Data Analysis on Diabetes Dataset Using Machine Learning Algorithms

1. **Abstract**
2. **Objectives of Study**
3. **Existing Systems**
4. **Issues in Existing Systems**
5. **Proposed System**
6. **Advantages**
7. **Methodology**
8. **Snippets of Implementing Code**
9. **Requirements**
10. **Literature Review**
11. **References**

ABSTRACT

1. **Project Objective:** To leverage machine learning for predicting diabetes in individuals.
2. **Dataset Overview:** The dataset consists of 100,000 entries initially, with each entry representing individual health records.
3. **Features Included:**
   1. Demographics: Age, Gender.
   2. Health Indicators: BMI, Blood Glucose Level.
   3. Medical History: Hypertension, Heart Disease.
   4. Lifestyle Information: Smoking History.
   5. Clinical Measures: HbA1c Level.
4. **Data Cleaning and Preparation:**
   1. Removed 3,854 duplicate entries.
   2. Addressed missing or incomplete data.
5. **Class Imbalance Solution:** Employed SMOTE to balance the dataset, enhancing the model's ability to predict minority class instances.
6. **Model Exploration:**
   1. Tested a range of models including Logistic Regression, Decision Trees, Random Forest, KNN, XGBoost, CatBoost, and LightGBM.
   2. Evaluated models based on accuracy, precision, recall, and AUC.
7. **Optimal Model Identification:**
   1. CatBoost was identified as the best performing model, particularly effective in handling categorical data and complex relationships within the dataset.
8. **Project Significance:**
   1. Highlights the effectiveness of machine learning in healthcare, especially for predictive diagnostics.
   2. Emphasizes the importance of accurate and comprehensive data for building reliable predictive models.

- This abstract gives a detailed summary of your project, emphasizing the dataset's composition and the methodological rigor in model selection and evaluation.

OBJECTIVE OF THE STUDY

- To develop a predictive model using machine learning techniques for identifying individuals at risk of diabetes.
- Leverage medical and demographic data, such as age, gender, BMI, hypertension, heart disease, smoking history, HbA1c, and blood glucose levels.
- Aim to enhance early detection of diabetes, potentially improving patient outcomes.
- Provide healthcare professionals with a tool for better identifying high-risk patients.
- Explore the interplay of various health indicators and lifestyle factors in diabetes risk.
- Contribute to personalized healthcare by enabling targeted intervention strategies

EXISTING SYSTEMS

- Existing systems primarily utilized traditional statistical and machine learning techniques.
- Focused on simpler models like Linear Regression and Logistic Regression.
- The approach involved minimal data preprocessing and feature engineering.
- Systems often did not fully address issues like class imbalance, impacting model accuracy.
- Relied on manual methods for data analysis, leading to inefficiencies.
- Lacked real-time data processing capabilities, crucial for prompt diabetes detection.
- Overall, these systems provided basic insights but lacked the advanced methodologies required for highly accurate and efficient diabetes prediction.

