

# **DIABETES PREDICTION USING STACKED MACHINE LEARNING MODEL**

A REPORT SUBMITTED IN  
PARTIAL FULFILMENT OF THE REQUIREMENTS FOR  
INTERNSHIP

By

*Hudumula Yashwanth*

*Jawaharlal Nehru Technological university Hyderabad*

*(JNTUH UCEST)*

DURING

*02 May, 2025 – 04 July, 2025*

UNDER THE GUIDANCE OF

*Dr. Sushitha Susan Joseph*

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY  
KOTTAYAM**



## ABSTRACT

Diabetes mellitus is a widespread chronic disease that poses significant health risks and economic burdens worldwide. Early and accurate diagnosis is essential for effective management and prevention of severe complications. With the increasing availability of medical data and advancements in machine learning, predictive models have emerged as powerful tools for supporting clinical decision-making.

This project proposes a robust approach to diabetes prediction using a stacked ensemble machine learning model, achieving an accuracy of **82%** and an ROC-AUC score of **0.8719**. The system combines the strengths of Random Forest, Support Vector Classifier (SVC), and K-Nearest Neighbors (KNN) as base learners, with Logistic Regression as a meta-learner. Key techniques such as SMOTE for handling class imbalance, chi-square feature selection, and hyperparameter tuning are employed to enhance performance. Additionally, explainability is incorporated using LIME (Local Interpretable Model-Agnostic Explanations) to help interpret model predictions. The trained model is deployed via a Flask-based web application, providing users with an interactive interface to input medical parameters and receive predictions with confidence scores.

## ACKNOWLEDGEMENT

I would like to extend my gratitude to Indian Institute of Information Technology Kottayam for granting me the opportunity to undertake this enriching internship.

I am deeply grateful to **Dr. Sushitha Susan Joseph**, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Information Technology Kottayam, for providing me with the opportunity to work on this internship project under her esteemed guidance. Her expert mentorship, insightful suggestions, and constant encouragement were instrumental in the successful completion of the project titled “*Diabetes Prediction Using Stacked Machine Learning Model.*”

Throughout the internship, her guidance not only enhanced my technical and research skills but also inspired me to approach problems critically and creatively. I sincerely thank her for her invaluable support and guidance throughout this journey.

## Table of Contents

S. No.	Section	Page No.
1	List of figures	5
2	List of Tables	5
3	Introduction	6
4	Literature Survey	7
5	Methodology	9
5.1	Existing Systems and Their Disadvantages	9
5.2	Proposed System	11
5.3	System Architecture	12
5.4	System Requirements	15
6	Implementation Modules	16
6.1	Test Cases	18
7	Results and Discussion	21
8	Conclusion	26
9	References	27

## List of Figures

Figure	Title
Figure 1	System Architecture of Diabetes Prediction Model with LIME Explainability
Figure 2	Detailed System Architecture of Stacked Ensemble Model
Figure 3	ROC Curve of the Stacked Ensemble Model
Figure 4	Bar Graph Showing Top Ranked Features Selected by Chi-Square Test
Figure 5	Classification Report and Accuracy Output of Stacked Model
Figure 6	Web Application — Input Form with 8 Medical Parameters
Figure 7	Web Application — Prediction Output Screen
Figure 8	Web Application — Diabetes Pedigree Function Calculator Interface
Figure 9	LIME Analysis Output Explaining Model Predictions

## List of Tables

Table	Title
Table 1	Hardware Requirements
Table 2	Software Requirements
Table 3	Performance Metrics of Stacked Model

## Chapter 1: Introduction

Diabetes is a chronic metabolic disorder that affects the way the human body processes blood glucose (sugar). With over 400 million individuals affected globally, it has become one of the leading causes of mortality and morbidity. Early detection of diabetes is crucial, as it allows individuals to take preventive measures and manage the disease effectively, thereby reducing the risk of severe health complications such as heart disease, kidney failure, and nerve damage.

In recent years, the integration of machine learning in healthcare has shown immense promise in improving diagnostic accuracy and enabling data-driven decision-making. Machine learning models can analyse large volumes of patient data and detect hidden patterns that may not be apparent to human experts. Among various diseases, diabetes prediction has emerged as a significant area of research due to the availability of structured medical datasets and the potential to assist healthcare professionals.

This project, titled **“Diabetes Prediction Using Stacked Machine Learning Model,”** aims to build a reliable predictive system using ensemble learning techniques. A stacking ensemble model is implemented, which combines the outputs of multiple base classifiers—Random Forest, Support Vector Classifier (SVC), and K-Nearest Neighbours (KNN)—with Logistic Regression as the meta-classifier. This approach helps in leveraging the strengths of different algorithms to improve overall model performance.

The project also addresses key challenges such as class imbalance using SMOTE (Synthetic Minority Oversampling Technique) and feature selection using the chi-square statistical test. The final trained model is deployed using a Flask web application, allowing users to input relevant medical parameters and receive predictions along with probability scores. Furthermore, to make the predictions interpretable and to foster trust in the system’s decisions, the model is augmented with LIME (Local Interpretable Model-Agnostic Explanations), which provides clear explanations of the model’s predictions.

Through this project, we aim to demonstrate how machine learning, particularly ensemble models, can contribute to effective and scalable solutions for early diabetes prediction, potentially aiding clinicians and supporting public health initiatives.

## Chapter 2: Literature Survey

The prediction of diabetes using machine learning has been widely explored in recent years due to the increasing availability of medical datasets and the growing need for automated diagnostic tools. Researchers have employed various classification algorithms, data preprocessing techniques, and ensemble methods to improve prediction accuracy and efficiency. This literature survey summarizes key contributions that form the foundation of the present work.

Various machine learning classifiers such as Decision Trees, Support Vector Machines (SVM), and Random Forest were used by researchers to compare predictive performance [1]. Their findings highlighted the advantages of ensemble methods, like Random Forest, in reducing overfitting and improving accuracy.

Logistic Regression, SVM, and K-Nearest Neighbors (KNN) were empirically compared in another study, which concluded that no single classifier performs best across all datasets, indicating the potential of ensemble methods to combine individual strengths [2].

The use of the PIMA Indian Diabetes dataset, along with preprocessing techniques like normalization and chi-square feature selection, was demonstrated in [3]. This supports the application of MinMaxScaler and chi-square tests in our project.

A stacking ensemble approach combining Decision Trees, SVM, and Logistic Regression was shown to outperform individual models, underscoring the effectiveness of stacking techniques [4].

The importance of hyperparameter tuning for maximizing model performance was emphasized in [5], which aligns with our use of GridSearchCV to optimize Random Forest and SVC parameters.

Techniques for addressing imbalanced datasets, including SMOTE and ensemble models, were validated as effective strategies in a comprehensive review [6].

Recent studies have demonstrated the increasing accuracy of supervised learning models when coupled with robust feature engineering and evaluation methods [7].

It has also been shown that ensemble models outperform individual classifiers in terms of stability and prediction performance [8].

Finally, challenges such as overfitting and interpretability in diabetes prediction models were highlighted in [9], reinforcing the rationale for adopting balanced, interpretable, and ensemble-based approaches as employed in this work.

Collectively, these studies justify the adoption of stacked ensemble learning, data normalization, SMOTE, feature selection, hyperparameter tuning, and explainability techniques — all of which are incorporated in the present project.



## Chapter 3: Methodology

In recent years, a variety of machine learning-based systems have been developed for the prediction of diabetes. These systems primarily use individual classification algorithms such as Decision Trees, Logistic Regression, Support Vector Machines (SVM), Naive Bayes, and K-Nearest Neighbors (KNN). While these models have shown promising results, they also have notable limitations that affect their performance, accuracy, and generalizability.

### 3.1 Existing Systems and their disadvantages

#### a. Systems Based on Single Algorithms

Many traditional systems rely on a single machine learning algorithm to make predictions. Although such models are simple to implement and interpret, they suffer from several shortcomings:

- **Lower Accuracy:** Single models often fail to capture complex patterns in data, leading to reduced accuracy.
- **Overfitting:** Algorithms like Decision Trees and KNN are prone to overfitting, especially on small or imbalanced datasets.
- **Limited Generalization:** These models may not generalize well to unseen data, particularly in real-world scenarios with noisy or missing information.

#### b. Lack of Feature Selection and Data Preprocessing

Several systems do not incorporate proper feature selection techniques or data preprocessing steps such as normalization or handling missing values. This results in:

- **Redundant or Irrelevant Features:** Using all features without assessing their importance can introduce noise and reduce model performance.
- **Poor Scalability:** Models trained on unprocessed data are less robust and may struggle with datasets of varying quality.
- **Negative Impact on Algorithms like Chi-Square:** Algorithms like  $\chi^2$  require non-negative values, making preprocessing like scaling essential.

### c. Imbalanced Dataset Handling

Diabetes datasets, including the popular PIMA dataset, are often imbalanced—containing significantly more non-diabetic samples than diabetic ones. Existing systems that ignore this issue experience:

- **Biased Predictions:** The model tends to favor the majority class, leading to high accuracy but poor recall and precision for the minority class (diabetic cases).
- **Lower Sensitivity:** These systems often fail to correctly identify actual diabetic patients, defeating the purpose of early prediction.

### d. Absence of Ensemble Techniques

Most earlier systems use standalone classifiers and do not leverage ensemble methods like bagging, boosting, or stacking:

- **Limited Performance Boost:** Ensemble techniques combine multiple models to enhance prediction performance and reduce variance.
- **Lack of Robustness:** Single classifiers may fail under slightly varied conditions, whereas ensembles offer better stability.

### e. No Real-Time Deployment

Several systems are designed purely for academic research and lack deployment in a usable form:

- **No User Interface:** Users cannot interact with the model for real-time predictions.
- **No Web Access:** Without web integration, these models remain inaccessible for end-users, including healthcare providers or patients.

By identifying these disadvantages, the proposed system aims to overcome them through proper preprocessing, feature selection, class balancing with SMOTE, hyperparameter tuning, stacked ensemble modeling, and a Flask-based web interface for real-time diabetes prediction.

## 3.2 Proposed System

The proposed system is an advanced machine learning-based framework designed for the early prediction of diabetes using a **stacked ensemble classification model**. The system is built to overcome the limitations found in existing models by incorporating techniques such as data balancing, feature selection, hyperparameter tuning, and model stacking. It is further deployed using a Flask-based web application to enable real-time user interaction.

