Diabetic Diagnosis System - Detailed Report

1. Introduction

This report details the process of building a machine learning model to predict whether a patient has diabetes based on certain diagnostic measurements. The project utilizes the PIMA Indians Diabetes Database. The primary goal is to achieve a model with high accuracy and, more importantly, a high recall for the positive class (diabetic), to minimize false negatives.

2. Data Loading and Initial Analysis

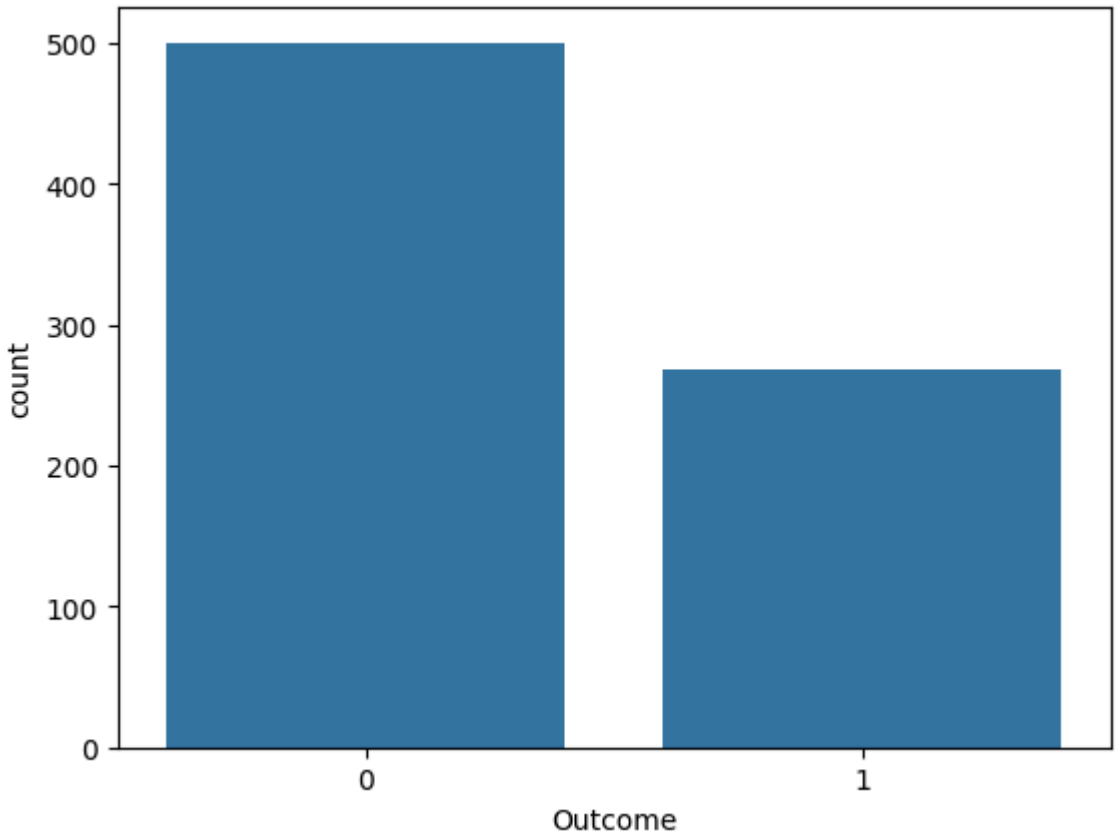
- The dataset is loaded from `diabetes.csv`.
- Initial analysis is performed using `df.head()`, `df.describe()`, and `df.info()`.
- The dataset contains 768 entries and 9 columns.
- There are no null values, but some columns like 'Glucose', 'BloodPressure', 'SkinThickness', and 'BMI' have zero values, which are practically impossible and are treated as missing values.

3. Data Preprocessing

- **Handling Missing Values:** The zero values in 'Glucose', 'BloodPressure', 'SkinThickness', and 'BMI' are replaced with the mean of their respective columns. The zero values in 'Insulin' are also replaced by the mean of the column.
- **Feature Scaling:** The features (X) are scaled using `StandardScaler` to ensure that all features contribute equally to the model's performance.

