Phase 5

Documentation

Date	1 November 2023
Team id	Proj-212176-Team-2
Project Name	AI based Diabetes Prediction System
Maximum mark	

INDEX

List of figures and tables:

Abstract:

AI based diabetes prediction is used to view the details about the diabetic patient diabetic conditions using the dataset. This abstract provides an overview of the key aspects of AI-based diabetes prediction, including data collection, visualization, model development, and evaluation. It highlights the significance of early detection and personalized risk assessment in diabetes care, showcasing the potential for AI to transform healthcare by enabling proactive interventions and reducing the burden of diabetes-related complications. This feature is very much useful in treating the diabetic patients by seeing the data given by the AI analysis. This study explores the development and evaluation of an AI-based diabetes prediction system. Artificial intelligence (AI) has emerged as a potent tool for harnessing the power of data and technology to improve diabetes prediction and management.

Introduction:

This project explores the development and evaluation of the AI based diabetic prediction system. Initially the project starts with the problem definition and design thinking which gives information about the likelihood of the patient detail. The next process is importing the dataset and perform data cleaning and analysis. These are used in data preprocessing and then data visualization that helps in visualizing the data. Finally in model development and model evaluation the data are split and visualized in tree structure. the integration of AI-based prediction systems into healthcare practices represents a critical step towards improving patient outcomes, reducing healthcare costs, and addressing the global diabetes epidemic.

Literature Survey:

- A literature survey on AI-based diabetic prediction provides an overview of the research and developments in the field of using artificial intelligence for predicting diabetes. It helps to understand the existing approaches, challenges, and future directions in this area.
- AI systems are being used for early detection of diabetes risk factors and complications, such as diabetic retinopathy and neuropathy. Early intervention and preventative measures can significantly improve patient outcomes.
- Challenges in AI-based diabetic prediction include data privacy concerns, data quality, and model generalization. Additionally, addressing class imbalance in datasets and handling unstructured data like free-text clinical notes are ongoing challenges.
- AI-based diabetic prediction models are increasingly finding their way into clinical practice. They assist healthcare professionals in identifying at-risk patients and tailoring treatment plans. They are also used in telemedicine, remote monitoring, and mobile health applications.
- A literature survey on AI-based diabetic prediction reveals a dynamic and rapidly evolving field with significant potential to improve the diagnosis, treatment, and management of diabetes. Researchers and healthcare practitioners are continually exploring innovative AI solutions to address the challenges posed by this chronic condition and to promote patient-centric care.

Problem definition:

1. Developing an AI model to predict the likelihood of an individual developing diabetes based on their medical history and lifestyle factor.

2. Creating an AI system that can accurately classify patients as diabetic or non-diabetic using their blood sugar level, BMI and other relevant health indicators.

3. Designing an AI-powered tool that can provide early detection of diabetes by analysing patterns in a person's glucose levels over time.

4. Building an AI model that can predict the risk of diabetes complications such as kidney disease or retinopathy, based on a patient's medical records and lifestyle data.

5. Developing an Ai system that can provide personalized recommendations for managing and preventing diabetes based on an individual's specific risk factors and health goals.

Design thinking:

Empathy :

