DIABETES PREDICTION USING MACHINE LEARNING

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

S.SAI CHARAN (20UECS0905) **(15980)**
M.DINESH (20UECS0613) **(18160)**
B.CHARAN SAI (20UECS0088) **(17645)**

Under the guidance of
Dr.G.MARIAMMAL,M.E,Ph.D.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

DIABETES PREDICTION USING MACHINE LEARNING

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

S.SAI CHARAN	(20UECS0905)	**(15980)**
M.DINESH	(20UECS0613)	**(18160)**
B.CHARAN SAI	(20UECS0088)	**(17645)**

Under the guidance of
Dr.G.MARIAMMAL,M.E,Ph.D.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "DIABETES PREDICTION USING MACHINE LEARNING" by "S.SAI CHARAN (20UECS0905), M.DINESH (20UECS0613), B.CHARAN SAI (20UECS0088) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor	**Signature of Professor In-charge**
Computer Science & Engineering	**Computer Science & Engineering**
School of Computing	**School of Computing**
Vel Tech Rangarajan Dr. Sagunthala R&D	**Vel Tech Rangarajan Dr. Sagunthala R&D**
Institute of Science & Technology	**Institute of Science & Technology**
May, 2024	**May, 2024**

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(S.SAI CHARAN)
Date: / /

(Signature)
(M.DINESH)
Date: / /

(Signature)
(B.CHARAN SAI)
Date: / /

APPROVAL SHEET

This project report entitled DIABETES PREDICTION USING MACHINE LEARNING by S.SAI CHARAN (20UECS0905), M.DINESH (20UECS0613), B.CHARAN SAI (20UECS0088) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr.G.MARIAMMAL,M.E,Ph.D.,ASSISTANT PROFESSOR,.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering,Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Dr.G.MARIAMMAL,M.E,Ph.D.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

S.SAI CHARAN (20UECS0905)
M.DINESH (20UECS0613)
B.CHARAN SAI (20UECS0088)

ABSTRACT

Diabetes mellitus is a chronic metabolic disorder characterized by elevated blood sugar levels, affecting millions of individuals worldwide. Early detection and management of diabetes are crucial in preventing complications and improving patient outcomes. In recent years, machine learning (ML) techniques have shown promise in predicting diabetes risk based on various factors such as demographic information, medical history, and lifestyle habits. This paper provides a comprehensive review and comparative analysis of machine-learning approaches for diabetes prediction. We systematically explore the methodologies, datasets, features, performance metrics, and challenges associated with existing ML-based models. Furthermore, we identify the strengths and limitations of different algorithms including support vector machines (SVM), decision trees, random forests, logistic regression, neural networks, and ensemble methods.

Keywords: Decision trees, Ensemble methods, Logistic regression, Neural networks, Performance metrics, Random forests, Support vector machines (SVM).

LIST OF FIGURES

4.1	**General Architecture**	11
4.2	**Data Flow Diagram**	12
4.3	**Use Case Diagram**	13
4.4	**Class Diagram**	14
4.5	**Sequence Diagram**	15
4.6	**Collaboration Diagram**	16
4.7	**Activity Diagram**	17
5.1	**Input Data**	22
5.2	**Output**	23
5.3	**Unit Testing Output Data**	24
5.4	**Integration Testing Output Data**	25
5.5	**System Testing Output Data**	26
5.6	**Test Image**	27
6.1	**Result For Diabetes Patient**	34
6.2	**Result For Normal Patient**	35
8.1	**Plagiarism report**	38
9.1	**Poster Representation**	45

LIST OF ACRONYMS AND ABBREVIATIONS

EHR	Electronic Health Records
HPC	High Performence Computing
IDF	International Diabetes Federation
ML	Machine Learning
ROC	Receiver Operating Characteristic Curve
SVM	Support Vector Machine

TABLE OF CONTENTS

Page.No

ABSTRACT **v**

LIST OF FIGURES **vi**

LIST OF ACRONYMS AND ABBREVIATIONS **vii**

1 INTRODUCTION **1**

1.1 Introduction 1
1.2 Aim of the Project 3
1.3 Project Domain 3
1.4 Scope of the Project 3

2 LITERATURE REVIEW **4**

3 PROJECT DESCRIPTION **6**

3.1 Existing System 6
3.2 Proposed System 7
3.3 Feasibility Study 8
3.3.1 Economic Feasibility 9
3.3.2 Technical Feasibility 9
3.3.3 Social Feasibility 9
3.4 System Specification 9
3.4.1 Hardware Specification 10
3.4.2 Software Specification 10
3.4.3 Standards and Policies 10

4 METHODOLOGY **11**

4.1 General Architecture 11
4.2 Design Phase 12
4.2.1 Data Flow Diagram 12
4.2.2 Use Case Diagram 13
4.2.3 Class Diagram 14

4.2.4 Sequence Diagram . . . 15
4.2.5 Collaboration diagram . . . 16
4.2.6 Activity Diagram . . . 17
4.3 Algorithm & Pseudo Code . . . 18
4.3.1 Algorithm . . . 18
4.3.2 Pseudo Code . . . 18
4.4 Module Description . . . 19
4.4.1 Module1 . . . 19
4.4.2 Module2 . . . 19
4.5 Steps to execute/run/implement the project . . . 20
4.5.1 Step1 . . . 20
4.5.2 Step2 . . . 20
4.5.3 Step3 . . . 20

5 IMPLEMENTATION AND TESTING 22
5.1 Input and Output . . . 22
5.1.1 Input Design . . . 22
5.1.2 Output Design . . . 23
5.2 Testing . . . 23
5.3 Types of Testing . . . 23
5.3.1 Unit testing . . . 23
5.3.2 Integration testing . . . 24
5.3.3 System testing . . . 25
5.3.4 Test Result . . . 27

6 RESULTS AND DISCUSSIONS 28
6.1 Efficiency of the Proposed System . . . 28
6.2 Comparison of Existing and Proposed System . . . 29
6.3 Sample Code . . . 29

7 CONCLUSION AND FUTURE ENHANCEMENTS 36
7.1 Conclusion . . . 36
7.2 Future Enhancements . . . 36

8 PLAGIARISM REPORT 38

9 SOURCE CODE & POSTER PRESENTATION **39**
9.1 Source Code 39
9.2 Poster Presentation 45

References **45**

Chapter 1

INTRODUCTION

1.1 Introduction

Diabetes mellitus is a chronic metabolic disorder characterized by elevated blood sugar levels, affecting millions of individuals worldwide. Early detection and effective management of diabetes are paramount in preventing complications and improving patient outcomes. With the advancement of technology, particularly in the realm of machine learning (ML), there has been a growing interest in leveraging ML techniques for predicting diabetes risk based on various factors such as demographic information, medical history, and lifestyle habits.

This project aims to explore and implement machine learning algorithms to predict the risk of diabetes in individuals. By analyzing diverse datasets and employing sophisticated ML models, we seek to develop an accurate and reliable prediction system that can assist healthcare professionals in identifying individuals at high risk of developing diabetes.

In this project, we will conduct a comprehensive review and comparative analysis of existing ML-based models for diabetes prediction. We will systematically explore the methodologies, datasets, features, performance metrics, and challenges associated with these models. Furthermore, we will identify the strengths and limitations of different ML algorithms, including support vector machines (SVM), decision trees, random forests, logistic regression, neural networks, and ensemble methods, in the context of diabetes prediction.

By the end of this project, we aim to deliver a robust and scalable ML-based diabetes prediction system that can be potentially integrated into healthcare systems to aid in early detection and proactive management of diabetes, thereby improving patient outcomes and reducing the burden of this chronic disease on individuals and healthcare systems alike.

Diabetes mellitus, a chronic metabolic disorder characterized by elevated blood glucose levels, has emerged as a global health challenge with significant implications for public health, healthcare systems, and individual well-being. According to the International Diabetes Federation (IDF), approximately 463 million adults were living with diabetes worldwide in 2019, and this number is projected to rise to 700 million by 2045. Diabetes not only imposes a substantial economic burden on healthcare systems but also increases the risk of various complications, including cardiovascular disease, neuropathy, retinopathy, and kidney failure, leading to reduced quality of life and increased mortality rates.

Early detection and intervention are critical for effectively managing diabetes and mitigating its complications. Machine learning (ML) techniques, with their ability to analyze large and complex datasets, hold immense promise for improving the accuracy and efficiency of diabetes prediction. By leveraging diverse sets of patient data, including demographic information, clinical variables, lifestyle factors, and biomarkers, ML algorithms can identify patterns and relationships that may not be apparent through traditional statistical methods.

In recent years, there has been a surge of interest in developing ML-based models for diabetes prediction, spurred by advancements in data science, computational power, and data availability. These models range from traditional regression-based approaches to more sophisticated ensemble methods and deep learning architectures. However, despite the proliferation of research in this area, several challenges remain, including data heterogeneity, model interpretability, and generalizability across diverse populations.

The findings of this study have the potential to inform clinical practice, public health policy, and future research directions in the field of predictive healthcare analytics. By elucidating the strengths and limitations of ML-based approaches for diabetes prediction, and clinically relevant prediction models that can assist healthcare practitioners in early risk stratification and personalized intervention strategies.

1.2 Aim of the Project

The aim of the project "Diabetes prediction using machine learning" would likely be to develop a predictive model that can analyze various factors or features associated with individuals and predict the likelihood of them developing diabetes in the future. This predictive model could help in early diagnosis and intervention, enabling better management and prevention strategies for diabetes.

1.3 Project Domain

The project domain for diabetes prediction using machine learning resides within the realm of healthcare and medical informatics. Specifically, it focuses on leveraging advanced machine learning techniques to address the pressing challenge of diabetes prevention and management. Within this domain, the project delves into the intricate interplay of physiological, genetic, lifestyle, and environmental factors that contribute to the onset and progression of diabetes. By harnessing healthcare data, including electronic health records (EHR), medical imaging data, genetic information, and wearable device data, the project aims to develop accurate predictive models capable of identifying individuals at high risk of developing diabetes.

1.4 Scope of the Project

The project on diabetes prediction using machine learning encompasses a multifaceted approach aimed at developing a robust predictive model to ascertain the likelihood of individuals having diabetes based on various factors. Beginning with the precise definition of the problem, the scope extends to encompassing data collection and preprocessing, where relevant datasets are curated and processed to ensure their suitability for analysis.

Chapter 2

LITERATURE REVIEW

[1]Luo, W, et al, (2010), Author proposed this systematic review paper provides a comprehensive overview of the application of machine learning techniques for diabetes prediction. It explores various methodologies, algorithms, and datasets used in previous studies.

[2]Kavakiotis, I., et al, (2017), Author proposed this research article presents an overview of machine learning and data mining methods employed in diabetes research. The authors discuss various techniques used for data analysis, including classification, clustering, and regression.

[3]Fernández, A., et al, (2014), Author proposed this book provides a comprehensive overview of learning from imbalanced data sets, a common challenge in machine learning applications. The authors discuss various techniques and algorithms for addressing class imbalance, including resampling methods, cost-sensitive learning, and ensemble techniques.

[4]Sathyanarayana, A., et al, (2016), Author published this preprint presents strategies for addressing the challenge of imbalanced datasets in the context of deep learning. The authors discuss techniques such as class weighting, oversampling, and threshold adjustment to mitigate the effects of class imbalance on deep learning models.

[5]Banerjee, M., et al, (2017), Author proposed this research article focuses on predicting diabetes using machine learning methods. The authors explore the application of various machine learning techniques for diabetes prediction, including logistic regression, decision trees, support vector machines, and neural networks.

[6]Al-Masri, E., et al, (2018), Author proposed this research article presents a study on diabetes prediction using machine learning techniques. The authors explore

the application of various machine learning algorithms, such as logistic regression, decision trees, random forests, and support vector machines, for predicting the likelihood of diabetes occurrence.

[7]Waseem, M., et al, (2020), Author proposed the analyze the state-of-the-art methodologies, algorithms, and datasets employed in previous studies. They discuss the strengths and limitations of different machine learning approaches for diabetes prediction and highlight emerging trends and future research directions in the field.

[8]Anwar, et al, (2021), Author proposed this review article provides an extensive overview of medical image analysis techniques using convolutional neural networks (CNNs). While the primary focus is on medical image analysis, the application of CNNs in healthcare extends to various domains, including diabetes prediction.

[9]Chaudhary, et al, (2022), Author proposed this review article provides an in-depth examination of data mining techniques employed in healthcare data analysis. While not specific to diabetes prediction, the review encompasses various data mining methodologies relevant to healthcare applications, including predictive modeling, clustering, association rule mining, and anomaly detection.

[10]Sharma, A., et al, (2010), Author proposed this review article provides a comprehensive examination of diabetes prediction using machine learning techniques. The authors critically analyze the methodologies, algorithms, and datasets employed in previous studies focused on predicting diabetes risk.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system for diabetes prediction using machine learning relies primarily on structured data sources such as electronic health records (EHR) and laboratory test results. It typically employs traditional machine learning algorithms such as logistic regression, decision trees, and support vector machines for predicting diabetes risk. Feature engineering techniques used in the existing system are often based on expert knowledge and domain-specific rules, aiming to extract relevant features from the data.

Model interpretability is a key consideration in the existing system, with an emphasis on building models that are easy to understand and interpret by healthcare professionals. However, the existing system may face challenges in scalability and adaptability, as it may struggle to incorporate new data sources or adapt to changes in patient populations or healthcare practices.

Advantages

- The existing system utilizes machine learning algorithms to achieve higher accuracy in diabetes prediction compared to traditional risk assessment methods.
- Machine learning models employed in the existing system enable early detection of individuals at risk of developing diabetes.
- The system provides personalized risk assessment by considering individual characteristics.
- By integrating diverse sources of data including clinical, genetic, and lifestyle information.
- The system is scalable for population-level screening and healthcare applications, allowing for efficient management of large volumes of patient data.

Disadvantages

- Some machine learning algorithms employed in the existing system lack interpretability.
- Issues related to data quality, such as missing values, data incompleteness, and biases, may affect the performance.
- The existing system may be susceptible to overfitting, where the model learns noise or irrelevant patterns from the training data.
- Complex machine learning models utilized in the existing system may require significant computational resources.
- The use of sensitive health data in the existing system raises ethical and privacy concerns, including patient consent.

3.2 Proposed System

The proposed system for diabetes prediction using machine learning introduces several advancements over the existing approach. It integrates a broader range of data sources, including wearable device data, genetic information, dietary records, and social determinants of health, for more comprehensive risk assessment. The proposed system utilizes advanced feature engineering methods, including automatic feature selection, feature extraction from unstructured data, and deep learning-based feature representation learning, to capture complex relationships and patterns in the data.

Moreover, the proposed system moves towards personalized risk assessment by incorporating individual characteristics (e.g., age, gender, ethnicity, comorbidities) and adapting models to specific patient profiles, enabling tailored preventive interventions and treatment strategies. With a focus on rigorous clinical validation, stakeholder engagement, and regulatory compliance, the proposed system aims to address the limitations of the existing approach and ensure the safety, effectiveness, and usability of diabetes prediction in clinical practice.

Advantages

- The proposed system aims to improve prediction accuracy by employing advanced machine learning algorithms and incorporating novel features.
- The proposed system may utilize sophisticated feature selection and engineering techniques to identify the most informative predictors of diabetes.
- By integrating interpretable machine learning models or providing post-hoc explanations for predictions.
- Through personalized risk assessment, the proposed system can support the development of targeted intervention strategies.
- The proposed system may incorporate mechanisms to address ethical and privacy concerns, such as ensuring data confidentiality.

Disadvantages

- Developing and implementing the proposed system may require significant computational resources, expertise in machine learning.
- Ensuring the validity and generalizability of the proposed system across diverse populations and healthcare contexts may be challenging.
- The effectiveness of the proposed system may depend on the availability and quality of relevant data sources.
- Compliance with regulatory requirements, such as data protection regulations (e.g., GDPR, HIPAA), and legal considerations, including liability for errors or misuse of predictive models.
- The success of the proposed system may hinge on user acceptance and adoption by healthcare providers.

3.3 Feasibility Study

A feasibility study for diabetes prediction using machine learning involves assessing the practicality and viability of implementing such a system. Evaluate the availability and accessibility of relevant data sources for diabetes prediction, including electronic health records, laboratory test results, and patient demographics.

3.3.1 Economic Feasibility

Evaluate the costs associated with developing and implementing the machine learning system for diabetes prediction, including data acquisition, model development, infrastructure, and maintenance. Compare these costs to the potential benefits, such as improved patient outcomes, reduced healthcare costs, and increased efficiency in diabetes management.

3.3.2 Technical Feasibility

Evaluate the availability and quality of data required for training and validating machine learning models for diabetes prediction. Assess whether sufficient data sources are accessible, including electronic health records, laboratory tests, wearable device data, and patient-reported information.

3.3.3 Social Feasibility

Assess the social acceptance and adoption of the diabetes prediction system among stakeholders, including healthcare professionals, patients, policymakers, and the broader community. Consider factors such as perceived usefulness, ease of use, privacy concerns, and ethical implications related to data privacy and security.

3.4 System Specification

- Develop a system for predicting diabetes risk using machine learning techniques.
- The system should be able to Collect and preprocess data from various sources, including electronic health records, laboratory tests, and wearable devices.
- Achieve high prediction accuracy, with metrics such as sensitivity, specificity, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC).
- Handle large volumes of data efficiently, with scalable algorithms and computational resources.

- Ensure data privacy and security by implementing measures such as encryption, access control, and compliance with healthcare regulations.

3.4.1 Hardware Specification

- High-performance computing (HPC) infrastructure for training machine learning models, including CPU, GPU, or specialized accelerators (e.g., TPU).
- Sufficient memory and storage capacity to handle large datasets and model parameters.
- High-speed network connectivity for data transfer between servers and storage systems.
- Cloud-based infrastructure or distributed computing frameworks for elastic scaling and resource provisioning.

3.4.2 Software Specification

- Python as the primary programming language for data preprocessing, model training, and prediction generation.
- Machine learning libraries and frameworks such as scikit-learn, TensorFlow, and PyTorch for implementing algorithms and building models.
- Data visualization libraries (e.g., Matplotlib, Seaborn) for exploring and visualizing datasets.

3.4.3 Standards and Policies

Python

Python is a popular programming language widely used for diabetes prediction using machine learning due to its simplicity, versatility, and extensive ecosystem of libraries and frameworks tailored for data science and machine learning tasks. Its flexibility and scalability make it suitable for both research and production environments, enabling developers to create robust, efficient, and user-friendly applications for diabetes prediction and management.

Standard Used: ISO/OS /2 PEP8

Chapter 4

METHODOLOGY

4.1 General Architecture

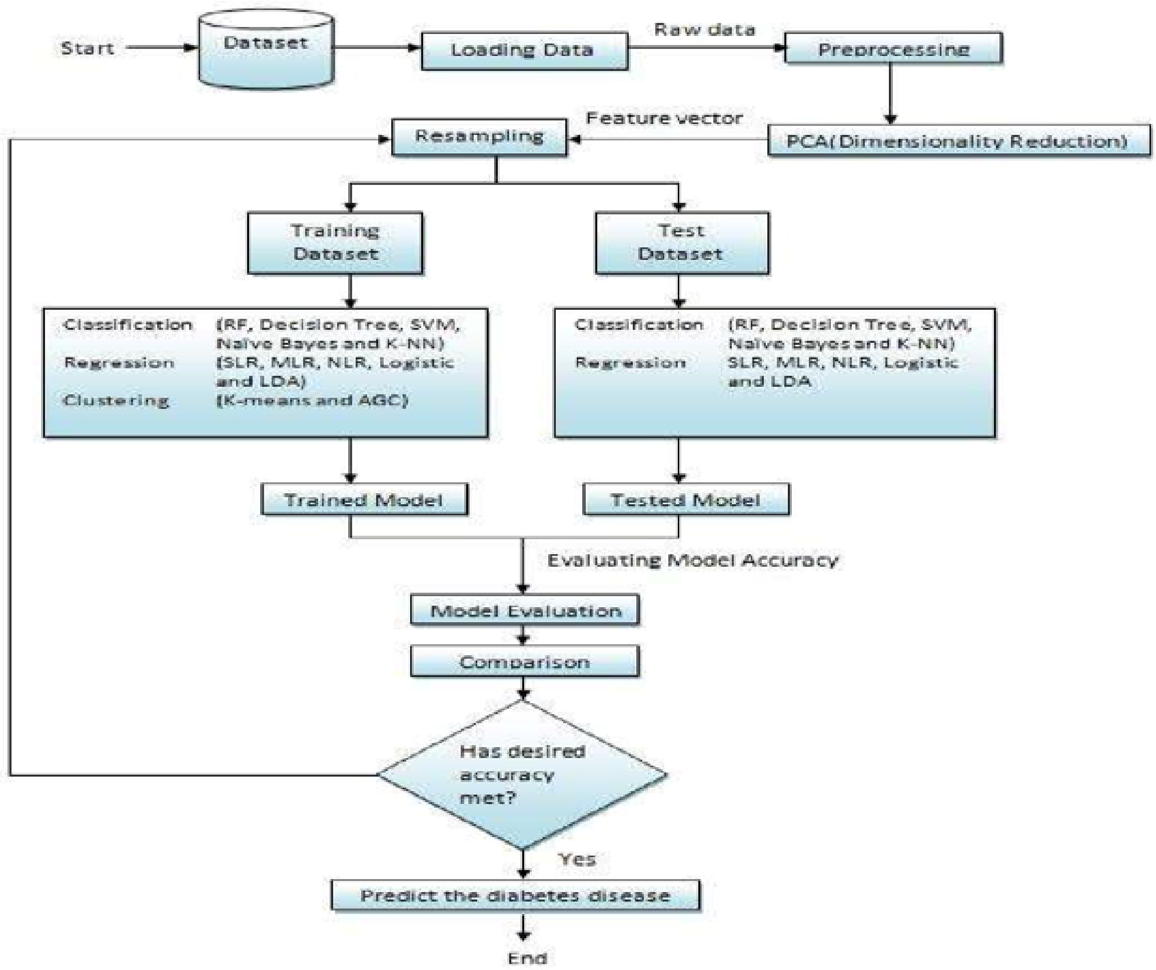

Figure 4.1: **General Architecture**

In Figure 4.1 it depicts the various components involved, such as data sources (e.g., electronic health records, wearable devices), machine learning models, data preprocessing modules, and user interfaces. The architecture diagram also shows the interactions between these components and the flow of data through the system, from data acquisition to prediction generation.

4.2 Design Phase

4.2.1 Data Flow Diagram

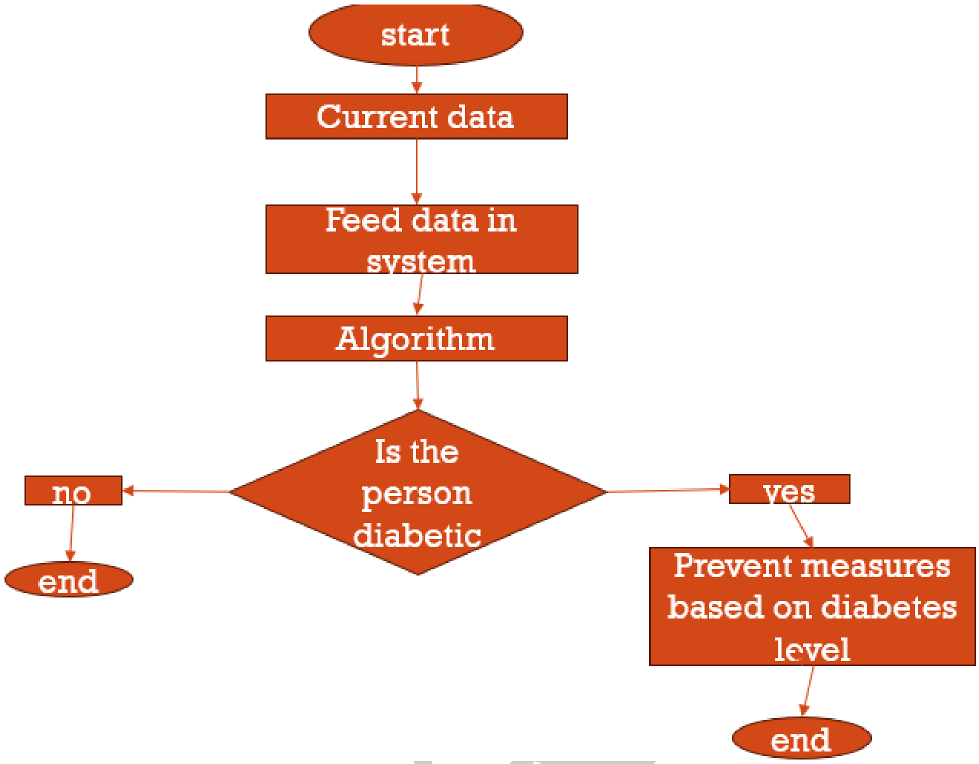

Figure 4.2: **Data Flow Diagram**

In Figure 4.2 the dataflow diagram (DFD) illustrates the flow of data through the system for diabetes prediction using machine learning. It shows how data moves from data sources (e.g., patient records, sensor data) to data processing modules (e.g., preprocessing, feature extraction) to machine learning models (e.g., training, prediction) and finally to output interfaces (e.g., visualization, reporting).

4.2.2 Use Case Diagram

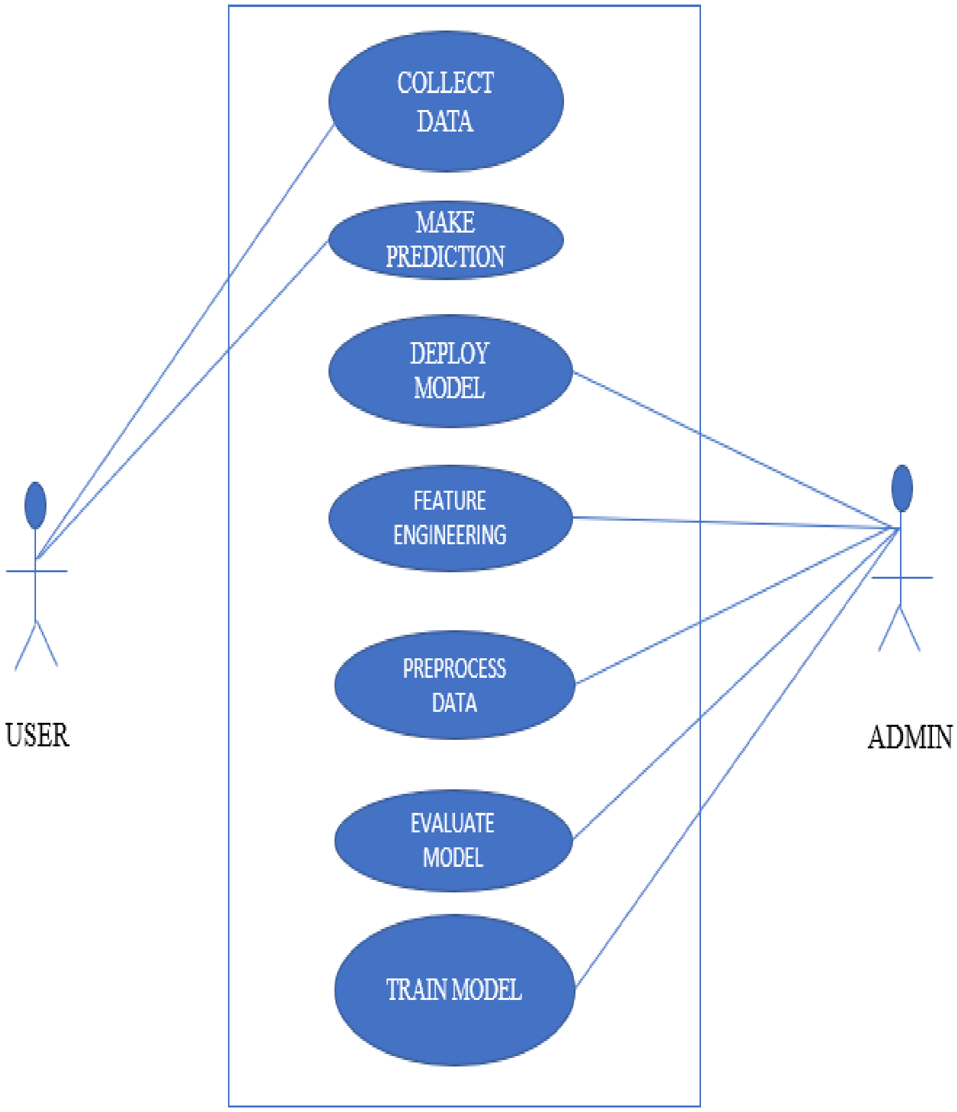

Figure 4.3: **Use Case Diagram**

In Figure 4.3 the use case diagram provides a high-level overview of the functionalities and interactions within the diabetes prediction system using machine learning. It helps stakeholders understand the system's capabilities and how different actors interact with the system to achieve specific goals related to diabetes prediction and management.

4.2.3 Class Diagram

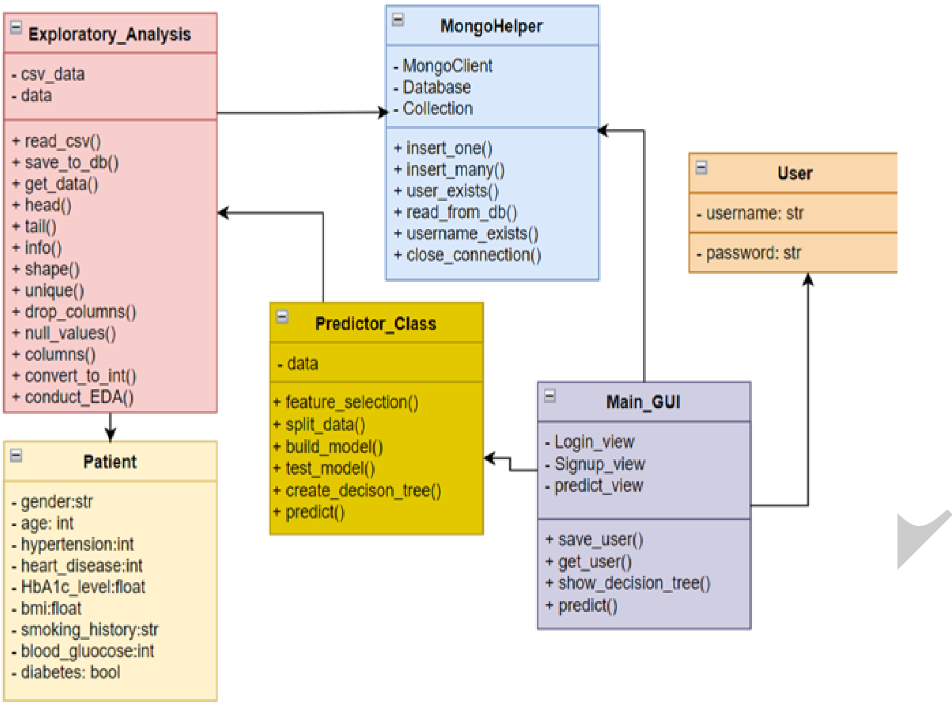

Figure 4.4: **Class Diagram**

In Figure 4.4 in the context of diabetes prediction using machine learning, the class diagram represents the static structure of the system by illustrating the classes, attributes, methods, and relationships between objects. It includes classes such as ”PatientData,” ”FeatureExtractor,” ”ModelTrainer,” ”PredictionModel,” and ”User-Interface.” The class diagram helps developers understand the organization of code and data within the system and facilitates software design and implementation.