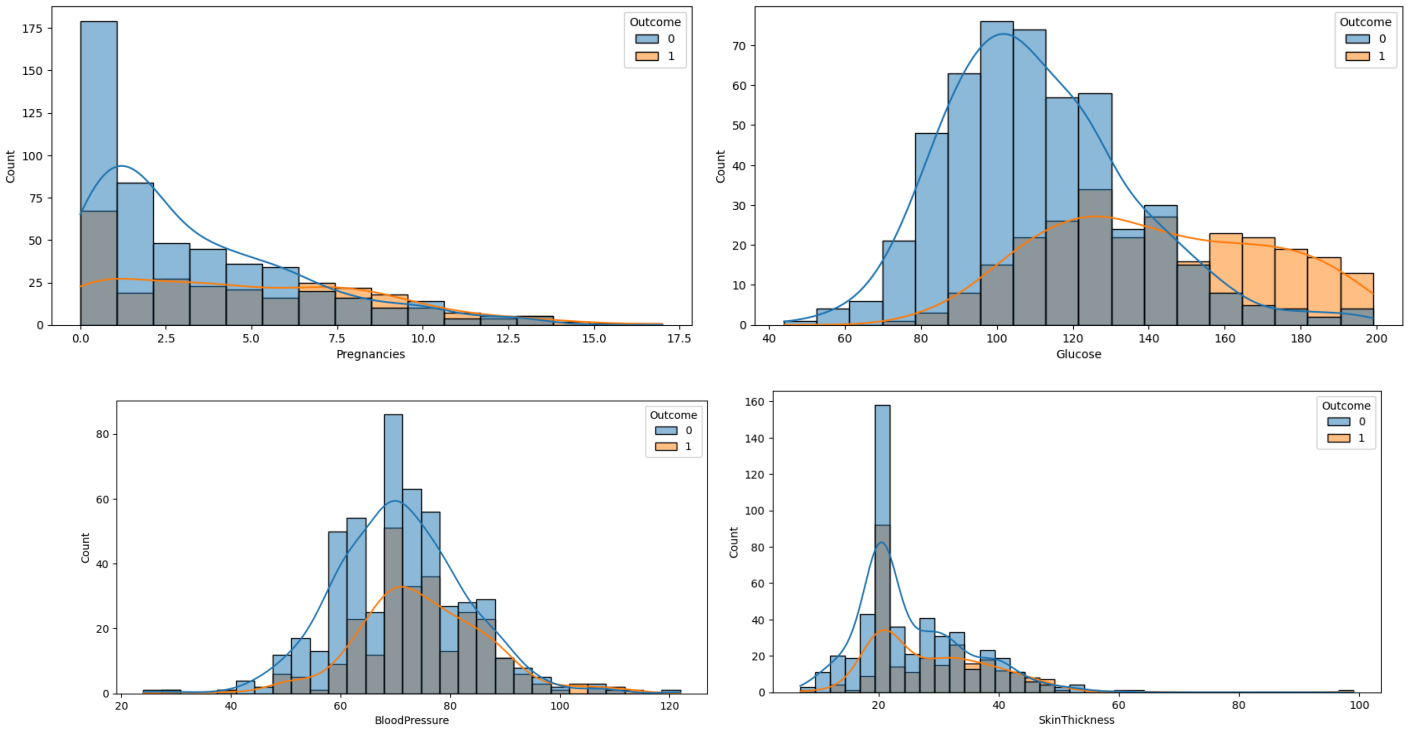
4. Exploratory Data Analysis (EDA)

