

```

import pandas as pd
import numpy as np
import matplotlib as plt

df=pd.read_csv("diabetes.csv")

df

```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI |
|-----|-------------|---------|---------------|---------------|---------|------|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 |
| .. | ... | ... | ... | ... | ... | ... |
| 763 | 10 | 101 | 76 | 48 | 180 | 32.9 |
| 764 | 2 | 122 | 70 | 27 | 0 | 36.8 |
| 765 | 5 | 121 | 72 | 23 | 112 | 26.2 |
| 766 | 1 | 126 | 60 | 0 | 0 | 30.1 |
| 767 | 1 | 93 | 70 | 31 | 0 | 30.4 |

| | DiabetesPedigreeFunction | Age | Outcome |
|-----|--------------------------|-----|---------|
| 0 | 0.627 | 50 | 1 |
| 1 | 0.351 | 31 | 0 |
| 2 | 0.672 | 32 | 1 |
| 3 | 0.167 | 21 | 0 |
| 4 | 2.288 | 33 | 1 |
| .. | ... | ... | ... |
| 763 | 0.171 | 63 | 0 |
| 764 | 0.340 | 27 | 0 |
| 765 | 0.245 | 30 | 0 |
| 766 | 0.349 | 47 | 1 |
| 767 | 0.315 | 23 | 0 |

```

[768 rows x 9 columns]

df.columns

Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin',

```

```
'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],  
dtype='object')
```

```
df.isnull()
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin |
|-------|-------------|---------|---------------|---------------|---------|
| BMI \ | | | | | |
| 0 | False | False | False | False | False |
| False | | | | | |
| 1 | False | False | False | False | False |
| False | | | | | |
| 2 | False | False | False | False | False |
| False | | | | | |
| 3 | False | False | False | False | False |
| False | | | | | |
| 4 | False | False | False | False | False |
| False | | | | | |
| .. | ... | ... | ... | ... | ... |
| . | | | | | |
| 763 | False | False | False | False | False |
| False | | | | | |
| 764 | False | False | False | False | False |
| False | | | | | |
| 765 | False | False | False | False | False |
| False | | | | | |
| 766 | False | False | False | False | False |
| False | | | | | |
| 767 | False | False | False | False | False |
| False | | | | | |

| | DiabetesPedigreeFunction | Age | Outcome |
|-----|--------------------------|-------|---------|
| 0 | False | False | False |
| 1 | False | False | False |
| 2 | False | False | False |
| 3 | False | False | False |
| 4 | False | False | False |
| .. | ... | ... | ... |
| 763 | False | False | False |
| 764 | False | False | False |
| 765 | False | False | False |
| 766 | False | False | False |
| 767 | False | False | False |

```
[768 rows x 9 columns]
```

```
x = df.drop(['Outcome'], axis=1)  
y = df['Outcome']
```

```

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x, y,
test_size=0.4,random_state=10)

from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit(X_train, Y_train)

GaussianNB()

Y_pred = gaussian.predict(X_test)

from sklearn.metrics import accuracy_score, precision_score,
recall_score

accuracy = accuracy_score(Y_test,Y_pred)
precision =precision_score(Y_test, Y_pred,average='micro')
recall = recall_score(Y_test, Y_pred,average='micro')

from sklearn.metrics import
precision_score,confusion_matrix,accuracy_score,recall_score
cm = confusion_matrix(Y_test, Y_pred)

cm =confusion_matrix(Y_test,Y_pred)
print("ConfusionMatrix:\n",cm)

ConfusionMatrix:
[[166  35]
 [ 47  60]]

print("Accuracy:", accuracy_score(Y_test, Y_pred))
print("Precision:", precision_score(Y_test, Y_pred,
average='weighted'))
print("Recall:", recall_score(Y_test, Y_pred, average='weighted'))

Accuracy: 0.7337662337662337
Precision: 0.7280092035466387
Recall: 0.7337662337662337

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Generate confusion matrix
cm = confusion_matrix(Y_test, Y_pred)
# Correct class labels
labels = ['No Diabetes', 'Diabetes'] # Use appropriate labels
# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Greens',
xticklabels=labels,yticklabels=labels)
plt.xlabel('Predicted Labels')

```

```
plt.ylabel('True Labels')  
plt.title('Confusion Matrix')  
plt.show()
```