4.2.4 Sequence Diagram

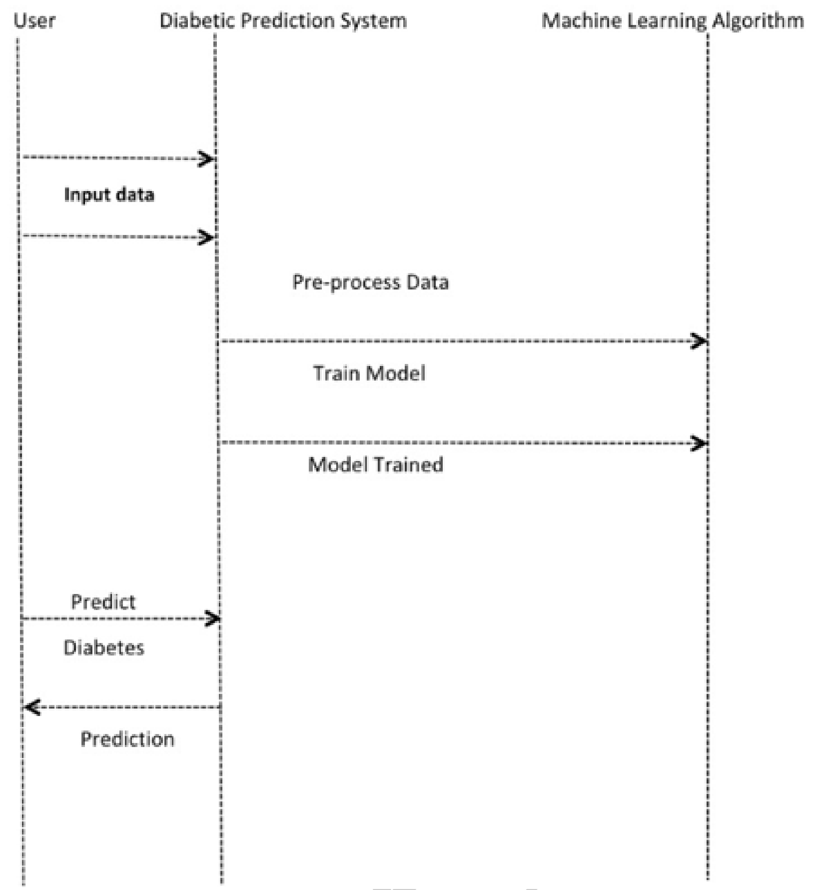

Figure 4.5: **Sequence Diagram**

In Figure 4.5 the sequence diagram for diabetes prediction using machine learning illustrates the interactions between system components over time. It shows the sequence of steps involved in the prediction process, including data preprocessing, model training, model evaluation, and prediction generation. The sequence diagram helps visualize the flow of control and data between different modules or layers in the system.

4.2.5 Collaboration diagram

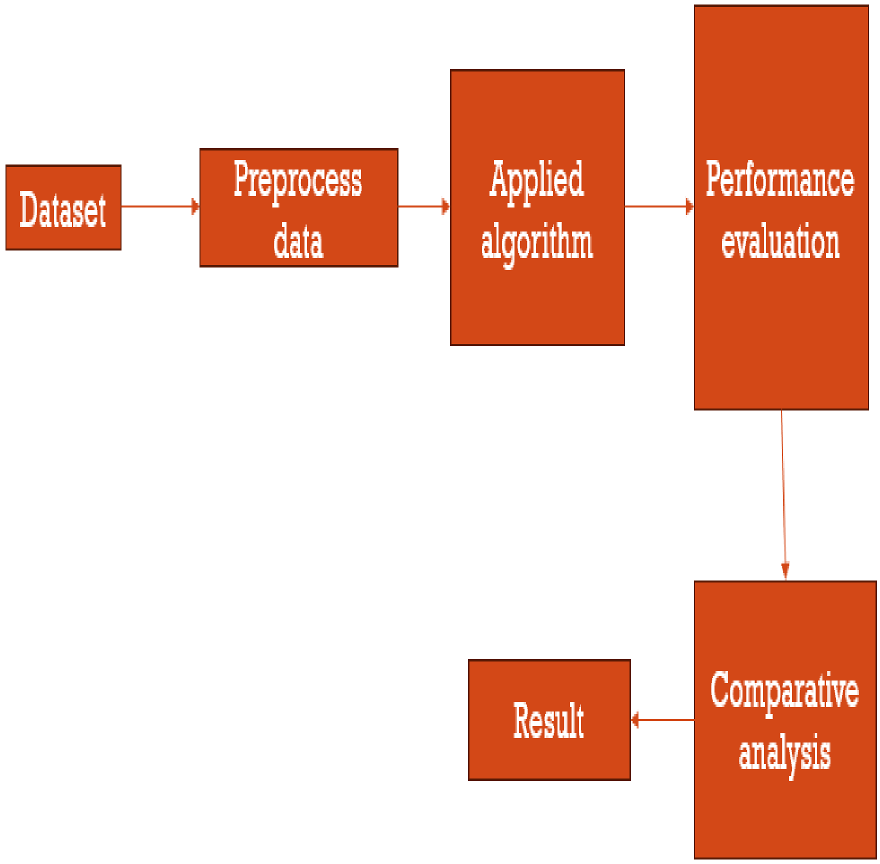

Figure 4.6: **Collaboration Diagram**

In Figure 4.6 a collaboration diagram, also known as a communication diagram, illustrates the interactions between system components for diabetes prediction using machine learning. It shows the relationships and messages exchanged between objects or modules, including data transfers, method calls, and control flows. The collaboration diagram helps visualize how different components collaborate to perform tasks and achieve the system's objectives.

4.2.6 Activity Diagram

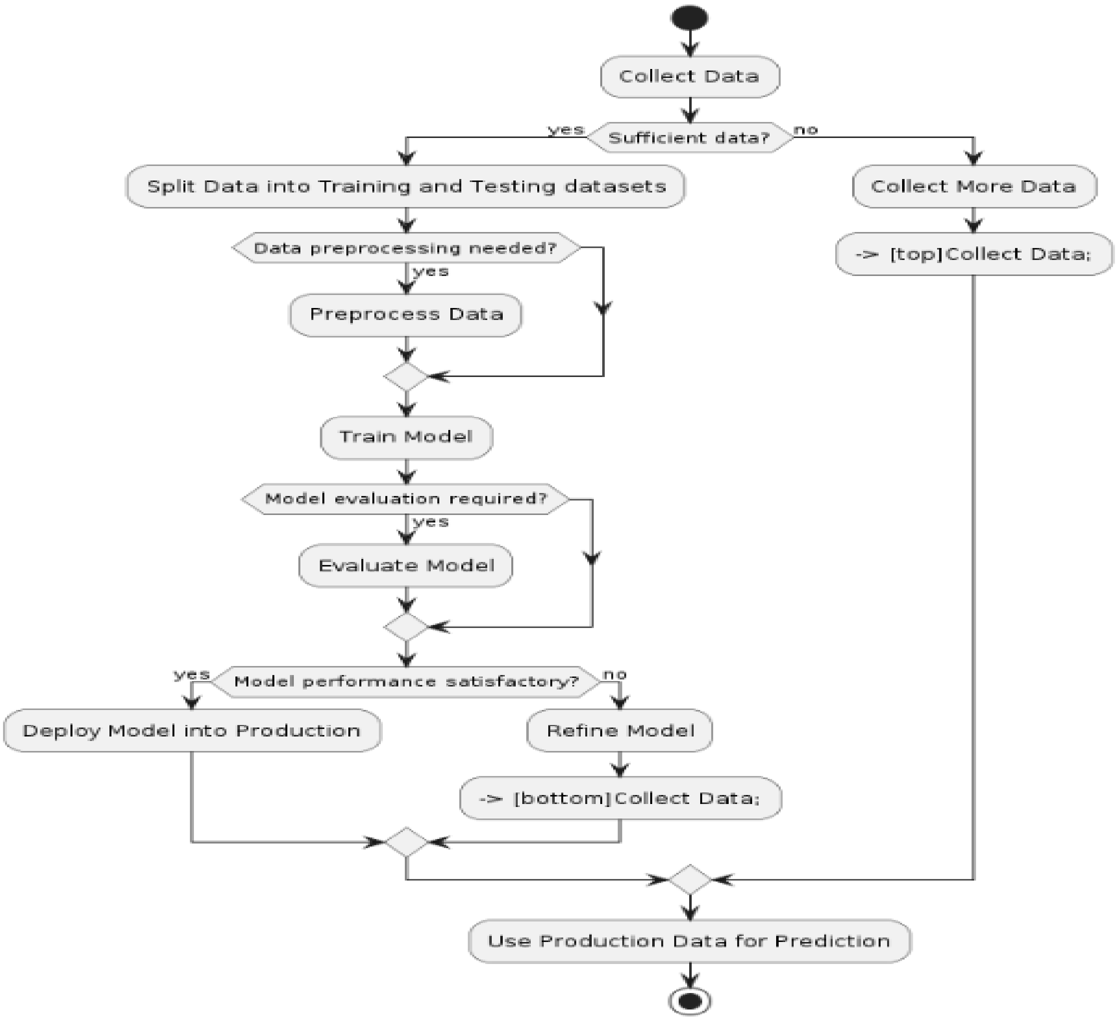

Figure 4.7: **Activity Diagram**

In Figure 4.7 an activity diagram for diabetes prediction using machine learning models the workflow of the prediction process. It depicts the sequence of activities involved, such as data collection, data preprocessing, feature extraction, model training, model evaluation, and prediction generation. Decision points, branching paths, and loops may be included to represent different scenarios or conditions in the prediction process.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

- Loads the dataset.
- Splits the dataset into features (X) and the target variable (y).
- Splits the data into training and testing sets.
- Standardizes the features to have mean 0 and variance 1.
- Initializes a logistic regression model.
- Trains the model on the training data.
- Makes predictions on the testing data.
- Evaluates the model's performance using accuracy and classification report.

4.3.2 Pseudo Code

```
1. Import necessary libraries: pandas, StandardScaler, train_test_split, accuracy_score
2. Load the PIMA Diabetes Dataset into a pandas DataFrame.
3. Display the first 5 rows of the dataset and its shape.
4. Display statistical measures of the dataset.
5. Display the count of diabetic and non-diabetic individuals.
6. Data Cleaning:
   - Drop duplicate rows.
   - Check for NULL values and handle them if any.
   - Check for missing values in specific columns and replace them with the mean.
7. Data Visualization:
   - Create a pie chart to visualize the distribution of outcomes (diabetic and non-diabetic).
   - Create a count plot to visualize the distribution of outcomes.
8. Check if the dataset is balanced or skewed by plotting a histogram.
9. Analyze relationships between variables:
   - Perform correlation analysis and visualize it using a heatmap.
10. Separate the independent variables (X) and dependent variable (y).
11. Standardize the independent variables using StandardScaler.
12. Split the dataset into training and testing sets.
13. Classification Models:
    1) Logistic Regression:
       - Fit the logistic regression model to the training data.
    2) k-Nearest Neighbors (KNN):
       - Fit the KNN model to the training data.
    3) Naive Bayes:
       - Fit the Naive Bayes model to the training data.
    4) Support Vector Machine (SVM):
```

```
      - Fit the SVM model to the training data.
    5) Decision Tree:
      - Fit the Decision Tree model to the training data.
    6) Random Forest:
      - Fit the Random Forest model to the training data.
14. Predictions & Evaluation:
    - Make predictions using the testing data for all models.
    - Calculate the accuracy score for each model.
15. Save the model with the highest accuracy using pickle.
```

4.4 Module Description

4.4.1 Module1

Data Collection and Preprocessing

Gather relevant datasets containing information about individuals, including features like age, weight, height, blood pressure, cholesterol levels, family history of diabetes, etc. Ensure that the data is from reliable sources and is representative of the population you want to predict for.

