Project Report

Diabetes prediction using Machine Learning

1. INTRODUCTION

1.1 Project Overview

The Diabetes Prediction project is geared towards crafting a precision-driven system capable of accurately assessing the probability of an individual developing diabetes. Central to the project is an in- depth analysis of a diverse health dataset, encompassing critical parameters such as blood sugar levels, BMI, age, and familial medical history. Leveraging the power of machine learning algorithms, the project focuses on training a robust predictive model that can distil meaningful patterns and associations within the dataset. By delving into advanced techniques such as supervised learning, the model is equipped to make informed predictions when presented with novel input parameters. Throughout this process, a commitment to data integrity, model interpretability, and ethical considerations remains paramount, ensuring the reliability and ethical deployment of the predictive system. Ultimately, the Diabetes Prediction project aspires to revolutionize early diabetes detection, offering a valuable tool for both healthcare professionals and individuals seeking proactive health management.

1.2 Purpose

This project serves a critical purpose in combating the escalating global prevalence of diabetes by developing a dependable and precise prediction system. By furnishing healthcare professionals with this tool, the project aims to facilitate early identification of individuals at risk of developing diabetes. This early recognition paves the way for timely implementation of preventive measures and personalized healthcare interventions, significantly enhancing the overall management of diabetes and its associated health risks on a global scale.

2. LITERATURE SURVEY

2.1 Existing problem

A review of existing literature reveals that diabetes has become a global health concern, affecting millions of people worldwide and Discuss the challenges associated with diabetes prediction, such as the reliance on manual diagnostic methods, limited accessibility to healthcare facilities for screening, and the need for early detection to improve patient outcomes. Highlight the shortcomings of existing prediction models and emphasize the significance of developing an accurate and reliable diabetes prediction system.

2.2 References

Gill, N.S.; Mittal, P. A computational hybrid model with two level classification using SVM and neural network for predicting the diabetes disease. *J. Theor. Appl. Inf. Technol.* **2016**, *87*, 1–10.

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

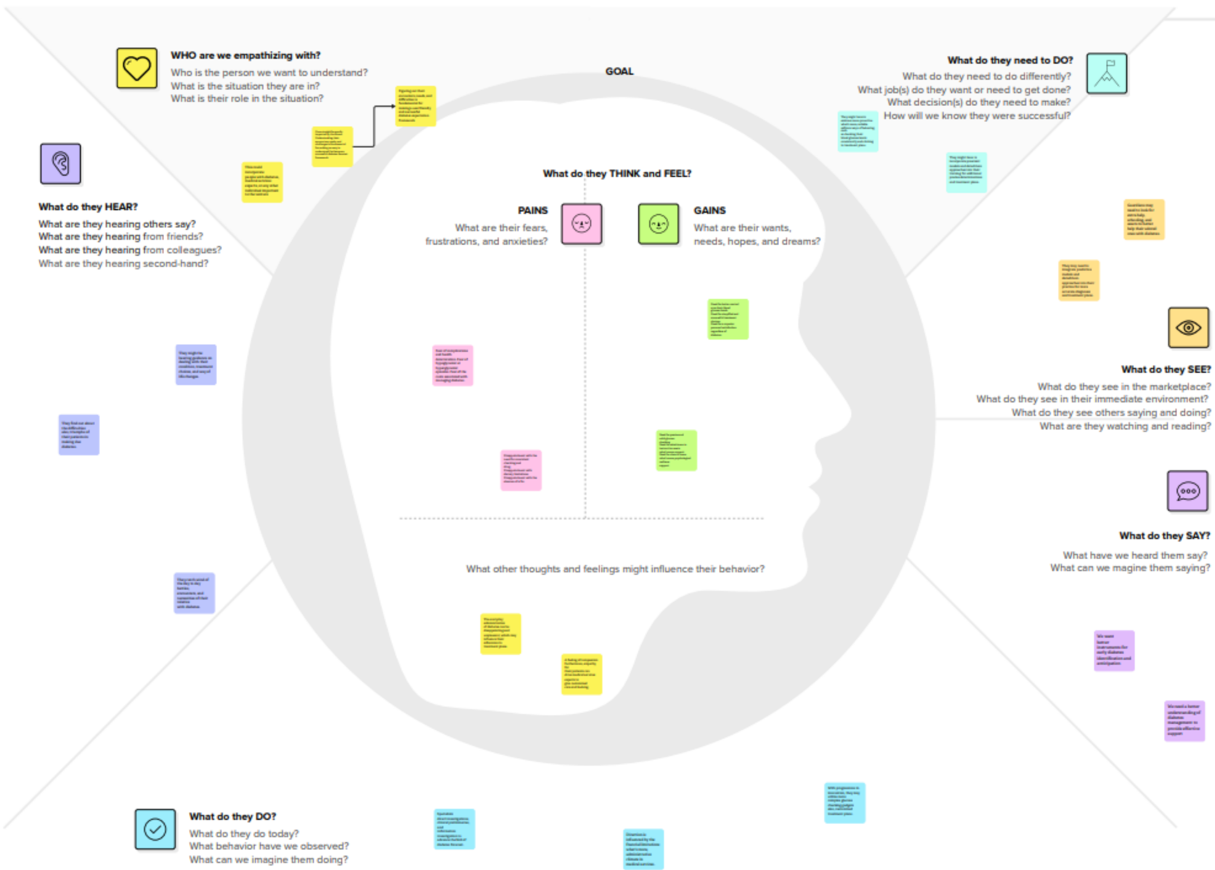
Nai-Arun N., Moungmai R. Comparison of Classifiers for the Risk of Diabetes Prediction Procedia Computer Science, 69 (2015), pp. 132-142

2.3 Problem Statement Definition

The problem at hand involves the widespread lack of early detection and preventive measures for individuals at high risk of developing diabetes. This issue extends across diverse demographics, impacting the well-being of affected individuals and communities. The urgency lies in addressing this problem promptly to avert potential health complications, reduce healthcare costs, and enhance overall quality of life. Failure to solve this problem could lead to an increased incidence of diabetes-related complications, exacerbating healthcare burdens and compromising the well-being of those at risk.