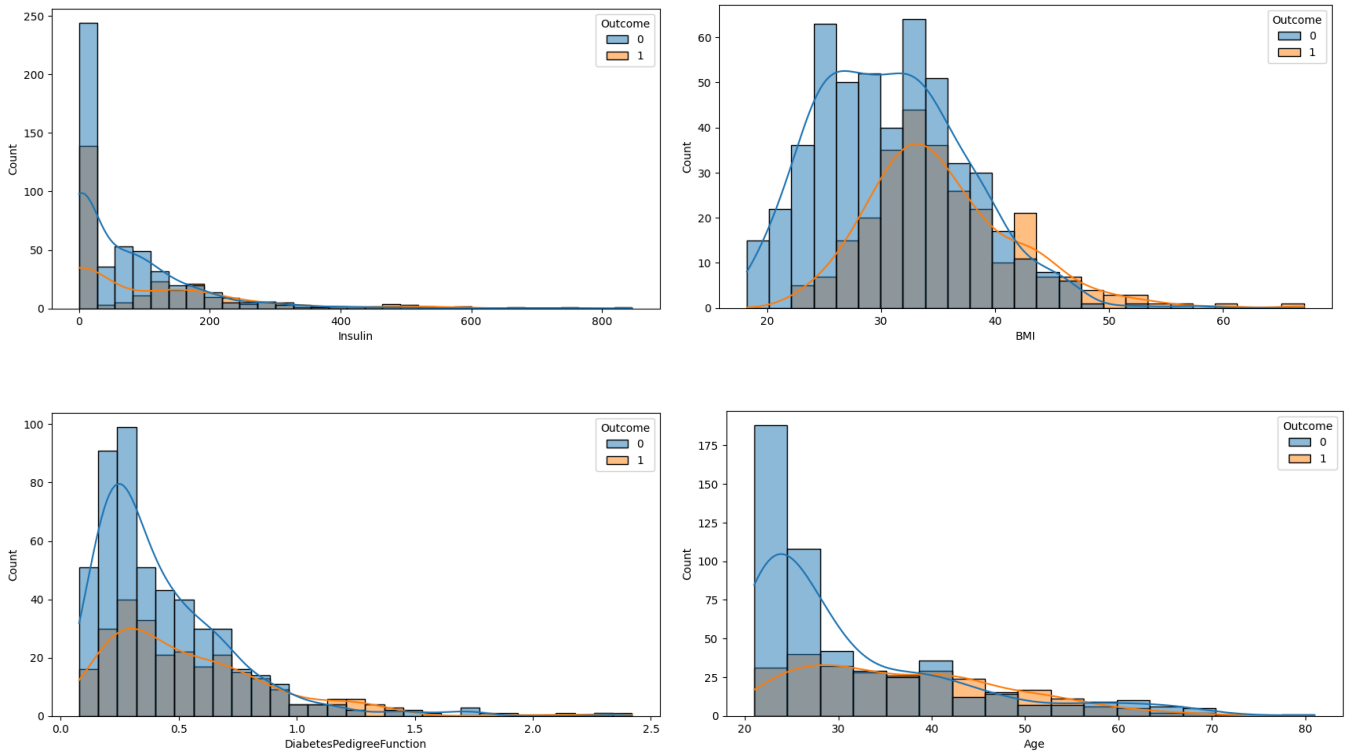
4.1. Outcome Distribution



- The countplot shows that the dataset is imbalanced, with more non-diabetic patients than diabetic ones.