Impute missing values using techniques like mean, median, or sophisticated imputation methods. Identify the most relevant features for predicting diabetes using techniques like correlation analysis, feature importance ranking, or domain expertise. Data normalization or standardization: Scale the features to ensure they have similar ranges and distributions.

4.4.2 Module2

Model Development and Evaluation Choose appropriate machine learning algorithms for classification tasks. Common choices include logistic regression, decision trees, random forests. Train the selected models using the training data. Evaluate the trained models using the testing data. Measure performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).

4.5 Steps to execute/run/implement the project

4.5.1 Step1

Data collection, Data preprocessing

- Load the dataset into your development environment.
- Handle missing values, outliers, and inconsistencies in the data.
- Encode categorical variables into numerical representations.
- Further preprocess the data as needed, such as feature scaling or normalization.

4.5.2 Step2

Model Building, Model Evaluation, Model Deployment

- Choose appropriate machine learning algorithms for diabetes prediction, such as logistic regression.
- Compute evaluation metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix.
- Visualize model performance using plots such as ROC curves, precision-recall curves, and confusion matrices.
- Deploy the best-performing model into a production environment.
- Integrate the model into the deployment environment and ensure it meets regulatory requirements and data privacy regulations.

4.5.3 Step3

Documentation and Testing, Execution and Monitoring

- Document the project, including details about data preprocessing, model building, evaluation, and deployment steps.
- Write unit tests to ensure the correctness and robustness of the implemented functionalities.
- Provide usage examples and tutorials for other users or team members.

- Execute the diabetes prediction system in the production environment.
- Continuously monitor model performance and update the model periodically.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

Pregnancie	Glucose	BloodPres	SkinThickn	Insulin	BMI	DiabetesPe	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0
8	99	84	0	0	35.4	0.388	50	0
7	196	90	0	0	39.8	0.451	41	1
9	119	80	35	0	29	0.263	29	1
11	143	94	33	146	36.6	0.254	51	1
10	125	70	26	115	31.1	0.205	41	1

diabetes

Figure 5.1: **Input Data**

In Figure 5.1 the input features for a diabetes prediction model using machine learning involves selecting the most relevant information from available data that can help in accurately predicting the likelihood of someone having diabetes.

5.1.2 Output Design

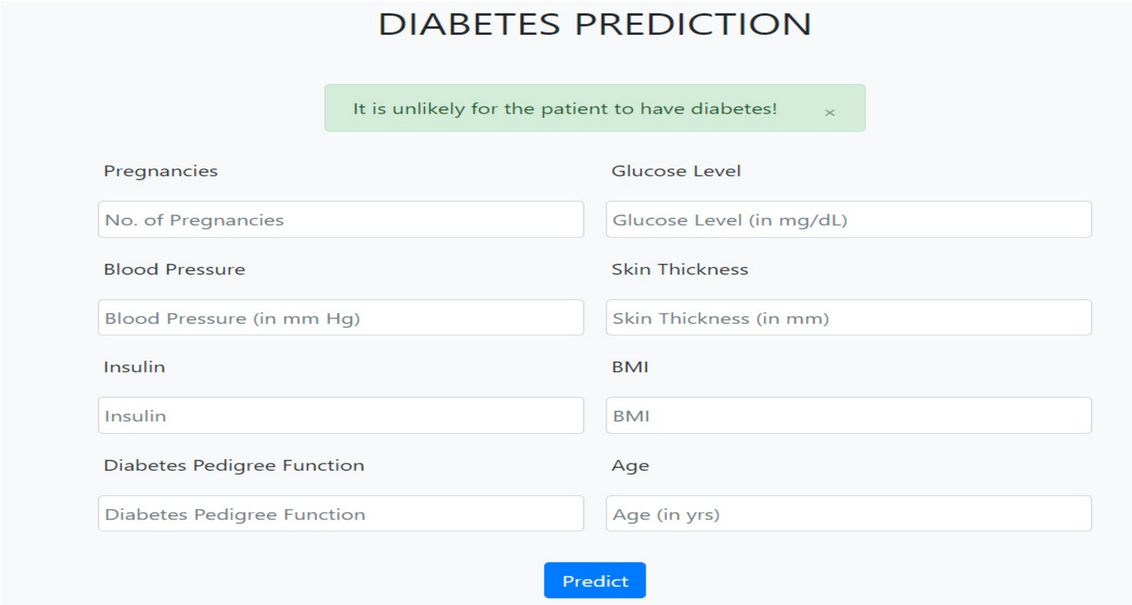

Figure 5.2: **Output**

In Figure 5.2 the output design for a diabetes prediction model using machine learning involves determining how the model's predictions will be presented or used to make informed decisions.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Input

```
import unittest

class TestDiabetesPrediction(unittest.TestCase):
    def test_model_prediction(self):
        # Mock input features
        input_features = np.array([[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]])

        # Perform prediction
        prediction = model.predict(input_features)

        # Assert prediction result
```

```
        self.assertIn(prediction[0], [0, 1])  # Ensure prediction is binary (0 or 1)

if __name__ == '__main__':
    unittest.main()
```

Test result

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure 5.3: **Unit Testing Output Data**

In Figure 5.3 Unit testing in the context of diabetes prediction using machine learning involves testing individual components or units of the codebase to ensure they function correctly.

5.3.2 Integration testing

Input

```
def test_diabetes_prediction_integration():
    # Mock input features
    input_features = np.array([[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]])

    # Perform prediction
    prediction = model.predict(input_features)
```

```

    # Assert prediction result
    assert prediction[0] in [0, 1]  # Ensure prediction is binary (0 or 1)

test_diabetes_prediction_integration()
```

Test result

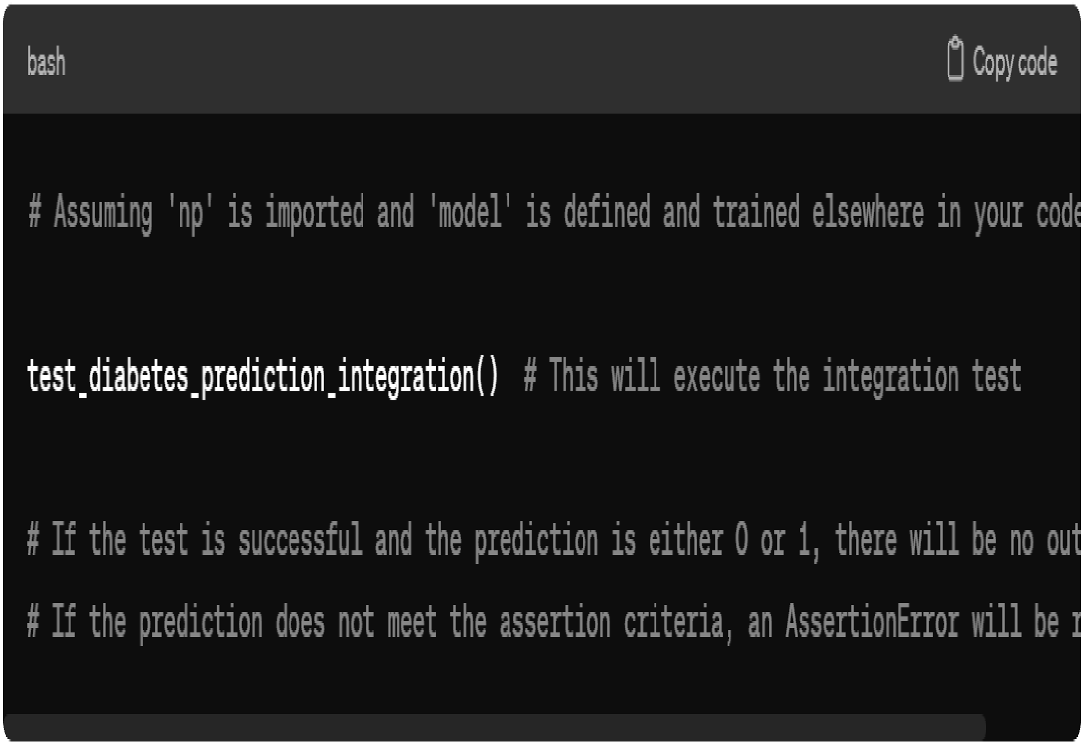

Figure 5.4: **Integration Testing Output Data**

In Figure 5.4 Integrating testing for a diabetes prediction model using machine learning involves ensuring that the model performs as expected within the broader system or application where it will be deployed.

5.3.3 System testing

Input

```
def test_diabetes_prediction_system():
    # Mock input features
    input_features = np.array([[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]])

    # Perform prediction
```

```
    prediction = model.predict(input_features)

    # Assert prediction result
    assert prediction[0] in [0, 1]  # Ensure prediction is binary (0 or 1)

test_diabetes_prediction_system()
```

Test Result

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure 5.5: **System Testing Output Data**

In Figure 5.5 System testing of a diabetes prediction system using machine learning is crucial to ensure its accuracy, reliability, and effectiveness before deployment in real-world settings.

5.3.4 Test Result

Figure 5.6: **Test Image**

In Figure 5.6 the output of a diabetes prediction model using machine learning typically consists of the model's prediction regarding whether an individual is likely to have diabetes or not.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of a proposed system for diabetes prediction using logistic regression depends on various factors, including the quality of data, feature selection, model training, and evaluation metrics. The accuracy and reliability of predictions heavily rely on the quality of the input data. High-quality data, free from errors and inconsistencies, can lead to more reliable predictions. Selecting relevant features or variables that have a significant impact on diabetes prediction is crucial. Feature selection techniques can help identify the most informative features, leading to more efficient models with improved performance.

Logistic regression models are relatively simple and computationally efficient compared to more complex models like neural networks. Training logistic regression models typically requires less computational resources and time, making them suitable for large datasets and real-time applications. Efficiency can be assessed using evaluation metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). A higher AUC-ROC value indicates better discrimination ability of the logistic regression model in distinguishing between positive and negative instances of diabetes.

The efficiency of the proposed system should also be evaluated in real-world settings, considering factors such as usability, scalability, and deployment feasibility. User feedback and performance monitoring can help identify areas for improvement and optimization. Overall, logistic regression can be an efficient and effective method for diabetes prediction, especially when coupled with appropriate data preprocessing, feature selection, and model evaluation techniques.

6.2 Comparison of Existing and Proposed System

Existing system:(Random Forest)

This existing system employs the Random Forest algorithm, a powerful ensemble learning method that combines multiple decision trees to make predictions. Historical data, including patient demographics, medical history, laboratory results, and lifestyle factors, is preprocessed to handle missing values, normalize features, and encode categorical variables. Feature selection techniques may be applied to identify the most relevant predictors of diabetes. The Random Forest model is then trained using the preprocessed data, where each decision tree is trained on a bootstrapped subset of the data and makes predictions independently. The final prediction is determined by aggregating the predictions of all individual trees, typically through a majority voting mechanism.

