AI BASED DIABETES PREDICTION SYSTEM

TEAM MEMBER

510521205043 :SEENURAO

INTRODUCTION:

- Diabetes (T2D) is defined as hyperglycemia caused by abnormal insulin action, resulting in metabolic malfunction in energy generation from ingested glucose (Association 2020). T2D is achronic disease that is increasing the levels of mortality, morbidity, severe complications, and health expenditure** (Khan et al. 2020; Zhou et al. 2016)
- The rate of diabetes diagnosis in the Republic of Korea (ROK) irapidly increasing, similar to global trends. Among people with diabetes, 60% were treated with oral hypogly-cemic regimen or insulin therapy, but only 28.3% of patients achieved the glycemic goal (HbA1c < 6.5%) from 2016 to 2018 (Jung et al. 2021).

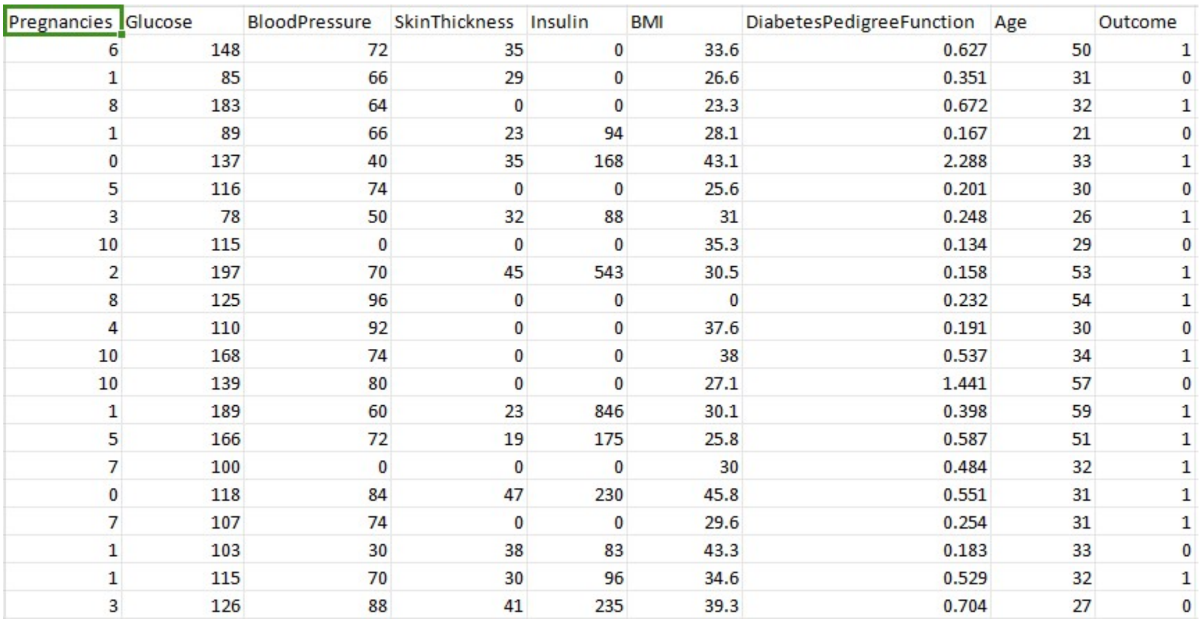
HOW DOES ARTIFICIAL INTELLIGENCE BASED DIABETES SYSTEM WORK:

- ✓ Clinical and laboratory variables were collected from the clinical data warehouse platform and the electronic medical records in Ulsan University Hospital.

- ✓ This study is categorized under the records-based retrospective research; therefore, informed consent by participants was not required.
- ✓ The study was reviewed and the protocol approved by the Institutional Human Experimentation Committee Review Board of Ulsan University Hospital, Republic of Korea (UUH 2020-09-003). The study was conducted in accordance with the ethical standards set forth in the 1964 Declaration of Helsinki.

