B7- PRIVACY PRESERVING LOCATION DATA PUBLISHING A MACHINE LEARNING APPROACH

In [1]:
```
!pip install Bio
```

```
Requirement already satisfied: Bio in c:\users\hi\anaconda3\lib\site-packages (1.
3.9)
Requirement already satisfied: tqdm in c:\users\hi\anaconda3\lib\site-packages (fr
om Bio) (4.64.0)
Requirement already satisfied: requests in c:\users\hi\anaconda3\lib\site-packages
(from Bio) (2.27.1)
Requirement already satisfied: biopython>=1.79 in c:\users\hi\anaconda3\lib\site-p
ackages (from Bio) (1.79)
Requirement already satisfied: mygene in c:\users\hi\anaconda3\lib\site-packages
(from Bio) (3.2.2)
Requirement already satisfied: numpy in c:\users\hi\anaconda3\lib\site-packages (f
rom biopython>=1.79->Bio) (1.21.5)
Requirement already satisfied: biothings-client>=0.2.6 in c:\users\hi\anaconda3\li
b\site-packages (from mygene->Bio) (0.2.6)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\hi\anaconda3
\lib\site-packages (from requests->Bio) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hi\anaconda3\lib\sit
e-packages (from requests->Bio) (2021.10.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\hi\anaconda3\lib
\site-packages (from requests->Bio) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hi\anaconda3\lib\site-pack
ages (from requests->Bio) (3.3)
Requirement already satisfied: colorama in c:\users\hi\anaconda3\lib\site-packages
(from tqdm->Bio) (0.4.4)
```

In [2]:
```
from Bio import pairwise2
from Bio.Seq import Seq
import numpy as np
import pandas as pd
```

In [ ]:
```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import numpy as np
from tkinter import simpledialog
from tkinter import filedialog

from Bio import pairwise2
from Bio.Seq import Seq
import numpy as np
import pandas as pd
from Bio.Seq import Seq
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score
import random
import matplotlib.pyplot as plt


main = tkinter.Tk()
main.title("Privacy Preserving Location Data Publishing: A Machine Learning Approac
```

```
main.geometry("1300x1200")

global filename
global dataset
global train
global cluster_labels
global kmeans_loss, heuristic_loss
trajectory_append = []
store_loss = []
sa_correct = 0

def uploadDataset():
    global filename
    global dataset
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="TaxiDataset")
    pathlabel.config(text=str(filename)+" loaded")
    dataset = pd.read_csv(filename,nrows=100)
    dataset['querydate']= pd.to_datetime(dataset['querydate'])
    text.insert(END,str(dataset.head()))


def processDataset():
    global train
    global dataset
    text.delete('1.0', END)
    dataset.fillna(0, inplace = True)
    train = dataset[['latitude','longitude']]
    text.insert(END,"Total records contains in dataset : "+str(train.shape[0])+"\n'
    text.insert(END,"\ndataset preprocessing completed\n")

def dynamicSA(src_lat,src_lon,cls_id):
    global sa_correct
    dups = []
    max1 = 0
    max2 = 0
    choosen_lat = 0
    choosen_lon = 0
    while len(dups) < 10:
        random_record_dataset = 0
        flag = True
        while flag:
            random_record_dataset = random.randint(0,(len(dataset)-1))
            if random_record_dataset not in dups:
                dups.append(random_record_dataset)
                flag = False
        des_lat = dataset[random_record_dataset,2]
        des_lon = dataset[random_record_dataset,3]
        seq1 = Seq(str(src_lat))
        seq2 = Seq(str(des_lat))
        seq3 = Seq(str(src_lon))
        seq4 = Seq(str(des_lon))
        alignments1 = pairwise2.align.globalxx(seq1, seq2)
        alignments2 = pairwise2.align.globalxx(seq3, seq4)
        for match in alignments1:
            score = match[2]
            if score > max1:
                max1 = score
                choosen_lat = des_lat
        for match in alignments2:
            score = match[2]
            if score > max2:
                max2 = score
                choosen_lon = des_lon
```

```
    cls = 0
    if max1 <= 5 and max2 <= 5:
        cls = 0
    else:
        cls = 1
    if cls == cls_id:
        sa_correct = sa_correct + 1
    print(str(sa_correct)+" "+str(cls_id)+" "+str(max1)+" "+str(max2))
    return str(choosen_lat)+","+str(choosen_lon),(max1+max2)/2


def runKmeansSA():
    text.delete('1.0', END)
    global trajectory_append
    global store_loss
    global train
    global dataset
    global cluster_labels
    global kmeans_loss
    global heuristic_loss
    global sa_correct
    sa_correct = 0
    trajectory_append.clear()
    store_loss.clear()
    kmeans = KMeans(n_clusters=2)
    kmeans.fit(train)
    predict = kmeans.predict(train)
    cluster_labels = kmeans.labels_
    for i in range(0,10):
        predict[i] = 3
    acc = accuracy_score(cluster_labels,predict)
    kmeans_loss = 1.0 - acc
    dataset['clusterID'] = cluster_labels
    dataset = dataset.values

    for i in range(len(cluster_labels)):
        src_lat = dataset[i,2]
        src_lon = dataset[i,3]
        cls_id = dataset[i,4]
        trajectory_value, trajectory_loss = dynamicSA(src_lat,src_lon,cls_id)
        trajectory_append.append(trajectory_value)
        store_loss.append(trajectory_loss)
        text.insert(END,"Processed Location Data : "+trajectory_value+" with loss
        text.update_idletasks()
    heuristic_loss = sa_correct / 100.0
    text.delete('1.0', END)
    text.insert(END,"KMEANS Loss on Dataset : "+str(kmeans_loss)+"\n\n")
    text.insert(END,"Heuristic Loss on Dataset : "+str(heuristic_loss)+"\n\n")


def dataGeneralization():
    text.delete('1.0', END)
    global trajectory_append
    global store_loss
    for i in range(len(trajectory_append)):
        arr = trajectory_append[i].split(",")
        lat = float(arr[0])
        lon = float(arr[1])
        lat = lat + store_loss[i]
        lon = lon + store_loss[i]
        text.insert(END,"Latitude After Generalization : "+str(lat)+" Longitude Af

def graph():
    global heuristic_loss
```

