Implement Naive Bayes algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
df = pd.read_csv("diabetes.csv")
```

In [ ]:
```
df.head()
```

In [ ]:
```
df.describe().T
```

In [ ]:
```
df.isnull().sum()
```

In [ ]:
```
df.head()
```

In [ ]:
```
X = df.drop('Outcome', axis = 1)
X.head()
```

In [ ]:
```
Y = df['Outcome']
Y.head()
```

In [ ]:
```
cat_list = X.columns
cat_list
```

In [ ]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size = 0.3,
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train1 = sc.fit_transform(x_train)
x_test1 = sc.fit_transform(x_test)
```

In [ ]:
```
from sklearn.naive_bayes import GaussianNB
gn = GaussianNB().fit(x_train, y_train)
```

In [ ]:
```
y_pred = gn.predict(x_test)
y_pred
```

In [ ]:
```
from sklearn.metrics import classification_report, confusion_matrix
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
confusion_matrix(y_test, y_pred)
```

In [ ]:
```
tn, fp, fn, tp
```

In [ ]:
```
print("Accuracy ", (tp+tn)/(tn+tp+fn+fp))
print("Error Rate ", (fp+fn)/(tn+tp+fn+fp))
print("Precision ", (tp)/(tp+fp))
print("Recall ", (tp)/(tp+fn))
```

In [ ]:
```
print(classification_report(y_test, y_pred))
```