**CS634 Final Term Project Report**

**Student Name:** Mahesh Chandra Chowdary Bodepudi (mb2327)  
**Course:** CS634 – Data Mining  
**Instructor:** Dr. Yasser Abduallah  
**Project Title:** Binary Classification of Diabetes Health Indicators Using Random Forest, LSTM, and KNN

**1. Introduction**

The purpose of this project is to design, implement, and evaluate multiple binary classification algorithms to predict diabetes using a real-world health indicators dataset. Binary classification aims to categorize each instance into one of two possible outcomes. in this case, “Diabetic” or “Non-Diabetic.” This task is significant in healthcare analytics because early detection enables clinical intervention and reduces long-term medical complications.

In compliance with project requirements, this study uses three supervised classification models:

1. **Random Forest** – required ensemble baseline
2. **LSTM (Long Short-Term Memory)** – selected deep learning model
3. **K-Nearest Neighbors (KNN)** – selected traditional machine learning algorithm

The dataset used is the **Diabetes Health Indicators Dataset** from Kaggle. All models are evaluated using **10-fold cross-validation**, and all performance metrics—including TP, TN, FP, FN, accuracy, precision, recall, F1-score, and advanced metrics such as TSS, HSS, BS, and BSS—are computed manually as required.

**2. Dataset**

**Name & Source**

* **Dataset:** Diabetes Health Indicators Dataset
* **Source:** Kaggle
* **Link:** <https://www.kaggle.com/datasets/mohankrishnathalla/diabetes-health-indicators-dataset>

**2.2 Dataset Description**

The dataset contains **10,000 rows** and **30 features**, covering demographic, lifestyle, clinical, and biochemical indicators relevant to diabetes prediction.  
The columns include:

* **Demographic features:**  
  age, gender, ethnicity, education\_level, income\_level, employment\_status
* **Lifestyle habits:**  
  smoking\_status, alcohol\_consumption\_per\_week,  
  physical\_activity\_minutes\_per\_week, diet\_score,  
  sleep\_hours\_per\_day, screen\_time\_hours\_per\_day
* **Medical history:**  
  family\_history\_diabetes, hypertension\_history, cardiovascular\_history
* **Clinical measurements:**  
  bmi, waist\_to\_hip\_ratio, systolic\_bp, diastolic\_bp, heart\_rate
* **Lab results:**  
  cholesterol\_total, hdl\_cholesterol, ldl\_cholesterol, triglycerides,  
  glucose\_fasting, glucose\_postprandial, insulin\_level, hba1c
* **Derived indicators:**  
  diabetes\_risk\_score
* **Target Variable:** Diabetes\_state (0 = No, 1 = Yes)

**2.3 Preprocessing Steps**

1. **Loading the dataset**  
   The CSV file diabetes\_dataset.csv is read into a pandas DataFrame.

A screenshot of a computer

AI-generated content may be incorrect.

**2. Handling Missing or Invalid Values**

* Features with unrealistic values (e.g., 0 for glucose or BMI) were replaced with NaN.
* Missing values were imputed using **median imputation**, consistent with your helper function.

A screenshot of a phone

AI-generated content may be incorrect.

**3. Categorical Encoding**

* All categorical variables such as gender, ethnicity, education\_level, smoking\_status, etc., were converted into numeric form using **LabelEncoder**.

A screenshot of a computer code

AI-generated content may be incorrect.

1. **Feature Standardization**

* Because KNN and LSTM are sensitive to feature scale, all numerical predictors were standardized using z-score normalization:

Xscaled​ = x−μ​ / σ

1. **Train-Test Split**

* A 10% held-out test set was created using **stratified sampling**.
* This preserves the diabetes class distribution.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Stratified 10-Fold Cross-Validation**

* Used for training and evaluating all algorithms.
* Ensures balanced representation of diabetic vs. non-diabetic cases in each fold.

A screenshot of a computer code

AI-generated content may be incorrect.

**2.5 Class Distribution**

After converting diabetes\_stage to binary:

* The dataset exhibits **class imbalance**, with “Non-Diabetic” being more frequent.
* You handled this correctly using **StratifiedKFold** to preserve ratios in all training/testing splits.

