The Hong Kong Polytechnic University

Department of Electrical and Electronics Engineering

EIE4430 Honours Project

2024-2025 Semester 1

Student Name: Chan Hou Ting Constant (21034774d)

Project Title: **Machine learning model to predict the risk of diabetes**

Progress Report (1/9/2024)

Works did in past month

In the past month, I tried to train the machine learning model and implemented metrics to evaluate the performance of the model. The application I used was Jupyter Notebook. For the testing, I used Random Forest temporarily and did some adjustments on its hyperparameter. In the following months, I will try to implement other models and adjust the hyperparameter on Random Forest to get a better result.

Enviornment Setup & Display all data in dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.utils import resample
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load the dataset
df = pd.read_csv('C:/Users/user/FYP_DiabetePrediction/Dataset/diabetes.csv')


# Display the styled DataFrame
rows , col =  df.shape
print("Number of Rows :",rows)
print("Number of Columns :", col)
df
```

```
Number of Rows : 768
Number of Columns : 9
```

	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
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Display dataset information

```
<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
```

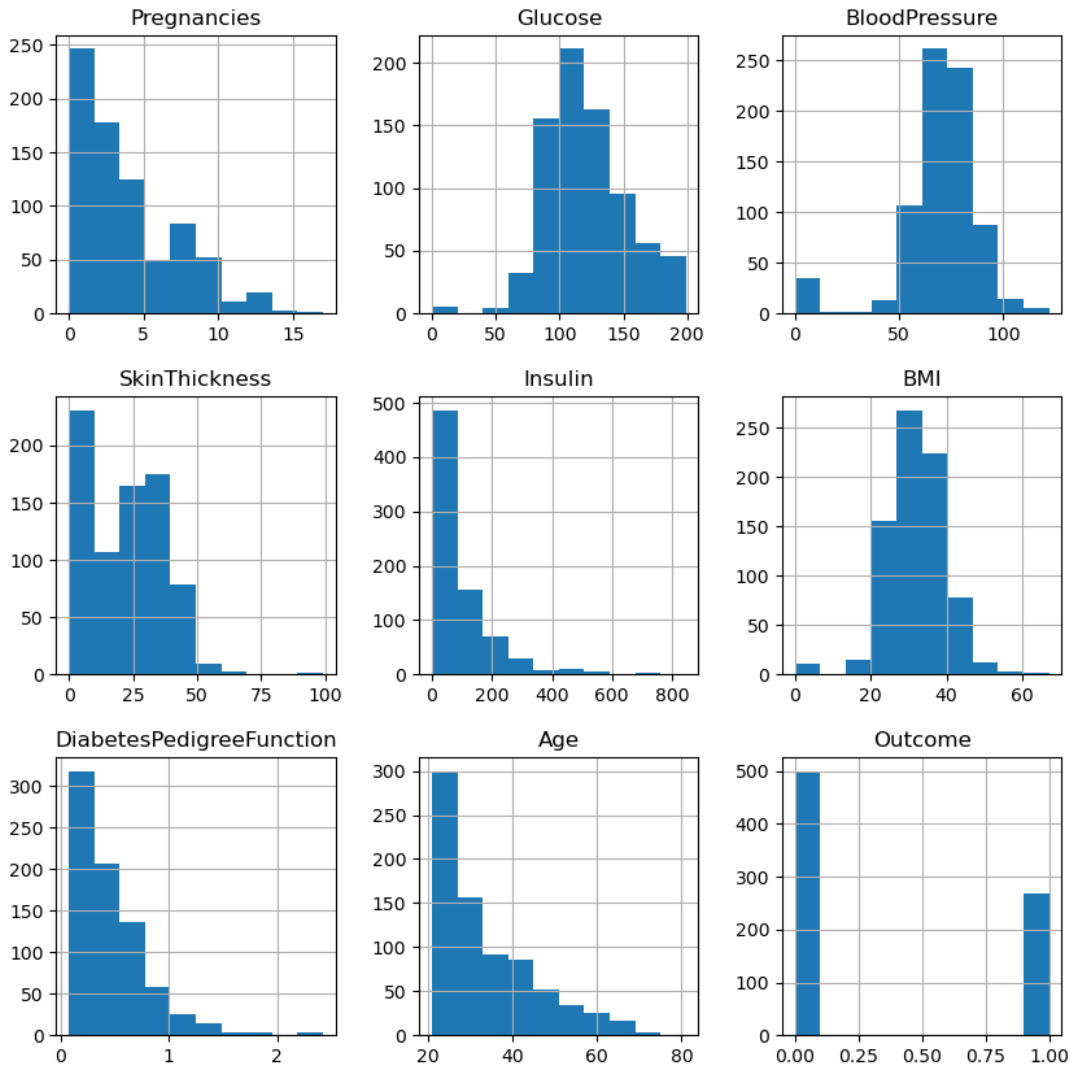
Null value check

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Display dataframe in Statistics form

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Display histogram of dataset



Display histogram of dataset

