

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
```