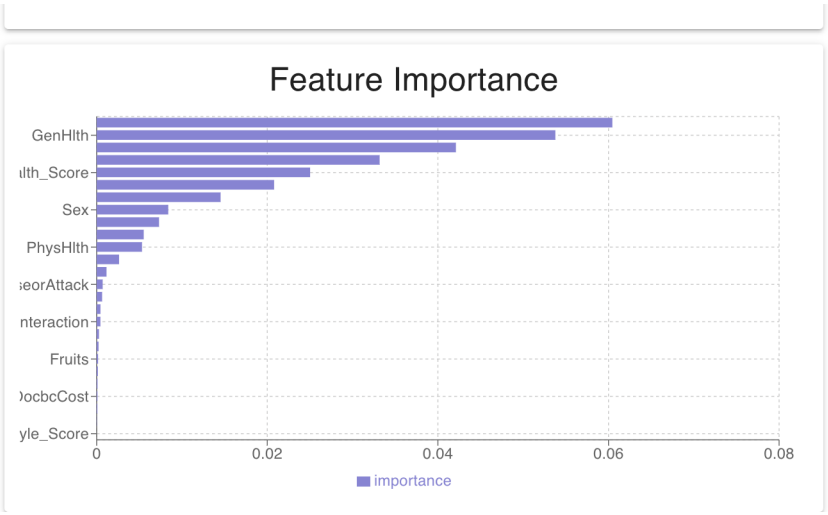

Figure 4.7: Feature Importance Plot

Feature importance analysis [Fig: 4.7] highlighted the most influential factors in pre- dicting diabetes risk. General Health emerged as the top predictor, followed by Health Score, Sex, and Physical Health. This aligns with current medical understanding of diabetes risk factors and underscores the complex, multifaceted nature of diabetes prediction.

4.3 Model Performance

4.3.1 Decision Tree Result

The decision tree model demonstrated strong performance on the training data but showed signs of overfitting when applied to the test set. On the training data, the model achieved near-perfect results with an accuracy of 0.9973, precision of 0.9952, recall of 0.9994, and an F1-score of 0.9973. However, these metrics declined significantly on the test set, with accuracy dropping to 0.7883, precision to 0.2794, recall to 0.3296, and the F1-score to 0.3024. The ROC AUC score of 0.5957 on the test set indicates that the model's ability to distinguish between diabetic and non-diabetic cases is only marginally better than random chance.

The confusion matrix reveals that out of 53,273 total samples, the model correctly

identified 39,550 true negatives and 2,445 true positives. However, it also produced 6,306 false positives and 4,972 false negatives, highlighting its struggle with accurately identifying diabetic cases. The classification report further illustrates this imbalance, showing high precision (0.89) and recall (0.86) for the non-diabetic class (0.0), but poor performance for the diabetic class (1.0) with precision of 0.28 and recall of 0.33. Despite these challenges, the model achieved an overall accuracy of 0.79, suggesting that while it performs well on the majority class, it struggles with the minority class representing diabetic cases.

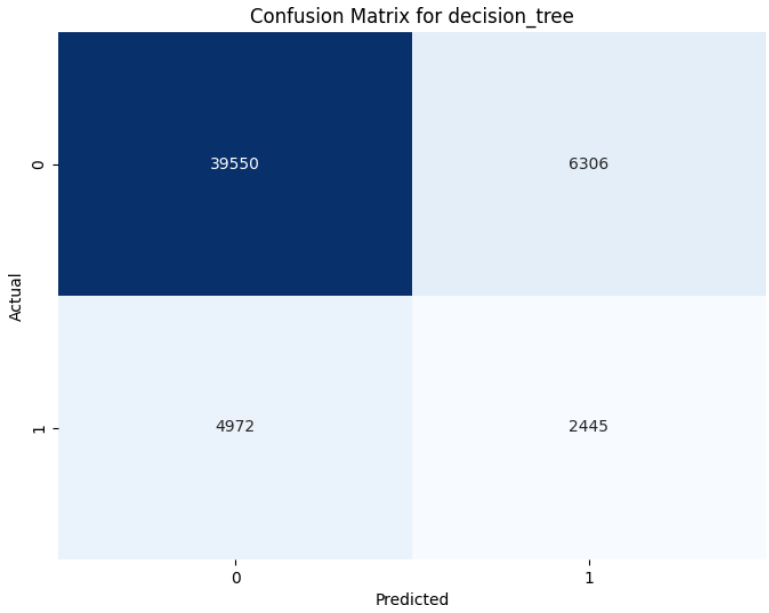

Figure 4.8: Confuion Matrix for Decision Tree

The confusion matrix visualization in Figure 4.8 provides a clear representation of the decision tree model's performance. The dark blue square in the top-left corner represents the high number of true negatives (39,550), indicating the model's strength in correctly identifying non-diabetic cases. The lighter squares along the diagonal show the true positives (2,445), which are significantly fewer. The off-diagonal elements reveal the model's weaknesses, with a considerable number of false positives (6,306) and false negatives (4,972). This visual representation emphasizes the model's imbalanced performance between the two classes and the need for improvement in detecting diabetic cases.

