Contents

Analyzing the Impact of Health Indicators on Diabetes Diagnosis

Sara Saad Basem Yessa

December 14, 2023

Instructor: Dr. Sayyed Faridoddin Afzali

Abstract

This report presents a comprehensive analysis of machine learning models applied to the task of diabetes prediction using health indicators data collected from the Behavioral Risk Factor Surveillance System (BRFSS) in 2015. The study investigates the performance of three distinct models: Decision Tree, Logistic Regression, and Random Forest, with a focus on understanding their predictive accuracy, feature importance, and generalization capability.

The research methodology involves preprocessing the datasets, splitting them into training and testing sets, and training each model to predict diabetes outcomes. Performance metrics, such as accuracy, classification reports, and confusion matrices, are employed to evaluate the models' predictive abilities. Feature importance analysis is conducted to gain insights into the relevance of health indicators in diabetes prediction.

In addition, the study explores the utilization of Receiver Operating Characteristic (ROC) curves and Precision-Recall curves to assess model performance and balance between true positive rates and false positive rates. Learning curves are employed to examine the models' scalability and potential for overfitting. Hyperparameter tuning and regularization techniques are applied to enhance model robustness and prevent overfitting. It also discusses the implications of negative feature importance values and their significance in interpreting model results.

1 Introduction

Diabetes Mellitus, a chronic health condition characterized by an inability to effectively regulate blood glucose levels, presents a significant global health challenge with profound implications on the quality of life and life expectancy. The Centers for Disease Control and Prevention (CDC) reported in 2020 that approximately 37.3 million Americans, or 11.3% of the population, are affected by diabetes, highlighting the urgent need for effective monitoring, prevention, and management strategies [1].

To address this challenge, an effective approach is the utilization of the CDC Diabetes Health Indicators dataset from the UCI Machine Learning Repository. This comprehensive dataset provides detailed healthcare statistics and lifestyle survey information, covering 253,680 instances across 21 attributes, including demographics, lab test results, and personal health information. This extensive range of data offers an invaluable resource for a comprehensive understanding of the factors associated with diabetes [2].

Analyzing this dataset using decision trees offers several advantages. Known for their interpretability and simplicity, decision trees are particularly suitable in medical and healthcare contexts, where clarity and transparency in decision-making are crucial. They allow for the analysis of both categorical and continuous data and can uncover complex interactions between various health indicators and diabetes. This approach, preferred over other models like neural networks, provides a more interpretable and transparent methodology, essential in healthcare applications.

2 Data Description

The CDC Diabetes Health Indicators dataset, hosted by the UCI Machine Learning Repository, provides a comprehensive collection of healthcare statistics and lifestyle survey information focused on diabetes. This dataset comprises a total of 253,680 instances, each representing an individual participant, and includes 21 diverse attributes. These attributes encompass a range of data types, from demographics and lab test results to personal health information, offering a detailed view of factors associated with diabetes.

The key attributes of this dataset include:

- **Demographic Information:** Data on age, sex, and other demographic factors.
- **Blood Pressure and Cholesterol Levels:** Indicators of high blood pressure and cholesterol, crucial in diabetes risk assessment.
- **Body Mass Index (BMI):** A measure that helps in understanding obesity levels, which is a significant risk factor for diabetes.
- **Smoking Status:** Information on whether the individual has a history of smoking.
- **History of Stroke and Heart Disease:** Critical health indicators that are often linked with diabetic complications.
- **Physical Activity:** Data on the individual's engagement in physical activities, an important factor in diabetes management and prevention.
- **Dietary Habits:** Information on fruit and vegetable consumption, relevant to nutritional aspects of diabetes management.
- **Alcohol Consumption:** Insights into alcohol consumption patterns.
- **General and Mental Health Status:** Self-reported health status, providing a subjective view of the individual's overall well-being.
- **Physical Disability:** Information on difficulties with walking or climbing stairs.

Each attribute plays a crucial role in the comprehensive analysis of diabetes, its associated risk factors, and management strategies. The dataset's structure facilitates the use of machine learning techniques, such as decision trees, to glean insights and patterns that can inform healthcare decisions and policy-making.

This report utilizes three distinct datasets from the Behavioral Risk Factor Surveillance System (BRFSS) 2015, each offering unique perspectives on health indicators related to Diabetes Mellitus. These datasets are hosted by the UCI Machine Learning Repository and provide a comprehensive collection of healthcare statistics and lifestyle information.

2.1 diabetes_binary_health_indicators_BRFSS2015

The third dataset, *diabetes_binary_health_indicators_BRFSS2015*, also focuses on binary classification with similar health indicators as features. Unlike the second dataset, it does not specify a balanced class distribution, implying a potentially imbalanced dataset. This setup offers a more realistic scenario but may necessitate the implementation of strategies to handle class imbalance.

