**Name: Jigar Siddhpura SAPID:** 60004200155

**DIV: C/C2 Branch:** Computer Engineering

ML - Experiment 6

### 1st DATASET

import numpy as np

# Calculate Euclidean distance between two points

def euclidean\_distance(p1, p2):

    return np.sqrt(np.sum((p1 - p2)\*\*2))

# Define the dataset, target values, and test point

dataset1 = np.array([[5, 45], [5.11, 26], [5.6, 30], [5.9, 34], [4.8, 40], [5.8, 36], [5.3, 19], [5.8, 28], [5.5, 23], [5.6, 32]])

target1 = np.array([77, 47, 55, 59, 72, 60, 40, 60, 45, 58])

test1 = np.array([5.5, 38])

# Set the value of K

k = 3

# Calculate distances to all points in the dataset

distances = np.array([euclidean\_distance(test1, d) for d in dataset1])

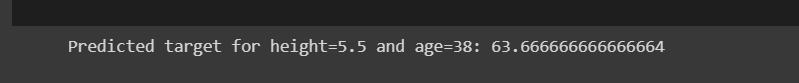
# Get indices of K nearest neighbors

nearest\_indices = np.argsort(distances)[:k]

# Predict the target value based on the average of K nearest neighbors

predicted\_target = np.mean(target1[nearest\_indices])

print("Predicted target for height=5.5 and age=38:", predicted\_target)



"""### 2nd DATASET"""

import math

def euclidean\_distance(p1, p2):

    return math.sqrt((p1[0] - p2[0])\*\*2 + (p1[1] - p2[1])\*\*2)

# Define the dataset

data = [

    [167, 51, 'under'],

    [182, 62, 'normal'],

    [176, 69, 'normal'],

    [173, 64, 'normal'],

    [172, 65, 'normal'],

    [174, 56, 'under'],

    [169, 58, 'normal'],

    [173, 57, 'normal'],

    [170, 55, 'normal']

]

point = [170, 57]

k = 3

distances = [(euclidean\_distance(point, d[:2]), d) for d in data]

# Sort distances and get the K nearest neighbors

nearest\_neighbors = sorted(distances)[:k]

print(nearest\_neighbors)

# Count occurrences of categories in nearest neighbors

category\_count = {}

for \_, neighbor in nearest\_neighbors:

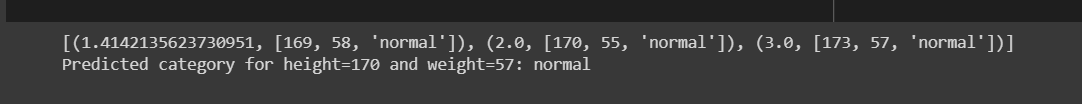
    category = neighbor[2]

    category\_count[category] = category\_count.get(category, 0) + 1

# Predict the category based on majority vote

predicted\_category = max(category\_count, key=category\_count.get)

print("Predicted category for height=170 and weight=57:", predicted\_category)



"""### 3rd DATASET"""

from google.colab import drive

drive.mount('/content/gdrive')

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.neighbors import KNeighborsClassifier

plt.style.use('ggplot')

df = pd.read\_csv('/content/gdrive/MyDrive/ML/diabetes.csv')

X = df.drop('Outcome',axis=1).values

y = df['Outcome'].values

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.4,random\_state=42, stratify=y)

neighbors = np.arange(1,6)

train\_accuracy =np.empty(len(neighbors))

test\_accuracy = np.empty(len(neighbors))

for i,k in enumerate(neighbors):

    #Setup a knn classifier with k neighbors

    knn = KNeighborsClassifier(n\_neighbors=k)

    #Fit the model

    knn.fit(X\_train, y\_train)

    #Compute accuracy on the training set

    train\_accuracy[i] = knn.score(X\_train, y\_train)

    #Compute accuracy on the test set

test\_accuracy[i] = knn.score(X\_test, y\_test)

plt.title('kNN Varying number of neighbors')

plt.plot(neighbors, test\_accuracy, label='Testing Accuracy')

plt.plot(neighbors, train\_accuracy, label='Training accuracy')

plt.legend()

plt.xlabel('Number of neighbors')

plt.ylabel('Accuracy')

plt.show()

