**Assignment 8 - Machine Learning (Cloud)**

**Name: Shweta Pathak**

**UTA ID: 1001154572**

**Net Id: ssp4572**

**Section: 1.00 pm to 3.00 pm**

**References:**

1. <http://pygments.org/demo/71873/>
2. <http://glowingpython.blogspot.com/2012/04/k-means-clustering-with-scipy.html>
3. <http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot>

**Code:**

**Main.py:**

**# Code to implement K mean clustering**

from bottle import route, run, template

from bottle import route, request, response, template, HTTPResponse

from bottle import static\_file

import matplotlib

import matplotlib.pyplot as plt

from pylab import plot,show

from numpy import vstack,array

from numpy.random import rand

from scipy.cluster.vq import kmeans,vq,whiten

import csv

from Tkinter import \*

import random

import json

# Fields in the data set

field\_names = ['Sepal\_length','Sepal\_width','Petal\_length','Petal\_width']

@route('/')

def index():

return template('user\_interface')

@route('/static/<filename>')

def server\_static(filename):

return static\_file(filename, root="static")

@route('/cluster', method='POST')

def clusterimage():

if request.headers.get('X-Requested-With') == 'XMLHttpRequest':

posted\_values = request.forms.dict

x\_param = posted\_values["x"][0]

y\_param = posted\_values["y"][0]

noCluster = posted\_values["cluster"][0]

clusters = int(noCluster)

data\_arr = get\_data\_arr(x\_param,y\_param)

data = vstack( data\_arr )

# Normalizing data

data = whiten(data)

# computing K-Means with clusters (clusters)

centroids, distortion = kmeans(data,clusters)

# assign each sample to a cluster

idx,\_ = vq(data,centroids)

total\_points =[]

for i in range(clusters):

result\_names = data[idx==i, 0]

print "================================="

print "Cluster " + str(i+1)

count = 0

for name in result\_names:

# print name

count +=1

print "Total cluster points: "

print(count)

total\_points.append(count)

# some plotting using numpy's logical indexing

print " ==================================="

print(centroids[0:])

centroid\_points = []

for row in centroids:

cent = []

cent.append(row[0])

cent.append(row[1])

centroid\_points.append(cent)

print(centroid\_points[0:])

colour = ["Db","Dr","Dg","Dy","Dk","Dc"]

fig = plt.figure()

for i in range(0,clusters):

for j in range(0,1):

plt.plot(data[idx==i,j], data[idx==i,j+1],colour[i])

plt.plot(centroids[:,0],centroids[:,1],'hm',markersize=12)

filename = "temp.png"

returnData = json.dumps({'filename':filename,'total\_points':total\_points,'centroids':centroid\_points})

fig.savefig("./static/"+filename)

resp = HTTPResponse(body=returnData,status=200)

return resp

else:

return 'This is a normal request'

def get\_data\_arr(x\_param, y\_param):

vector = []

with open('iris\_data.csv', 'rb') as f:

reader = csv.DictReader(f, fieldnames=field\_names)

for row in reader:

vector\_element = []

vector\_element.append(float(row[x\_param]))

vector\_element.append(float(row[y\_param]))

vector.append(vector\_element)

return vector

run(host="0.0.0.0",port=8080)

**User\_interface.tpl**

**# Web interface**

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>

<script>

$( document ).ready(function() {

$("#kmeans\_image").hide()

console.log( "ready!" );

$("#axes").hide();

$("#total").hide();

$("#total1").hide();

$("#cent").hide();

$("#centroid1").hide();

$( "#show" ).click(function() {

var xyObj = {};

xyObj["x"] = $( "#x" ).val();

xyObj["y"] = $( "#y" ).val();

xyObj["cluster"] = document.getElementById("no\_clusters").value;

console.log(xyObj);

var axis = $( "#x" ).val() + " " + "vs" + " " + $( "#y" ).val();

console.log(axis);

document.getElementById("axes").innerHTML = axis;

$.post("/cluster",xyObj,

function(data, status){

var json = JSON.parse(data);

console.log(json["filename"]);

console.log(json);

$("#kmeans\_image").attr("src", "./static/"+json["filename"])

$("#kmeans\_image").show()

document.getElementById("total").innerHTML = json["total\_points"];

document.getElementById("centroid1").innerHTML = json["centroids"];

$("#axes").show()

$("#total1").show()

$("#total").show()

$("#cent").show()

$("#centroid1").show()

});

});

});

</script>

</head>

<body>

<h1> Clustering </h1>

<br />

<b> Enter the number of clusters: </b>

<input type="text" id="no\_clusters"></input>

<br /><br />

Select the Parameter1 :

<select id="x">

<option value='Sepal\_length'>Sepal Length</option>

<option value='Sepal\_width'>Sepal Width</option>

<option value='Petal\_length'>Petal Length</option>

<option value='Petal\_width'>Petal Width</option>

</select>

<br><br>

Select the Parameter2:

<select id="y">

<option value='Sepal\_length'>Sepal Length</option>

<option value='Sepal\_width'>Sepal Width</option>

<option value='Petal\_length'>Petal Length</option>

<option value='Petal\_width'>Petal Width</option>

</select>

<br><br>

<input type="button" id="show" value="Get Cluster">

<br>

<img id="kmeans\_image" src='' ></img>

<h2 id="axes"></h2>

<h6 id ="total1">Total number of points in the clusters:</h6>

<h6 id="total"></h6>

<h6 id ="cent">Centroid points:</h6>

<h6 id="centroid1"></h6>

</body>

</html>

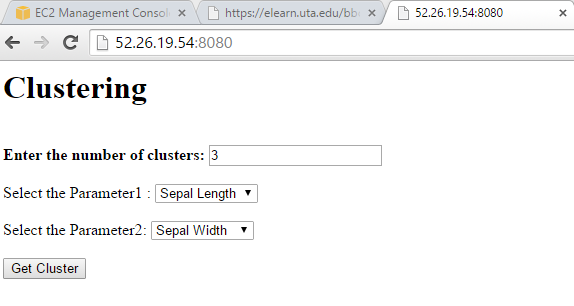
**Dataset used:**

**Iris\_data.csv: Contains the below sample data**

Sepal\_length Sepal\_width Petal\_length Petal\_width Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 4.9 | 3 | 1.4 | 0.2 | Iris-setosa |
|  |  |  |  |  |

**Input Given:**



**Output:**