DATA SOURCE:

Table 1 shows that among the 57 variables, including individual characteris-Tics, spirometry test, vital signs, complete blood count, blood type, inflamma-Tory marker test, liver function test, kidney function test, lipid panel test, Diabetes test, mineral/electrolytes, thyroid function test, infection test, urine Test, and medical examination by interview.



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

Data Set Link: (https://www.kaggle.com/datasets/mathchi/diabetes-data-set)

ABSTRACTION FOR PROBLEM DEFINITION AND DESIGN THINKING:

- ✓ The economic burden of Type 2 Diabetes (T2D) on society has Increased over time. Early prediction of diabetes and predia-Betes can reduce treatment cost and improve intervention. The Development of (pre)diabetes is associated with various health Conditions that can be monitored by routine health checkups.
- ✓ This study aimed to develop amachine learning-based model For predicting (pre)diabetes. Our frameworks were based on 22,722 patient samples collected from 2013 to 2020 in ageneral Hospital in Korea. The disease progression was divided into Three categories based on fasting blood glucose: normal, pre-Diabetes, and T2D. The risk factors at each stage were identified And compared. Based on the area under the curve, the support Vector machine appeared to have optimal performance.
- ✓ At the Normal and prediabetes stages, fasting blood glucose and HbA1c are prevalent risk features for the suggested models. Interestingly, HbA1c had the highest odds ratio among the Features even in the normal stage (FBG is less than 100).
- ✓ In Addition, factors related to liver function, such as gamma- Glutamyl transpeptidase can be used to predict progression From normal to prediabetes, while factors related to renal func-Tion, such as blood urea nitrogen and creatinine, are prediction Factors of T2D development.

DATA COLLECTION FOR AI I BASED DIABETES PREDUCTION SYSTEM:

Collecting relevant data for an AI-based diabetes prediction system is a crucial step in developing an accurate and effective model. The data you collect will serve as the foundation for training and testing your predictive model. Here are steps to guide you through the data collection process:

A. Define Data Requirements:

Before collecting data, clearly define the types of data you need for your diabetes prediction system. This should include various categories of information, such as:

- ✓ □ Patient demographics (e.g., age, gender, ethnicity)
- ✓ □ Clinical measurements (e.g., glucose levels, blood pressure, BMI)
- ✓ □ Lifestyle factors (e.g., physical activity, diet)
- ✓ □ Medical history (e.g., family history of diabetes, previous diagnoses)
- ✓ □ Medication usage (if applicable)
- ✓ □ Laboratory test results (e.g., HbA1c, lipid profiles)

□ Other relevant health metrics,

B.Identify Data Sources:

Determine where you will obtain the required data. Potential data sources include:

- ✓ **Electronic Health Records (EHRs):** Hospitals and healthcare providers often maintain EHR systems that contain comprehensive patient data.
- ✓ □ **Clinical databases:** Publicly available clinical databases like the National Health and Nutrition Examination Survey (NHANES) may provide relevant data.
- ✓ □ **Wearable Devices:** If collecting real-time health data, consider integrating with wearable devices that monitor glucose levels, activity, or other relevant metrics.
- ✓ □ **Surveys and Questionnaires:** Create surveys to gather additional information from patients regarding lifestyle and dietary habits.
- ✓ □ **Research Studies:** Collaborate with research institutions or access existing research data related to diabetes.

1) DATA COLLECTION METHODS:

Depending on the data source, you may use various methods to collect the data:

- ✓ □ **Extract Data from EHRs:** Work with healthcare institutions to extract relevant patient records from their EHR systems.
- ✓ □ **Surveys and Questionnaires:** Create online or paper-based surveys and distribute them to patients or volunteers.
- ✓ □ **Wearable Devices:** If using wearable devices, set up data collection protocols and ensure devices are properly calibrated.
- ✓ □ **Data APIs:** If accessing public health datasets or APIs, use appropriate data extraction methods.

