

```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

Loading the data for the breast cancer dataset

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data"
column_names = ["ID", "Clump Thickness", "Uniformity of Cell Size", "Uniformity of Cell Shape",
                 "Marginal Adhesion", "Single Epithelial Cell Size", "Bare Nuclei",
                 "Bland Chromatin", "Normal Nucleoli", "Mitoses", "Class"]
cancerData = pd.read_csv(url, names=column_names)
```

```
# cleaning and replacing the null values
cancerData.replace("?", np.nan, inplace=True)
cancerData.drop("ID", axis=1, inplace=True)
```

```
# filling missing values
cancerData = cancerData.apply(pd.to_numeric)
cancerData.fillna(cancerData.mean(), inplace=True)
```

```
# getting the x and y axis
X = cancerData.drop("Class", axis=1)
y = cancerData["Class"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

```
▼ KNeighborsClassifier
KNeighborsClassifier()
```

Prediction on the dataset

```
y_pred = knn.predict(X_test)
```

Evaluating the printing the model

```
dataReport = classification_report(y_test, y_pred, target_names=["Benign", "Malignant"])
accuracyOfData = accuracy_score(y_test, y_pred)
```

```
print("Accuracy of the data :", accuracyOfData)
print("\nClassification Report for the data : \n", dataReport)
```

```
Accuracy of the data : 0.9666666666666667
```

```
Classification Report for the data :
              precision    recall  f1-score   support

   Benign         0.97       0.99       0.98         143
  Malignant         0.97       0.93       0.95          67

   accuracy                   0.97         210
  macro avg         0.97       0.96       0.96         210
 weighted avg         0.97       0.97       0.97         210
```