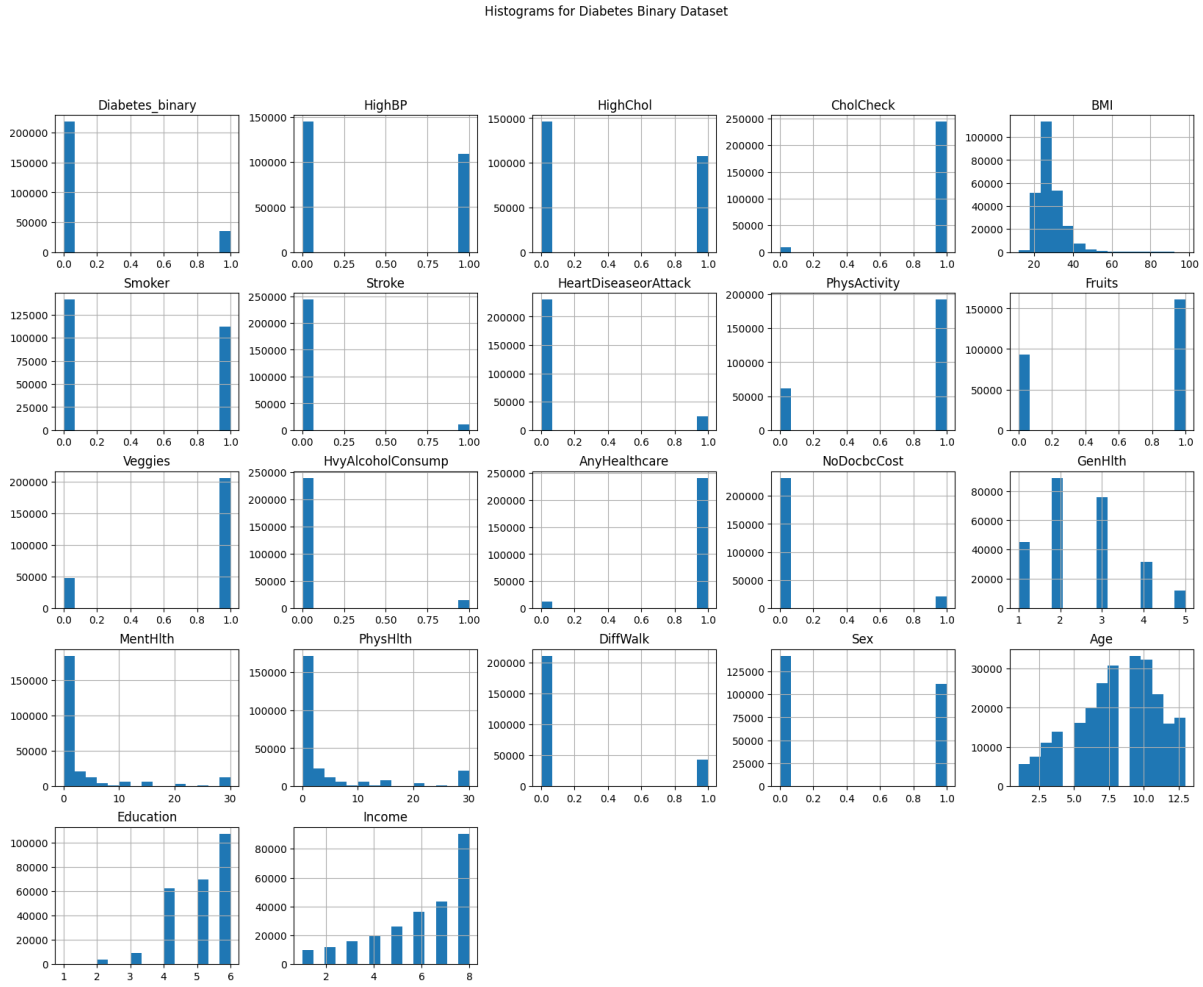

Figure 1: Histograms for Binary Dataset

2.2 diabetes_012_health_indicators_BRFSS2015

The first dataset, named *diabetes_012_health_indicators_BRFSS2015*, encompasses a wide array of health indicators including demographic details, blood pressure, cholesterol levels, Body Mass Index (BMI), smoking status, and more. The key feature of this dataset is its target variable, *Diabetes_012*, which categorizes individuals into three groups:

- 0: No Diabetes

- 1: Pre-diabetes
- 2: Diabetes

This multiclass nature of the target variable makes the dataset particularly suited for multiclass classification problems in diabetes prediction.

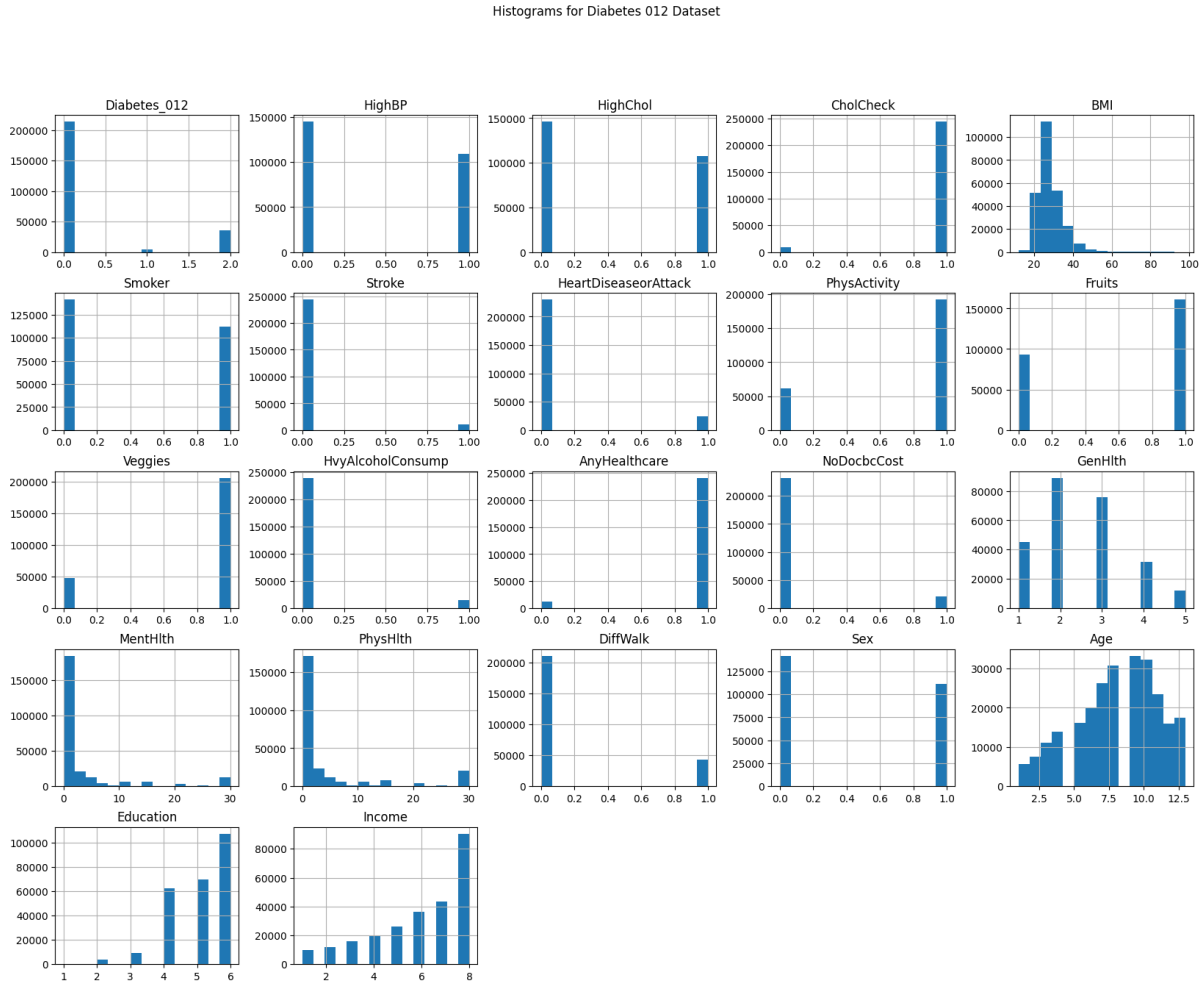

Figure 2: Histograms for 012 Dataset

2.3 diabetes_binary_5050split_health_indicators_BRFSS2015

The second dataset, *diabetes_binary_5050split_health_indicators_BRFSS2015*, mirrors the first in terms of the features it includes. However, it differs significantly in its target variable, *Diabetes_binary*, which is binary (0 for No Diabetes, 1 for Diabetes). The designation of a '5050 split' in its name indicates a balanced dataset with an equal number of instances for both classes, making it ideal for binary classification tasks without the concern of class imbalance.