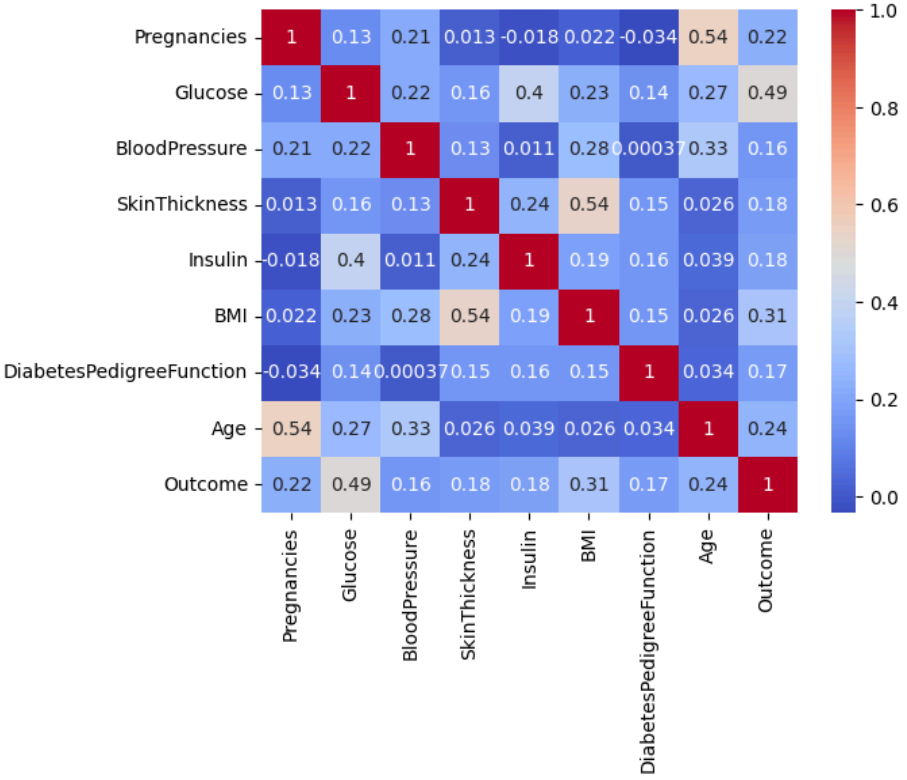
4.2. Feature Distribution





- The histograms show the distribution of each feature for both diabetic and non-diabetic patients.
- Higher values of Glucose, BMI, and Age show a strong indication of diabetes.
- Pregnancies and DiabetesPedigreeFunction also seem to be good predictors.
- BloodPressure and SkinThickness do not seem to be as strong predictors compared to others.

4.3. Correlation Matrix



- The heatmap shows the correlation between different features.
- 'Glucose' has the strongest correlation with 'Outcome', followed by 'BMI', 'Age', and 'Pregnancies'.
- There is a noticeable correlation between 'Age' and 'Pregnancies', and between 'BMI' and 'SkinThickness'.

5. Model Training and Evaluation

The data is split into training (80%) and testing (20%) sets.

5.1. Logistic Regression

- A Logistic Regression model is trained on the scaled training data.
- **Results:**
  - Accuracy: 76.62%
  - Confusion Matrix:
    [[83 16]
     [20 35]]
  - Classification Report:

	precision	recall	f1-score	support
0	0.81	0.84	0.82	99
1	0.69	0.64	0.66	55

5.2. Random Forest Classifier

- A Random Forest Classifier is trained.
- **Results:**
  - Accuracy: 74.03%
  - The performance is slightly worse than Logistic Regression.

- A balanced Random Forest Classifier (`class_weight='balanced'`) is also trained to handle the data imbalance.
- **Results (Balanced):**
  - Accuracy: 73.38%
  - This did not significantly improve the model's performance.

5.3. Random Forest with GridSearchCV

- To find the best hyperparameters for the Random Forest model, `GridSearchCV` is used with `recall` as the scoring metric.
- **Best Hyperparameters:** {'criterion': 'entropy', 'max_depth': 20, 'n_estimators': 100}
- **Results (Best Model):**

- Accuracy: 77.92%
- Confusion Matrix:
  [[80 19]
   [15 40]]
- Classification Report:

	precision	recall	f1-score	support
0	0.84	0.81	0.82	99
1	0.68	0.73	0.70	55

- This model gives the best recall score for the positive class (diabetic) and the highest accuracy so far. The number of false negatives is reduced to 15.

6. Conclusion

The Random Forest model tuned with GridSearchCV provided the best performance, with an accuracy of 77.92% and a recall of 0.73 for the positive class. This model is chosen as the final model for the diagnosis system.

7. Model Saving

The best performing model and the scaler are saved to disk using `joblib` for later use in the application.
- Model: `diabetes_model.joblib`
- Scaler: `scaler.joblib`