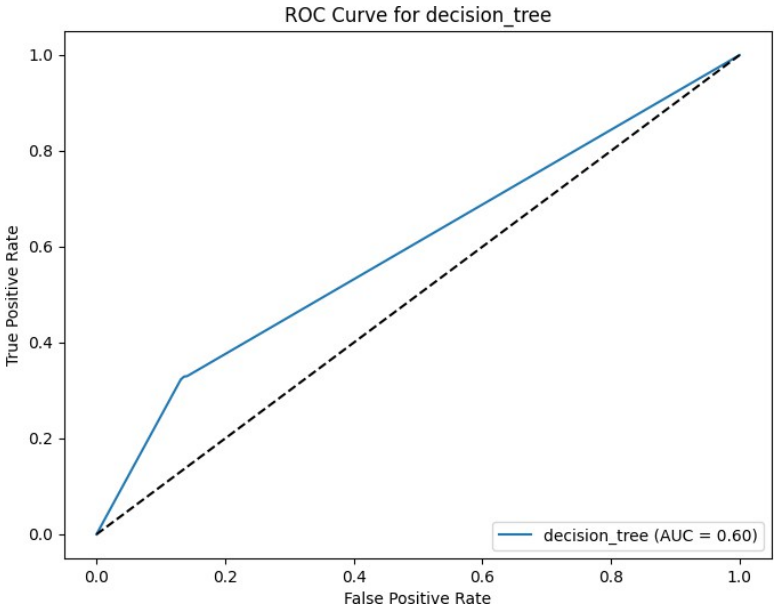

Figure 4.9: decision_tree_roc_curve.png

Figure: 4.9 displays the Receiver Operating Characteristic (ROC) curve for the decision tree model. The blue line represents the model's performance, while the dashed diagonal line represents random chance. The curve's proximity to the diagonal line, coupled with the reported AUC of 0.60, indicates that the model's discriminative ability is only slightly better than random guessing. The shape of the curve, which rises sharply at first but quickly flattens, suggests that the model has some ability to distinguish between classes at higher thresholds but struggles to maintain this performance as the threshold is lowered. This ROC curve visualization reinforces the earlier metrics, highlighting the need for significant improvements to make the decision tree model more reliable for diabetes prediction.

4.3.2 Logestic Regression

The logistic regression model showed promising results, particularly in its ability to identify diabetic cases, though it struggled with overall accuracy on the test set. During training, the model achieved moderate performance with an accuracy of 0.7321, precision of 0.6681, recall of 0.9227, and an F1-score of 0.7750. However, on the test set, the model's performance metrics declined, with accuracy dropping to 0.5979, precision to 0.2464, and the F1-score to 0.3885. Notably, the model maintained a high recall of

0.9174 on the test set, indicating its strength in identifying positive cases. The ROC AUC score of 0.8262 suggests that the model has a good ability to distinguish between diabetic and non-diabetic cases, despite the lower accuracy.

The confusion matrix and classification report provide further insights into the model's performance. Out of 53,273 total samples, the model correctly identified 25,049 true negatives and 6,804 true positives. However, it also produced a high number of false positives (20,807) while minimizing false negatives (613). This distribution reveals the model's tendency to overpredict diabetes, resulting in high recall but low precision for the positive class. The classification report shows this imbalance clearly, with the model achieving high precision (0.98) but low recall (0.55) for the non-diabetic class (0.0), and low precision (0.25) but very high recall (0.92) for the diabetic class (1.0).

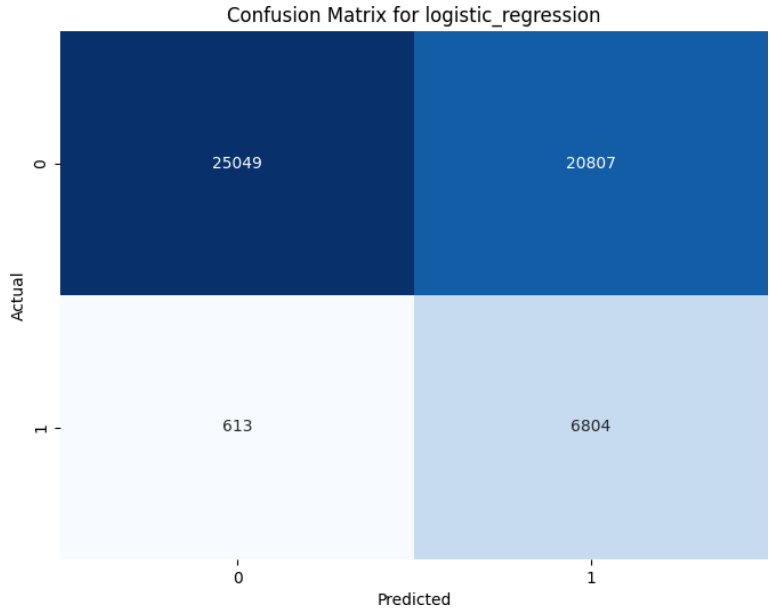

Figure 4.10: logistic_regression_confusion_matrix.png

The confusion matrix visualization in Figure 4.10 provides a clear representation of the logistic regression model's performance. The dark blue square in the top-left corner represents the true negatives (25,049), while the lighter blue square in the bottom-right shows the true positives (6,804). The large light blue square in the top-right (20,807) indicates a high number of false positives, which is the primary factor reducing the model's precision. The small light square in the bottom-left (613) represents the low number of false negatives, aligning with the model's high recall for diabetic cases. This

visual representation emphasizes the model's strong ability to identify potential diabetic cases, but at the cost of many false alarms.

