**HW\_03\_KNN**

**Poorvi Raut -20009560**

The “breast cancer dataset” in CANVAS was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The features in the dataset, described below, have been categorized from 1 to 10. Use the knn methodology (k=3,5 and 10) to develop a classification model for the Diagnosis. Important: make sure your categories are represented by the “factor” data type in python and delete the rows with missing value. Use 30% test 70% training data. Features Domain -- -----------------------------------------

Sample code number id number

F1. Clump Thickness 1 - 10 F2.

Uniformity of Cell Size 1 - 10 F3.

Uniformity of Cell Shape 1 - 10 F4.

Marginal Adhesion 1 - 10 F5.

Single Epithelial Cell Size 1 - 10 F6.

Bare Nuclei 1 - 10 F7.

Bland Chromatin 1 - 10 F8.

Normal Nucleoli 1 - 10 F9.

Mitoses 1 - 10 Diagnosis Class: (2 for benign, 4 for malignant)

Below is screenshot of code in Python with necessary steps generated using ChatGpt.Ai

Text

Description automatically generated with low confidence

Text

Description automatically generated

Text

Description automatically generated

**After running the code on jupyter lab :**

import pandas as pd

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

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix

# Assuming your dataset is a CSV file named "breast\_cancer\_data.csv"

data = pd.read\_csv("breast-cancer-wisconsin.csv")

# Convert all columns (except 'Sample' and 'Class') to numeric

columns\_to\_convert = [col for col in data.columns if col not in ['Sample', 'Class']]

data[columns\_to\_convert] = data[columns\_to\_convert].apply(pd.to\_numeric, errors='coerce')

# Remove rows with missing values

data = data.dropna()

# Split the dataset into training (70%) and testing (30%) sets

train\_data, test\_data = train\_test\_split(data, test\_size=0.3, random\_state=123)

# Remove the Sample code number (id) from the dataset

train\_data = train\_data.drop(columns=['Sample'])

test\_data = test\_data.drop(columns=['Sample'])

train\_labels = train\_data.pop('Class')

test\_labels = test\_data.pop('Class')

k\_values = [3, 5, 10]

for k in k\_values:

    knn = KNeighborsClassifier(n\_neighbors=k)

    knn.fit(train\_data, train\_labels)

    predicted\_class = knn.predict(test\_data)

    accuracy = accuracy\_score(test\_labels, predicted\_class)

    confusion = confusion\_matrix(test\_labels, predicted\_class)

We get output as:

Accuracy for k = 3: 0.9853658536585366

Confusion Matrix:

[[131 1]

[ 2 71]]

Accuracy for k = 5: 0.9853658536585366

Confusion Matrix:

[[130 2]

[ 1 72]]

Accuracy for k = 10: 0.9804878048780488

Confusion Matrix:

[[131 1]

[ 3 70]]