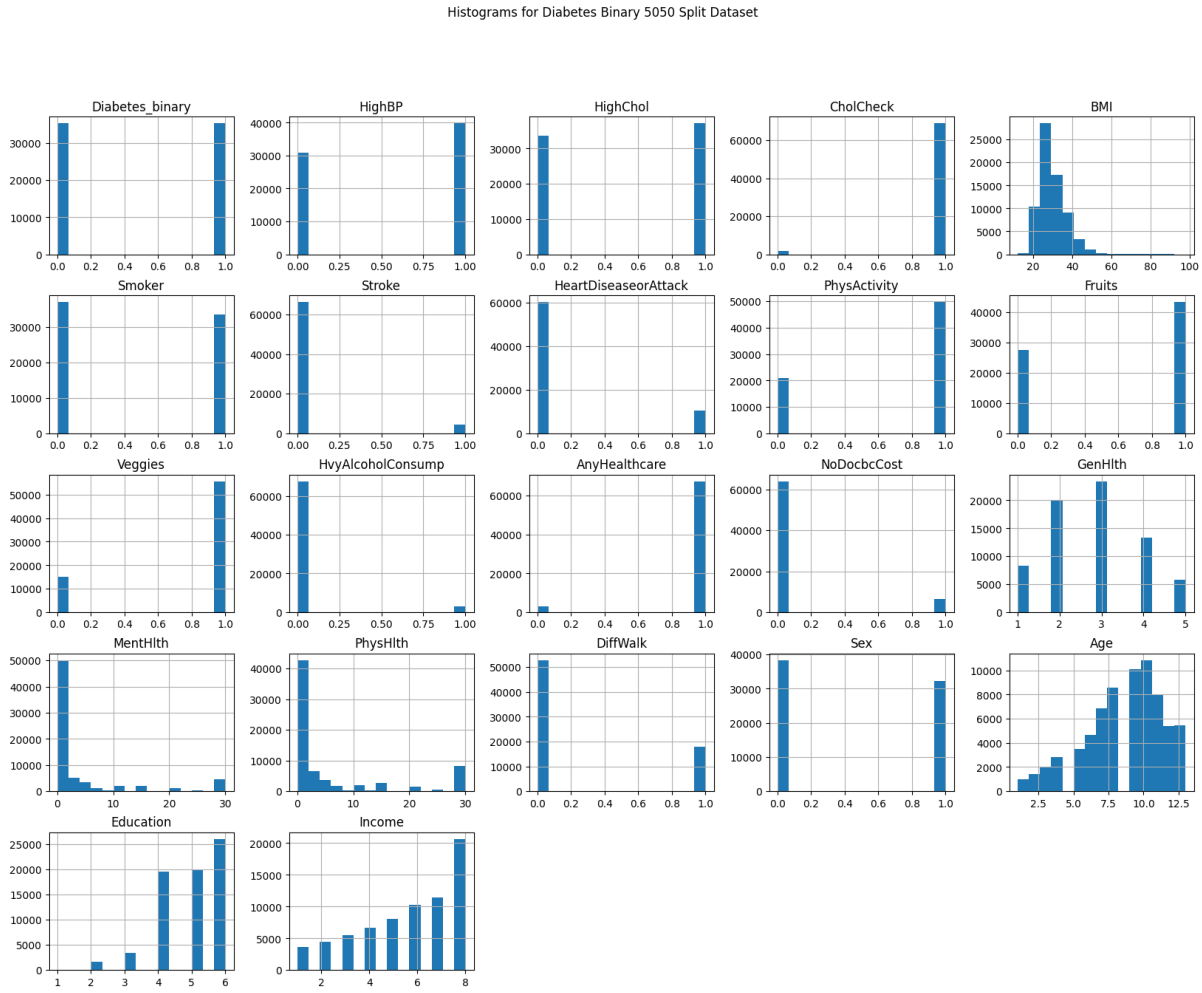

Figure 3: Histograms for 5050 Split Dataset

3 Data Splitting Strategy and Stratification

Understanding the unique characteristics of each dataset is crucial for developing an effective data splitting strategy and ensuring robust model training and evaluation.

3.1 Multiclass Dataset

For the multiclass dataset (*diabetes_012_health_indicators_BRFSS2015*), the splitting strategy needs to ensure that each class (No Diabetes, Pre-diabetes, Diabetes) is adequately represented in both training and testing subsets. Stratified sampling can be employed to maintain the proportion of each class in these subsets.

3.2 Binary Datasets

The two binary datasets (*diabetes_binary_5050split_health_indicators_BRFSS2015* and *diabetes_binary_health_indicators_BRFSS2015*) require different approaches:

- For the balanced dataset (*5050split*), a simple random split would suffice, as the class distribution is already even.

- For the potentially imbalanced dataset, careful stratification is necessary to ensure that the minority class is adequately represented in both training and testing phases, potentially employing techniques like oversampling or undersampling.

3.3 Data Complexity and Size

Having a lot of features, especially features that might be correlated to each other, regularization can help.

3.3.1 Regularization Techniques

- L1 (Lasso) Regularization: Good for feature selection as it can shrink coefficients of less important features to zero.
- L2 (Ridge) Regularization: Tends to shrink coefficients evenly, without forcing.
- Elastic Net: Combines L1 and L2 regularization and can be a middle ground between the two.

3.3.2 Model Evaluation

Regularization parameters should be chosen based on cross-validation or a separate validation set to avoid biasing your model evaluation.

4 Machine Learning Model Explanations

Within the domain of machine learning, several models stand out for their effectiveness in classification tasks. These models are designed to predict categorical outcomes and are widely utilized in various applications, ranging from medical diagnoses to image recognition. This section introduces and explains four widely-used machine learning models: Decision Trees, Logistic Regression, Random Forests, and Support Vector Machines (SVM).