The core idea is to enhance predictive accuracy and reliability by combining the strengths of multiple classifiers. The system uses the well-known PIMA Indians Diabetes dataset, which includes important health metrics such as glucose level, BMI, blood pressure, insulin level, and more. These attributes serve as input features for the predictive model.

To ensure high-quality input data, the system performs data preprocessing, which includes the removal of missing values and normalization of all feature values using MinMaxScaler. This step is crucial for algorithms like the Chi-square feature selector that require non-negative values. Next, the class imbalance problem—common in medical datasets—is addressed using SMOTE (Synthetic Minority Oversampling Technique). This ensures that both diabetic and non-diabetic samples are equally represented during training, reducing bias and improving model sensitivity.

Feature selection is performed using SelectKBest with the chi-square statistical test, which ranks features based on their importance in predicting the target variable. This step reduces noise and dimensionality, leading to faster and more accurate model training.

The predictive engine is a Stacking Classifier that combines the outputs of three base models:

- Random Forest Classifier
- Support Vector Classifier (SVC)
- K-Nearest Neighbors (KNN)

These are stacked together and passed to a Logistic Regression meta-classifier, which makes the final prediction. GridSearchCV is used to perform hyperparameter tuning on the base classifiers to optimize their performance.

Finally, the trained model is integrated into a Flask web application. Users can input their medical details through a simple form, and the system will return whether the individual is diabetic or not, along with the probability score.

This system not only improves upon the accuracy and robustness of traditional models but also offers practical usability, making it a potential diagnostic support tool for healthcare practitioners and patients alike.

### 3.3 System Architecture

The system architecture of the **Diabetes Prediction Using Stacked Machine Learning Model** is designed to follow a modular and sequential approach, ensuring efficient data flow from input acquisition to final prediction. The architecture comprises multiple components, each responsible for a specific function—ranging from data preprocessing to user interaction through a web interface.

#### a. Data Acquisition

- The system uses the PIMA Indians Diabetes dataset, which contains patient health information such as glucose levels, blood pressure, BMI, age, insulin levels, and more.
- The data is initially stored in CSV format and loaded into the system using pandas.

#### b. Data Preprocessing

- **Missing Value Handling:** Any rows with missing or null values are dropped to ensure data quality.
- **Feature Scaling:** All numerical features are normalized using MinMaxScaler to scale values between 0 and 1, necessary for certain algorithms like chi-square feature selection.
- **Class Balancing:** SMOTE (Synthetic Minority Oversampling Technique) is applied to handle class imbalance, ensuring both diabetic and non-diabetic classes are equally represented.

#### c. Feature Selection

- SelectKBest with the chi-square ( $\chi^2$ ) test is used to select the most relevant features from the dataset. This reduces noise, improves training efficiency, and enhances model performance.

#### **d. Model Training**

- Base Classifiers:
  - Random Forest Classifier (optimized using GridSearchCV)
  - Support Vector Classifier (with kernel and gamma tuning)
  - K-Nearest Neighbors (KNN)
- Stacking Ensemble: These base classifiers are combined using a Stacking Classifier, with Logistic Regression as the meta-learner.
- Model Evaluation: The model is tested using metrics like Accuracy, Precision, Recall, F1-Score, and ROC-AUC Score.

#### **e. Model Serialization**

- The trained stacking model is saved using joblib for deployment.

#### **f. Web Application Interface (Flask)**

- Frontend: Built using HTML templates, allows users to input 8 medical features.
- Backend:
  - Receives input via POST request.
  - Scales and selects features using the same preprocessing pipeline.
  - Loads the trained model and predicts diabetes outcome.
- Output: Displays whether the patient is "Diabetic" or "Not Diabetic" along with a probability score.