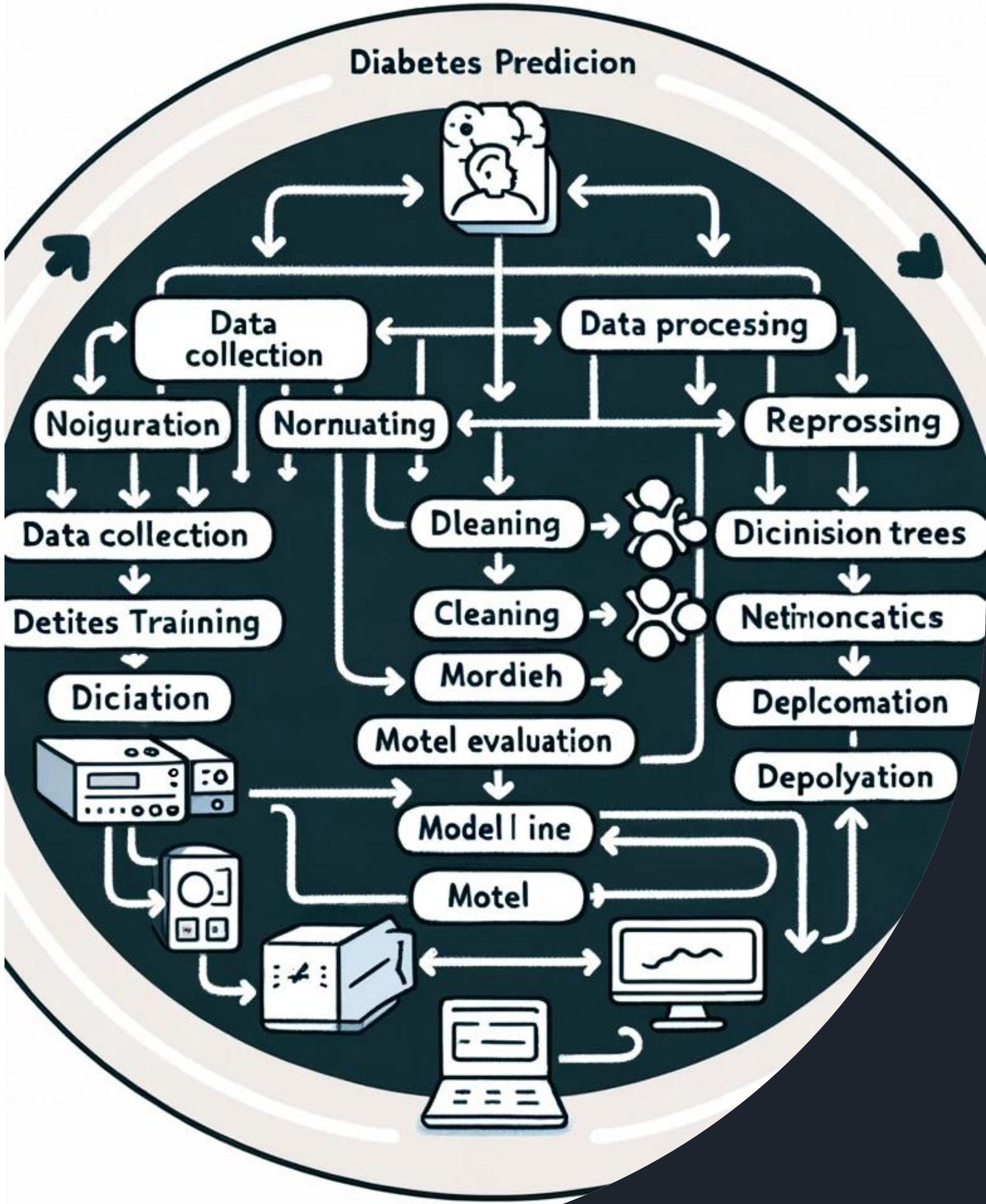
ISSUES IN EXISTING SYSTEM

- The system's limited adaptability to new and diverse datasets impacted its predictive capabilities.
- Faced challenges in integrating newer, more advanced machine learning techniques.
- Inadequate handling of categorical and continuous variables in the dataset.
- Struggled to provide interpretable results for non-technical stakeholders.
- Encountered difficulties in maintaining model performance over time.
- Issues with data privacy and security were not adequately addressed.
- Overall, these factors contributed to a less robust and versatile system for diabetes prediction.

PROPOSED SYSTEM

- The proposed system integrates advanced machine learning algorithms for enhanced diabetes prediction accuracy.
- Utilizes comprehensive data preprocessing, including SMOTE for class balancing and outlier removal.
- Incorporates sophisticated models like CatBoost, XGBoost, and LightGBM for better pattern recognition.
- Employs real-time data processing for timely and effective diabetes management.
- Designed for scalability and adaptability to handle larger and more diverse datasets.
- Focuses on producing interpretable results for a wider range of stakeholders.
- Prioritizes data privacy and security in the system's architecture and operations.