```
Pregnancies                 11.354056
Glucose                   1022.248314
BloodPressure              374.647271
SkinThickness              254.473245
Insulin                  13281.180078
BMI                         62.159984
DiabetesPedigreeFunction     0.109779
Age                        138.303046
Outcome                      0.227483
dtype: float64
```

Split the data

```
# Split the data
X = df.drop('Outcome',axis = 1)
y = df['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

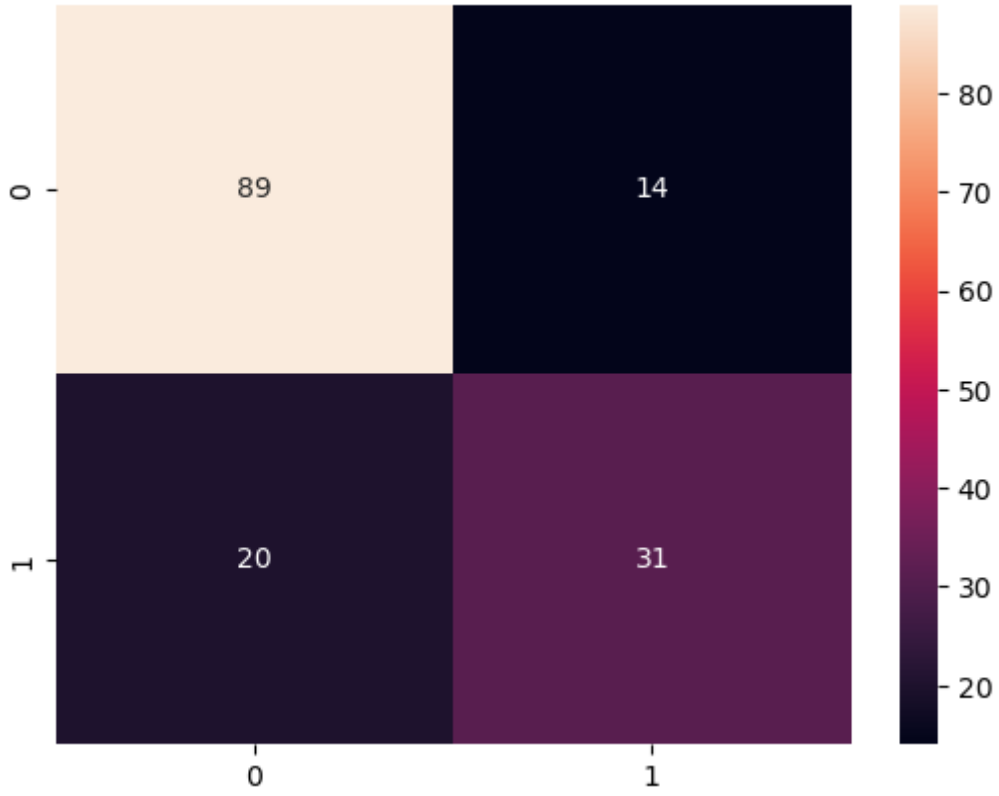
```
((614, 8), (154, 8), (614,), (154,))
```

Display Metrics and Confusion Matrix of Random Forest (n jobs = -1, random state = 42)

```
              precision    recall  f1-score   support

           0       0.82      0.86      0.84       103
           1       0.69      0.61      0.65        51

    accuracy                           0.78       154
   macro avg       0.75      0.74      0.74       154
weighted avg       0.77      0.78      0.78       154

[[89 14]
 [20 31]]
```

Display Confusion Matrix of Random Forest in heatmap

```
Accuracy =  0.7792207792207793
```



Tunning hyperparameter of Random Forest

```python
#Hyperparameter tunning
param_grid = {
    'n_estimators': [10, 20, 40],
    'min_samples_split': [2, 5, 10, 20, 40],
    'min_samples_leaf': [1, 2, 4, 8, 16],
    'max_depth': [10, 20, 40],
}

RF = RandomForestClassifier(n_jobs=-1, random_state=42)
grid_search = GridSearchCV(RF, param_grid, cv=5, n_jobs=-1, scoring='accuracy')
grid_search.fit(X_train, y_train)
RF_Best = grid_search.best_estimator_

RF_Best.fit(X_train, y_train)

# Evaluate the model on the training and validation data
train_accuracy = RF_Best.score(X_train, y_train)
val_accuracy = RF_Best.score(X_test, y_test)

# Print the results
print("Training Accuracy:", train_accuracy)
print("Validation Accuracy:", val_accuracy)
```

```
Training Accuracy: 0.8534201954397395
Validation Accuracy: 0.8116883116883117
```