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**}\n**Patients without Diabetes: **{**without\_diabetes**}**")

sns.countplot(x="Outcome", data=dataset)

Outcome

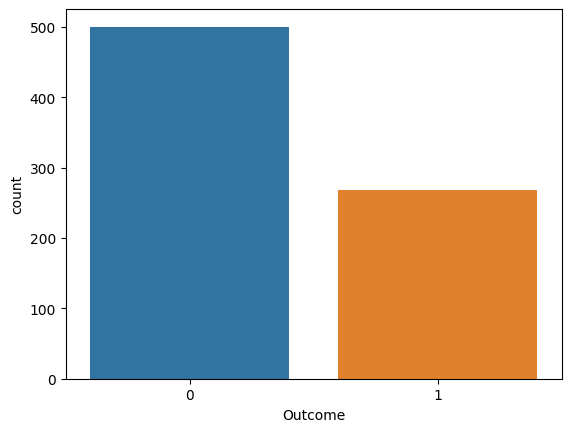
0 65.104167

1 34.895833

Name: count, dtype: float64

Patients with Diabetes: 268

Patients without Diabetes: 500



## **Visualizing the distribution of data in each column**

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')

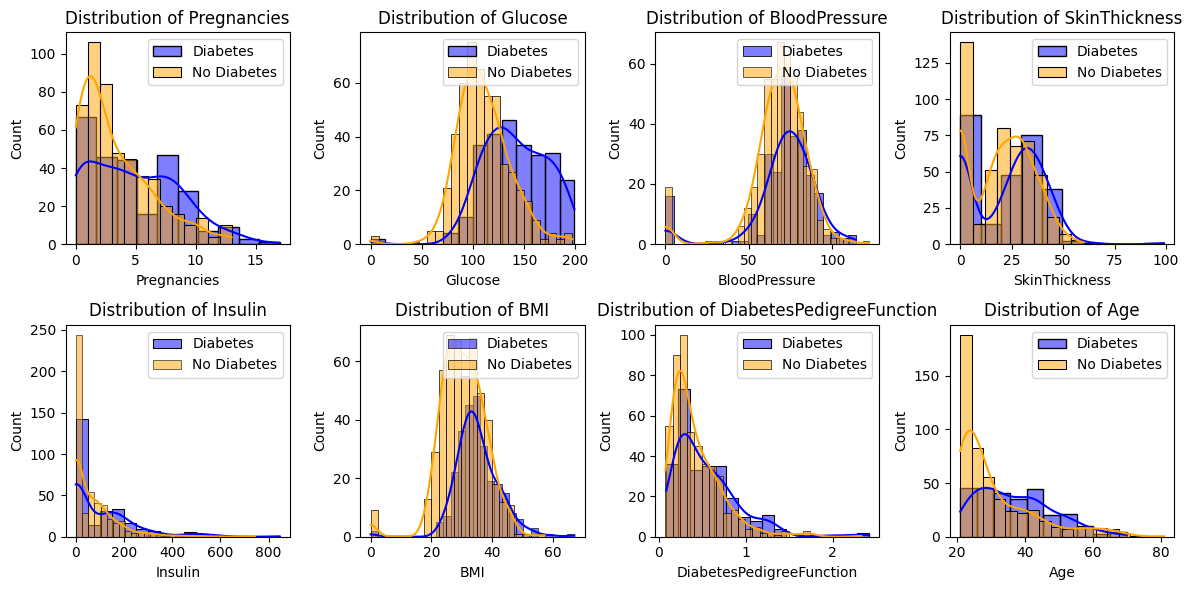
sns.histplot(dataset[dataset['Outcome'] == 0][col], kde=True, label='No Diabetes', color='orange')

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,)