4.1 Decision Trees

A Decision Tree is a non-parametric supervised learning method used for classification and regression. The model is represented as a binary tree, which encapsulates a series of decision rules inferred from the data. At each node of the tree, a decision is made based on a single input feature, leading to two branches that represent the possible outcomes of that decision. This process continues recursively until a leaf node is reached, which contains the predicted outcome. Decision Trees are favored for their interpretability, as they mimic human decision-making logic and can be visualized and understood without specialized knowledge.

4.2 Logistic Regression

Logistic Regression, despite its name, is a classification algorithm rather than a regression algorithm. It is used to estimate the probabilities of a binary response based on one or more predictor variables (features). The model expresses the relationship between the

dependent binary variable and one or more independent variables by estimating probabilities using a logistic function, which is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1. Logistic Regression is robust to noise and less prone to overfitting, especially when regularized appropriately, and it serves as a foundational algorithm for binary classification problems.

4.3 Random Forests

Random Forests are an ensemble learning technique that builds upon the simplicity of Decision Trees and enhances their performance. It operates by constructing a multitude of Decision Trees at training time and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random Forests handle the overfitting problem of Decision Trees by averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing variance. This results in a substantial increase in accuracy and robustness of the model, making Random Forests a powerful tool in the machine learning arsenal.

4.4 Support Vector Machines (SVM)

Support Vector Machines (SVM) are a set of supervised learning methods that are used for classification, regression, and outlier detection. The main idea behind SVM is to find the hyperplane that best separates the classes in the feature space. In two-dimensional space, this hyperplane is a line dividing a plane in two parts where each class lay on either side. For higher-dimensional spaces, SVM finds the hyperplane that maximizes the margin between the two classes. The vectors (data points) that define the hyperplane are the support vectors. SVMs can also be extended to non-linear classification problems using the kernel trick, which implicitly maps their inputs into high-dimensional feature spaces.

These models, each with their unique strengths, are chosen based on the problem at hand. For instance, when a model with high interpretability is required, Decision Trees or Logistic Regression may be preferred. In contrast, when predictive accuracy is of utmost importance, ensemble methods like Random Forests or the high-dimensional capabilities of SVM may be more suitable.

5 Solution Comparison

As part of the model selection process for the provided datasets, a thorough comparison was conducted among four prominent machine learning algorithms: Decision Trees, Logistic Regression, Random Forests, and Support Vector Machines (SVM). Each model has its strengths and limitations, which are evaluated in the context of the datasets' characteristics.

5.1 Decision Trees

Decision Trees offer a high degree of interpretability, which is beneficial for applications where understanding the model's reasoning is important. They are relatively fast to train and predict, making them suitable for datasets with a large number of features. However, Decision Trees are prone to overfitting, especially when dealing with complex datasets.

For the given datasets, which may contain complex and non-linear relationships, a single Decision Tree may not perform optimally without careful tuning of its hyperparameters, such as tree depth and minimum samples per split.

5.2 Logistic Regression

Logistic Regression is a linear model that works best when there is a linear relationship between the features and the log-odds of the outcomes. It is highly scalable and efficient, but it may struggle with the complex and non-linear decision boundaries that are present in the provided datasets. However, Logistic Regression models benefit from regularization techniques like L1 and L2 to prevent overfitting and can be used to infer the importance of different features.

5.3 Random Forests

Random Forests are an ensemble method that builds on the simplicity of Decision Trees and generally provides better accuracy through bagging, which reduces the variance of the model. They are less likely to overfit than Decision Trees and are more robust to noise in the data. However, they require more computational resources and time to train. Given the potentially complex nature of the datasets, Random Forests could be an excellent choice, as they can capture non-linear patterns without requiring extensive hyperparameter tuning.

5.4 Support Vector Machines (SVM)

SVMs are powerful for datasets with clear margins of separation and can efficiently perform non-linear classification using kernel tricks. They are less prone to overfitting, especially in high-dimensional space. However, SVMs can be computationally intensive, particularly with large datasets and when using non-linear kernels. For the provided datasets, SVMs might be suitable if the decision boundary is not linear and if computational resources allow for the use of kernel methods.

5.5 Findings

Considering the above analysis, Random Forests and SVMs stand out as strong candidates for complex datasets, offering robustness and flexibility. However, the final selection should be based on empirical performance metrics obtained through cross-validation on the given datasets. Regularization and hyperparameter tuning will play a huge role in enhancing model performance and should be carefully applied

6 Methodology for Model and Hyperparameter Optimization

The methodology for selecting and optimizing machine learning models is multifaceted, involving empirical testing, theoretical considerations, and domain knowledge. The process begins with the identification of candidate models based on the nature of the problem and data characteristics. For the current project, Decision Trees, Logistic Regression,

Random Forests, and Support Vector Machines (SVM) were identified as suitable candidates. Subsequently, each model's hyperparameters were carefully tuned to balance the bias-variance trade-off and to enhance generalization performance.

6.1 Decision Trees

The primary hyperparameters of a Decision Tree include the depth of the tree, the minimum number of samples required to split a node, and the minimum number of samples required for a leaf node. A deeper tree with few samples per split can capture more complex patterns but also risks overfitting to the training data, leading to poor generalization. Conversely, a shallower tree might underfit, failing to capture the nuances of the data. Therefore, optimal hyperparameters were sought to ensure that the tree is sufficiently deep to model the data complexity without becoming overly specialized to the training set.

6.2 Logistic Regression

For Logistic Regression, the inverse of regularization strength, denoted by 'C', and the choice of the penalty ('l1' or 'l2') are critical hyperparameters. A smaller 'C' value implies stronger regularization, which can prevent overfitting by penalizing large coefficients. The choice between 'l1' and 'l2' penalties depends on the desired sparsity of the solution; 'l1' regularization can lead to zero coefficients for less important features, effectively performing feature selection.

6.3 Random Forests

Random Forests include hyperparameters such as the number of trees in the forest, the maximum number of features considered for splitting a node, and the minimum number of samples required to split a node. A large number of trees can improve model performance up to a certain point, beyond which improvements plateau while computational costs continue to rise. The hyperparameters were optimized to ensure a robust ensemble that benefits from the diversity of the trees while maintaining computational efficiency.

