IBM NAAN MUDHALVAN AI – PHASE 1

AI BASED DIABETES PREDICTION SYSTEM

TEAM MEMBER

NAME: Muthu R

REGISTER NO: 721921243072

Project Title: AI Based Diabetes Prediction System

Phase-1 : Problem Defination and design thinking

Problem Definition:

The problem is to develop an AI-based diabetes prediction system that can accurately predict the likelihood of an individual developing diabetes in the future. Diabetes is a chronic disease with serious health implications, and early detection and intervention can significantly improve a person's quality of life. The system should be able to analyze various health-related data and provide predictions with a high degree of accuracy. The key challenges include data collection, feature selection, model development, and user interface design.

Design Thinking Process:

1. Empathize:
   - Understand the needs and concerns of individuals at risk of diabetes, as well as healthcare professionals.
   - Conduct interviews, surveys, and observations to gather insights into their experiences and expectations regarding diabetes prediction.
2. Define:
   - Clearly define the problem, objectives, and goals of the AI-based diabetes prediction system.
   - Create personas representing potential users and stakeholders, such as patients, doctors, and researchers.
   - Identify the key metrics for success, such as prediction accuracy, user satisfaction, and ease of use.
3. Ideate:
   - Brainstorm potential solutions and features that can address the defined problem.
   - Explore different data sources, such as electronic health records, wearable devices, and lifestyle data, for collecting relevant information.
   - Consider AI and machine learning techniques for predictive modeling, such as logistic regression, decision trees, or deep learning.
4. Prototype:
   - Develop a prototype of the AI-based diabetes prediction system.
   - Create wireframes or mockups for the user interface to visualize the system's functionality and user interactions.

- Build a proof-of-concept machine learning model using sample data to demonstrate its predictive capabilities.

5. Test:
   - Gather feedback from potential users and stakeholders on the prototype.
   - Evaluate the accuracy and reliability of the predictive model using a dataset with known outcomes.
   - Iterate on the design and functionality based on user feedback and model performance.
6. Implement:
   - Develop the AI-based diabetes prediction system with a robust back-end for data processing and machine learning.
   - Create a user-friendly front-end interface accessible through web or mobile applications.
   - Integrate data sources and ensure data security and privacy compliance.
7. Iterate:
   - Continuously improve the system based on user feedback and real-world performance.
   - Monitor the predictive model's accuracy and update it as new data becomes available.
   - Stay updated with advancements in AI and healthcare to incorporate new features and improvements.
8. Launch and Scale:
   - Deploy the AI-based diabetes prediction system in a real-world healthcare setting, partnering with healthcare providers or organizations.
   - Monitor system performance, scalability, and user adoption.
   - Explore opportunities for expanding the system's reach to a wider audience

Source Code

```
# Sample code for data preprocessing

import pandas as pd

# Load your dataset (replace 'data.csv' with your dataset)

data = pd.read_csv('data.csv')

# Perform data preprocessing steps (e.g., handling missing values, encoding categorical variables)

# ...

# Split the data into features (X) and target variable (y)

X = data.drop('diabetes_label', axis=1)

y = data['diabetes_label']

# Sample code for model development (using scikit-learn)

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

# Split the data into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and train a Random Forest classifier (replace with your chosen model)

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)

# Sample code for model evaluation

from sklearn.metrics import accuracy_score, classification_report

# Make predictions on the test data

y_pred = model.predict(X_test)

# Evaluate model performance

accuracy = accuracy_score(y_test, y_pred)

classification_rep = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')

print(f'Classification Report:\n{classification_rep}')
```

Data Sets:

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	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
7	147	76	0	0	39.4	0.257	43	1
1	97	66	15	140	23.2	0.487	22	0
13	145	82	19	110	22.2	0.245	57	0
5	117	92	0	0	34.1	0.337	38	0
5	109	75	26	0	36	0.546	60	0
3	158	76	36	245	31.6	0.851	28	1
3	88	58	11	54	24.8	0.267	22	0
6	92	92	0	0	19.9	0.188	28	0
10	122	78	31	0	27.6	0.512	45	0
4	103	60	33	192	24	0.966	33	0
11	138	76	0	0	33.2	0.42	35	0
9	102	76	37	0	32.9	0.665	46	1
2	90	68	42	0	38.2	0.503	27	1
4	111	72	47	207	37.1	1.39	56	1
3	180	64	25	70	34	0.271	26	0
7	133	84	0	0	40.2	0.696	37	0
7	106	92	18	0	22.7	0.235	48	0
9	171	110	24	240	45.4	0.721	54	1
7	159	64	0	0	27.4	0.294	40	0
0	180	66	39	0	42	1.893	25	1
1	146	56	0	0	29.7	0.564	29	0
2	71	70	27	0	28	0.586	22	0
7	103	66	32	0	39.1	0.344	31	1
7	105	0	0	0	0	0.305	24	0
1	103	80	11	82	19.4	0.491	22	0
1	101	50	15	36	24.2	0.526	26	0
5	88	66	21	23	24.4	0.342	30	0
8	176	90	34	300	33.7	0.467	58	1
7	150	66	42	342	34.7	0.718	42	0
1	73	50	10	0	23	0.248	21	0
7	187	68	39	304	37.7	0.254	41	1
0	100	88	60	110	46.8	0.962	31	0
0	146	82	0	0	40.5	1.781	44	0
0	105	64	41	142	41.5	0.173	22	0
2	84	0	0	0	0	0.304	21	0
8	133	72	0	0	32.9	0.27	39	1
5	44	62	0	0	25	0.587	36	0
2	141	58	34	128	25.4	0.699	24	0
7	114	66	0	0	32.8	0.258	42	1
5	99	74	27	0	29	0.203	32	0
0	109	88	30	0	32.5	0.855	38	1
2	109	92	0	0	42.7	0.845	54	0
1	95	66	13	38	19.6	0.334	25	0
4	146	85	27	100	28.9	0.189	27	0
2	100	66	20	90	32.9	0.867	28	1
5	139	64	35	140	28.6	0.411	26	0
13	126	90	0	0	43.4	0.583	42	1
4	129	86	20	270	35.1	0.231	23	0
1	79	75	30	0	32	0.396	22	0
1	0	48	20	0	24.7	0.14	22	0
7	62	78	0	0	32.6	0.391	41	0
5	95	72	33	0	37.7	0.37	27	0
0	131	0	0	0	43.2	0.27	26	1
2	112	66	22	0	25	0.307	24	0
3	113	44	13	0	22.4	0.14	22	0

2	74	0	0	0	0	0.102	22	0
7	83	78	26	71	29.3	0.767	36	0
0	101	65	28	0	24.6	0.237	22	0
5	137	108	0	0	48.8	0.227	37	1
2	110	74	29	125	32.4	0.698	27	0
13	106	72	54	0	36.6	0.178	45	0
2	100	68	25	71	38.5	0.324	26	0
15	136	70	32	110	37.1	0.153	43	1
1	107	68	19	0	26.5	0.165	24	0
1	80	55	0	0	19.1	0.258	21	0
4	123	80	15	176	32	0.443	34	0