PROGRAM:

```
import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
import seaborn as sns

In [2]:
dataset=pd.read_csv("D:/dataset/diabetes.csv")

In [3]:
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.	0.167	21	0

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
						1			
4	0	137	40	35	168	43.1	2.288	33	1

2) DATA PREPROCESSING:

Once you've collected the data, you'll likely need to preprocess it to ensure it's clean and suitable for modeling. Data preprocessing steps may include handling missing values, outlier detection and removal, feature engineering, and data transformation.

PROGRAM:

IN[1]:

```
#check if null value is present

dataset.isnull().values.any()

>>>False
```

IN[2]:

```
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
```

DATA PREPROCESSING FOR AI I BASED DIABETES PREDUCTION SYSTEM:

Data preprocessing is a critical step in preparing the collected data for use in an AI-based diabetes prediction system. Proper data preprocessing ensures that the data is clean, consistent, and ready for model training and evaluation. Here are the key data preprocessing steps for an AI-based diabetes prediction system:

A) Data Cleaning:

□ Handle Missing Values: Identify and handle missing data points. Depending on the amount of missing data, you can either impute missing values (e.g., with mean or median values) or remove rows or columns with excessive missing data.

□ Outlier Detection: Identify and handle outliers in the data. Outliers can significantly affect model performance. You can choose to remove, transform, or cap outliers based on domain knowledge.

B)Data Transformation:

- ✓ □ **Feature Encoding:** Convert categorical variables (if any) into numerical format. You can use techniques like one-hot encoding or label encoding.
- ✓ □ **Scaling and Normalization:** Normalize numerical features to a common scale, such as Min-Max scaling (scaling features to a range between 0 and 1) or standardization (scaling features to have mean 0 and standard deviation 1).
- ✓ □ **Feature Engineering:** Create new features that may be informative for diabetes prediction. For example, you can calculate the body mass index (BMI) from height and weight data.
- ✓ □ **Handling Imbalanced Data:** If your dataset has an imbalance between the number of diabetic and non-diabetic cases, consider techniques like oversampling, undersampling, or using synthetic data generation methods.

3) FEATURE SELECTION FOR AI I BASED DIABETES PREDUCTION SYSTEM:

Feature selection is a crucial step in building an AI-based diabetes prediction system. Selecting the most relevant features (variables) helps improve model accuracy, reduces overfitting, and simplifies the model. Here are steps and techniques for feature selection in your diabetes prediction system:

1. Initial Feature Exploration:

 Start by examining the dataset and getting an initial sense of the features. Look at data summaries, visualizations, and correlation matrices to identify potential patterns and relationships.

2. Domain Knowledge:

 Consult with domain experts, such as healthcare professionals, to gain insights into which features are likely to be the most relevant for diabetes prediction. Their expertise can guide your feature selection process.

3. Univariate Feature Selection:

 Use statistical tests to assess the relationship between individual features and the target variable (diabetes status). Common techniques include:

 Chi-Square Test: For categorical features.

 ANOVA F-Test: For numerical features.

4. Feature Importance Scores:

 If you plan to use machine learning algorithms like decision trees or random forests, you can calculate feature importance scores. These scores indicate the contribution of each feature to the model's predictive power.

5. Correlation Analysis:

 Analyze the correlation between features. Features that are highly correlated with each other may not provide additional information and can be candidates for removal.

4) MODEL SELECTION FOR AI I BASED DIABETES PREDUCTION SYSTEM:

Selecting an appropriate model for your AI-based diabetes prediction system is a crucial step that can significantly impact the system's performance and accuracy. Model selection should consider factors such as the nature of the data, available computational resources, interpretability, and the specific requirements of your application. Here are some commonly used models for medical prediction tasks like diabetes prediction:

A) Logistic Regression:

□ Logistic regression is a simple yet effective model for binary classification tasks like diabetes prediction. It provides interpretable results and is less prone to overfitting.

PROGRAM:

In[1]:

```
#Logistic regression
y = dataset_new['Outcome']
X = dataset_new.drop('Outcome', axis=1)
```

In[2]:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, Y_train)
y_predict = model.predict(X_test)
```

```
        /opt/conda/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
        STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

        Increase the number of iterations (max_iter) or scale the data as shown in:
```