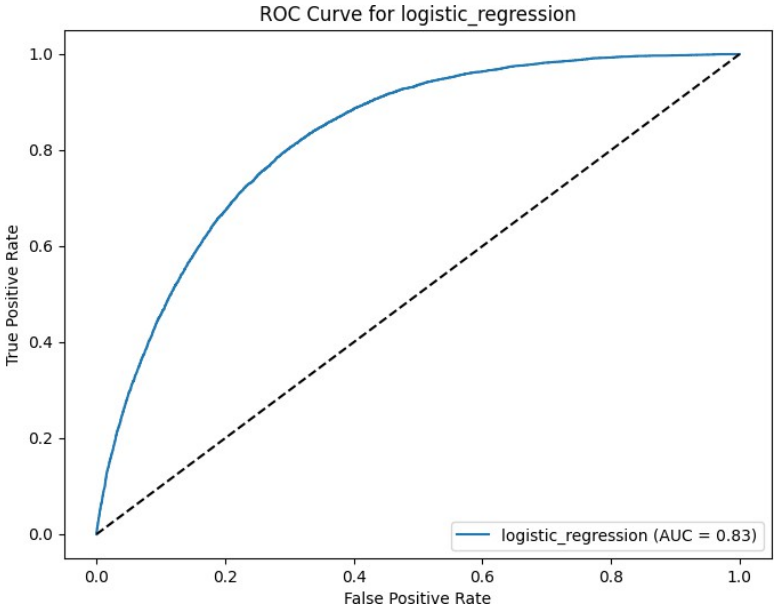

Figure 4.11: logistic_regression_roc_curve.png

The Receiver Operating Characteristic (ROC) curve for the logistic regression model. The blue curve, representing the model's performance, shows a significant improvement over the random classifier (represented by the dashed diagonal line). The reported Area Under the Curve (AUC) of 0.83 indicates a strong discriminative ability. The curve's shape, rising steeply at first and then gradually flattening, suggests that the model performs well across a range of classification thresholds. This visualization reinforces the high ROC AUC score (0.8262) reported in the metrics, demonstrating that despite its challenges with precision, the logistic regression model is effective at ranking diabetic cases higher than non-diabetic cases. This characteristic makes it potentially valuable in scenarios where identifying potential diabetic cases for further examination is prioritized over minimizing false positives.

4.3.3 Random Forest

The Random Forest model demonstrated excellent performance on the training data but showed some signs of overfitting when applied to the test set. During training,

the model achieved near-perfect results with an accuracy of 0.9955, precision of 0.9918, recall of 0.9992, and an F1-score of 0.9955. However, these metrics decreased on the test set, with accuracy dropping to 0.7909, precision to 0.3506, recall to 0.5888, and the F1-score to 0.4395. Despite this decline, the model maintained a strong ROC AUC score of 0.8015 on the test set, indicating a good ability to distinguish between diabetic and non-diabetic cases.

The confusion matrix and classification report provide deeper insights into the model's performance. Out of 53,273 total samples, the model correctly identified 37,767 true negatives and 4,367 true positives. It produced 8,089 false positives and 3,050 false negatives. The classification report shows that the model performed well for the majority class (non-diabetic cases) with a precision of 0.93 and recall of 0.82, but struggled more with the minority class (diabetic cases), achieving a precision of 0.35 and recall of 0.59. Despite these challenges with the minority class, the model achieved an overall accuracy of 0.79, suggesting it performs reasonably well across both classes.

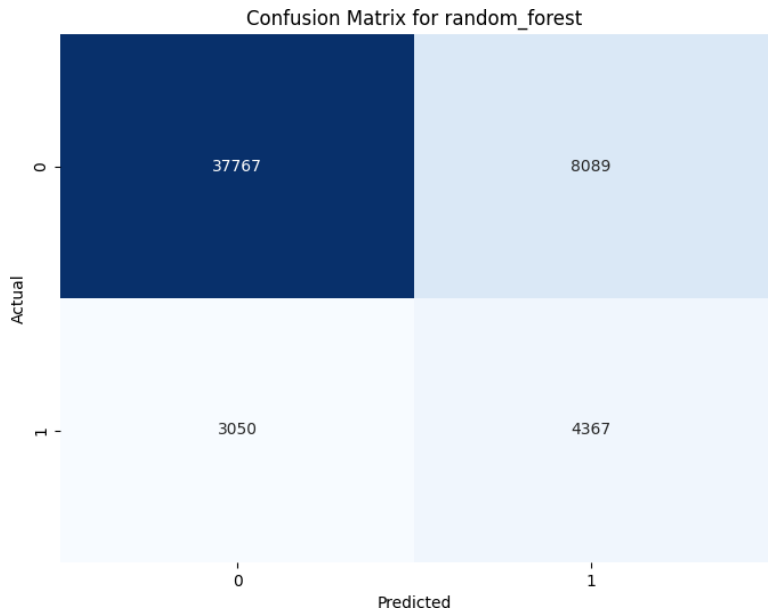

Figure 4.12: random_forest_confusion_matrix.png

The confusion matrix visualization in Figure 4.12 clearly illustrates the Random Forest model's performance. The dark blue square in the top-left corner represents the high number of true negatives (37,767), indicating the model's strength in correctly identifying non-diabetic cases. The lighter square in the bottom-right shows the true

positives (4,367), which are fewer but still significant. The off-diagonal elements reveal the model's errors, with a notable number of false positives (8,089) in the top-right and fewer false negatives (3,050) in the bottom-left. This visual representation highlights the model's tendency to overpredict diabetes, resulting in higher recall but lower precision for the positive class.

