Name: Jigar Siddhpura **SAPID:** 60004200155

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

ML - Experiment 6

1st DATASET

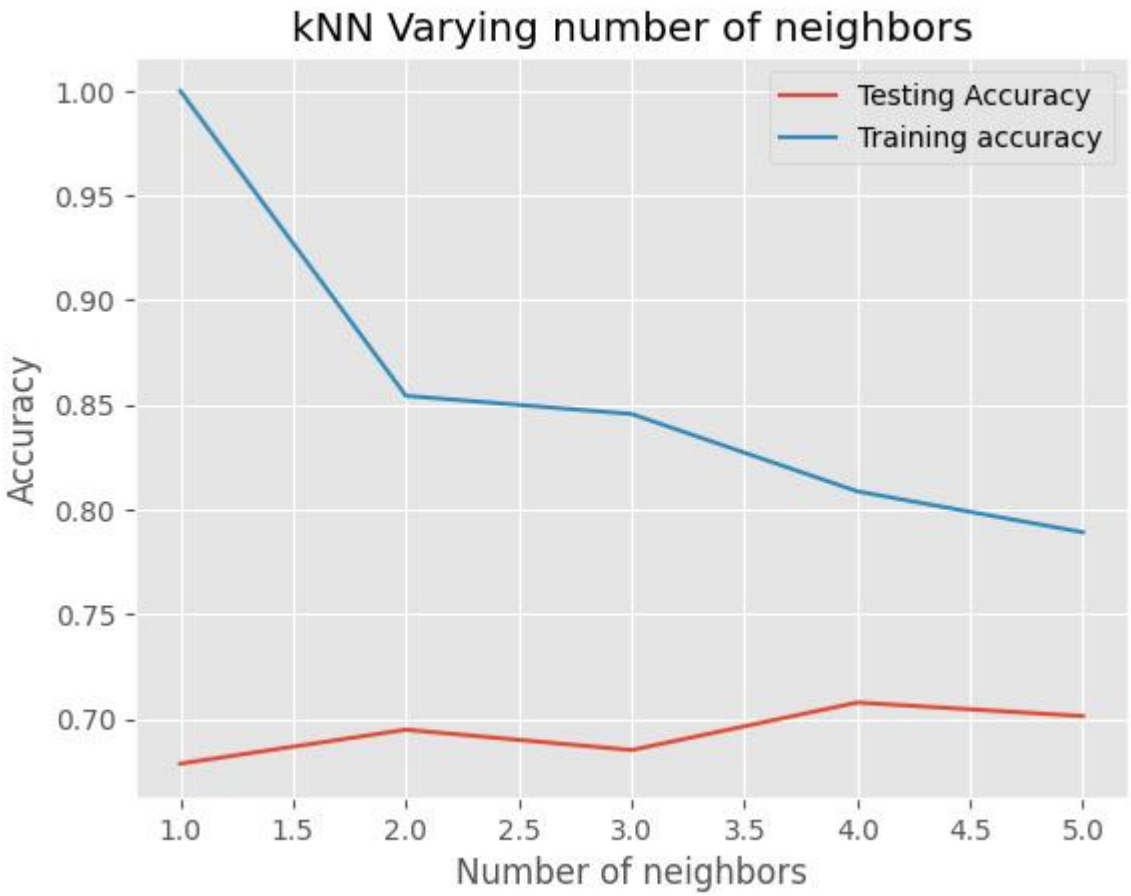
```
import numpy as np

# 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)
```

```
Predicted target for height=5.5 and age=38: 63.666666666666664
```

```
"""### 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)
```

```
[(1.4142135623730951, [169, 58, 'normal']), (2.0, [170, 55, 'normal']), (3.0, [173, 57, 'normal'])]
Predicted category for height=170 and weight=57: normal
```

```
"""### 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()
```


kNN Varying number of neighbors
Testing Accuracy
Training accuracy
Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of neighbors