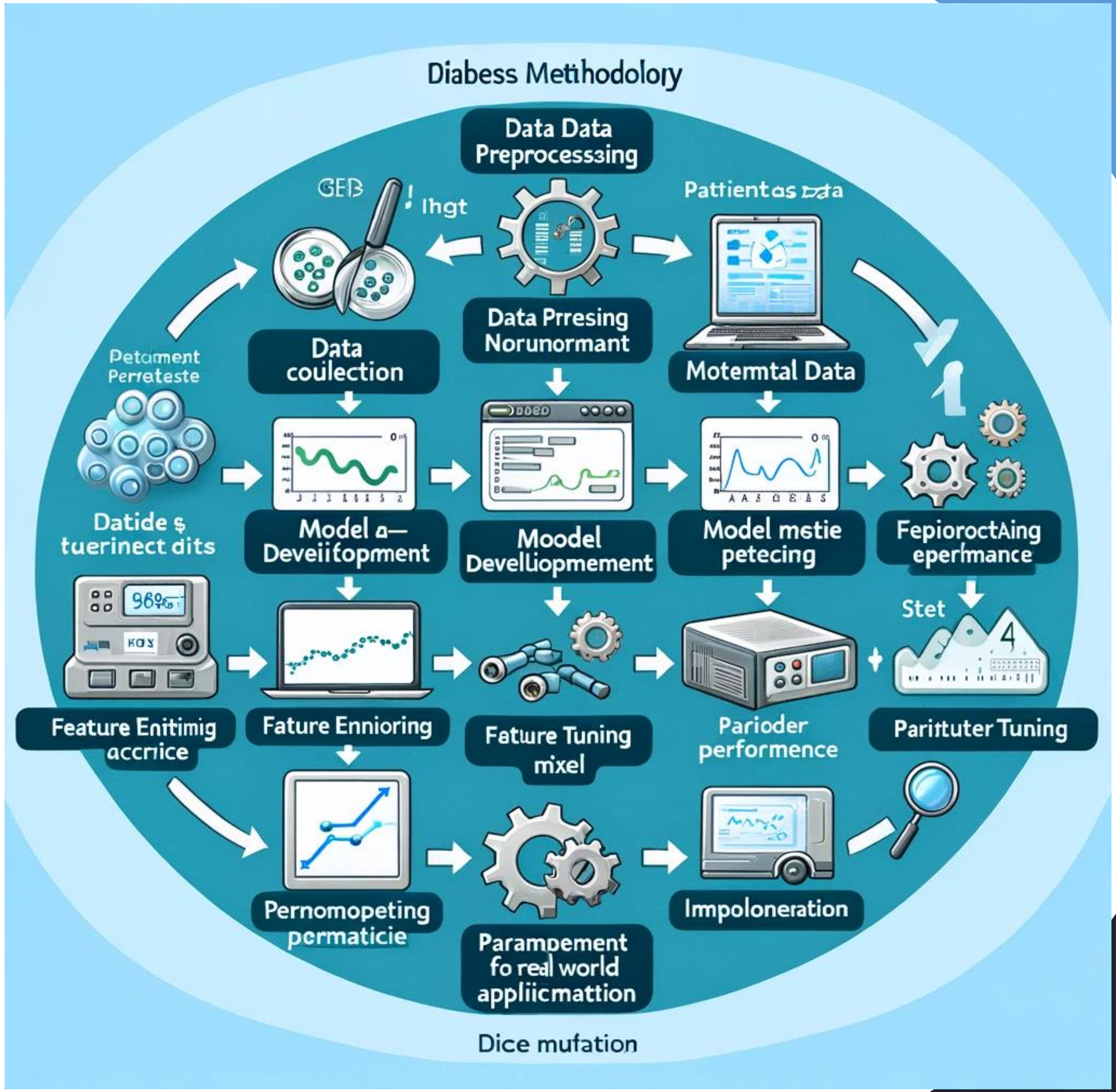
- This flowchart presents a streamlined overview of a diabetes prediction system, detailing key phases from data acquisition and cleansing to model development and testing, culminating in practical implementation.
- It visually guides through the systematic progression of tasks essential for effective diabetes prediction using machine learning techniques.



ADVANTAGES

- Advanced algorithms like CatBoost and XGBoost significantly improve prediction accuracy.
- The SMOTE technique ensures balanced data, reducing bias in predictions.
- Automated data preprocessing and outlier removal enhance dataset quality and reliability.
- Complex models enable identification of intricate patterns in data.
- The system's real-time data processing allows for prompt and effective diabetes management.
- Designed for scalability, efficiently handling larger and more diverse datasets.
- Produces results that are understandable to a broad audience, including non-technical stakeholders.
- Incorporates measures to ensure the security and privacy of sensitive patient data.

METHODOLOGY



Data Exploration: Initial examination of the dataset using methods like head(), info(), and describe(). Assessment of null values, duplicates, and unique values in the data.

Data Visualization: Utilizing histograms, count plots, and stacked area charts to visualize the distribution of various features and relationships between variables.

Data Preprocessing: Including label encoding for categorical variables and handling class imbalance with SMOTE (Synthetic Minority Over-sampling Technique).

