

# Diabetes Prediction System Report

## 1. Introduction

Diabetes is a chronic health condition that affects how your body turns food into energy. The purpose of this project is to develop a machine learning model to predict the likelihood of a person having diabetes based on various health metrics. Early detection of diabetes can significantly improve the management and treatment of the disease, thus reducing the risk of severe health complications.

## 2. Requirements

### Functional Requirements

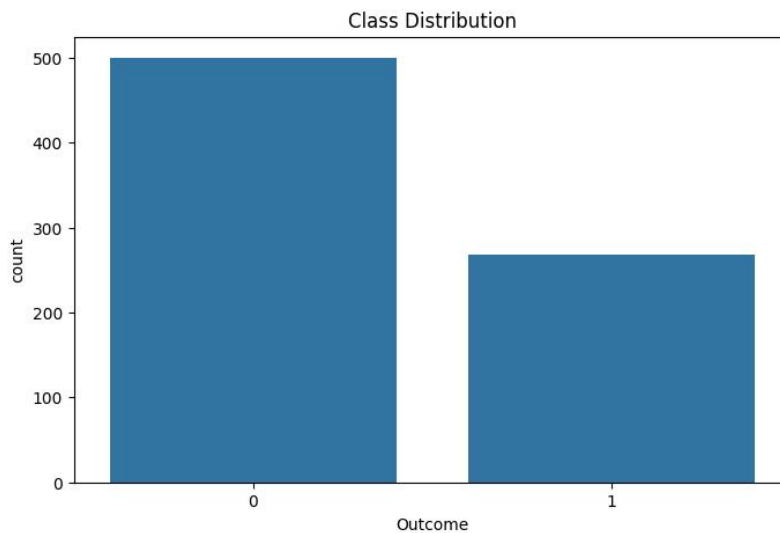
1. Data Preprocessing: Clean and preprocess the data to handle missing values, duplicates, and outliers.
2. Model Training: Develop and train a machine learning model using the processed data.
3. Model Evaluation: Evaluate the model's performance using appropriate metrics such as accuracy
4. Prediction: Predict diabetes on new, unseen data.

## 3. Technologies Used

1. Python: The primary programming language used for data analysis and model development due to its extensive libraries and ease of use.
2. Pandas and NumPy: Libraries for data manipulation and analysis.
3. Matplotlib and Seaborn: Libraries for data visualization.
4. Scikit-Learn: A machine learning library used for model development, training, and evaluation.

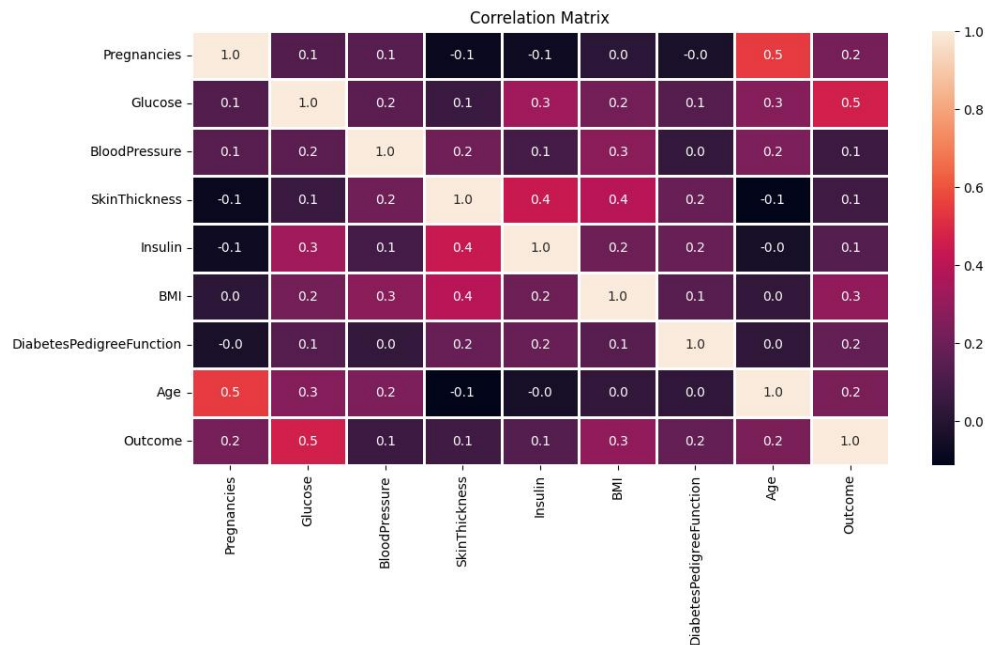
## 4. Methodology

1. **Data Collection & Analysis:** The dataset used is <https://www.kaggle.com/datasets/mathchi/diabetes-data-set>, containing various health metrics of individuals. We analyzed our data.



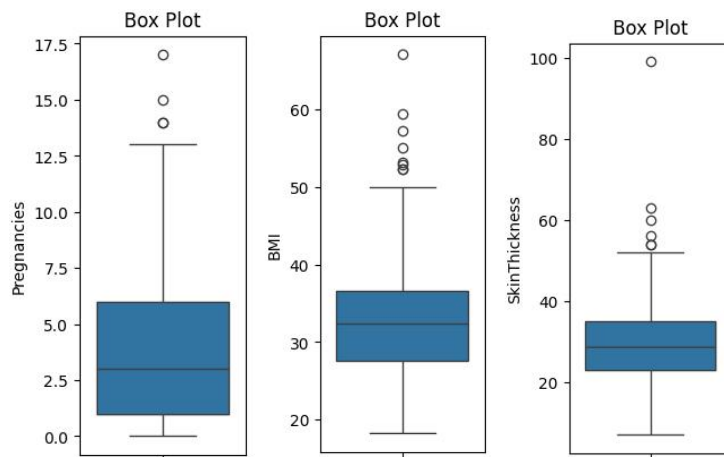
The graph above shows the number of diabetic and non-diabetic persons.