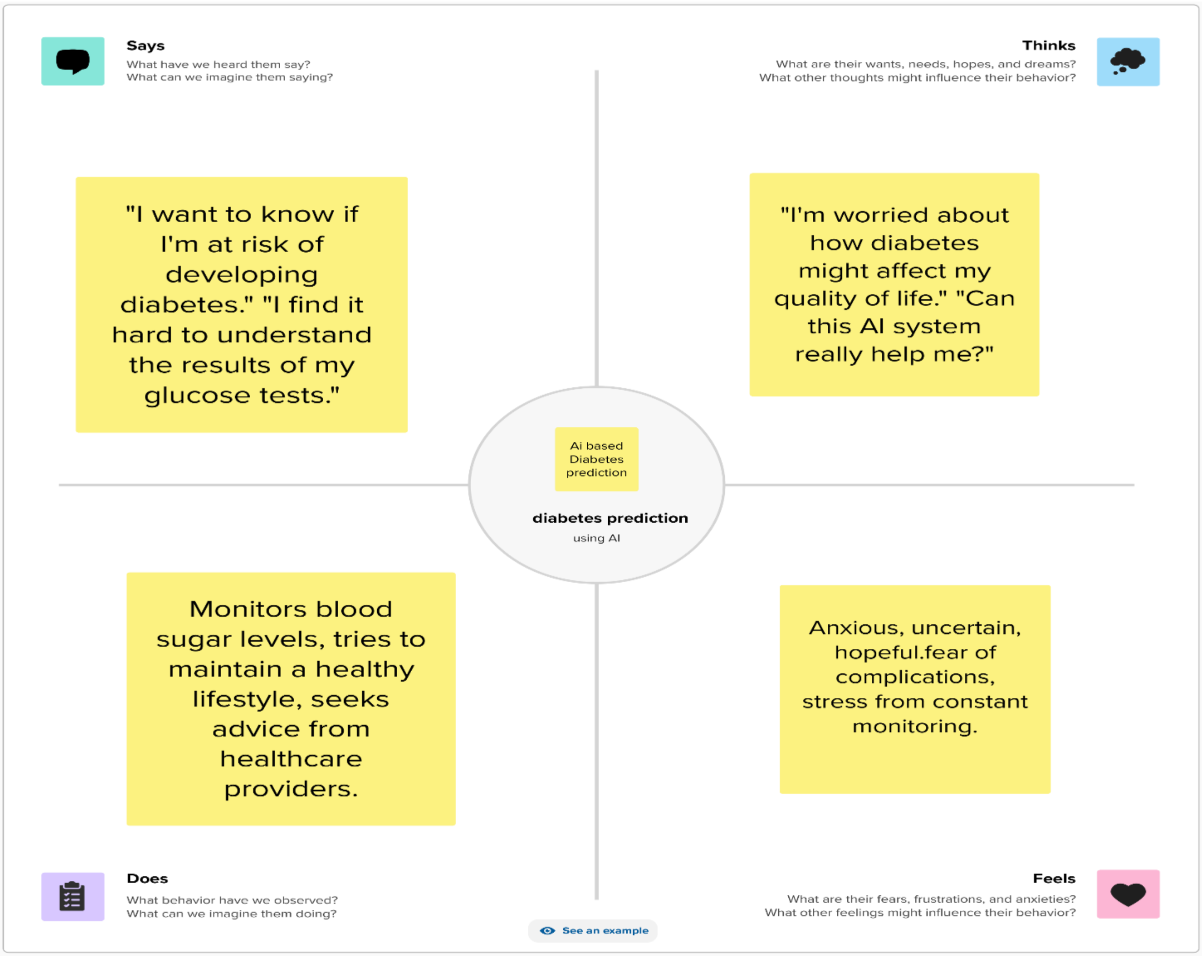
Diabetes Patients: Those who have been diagnosed with diabetes or are at risk of developing it.

Healthcare Providers: Doctors, nurses, dietitians, and other professionals involved in diabetes care.

Caregivers: Family members or friends who support diabetes patient.

Regulators: Those responsible for ensuring that the AI system complies with healthcare regulations.

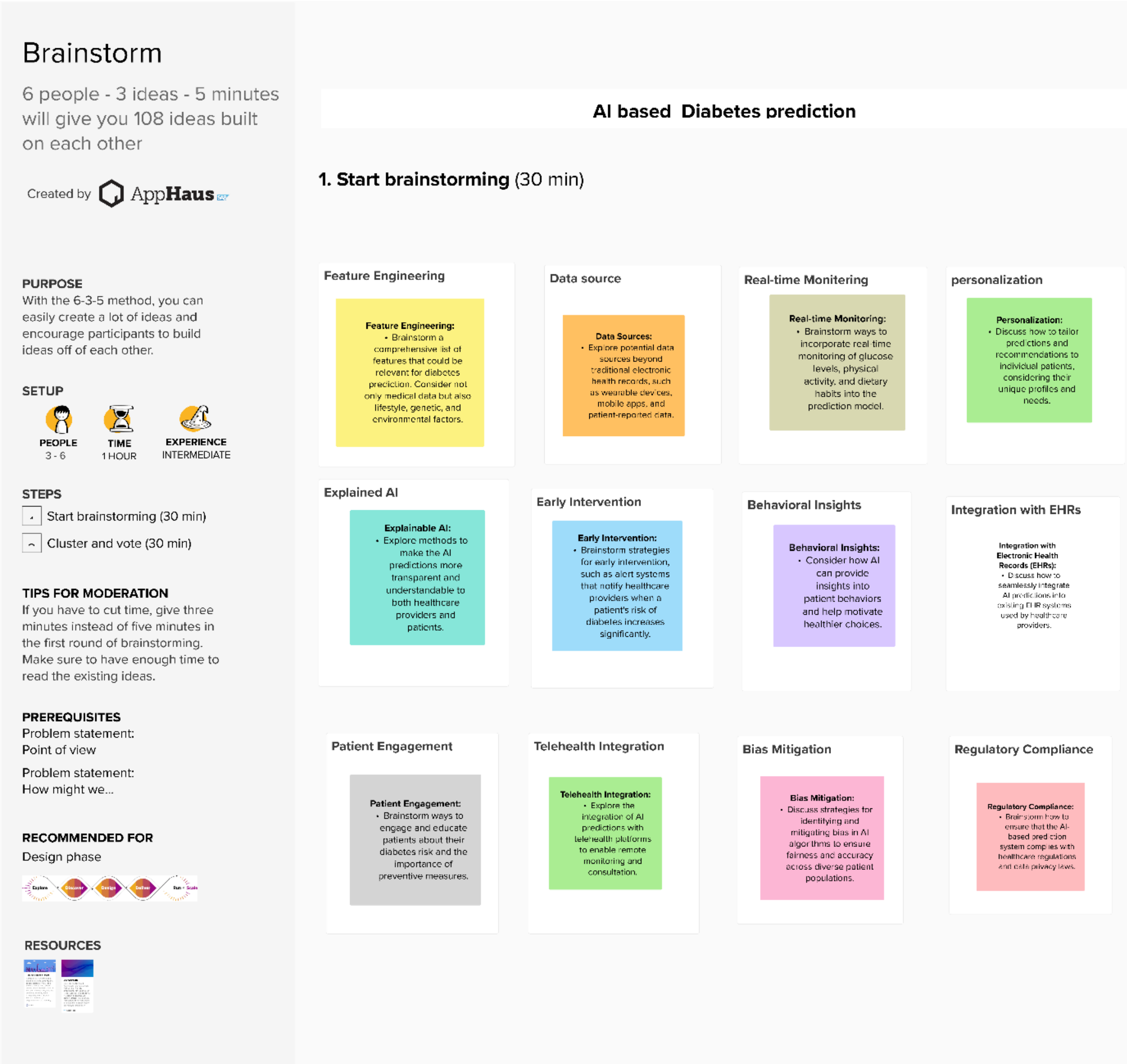
Researchers: Individuals or teams conducting research on diabetes prediction and management



Figure(1)

Brainstorm:

Brainstorming is a creative process that can help generate innovative ideas and solutions for AI-based diabetes prediction. When brainstorming for this context, it's important to involve a diverse group of experts, including data scientists, healthcare professionals, and domain specialists. Here are some brainstorming ideas to get you started.



Figure(2)

Innovation and Problem solving

Problem Statement:

Early Detection of Diabetes Risk	Problem statement develop an AI based predictive model to identify individual at an early stage of diabetes risk, allowing for timely intervention and prevention strategies.
Personalized Diabetes Risk Assessment	Create an AI algorithm that provides personalized diabetes risk assessments by considering a patients.
Real time glucose Level prediction	Build an algorithm that predicts real-time glucose levels
Integrating wearable data	Integrate data from wearable devices such as continuous glucose monitors to enhance diabetes risk prediction and management
Reducing false positives	Develop AI algorithms that minimize false positive predictions of diabetes risk, are directed towards those who truly need them.
Ethical use of patient data	Address ethical concerns and ensure the responsible use of patient data in AI-based diabetes prediction, respecting privacy and confidentiality

Data Cleaning and Analysis:

Data cleaning and analysis comes under data preprocessing. Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining tasks

Data Cleaning:

• Handle missing values: Identify and handle missing data. You can either impute missing values or remove rows/columns with missing data depending on the extent of missingness.

• Outlier detection and treatment: Identify and deal with outliers in your data. Outliers can negatively impact model performance.

Data Analysis:

Data analysis is a crucial step in developing an AI-based diabetes detection model. Through data analysis, you can gain insights into the dataset, understand the relationships between features, and make informed decisions about feature selection, preprocessing, and model development. Below are some key steps and code examples for data analysis in Python using popular libraries like Pandas, NumPy, and Matplotlib.

Data visualization:

Data visualization is an important step in understanding your dataset when working on an AI-based diabetes detection project. You can use libraries like Matplotlib and Seaborn in Python to create various types of visualizations.

The following codes will provides examples of various data visualization techniques:

1. Displaying the first few rows of the dataset to get an overview.
2. Generating summary statistics for numerical features.
3. Creating histograms to visualize the distribution of numerical features.
4. Generating boxplots to identify potential outliers.
5. Creating a pair plot to visualize relationships between features, with hue indicating the outcome class.
6. Creating a correlation heatmap to visualize feature correlations.

Model development and evaluation:

Module development in AI typically refers to the creation and organization of code, functions, or components that serve specific purposes within an artificial intelligence system. These modules are designed to perform well-defined tasks, such as data preprocessing, feature extraction, model training, evaluation, or deployment. They are created to enhance code modularity, reusability, and maintainability, making AI projects more organized and manageable.

Some of the key aspects of modularity are,

Modularity: AI module development involves breaking down complex AI systems into smaller, manageable modules or components. Each module is responsible for a specific part of the AI workflow.

Reusability: Modules are designed to be reusable in different parts of the project or even in other AI projects. This encourages efficient code reuse and reduces redundancy.

Encapsulation: Modules encapsulate specific functionality, and their internal details may not be visible to other parts of the system. This concept aligns with the principles of object-oriented programming and helps control complexity.

Abstraction: Modules are often designed with a clear and high-level interface, abstracting away the internal complexities. This makes it easier for other developers to use the modules without needing to understand the internal workings.

Testing and Validation: Modules can be tested in isolation, making it easier to identify and fix issues in smaller, self-contained components. This can lead to improved overall system reliability.

Collaboration: In team-based AI development, different team members may be responsible for developing various modules. Properly designed modules enable effective collaboration among team members with different expertise.

Code samples:

#import packages

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler , Normalizer
from sklearn.compose import make_column_transformer, make_column_select
or from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

#import dataset

```
dataset = pd.read_csv('C:/Users/91638/Documents/diabetes.csv')
```

#Data cleaning and analysis

```
dataset.head()

dataset.info()

X = dataset.copy()
y = X.pop('Outcome')

dataset.rename(columns={'DiabetesPedigreeFunction': 'DPF'}, inplace= True)
to_nan = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin']
```

```
to_nan.append(['BMI', 'DPF', 'Age'])
for i in range(len(to_nan)):
    dataset[to_nan[i]] = dataset[to_nan[i]].replace(0, np.nan)
dataset.head(10)
```

#data visualization

```
dataset.plot()

from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
cm = confusion_matrix(y__predict, y__real)
from mlxtend.plotting import plot_confusion_matrix
fig, ax = plot_confusion_matrix(conf_mat=cm)
plt.show()

corrmat=dataset.corr()
sns.heatmap(corrmat, annot=True)
```

#model development and evaluation

```
X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model = RandomForestClassifier(random_state=42)
```

```
model.fit(X_train, y_train)

dataset_true = dataset[(dataset.Outcome>0)]
dataset_true.describe().T

dataset_false = dataset[(dataset.Outcome<1)]
dataset_false.describe().T


y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print("Classification Report:")
print(report)
```

Output:

#data cleaning and analysis

```
In [1]: import numpy as np
        import pandas as pd
        from sklearn.preprocessing import StandardScaler , Normalizer
        from sklearn.compose import make_column_transformer, make_column_selector
        from sklearn.model_selection import train_test_split
```

```
In [3]: dataset = pd.read_csv('C:/Users/91638/Documents/diabetes.csv')
        dataset.head()
```

Out[3]:

	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

```
In [ ]:
```

Figure (3)

```
In [4]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Figure (4)

```
In [8]: X = dataset.copy()
        y = X.pop('Outcome')
```

```
In [9]: dataset.head()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
6	3	78.0	50.0	32.0	88.0	31.0	0.248	26	1
8	2	197.0	70.0	45.0	543.0	30.5	0.158	53	1
13	1	189.0	60.0	23.0	846.0	30.1	0.398	59	1

```
In [ ]:
```

Figure (5)

Out[21]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
6	3	78.0	50.0	32.0	88.0	31.0	0.248	26	1
8	2	197.0	70.0	45.0	543.0	30.5	0.158	53	1
13	1	189.0	60.0	23.0	846.0	30.1	0.398	59	1
14	5	166.0	72.0	19.0	175.0	25.8	0.587	51	1
16	0	118.0	84.0	47.0	230.0	45.8	0.551	31	1
18	1	103.0	30.0	38.0	83.0	43.3	0.183	33	0
19	1	115.0	70.0	30.0	96.0	34.6	0.529	32	1
20	3	126.0	88.0	41.0	235.0	39.3	0.704	27	0

Figure (6)

In [22]:
```
dataset_true = dataset[(dataset.Outcome>0)]
dataset_true.describe().T
```

Out[22]:

	count	mean	std	min	25%	50%	75%	max
Pregnancies	130.0	4.469231	3.916153	0.000	1.00000	3.000	7.0000	17.00
Glucose	130.0	145.192308	29.839388	78.000	124.25000	144.500	171.7500	198.00
BloodPressure	130.0	74.076923	13.021518	30.000	66.50000	74.000	82.0000	110.00
SkinThickness	130.0	32.961538	9.642770	7.000	26.00000	33.000	39.7500	63.00
Insulin	130.0	206.846154	132.699898	14.000	127.50000	169.500	239.2500	846.00
BMI	130.0	35.777692	6.734687	22.900	31.60000	34.600	38.3500	67.10
DPF	130.0	0.625585	0.405910	0.127	0.32975	0.546	0.7865	2.42
Age	130.0	35.938462	10.634705	21.000	27.25000	33.000	43.0000	60.00
Outcome	130.0	1.000000	0.000000	1.000	1.00000	1.000	1.0000	1.00

