**Fall**

15

Machine Learning Assignment 4

Sundari Selvarajan

ss2738

08

**Fall**

**k-Means Clustering Algorithm**

**Location:**

/afs/cad.njit.edu/courses/ccs/f15/cs/675/001/ss2738/Assignment4

**File Name:**

k\_cluster.py

**Argument:**

<data file>

**Program:**

|  |
| --- |
| \_\_author\_\_ = **'Sundu'  import** sys **import** copy  **def** dist(point,mean):  distance1=0.0  **for** col **in** range(len(point)):  distance1+=(point[col]- mean[col])\*\*2  **return** distance1  *####################### Reading Data and Label Sets #######################################* input\_file = sys.argv[1]  **with** open(input\_file) **as** f:  data = f.read()  dataset = [] dataset1 = [] **for** line **in** data.split(**"\n"**):  **if** line:  dataset = []  a = line.split()  **for** i **in** range(len(a)):  dataset.append(float(a[i]))   dataset1.append(dataset)   *########################## Step 1 : Assign x[i]'s to classes with equal probablitity ##############*  k=input(**"Enter the total no. of clusters to be formed:"**)  count = len(dataset1) labelset = {} div=count/k q=0  **for** i **in** range(0,k):  **for** j **in** range(0,div):  **if** i **in** labelset:  labelset[i].append(q)  q=q+1  **else**:  labelset[i] = [q]  q=q+1  **while**(q<count):  labelset[i].append(q)  q=q+1  *####################Calculating means for every class#######################################* means = {} **for** k,v **in** labelset.items():  **for** data\_item **in** v:  **if** k **in** means:  t =list( zip(means[k],dataset1[data\_item]))  **for** col **in** range(len(dataset1[data\_item])):  means[k][col]=sum(t[col])  **else**:  means[k] = copy.copy(dataset1[data\_item]) **for** k,v **in** means.items():  count = len(labelset[k])  means[k] = [x/count **for** x **in** means[k]] *########################### Previous obj ################################################* prev\_obj = float(**"Inf"**) obj1=[] **for** x,y **in** labelset.items():  **for** t **in** y:  **for** i,m **in** means.items():  **if** x==i:  obj1.append(dist(dataset1[t],m))  obj=sum(obj1)  *#############################Step 2: Recompute clusters ##################################* a=0 **while**(prev\_obj-obj>=0.0001):  a=a+1  prev\_obj=obj  labelset2={}  obj1=[]  q=0  **for** x **in** dataset1:  distance=[]  **for** i,m **in** means.items():  distance.append(dist(x,m))  short\_dist=min(distance)  obj1.append(short\_dist)  t=0  **while**(short\_dist!=distance[t]):  t=t+1  **if** t **in** labelset2:  labelset2[t].append(q)  **else**:  labelset2[t]=[q]   q=q+1   means = {}  **for** k,v **in** labelset2.items():  **for** data\_item **in** v:  **if** k **in** means:  t =list( zip(means[k],dataset1[data\_item]))  **for** col **in** range(len(dataset1[data\_item])):  means[k][col]=sum(t[col])  **else**:  means[k] = copy.copy(dataset1[data\_item])   **for** k,v **in** means.items():  count = len(labelset2[k])  means[k] = [x/count **for** x **in** means[k]]   obj=sum(obj1)  **print**(labelset2) **print**() **print**(a,**"times iterated."**)  *#############################Writing predictions in a file## #################################* name=sys.argv[1]+**".predictions"** op = open(name,**"w"**) **for** data **in** range(len(dataset1)):   cls = None  **for** k,v **in** labelset2.items():  **for** t **in** v:  **if** data == t:  cls = int(k)+1  **break** op.write(str(cls)+**"\n"**) |