**3. Visualizations**

**3.1 Histogram Plots of the Dataset**

**A group of blue and white graphs

AI-generated content may be incorrect.**

**3.2 Scatter Plots of Dataset**

A screenshot of a graph

AI-generated content may be incorrect.

**3.3 Heatmap of the Dataset**

A screenshot of a graph

AI-generated content may be incorrect.

**3.4 Violin Plot**

A graph with different colored lines

AI-generated content may be incorrect.

**3.5 Join Plot of the Dataset**

A graph of alcohol consumption

AI-generated content may be incorrect.

**4. Algorithms Overview**

**4.1 Random Forest**

Random Forest aggregates predictions from multiple decision trees built on bootstrapped samples. It reduces variance and handles complex feature interactions well ideal for medical datasets.

**A screenshot of a computer code

AI-generated content may be incorrect.**

**4.2 LSTM**

Long Short-Term Memory (LSTM) networks model long-range dependencies. In this project, features are treated as ordered sequences, allowing the LSTM to learn deeper patterns.

A screenshot of a computer program

AI-generated content may be incorrect.

**4.3 K-Nearest Neighbors (KNN)**

KNN classifies samples based on the majority label of the nearest neighbors in feature space. It is simple but sensitive to feature scaling and noise.

A screenshot of a computer code

AI-generated content may be incorrect.

**5. Implementation**

**5.1 Development Environment**

* Python
* Jupyter Notebook (.ipynb)
* Standalone .py scripts

5.2 Required Packages

* Numpy
* Pandas
* Scikit – Learn
* Keras
* Matplotlib
* Seaborn

**6. Evaluation Metrics**

**6.1 Manual Evaluation Metric Calculation of Confusion Matrix**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**6.2 Manual Metric Evaluation**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**7. Results**

**7.1 Random Forest Results**

**A screen shot of a graph

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**7.2 LSTM Results**

**A table of numbers with numbers

AI-generated content may be incorrect.**

**A screen shot of a graph

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a spreadsheet

AI-generated content may be incorrect.**

**7.3 KNN Results**

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a data

AI-generated content may be incorrect.

**8. Discussion**

**8.1 Best Performing Algorithm**

Random Forest produced the best overall results across accuracy, balanced accuracy, skill scores, and AUC.

**8.2 Challenges**

* Handling multiple categorical fields
* Preparing the dataset for LSTM (3D reshape)
* Ensuring all metrics were manually calculated as required

**8.3 Future Improvements**

* Try Bi-LSTM or Conv1D architectures
* Apply SMOTE for imbalance
* Hyperparameter tuning for all algorithms
* Feature engineering and selection

**9. Conclusion**

* The Diabetes Health Indicators Dataset was used to test and compare three binary classification algorithms: Random Forest, LSTM, and KNN. Each model was assessed using stratified 10-fold cross-validation after the data was preprocessed using encoding, imputation, normalization, and binary target transformation. All performance indicators were manually calculated as needed.
* Random Forest consistently produced the best results across all folds and aggregated results, demonstrating high accuracy, balanced accuracy, and great skill scores (TSS and HSS). It was a good fit for this tabular medical dataset because of its capacity to manage nonlinear interactions and heterogeneous feature types.
* Even after normalization, KNN yielded mediocre results, highlighting the shortcomings of distance-based algorithms when classes overlap and feature scales differ. In this situation, LSTM did not perform better than Random Forest, despite its ability to learn complex patterns. Deep learning's full potential is limited by the dataset's modest size and lack of sequential organization.
* The findings show that ensemble tree models continue to be a dependable and effective option for structured health datasets. The experiment also demonstrated the importance of using a variety of indicators to assess models instead of just accuracy. Future enhancements might involve experimenting with different deep learning architectures, treating class imbalance more explicitly, feature engineering, and hyperparameter tuning.  
  This work highlights how model selection, data preprocessing, and evaluation techniques directly affect predictive accuracy and illustrates the practical difficulties and implications of applying machine learning to medical prediction problems.

**10. GitHub**

* Link to the GitHub Repository is attached below.

<https://github.com/maheshchandra2002/Bodepudi_Mahesh_FinalProject>