AI BASED DIABETES PREDICTION SYSTEM

Phase 3: Submission

Project: Diabetes prediction system

Introduction:

- An AI-based diabetes prediction system is a sophisticated healthcare application that employs artificial intelligence (AI) and machine learning (ML) techniques to analyze and interpret diverse sets of data to predict the likelihood of an individual developing diabetes.
- Step into the future of healthcare with our AI-based Diabetes Prediction System. This innovative solution leverages the prowess of artificial intelligence to analyze comprehensive datasets, ranging from medical history to lifestyle factors.
- Through advanced machine learning algorithms, it accurately predicts the likelihood of diabetes onset, empowering individuals and healthcare professionals with proactive insights for early intervention and personalized preventive measures.

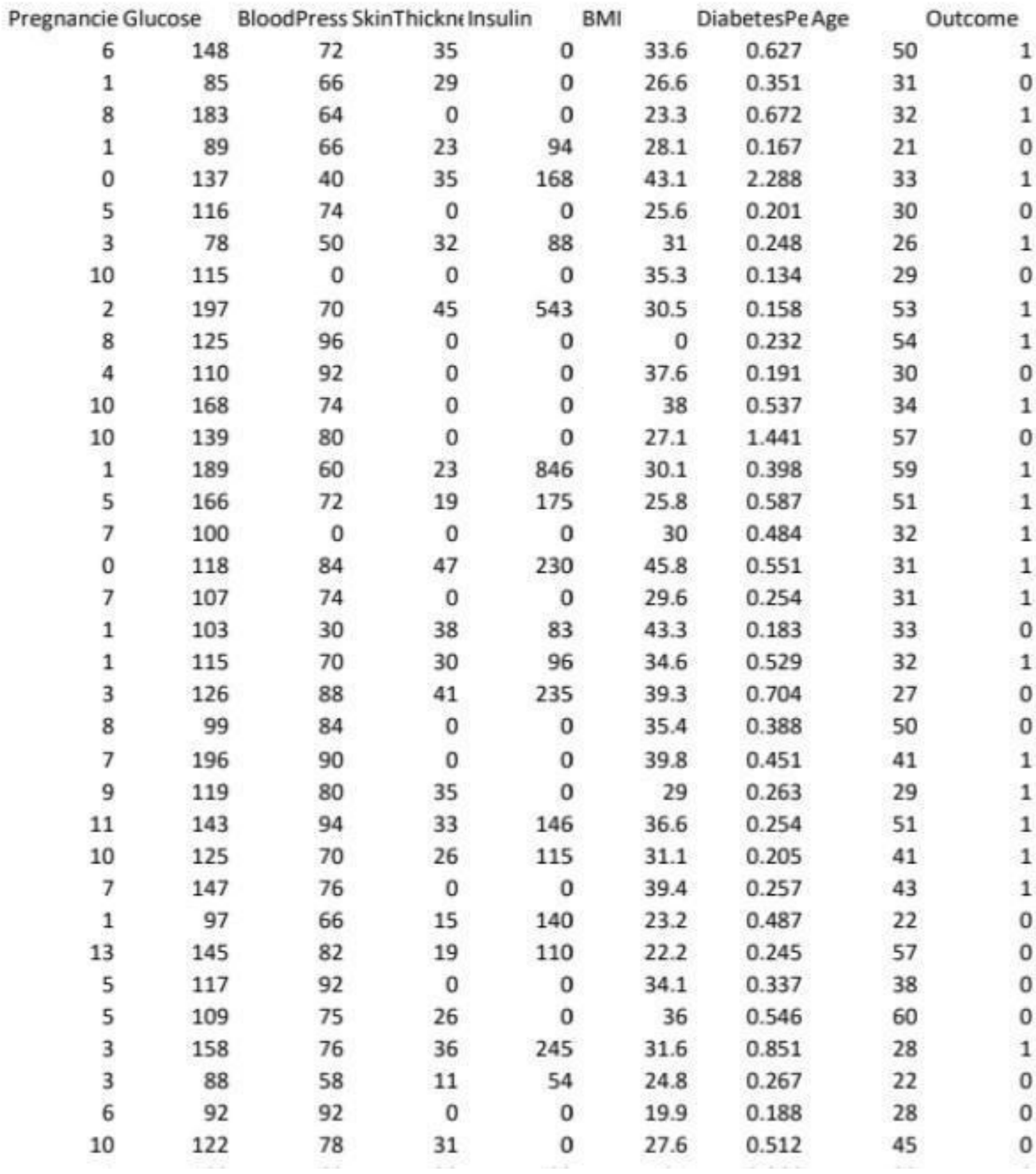
Phase 3: Development Part 1

In this part you will begin building your project by loading and preprocessing the dataset.Start building the AI based diabetes prediction model by loading and preprocessing the dataset.Load the diabetes prediction dataset and preprocess the data.

Data Source

A good data source for diabetes prediction using machine learning should be Accurate whether the person has diabetes or not.

Dataset link :([https://www .kaggle.com/datasets/mathchi/diabetes-data-set](https://www.kaggle.com/datasets/mathchi/diabetes-data-set))



Pregnancie	Glucose	BloodPress	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
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

Import Libraries:

Import the necessary libraries for your project. You'll likely need libraries such as Pandas and NumPy and seaborn.

Source Code:

```
import  pandas as pd import
numpy as np
importmatplotlib.pyploaslt
import  seaborn as sns
```

Load the dataset:

Load your diabetes prediction dataset into a Pandas Data Frame. You can typically load data from a CSV file using the pd.read_csv() function .

Source Code:

```
df = pd.read_csv('../input/pima-indians-diabetes-database/diabetes.csv')

df.head()
```

Explore the data:

Take a look at the data to understand its structure and content. You can use functions like data.head(), data.info(), and data.describe() to get an overview.

Source Code:

```
# Display the first few rows of the dataset print(data.head())
# Get information about the dataset  print(data.info())
# Summary statistics of the data print(data.describe())
```

Data Preprocessing:

- Handling Missing Data:

Check for missing values in the dataset and decide how to handle them (e.g., by filling missing values with the mean, median, or using other techniques).

Source Code:

```
  # Handle missing values (if any)
data.fillna(method='ffill', inplace=True)  # Example: Forward fill missing values
```

- Categorical Data:

If your dataset contains categorical variables, you may need to encode them using techniques like one-hot encoding.

Source Code:

```
# Example one-hot encoding
data = pd.get_dummies(data, columns=['categorical_column'])
```

- Scaling:

Normalize or standardize numerical features to ensure that they are on a similar scale.

Source Code:

```
        from sklearn.preprocessing import StandardScaler
        scaler = StandardScaler()
        data[['numerical_column1', 'numerical_column2']] =
scaler.fit_transform(data[['numerical_column1', 'numerical_column2']])
```

Conclusion and Future Work:

We have loaded and preprocessed the dataset. Next step we can proceed to build the model of diabetes prediction system in the next phase of the project.

Project Conclusion:

Using Random forest algorithm in our prediction system, we can evaluate the performance using the accuracy score, comparing the performance between train and test data and produce accurate prediction values of whether a patient has diabetes or not.