DATA MODELING

Assignment-1

Ashirbad Sarangi
SC23M002

[1]:
```
import pandas as analytics
import numpy as maths
from math import exp , pi
import warnings
warnings.filterwarnings("ignore")
```

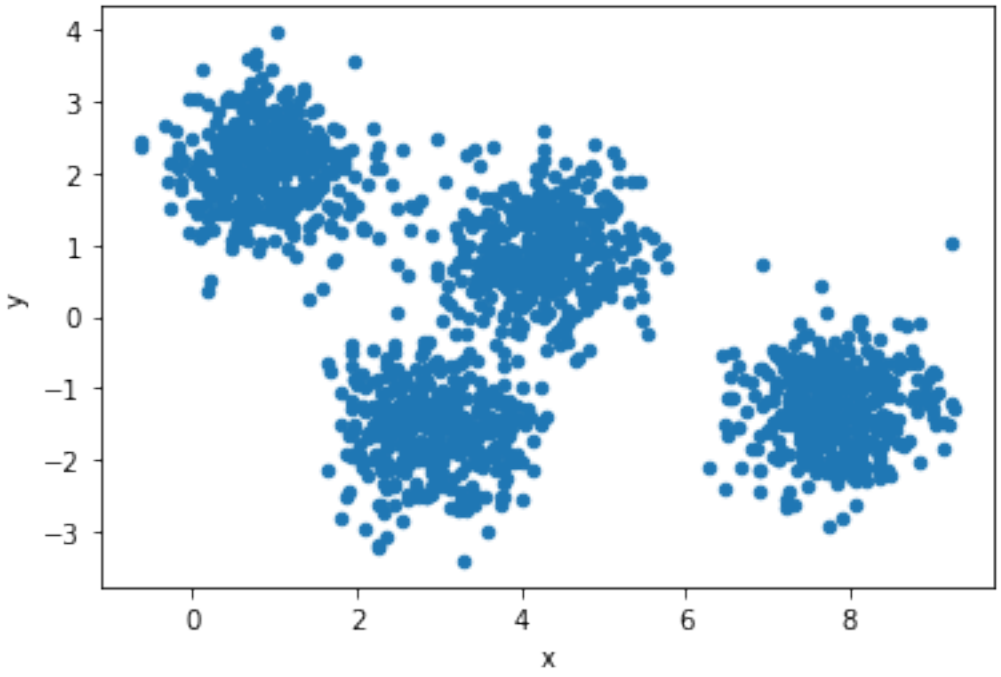
[2]:
```
df_raw = analytics.read_csv('../data/Data1.csv')
df_raw = df_raw.drop('Unnamed: 0',axis=1)
df_raw = df_raw.rename(columns = {"0":'x',"1":'y'})
df_raw
```

[2]:
```
              x         y
0     1.004939  2.319887
1     3.412653 -1.637157
2     7.483318 -1.399250
3     0.702826  2.038150
4     0.287620  2.191703
...        ...       ...
1595  1.475069  2.329653
1596  4.277030  2.183024
1597  0.814996  2.246927
1598  7.999698 -1.811024
1599  4.007795  0.121834

[1600 rows x 2 columns]
```

[3]:
```
df_raw.plot('x','y',kind = 'scatter')
```

[3]:
```
<Axes: xlabel='x', ylabel='y'>
```



[4]:
```
number_of_clusters = 4
number_of_datapoints = df_raw.shape[0]
number_of_attributes = df_raw.shape[1]
```

[109]:
```
max_loops = 100
```

[122]:
```
sigmas = maths.random.
 ↪random(size=(number_of_clusters,number_of_attributes,number_of_attributes))
means = maths.random.random(size=(number_of_clusters,number_of_attributes,1))
probabilities = maths.random.random(size = number_of_clusters)
probabilities = probabilities/sum(probabilities)
```

[123]:
```
print(means)
print("===")
print(sigmas)
print("===")
print(probabilities)
```

```
[[[0.78344473]
  [0.16494322]]

 [[0.37343133]
  [0.61292204]]

 [[0.02288354]
```

```
  [0.33344796]]

 [[0.9500946 ]
  [0.48390508]]]
===
[[[0.44029709 0.41213522]
  [0.93730204 0.89850869]]

 [[0.55527411 0.04670263]
  [0.84975036 0.84799134]]

 [[0.58183691 0.2988028 ]
  [0.34473284 0.08085565]]

 [[0.98426551 0.36286571]
  [0.58635703 0.88036117]]]
===
[0.4632705  0.41021202 0.00431618 0.1222013 ]
```

[124]:
```
max_loops = 10
for _ in range(max_loops) :
    p_i = []
    for i in range(number_of_clusters):
        mean = means[i]
        sigma = sigmas[i]
        probability = probabilities[i]
        conditional_probabilies = []
        for j in range(number_of_datapoints):
            x = maths.matrix(df_raw.iloc[j]).reshape(-1,1)
            mahalonabis_distance = float((x - mean).T @ sigma @ (x - mean))
            conditional_probability = 1/(sigma * pi ** (number_of_attributes/2))␣
↪* exp(-1/2 * mahalonabis_distance)
            conditional_probabilies.append(conditional_probability)
        conditional_probabilities = conditional_probabilities /␣
↪sum(conditional_probabilities)
        p_i.append(probability*conditional_probabilities)
    for i in range(number_of_clusters):
        probability_sum = sum(p_i[i])
        mean_sum = []
        sigma_sum = []
        prob_sum = []
        for j in range(number_of_datapoints):
            x = maths.matrix(df_raw.iloc[j]).reshape(-1,1)
            mean_sum.append(p_i[i][j] / probability_sum * x)
            sigma_sum.append((p_i[i][j] / (probability_sum - 1)) * float((x -␣
↪means[i]).T @ sigmas[i] @ (x-means[i])) )
```

```
        probabilities[i] = sum(p_i[i])/number_of_datapoints
        means[i] = sum(mean_sum)
        sigmas[i] = sum(sigma_sum)
```

[131]:
```
sum(p_i[3])
probability_sum
```

[131]:
```
1.7782629020369593e-30
```

[125]:
```
print(means)
print("===")
print(sigmas)
print("===")
print(probabilities)
```

```
[[[ 3.21331011]
  [-0.9141329 ]]

 [[ 3.21331011]
  [-0.9141329 ]]

 [[ 3.21331011]
  [-0.9141329 ]]

 [[ 3.21331011]
  [-0.9141329 ]]]
===
[[[3.15593424e-144 3.15593424e-144]
  [3.15593424e-144 3.15593424e-144]]

 [[3.04826996e-144 3.04826996e-144]
  [3.04826996e-144 3.04826996e-144]]

 [[2.96580957e-164 2.96580957e-164]
  [2.96580957e-164 2.96580957e-164]]

 [[1.77110018e-149 1.77110018e-149]
  [1.77110018e-149 1.77110018e-149]]]
===
[4.21342068e-33 3.73085661e-33 3.92554132e-35 1.11141431e-33]
```