In [ ]:

Figure (7)

In [24]:
```
dataset_false = dataset[(dataset.Outcome<1)]
dataset_false.describe().T
```

Out[24]:

	count	mean	std	min	25%	50%	75%	max
Pregnancies	262.0	2.721374	2.617844	0.000	1.000	2.0000	4.00000	13.000
Glucose	262.0	111.431298	24.642133	56.000	94.000	107.5000	126.00000	197.000
BloodPressure	262.0	68.969466	11.892841	24.000	60.000	70.0000	76.00000	106.000
SkinThickness	262.0	27.251908	10.434135	7.000	18.250	27.0000	34.00000	60.000
Insulin	262.0	130.854962	102.626177	15.000	66.000	105.0000	163.75000	744.000
BMI	262.0	31.750763	6.794971	18.200	26.125	31.2500	36.10000	57.300
DPF	262.0	0.472168	0.299240	0.085	0.261	0.4135	0.62425	2.329
Age	262.0	28.347328	8.989008	21.000	22.000	25.0000	30.00000	81.000
Outcome	262.0	0.000000	0.000000	0.000	0.000	0.0000	0.00000	0.000

Figure (8)

#data visualization

```
In [14]: dataset.plot()
Out[14]: <Axes: >
```

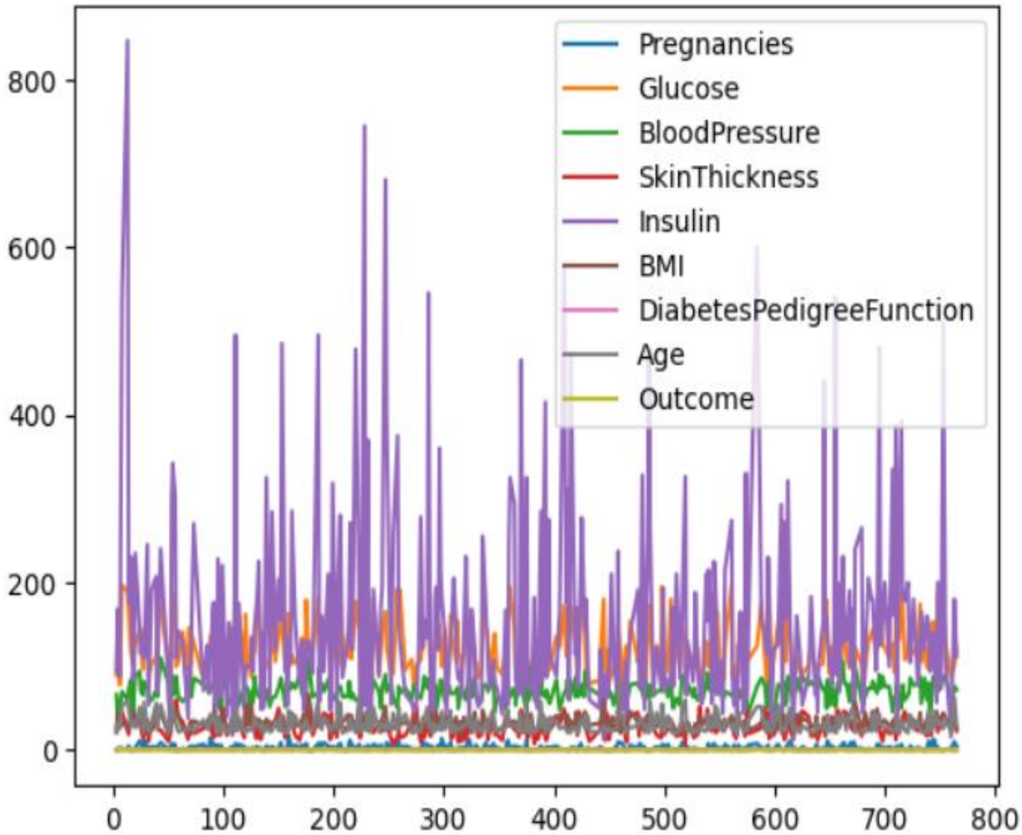


Figure (9)