Proposed system:(logistic Regression)

The proposed system aims to enhance diabetes prediction using logistic regression by incorporating a broader range of data sources, including genetic information, lifestyle data, and social determinants of health. Advanced feature selection methods and model training techniques, such as transfer learning and ensemble learning, are utilized to improve model performance. Evaluation metrics are expanded to include measures such as area under the receiver operating characteristic curve (AUC-ROC) and calibration plots, providing a more comprehensive assessment of model calibration and discrimination ability. Moreover, the proposed system emphasizes real-world deployment, stakeholder engagement, and regulatory compliance, conducting rigorous clinical validation studies and addressing ethical considerations to ensure the safety, effectiveness, and usability of the system in clinical practice. Overall, the proposed system represents an evolution beyond the existing approach, leveraging advanced methodologies to enhance prediction accuracy, interpretability, personalization, and generalization for diabetes prediction using logistic regression.

6.3 Sample Code

```
Automatically generated by Colaboratory.

# *DIABETES PREDICTION*

```

```
---

Importing Dependencies
"""

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

"""Data Collection and Analysis

PIMA Diabetes Dataset
"""

#loading the dataset to a pandas df
df = pd.read_csv('diabetes.csv')

#printing the first 5 rows
df.head()

#no of rows and cols
df.shape

#getting the statistical measures of the df
df.describe()

#no of diabetics and non-diabetics
df['Outcome'].value_counts()

"""0 --> Non-Diabetic

1 --> Diabetic
"""



"""Data Cleaning

Drop duplicates
"""

print('Before dropping duplicates: ', df.shape)
df = df.drop_duplicates()
print('After dropping duplicates: ', df.shape)

"""Check for NULL values"""

df.isnull().sum()

"""Check for missing values"""
```

```

print('No of missing values in Glucose: ', df[df['Glucose'] == 0].shape[0])
print('No of missing values in BloodPressure: ', df[df['BloodPressure'] == 0].shape[0])
print('No of missing values in SkinThickness: ', df[df['SkinThickness'] == 0].shape[0])
print('No of missing values in Insulin: ', df[df['Insulin'] == 0].shape[0])
print('No of missing values in BMI: ', df[df['BMI'] == 0].shape[0])

"""Replace missing values with mean"""

df['Glucose'] = df['Glucose'].replace(0, df['Glucose'].mean())
df['BloodPressure'] = df['BloodPressure'].replace(0, df['BloodPressure'].mean())
df['SkinThickness'] = df['SkinThickness'].replace(0, df['SkinThickness'].mean())
df['Insulin'] = df['Insulin'].replace(0, df['Insulin'].mean())
df['BMI'] = df['BMI'].replace(0, df['BMI'].mean())

df.describe()

"""Data Visualisation

Count plot
"""

import matplotlib.pyplot as plt
f, ax = plt.subplots(1,2,figsize=(10,5))
df['Outcome'].value_counts().plot.pie(explode=[0,0.1],autopct='%1.1f%%', ax=ax[0],shadow=True)
ax[0].set_title('Outcome')
ax[0].set_ylabel('')

import seaborn as sns
sns.countplot('Outcome', data=df, ax=ax[1])
ax[1].set_title('Outcome')
N, P = df['Outcome'].value_counts()
print('Negative(0) ->', N)
print('Positive(1) ->', P)

plt.grid()
plt.show()

"""Dataset is not balanced

Histogram (data is balanced or skewed)
"""

df.hist(bins=10,figsize=(10,10))
plt.show()

"""Analysing relationships bw variables

Correlation analysis
"""
```

```
#get correlations of each feature in the dataset
corr_mat = df.corr()
top_corr_features = corr_mat.index
plt.figure(figsize=(10,10))
#plot heat map
g = sns.heatmap(df[top_corr_features].corr(), annot=True, cmap='RdYlGn')

"""Split data into X and y"""

#separating the independent and dependent variables
X = df.drop(columns='Outcome', axis=1)
y = df['Outcome']
print(X.head())
print(y.head())

"""Data Standardisation - Feature Scaling"""

scaler = StandardScaler()
scaler.fit(X)
standardised_data = scaler.transform(X)
print(standardised_data)

X = standardised_data
y = df.Outcome
print(X)
print(y)

"""Split data into training and testing data"""

#80% is train, 20% is test
#random state is used to ensure a specific split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=7)

print(X.shape, X_train.shape, X_test.shape)

"""Classification Models

1) Logistic Regression
"""

from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression(solver='liblinear', multi_class='ovr')
lr_model.fit(X_train, y_train)

"""2) K Neighbours Classifier"""

from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier()
knn_model.fit(X_train, y_train)
```

```python

"""3) Naive Bayes Classifier"""

from sklearn.naive_bayes import GaussianNB
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

"""4) Support Vector Machine(SVM)"""

from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(X_train, y_train)

"""5) Decision tree"""

from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)

"""6) Random Forest"""

from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(criterion='entropy')
rf_model.fit(X_train, y_train)

"""Predicting & Evaluating the Models"""

#make the predictions using test data for all 6 models
lr_preds = lr_model.predict(X_test)

knn_preds = knn_model.predict(X_test)

nb_preds = nb_model.predict(X_test)

svm_preds = svm_model.predict(X_test)

dt_preds = dt_model.predict(X_test)

rf_preds = rf_model.predict(X_test)

#get the accuracy of the models
print('Accuracy score of Logistic Regression:', round(accuracy_score(y_test, lr_preds) * 100, 2))
print('Accuracy score of KNN:', round(accuracy_score(y_test, knn_preds) * 100, 2))
print('Accuracy score of Naive Bayes:', round(accuracy_score(y_test, nb_preds) * 100, 2))
print('Accuracy score of SVM:', round(accuracy_score(y_test, svm_preds) * 100, 2))
print('Accuracy score of Decision Tree:', round(accuracy_score(y_test, dt_preds) * 100, 2))
print('Accuracy score of Random Forest:', round(accuracy_score(y_test, rf_preds) * 100, 2))

"""Save the Model with the Highest Accuracy using pickle"""

```

```
import pickle
pickle.dump(svm_model, open('svm_model.pkl', 'wb')) #svm has the highest accuracy

pickle.dump(scaler, open('scaler.pkl', 'wb')) #save the std scaler too
```

Output

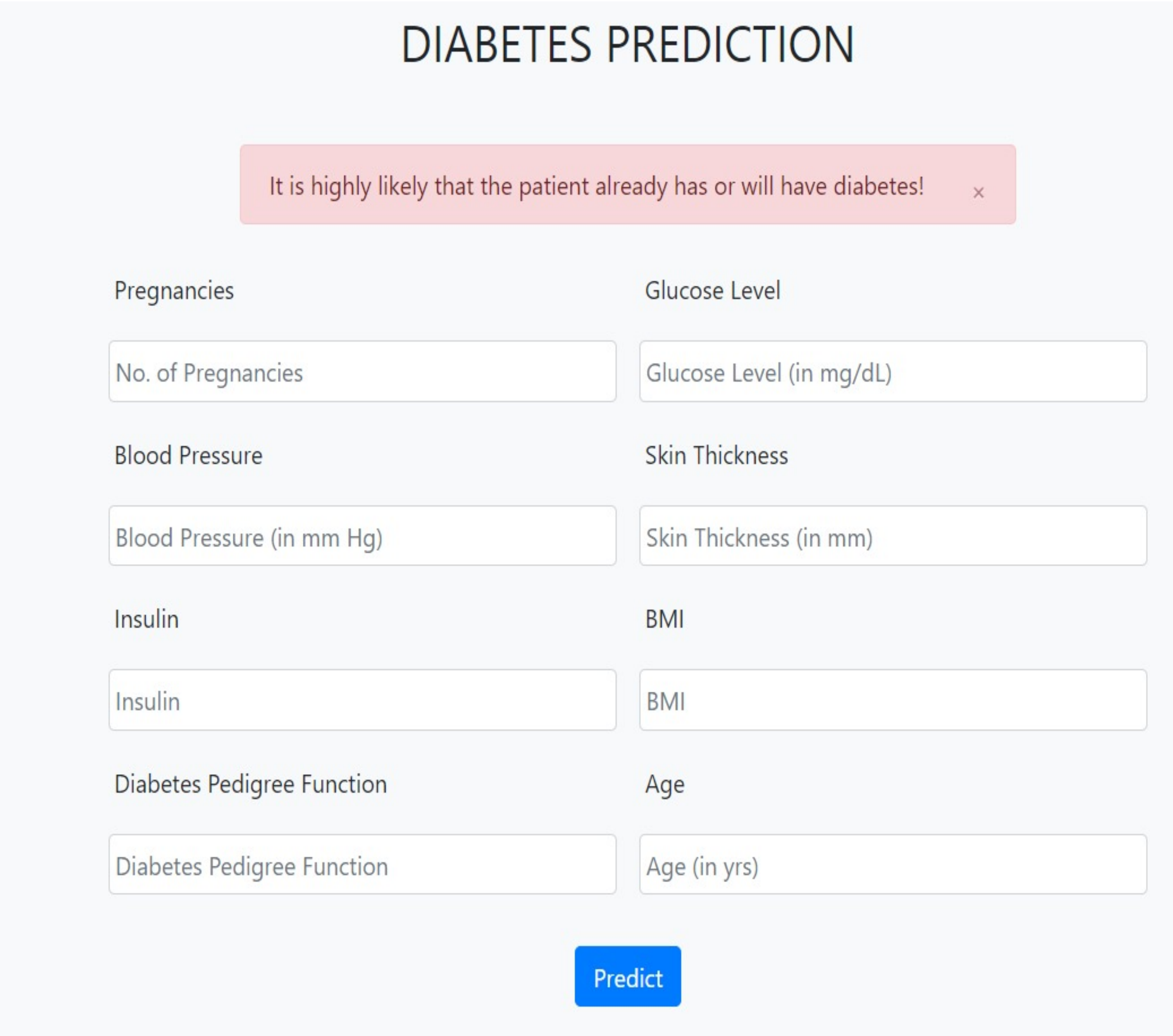

Figure 6.1: **Result For Diabetes Patient**

In Figure 6.1 the output of a diabetes prediction model using machine learning typically consists of the model's prediction regarding whether an individual is likely to have diabetes or not.

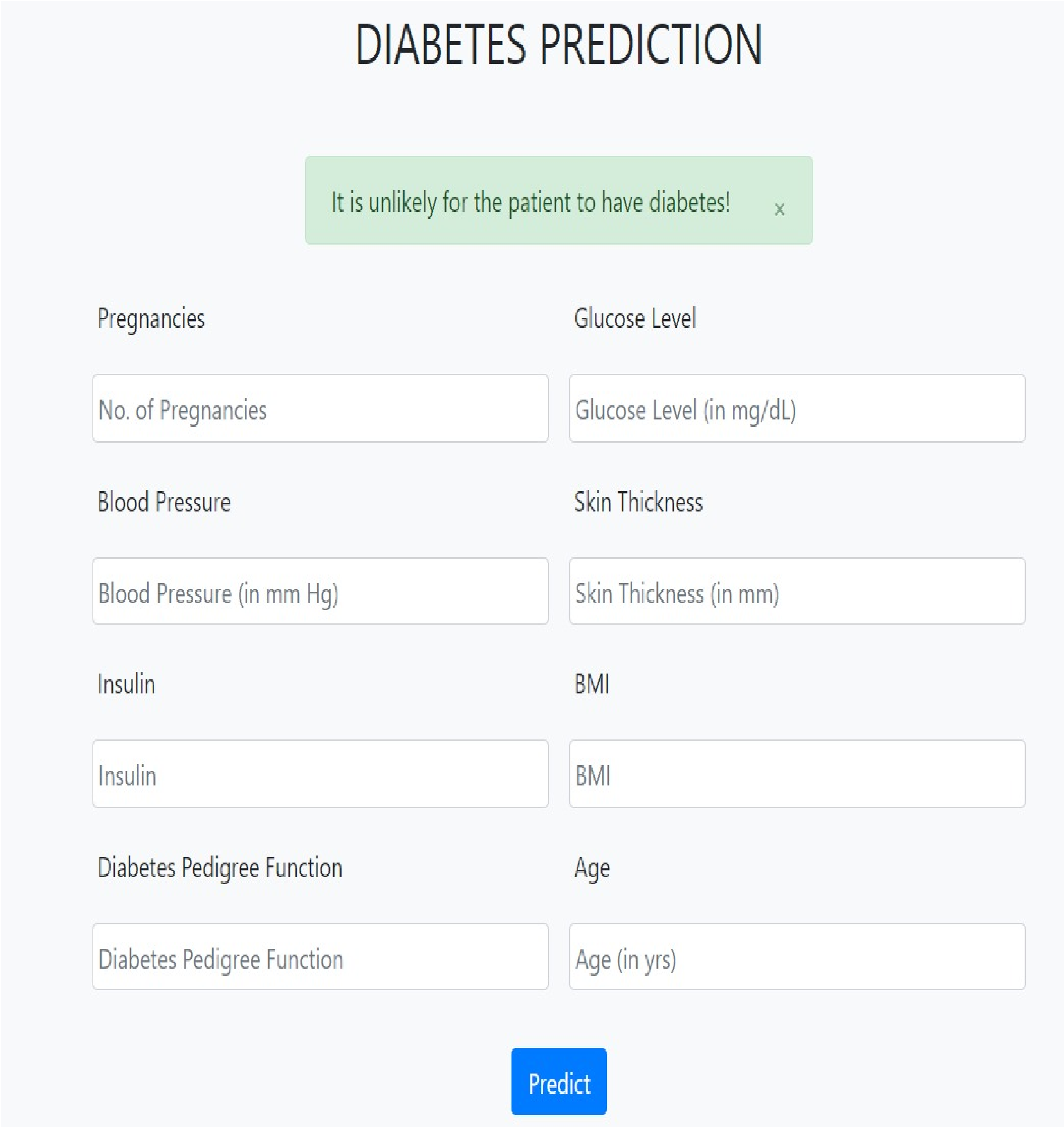

Figure 6.2: **Result For Normal Patient**

In Figure 6.2 the output of a diabetes prediction model using machine learning typically consists of the model's prediction regarding whether an individual is likely to have diabetes or not.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Diabetes prediction using machine learning techniques has emerged as a promising approach for early detection and proactive management of diabetes. Through the integration of advanced machine learning algorithms, predictive models can effectively analyze diverse datasets containing demographic information, medical history, lifestyle factors, and biomarkers to identify individuals at high risk of developing diabetes.

Several studies and reviews have highlighted the significance of machine learning in improving the accuracy and reliability of diabetes prediction models. By leveraging techniques such as logistic regression, decision trees, random forests, support vector machines, neural networks, and ensemble methods, researchers have been able to develop robust predictive models capable of accurately assessing diabetes risk.

7.2 Future Enhancements

Integrating diverse data sources such as electronic health records, genetic information, wearable device data, and dietary information can provide a more comprehensive understanding of individual health profiles and enhance predictive accuracy.

Exploring advanced feature engineering techniques and automated feature selection methods can help identify the most informative features for diabetes prediction, leading to more efficient and interpretable models.

Chapter 8

PLAGIARISM REPORT

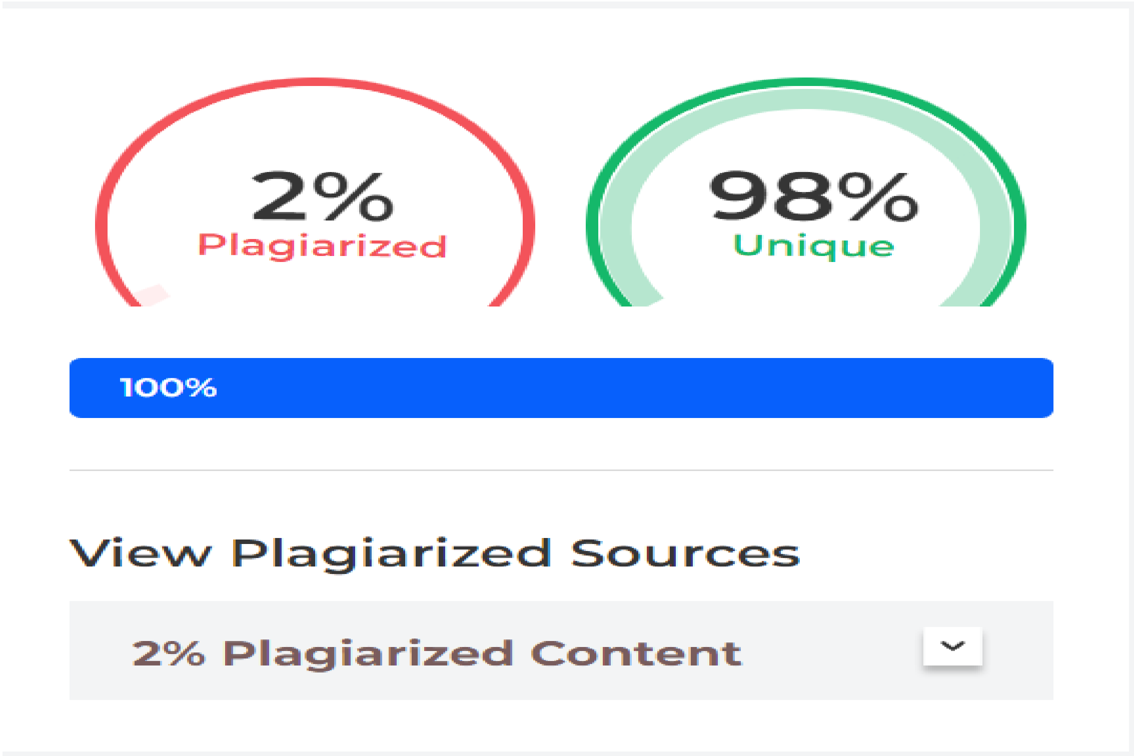

Figure 8.1: **Plagiarism report**

In Figure 8.1 these tools compare your content against a vast database of academic papers, articles, and web pages to identify any instances of potential plagiarism. Simply copy and paste your text into one of these tools to generate a detailed report. Remember to properly cite your sources to avoid plagiarism and give credit to the original authors.

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
from flask import Flask, request, render_template, flash

import pickle

app = Flask(_name_)
app.config['SECRET_KEY'] = 'supersecret'

scaler = pickle.load(open('scaler.pkl', 'rb'))
model = pickle.load(open('svm_model.pkl', 'rb'))

@app.route('/', methods=['GET', 'POST'])
def home():
    prediction = -1
    if request.method == 'POST':
        pregs = int(request.form.get('pregs'))
        gluc = int(request.form.get('gluc'))
        bp = int(request.form.get('bp'))
        skin = int(request.form.get('skin'))
        insulin = float(request.form.get('insulin'))
        bmi = float(request.form.get('bmi'))
        func = float(request.form.get('func'))
        age = int(request.form.get('age'))

        input_features = [[pregs, gluc, bp, skin, insulin, bmi, func, age]]
        # print(input_features)
        prediction = model.predict(scaler.transform(input_features))
        # print(prediction)

    return render_template('index.html', prediction=prediction)

if _name_ == '_main_':
    app.run(debug=True)
Automatically generated by Colaboratory.

# *DIABETES PREDICTION*
```

```python

---

Importing Dependencies
"""

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

"""Data Collection and Analysis

PIMA Diabetes Dataset
"""

#loading the dataset to a pandas df
df = pd.read_csv('diabetes.csv')

#printing the first 5 rows
df.head()

#no of rows and cols
df.shape

#getting the statistical measures of the df
df.describe()

#no of diabetics and non-diabetics
df['Outcome'].value_counts()

"""0 --> Non-Diabetic

1 --> Diabetic
"""


"""Data Cleaning

Drop duplicates
"""

print('Before dropping duplicates: ', df.shape)
df = df.drop_duplicates()
print('After dropping duplicates: ', df.shape)

"""Check for NULL values"""

df.isnull().sum()

```

```
"""Check for missing values"""

print('No of missing values in Glucose: ', df[df['Glucose'] == 0].shape[0])
print('No of missing values in BloodPressure: ', df[df['BloodPressure'] == 0].shape[0])
print('No of missing values in SkinThickness: ', df[df['SkinThickness'] == 0].shape[0])
print('No of missing values in Insulin: ', df[df['Insulin'] == 0].shape[0])
print('No of missing values in BMI: ', df[df['BMI'] == 0].shape[0])

"""Replace missing values with mean"""

df['Glucose'] = df['Glucose'].replace(0, df['Glucose'].mean())
df['BloodPressure'] = df['BloodPressure'].replace(0, df['BloodPressure'].mean())
df['SkinThickness'] = df['SkinThickness'].replace(0, df['SkinThickness'].mean())
df['Insulin'] = df['Insulin'].replace(0, df['Insulin'].mean())
df['BMI'] = df['BMI'].replace(0, df['BMI'].mean())

df.describe()

"""Data Visualisation

Count plot
"""

import matplotlib.pyplot as plt
f, ax = plt.subplots(1,2,figsize=(10,5))
df['Outcome'].value_counts().plot.pie(explode=[0,0.1],autopct='%1.1f%%', ax=ax[0],shadow=True)
ax[0].set_title('Outcome')
ax[0].set_ylabel('')

import seaborn as sns
sns.countplot('Outcome', data=df, ax=ax[1])
ax[1].set_title('Outcome')
N, P = df['Outcome'].value_counts()
print('Negative(0) ->', N)
print('Positive(1) ->', P)

plt.grid()
plt.show()

"""Dataset is not balanced

Histogram (data is balanced or skewed)
"""

df.hist(bins=10,figsize=(10,10))
plt.show()

"""Analysing relationships bw variables

Correlation analysis
```

```
"""

