Assignment – 11

# B.Rithwik

# 2303A52330

# Batch – 35

**Question – 1**

**Read the Data with Pandas and Find Features and Target Variables**

import pandas as pd

from sklearn.preprocessing import LabelEncoder

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**

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score

from sklearn.preprocessing import StandardScaler

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

**Train SVM Model and Find Accuracy with Different Test Sizes**

from sklearn.svm import SVC

# Train SVM model and evaluate accuracy with different test sizes

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

**Train Logistic Regression Model and Find Accuracy with Different Test Sizes**

from sklearn.linear\_model import LogisticRegression

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)

log\_reg = LogisticRegression(max\_iter=1000)

log\_reg.fit(X\_train, y\_train)

y\_pred = log\_reg.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Logistic Regression Accuracy with test size {test\_size}%: {accuracy:.2f}")

**OUTPUT -**

Logistic Regression Accuracy with test size 20%: 0.78

Logistic Regression Accuracy with test size 25%: 0.78

Logistic Regression Accuracy with test size 30%: 0.78

Logistic Regression Accuracy with test size 35%: 0.81

**Dimensionality Reduction with PCA and Model Training**

from sklearn.decomposition import PCA

pca = PCA(n\_components=5)

X\_reduced = pca.fit\_transform(X)

for test\_size in test\_sizes:

split = test\_size / 100.0

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_reduced, y, test\_size=split, random\_state=42)

knn.fit(X\_train, y\_train)

y\_pred = knn.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"KNN with PCA Accuracy (test size {test\_size}%): {accuracy:.2f}")

svm.fit(X\_train, y\_train)

y\_pred = svm.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"SVM with PCA Accuracy (test size {test\_size}%): {accuracy:.2f}")

log\_reg.fit(X\_train, y\_train)

y\_pred = log\_reg.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Logistic Regression with PCA Accuracy (test size {test\_size}%): {accuracy:.2f}")

**OUTPUT -**

KNN with PCA Accuracy (test size 20%): 0.81

SVM with PCA Accuracy (test size 20%): 0.66

Logistic Regression with PCA Accuracy (test size 20%): 0.80

KNN with PCA Accuracy (test size 25%): 0.82

SVM with PCA Accuracy (test size 25%): 0.56

Logistic Regression with PCA Accuracy (test size 25%): 0.79

KNN with PCA Accuracy (test size 30%): 0.81

SVM with PCA Accuracy (test size 30%): 0.41

Logistic Regression with PCA Accuracy (test size 30%): 0.79

KNN with PCA Accuracy (test size 35%): 0.82

SVM with PCA Accuracy (test size 35%): 0.34

Logistic Regression with PCA Accuracy (test size 35%): 0.81