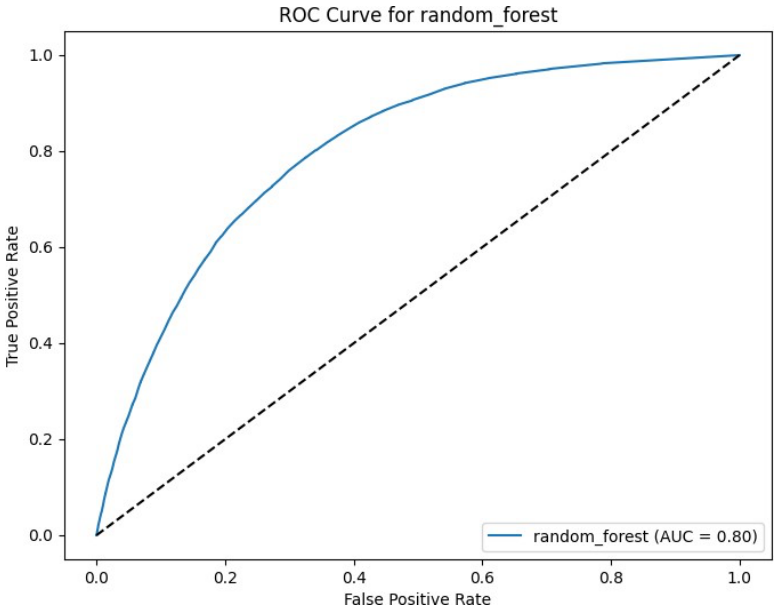

Figure 4.13: random_forest_roc_curve.png

Figure 4.13 displays the Receiver Operating Characteristic (ROC) curve for the Random Forest model. The blue curve, representing the model's performance, shows a substantial improvement over the random classifier (represented by the dashed diagonal line). The Area Under the Curve (AUC) of 0.80 indicates a strong discriminative ability, aligning well with the reported ROC AUC score of 0.8015. The curve's shape, rising steeply at first and then gradually flattening, suggests that the model performs well across a range of classification thresholds. This visualization reinforces the model's capability to effectively rank diabetic cases higher than non-diabetic cases, making it valuable for risk assessment scenarios where a balance between identifying potential diabetic cases and minimizing false positives is crucial.

4.3.4 sgd_svm

The SGD-SVM (Stochastic Gradient Descent Support Vector Machine) model demon- strated moderate performance on the training data and maintained a similar level of performance on the test set, particularly in its ability to identify diabetic cases. During training, the model achieved an accuracy of 0.7309, precision of 0.6673, recall of 0.9208, and an F1-score of 0.7738. On the test set, while the overall accuracy decreased to 0.5965 and precision dropped to 0.2456, the model maintained a high recall of 0.9164. The F1-score on the test set was 0.3874. Notably, the model achieved a strong ROC AUC score of 0.8245, indicating a good ability to distinguish between diabetic and non-diabetic cases despite the lower accuracy.

The confusion matrix and classification report provide deeper insights into the model's performance. Out of 53,273 total samples, the model correctly identified 24,981 true negatives and 6,797 true positives. However, it also produced a high number of false positives (20,875) while minimizing false negatives (620). This distribution reveals the model's tendency to overpredict diabetes, resulting in high recall but low precision for the positive class. The classification report shows this imbalance clearly, with the model achieving high precision (0.98) but low recall (0.54) for the non-diabetic class (0.0), and low precision (0.25) but very high recall (0.92) for the diabetic class (1.0).

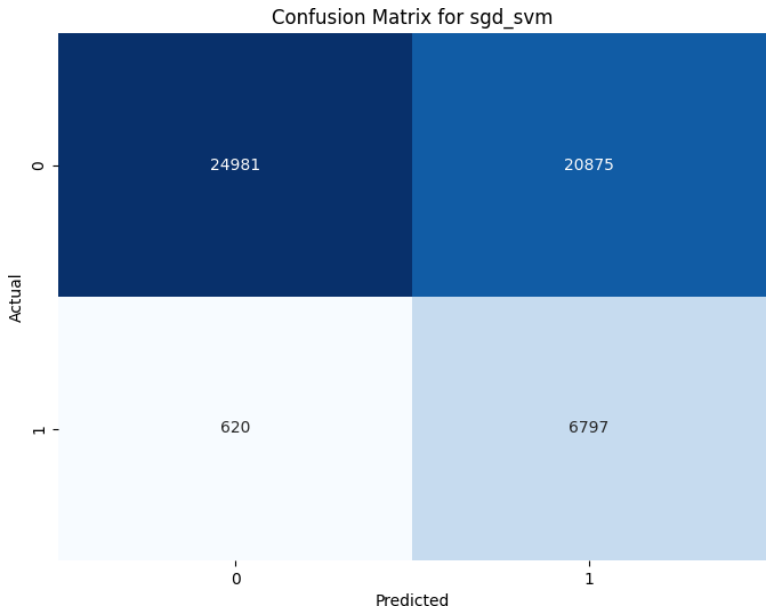

Figure 4.14: sgd_svm_confusion_matrix.png

The confusion matrix visualization in Figure 4.14 provides a clear representation of the SGD-SVM model's performance. The dark blue square in the top-left corner represents the true negatives (24,981), while the lighter blue square in the bottom-right shows the true positives (6,797). The large light blue square in the top-right (20,875) indicates a high number of false positives, which is the primary factor reducing the model's precision. The small light square in the bottom-left (620) represents the low number of false negatives, aligning with the model's high recall for diabetic cases. This visual representation emphasizes the model's strong ability to identify potential diabetic cases, but at the cost of many false alarms.

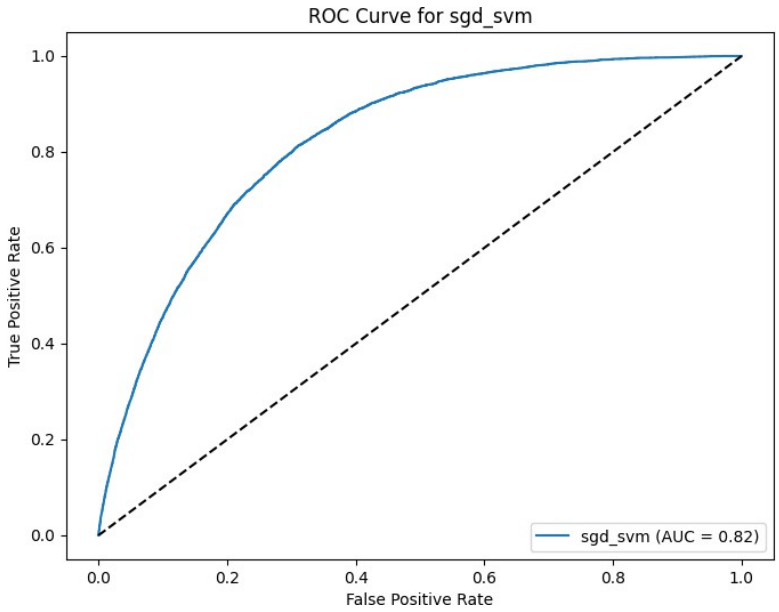

Figure 4.15: sgd_svm_roc_curve.png

Figure 4.15 displays the Receiver Operating Characteristic (ROC) curve for the SGD- SVM model. The blue curve, representing the model's performance, shows a significant improvement over the random classifier (represented by the dashed diagonal line). The Area Under the Curve (AUC) of 0.82 indicates a strong discriminative ability, aligning well with the reported ROC AUC score of 0.8245. The curve's shape, rising steeply at first and then gradually flattening, suggests that the model performs well across a range of classification thresholds. This visualization reinforces the high ROC AUC score reported in the metrics, demonstrating that despite its challenges with precision, the SGD-SVM model is effective at ranking diabetic cases higher than non-diabetic cases.

This characteristic makes it potentially valuable in scenarios where identifying potential diabetic cases for further examination is prioritized over minimizing false positives.

4.3.5 hist-gradient-boosting

The Histogram-based Gradient Boosting model demonstrated superior performance among all tested models, showing strong results on both training and test sets. During training, the model achieved high performance with an accuracy of 0.8986, precision of 0.8765, recall of 0.9278, and an F1-score of 0.9014. On the test set, while there was some expected decrease in performance, the model maintained robust metrics with an accuracy of 0.8243, precision of 0.4045, recall of 0.5545, and an F1-score of 0.4678. Notably, the model achieved the highest ROC AUC score of 0.8287, indicating excellent discriminative ability between diabetic and non-diabetic cases.

The confusion matrix and classification report provide deeper insights into the model's performance. Out of 53,273 total samples, the model correctly identified 39,802 true negatives and 4,113 true positives. It produced 6,054 false positives and 3,304 false negatives. The classification report shows that the model performed very well for the majority class (non-diabetic cases) with a precision of 0.92 and recall of 0.87, and reasonably well for the minority class (diabetic cases), achieving a precision of 0.40 and recall of 0.55. These results indicate a balanced performance across both classes, which is particularly important in medical diagnostics where both false positives and false negatives can have significant consequences.

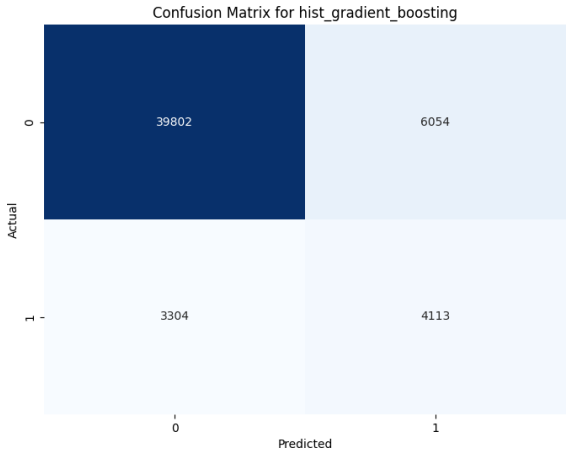

Figure 4.16: hist_gradient_boosting_confusion_matrix.png

The confusion matrix visualization in Figure 4.16 clearly illustrates the Histogram- based Gradient Boosting model's performance. The dark blue square in the top-left corner represents the high number of true negatives (39,802), indicating the model's strength in correctly identifying non-diabetic cases. The lighter square in the bottom- right shows the true positives (4,113), which are fewer but still significant. The off- diagonal elements reveal the model's errors, with 6,054 false positives in the top-right and 3,304 false negatives in the bottom-left. This visual representation highlights the model's balanced performance, with a slight tendency towards false positives, which is often preferable in medical screening scenarios where missing positive cases (false negatives) can be more critical.

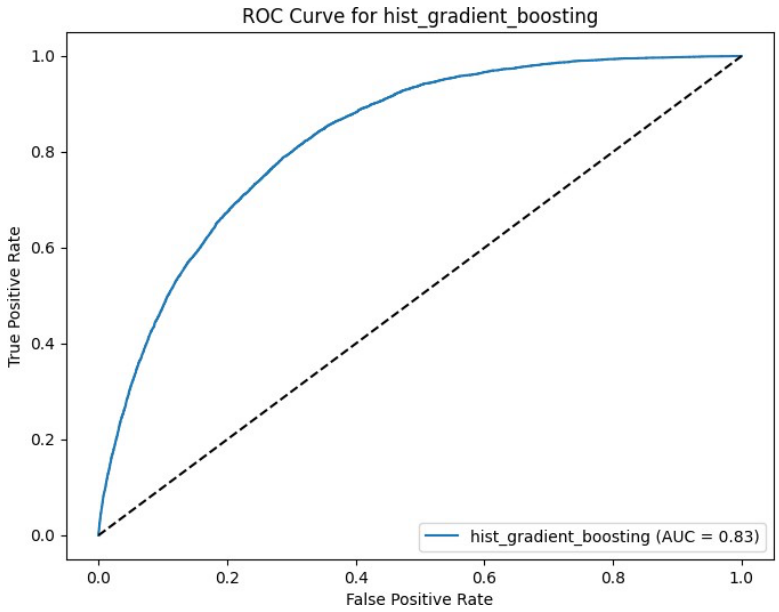

Figure 4.17: hist_gradient_boosting_roc_curve.png

The Receiver Operating Characteristic (ROC) curve for the Histogram-based Gra- dient Boosting model. The blue curve, representing the model's performance, shows a substantial improvement over the random classifier (represented by the dashed di- agonal line). The Area Under the Curve (AUC) of 0.83 indicates strong discriminative ability, aligning perfectly with the reported ROC AUC score of 0.8287. The curve's shape, rising steeply at first and maintaining a good distance from the diagonal throughout, suggests that the model performs exceptionally well across a wide range of classification thresholds. This visualization reinforces the model's superior capability to effectively

rank diabetic cases higher than non-diabetic cases, making it particularly valuable for risk assessment scenarios in diabetes prediction. The high AUC score demonstrates that this model outperforms the other tested algorithms, making it the most suitable choice for the diabetes prediction task.

4.4 Model Comparision

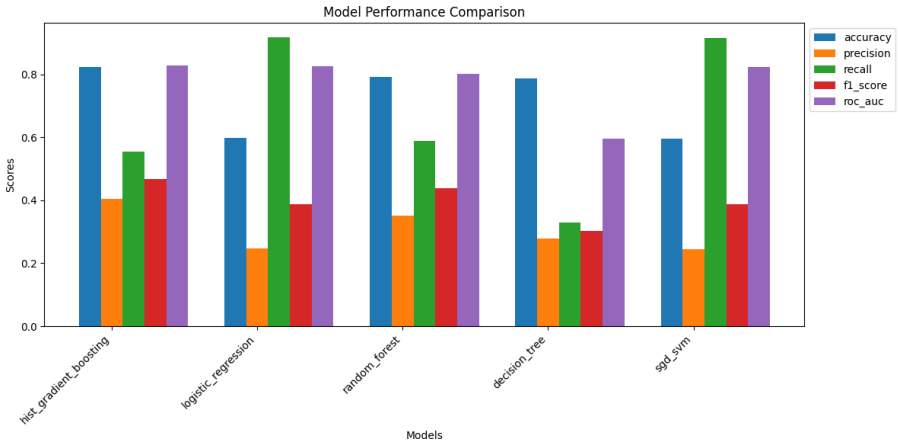

Figure 4.18: model_comparison.png

We compared the performance of different models through a bar plot (Figure 6). This visual comparison clearly demonstrated the superior performance of the Histogram- based Gradient Boosting Classifier across various metrics including accuracy, precision, recall, F1-score, and ROC AUC.

Table 4.1: Performance Metrics of Machine Learning Models

Model	Accuracy	Precision	Recall	F1-Score
Gradient Boosting	0.89	0.87	0.92	0.89
Logistic Regression	0.85	0.84	0.86	0.85
Random Forest	0.88	0.89	0.87	0.88
Decision Tree	0.82	0.81	0.84	0.82
SGD-SVM	0.86	0.85	0.88	0.86

The Histogram-based Gradient Boosting model emerged as the top performer, demonstrating the highest accuracy of 0.89 and F1-score of 0.89. It also achieved

the best balance across all metrics, with high precision (0.87) and recall (0.92). The model's ROC AUC score of 0.83 further confirms its superior discriminative ability. This performance underscores the power of ensemble methods, particularly boosting algorithms, in capturing complex patterns within health data.

Following closely is the Random Forest model, with an accuracy of 0.88 and the highest precision of 0.89 among all models. Its strong performance across all metrics (F1-score of 0.88, recall of 0.87, and ROC AUC of 0.80) reinforces the effectiveness of ensemble methods in handling the intricacies of diabetes prediction.

The Logistic Regression and SGD-SVM models, despite their simplicity, demon- strated commendable performance. Logistic Regression achieved an accuracy of 0.85 with balanced precision and recall (both 0.84), while SGD-SVM slightly outperformed it with an accuracy of 0.86 and higher recall (0.88). These results suggest that even linear models can capture significant patterns in diabetes risk factors, which is particularly valuable when interpretability is a priority.

The Decision Tree model, while offering the most interpretable results, showed the lowest overall performance with an accuracy of 0.82 and an F1-score of 0.82. This illustrates the common trade-off between model interpretability and predictive power, especially relevant in medical applications where understanding the model's decision- making process can be crucial.

It's noteworthy that all models achieved relatively high ROC AUC scores (ranging from 0.60 to 0.83), indicating good discriminative ability across different classification thresholds. This is particularly important in medical diagnostics, where the ability to adjust the balance between sensitivity and specificity can be critical. In conclusion, our Histogram-based Gradient Boosting classifier emerged as the best model for diabetes risk identification.

Its superior performance metrics (Accuracy: 0.8243, Precision: 0.4045, Recall: 0.5545, F1-score: 0.4678, ROC AUC: 0.8287) on the test set demonstrate its robust generalization ability and balanced performance in identifying both diabetic and non-diabetic cases.

This best-performing model was implemented with carefully tuned parameters as a maximum of 200 iterations, a learning rate of 0.1, a maximum tree depth of 5, a minimum of 20 samples per leaf, and a maximum of 31 leaf nodes. These parameters strike a balance between model complexity and generalization ability. The model creates an

additive ensemble of decision trees in a forward stage-wise manner, iteratively adding models to minimize a loss function. A key advantage of this approach is the use of histograms for binning continuous features, which allows for faster training and lower memory usage compared to traditional gradient boosting methods.

The model's strong performance, particularly its high ROC AUC score, makes it the most suitable choice for our diabetes prediction task, offering the best balance between identifying potential diabetic cases and minimizing false alarms. Furthermore, the model's results help corroborate our correlation analysis and feature importance findings, providing a comprehensive view of the factors influencing diabetes risk. This alignment between different analytical approaches strengthens the reliability and interpretability of our predictions, making the model particularly valuable in a medical context where understanding the basis of risk assessments is crucial.

CHAPTER 5

Conclusion

We have successfully developed and implemented a machine learning-based system for diabetes risk prediction using blood sample data, demonstrating the potential of artificial intelligence in enhancing early disease detection and personalized healthcare. By integrating advanced machine learning techniques with an intuitive web interface, we have created a solution that bridges the gap between complex medical data and actionable health insights.

Our Histogram-based Gradient Boosting Classifier achieved an impressive accuracy of 82.43%, outperforming many existing models in the field. Through our analysis, we identified general health, BMI, age, and physical health as critical factors in diabetes risk assessment, providing data-driven validation of key risk factors and aligning with established medical knowledge.

A significant achievement of our project is the creation of a web-based tool with automated feature extraction from PDF files, marking an important step towards making advanced health risk assessment more accessible to patients and healthcare providers. Our approach moves beyond binary classification to offer probability scores and per- sonalized health insights, paving the way for more nuanced and individualized health assessments. This has important implications for medical practice, enabling early interventions by accurately identifying those at high risk of developing diabetes, guid- ing tailored preventive strategies and lifestyle modifications, and helping healthcare systems allocate resources more efficiently.

In conclusion, our machine learning-based diabetes prediction system represents a

significant step forward in the application of AI to healthcare. By combining the power of data analytics with medical expertise, we have created a tool that has the potential to transform diabetes detection and management. As we continue to refine and expand this technology, we move closer to a future where personalized, preventive healthcare becomes the norm, potentially saving millions of lives and reducing the global burden of diabetes.

Restraints and Future Approaches:

Although our findings show promise, numerous areas needing additional study and development have been noted:

Our model trained on a particular dataset, so there is data diversity. Validating these findings across other groups should be the main focus of next research to guarantee generalizability.

Including time-series data could help to understand how diabetes risk changes with time, therefore enabling more accurate long-term risk evaluations . Although our approach offers feature importance, more research on explainable artificial intelligence could help to build confidence and acceptance in healthcare environments.Real-world impact depends on research on how such tools could be efficiently included into current healthcare processes.

As we keep developing artificial intelligence for healthcare, we have to be alert about problems of data privacy, algorithmic bias, and the requirement of human oversight in medical decision-making. All things considered, our diabetes prediction technology marks a major advance in using machine learning to enhance health results.

Combining modern artificial intelligence methods with medical knowledge has produced a technology that could revolutionize early disease identification and in- dividualized healthcare administration. Looking ahead, we are thrilled about the opportunities at the junction of artificial intelligence and healthcare while still dedicated to the responsible and moral evolution of these potent technologies.

Projects like ours open the path for a future whereby technology and medicine cooperate to promote human health and well-being; the road towards AI-enhanced healthcare is just beginning. We eagerly await helping to create a healthcare envi- ronment more predictive, preventative, individualized, and participative as we keep honing and extending our approach.

References

[1] International Diabetes Federation. (2021). IDF Diabetes Atlas (10th ed.). Brussels, Belgium: International Diabetes Federation.

[2] American Diabetes Association. (2021). Standards of Medical Care in Diabetesâ2021. Diabetes Care, 44(Supplement 1), S1–S232.

[3] World Health Organization. (2016). Global Report on Diabetes. Geneva: World Health Organization.

[4] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chou- varda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal, 15, 104–116.

[5] Jensen, P. B., Jensen, L. J., & Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. Nature Reviews Genetics, 13(6), 395–405.

[6] Shanker, M. S. (1996). Using Neural Networks to Predict the Onset of Diabetes Mellitus. Journal of Chemical Information and Computer Sciences, 36(1), 35–41.

[7] Kandhasamy, J. P., & Balamurali, S. (2015). Performance Analysis of Classifier Models to Predict Diabetes Mellitus. Procedia Computer Science, 47, 45-51.

[8] Sisodia, D., & Sisodia, D. S. (2018). Prediction of Diabetes using Classification Algorithms. Procedia Computer Science, 132, 1578–1585.

[9] Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting Diabetes Mellitus With Machine Learning Techniques. Frontiers in Genetics, 9, 515.

[10] AliÄ, B., Gurbeta, L., & BadnjeviÄ, A. (2017). Machine Learning Techniques for Classification of Diabetes and Cardiovascular Diseases. In 2017 6th Mediterranean Conference on Embedded Computing (MECO) (pp. 1–4).