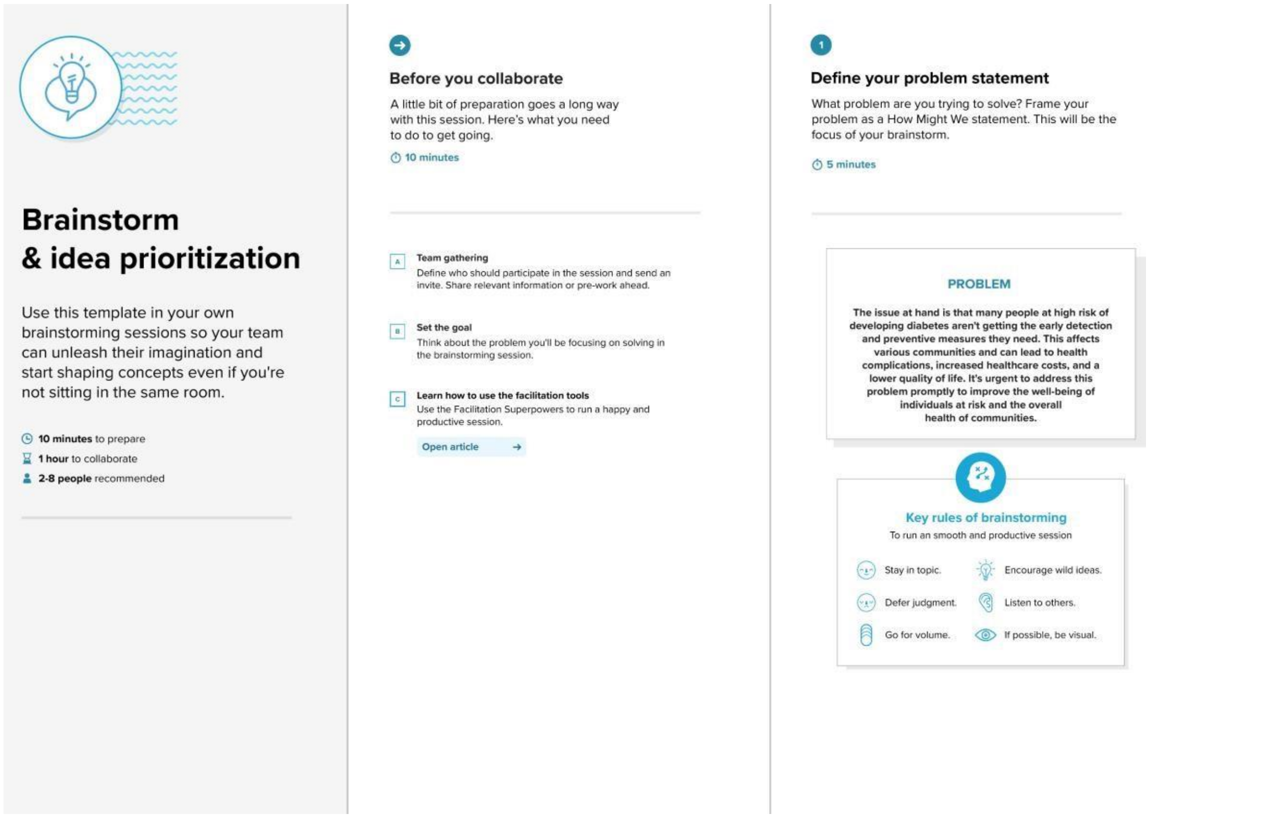
3. IDEATION & PROPOSED SOLUTION

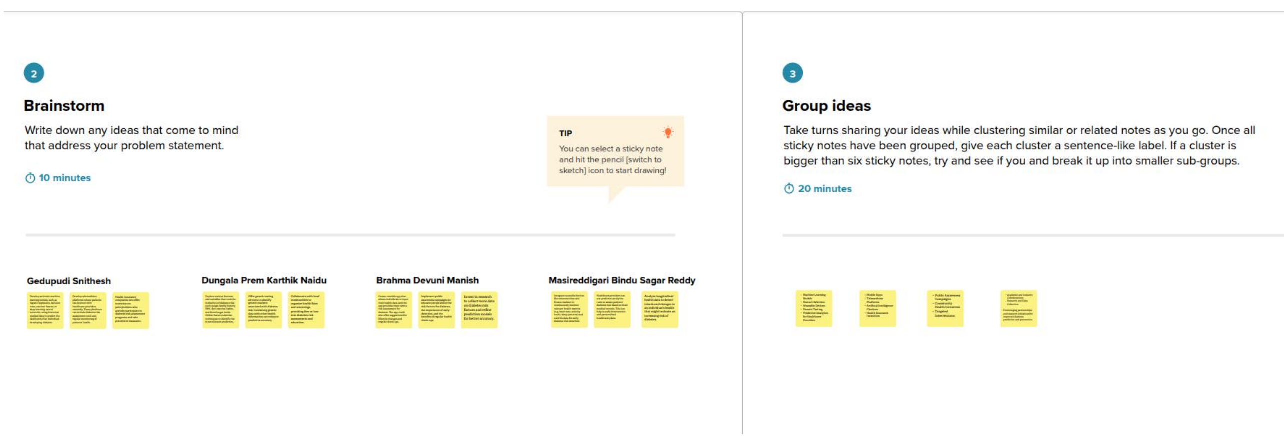
3.1 Empathy Map Canvas



WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we magine them saying?
What other thoughts and feelings might influence their behavior?
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

3.2 Ideation & Brainstorming





4. REQUIREMENT ANALYSIS

4.1 Functional requirement

The functional requirements for the diabetes prediction project include:

- Data collection and preprocessing: Collecting relevant health-related data and preprocessing it to remove any inconsistencies or missing values.