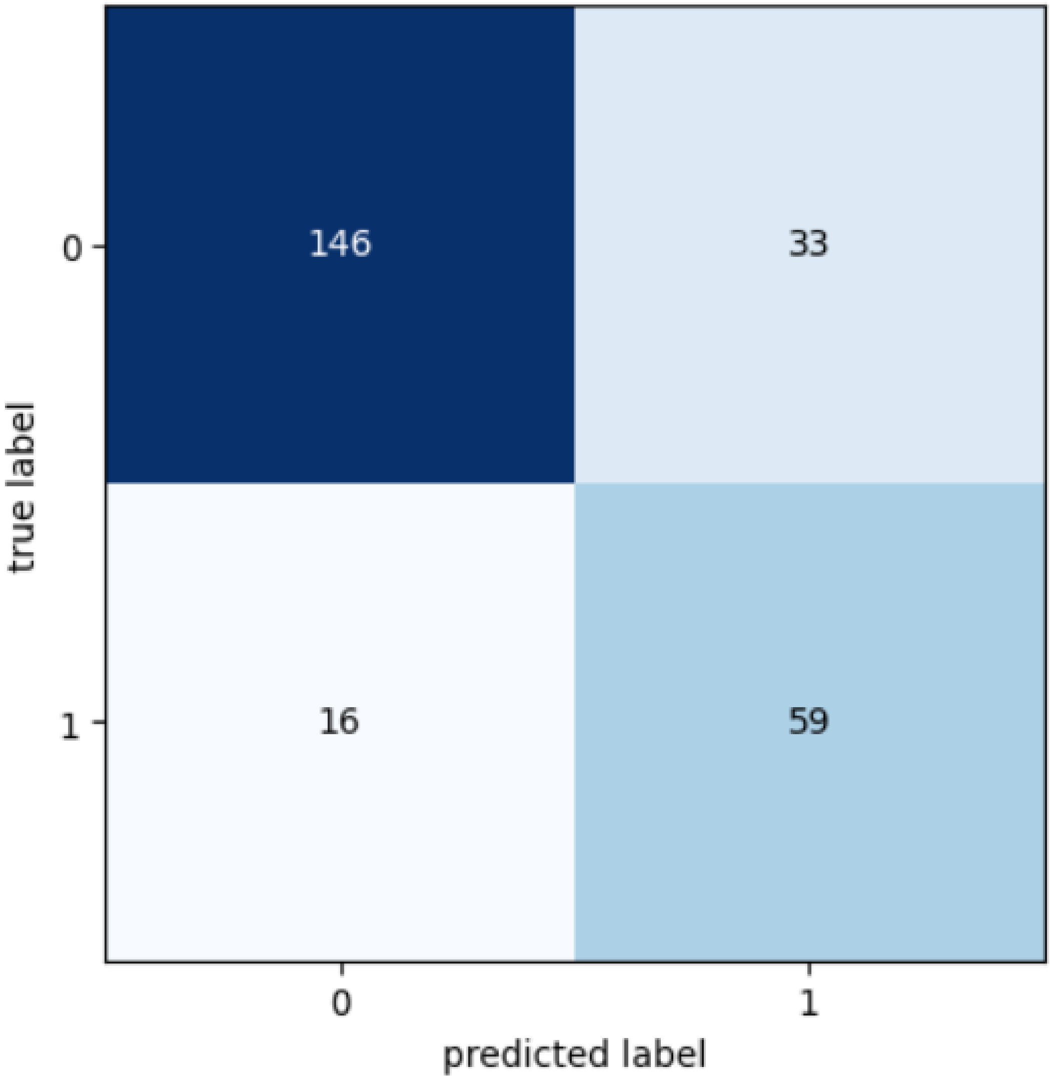


Figure (10)

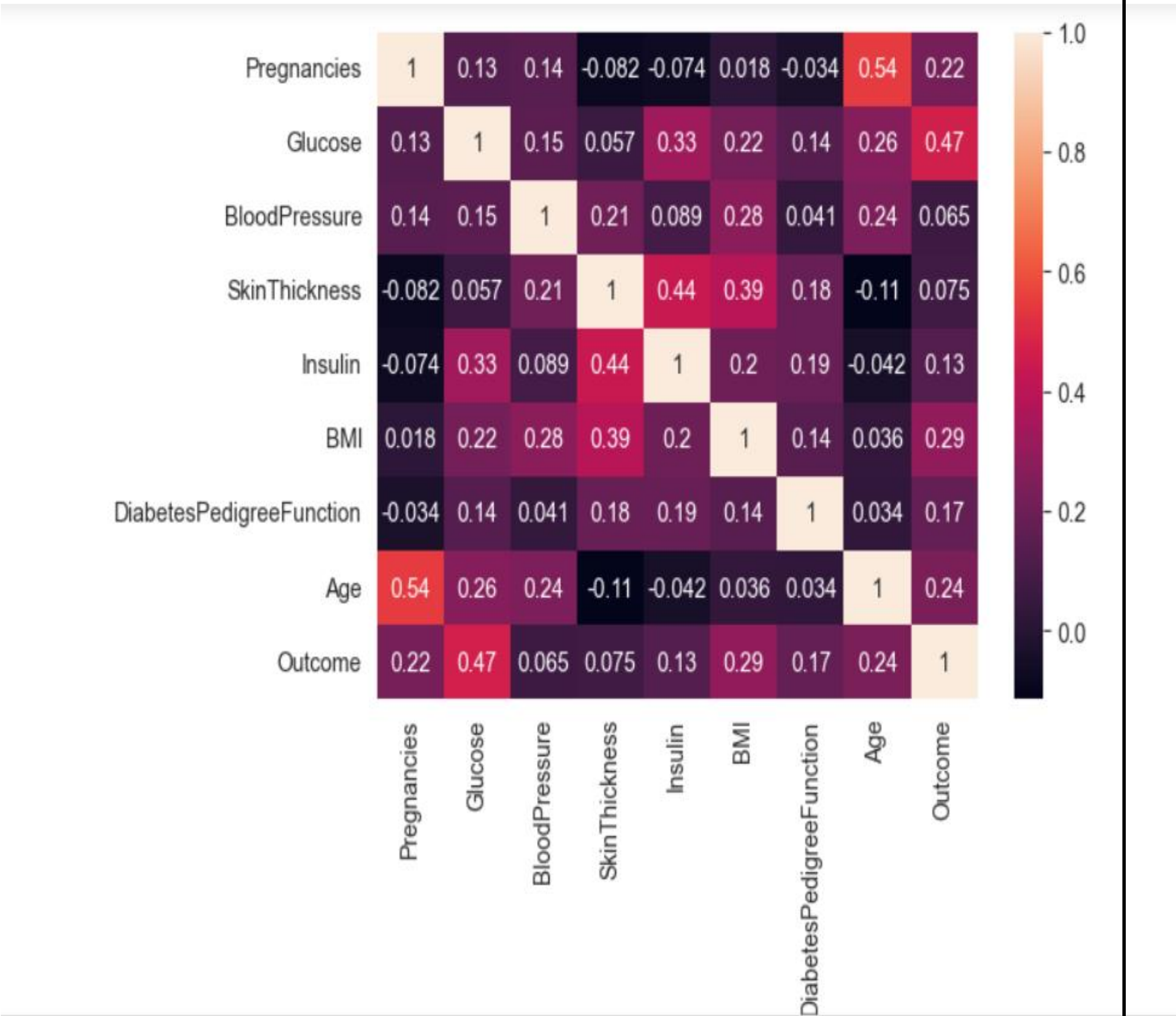

Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1 0.13 0.14 -0.082 -0.074 0.018 -0.034 0.54 0.22
0.13 1 0.15 0.057 0.33 0.22 0.14 0.26 0.47
0.14 0.15 1 0.21 0.089 0.28 0.041 0.24 0.065
-0.082 0.057 0.21 1 0.44 0.39 0.18 -0.11 0.075
-0.074 0.33 0.089 0.44 1 0.2 0.19 -0.042 0.13
0.018 0.22 0.28 0.39 0.2 1 0.14 0.036 0.29
-0.034 0.14 0.041 0.18 0.19 0.14 1 0.034 0.17
0.54 0.26 0.24 -0.11 -0.042 0.036 0.034 1 0.24
0.22 0.47 0.065 0.075 0.13 0.29 0.17 0.24 1
1.0
0.8
0.6
0.4
0.2
0.0

Figure (11)

#model development and evaluation

```
In [7]: accuracy = accuracy_score(y_test, y_pred)
        report = classification_report(y_test, y_pred)
```

```
In [8]: print(f"Accuracy: {accuracy:.2f}")
        print("Classification Report:")
        print(report)
```

```
Accuracy: 0.72
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.78      0.78        99
           1       0.61      0.62      0.61        55

    accuracy                           0.72       154
   macro avg       0.70      0.70      0.70       154
weighted avg       0.72      0.72      0.72       154
```

```
In [ ]:
```

Figure (12)

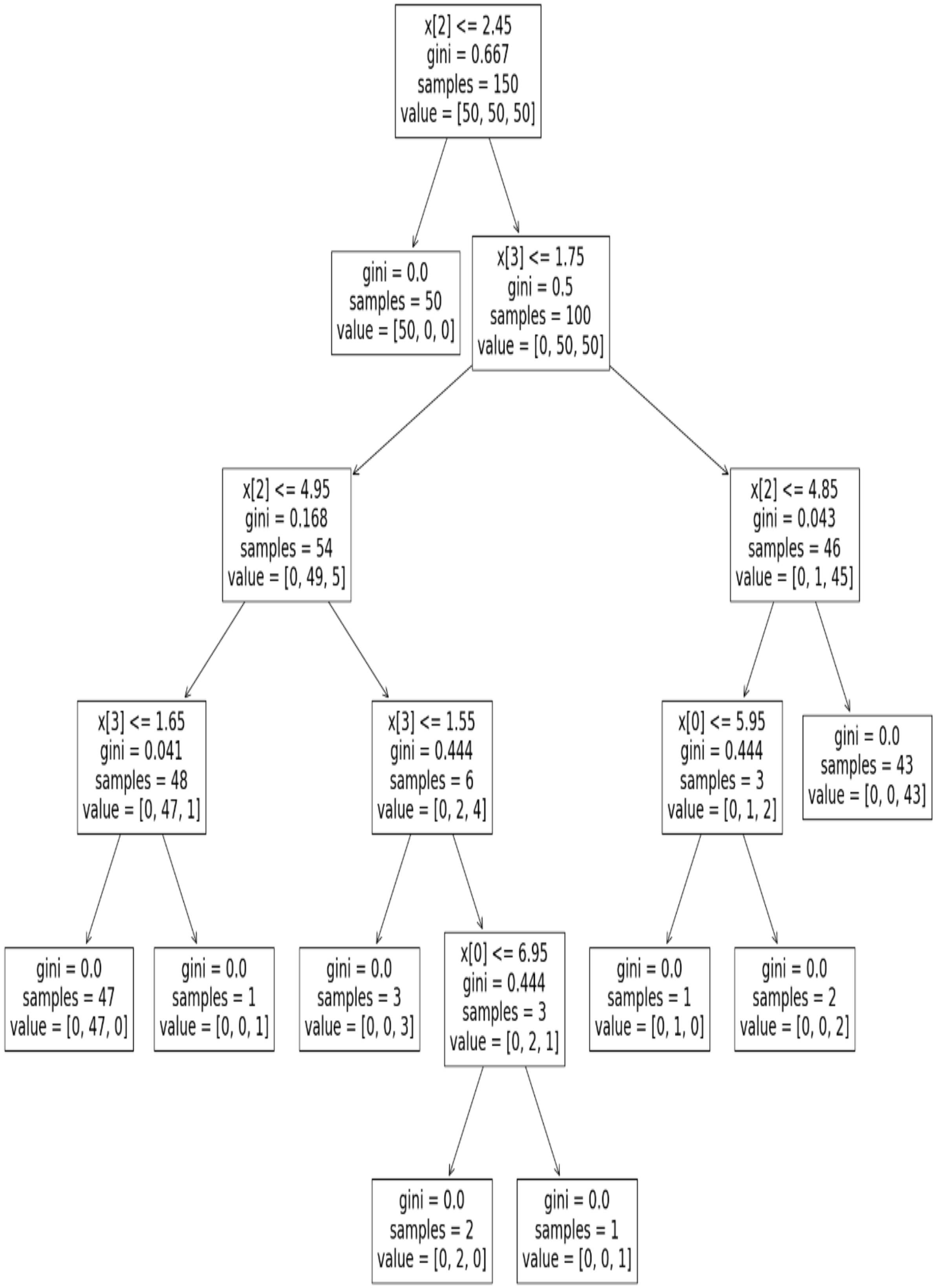

x[2] <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
gini = 0.0
samples = 50
value = [50, 0, 0]
x[3] <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
x[2] <= 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
x[2] <= 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
x[3] <= 1.65
gini = 0.041
samples = 48
value = [0, 47, 1]
x[3] <= 1.55
gini = 0.444
samples = 6
value = [0, 2, 4]
x[0] <= 5.95
gini = 0.444
samples = 3
value = [0, 1, 2]
gini = 0.0
samples = 43
value = [0, 0, 43]
gini = 0.0
samples = 47
value = [0, 47, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 3
value = [0, 0, 3]
x[0] <= 6.95
gini = 0.444
samples = 3
value = [0, 2, 1]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 2
value = [0, 0, 2]
gini = 0.0
samples = 2
value = [0, 2, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]

Figure (13)

Conclusion:

AI-based diabetes prediction is a transformative force in the healthcare landscape, offering innovative solutions to the complex challenges posed by diabetes. As this field continues to evolve, it holds the potential to empower individuals to take proactive control of their health and to enable healthcare professionals to provide more precise and personalized care. By leveraging the capabilities of AI, we move closer to a future where diabetes is managed with greater efficiency, effectiveness, and compassion.

Reference:

R. Punn, A. Agarwal, and R. K. Ahuja, "Deep learning and medical image processing for diabetic retinopathy: a study," Journal of King Saud University - Computer and Information Sciences, 2020. - This study explores the use of deep learning in the context of diabetic retinopathy detection.