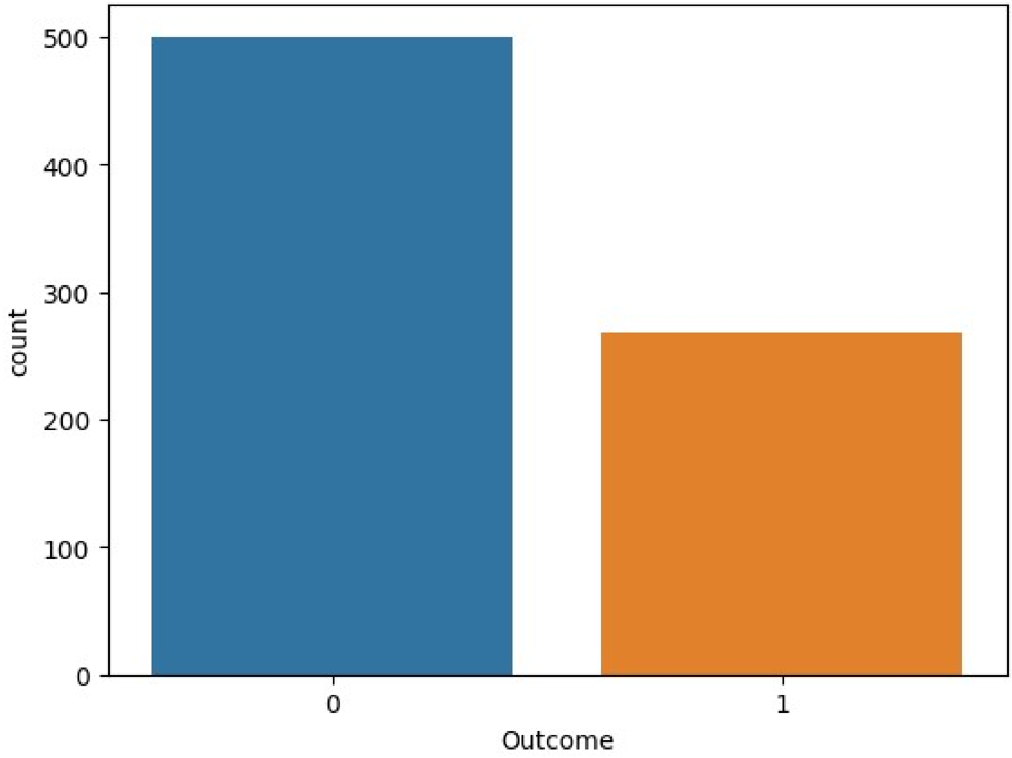
```
      https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
    n_iter_i = _check_optimize_result(
y_predict
array([1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0])
```

B) Decision Trees:

□ Decision trees can capture complex relationships in the data and are easy to visualize. However, they are prone to overfitting, which can be mitigated with techniques like pruning.

PROGRAM:

```
#data visualizationsns.countplot(x = 'Outcome',data = dataset)
Out[1]:
<Axes: xlabel='Outcome', ylabel='count'>
```



C) Random Forest:

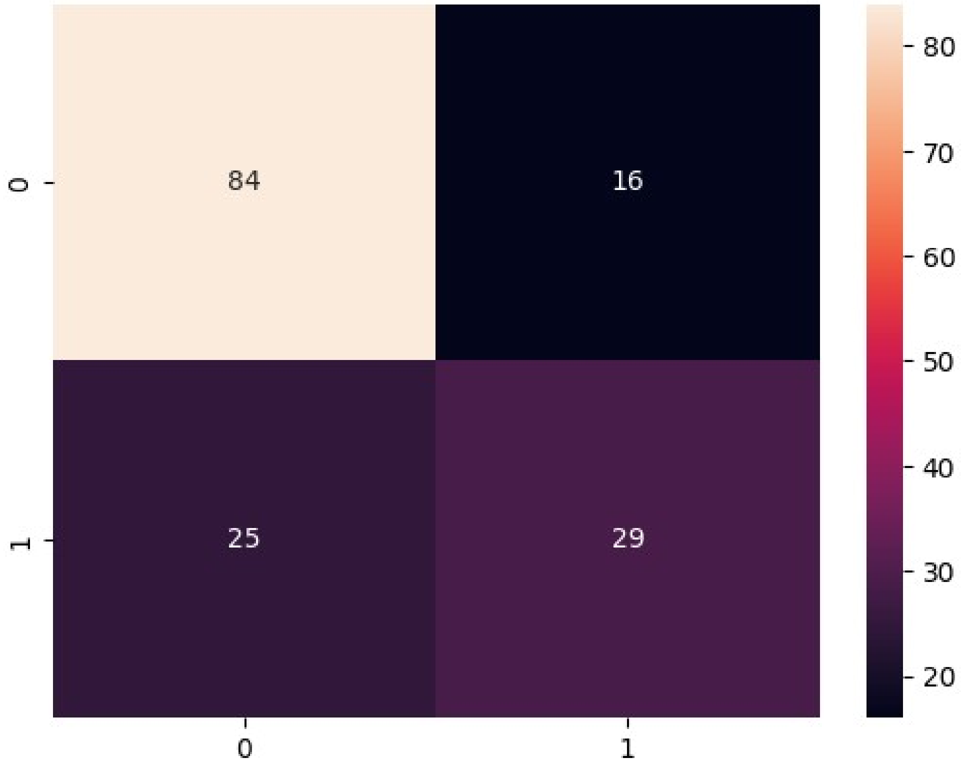
 Random forests are an ensemble of decision trees, offering improved generalization and robustness. They are well-suited for handling high-dimensional data and capturing non-linear relationships.

PROGRAM:

In[1]:

```
# Heatmap of Confusion matrix
sns.heatmap(pd.DataFrame(cm), annot=True)
```

Out[1]:



ACCURACY:

```
accuracy =accuracy_score(Y_test, y_predict)

accuracy

0.7337662337662337
```

#Example: Let's check whether the person have diabetes or not using some random values

```
y_predict = model.predict([[1,148,72,35,79.799,33.6,0.627,50]])

print(y_predict)

if y_predict==1:

   print("Diabetic")

else:

   print("Non Diabetic")
```

[1]:

Diabetic

ITERATIVE IMPROVEMENT FOR AI I BASED DIABETES PREDUCTION SYSTEM:

Iterative improvement is a vital aspect of developing and maintaining an AI-based diabetes prediction system. The process involves continuously refining the system, its models, and its performance to ensure that it remains accurate, up-to-date, and aligned with the evolving needs of healthcare providers and patients. Here are steps to guide you in implementing iterative improvement for your diabetes prediction system:

1. Continuous Data Collection and Update:

□ Maintain a mechanism for collecting new data, especially if your system relies on real-world patient data. Ensure that the system remains current with the latest patient records and health measurements.

2. Model Retraining:

□ Periodically retrain your predictive models using the updated data. Machine learning models can degrade over time as data distributions change, and retraining helps maintain their accuracy.

3. Performance Monitoring:

□ Implement continuous monitoring of model performance in a production environment. Set up alerts to notify you of any significant changes in performance metrics or issues.

4. Feedback Mechanisms:

□ Establish feedback loops with healthcare professionals, users, and other stakeholders. Encourage them to provide feedback on the system's predictions and recommendations.

5. Ethical and Bias Monitoring:

□ Continuously monitor the system for biases and ethical considerations. Assess whether the system's predictions align with fairness and ethical guidelines, and take corrective actions if biases are detected.

CONCLUSION:

Contribution of the Explainable AI in Diabetes Prediction system makes it easy for the end-user to understand the AI systems' complex working. It provides a human-centered interface to the user. Explainability is a key to producing a transparent, proficient, and accurate AI system that can help the healthcare practitioner, patients, and researcher understand and use the system.