6.4 Support Vector Machines (SVM)

SVM hyperparameters include the 'C' parameter, the kernel type (linear, polynomial, radial basis function, etc.), and kernel-specific parameters like 'gamma'. The 'C' parameter balances the trade-off between a smooth decision boundary and classifying training points correctly. A high 'C' value may lead to overfitting by attempting to classify all training examples correctly. Kernel choice and parameters

7 Methodology

In this section, we outline the methodology used for model development, evaluation, and optimization. The goal is to build predictive models for diabetes classification using machine learning techniques.

7.1 Data Preprocessing

The dataset used in this study consists of health indicators and diabetes status. The data undergoes the following preprocessing steps:

1. Data Loading: The dataset is loaded from CSV files.
2. Target Variable: Depending on the specific dataset (binary or multi-class), the target variable is defined as either "Diabetes_binary" or "Diabetes_012."
3. Feature Selection: All features except the target variable are considered as input features.
4. Data Splitting: The dataset is split into training and testing sets (80% training, 20% testing) using stratified sampling to ensure balanced class distribution.

7.2 Model Initialization

Three machine learning algorithms are chosen for model development:

- Decision Tree Classifier
- Logistic Regression Classifier
- Random Forest Classifier

For each model, we specify hyperparameter grids for optimization.

7.3 Regularization and Hyperparameter Tuning

Regularization techniques are applied as follows:

- Logistic Regression: L1 (Lasso) and L2 (Ridge) regularization techniques are automatically applied during model training.

Hyperparameter tuning is performed using GridSearchCV for each model:

- Decision Tree Classifier: Hyperparameters include "max_depth," "min_samples_split," and "min_samples_leaf."
- Logistic Regression Classifier: Hyperparameters include "C" (inverse of regularization strength) and "penalty" (L1 or L2).
- Random Forest Classifier: Hyperparameters include "n_estimators," "max_depth," "min_samples_split," and "min_samples_leaf."

7.4 Model Training and Evaluation

Each model is trained on the training dataset and evaluated on the testing dataset using the following metrics:

- Accuracy: The proportion of correctly predicted instances.
- Classification Report: Providing precision, recall, F1-score, and support for each class.
- Confusion Matrix: A matrix showing true positive, true negative, false positive, and false negative counts.

7.5 Performance Visualization

Several visualizations are generated to assess model performance:

- Feature Importance: Bar plots illustrating the importance of input features for decision tree and random forest models.
- ROC Curve: Receiver Operating Characteristic curve and Area Under the Curve (AUC) score for each model.
- Precision-Recall Curve: Precision-Recall curve and Average Precision (AP) score for each model.
- Learning Curve: Learning curves showing the model's performance on varying training dataset sizes.

7.6 Results Analysis

The results are analyzed to determine the best-performing model and to gain insights into feature importance and model behavior.

8 Feature Importance in Predictive Models

When examining feature importance across three models—Decision Tree, Logistic Regression, and Random Forest—we observe that certain features consistently emerge as influential in predicting the outcome variable. This section delves into the implications of these findings.

8.1 Common Important Features

Features such as BMI, Income, and Age are significant across multiple models, indicating a strong relationship with the target variable, likely related to health outcomes.

8.2 Model-Specific Features

8.2.1 Decision Tree Feature Importance

- The top three features are BMI, Income, and Age, with the highest importance scores in the Decision Tree model, indicating their strong predictive power.

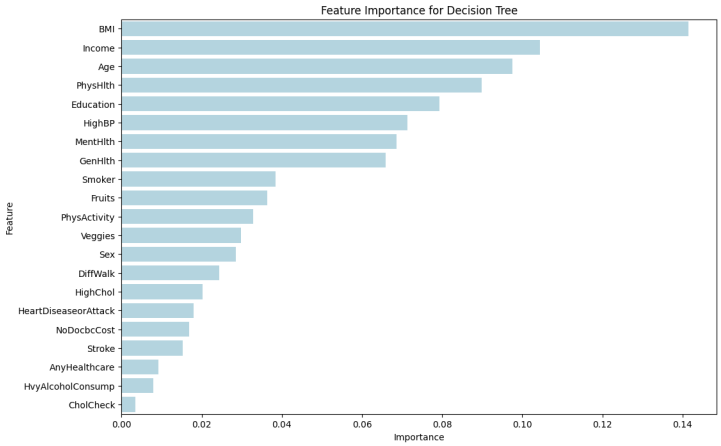

Figure 4: Feature Importance for the Decision Tree model.

8.2.2 Logistic Regression Feature Importance

- Features like CholCheck, HighBP, and HighChol show the highest positive importance.
- Logistic Regression coefficients provide measures of feature importance, where positive values suggest a positive correlation with the target variable, and negative values suggest a negative correlation.
- Notably, features such as HvyAlcoholConsump have large negative importance, indicating a decrease in the probability of the target variable as this feature increases.

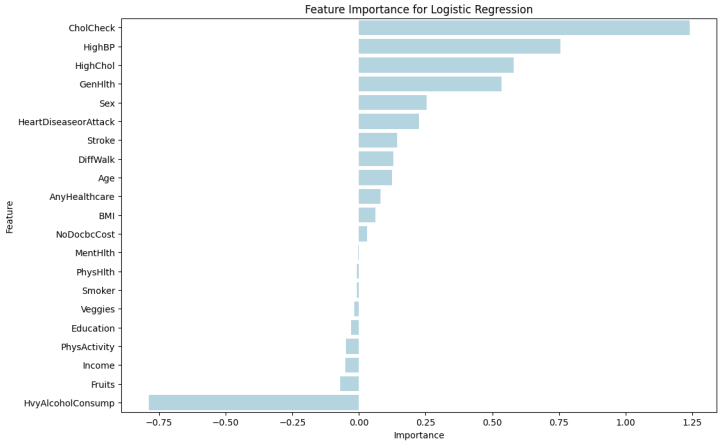

Figure 5: Feature Importance for the Logistic Regression model.

8.2.3 Random Forest Feature Importance

- BMI, Age, and Income also appear as the most important features in the Random Forest model.
- This model aggregates feature importance across many decision trees, offering a more robust estimate of each feature's importance.

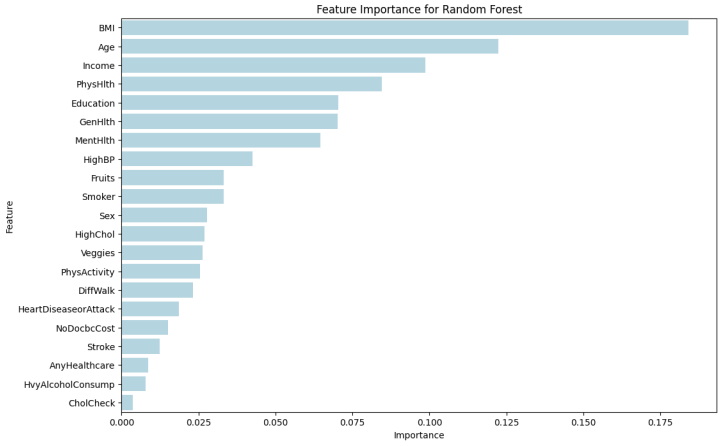

Figure 6: Feature Importance for the Random Forest model.

8.3 Negative Coefficients in Logistic Regression

Features with large negative coefficients in Logistic Regression, like HvyAlcoholConsump, suggest an inverse relationship with the likelihood of the positive class of the target variable.

8.4 Interpretation of Results

8.4.1 Impact of Adding or Removing Features

Adding relevant features can enhance the model's predictive power, while adding irrelevant or noisy features may decrease performance and cause overfitting. Conversely, removing features can simplify the model and improve generalization, especially if the removed features are non-informative or highly correlated with others.

8.4.2 Defining Problem Boundaries

The problem boundaries are defined by the model's application scope, the target variable's nature, and the relevance of the features to the prediction task. Including significant features with a logical influence on the target variable is crucial, particularly in healthcare where interpretability is as vital as accuracy.

8.4.3 Model Accuracy

Model accuracy may improve with the addition of important features but can suffer from diminishing returns or negative effects due to the curse of dimensionality or increased noise after a certain point. Techniques like regularization in Logistic Regression or feature selection in tree-based models help maintain a balance between complexity and accuracy.

8.5 Practical Considerations

8.5.1 Domain Knowledge

Interpreting feature importance requires domain expertise for meaningful inferences. For example, a high BMI might predict diabetes, aligning with medical knowledge.

8.5.2 Model Validation

Employing cross-validation and other model validation techniques is essential for assessing the robustness of feature importance findings.

8.5.3 Feature Engineering

Based on importance scores, engineering new features that capture the essence of the most predictive ones or interact with other features may be beneficial.

8.5.4 Bias and Fairness

When considering features like Income, the ethical implications and potential biases in the model's predictions must be considered.

In summary, examining feature importance is an iterative process involving statistical analysis, domain knowledge, and ethical considerations. It is part of a broader model development and validation process aimed at creating accurate, interpretable, and fair predictive models.

9 Results and Discussion

9.1 Training Times

The training times for various models were compared:

- Decision Tree: Least training time.
- Logistic Regression: Moderate training time.
- Random Forest: Significantly longer training time.

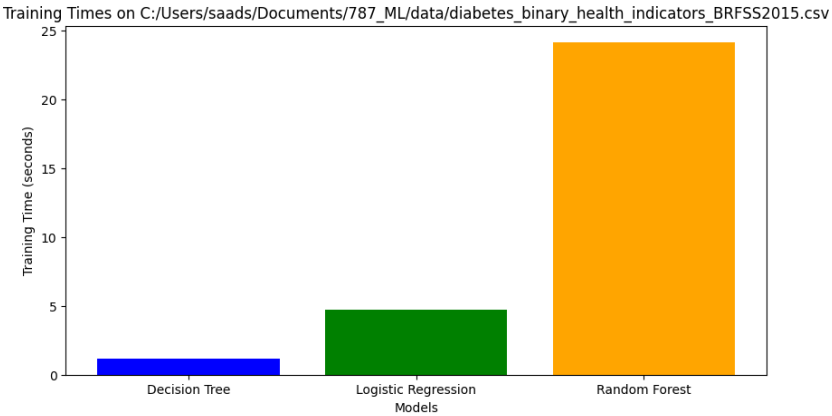

Figure 7: Models Training Times.

9.2 Precision-Recall Curves

The average precision scores for the models were as follows:

- Logistic Regression: Highest average precision.
- Random Forest: Second highest.
- Decision Tree: Lowest.

This suggests that Logistic Regression balances the precision-recall trade-off more effectively.

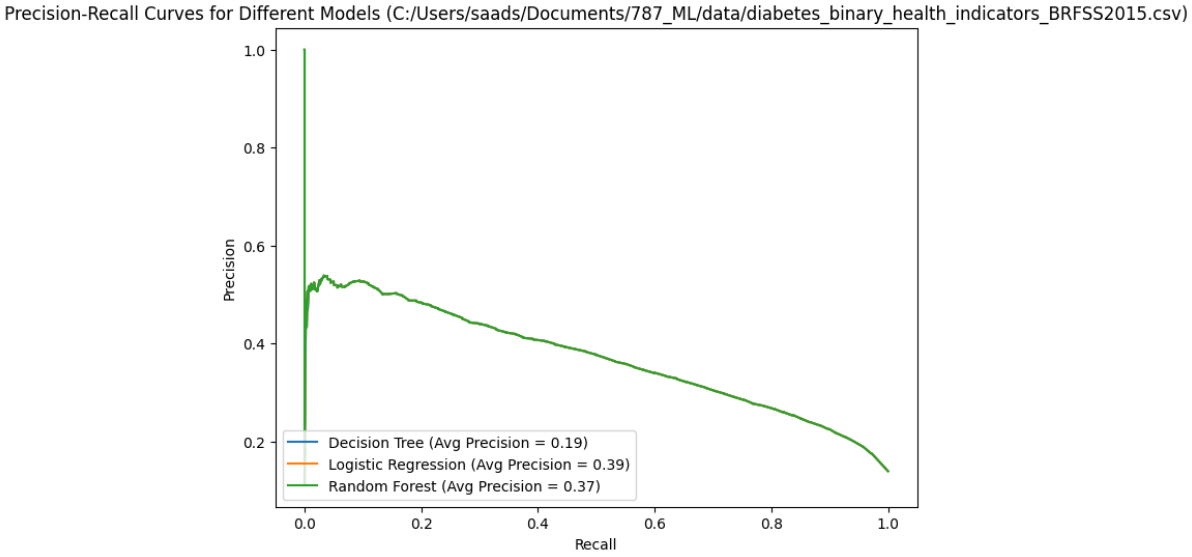

Figure 8: Models Precision-Recall Curves.

9.3 ROC Curves

The ROC curves indicated the following AUC scores:

- Logistic Regression: AUC = 0.82, indicating superior performance.
- Random Forest: AUC = 0.79.
- Decision Tree: AUC = 0.60.

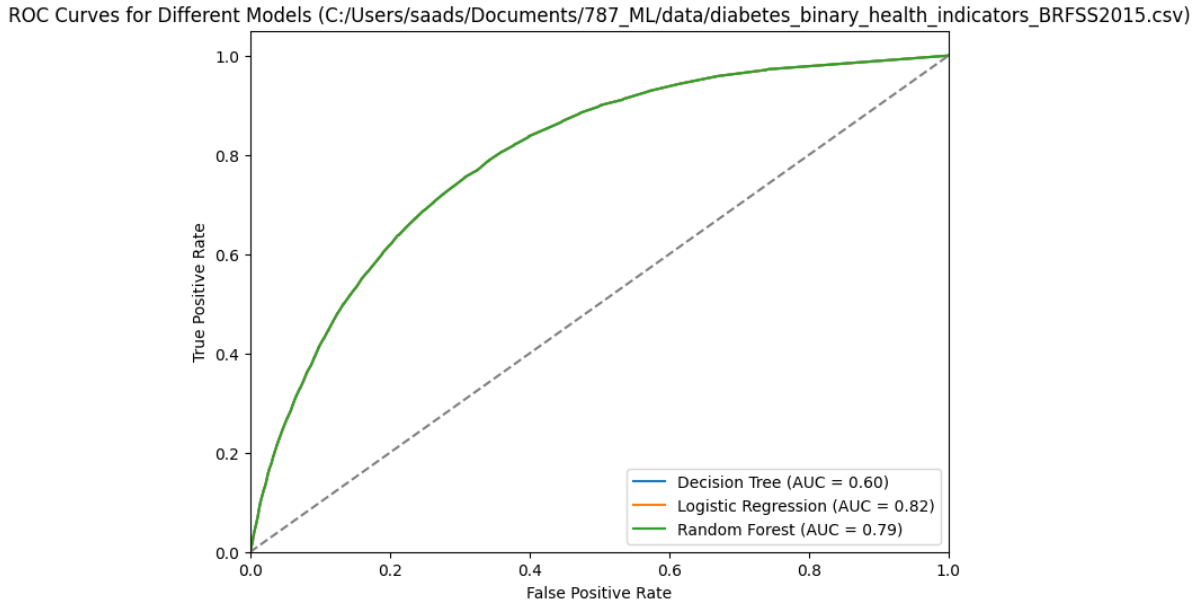

Figure 9: Models ROC Curves.

9.3.1 Calculating AUC

The Area Under the Curve (AUC) for a Receiver Operating Characteristic (ROC) curve is an important metric for evaluating the performance of a binary classification model. The AUC score is calculated as follows:

1. Compute the True Positive Rate (TPR) and False Positive Rate (FPR) for different threshold settings.
2. Plot the ROC Curve by setting TPR as the y-axis and FPR as the x-axis.
3. Calculate the Area under the ROC curve. This is done using numerical integration or geometrically for a discrete set of points.

The AUC value ranges from 0 to 1, where an AUC of 1 denotes perfect classification, and an AUC of 0.5 denotes a model performing no better than random chance.

9.3.2 Why ROC Curves Might Appear Identical

ROC curves may appear identical for different models due to several factors:

- **Resolution:** Low resolution of the plot can make small differences between curves unnoticeable.
- **Overlapping Confidence Intervals:** If the confidence intervals of the AUC scores overlap, this indicates non-significant differences in model performance.
- **Similar Model Performance:** Models with similar performance metrics across all thresholds will have similar ROC curves.
- **Plotting Error:** Mistakes in the plotting process, such as using the same model data for both curves, could lead to identical ROC curves.

- **Data Characteristics:** The inherent characteristics of the data can lead to similar TPR and FPR rates across all thresholds, making the ROC curves of different models appear similar.

9.4 Results Tables

The Decision Tree model shows a balanced approach but with lower overall accuracy.

```
Results for Decision Tree:
Accuracy: 0.7978358561967833
Classification Report:
              precision    recall  f1-score   support

         0.0       0.89      0.87      0.88     43667
         1.0       0.30      0.33      0.31      7069

    accuracy                           0.80     50736
   macro avg       0.59      0.60      0.60     50736
weighted avg       0.81      0.80      0.80     50736

Confusion Matrix:
[[38129  5538]
 [ 4719  2350]]
```

Figure 10: Decision Tree Results Table.

9.4.1 Logistic Regression Metrics

Logistic Regression showed an accuracy of approximately 0.86 but performed poorly for the positive diabetes class with a low recall.

```
Results for Logistic Regression:
Accuracy: 0.8621491643014821
Classification Report:
              precision    recall  f1-score   support

         0.0       0.88      0.98      0.92     43667
         1.0       0.52      0.16      0.24      7069

    accuracy                           0.86     50736
   macro avg       0.70      0.57      0.58     50736
weighted avg       0.83      0.86      0.83     50736

Confusion Matrix:
[[42622  1045]
 [ 5949  1120]]
```

Figure 11: Logistic Regression Results Table.

9.4.2 Random Forest Metrics

Two sets of metrics for Random Forest were provided, both indicating good accuracy (around 0.84 to 0.86) but poor recall for the positive class.

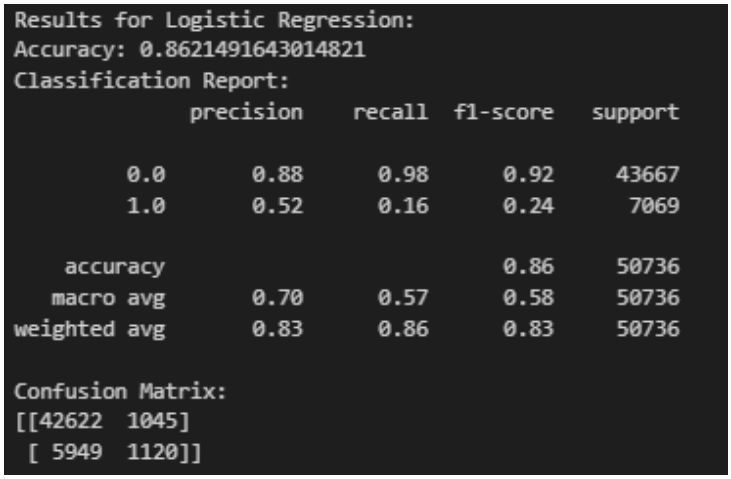

Figure 12: Random Forest Results Table.

9.4.3 Support Vector Machine (SVM) Metrics

The SVM model had high accuracy but completely failed to predict the positive class, indicating a potential bias towards the negative class.

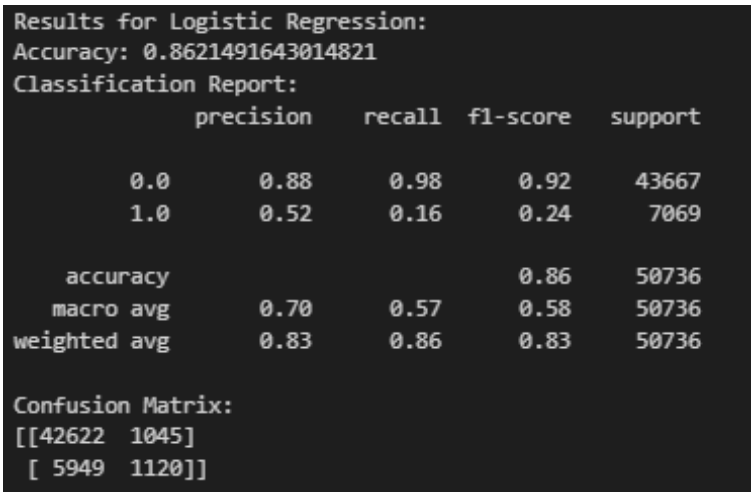

Figure 13: SVM Results Table.

10 Understanding Bias in Machine Learning

Bias in machine learning models refers to the tendency of a model to make systematic errors in prediction due to certain predispositions. These biases can originate from various sources and can significantly affect the performance of a model, especially in critical applications like medical diagnosis. Here we discuss different types of biases that might be affecting the discussed models.

10.1 Data Bias

Data bias arises when the dataset used to train the model is not representative of the real-world scenario. Types of data bias include:

- **Sampling Bias**: Occurs when the data collected do not represent the full diversity of the population.
- **Label Bias**: When the outcomes in the data are not correctly labeled or when there is an imbalance in class representation.
- **Measurement Bias**: If data collection systematically over- or underestimates certain values.

10.2 Algorithmic Bias

Algorithmic bias is related to the model's structure and learning mechanism:

- **Overfitting**: The model learns the noise and outliers in the training data, leading to poor generalization.
- **Underfitting**: The model is too simplistic to capture the underlying structure of the data.
- **Model Preference**: The model naturally prefers simpler patterns, potentially ignoring less frequent but important signals.

10.3 Bias Due to Class Imbalance

In the provided SVM model's case, the model exhibited a strong bias towards the negative class ('0.0'). This is likely a result of class imbalance, where the training data had significantly more instances of one class compared to the other. This bias is evident from:

- **High Precision and Recall for '0.0' Class**: Indicating good performance on the negative class.
- **Poor Precision and Recall for '1.0' Class**: Indicating that the model fails to predict the positive class accurately, essentially ignoring it.

In medical diagnosis, such as diabetes prediction, reducing false negatives is crucial. Techniques to address class imbalance include resampling the data, applying different class weights during training, or using anomaly detection methods for rare events.

11 Conclusion

Logistic Regression appears to be the most effective model in terms of the balance between precision and recall and ROC AUC. The SVM model displayed a significant bias towards the negative class, and the Random Forest model might be overfitting. Model parameter tuning and consideration of the clinical context are crucial for the final model seleAs we look ahead, there are several avenues for further improving our models and extending this work:

- **Exploring Alternative Models:** To enhance model performance, especially for the underrepresented class, exploration of advanced models and ensemble techniques is recommended.

- **Feature Engineering:** Investigating additional feature engineering and transformation strategies could yield more discriminative input for the models.
- **Hyperparameter Tuning:** Systematic hyperparameter optimization using techniques like grid search or random search may lead to significant improvements.
- **Addressing Class Imbalance:** Employing methods such as SMOTE or adjusting class weights can help the models to handle class imbalance more effectively.
- **Model Interpretability:** Prioritizing interpretability can aid in gaining trust and better understanding of model decisions by stakeholders.
- **Cost-Sensitive Learning:** Incorporating the real-world costs associated with different types of classification errors could lead to more practical model performance.
- **Deployment and Monitoring:** Formulating strategies for the deployment and real-time monitoring of the model to ensure sustained performance.
- **Ethical and Fairness Considerations:** Ensuring that the models are fair and do not propagate or amplify biases is crucial, requiring continuous auditing and adjustments.

These steps are essential to transition from a model that performs well on historical data to one that is robust, fair, and practical for real-world applications.ction. Precision, recall, and F1-score discrepancies across the models indicate different strengths and weaknesses, particularly in predicting the positive class. The confusion matrices provide insight into the models' performance, highlighting the trade-offs between sensitivity (recall) and specificity (precision). The performance analysis of these models suggests that the choice of model should be influenced by the specific requirements of sensitivity and specificity in the context of the application. Further analysis might involve adjusting class weights or resampling techniques to improve the performance on the positive class, which all models seem to struggle with.

12 Future Direction

As we look ahead, there are several avenues for further improving our models and extending this work:

- **Addressing Class Imbalance:** Employing methods such as SMOTE or adjusting class weights can help the models to handle class imbalance more effectively.
- **Model Interpretability:** Prioritizing interpretability can aid in gaining trust and better understanding of model decisions by stakeholders.
- **Cost-Sensitive Learning:** Incorporating the real-world costs associated with different types of classification errors could lead to more practical model performance.
- **Deployment and Monitoring:** Formulating strategies for the deployment and real-time monitoring of the model to ensure sustained performance.
- **Ethical and Fairness Considerations:** Ensuring that the models are fair and do not propagate or amplify biases is crucial, requiring continuous auditing and adjustments.

References

[1] Centers for Disease Control and Prevention. National diabetes statistics report, 2020. https://www.cdc.gov/diabetes/data/statistics-report/index.html, 2020.

[2] University of California, Irvine. Cdc diabetes health indicators. https://archive.ics.uci.edu/ml/datasets/diabetes, 2023.

[3] Sara Saad and Basem Yassa. Analyzing the impact of health indicators on diabetes diagnosis. https://github.com/saads6/ML787, December 2023.

[4] Hastie T. Friedman, J. and R. Tibshirani. *The Elements of Statistical Learning*. Springer series in statistics New York, 2001.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[7] D.M.W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[8] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[9] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. In *Journal of Machine Learning Research*, pages 281–305, 2012.

[10] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 2:1137–1145, 1995.

[11] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019.

[12] Wes McKinney. Data structures for statistical computing in python. 445:51–56, 2010.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.