- Model development: Developing a predictive model using machine learning algorithms such as logistic regression, decision trees, or support vector machines.
- Model evaluation: Evaluating the performance of the developed model using appropriate evaluation metrics such as accuracy, sensitivity, specificity, and area under the ROC curve.
- User interface: Designing a user-friendly interface that allows users to input their health-related features and obtain the predicted likelihood of developing diabetes.

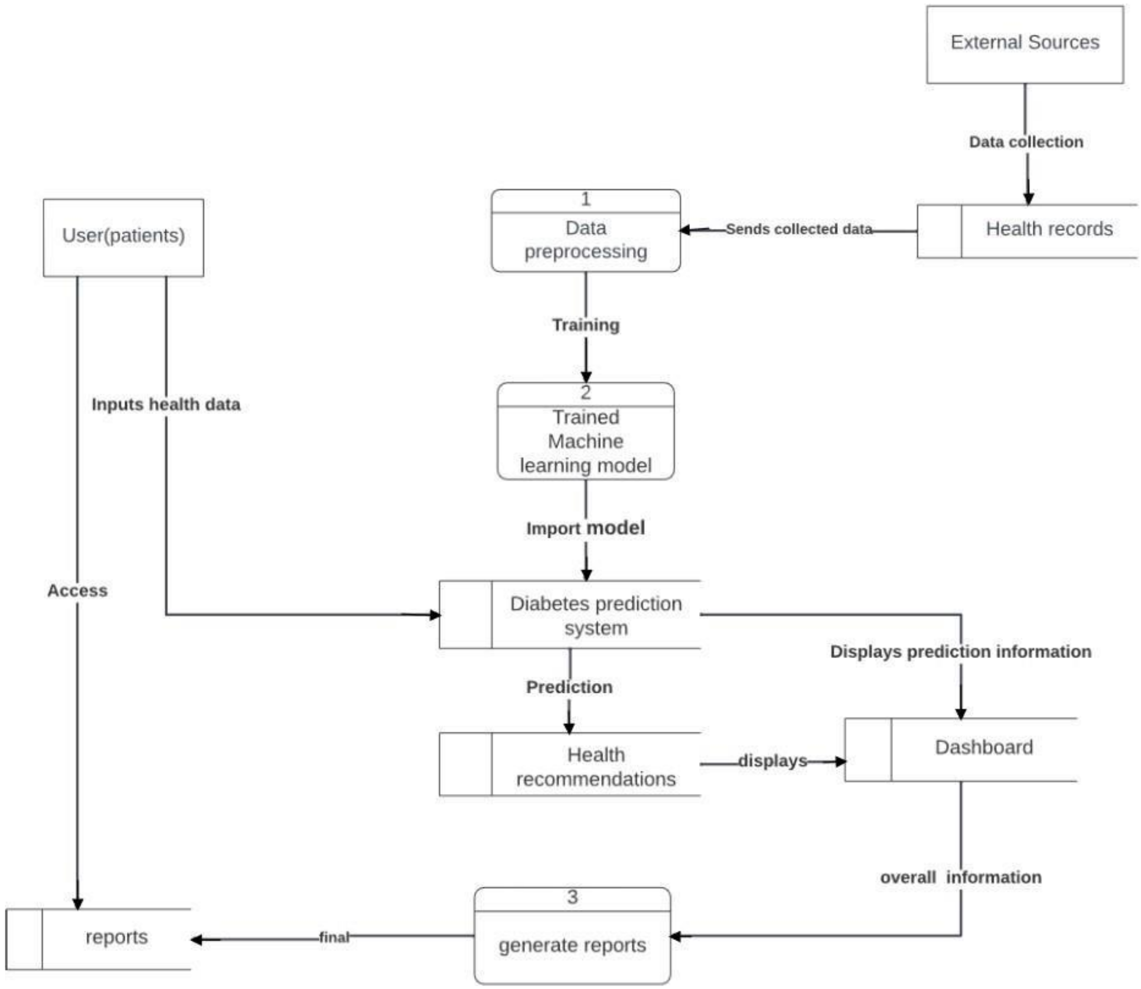
4.2 Non-Functional requirements

The non-functional requirements for the diabetes prediction project include:

- Accuracy: The predictive model should achieve a high level of accuracy in predicting the likelihood of diabetes.
- Performance: The system should be able to process input data and provide predictions in a timely manner.
- Usability: The user interface should be intuitive and easy to use, requiring minimal technical expertise.
- Security: The system should ensure the confidentiality and privacy of user data.
- Scalability: The system should be able to handle a large volume of data and accommodate future growth.

5. PROJECT DESIGN

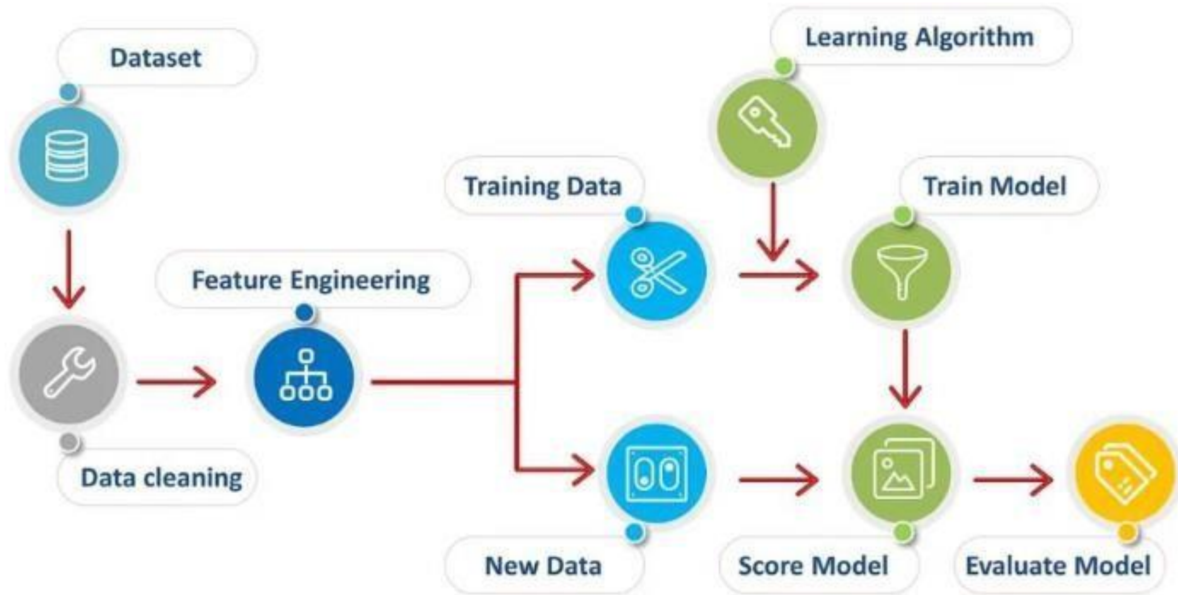
5.1 Data Flow Diagrams & User Stories



User Stories

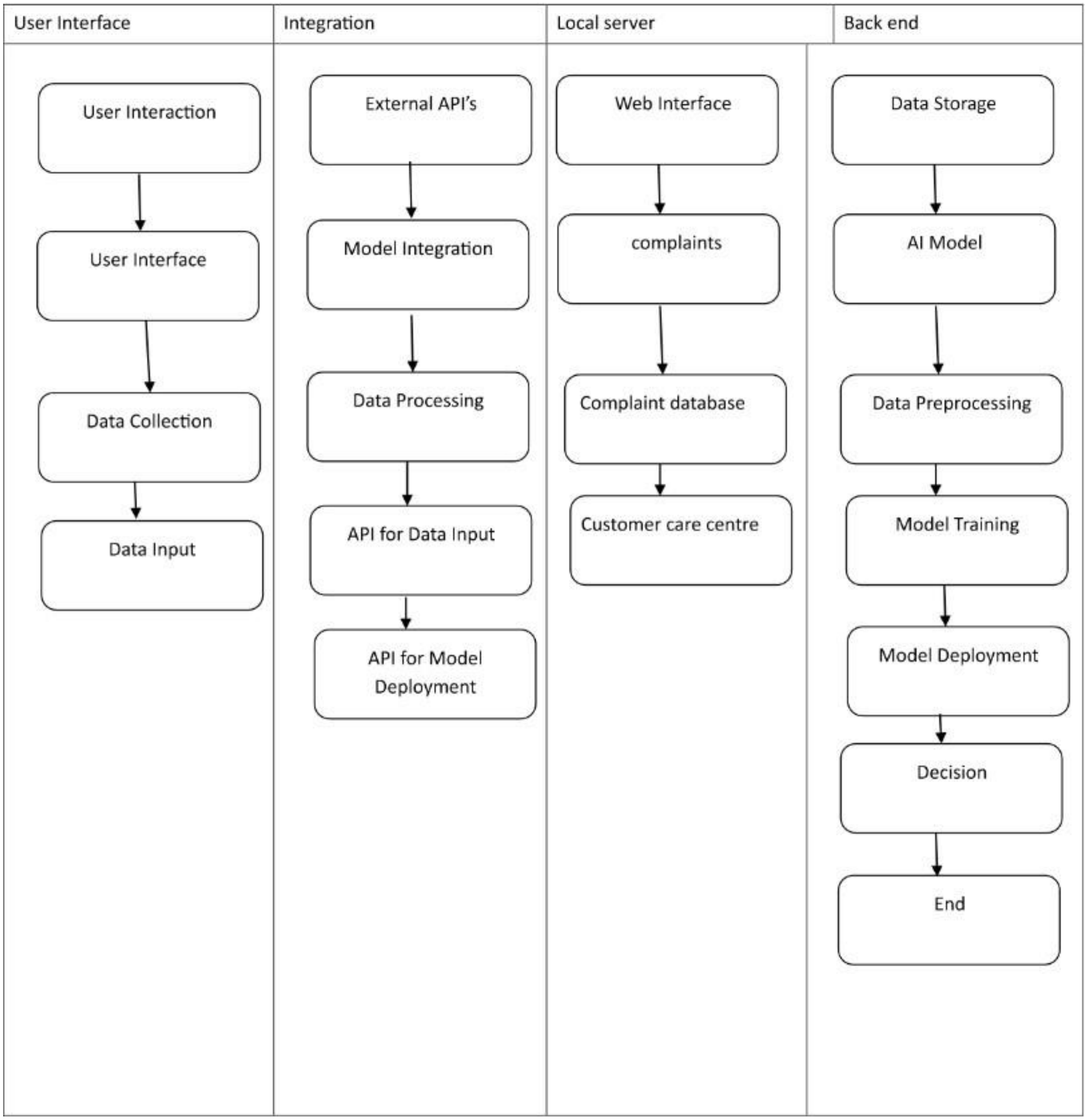
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Healthcare Professionals	Predict Diabetes Onset	USN-1	As a healthcare professional, I want to input patient health records and relevant parameters to predict the onset of diabetes.	The system should accept input data including blood pressure, BMI, heart diseases, cholesterol levels, age, family history, and lifestyle habits.	High	Sprint 1
Healthcare Institutions and local government	Data collection	USN-2	As a health care institutions, I want to collect and Gather a comprehensive dataset of health records and relevant parameters for training the diabetes prediction model.	Collect a diverse and representative dataset containing information such as blood pressure, BMI, heart diseases, cholesterol levels, age, family history, and lifestyle habits	High	Sprint 1
Researchers and Academics	data preprocessing	USN-3	Preprocess the collected dataset by cleaning, normalizing, and splitting it into training and validation sets.	Successfully clean and preprocess the dataset, handling missing values, outliers, and data inconsistencies.	High	Sprint 2
Healthcare Professionals	Model Development & Training	USN-4	select the most suitable model for predicting diabetes onset and Train the selected machine learning model using the preprocessed dataset.	Train the model using the preprocessed dataset. Monitor and optimize the model's performance on the validation set	High	Sprint 3
System Administrators	Model Deployment & Integration	USN-5	As a system Administrator, I want to Deploy the trained machine learning model as a service or API and integrate it into a user-friendly interface.	Develop a user interface for individuals to input their health records and receive diabetes prediction results.	medium	Sprint 4
Individuals/Patients	Personalized Risk Assessment	USN-6	As an individual, I want to input my health data into the system to receive a personalized risk assessment for diabetes onset.	The report should explain the factors that contribute to their risk of diabetes, and provide recommendations for reducing their risk	medium	Sprint 5
Researchers And Academics	Model Evaluation and Enhancement	USN-7	As a researcher, I want tools to evaluate the effectiveness of the diabetes prediction model and continuously enhance its performance.	Implement model evaluation metrics (e.g., accuracy, precision, recall).	medium	Sprint 5

5.2 Solution Architecture



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

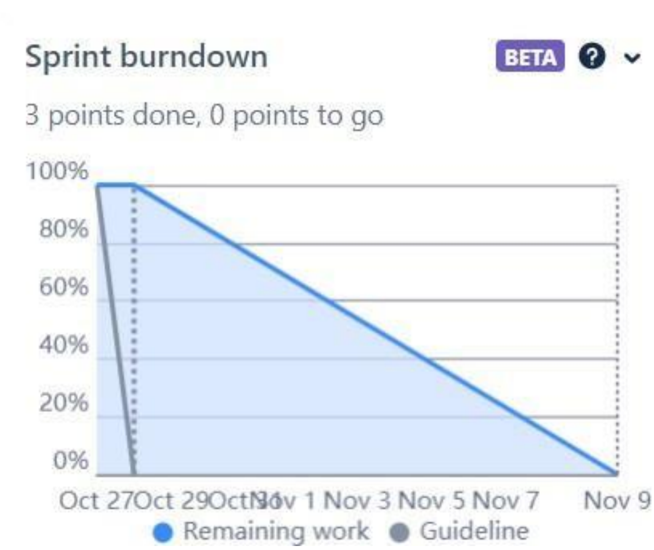

User Interface
Integration
Local server
Back end
User Interaction
User Interface
Data Collection
Data Input
External API's
Model Integration
Data Processing
API for Data Input
API for Model Deployment
Web Interface
complaints
Complaint database
Customer care centre
Data Storage
AI Model
Data Preprocessing
Model Training
Model Deployment
Decision
End

6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project Initialization & Infrastructure Setup	USN-1	As a healthcare professional, I want Set up the development environment with the required tools and frameworks to start the diabetes prediction project.	1	High	Tasneem
Sprint-1	Data collection	USN-2	As a health care institutions, I want to collect and Gather a comprehensive dataset of health records and relevant parameters for training the diabetes prediction model.	2	High	Prasuna
Sprint-2	data preprocessing	USN-3	Preprocess the collected dataset by cleaning, normalizing, and splitting it into training and validation sets.	3	High	Prasuna
Sprint-3	Model Development & Training	USN-4	select the most suitable model for predicting diabetes onset and Train the selected machine learning model using the preprocessed dataset.	5	High	Shreya
Sprint-4	model deployment & Integration	USN-5	As a system Administrator, I want to Deploy the trained machine learning model as a service or API and integrate it into a user-friendly interface.	6	High	Harini
Sprint-5	Personalized Risk Assessment	USN-6	As an individual, I want to input my health data into the system to receive a personalized risk assessment for diabetes onset.	1	medium	Tasneem
Sprint-5	Model Evaluation and Enhancement	USN-7	As a researcher, I want tools to evaluate the effectiveness of the diabetes prediction model and continuously enhance its performance.	2	High	Tasneem

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	3	1 Day	28 oct 2023	28 oct 2023	3	28 oct 2023
Sprint-2	3	1 Day	29 oct 2023	29 oct 2023	3	29 oct 2023
Sprint-3	5	3 Days	30 oct 2023	1 nov 2023	5	1 nov 2023
Sprint-4	6	3 Days	2 nov 2023	4 nov 2023	6	4 nov 2023
Sprint-5	3	2 Days	5 nov 2023	6 nov 2023	3	6 nov 2023