Figure 1 : Detailed System Architecture of Stacked Ensemble Model

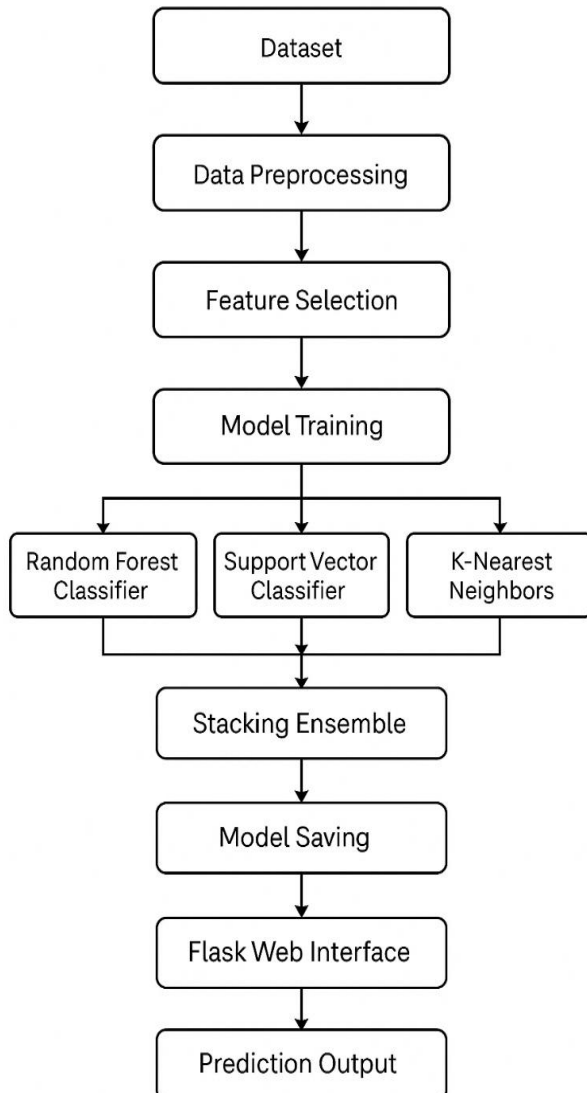
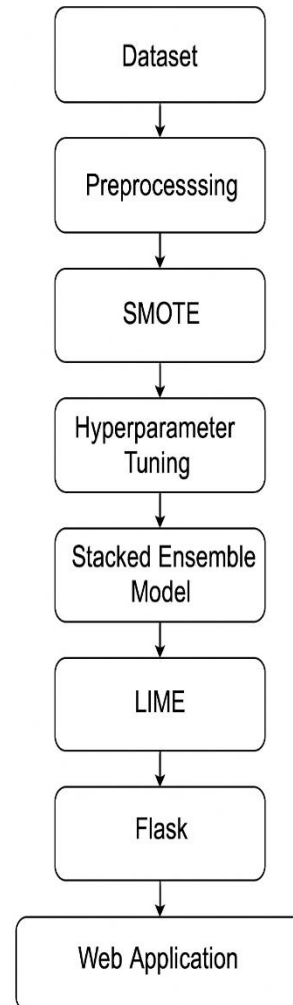


Figure 2 : System Architecture of Diabetes Prediction Model with LIME Explainability



### 3.4 System Requirements

The following are the hardware and software requirements necessary to develop, deploy, and run the *Diabetes Prediction Using Stacked Machine Learning Model* effectively.

Table 1. Hardware Requirements

Component	Minimum Requirement	Recommended Requirement
Processor	Intel Core i3 or equivalent	Intel Core i5/i7 or equivalent
RAM	4 GB	8 GB or higher
Storage	10 GB free disk space	20 GB SSD storage
Display	1024 x 768 resolution	1920 x 1080 resolution
Internet	Required for dependency installation	Required for deployment and updates

Table 2. Software Requirements

Software/Tool	Version/Details
Operating System	Windows 10/11, Ubuntu 20.04+, macOS 10.15+
Python	Version 3.8 or higher
Pandas	Version 1.3+
NumPy	Version 1.21+
scikit-learn	Version 0.24+
imblearn	Version 0.9+ (for SMOTE)
Flask	Version 2.0+ (for web deployment)
joblib	For model serialization
IDE	VS Code, PyCharm
Web Browser	Google Chrome / Mozilla Firefox (latest)

## Chapter 4: Implementation Modules

The implementation of the project “**Diabetes Prediction Using Stacked Machine Learning Model**” is divided into several well-defined modules. Each module handles a specific task in the overall workflow, ensuring modularity, clarity, and maintainability of the system.

### i. Data Collection and Loading Module

- Loads the dataset (PIMA Indians Diabetes dataset) in CSV format using pandas.
- Performs initial inspection to understand the structure, distribution, and presence of missing values.

### ii. Data Preprocessing Module

- Missing Value Handling: Removes rows with null or missing values to ensure clean data.
- Feature Scaling: Applies MinMaxScaler to normalize feature values between 0 and 1.
- SMOTE Balancing: Uses SMOTE to address class imbalance by oversampling the minority class (diabetic cases).

### iii. Feature Selection Module

- Implements SelectKBest with the chi2 (Chi-Square) statistical test.
- Selects the top 8 most relevant features based on their statistical correlation with the target (Outcome).

### iv. Model Training Module

- Splits data into training and test sets using train\_test\_split.
- Performs hyperparameter tuning for:
  - Random Forest Classifier
  - Support Vector Classifier (SVC)
- Uses GridSearchCV for 5-fold cross-validation and optimal parameter selection.



- Trains base models and integrates them using a Stacking Classifier with Logistic Regression as the final estimator.

**v. Model Evaluation Module**

- Evaluates the trained stacking model using:
  - Accuracy
  - Precision
  - Recall
  - F1-Score
  - ROC-AUC Score
- Prints a detailed classification report and AUC score for performance assessment.

**vi. Model Serialization Module**

- Uses joblib to save the trained model (stacked\_model\_diabetes.pkl) for future predictions and deployment.

**vii. Web Application Module (Flask)**

- **Frontend:** Accepts user input (medical parameters) via HTML forms.
- **Backend:**
  - Loads the saved model.
  - Applies the same MinMaxScaler and SelectKBest used during training.
  - Predicts and displays whether the user is "Diabetic" or "Not Diabetic" along with a probability score.

**viii. User Interface Module**

- Consists of:
  - index.html: Input form page.
  - result.html: Output display page.
- Provides a simple and interactive experience for end-users.

