|  |
| --- |
| Project Documentation: Diabetes Prediction Model |
| [Company Name]  [Street Address, City, ST ZIP Code]  **[**Phone]  **[**Email]  [Website] |

Overview

This project builds a predictive model to identify diabetic patients based on clinical and lifestyle data. Using supervised machine learning techniques, the goal is to create a model that maximizes recall and F1-score, allowing for accurate detection of diabetic individuals while balancing false positive rates.

Table of Contents

1. [Data Exploration](#data-exploration)
2. [Data Preprocessing](#data-preprocessing)
3. [Exploratory Data Analysis (EDA)](#exploratory-data-analysis-eda)
4. [Model Selection and Training](#model-selection-and-training)
5. [Model Evaluation and Tuning](#model-evaluation-and-tuning)
6. [Results](#results)
7. [Conclusion and Next Steps](#conclusion-and-next-steps)

1. Data Exploration

* Data Source: A dataset containing information on several health indicators and whether the patient has diabetes (labeled as Outcome).
* Objective: Understand the structure and properties of the data, including potential data imbalance between diabetic and non-diabetic cases.

Key Steps

* Load the Data: Import and display the data to get a first look at feature names and their types.
* Data Description: Use .describe() and .info() functions to summarize the dataset, including counts, data types, and descriptive statistics.
* Target Variable: Examine the Outcome variable to check class distribution, revealing any class imbalance between diabetic (1) and non-diabetic (0) samples.

2. Data Preprocessing

* Handling Missing Values: Replace zero values in key columns (Glucose, Insulin, SkinThickness, BloodPressure, BMI) with the median (for skewed distributions like Glucose and Insulin) or mean (for relatively normal distributions like BMI and BloodPressure).
* Outliers: Use box plots to identify and possibly treat outliers, focusing on avoiding extreme values that might skew model performance.
* Scaling: Standardize or normalize numerical features to improve model convergence, especially for distance-based models like K-Nearest Neighbors (KNN).

**3. Exploratory Data Analysis (EDA)**

* **Visualization**:
  + **Histograms**: Visualize the distribution of each feature to understand ranges and skewness.
  + **Correlation Matrix**: Compute and plot a heatmap to assess feature correlations with each other and with the Outcome variable.
  + **Joint Plot for Glucose vs. Outcome**: Examine the relationship between Glucose levels and the likelihood of diabetes.
  + **Key Findings**:
  + Some features, such as Glucose, are strongly correlated with Outcome.
  + The class imbalance is evident, with fewer cases labeled as diabetic.

**4. Model Selection and Training**

* **K-Nearest Neighbors (KNN)**:
  + Initialize the KNN model and split the dataset into training and testing sets.
  + Use a loop to find the optimal value of k by evaluating accuracy across a range of k values.
  + Visualize training and test accuracy to identify the best k based on performance stability.
* **Support Vector Machine (SVM)**:
  + Train an SVM classifier with class weights to handle the imbalanced data.
  + Use GridSearchCV to tune hyperparameters and identify the best model settings.

**5. Model Evaluation and Tuning**

* **Metrics**: Evaluate model performance using:
  + **Recall**: High recall ensures that most diabetic cases are correctly identified.
  + **Precision**: Controls false positives, reducing misdiagnosis.
  + **F1-Score**: Balances recall and precision, giving a single metric for model effectiveness.
* **Confusion Matrix**: Display the confusion matrix to visualize correct and incorrect classifications.
* **Class Weights**: Adjust class weights to improve recall for the minority diabetic class.

**6. Results**

* **KNN Model**:
  + Achieves an accuracy of X%, with a recall of Y% for the diabetic class. The optimal k value found was Z.
* **SVM Model**:
  + With tuned parameters, the SVM achieved a recall of Y% and an F1-score of Z% on the diabetic class, balancing sensitivity and precision.
* **Comparison**: The SVM model with tuned parameters performed slightly better in terms of recall and F1-score, indicating it may be more effective in identifying diabetic cases without excessive false positives.

**7. Conclusion and Future Steps**

* **Summary**: The project successfully built and evaluated a diabetes prediction model using both KNN and SVM, with SVM showing better performance for the imbalanced dataset.
* **Future Work**:
  + Experiment with additional algorithms such as **Random Forests** or **XGBoost** for potentially improved accuracy.
  + Consider using **oversampling techniques** (e.g., SMOTE) or **under-sampling** to further handle class imbalance.
  + Conduct **feature engineering** or selection to test additional combinations and reduce noise.