Model Training and Evaluation: Implementing a range of machine learning models such as Logistic Regression, KNN, Decision Trees, Random Forest, XGBoost, CatBoost, Gradient Boosting, and LightGBM. Evaluation based on accuracy, precision, recall, F1-score, and ROC-AUC scores.

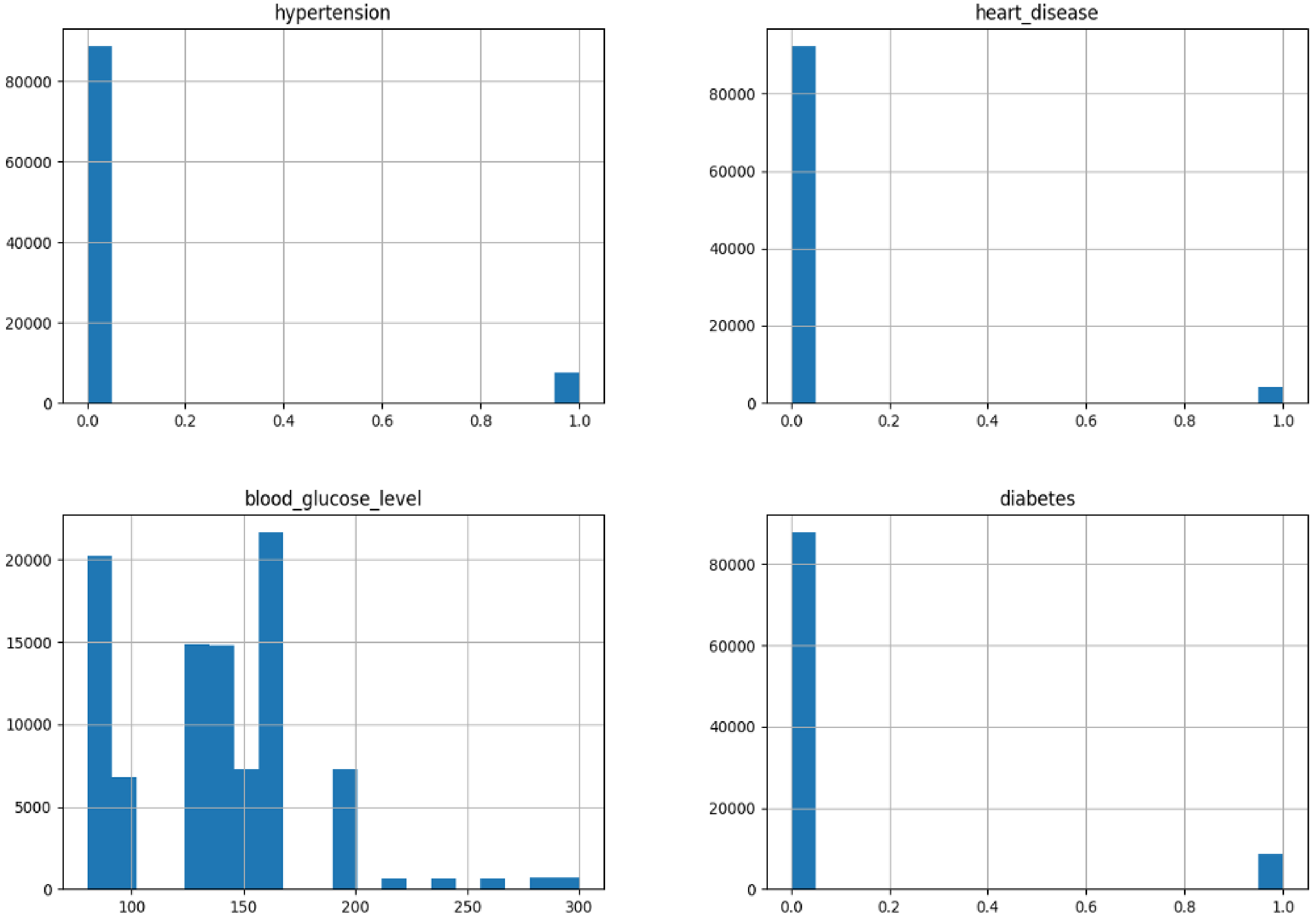
Cross-Validation: Performing k-fold cross-validation to assess the CatBoost model's performance.