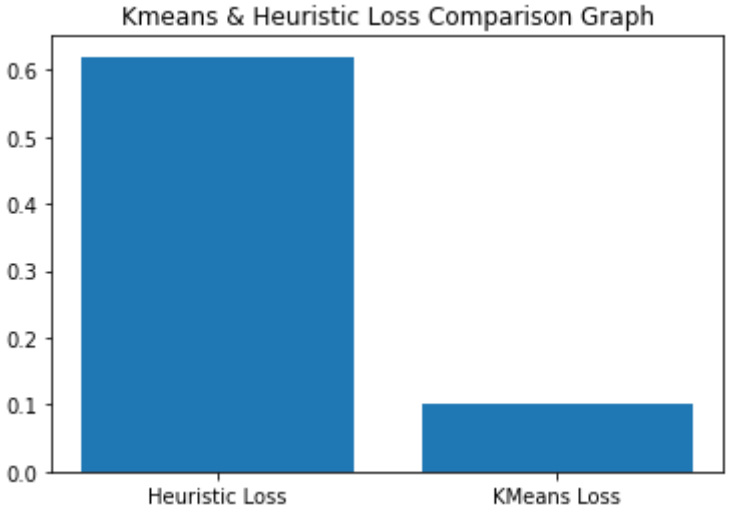
```
    global kmeans_loss
    height = [heuristic_loss,kmeans_loss]
    bars = ('Heuristic Loss','KMeans Loss')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.title("Kmeans & Heuristic Loss Comparison Graph")
    plt.show()

font = ('times', 16, 'bold')
title = Label(main, text='Privacy Preserving Location Data Publishing: A Machine Le
title.config(bg='black', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)


font1 = ('times', 15, 'bold')
uploadButton = Button(main, text="Upload Taxi Trajectory Dataset", command=uploadDa
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='DarkOrange1', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=50,y=150)


processButton = Button(main, text="Preprocess Dataset", command=processDataset)
processButton.place(x=50,y=200)
processButton.config(font=font1)

kmeansButton = Button(main, text="Run KMeans with DynamicSA Algorithm", command=ru
kmeansButton.place(x=50,y=250)
kmeansButton.config(font=font1)

generalButton = Button(main, text="Run Data Generalization Algorithm", command=data
generalButton.place(x=50,y=300)
generalButton.config(font=font1)

graphButton = Button(main, text="Loss Comparison Graph", command=graph)
graphButton.place(x=50,y=350)
graphButton.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=28,width=100)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=480,y=100)
text.config(font=font1)

main.config(bg='DarkOrange1')
main.mainloop()
```

