Principles of Machine Learning(CS4011)
Programming Assignment#3

S Ram Ananth
ME15B153

November 19, 2017

1 Clustering

The given txt files were converted to ARFF format to make it readable by Weka by adding suitable lines to start of file to indicate name of relation,feature names, label names,etc.

The given 8 datasets were visualised using Weka.

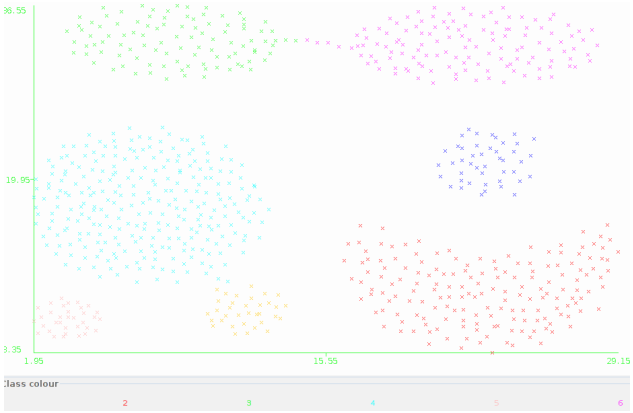

Figure 1: Aggregation

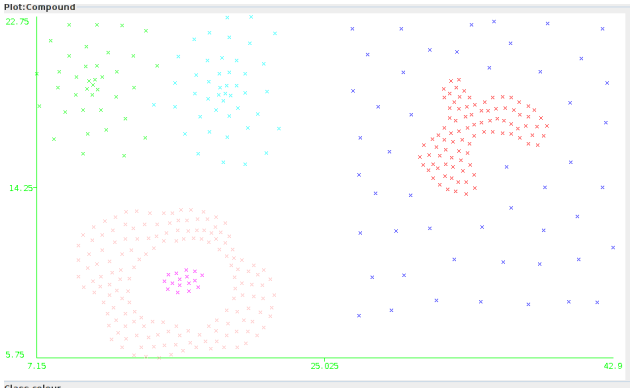

Figure 2: Compound

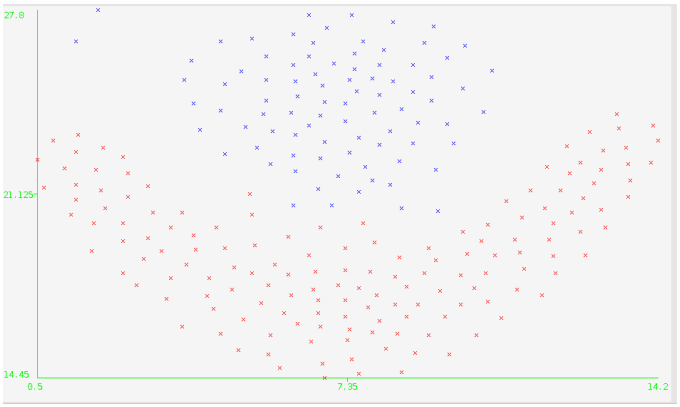

Figure 4: Flames

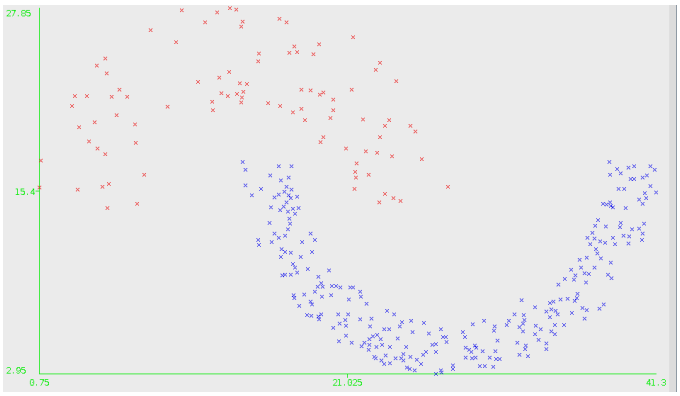

Figure 5: Jain

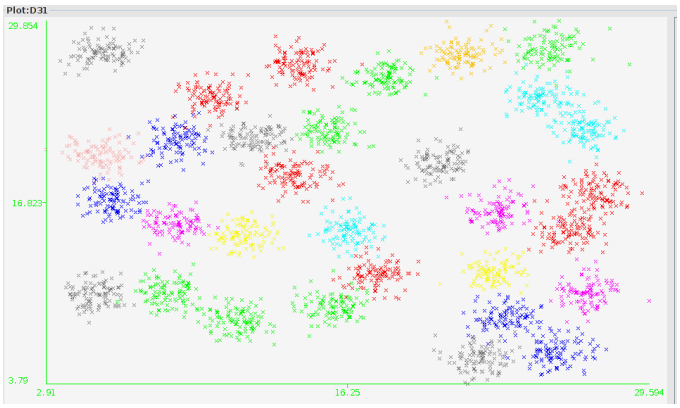

Figure 3: D31

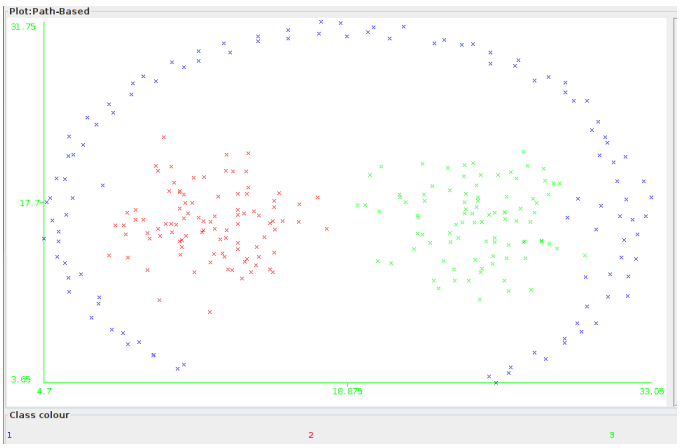

Figure 6: Path-Based

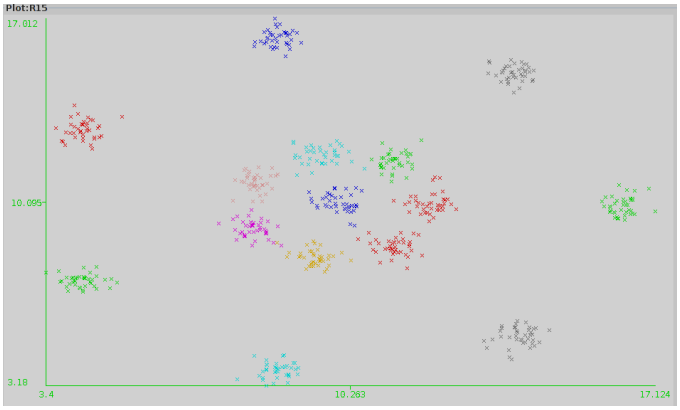

Figure 7: R-15

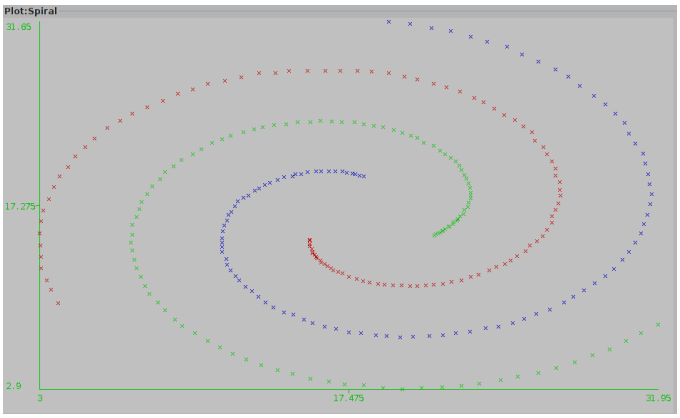

Figure 8: Spiral

Different clustering algorithms are suited for different types of datasets and their effect on the above visualised datasets are as follows:

Aggregation

As observed from the visualisations the clusters are well separated from one another. In this case k-means clustering will work pretty well. Hierarchical Clustering-Single will work well everywhere except maybe in the region where two clusters seem to meet. DBSCAN will work well too if we provide a correct value of minpts (should be sufficiently large)

Compound

This dataset has a cluster that is non-convex and lies completely inside another cluster. In this case k-means clustering will not work well for that particular region. Hierarchical Clustering-Single will work decently but that there is a chance of overlapped clusters being merged as one. DBSCAN works the best among the three for the correct values of parameters but otherwise it might also merge the two clusters mentioned.