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1 data analysis, data preprocessing, model deployment:

```
import pandas as pd import numpy
as np import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler from
sklearn.model_selection import train_test_split from
imblearn.over_sampling import SMOTE from
sklearn.linear_model import LogisticRegression from
sklearn.neighbors import KNeighborsClassifier from
sklearn.naive_bayes import GaussianNB from
sklearn.tree import DecisionTreeClassifier from
sklearn.ensemble import RandomForestClassifier from
sklearn.svm import SVC
from sklearn.metrics import accuracy_score,
confusion_matrix,classification_report

#load the dataset
df=pd.read_csv('/content/diabetes_012_health_indicators_BRFSS2015.csv')
df.head()
# descriptive statistics
# statistical analysis of the data
df.describe()
#comparing all attributes with Diabetes_012
for i in attributes:

plt.figure(figsize=(5,3))

pd.crosstab(df[i],df.Diabetes_012).plot(kind="bar",figsize=(30,12),color=['
green', 'yellow','red' ])

 plt.show()
up_outlier =['Stroke',
```

```
'HeartDiseaseorAttack','HvyAlcoholConsump','NoDocbcCost','GenHlth',
'MentHlth', 'PhysHlth', 'DiffWalk']
low_outlier = ['CholCheck','PhysActivity','Veggies','AnyHealthcare']
#outlier removal by replacement with median
q1=df.BMI.quantile(0.25)          #          q1
q3=df.BMI.quantile(0.75) # q3
IQR=q3-q1

upper_limit=q3+1.5*IQR # upper linit value

lower_limit=q1-1.5*IQR

df['BMI']=np.where(df['BMI']>upper_limit,27,df['BMI'])
df['BMI']=np.where(df['BMI']<lower_limit,27,df['BMI'])

#outlier removal by replacement with median
for i in up_outlier:
 q1=df[i].quantile(0.25) # q1
q3=df[i].quantile(0.75) # q3  IQR=q3-q1
 upper_limit=q3+1.5*IQR # upper linit value  lower_limit=q1-1.5*IQR
 df[i]=np.where(df[i]>upper_limit,df[i].median(),df[i])

for i in low_outlier:
 q1=df[i].quantile(0.25) # q1
q3=df[i].quantile(0.75) # q3  IQR=q3-q1
 upper_limit=q3+1.5*IQR # upper linit value  lower_limit=q1-1.5*IQR
 df[i]=np.where(df[i]<lower_limit,df[i].median(),df[i])

 q1=df['PhysHlth'].quantile(0.25) # q1
q3=df['PhysHlth'].quantile(0.75) # q3  IQR=q3-q1
```

```
 upper_limit=q3+1.5*IQR # upper linit value  lower_limit=q1-1.5*IQR

df['PhysHlth']=np.where(df['PhysHlth']>upper_limit,df['PhysHlth'].m
edian()
,df['PhysHlth'])
 q1=df['MentHlth'].quantile(0.25) # q1
q3=df['MentHlth'].quantile(0.75) # q3  IQR=q3-q1
 upper_limit=q3+1.5*IQR # upper linit value  lower_limit=q1-1.5*IQR

df['MentHlth']=np.where(df['MentHlth']>upper_limit,df['MentHlth'].median(
)
,df['MentHlth'])
x=df.iloc[:,1:]
x.head()
y=df.Diabetes_012
y.head()
#balancing the data  smote=SMOTE()
x_smote,y_smote = smote.fit_resample(x,y)
# train test split  x_train,x_test,y_train,y_test
=
train_test_split(x_smote,y_smote,test_size=0.3,random_state=47)
# scaling independent variables
scale      =      MinMaxScaler()
x_train_scaled=
pd.DataFrame(scale.fit_transform(x_train),columns=x_train.columns)
x_test_scaled=

## Model building

model1 = LogisticRegression()
```

```
model1.fit(x_train_scaled,y_train)          y_pred1          =
model1.predict(x_test_scaled)          y_pred1_train          =
model1.predict(x_train_scaled)   print('Testing  Accuracy  =  ',
accuracy_score(y_test,y_pred1))   print('Training  Accuracy  =  ',
accuracy_score(y_train,y_pred1_train)) # model building -2

model2                    =
KNeighborsClassifier()
model2.fit(x_train_scaled,y_train)

y_pred2= model2.predict(x_test_scaled)
y_pred2_train   =   model2.predict(x_train_scaled)   print('Testing
Accuracy  =  ',  accuracy_score(y_test,y_pred2))   print('Training
Accuracy  =  ',  accuracy_score(y_train,y_pred2_train))  model3  =
GaussianNB()

model3.fit(x_train_scaled,y_train)

y_pred3                          =
model3.predict(x_test_scaled)

y_pred3_train                          =
model3.predict(x_train_scaled)

print("Test accuracy", accuracy_score(y_test,y_pred3))
print("Train accuracy", accuracy_score(y_train,y_pred3_train))
model4        =        DecisionTreeClassifier(max_depth=12)
model4.fit(x_train_scaled,y_train)          y_pred4          =
model4.predict(x_test_scaled)

y_pred4_train  =  model4.predict(x_train_scaled)  print("Test  accuracy",
accuracy_score(y_test,y_pred4))          print("Train          accuracy",
```

```
accuracy_score(y_train,y_pred4_train))
pd.DataFrame(scale.fit_transform(x_test),columns=x_test.columns)model5.fi
t(x
_train_scaled,y_train)
y_pred5 = model5.predict(x_test_scaled)

y_pred5_train  =  model5.predict(x_train_scaled)  print("Test
accuracy",    accuracy_score(y_test,y_pred5))    print("Train
accuracy", accuracy_score(y_train,y_pred5_train))

pd.crosstab(y_test,y_pred5)
print(classification_report(y_test,y_pred5))
from sklearn.metrics import confusion_matrix, classification_report
# Plot the confusion matrix using Seaborn's heatmap cm5
= confusion_matrix(y_test,y_pred5)
sns.heatmap(cm5, annot=True, fmt='d', cmap='Blues', cbar=True)

plt.show()
import pickle
pickle.dump(model5,open('db_prediction.pkl','wb'))

model5 =RandomForestClassifier(max_depth = 12,n_estimators =
10, random_state=47)
```

7.2 Feature 2

Flask file code:

```
from   flask   import   Flask,   render_template,
request
import pickle import
numpy  as  np  import
```

```
pandas as pd  app =
Flask(__name__)


# Load the model from the pickle file
model = pickle.load(open('db_prediction.pkl','rb'))


@app.route('/')
def start():
    return render_template('index.html')


@app.route("/login", methods=['POST'])
def login():
    Sex = request.form['Sex']
    if (Sex=='Male'):
    Sex=1
    else:
        Sex=0


    HighBP = request.form['HighBP']
    if (HighBP=='yes'):
    HighBP=1   else:
        HighBP=0


    Fruits = request.form['Fruits']
    if (Fruits=='yes'):
    Fruits=1   else:
        Fruits=0


Veggies = request.form['Veggies']  if
(Veggies=='yes'):
    Veggies=1  else:
```

```
        Veggies=0

    AnyHealthcare = request.form['AnyHealthcare']
    if (AnyHealthcare=='yes'):
        AnyHealthcare=1  else:
            AnyHealthcare=0

Highchol =request.form['Highchol']  if
(Highchol=='yes'):
        Highchol=1  else:
        Highchol=0

Age= request.form['Age']  if
(Age=='1'):
        Age=1
        elif
(Age=='2'):
        Age=2
    elif (Age=='3'):
        Age=3
        elif
(Age=='4'):
        Age=4
        elif
(Age=='5'):
        Age=5
        elif
(Age=='6'):
        Age=6
        elif
(Age=='7'):
```

```
        Age=7
        elif
(Age=='8'):
        Age=8
        elif
(Age=='9'):
        Age=9       elif
(Age=='10'):
        Age=10      elif
(Age=='11'):
        Age=11      elif
(Age=='12'):
        Age=12
        else:
             Age=13

      BMI = request.form['BMI']

        Smoker = request.form['Smoker']
        if (Smoker=='yes'):
        Smoker=1    else:
             Smoker=0

        CholCheck = request.form['CholCheck']
        if (CholCheck=='yes'):
        CholCheck=1
        else:
             CholCheck=0

      Stroke = request.form['Stroke']
```

```
    if (Stroke=='yes'):
        Stroke=1    else:
            Stroke=0

    HvyAlcoholConsump= request.form['HvyAlcoholConsump']
        if
(HvyAlcoholConsump=='yes'):
        HvyAlcoholConsump=1    else:
            HvyAlcoholConsump=0

Diffwalk= request.form['Diffwalk'] if
(Diffwalk=='yes'):
        Diffwalk=1 else:
            Diffwalk=0

    HeartDiseaseorAttack = request.form['HeartDiseaseorAttack']
        if
(HeartDiseaseorAttack=='yes'):
        HeartDiseaseorAttack=1
        else:
            HeartDiseaseorAttack=0

PhysActivity= request.form['PhysActivity'] if
(PhysActivity=='yes'):
        PhysActivity=1    else:
            PhysActivity=0

NoDocbcCost = request.form['NoDocbcCost'] if
(NoDocbcCost=='yes'):
```

```
    NoDocbcCost=1
    else:
        NoDocbcCost=0


    GenHlth =request.form['GenHlth']
    if (GenHlth=='1'):
        GenHlth=1
    elif (GenHlth=='2'):
        GenHlth=2
    elif (GenHlth=='3'):
        GenHlth=3
    elif (GenHlth=='4'):
    GenHlth=4  else:
        GenHlth=5


    MentHlth = request.form['MentHlth']
    PhysHlth =request.form['PhysHlth']


Education =request.form['Education']  if
(Education=='1'):
        Education=1
    elif (Education=='2'):
        Education=2
    elif (Education=='3'):
        Education=3
    elif (Education=='4'):
    Education=4
    else:
        Education=5

```

```
Income =request.form['Income'] if
(Income=='1'):
        Income=1
    elif (Income=='2'):
        Income=2
    elif (Income=='3'):
        Income=3
    elif (Income=='4'):
        Income=4
    elif (Income=='5'):
        Income=5
    elif (Income=='6'):
     Income=6    elif
(Income=='7'):
     Income=7      else:
        Income=8


  input_data =[[float(HighBP) ,float(Highchol) ,
float(CholCheck),float(BMI),float(Smoker),float(Stroke),float(HeartDiseaseo
rAtt ack),float(PhysActivity),float(Fruits),float(Veggies),
float(HvyAlcoholConsump), float(AnyHealthcare), float(NoDocbcCost),
float(GenHlth),float(PhysHlth),float(MentHlth),float(Diffwalk),float(Sex),f
loat(A ge),float(Education),float(Income) ]]

   output= model.predict(input_data)


    if (output == 0):
     return render_template('result.html', result='Good News, Diabetes not
present',rec="According to our prediction model, there is currently no
indication that you have diabetes. However, it's important to view this
```

```
as a snapshot, and maintaining a healthy lifestyle is key to ongoing well- being.",rec1="To continue minimizing the risk of diabetes, we recommend staying active, eating a balanced diet rich in fruits and vegetables, and maintaining a healthy weight. Regular exercise and a nutritious diet contribute to overall health and well-being.")     elif (output == 1):
     return render_template('result.html', result='there is a risk of getting diabetes',rec="Our analysis suggests that there is a potential risk for developing diabetes in the future. It's essential to understand that this is not a definite prediction but an indication to be mindful of your health.",rec1="To reduce the risk of developing diabetes, consider adopting preventive measures. Focus on maintaining a balanced diet, engaging in regular physical activity, and scheduling routine health check-ups. These lifestyle changes can play a significant role in minimizing the risk.")     else:
     return render_template('result.html', result='Oh no! you are sufferning from diabetes.',rec="Based on our analysis, it appears that you currently have diabetes. We understand that this may be concerning, but it's important not to panic. This prediction is not a diagnosis, and we strongly recommend consulting with a healthcare professional for a comprehensive evaluation.",rec1="To address this concern promptly, we recommend scheduling a check-up with your healthcare provider. They can provide personalized advice and guidance. Additionally, consider making immediate lifestyle changes such as adjusting your diet and incorporating regular exercise into your routine.")

if __name__ == '__main__':
    app.run(debug=True)
```

8. PERFORMANCE TESTING 8.1

Performace Metrics logistic regression:

```python
print('Testing Accuracy = ', accuracy_score(y_test,y_pred1))
print('Training Accuracy = ', accuracy_score(y_train,y_pred1_train))
```

```
Testing Accuracy =  0.5076663911029309
Training Accuracy =  0.5078034475996934
```

```python
pd.crosstab(y_test,y_pred1)
```

col_0	0.0	1.0	2.0
Diabetes_012			
0.0	42438	8602	13177
1.0	17822	14855	31431
2.0	11428	12232	40348

```python
print(classification_report(y_test,y_pred1))
```

```
              precision    recall  f1-score   support

         0.0       0.59      0.66      0.62     64217
         1.0       0.42      0.23      0.30     64108
         2.0       0.47      0.63      0.54     64008

    accuracy                           0.51    192333
   macro avg       0.49      0.51      0.49    192333
weighted avg       0.49      0.51      0.49    192333
```

Knn:

```
print('Testing Accuracy = ', accuracy_score(y_test,y_pred2))
print('Training Accuracy = ', accuracy_score(y_train,y_pred2_train))
```

```
Testing Accuracy =  0.8205664134599887
Training Accuracy =  0.8816759363245806
```

```
pd.crosstab(y_test,y_pred2)
```

col_0	0.0	1.0	2.0
Diabetes_012			
0.0	51158	2716	10343
1.0	1285	61188	1635
2.0	9534	8998	45476

```
print(classification_report(y_test,y_pred2))
```

```
              precision    recall  f1-score   support

         0.0       0.83      0.80      0.81     64217
         1.0       0.84      0.95      0.89     64108
         2.0       0.79      0.71      0.75     64008

    accuracy                           0.82    192333
   macro avg       0.82      0.82      0.82    192333
weighted avg       0.82      0.82      0.82    192333
```

Naive bayes:

```
print("Test accuracy", accuracy_score(y_test,y_pred3))
print("Train accuracy", accuracy_score(y_train,y_pred3_train))
```

```
Test accuracy 0.5113059121419621
Train accuracy 0.5119591956789133
```

```
pd.crosstab(y_test,y_pred3)
```

col_0	0.0	1.0	2.0
Diabetes_012			
0.0	42975	7094	14148
1.0	18040	13005	33063
2.0	11843	9804	42361

```
print(classification_report(y_test,y_pred3))
```

```
              precision    recall  f1-score   support

         0.0       0.59      0.67      0.63     64217
         1.0       0.43      0.20      0.28     64108
         2.0       0.47      0.66      0.55     64008

    accuracy                           0.51    192333
   macro avg       0.50      0.51      0.49    192333
weighted avg       0.50      0.51      0.49    192333
```

Decision tree:

```
print("Test accuracy", accuracy_score(y_test,y_pred4))
print("Train accuracy", accuracy_score(y_train,y_pred4_train))
```

```
Test accuracy 0.6845471135998502
Train accuracy 0.6935910119970765
```

```
pd.crosstab(y_test,y_pred4)
```

col_0	0.0	1.0	2.0
Diabetes_012			
0.0	50924	1947	11346
1.0	5532	43746	14830
2.0	10585	16432	36991

```
print(classification_report(y_test,y_pred4))
```

```
              precision    recall  f1-score   support

         0.0       0.76      0.79      0.78     64217
         1.0       0.70      0.68      0.69     64108
         2.0       0.59      0.58      0.58     64008

    accuracy                           0.68    192333
   macro avg       0.68      0.68      0.68    192333
weighted avg       0.68      0.68      0.68    192333
```

Random forest :

```
print("Test accuracy", accuracy_score(y_test,y_pred5))
print("Train accuracy", accuracy_score(y_train,y_pred5_train))
```

```
Test accuracy 0.7291364456437532
Train accuracy 0.7413475765192434
```

```
pd.crosstab(y_test,y_pred5)
```

col_0	0.0	1.0	2.0
Diabetes_012			
0.0	50611	586	13020
1.0	2988	47276	13844
2.0	8231	13427	42350

```
print(classification_report(y_test,y_pred5))
```

```
              precision    recall  f1-score   support

         0.0       0.82      0.79      0.80     64217
         1.0       0.77      0.74      0.75     64108
         2.0       0.61      0.66      0.64     64008

    accuracy                           0.73    192333
   macro avg       0.73      0.73      0.73    192333
weighted avg       0.73      0.73      0.73    192333
```

9. ADVANTAGES & DISADVANTAGES

Advantages:

- Early detection: The diabetes prediction project allows for the early identification of individuals at risk of developing diabetes, enabling preventive measures and lifestyle modifications.
- Improved outcomes: By predicting the likelihood of diabetes, healthcare professionals

can intervene early, leading to better management of the condition and improved health outcomes for patients.

- Personalized care: Predictive models can help tailor treatment plans based on individual risk factors, leading to more targeted and effective interventions.
- Cost-effective: Identifying high-risk individuals for diabetes early on can help healthcare systems allocate resources more efficiently and reduce the economic burden associated with the condition.

Disadvantages:

- False positives/negatives: Predictive models may occasionally produce false-positive or false-negative results, leading to unnecessary interventions or missed opportunities for early intervention.
- Privacy concerns: Collecting and analyzing personal health data for predictive purposes raises privacy concerns, requiring stringent data protection measures and patient consent.
- Limited accuracy: While predictive models have shown promising results, their accuracy may vary based on the quality and quantity of available data, as well as the complexity of individual risk factors.
- Ethical considerations: The utilization of predictive models for diabetes prediction necessitates careful ethical considerations, such as ensuring equity, transparency, and avoiding discrimination based on predicted risk profiles.

10. CONCLUSION

The diabetes prediction project holds significant potential in improving public health by enabling early detection and prevention of diabetes. By leveraging advanced data analytics and personalized approaches, the project can empower individuals to take control of their health and make informed decisions. However, it is crucial to address the project's limitations, such as data quality, privacy concerns, and equitable access, to maximize its benefits. With proper implementation and continuous refinement, the diabetes prediction project can contribute to reducing the burden of diabetes and improving overall health outcomes.

11. FUTURE SCOPE

The future scope of the diabetes prediction project is promising and can be expanded in several ways:

- Refinement of Prediction Models: Continuous improvement of the prediction models can enhance accuracy and reliability by incorporating new data sources, advanced machine learning algorithms, and feedback from real-world implementations.
- Integration with Wearable Devices: Integrating the prediction project with wearable

devices, such as fitness trackers or continuous glucose monitors, can provide real-time health data for more accurate predictions and personalized interventions.

- Behavioral and Lifestyle Interventions: The project can integrate behavioral and lifestyle interventions to support individuals in making healthier choices, such as personalized diet plans, exercise recommendations, and stress management techniques.
- Telemedicine and Remote Monitoring: Leveraging telemedicine technologies and remote monitoring systems can enable individuals to receive timely interventions, monitor their health remotely, and provide healthcare professionals with valuable data for continuous assessment.
- Collaborative Efforts: Collaboration between healthcare providers, researchers, and technology companies can facilitate the development of comprehensive and standardized prediction models, ensuring wider adoption and impact.