0 0 6.0 5.0
0 0 9.0 8.0
0 0 6.0 6.0
0 0 6.0 6.0
0 0 9.0 8.0
0 0 6.0 5.0
0 0 7.0 8.0
0 0 6.0 6.0
0 0 6.0 6.0
0 0 8.0 7.0
0 0 9.0 8.0
0 0 8.0 7.0
0 0 8.0 7.0
0 0 7.0 6.0
0 0 7.0 8.0
0 0 7.0 6.0
0 0 7.0 6.0
0 0 8.0 7.0
0 0 6.0 6.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 7.0 6.0
0 0 9.0 8.0
0 0 9.0 8.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 6.0 7.0
0 0 6.0 6.0
0 0 7.0 6.0
0 0 7.0 8.0
0 0 8.0 7.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 6.0 6.0
0 0 7.0 7.0
0 0 7.0 7.0
1 1 7.0 5.0
2 1 7.0 6.0
3 1 7.0 6.0
4 1 7.0 6.0
5 1 7.0 6.0
6 1 8.0 7.0
7 1 9.0 8.0
8 1 8.0 7.0
9 1 8.0 7.0
10 1 8.0 6.0
11 1 8.0 6.0
12 1 8.0 6.0
13 1 7.0 7.0
14 1 8.0 6.0
15 1 9.0 8.0
16 1 7.0 6.0
17 1 7.0 6.0
18 1 8.0 6.0
19 1 7.0 5.0
20 1 8.0 6.0
21 1 7.0 6.0
22 1 7.0 6.0
23 1 8.0 5.0
24 1 9.0 8.0
25 1 9.0 8.0
26 1 7.0 6.0

```
27 1 7.0 6.0
28 1 6.0 6.0
29 1 7.0 6.0
30 1 7.0 5.0
31 1 8.0 6.0
32 1 9.0 8.0
33 1 7.0 6.0
34 1 8.0 7.0
35 1 7.0 6.0
36 1 7.0 5.0
37 1 7.0 6.0
38 1 8.0 6.0
39 1 9.0 7.0
40 1 7.0 6.0
41 1 7.0 6.0
42 1 8.0 6.0
43 1 8.0 6.0
44 1 7.0 6.0
45 1 7.0 6.0
46 1 8.0 6.0
47 1 9.0 8.0
48 1 7.0 7.0
49 1 8.0 6.0
50 1 8.0 7.0
51 1 9.0 7.0
52 1 7.0 6.0
53 1 9.0 8.0
54 1 7.0 6.0
55 1 7.0 5.0
56 1 7.0 5.0
57 1 6.0 5.0
58 1 7.0 6.0
59 1 8.0 8.0
60 1 8.0 8.0
61 1 8.0 8.0
62 1 9.0 8.0
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_7032\1412466078.py", line 48, in
processDataset
    dataset.fillna(0, inplace = True)
AttributeError: 'numpy.ndarray' object has no attribute 'fillna'
```

0 0 8.0 6.0
0 0 7.0 5.0
0 0 9.0 8.0
0 0 9.0 8.0
0 0 6.0 6.0
0 0 6.0 6.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 8.0 8.0
0 0 6.0 6.0
0 0 8.0 7.0
0 0 7.0 7.0
0 0 7.0 6.0
0 0 8.0 7.0
0 0 9.0 8.0
0 0 7.0 7.0
0 0 7.0 5.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 7.0 6.0
0 0 9.0 7.0
0 0 9.0 8.0
0 0 7.0 7.0
0 0 9.0 7.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 7.0 7.0
0 0 6.0 7.0
0 0 7.0 7.0
0 0 7.0 6.0
0 0 7.0 7.0
0 0 8.0 7.0
0 0 7.0 6.0
0 0 8.0 8.0
0 0 7.0 7.0
0 0 6.0 6.0
0 0 6.0 7.0
1 1 7.0 5.0
2 1 7.0 6.0
3 1 7.0 6.0
4 1 6.0 6.0
5 1 7.0 6.0
6 1 9.0 8.0
7 1 7.0 6.0
8 1 9.0 8.0
9 1 8.0 7.0
10 1 7.0 6.0
11 1 7.0 6.0
12 1 8.0 7.0
13 1 7.0 7.0
14 1 9.0 7.0
15 1 7.0 7.0
16 1 7.0 7.0
17 1 7.0 6.0
18 1 9.0 8.0
19 1 7.0 6.0
20 1 7.0 7.0
21 1 7.0 7.0
22 1 7.0 6.0
23 1 7.0 5.0
24 1 7.0 5.0
25 1 7.0 6.0
26 1 9.0 7.0

```
27 1 7.0 6.0
28 1 6.0 5.0
29 1 7.0 7.0
30 1 7.0 5.0
31 1 8.0 7.0
32 1 7.0 7.0
33 1 7.0 8.0
34 1 7.0 7.0
35 1 7.0 5.0
36 1 6.0 6.0
37 1 6.0 6.0
38 1 7.0 6.0
39 1 7.0 6.0
40 1 7.0 6.0
41 1 7.0 6.0
42 1 8.0 6.0
43 1 8.0 6.0
44 1 7.0 7.0
45 1 7.0 6.0
46 1 8.0 6.0
47 1 8.0 7.0
48 1 7.0 6.0
49 1 9.0 8.0
50 1 7.0 7.0
51 1 8.0 7.0
52 1 8.0 8.0
53 1 7.0 6.0
54 1 7.0 6.0
55 1 7.0 5.0
56 1 7.0 6.0
57 1 7.0 5.0
58 1 7.0 6.0
59 1 9.0 8.0
60 1 6.0 6.0
61 1 7.0 6.0
62 1 8.0 7.0
```



In [ ]:

In [ ]: