Assignment – 10

B.Rithwik

2303A52330

Batch – 35

Question – 1

Loading the Data and Finding Features and Target Variables

```
import pandas as pd
from google.colab import drive

import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler

data = pd.read_csv("/content/drive/MyDrive/SML Dataset/breast_cancer_survival.csv")

data = data.dropna()

X = data.iloc[:, :-1]
y = data.iloc[:, -1]

for column in X.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    X[column] = le.fit_transform(X[column])

print("Features:", X.columns)
print("Target variable:", y.name)
```

OUTPUT –

```
Features: Index(['Age', 'Gender', 'Protein1', 'Protein2', 'Protein3', 'Protein4',
       'Tumour_Stage', 'Histology', 'ER status', 'PR status', 'HER2 status',
       'Surgery_type', 'Date_of_Surgery', 'Date_of_Last_Visit'],
      dtype='object')
Target variable: Patient_Status
```

Train KNN Model and Find Accuracy with Different Test Sizes

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

data = pd.read_csv("/content/drive/MyDrive/SML Dataset/breast_cancer_survival.csv")
data = data.dropna()

X = data.iloc[:, :-1]
y = data.iloc[:, -1]

for column in X.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    X[column] = le.fit_transform(X[column])

scaler = StandardScaler()
test_sizes = [20, 25, 30, 35]

for test_size in test_sizes:
    split = test_size / 100.0
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=split, random_state=42)
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

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

    y_pred = knn.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    print(f"KNN Accuracy with test size {test_size}%: {accuracy:.2f}")
```

OUTPUT –

```
KNN Accuracy with test size 20%: 0.81
KNN Accuracy with test size 25%: 0.79
KNN Accuracy with test size 30%: 0.79
KNN Accuracy with test size 35%: 0.80
```

Question – 2

Implement SVM with Different Training and Testing Splits

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

for test_size in test_sizes:
    split = test_size / 100.0
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=split, random_state=42)
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    svm = SVC(kernel='linear', max_iter=1000)
    svm.fit(X_train, y_train)

    y_pred = svm.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    print(f"SVM Accuracy with test size {test_size}%: {accuracy:.2f}")
```

OUTPUT -

```
SVM Accuracy with test size 20%: 0.80
SVM Accuracy with test size 25%: 0.79
SVM Accuracy with test size 30%: 0.79
SVM Accuracy with test size 35%: 0.81
```