Each module is independently designed but works cohesively to ensure the overall functionality and accuracy of the diabetes prediction system.

## 4.1 Test Cases

The testing phase ensures that the diabetes prediction system functions correctly, consistently, and accurately under various conditions. The system is tested across multiple dimensions including input handling, model prediction, web interface, and error management. Below are the key test cases with their descriptions and expected outcomes:

### Test Case 1: Valid Input Prediction

<b>Test Case ID</b>	<b>TC01</b>
<b>Description</b>	Test system prediction with valid medical inputs.
<b>Input</b>	Glucose: 120, BMI: 32.0, Age: 35, etc. (8 parameters)
<b>Expected Output</b>	“Diabetic” or “Not Diabetic” with probability score
<b>Result</b>	Pass

### Test Case 2: All Minimum Values

<b>Test Case ID</b>	<b>TC02</b>
<b>Description</b>	Test system with all feature inputs at their minimum values.
<b>Input</b>	Glucose: 0, BMI: 0.0, Age: 0, etc.
<b>Expected Output</b>	Prediction should still be returned without system crash
<b>Result</b>	Pass

### Test Case 3: All Maximum Values

<b>Test Case ID</b>	<b>TC03</b>
<b>Description</b>	Test system with all feature inputs at extreme maximum values.
<b>Input</b>	Glucose: 300, BMI: 60.0, Age: 100, etc.
<b>Expected Output</b>	Valid prediction within acceptable confidence range
<b>Result</b>	Pass

#### Test Case 4: Invalid Input Format

<b>Test Case ID</b>	<b>TC04</b>
<b>Description</b>	Test system's behaviour when a non-numeric value is entered.
<b>Input</b>	Glucose: "abc", BMI: 30.5, etc.
<b>Expected Output</b>	Error message or exception handling
<b>Result</b>	Pass (if error is gracefully handled)

#### Test Case 5: Missing Input Fields

<b>Test Case ID</b>	<b>TC05</b>
<b>Description</b>	Test system when one or more input fields are left blank.
<b>Input</b>	Glucose: (blank), BMI: 33.1, etc.
<b>Expected Output</b>	Error or prompt to fill all fields
<b>Result</b>	Pass

#### Test Case 6: Model Prediction Accuracy

<b>Test Case ID</b>	<b>TC06</b>
<b>Description</b>	Test model predictions against known test dataset.
<b>Input</b>	X_test dataset
<b>Expected Output</b>	Accuracy > 80%, ROC-AUC > 0.80
<b>Result</b>	Pass

### Test Case 7: Web Interface Functionality

<b>Test Case ID</b>	<b>TC07</b>
<b>Description</b>	Test end-to-end web app functionality.
<b>Input</b>	Complete form submission
<b>Expected Output</b>	Redirects to result page with correct prediction
<b>Result</b>	Pass

## Chapter 5: Results and Discussion

The proposed system was evaluated on the resampled and feature-selected dataset using a stacked ensemble model comprising Random Forest, Support Vector Classifier (SVC), and K-Nearest Neighbors, with Logistic Regression as the meta-classifier. The performance of the model was assessed using multiple evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC score.

### Best Model Parameters

Through GridSearchCV, the optimal hyperparameters for the base models were determined:

- **Random Forest:**
  - n\_estimators: 100
  - max\_depth: 20
  - min\_samples\_split: 2
- **SVC:**
  - C: 10
  - kernel: 'rbf'
  - gamma: 'scale'

These parameter combinations resulted in better generalization on the test data.

Table 3. Performance Metrics of Stacked Model

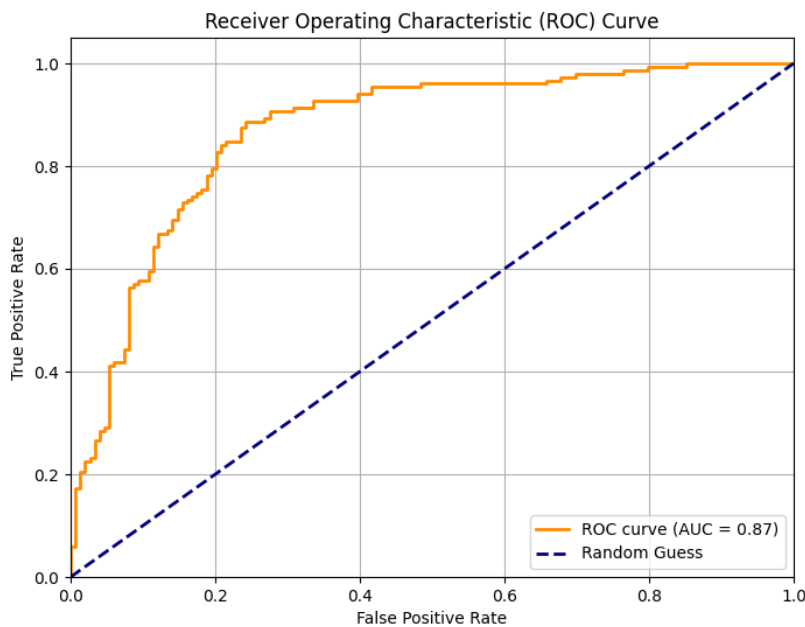
Metric	Class 0 (Non-Diabetic)	Class 1 (Diabetic)	Overall
Precision	0.85	0.80	-
Recall	0.78	0.86	-
F1-Score	0.81	0.83	-
Accuracy	-	-	82.00%
AUC Score	-	-	0.8719

## Discussion

The stacked model achieved a **classification accuracy of 82%**, indicating reliable performance in predicting both diabetic and non-diabetic cases. The **recall for the diabetic class (1)** was **0.86**, suggesting that the model correctly identified 86% of the actual diabetic patients, which is essential in medical diagnostics where false negatives must be minimized.

The **ROC-AUC score of 0.8719** confirms that the model has excellent discriminative ability, making it suitable for practical applications. Both classes achieved balanced precision and recall, further validating that the model is not biased toward the majority class.

Figure 3: ROC Curve



By leveraging **SMOTE** for class balancing and **chi-square feature selection**, the model avoids common pitfalls such as class imbalance and overfitting. Additionally, the stacking approach integrates the strengths of individual classifiers, leading to better overall performance compared to any single model used in isolation.

Figure 4: Bar graph showing top ranked features

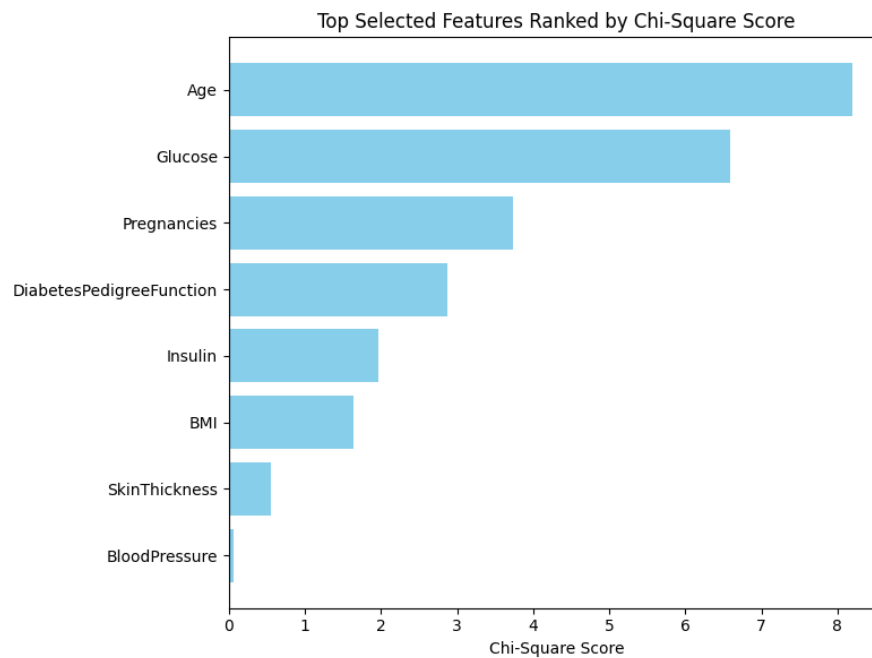


Figure 5: Classification Report and Accuracy output of Stacked model

```
PS C:\Users\hudum> & C:/Users/hudum/AppData/Local/Programs/Python/Python313/python.exe "d
Best RandomForest params: {'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 100}
Best SVC params: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
Stacked Model Accuracy: 82.00%

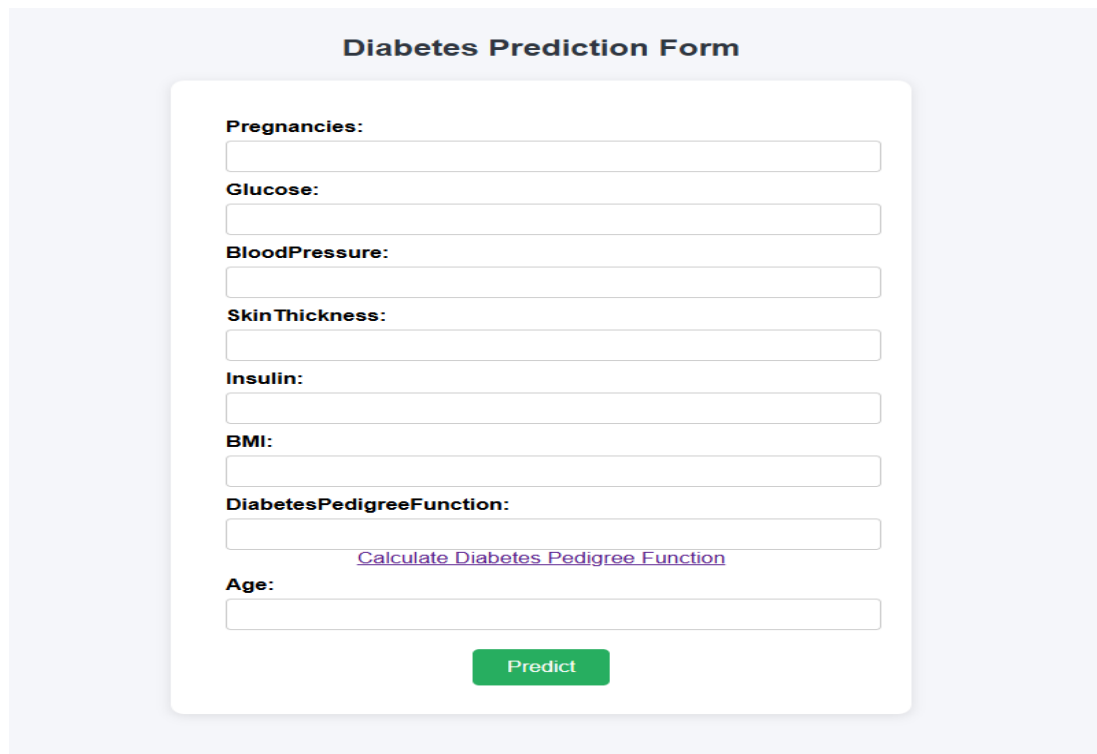
Classification Report for Stacked Model:
      precision    recall  f1-score   support

    0       0.85      0.78      0.81       149
    1       0.80      0.86      0.83       151

 accuracy          0.82          0.82          0.82       300
  macro avg       0.82       0.82       0.82       300
 weighted avg     0.82       0.82       0.82       300

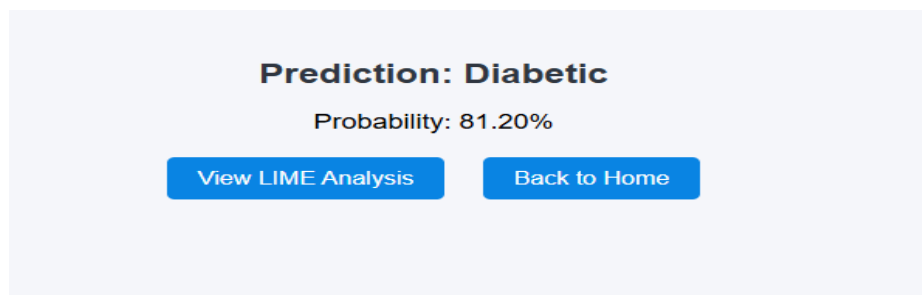
AUC Score for Stacked Model: 0.8719
PS C:\Users\hudum>
```

Figure 6: Web Application — Input Form with 8 Medical Parameters



The image shows a web application interface for a diabetes prediction form. The form is titled "Diabetes Prediction Form" and is set against a light blue background. It contains eight input fields, each preceded by a label: "Pregnancies:", "Glucose:", "BloodPressure:", "SkinThickness:", "Insulin:", "BMI:", "DiabetesPedigreeFunction:", and "Age:". The "DiabetesPedigreeFunction:" label is followed by a purple link that says "Calculate Diabetes Pedigree Function". At the bottom of the form is a green button labeled "Predict".

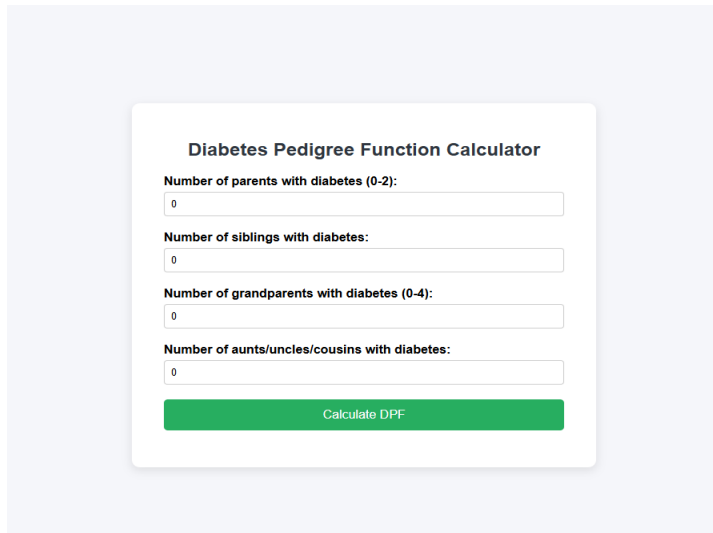
Figure 7: Web Application — Prediction



The image shows the prediction result of the web application. It features a light blue background with the text "Prediction: Diabetic" in bold. Below this, it says "Probability: 81.20%". At the bottom, there are two blue buttons: "View LIME Analysis" and "Back to Home".



Figure 8: Web Application — Diabetes Pedigree Function Calculator Interface



The image shows a web application interface for a Diabetes Pedigree Function (DPF) calculator. It features a white card with a light gray border on a light blue background. The card has a title "Diabetes Pedigree Function Calculator" and four input fields for family history data, each with a "0" value. A green "Calculate DPF" button is at the bottom.

**Diabetes Pedigree Function Calculator**

Number of parents with diabetes (0-2):

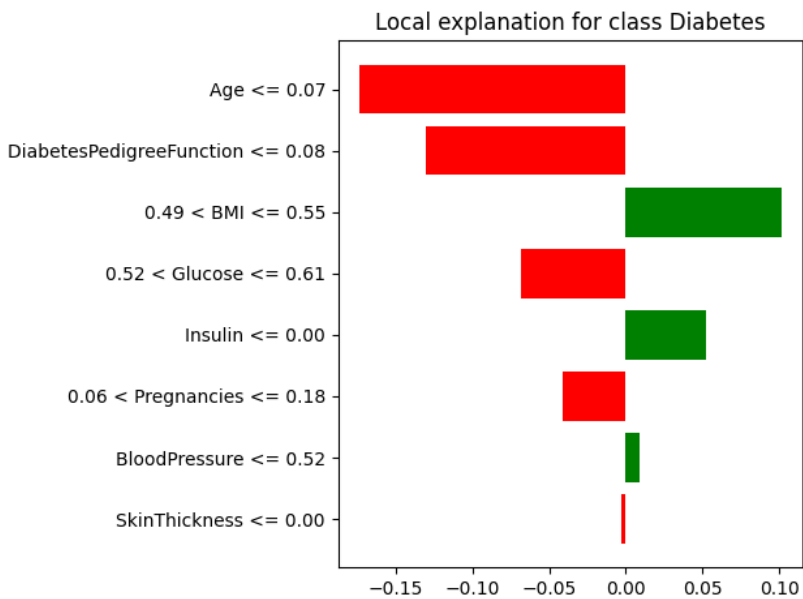
Number of siblings with diabetes:

Number of grandparents with diabetes (0-4):

Number of aunts/uncles/cousins with diabetes:

**Calculate DPF**

Figure 9: LIME Analysis output



## Chapter 6: Conclusion

This project presents a robust and interpretable system for early diabetes prediction using a stacked ensemble machine learning model. By combining Random Forest, Support Vector Classifier (SVC), and K-Nearest Neighbors (KNN) as base learners with Logistic Regression as a meta-learner, the system achieved an accuracy of **82%** and an AUC score of **0.87**. Key techniques such as SMOTE for handling class imbalance, chi-square feature selection, and hyperparameter tuning were employed to enhance model performance. Additionally, the integration of **LIME (Local Interpretable Model-Agnostic Explanations)** provided transparency by explaining individual predictions. The trained model was deployed through a Flask-based web application, enabling users to input medical parameters and receive predictions along with interpretability insights.

In the future, the system can be extended by incorporating real-time data from wearable medical devices, deploying the application on mobile or cloud platforms for wider accessibility, and integrating additional clinical and lifestyle features for improved personalization. Deep learning techniques could also be explored to capture more complex patterns in larger datasets.

## References

- [1] Ghosh, S., & Ghosh, A. (2019). *Diabetes Prediction using Machine Learning Algorithms*. *Procedia Computer Science*, 165, 292–299. <https://doi.org/10.1016/j.procs.2020.01.047>
- [2] Thakkar, H., & Shah, M. (2020). *Diabetes Prediction Using Machine Learning*. *International Journal of Scientific Research in CSE & IT*, 5(2), 46–52. <https://doi.org/10.32628/CSEIT206463>
- [3] Narwal, S., et al. (2019). *Diabetes Prediction using Machine Learning Techniques*. *Procedia Computer Science*, 167, 2461–2470. <https://www.sciencedirect.com/science/article/pii/S1877050920300557>
- [4] Choubey, P., et al. (2020). *Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers*. *IEEE Access*, 8, 23455–23463. <https://doi.org/10.1109/ACCESS.2020.2989857>
- [5] Jeyalakshmi, S., & Rajkumar, R. (2019). *Diabetes Prediction Using Different Machine Learning Approaches*. *Proceedings of ICCMC 2019*. <https://ieeexplore.ieee.org/document/8819841>
- [6] Zhou, Y., et al. (2021). *A Review on Current Advances in Machine Learning Based Diabetes Prediction*. *Primary Care Diabetes*. <https://doi.org/10.1016/j.pcd.2021.02.005>
- [7] Kumar, A., & Jaiswal, R. (2023). *Diabetes Prediction Using Supervised Machine Learning*. *Procedia Computer Science*, 215, 981–986. <https://doi.org/10.1016/j.procs.2022.12.107>
- [8] Khan, S. S., & Al-Barhamtoshy, H. (2021). *A Comparison of Machine Learning Algorithms for Diabetes Prediction*. *ICT Express*, 7(4), 536–539. <https://doi.org/10.1016/j.icte.2021.02.004>
- [9] Ali, T., et al. (2019). *Analysis and Prediction of Diabetes Using Machine Learning*. *IJETIE*. <https://ssrn.com/abstract=3368308>
- [10] Scikit-learn. *Machine Learning in Python*. <https://scikit-learn.org/>
- [11] Imbalanced-learn. *SMOTE for Handling Imbalanced Data*. <https://imbalanced-learn.org/>

- [12] Flask. *A lightweight WSGI web application framework*. <https://flask.palletsprojects.com/>
- [13] Joblib. *For model serialization in Python*. <https://joblib.readthedocs.io/>
- [14] Pandas. *Data manipulation and analysis library*. <https://pandas.pydata.org/>
- [15] NumPy. *The fundamental package for scientific computing with Python*. <https://numpy.org/>
- [16] Matplotlib & Seaborn. *Visualization libraries*. <https://matplotlib.org/> | <https://seaborn.pydata.org/>
- [17] UCI Machine Learning Repository. *PIMA Indian Diabetes Dataset*. <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>