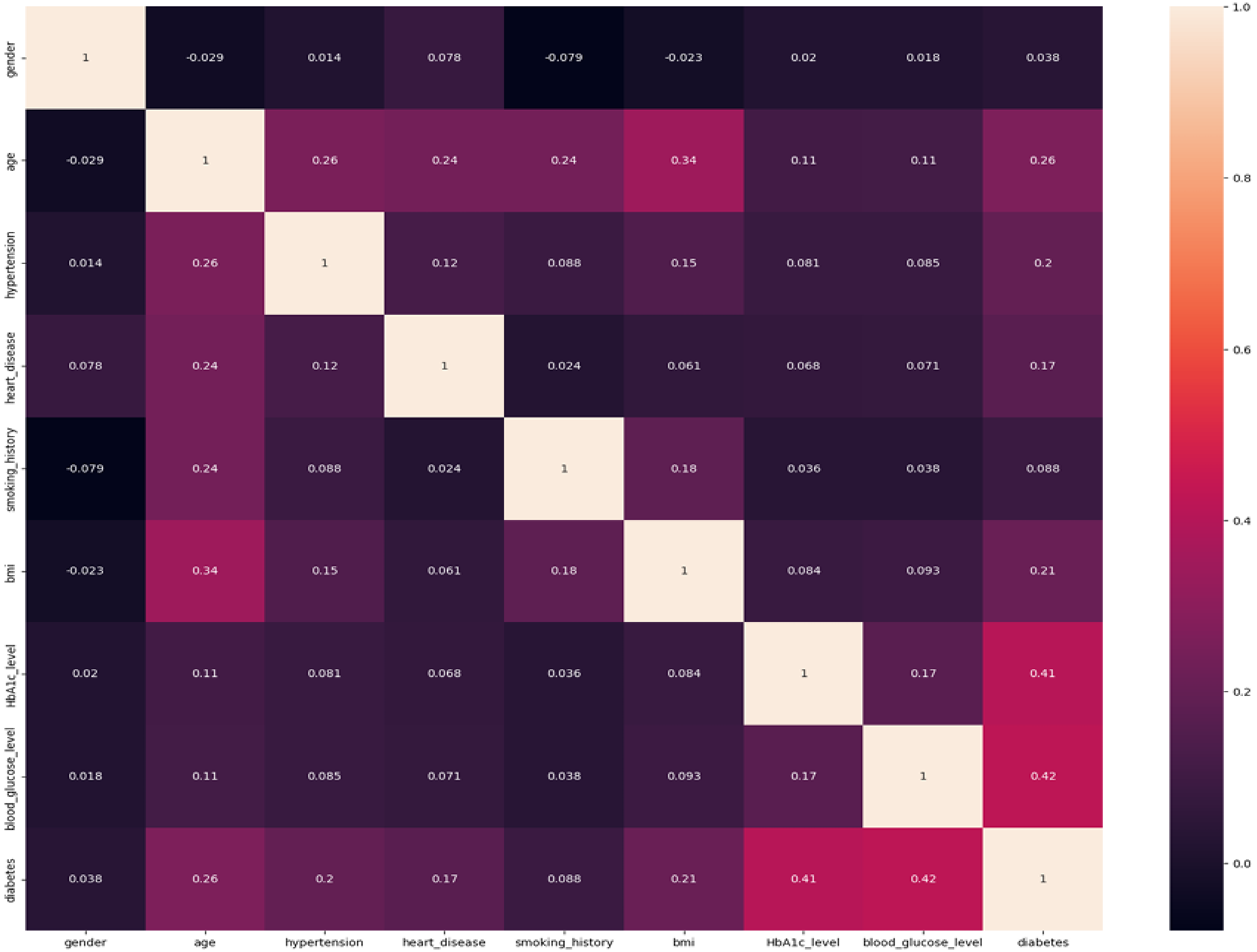
Neural Network Implementation: Building and training a neural network model using TensorFlow/Keras, including data standardization and model evaluation.

Model Deployment: Saving the trained model for future predictions.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   gender               100000 non-null  object 
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64  
 3   heart_disease        100000 non-null  int64  
 4   smoking_history      100000 non-null  object 
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64  
 8   diabetes             100000 non-null  int64  
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
```

```
Out[5]: (100000, 9)
```


hypertension
heart_disease
blood_glucose_level
diabetes
0
20000
40000
60000
80000
0.0
0.2
0.4
0.6
0.8
1.0
5000
10000
15000
100
150
200
250
300


gender
age
hypertension
heart_disease
smoking_history
bmi
HbA1c_level
blood_glucose_level
diabetes
1 -0.029 0.014 0.078 -0.079 -0.023 0.02 0.018 0.038
-0.029 1 0.26 0.24 0.24 0.34 0.11 0.11 0.26
0.014 0.26 1 0.12 0.088 0.15 0.081 0.085 0.2
0.078 0.24 0.12 1 0.024 0.061 0.068 0.071 0.17
-0.079 0.24 0.088 0.024 1 0.18 0.036 0.038 0.088
-0.023 0.34 0.15 0.061 0.18 1 0.084 0.093 0.21
0.02 0.11 0.081 0.068 0.036 0.084 1 0.17 0.41
0.018 0.11 0.085 0.071 0.038 0.093 0.17 1 0.42
0.038 0.26 0.2 0.17 0.088 0.21 0.41 0.42 1
1.0
0.8
0.6
0.4
0.2
0.0

Logistic Regression Accuracy: 0.89

Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.89	0.88	0.89	17439
1	0.88	0.90	0.89	17627
accuracy			0.89	35066
macro avg	0.89	0.89	0.89	35066
weighted avg	0.89	0.89	0.89	35066

```
Best Parameters: {'kneighborsclassifier__n_neighbors': 3, 'kneighborsclassifier__weights': 'distance'}
Model Accuracy: 0.934181258198825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.90      0.93     17439
           1       0.91      0.96      0.94     17627

    accuracy                           0.93     35066
   macro avg       0.94      0.93      0.93     35066
weighted avg       0.94      0.93      0.93     35066
```

Decision Tree Model Accuracy: 0.9709690298294644

Decision Tree Classification Report:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	17439
1	0.98	0.96	0.97	17627
accuracy			0.97	35066
macro avg	0.97	0.97	0.97	35066
weighted avg	0.97	0.97	0.97	35066

Model Accuracy: 0.9236582444533166

Classification Report:

	precision	recall	f1-score	support
0	0.93	0.91	0.92	17439
1	0.92	0.93	0.92	17627
accuracy			0.92	35066
macro avg	0.92	0.92	0.92	35066
weighted avg	0.92	0.92	0.92	35066

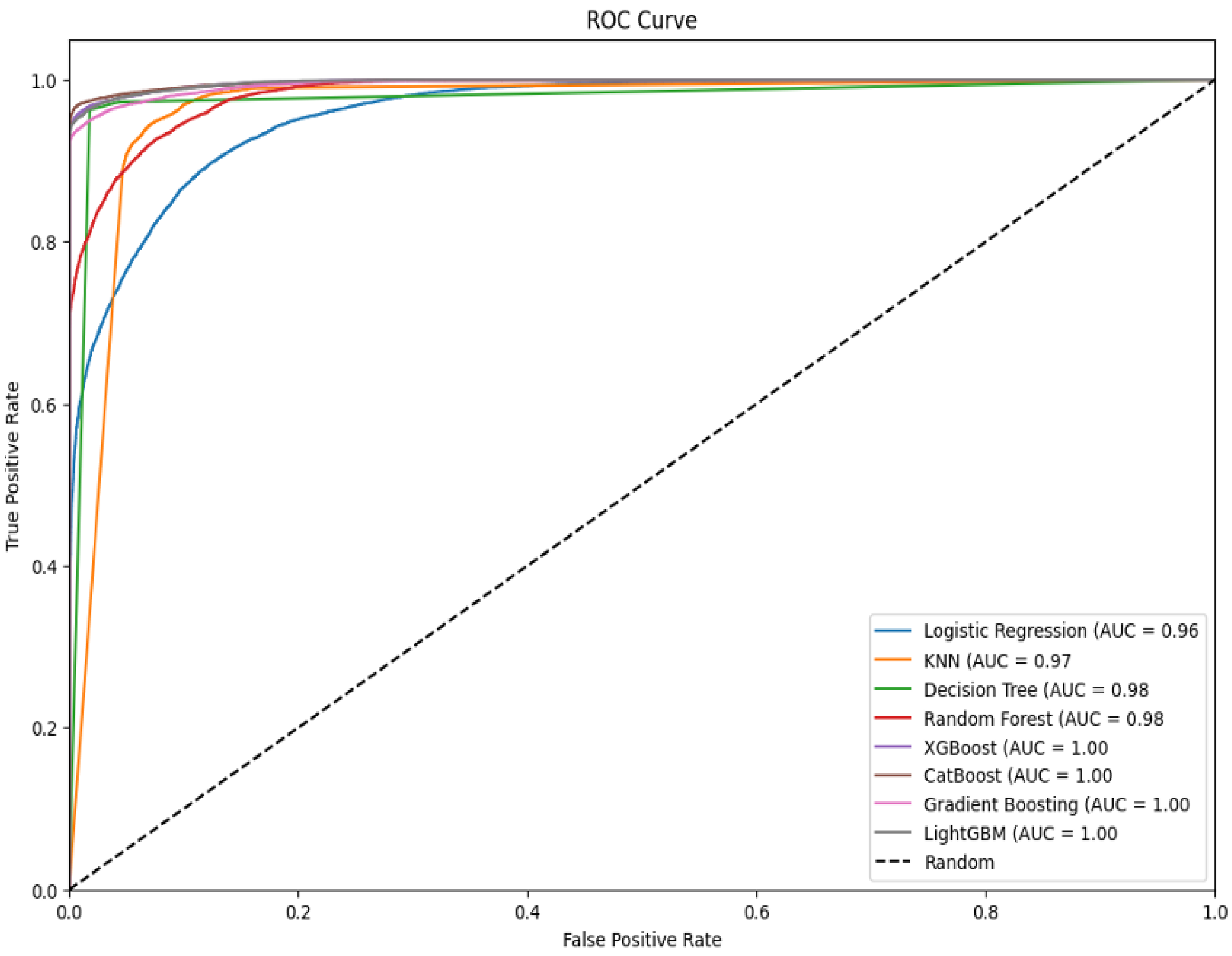
```
XGBoost Model Accuracy: 0.9750185364740774
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     17439
           1       0.99      0.96      0.97     17627

