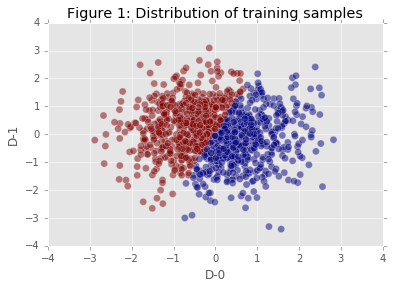
# Q1.

## a. Analysis of the data

The provided data has 2 numeric features. Figure 1 is the visualization of the data in a plane.



As displayed, the data is linearly separable.

I choose SVM to perform prediction as linear SVM classifiers are effective in classifying linearly separable sets such as the given dataset.

## b. Evaluation

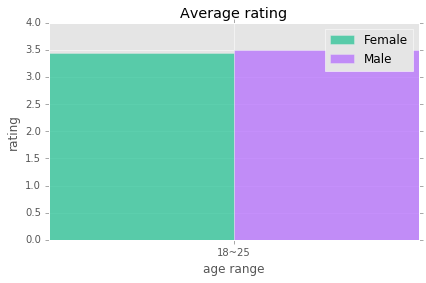
I used 3-round cross validation to evaluate the model. For each round, I calculated the mean and standard deviation of f1, recall, and precision scores. The result is as follows:

mean f1: 0.992901438023  
std f1: 0.00800981750292  
mean recall: 0.992901438023  
std recall: 0.00800981750292  
mean precision: 0.992901438023  
std precision: 0.00800981750292

# Q2.

## a. Compare average ratings among people aging from 18 to 25

Average rating of females: 3.438433  
Average rating of males: 3.504951



## b. 5 highest rated movies (based on average rating) in 1995

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Rating** |
| 0 | Someone Else's America (1995) | 5 |
| 1 | Close Shave, A (1995) | 4.491071 |
| 2 | Usual Suspects, The (1995) | 4.385768 |
| 3 | Stonewall (1995) | 4.2 |
| 4 | Braveheart (1995) | 4.151515 |

# Q3

I used LogisticRegression to predict the Value.

Before training the model using given features, I used “drop\_duplicates” method to remove

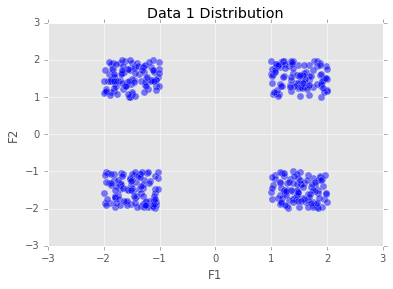
Redundancies.

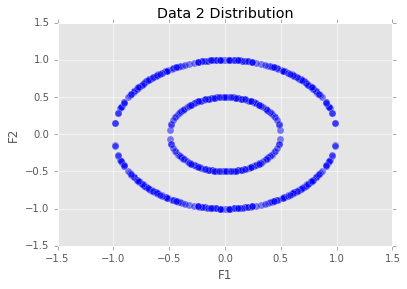
I used mean squared error to evaluate it, the result is:

5.0365819668375839e-22

# Q4

## a. Visualization





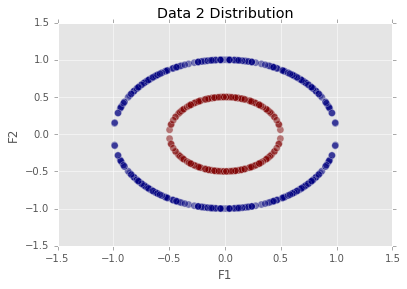
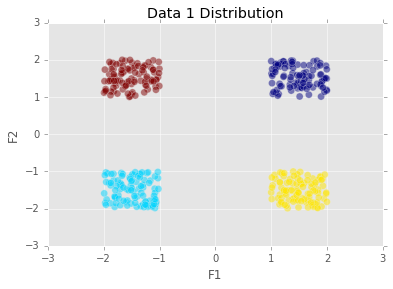
## b. Cluster given data

For data set 1 , I used KMeans with parameter k=4 to do clustering as in the visualization, data are grouped around 4 centers. By using KMeans I can find the 4 centers and the the corresponding cluster for each data.

For data set 2, I used nearest neighbor method to perform clustering. As the visualization shows, there is no center point that falls among the data. One significant factor is the distance between each data. Using nearest neighbor method, we can determine the clusters by measuring the distance between each neighbors.

For implementation, I used DBSCAN model provided by sklearn package which is using nearest neighbor under hood and performs fast as described in sklearn website. I set the eps parameter to increase some tolerance.

## C. Result



# Q5

First, I used TFIDFVectorizer to convert documents into tf-idf vectors.

Then, I calculated cosine similarity between the first document’s vector and other documents vectors.

At last, I plotted a bar chart to depict similarities between the first document and all other documents:

