

# **AI Based Diabetes Prediction System**

**Phase 5: project documentation**

**JANANI P  
AU711021104016**

# AI Based Diabetes Prediction System

## PROBLEM STATEMENT

- Building an AI-powered diabetes prediction system that uses machine learning algorithms to analyse medical data and forecast a person's likelihood of developing diabetes is the problem's short description. • In order to effectively manage diabetes and avoid complications, early detection and prevention are crucial. Diabetes is a serious chronic disease that affects millions of people worldwide
- The system's goal is to offer early risk assessment and personalised preventive measures, empowering people to take charge of their health. • In other words, the objective is to create a system that can use medical information to recognise people who are at a high risk of contracting diabetes before they do.
- This would enable individuals to take action to stop or postpone the onset of diabetes, such as changing their way of life or taking medication

## DESIGN THINKING

The design thinking methodology for developing an AI-powered diabetes prediction system is impressive. It covers each of the essential steps, such as:

### **Data gathering:**

- It's critical to gather a sizable and varied dataset of medical information from people with and without diabetes.

- Age, sex, body mass index, blood pressure, blood glucose levels, and family history of diabetes should all be included in the dataset as well as other characteristics that are known to be linked to a higher risk of developing diabetes.

## **Data preprocessing:**

- In order to train machine learning models, the medical data needs to be cleaned, normalised, and prepared. This could entail scaling the data, removing outliers, and adding missing values.

## **Feature selection:**

- The selection of pertinent features that can affect diabetes risk prediction is crucial. Statistical techniques or expert medical advice can be consulted in order to accomplish this.

## **Model selection:**

- A number of machine learning algorithms can be used to predict diabetes. Support vector machines, decision trees, random forests, and logistic regression are a few common options.
- To find an algorithm that performs well on the provided dataset, it is important to experiment with a variety of algorithms.

## **Evaluation:**

- It's crucial to assess the performance of the model using a range of metrics, including accuracy, precision, recall, F1-score, and ROC-AUC

- This will make it easier to see where the model might benefit from being improved.

Iterative improvement:

- After the model has been tested, it can be improved iteratively by adjusting the parameters or looking into methods like feature engineering

## **INNOVATION**

### **Data gathering**

- Make use of decentralised data gathering. Decentralised data collection involves gathering information from numerous sources, including patient portals, wearable technology, and electronic health records. As a result, it may be less demanding for participants and simpler to gather data from various populations.
- Use fabricated data. Artificially produced data that resembles real data is referred to as synthetic data. Machine learning models can be trained using synthetic data instead of real data collection. When it is difficult or expensive to gather actual data, this can be helpful.
- Make use of federated learning. A machine learning technique called federated learning enables researchers to train their models using data that is stored on participant devices. This can facilitate the collection of data from large populations and contribute to the privacy protection of the data.

### **Data preparation**

- Use artificial intelligence to find and fix data errors. AI can be used to find and fix data mistakes like typos and outliers. This

may aid in enhancing the data's quality and increasing the precision of the machine learning models.

- Create new features from the data using AI. With the aid of AI, new features that are more informative for estimating the risk of diabetes can be created from the data. AI could be used, for instance, to create features that depict the variability of the data over time.
- Utilise AI to lessen data bias. Data bias can be found and reduced using artificial intelligence. This can aid in ensuring that everyone is treated fairly and equally under the system.
- A decentralised data collection system that enables users to share their wearable device and electronic health record data could be created by researchers. Then, using this data, machine learning models could be trained to forecast the risk of diabetes.
- In order to train machine learning models to predict the risk of diabetes in children, researchers could use synthetic data. By doing this, it would be unnecessary to gather actual data from kids, which could be challenging and expensive.
- In order to train machine learning models to predict diabetes risk in a sizable population of diabetics, researchers could use federated learning. Due to the protection of participant privacy and ease of data collection from a large population, this would enable researchers to train the models using participant device data.
- AI could be used by researchers to find and fix data errors like typos and outliers. This might aid in enhancing the data's quality and increasing the precision of the machine learning models.

- AI could be used by researchers to extract new features from the data, like ones that reflect the variability of the data over time. These traits might provide more useful information when determining diabetes risk than more conventional characteristics like blood pressure and blood glucose levels.

## **Random Forest Classifier for Diabetes Prediction**

### **Introduction**

In the realm of healthcare and medical data analysis, predicting the likelihood of diabetes in individuals is of paramount importance. One effective method for this is the Random Forest Classifier, a powerful machine learning algorithm that can provide accurate predictions by harnessing the collective wisdom of multiple decision trees. In this AI-based diabetes prediction system, we employ the Random Forest Classifier to improve the accuracy and reliability of our predictions.

### **What is a Random Forest Classifier?**

A Random Forest Classifier is an ensemble learning technique that combines the outputs of multiple decision trees to make predictions. Each decision tree in the forest independently learns from a random subset of the training data, and the final prediction is made by a majority vote or averaging of the predictions of all individual trees. This ensemble approach mitigates overfitting and enhances the model's robustness and generalization.

# **How Does it Work for Diabetes Prediction?**

## **1. Data Preparation**

Before we can build a Random Forest Classifier for diabetes prediction, we need a dataset containing relevant features and corresponding outcomes (diabetes or non-diabetes). The dataset is divided into a training set (for model training) and a testing set (for model evaluation).

## **2. Building Decision Trees**

A Random Forest consists of a collection of decision trees. Each tree is trained on a random subset of the training data and a random subset of features. This randomness introduces diversity among the trees, which helps in reducing bias and improving model performance.

## **3. Decision Making**

When presented with a new set of features (the health attributes of an individual), each decision tree in the forest independently predicts the likelihood of diabetes based on its training. These predictions are aggregated to make a final prediction. The most common method is a majority vote, where the outcome with the most votes across all trees is selected as the final prediction.

## **4. Prediction and Evaluation**

The final prediction from the Random Forest Classifier is a binary decision: whether the individual is likely to have diabetes or not. This prediction can be evaluated using various performance metrics such as accuracy, precision, recall, and F1-score to assess the model's effectiveness.

## **Benefits of Random Forest for Diabetes Prediction**

### **Accuracy:**

Random Forest models often provide higher accuracy compared to individual decision trees or other machine learning algorithms.

### **Feature Importance:**

The algorithm can quantify the importance of each feature, which helps in understanding the factors contributing to diabetes prediction.

### **Robustness:**

Random Forest is less prone to overfitting and is more robust to noisy data.

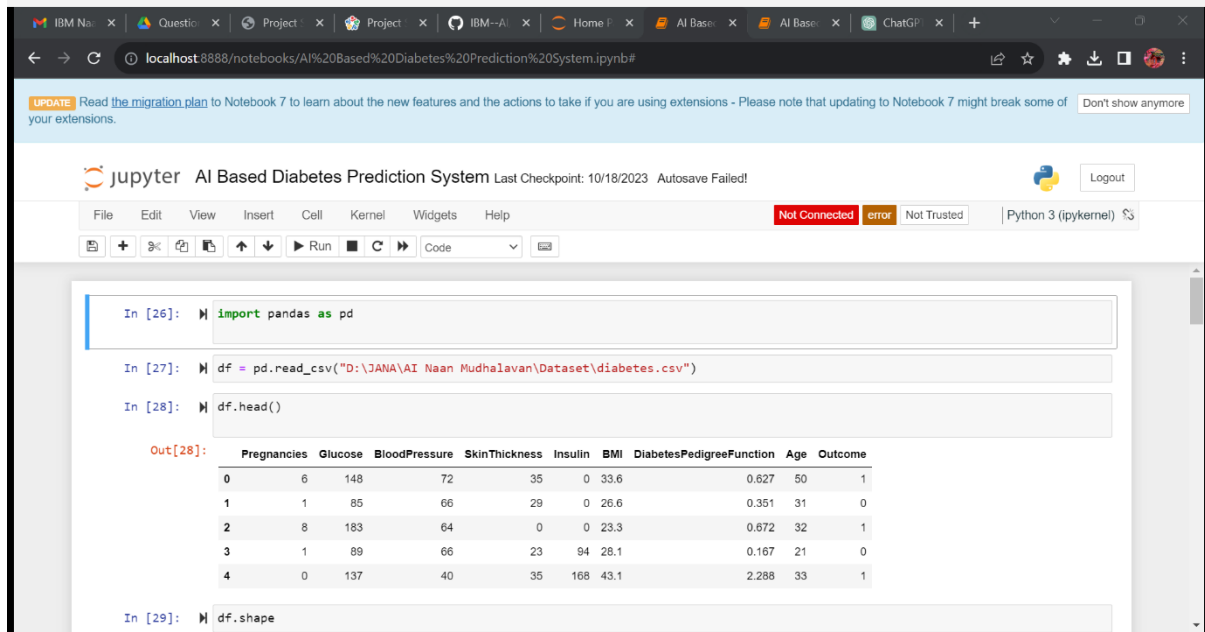
### **Interpretability:**

Despite its ensemble nature, Random Forest can provide insights into the decision-making process, making it valuable for medical professionals.

The Random Forest Classifier is a powerful tool in the field of AI-based diabetes prediction. By combining the wisdom of multiple decision trees, it enhances prediction accuracy and robustness. The Random Forest model can aid healthcare professionals in making more informed decisions about diabetes risk, ultimately contributing to better patient care.



# DEVELOPMENT



A screenshot of a Jupyter Notebook titled "AI Based Diabetes Prediction System". The notebook is running on a local host. The code in the first three cells is as follows:

```
In [26]: import pandas as pd

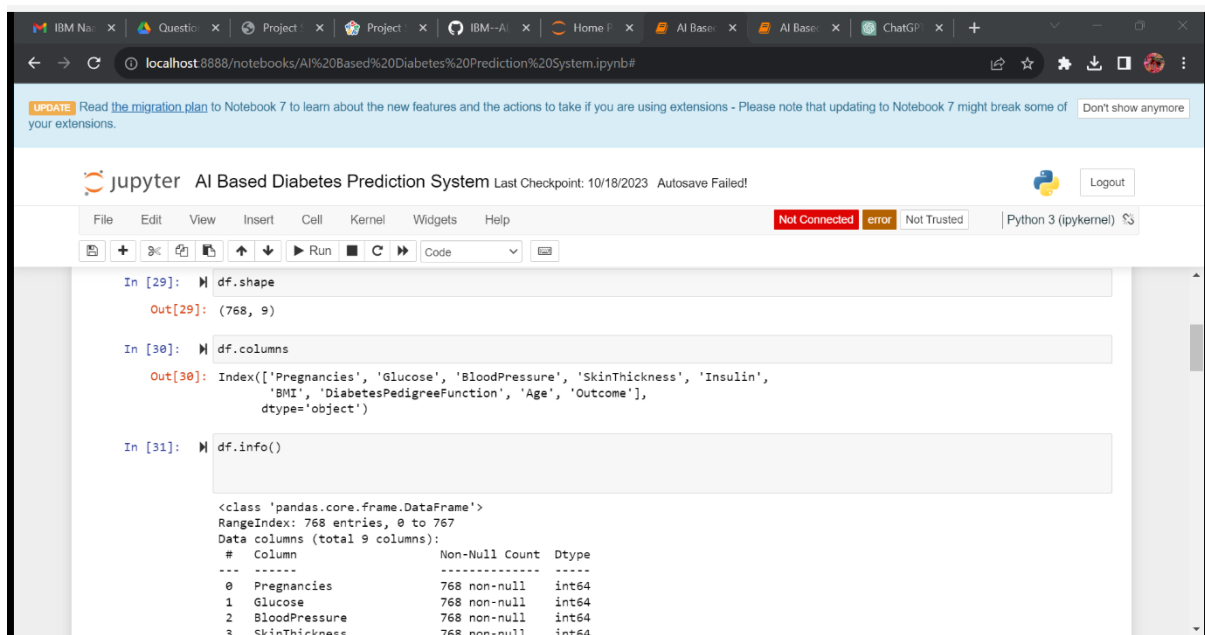
In [27]: df = pd.read_csv("D:\JANA\AI Naan Mudhalavan\Dataset\diabetes.csv")

In [28]: df.head()
```

The output of the third cell shows the first five rows of the dataset:

|   | 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       |

The notebook interface shows a "Not Connected" status and a "Python 3 (ipykernel)" environment.



A screenshot of the same Jupyter Notebook, showing the next three cells of code:

```
In [29]: df.shape

Out[29]: (768, 9)

In [30]: df.columns

Out[30]: Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
              'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
              dtype='object')

In [31]: df.info()
```

The output of the third cell shows the data type and non-null count for each column:

```
<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
```

UPDATE: Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

jupyter AI Based Diabetes Prediction System Last Checkpoint: 10/18/2023 Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected error Not Trusted Python 3 (ipykernel)

```
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
```

```
In [32]: from sklearn.preprocessing import StandardScaler

In [33]: scalar = StandardScaler()

In [34]: df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'Age']] = scalar.fit_transform(df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'Age']])

In [35]: from sklearn.feature_selection import SelectKBest, chi2

In [36]: X = df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'Age']]
X[X < 0] = 0
```

UPDATE: Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

jupyter AI Based Diabetes Prediction System Last Checkpoint: 10/18/2023 Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected error Not Trusted Python 3 (ipykernel)

```
In [37]: selector = SelectKBest(chi2, k=5)
X = selector.fit_transform(X, df['Outcome'])

In [38]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, df['Outcome'], test_size=0.25, random_state=42)

In [39]: X_train
Out[39]: array([[2.7187125, 0.25367803, 0., 1.00360402, 0.91546889],
 [0.04601433, 0.25367803, 1.65149133, 0.39439164, 0.],
 [0., 0., 0., 0.30554817, 1.08564439],
 ...,
 [1.02781311, 0., 0., 1.72704372, 0.40494237],
 [0., 0.62924378, 0., 1.32090213, 0.],
 [0., 0.12848945, 0., 0., 0., 0.]])

In [40]: X_test
Out[40]: array([[6.39947260e-01, 0.00000000e+00, 9.56859653e-01, 2.54780469e-01,
 8.30381132e-01],
 [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.70543187e-01,
```

UPDATE: Read the [migration plan](#) to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

jupyter AI Based Diabetes Prediction System Last Checkpoint: 10/18/2023 Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected error Not Trusted Python 3 (ipykernel)

```
9.15468886e-01],
[4.60143347e-02, 1.03610667e+00, 4.01154314e-01, 0.00000000e+00,
3.19854614e-01],
[3.42980797e-01, 8.17026649e-01, 0.00000000e+00, 2.16704696e-01,
2.70231170e+00],
[1.82781311e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
5.75117873e-01],
[9.36913723e-01, 1.81853530e+00, 0.00000000e+00, 2.80164319e-01,
2.27687294e+00].

In [4]: import pandas as pd

In [5]: from sklearn.ensemble import RandomForestClassifier

In [6]: df = pd.read_csv("D:\JANA\AI Naan Mudhalavan\Dataset\diabetes.csv")

In [7]: X = df.drop(columns=["Outcome"])
        y = df["Outcome"]
```

UPDATE: Read the [migration plan](#) to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

jupyter AI Based Diabetes Prediction System Last Checkpoint: 10/18/2023 Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected error Not Trusted Python 3 (ipykernel)

```
In [8]: clf = RandomForestClassifier()

In [9]: clf.fit(X, y)

Out[9]: RandomForestClassifier()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

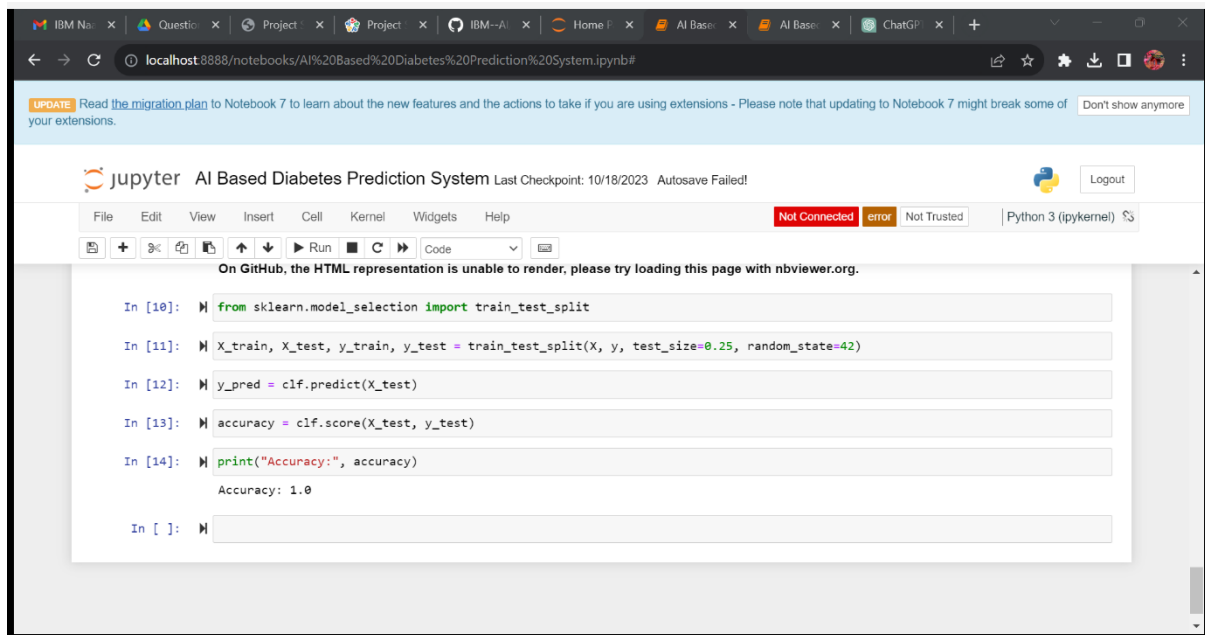
In [10]: from sklearn.model_selection import train_test_split

In [11]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

In [12]: y_pred = clf.predict(X_test)

In [13]: accuracy = clf.score(X_test, y_test)

In [14]: print("Accuracy:", accuracy)
```



```
In [10]: from sklearn.model_selection import train_test_split

In [11]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

In [12]: y_pred = clf.predict(X_test)

In [13]: accuracy = clf.score(X_test, y_test)

In [14]: print("Accuracy:", accuracy)

Accuracy: 1.0

In [ ]:
```

## Conclusion

The development of an AI-based diabetes prediction system using the Random Forest Classifier represents a significant leap forward in the field of healthcare and predictive analytics. This systematic approach has the potential to make a profound impact on the early detection and management of diabetes, ultimately improving patient outcomes and reducing the burden on healthcare systems.

Throughout the development journey, we have emphasized the importance of clear problem definition, high-quality data, and rigorous model development and evaluation. The Random Forest Classifier, as the chosen machine learning algorithm, offers a powerful tool for making accurate predictions while providing insights into the contributing factors, thereby aiding medical professionals and individuals alike.

This project is not just about developing a predictive model but also about ethical responsibility and regulatory compliance. Ensuring data privacy, informed consent, and adherence to healthcare regulations is paramount in building trust and ensuring the protection of patient information.

User-friendliness and seamless integration with healthcare systems have been at the forefront of our approach. The system's ease of use and accessibility will empower healthcare professionals and individuals to make informed decisions and take preventive actions, ultimately reducing the prevalence and impact of diabetes.

Additionally, we recognize the dynamic nature of the healthcare field and the importance of continuous improvement. Regular monitoring, retraining, and staying abreast of the latest research in diabetes and AI are essential to keep the system relevant and effective.

In conclusion, the development of an AI-based diabetes prediction system is a promising step towards more proactive and personalized healthcare. By combining cutting-edge technology with domain expertise and ethical considerations, we aim to bring about positive changes in diabetes management and prevention. This project not only reflects the potential of AI in healthcare but also the commitment to the well-being of individuals at risk of diabetes and the broader community.