    accuracy                           0.98     35066
   macro avg       0.98      0.98      0.98     35066
weighted avg       0.98      0.98      0.98     35066
```

```
CatBoost Model Accuracy: 0.9804083727827525
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     17439
           1       0.99      0.97      0.98     17627

    accuracy                           0.98     35066
   macro avg       0.98      0.98      0.98     35066
weighted avg       0.98      0.98      0.98     35066
```

Gradient Boosting Model Accuracy: 0.9658643700450579

Classification Report:

	precision	recall	f1-score	support
0	0.95	0.98	0.97	17439
1	0.98	0.95	0.97	17627
accuracy			0.97	35066
macro avg	0.97	0.97	0.97	35066
weighted avg	0.97	0.97	0.97	35066


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression (AUC = 0.96
KNN (AUC = 0.97
Decision Tree (AUC = 0.98
Random Forest (AUC = 0.98
XGBoost (AUC = 1.00
CatBoost (AUC = 1.00
Gradient Boosting (AUC = 1.00
LightGBM (AUC = 1.00
Random

```
Epoch 20/20
1754/1754 - 6s - loss: 0.1780 - accuracy: 0.9138 - val_loss: 0.1716 - val_accuracy: 0.9163 - 6s/epoch - 4ms/
step
1096/1096 [==============================] - 3s 3ms/step - loss: 0.1738 - accuracy: 0.9143
Test Accuracy: 91.43%
```

LITERATURE REVIEW

- CatBoost: https://arxiv.org/abs/1706.09516
- XGBoost: https://arxiv.org/abs/1603.02754
- LightGBM: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
- Gradient Boosting: https://arxiv.org/abs/1908.06951
- Neural Network: https://arxiv.org/abs/1404.7828

REQUIREMENTS

Hardware Requirements:

- Processor: Minimum 6-8 cores for efficient parallel processing.
- RAM: Minimum 16GB, recommended 32GB or higher for large datasets.
- Hard Drive: Minimum 1TB SSD, with higher capacities like 2TB beneficial for extensive data.

Software Requirements:

- Programming Language: Python for scripting and algorithm development.
- Data Analysis Tools: Jupyter Notebook or RStudio for exploratory data analysis.
- Machine Learning Libraries: Pandas, NumPy, Scikit-learn, TensorFlow, XGBoost, CatBoost.
- Database Management: SQL or NoSQL for data storage and retrieval.
- Version Control: Git for source code management and collaboration.
- Deployment Platforms: Cloud services like AWS, Azure, or GCP for scalability.
- Data Security: Software for data encryption and regulatory compliance.

REFRENCES

- Diabetes prediction dataset (kaggle.com)
- ChatGPT
- GitHub

THANK YOU