We created a heat-map to visualize the correlation between Features:



We can see in the chart that Glucose and Diabetes are highly correlated while blood pressure, skin thickness and insulin have less effect on onset of diabetes.

Then we created some box plots to see if we have any outliers.



We can see in the box plots that we have some outliers. We will treat these outliers in data cleaning step.

## 2. Data Cleaning:

Checked for duplicates and missing values. There were none!

Check for zero values. There were many. Specially skin thickness and insulin had highest number of Zero values 227 and 374 respectively.

We replaced the zero values with null. And then replaced the null values with mean. But the accuracy was not good enough.

So we replaced the null values using KNN method using 10 nearest neighbors. The accuracy optimized

## Handling Outliers

Handled outliers using the Inter-quartile Range (IQR) method.

### Objective

The objective of this analysis is to detect and remove outliers from a dataset using the Interquartile Range (IQR) method. Outliers are extreme values that differ significantly from the rest of the data and can negatively impact the accuracy of data analysis. Removing these outliers ensures more accurate and reliable results.

### Methodology

#### Calculating Quartiles and IQR:

The analysis begins with calculating the 25th percentile (Q1) and the 75th percentile (Q3) of the dataset. These quartiles divide the data into four equal parts.

The Interquartile Range (IQR) is then computed by subtracting Q1 from Q3. The IQR represents the middle 50% of the data and is used to identify outliers.

#### Determining Outlier Thresholds:

The next step involves determining the thresholds for detecting outliers using the IQR. The upper limit is set as Q3 plus 1.5 times the IQR, and the lower limit is set as Q1 minus 1.5 times the IQR.

These thresholds help identify data points that are significantly higher or lower than the majority of the data.

### Identifying Outliers:

Data points that fall outside the calculated upper and lower limits are identified as outliers. This identification process checks if each data point falls below  $Q1 - 1.5 * IQR$  or above  $Q3 + 1.5 * IQR$ .

Identifying these outliers is crucial because they can skew the results of the analysis if not addressed.

### Removing Outliers:

Once identified, outliers are removed to create a new, refined dataset. This refined dataset includes only the data points within the established thresholds, ensuring that the remaining data is more representative of the overall population.

Removing outliers helps in achieving a more accurate and reliable analysis by eliminating the influence of extreme values.

### Comparing Dataset Sizes:

The analysis then compares the sizes of the original dataset and the refined dataset to understand the impact of outlier removal. This comparison involves counting the number of data points in both datasets.

The number of outliers removed is determined by the difference between the size of the original dataset and the size of the refined dataset.

3. **Data Splitting:** Split the data into training and testing sets using an 80-20 ratio.

4. **Model Selection:** The performance of 3 models was evaluated using cross-validation scores. Logistic Regression outperformed both Random Forest and Decision Tree classifiers in terms of accuracy.

5. **Model Development:** Developed a Logistic Regression model.

6. **Hyperparameter Tuning:** Grid search was performed to tune hyperparameters for Logistic Regression, resulting in improved performance.

7. **Model Evaluation:** Evaluated the model's performance using accuracy

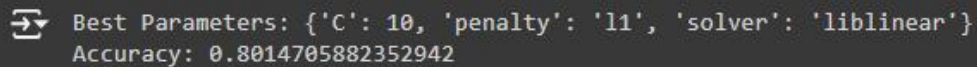
## 4.6 Hyperparameter Tuning

There are a lot of parameters in logistic regression. But most of them are often set to default and are not changed. The parameters we added are

- **C :** It is the regularization strength parameter. Regularization strength in the context of machine learning, particularly in models like Logistic Regression, refers to a technique used to prevent overfitting. We tried different values like 0.01, 0.1, 1, 10, 100. But value of 10 gave best results.
- **Penalty:** It is the regularization method that will be used. In our case The model will try both regularization methods to find the optimal one.
- **Solver:** It is the algorithm used. We used 'liblinear' which is a solver suitable for small datasets and supports both L1 and L2 regularization

## 5. Results

Performance of Logistic Regression after hyperparameter tuning:  
Accuracy: 0.8

A terminal window with a dark background and light green text. It displays the output of a hyperparameter tuning process, showing the best parameters found and the resulting accuracy.

```
Best Parameters: {'C': 10, 'penalty': 'l1', 'solver': 'liblinear'}  
Accuracy: 0.8014705882352942
```

## 6. Future Work

Potential enhancements and improvements include:

1. Feature Engineering: Create new features that might have a stronger correlation with the target variable.
3. Additional Algorithms: Explore other machine learning algorithms such as Gradient Boosting, Support Vector Machines, or Neural Networks for potentially better performance.
4. Data Augmentation: Gather more data to improve the robustness and generalizability of the model.
5. Deployment: Develop a user-friendly interface for the model and deploy it as a web or mobile application for broader accessibility.

## 7. Summary

This study demonstrates the application of machine learning techniques in predicting diabetes using health metrics. The models developed, particularly Logistic Regression with optimized parameters, show promising results in accurately predicting diabetes. Further refinement and validation of these models could potentially enhance their utility in clinical settings for early detection and management of diabetes.

## 8. References

- Pandas Documentation: <https://pandas.pydata.org/pandas-docs/stable/>
- NumPy Documentation: <https://numpy.org/doc/>
- Matplotlib Documentation: <https://matplotlib.org/stable/contents.html>
- Seaborn Documentation: <https://seaborn.pydata.org/>
- Scikit-Learn Documentation: <https://scikit-learn.org/stable/>
- RandomForestClassifier Documentation: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>