GOVERNMENT COLLEGE OF ENGINEERING BARGUR
(AUTONOMOUS)

Project : Cloud Application Development

Project Statement: Machine Learning Model Deployment with IBM Cloud Watson Studio

Team members:

ANJALI B

MUTHULAKSHMI R

AKSHAYA N

MONIKA Y

Phase 3: Development Part 1

In this part you will begin building your project.

Start building the machine learning model using IBM Cloud Watson Studio.

Define the predictive use case (e.g., customer churn prediction) and select a relevant dataset.

Use IBM Cloud Watson Studio's tools to import the dataset, preprocess the data, select features, and train the machine learning model.

1.Import the libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.stats import norm, skew

from sklearn.preprocessing import RobustScaler, StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.metrics import roc_auc_score, roc_curve, classification_report
from warnings import filterwarnings
filterwarnings("ignore")
```

Exploratory Data

```
dataset = pd.read_csv("/kaggle/input/diabetes-dataset/diabetes.csv")
```

Information of Dataset

```
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
```

```
dataset.shape
```

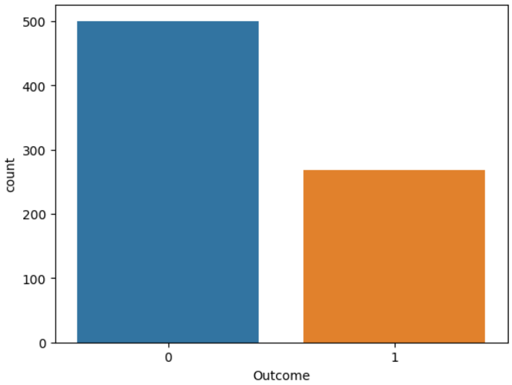
```
(768, 9)
```

Checking for missing values:

```
missing_values = dataset.isnull().sum()
print("Missing Values:")
print(missing_values)
Missing Values:
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

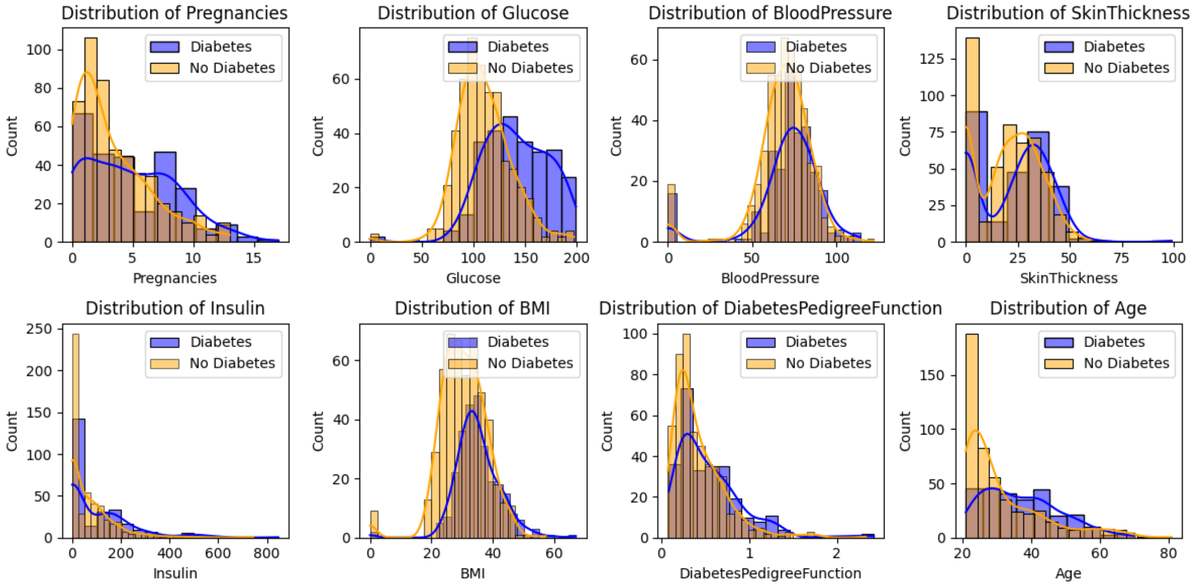
Diabetical and Non-diabetical Persons

```
dataset["Outcome"].value_counts()

#percentage distribution of the "Outcome"
print(100 * dataset["Outcome"].value_counts() / len(dataset))

with_diabetes = dataset['Outcome'].value_counts()[1]
without_diabetes = dataset['Outcome'].value_counts()[0]
print(f"Patients with Diabetes: {with_diabetes}\nPatients without Diabetes:
{without_diabetes}")

sns.countplot(x="Outcome", data=dataset)
Outcome
0    65.104167
1    34.895833
Name: count, dtype: float64
Patients with Diabetes: 268
Patients without Diabetes: 500
```



Visualizing the distribution of data in each column

```python
plt.figure(figsize=(12, 6))
for i, col in enumerate(dataset.columns[:-1]):
    plt.subplot(2, 4, i + 1)
    sns.histplot(dataset[dataset['Outcome'] == 1][col], kde=True,
label='Diabetes', color='blue')
    sns.histplot(dataset[dataset['Outcome'] == 0][col], kde=True,
label='No Diabetes', color='orange')
    plt.title(f"Distribution of {col}")
    plt.legend()
plt.tight_layout()
plt.show()
```



Splitting the Dataset into the Training set and Test Set

```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25, random_state=0)
```

```
print("X_train shape: ", X_train.shape)
print("X_test shape: ", X_test.shape)
print("y_train shape: ", y_train.shape)
print("y_test shape: ", y_test.shape)
```

```
X_train shape:  (576, 8)
X_test shape:  (192, 8)
y_train shape:  (576,)
y_test shape:  (192,)
```