[11] Xie, J., Liu, Y., Zeng, X., Li, W., Tao, Z., Shao, X., & Zhu, Z. (2021). Predicting the risk of type 2 diabetes mellitus using electronic health records: A retrospective cohort study. Journal of Diabetes Science and Technology, 15(5), 1021-1031.

[12] Lai, H., Huang, H., Keshavjee, K., Guergachi, A., & Gao, X. (2022). A novel random forest model with SMOTE and random undersampling for imbalanced classifica- tion of diabetes diagnosis. Scientific Reports, 12(1), 12529.

[13] Zhang, Y., Wang, Y., Zhou, Y., Zhang, J., & Xiao, F. (2023). An optimized decision tree algorithm for diabetes prediction. IEEE Access, 11, 37944-37953.

[14] Kumar, A., Sharma, A., Arora, A., & Sharma, A. (2022). A hybrid model based on particle swarm optimization and support vector machine for diabetes prediction. International Journal of Information Technology, 14(4), 2189-2195.

[15] Chen, X., Wang, L., Liu, M., & Zhao, L. (2023). XGBoost for diabetes prediction: Ex- planatory analysis using SHAP. IEEE Journal of Biomedical and Health Informatics, 27(3), 1290-1299.

[16] Fiarni, C., Sipayung, E. M., & Maemunah, S. (2019). Analysis and prediction of diabetes complication disease using data mining algorithm. Procedia Computer Science, 161, 449-457.

[17] Zheng, T., Xie, W., Xu, L., He, X., Zhang, Y., You, M., Yang, G., & Chen, Y. (2017). A machine learning-based framework to identify type 2 diabetes through electronic health records. International Journal of Medical Informatics, 97, 120-127.

[18] Teboul, A. (2022). Diabetes Health Indicators Dataset. Kaggle. [Online]. Avail- able: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators- dataset

[19] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statisti- cal Learning: With Applications in R. New York: Springer.

[20] Lucid Software Inc. (2021). Lucidchart. [Online]. Available: https://www.lucidchart.com

[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[22] Bruce, P., & Bruce, A. (2017). Practical Statistics for Data Scientists: 50 Essential Concepts. O'Reilly Media.

[23] Little, R. J., & Rubin, D. B. (2019). Statistical Analysis with Missing Data. John Wiley & Sons.

[24] Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd.

[25] Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning: Princi- ples and Techniques for Data Scientists. O'Reilly Media.

[26] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

[27] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.

[28] Zhu, D., Li, B., Lou, P., Li, J., Zhang, H., & Wang, L. (2020). Feature extraction from medical documents: A systematic literature review. IEEE Access, 8, 31408-31424.

[29] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[30] Hosmer Jr., D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression (3rd ed.). Wiley.

[31] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[32] Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.

[33] Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Jackel, L. D., LeCun, Y., Muller, U. A., Sackinger, E., Simard, P., & Vapnik, V. (1994). Comparison of classifier methods: a case study in handwritten digit recognition. In Proceedings of the 12th IAPR International Conference on Pattern Recognition (pp. 77-82).

[34] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

[35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[36] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statisti- cal Learning: With Applications in R. Springer.

[37] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[38] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[39] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

[40] Alpaydin, E. (2020). Introduction to Machine Learning. MIT Press.

[41] GÃ©ron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

[42] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[43] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[44] Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), 37-63.

[45] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.

[46] Van Rijsbergen, C. J. (1979). Information Retrieval (2nd ed.). Butterworths.

[47] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[48] Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7), 1145-1159.

[49] Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. Remote Sensing of Environment, 62(1), 77-89.

[50] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[51] Cleveland, W. S. (1993). Visualizing Data. Hobart Press.

[52] Tufte, E. R. (2001). The Visual Display of Quantitative Information. Graphics Press.

[53] Friendly, M. (2002). Corrgrams: Exploratory displays for correlation matrices. The American Statistician, 56(4), 316-324.

[54] Tukey, J. W. (1977). Exploratory Data Analysis. Addison-Wesley.

[55] Altmann, A., ToloÅi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. Bioinformatics, 26(10), 1340-1347.

[56] Chawla, N. V. (2009). Data Mining for Imbalanced Datasets: An Overview. In Data Mining and Knowledge Discovery Handbook (pp. 875-886). Springer.

[57] Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143(1), 29-36.

A.0.1 Dataset link

Diabetes Health Indicators Dataset on Kaggle

Note

This dataset is sourced from Kaggle and contains various health indicators related to diabetes. It is used in our analysis to predict diabetes risk based on these health factors. More detailed understanding of dataset is given under Chapter 3 Methodologies-Dataset Section.

Complete project link

Project Overview

I have developed a comprehensive diabetic prediction web application. The project is structured with separate backend and frontend components, employing a modular approach for enhanced readability and maintainability.

The complete project, including all necessary files, is accessible via the following SharePoint link: Complete Project Files

A.0.2 Project Execution

To run the project, use the following commands:

Backend Commands

```
python main.py --perform-eda
```

This command includes visualization graphs.

```
python main.py
```

This command runs without generating visualization graphs.

Frontend Command

```
npm start
```

A.0.3 Project Structure and Documentation

The project's code has been systematically organized into multiple files, enhancing readability and demonstrating a structured approach. This organization is particularly beneficial for including the project in a resume or portfolio. The complete code for both backend and frontend, along with the dataset, is provided in separate zip files within the shared project link.

Academic Version

For dissertation and academic requirements, a standalone backend file has been created. This version is executable with a single command, facilitating easy demonstration and evaluation.

The project's code, as presented in the report's appendix, is for reference and is not directly executable. To run the application, please use the provided zip files and follow the execution commands listed above.