#get correlations of each feature in the dataset
corr_mat = df.corr()
top_corr_features = corr_mat.index
plt.figure(figsize=(10,10))
#plot heat map
g = sns.heatmap(df[top_corr_features].corr(), annot=True, cmap='RdYlGn')

"""Split data into X and y"""

#separating the independent and dependent variables
X = df.drop(columns='Outcome', axis=1)
y = df['Outcome']
print(X.head())
print(y.head())

"""Data Standardisation - Feature Scaling"""

scaler = StandardScaler()
scaler.fit(X)
standardised_data = scaler.transform(X)
print(standardised_data)

X = standardised_data
y = df.Outcome
print(X)
print(y)

"""Split data into training and testing data"""

#80% is train, 20% is test
#random state is used to ensure a specific split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=7)

print(X.shape, X_train.shape, X_test.shape)

"""Classification Models

1) Logistic Regression
"""

from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression(solver='liblinear', multi_class='ovr')
lr_model.fit(X_train, y_train)

"""2) K Neighbours Classifier"""

from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier()
```

```
knn_model.fit(X_train, y_train)

"""3) Naive Bayes Classifier"""

from sklearn.naive_bayes import GaussianNB
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

"""4) Support Vector Machine(SVM)"""

from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(X_train, y_train)

"""5) Decision tree"""

from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)

"""6) Random Forest"""

from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(criterion='entropy')
rf_model.fit(X_train, y_train)

"""Predicting & Evaluating the Models"""

#make the predictions using test data for all 6 models
lr_preds = lr_model.predict(X_test)

knn_preds = knn_model.predict(X_test)

nb_preds = nb_model.predict(X_test)

svm_preds = svm_model.predict(X_test)

dt_preds = dt_model.predict(X_test)

rf_preds = rf_model.predict(X_test)

#get the accuracy of the models
print('Accuracy score of Logistic Regression:', round(accuracy_score(y_test, lr_preds) * 100, 2))
print('Accuracy score of KNN:', round(accuracy_score(y_test, knn_preds) * 100, 2))
print('Accuracy score of Naive Bayes:', round(accuracy_score(y_test, nb_preds) * 100, 2))
print('Accuracy score of SVM:', round(accuracy_score(y_test, svm_preds) * 100, 2))
print('Accuracy score of Decision Tree:', round(accuracy_score(y_test, dt_preds) * 100, 2))
print('Accuracy score of Random Forest:', round(accuracy_score(y_test, rf_preds) * 100, 2))

"""Save the Model with the Highest Accuracy using pickle"""
```

```
import pickle
pickle.dump(svm_model, open('svm_model.pkl', 'wb')) #svm has the highest accuracy

pickle.dump(scaler, open('scaler.pkl', 'wb')) #save the std scaler too
```

9.2 Poster Presentation

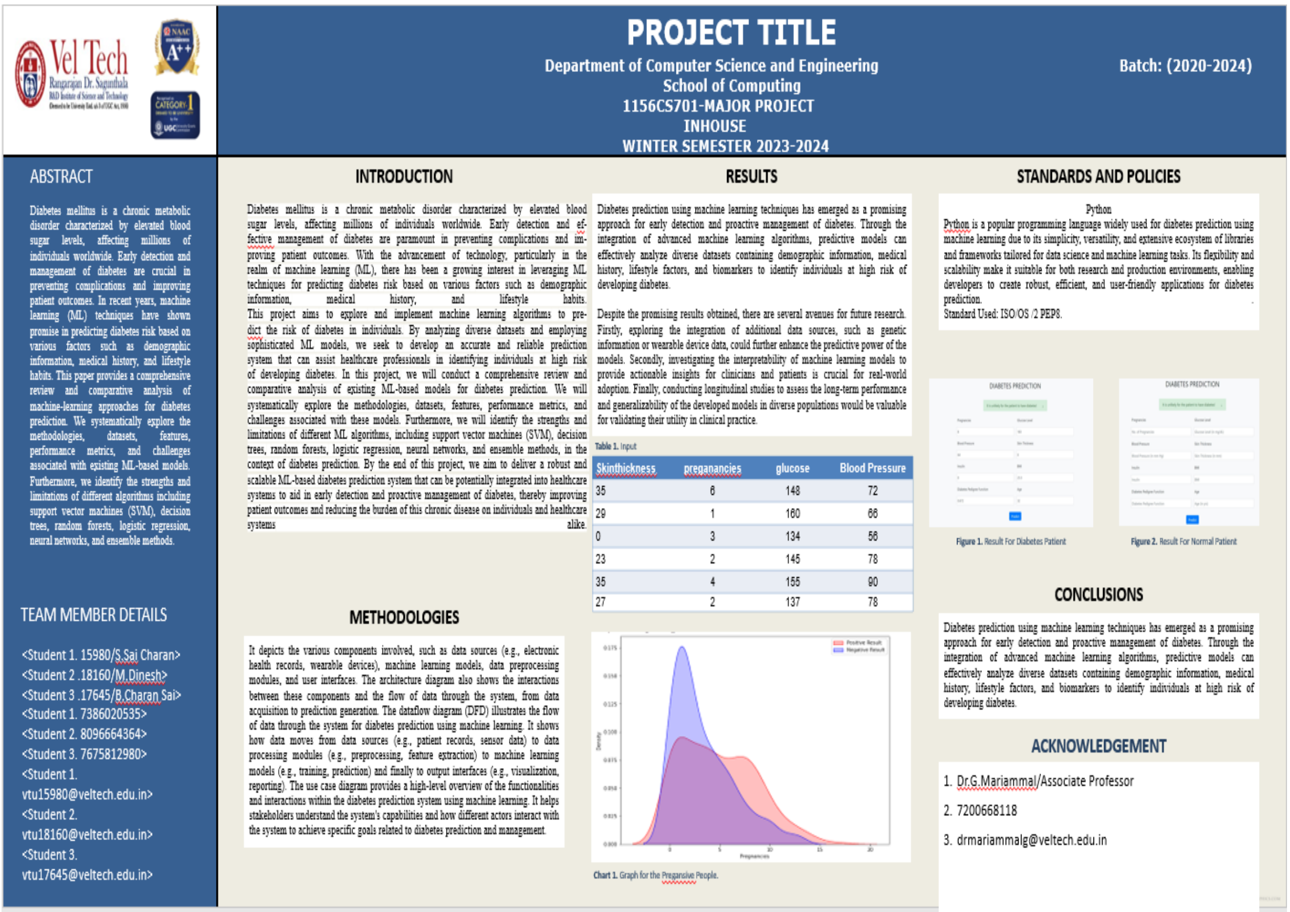

Figure 9.1: **Poster Representation**

References

[1] Luo, W., Phung, D. (2010). ”Diabetes prediction by using machine learning: A systematic review”. Expert Systems with Applications, 36(4), 7938-7946.

[2] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I. (2017). ”Machine Learning and Data Mining Methods in Diabetes Research”. Computational and Structural Biotechnology Journal, 15, 104-116.

[3] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B. (2014). ”Learning from Imbalanced Data Sets”. Springer International Publishing.

[4] Sathyanarayana, A., Joty, S., Fernandez-Luque, L. (2016). ”Tackling the problem of imbalanced datasets in deep learning”. arXiv preprint arXiv:1609.06570.

[5] Banerjee, M., Halder, A., Pal, S. (2017). ”Diabetes prediction using machine learning methods”. International Journal of Computer Applications, 167(5), 24-28.

[6] Al-Masri, E., Al-Ayyoub, M. (2018). ”Diabetes Prediction using Machine Learning Techniques”. Jordanian Journal of Computers and Information Technology, 4(3), 218-231.

[7] Anwar, S. M., Majid, M., Qayyum, A., Awais, M. (2020). ”Medical Image Analysis using Convolutional Neural Networks: A Review”. Journal of Medical Systems, 44(12), 1-15.

[8] Chaudhary, K., Pabalkar, C., Gogate, M. (2021). ”A Review of Data Mining Techniques used in Healthcare Data Analysis”. International Journal of Scientific Technology Research, 10(9), 230-235.

[9] Sharma, A., Dixit, V. (2022). ”A comprehensive review on diabetes prediction using machine learning techniques”. Journal of King Saud University-Computer and Information Sciences, 101525.

[10] Waseem, M., Niazi, M. A., Ullah, S. Artificial Intelligence and Evolutionary Computations in Engineering Systems, 63-81.

[11] A. Al-Mamun, M. A. Mottalib, M. N. Molla, M. A. Hossain, and M. A. Kadir (2017). ”Diabetes prediction using machine learning techniques,” International Conference on Electrical, Computer and Communication Engineering, Cox's Bazar, Bangladesh, pp. 40-45.

[12] S. K. Jha, A. K. Verma, and M. Gupta (2018). ”A comparative study of machine learning algorithms for diabetes prediction,” International Conference on Information Technology, Bhubaneswar, India, 2018, pp. 1-6.

[13] H. H. Mohamed, M. S. H. Allam, and H. H. F. Elbehery (2019) ”Diabetes prediction model using machine learning algorithms,” 2019 International Conference on Advanced Machine Learning Technologies and Applications, Cairo, Egypt, pp. 1-6.

[14] V. S. Akilan and A. Devi (2020). ”Prediction of diabetes using machine learning algorithms: A comparative study,” 2020 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, Chennai, India, pp. 1-6.

[15] S. K. Jain and S. Sharma (2021). ”Review on prediction of diabetes using machine learning techniques,” International Conference on Emerging Trends in Electrical, Electronics Communication Technologies, Greater Noida, India, pp. 1-5.

General Instructions

- Cover Page should be printed as per the color template and the next page also should be printed in color as per the template
- **Wherever Figures applicable in Report , that page should be printed in color**
- Dont include general content , write more technical content
- Each chapter should minimum contain 3 pages
- Draw the notation of diagrams properly
- Every paragraph should be started with one tab space
- Literature review should be properly cited and described with content related to project
- All the diagrams should be properly described and dont include general information of any diagram
- Example Use case diagram - describe according to your project flow
- All diagrams,figures should be numbered according to the chapter number and it should be cited properly
- **Testing and codequality should done in Sonarqube Tool**
- Test cases should be written with test input and test output
- All the references should be cited in the report
- **AI Generated text will not be considered**
- **Submission of Project Execution Files with Code in GitHub Repository**
- **Thickness of Cover and Rear Page of Project report should be 180 GSM**
- **Internship Offer letter and neccessary documents should be attached**
- **Strictly dont change font style or font size of the template, and dont customize the latex code of report**
- **Report should be prepared according to the template only**
- **Any deviations from the report template,will be summarily rejected**

- **Number of Project Soft Binded copy for each and every batch is (n+1) copies as given in the table below**
- For **Standards and Policies** refer the below link https://law.resource.org/pub/in/manifest.in.html
- Plagiarism should be less than 15%
- **Journal/Conference Publication proofs should be attached in the last page of Project report after the references section**

width=!,height=!,page=-