D31

K-means and Hierachical Clustering perform well in this case too similar to Aggregation. DBSCAN performs somewhat badly on this dataset because depending on minpts and eps it either classifies many as outliers or merges two clusters together.

Flames

K-means will not work in this case due to shape of the bottom clusters. Since the clusters are well separated and are comprised of two dense regions, DBSCAN will work really efficiently. It is necessary to choose the appropriate minpoints for DBSCAN otherwise it can result in more clusters.

Jain

Here too K-means will not work due to shape of both the clusters. Since, the clusters are well separated and almost of uniform density DBSCAN will again work efficiently for the right choices of parameters. The performance here will be better than Flames as the clusters are better separated and hence the chances of the cluster being density connected is lesser resulting in correct clustering.

Path-based

$K-$means will not do will in this case also due to the shape of the clusters. For certain values of the parameters DBSCAN can give completely different clusters than Hierarchical Clustering-Complete (which doesn't have crucial paramters to tune) as visible in the figures below.

R15

The dataset has convex shaped clusters and are clearly separated. Therefore all algorithms will work very well in this case.

Spiral

Owing to the spiral shape of the clusters $K-$means will definitely not work well. Here DBSCAN and Hierarchical-Complete will work really well

$K-$means on R15 Dataset

K-means clustering algorithm was run on R15 dataset for different values of k ranging from 1 to 20 and the values were plotted and for k= 8 clustering purity was 0.5333

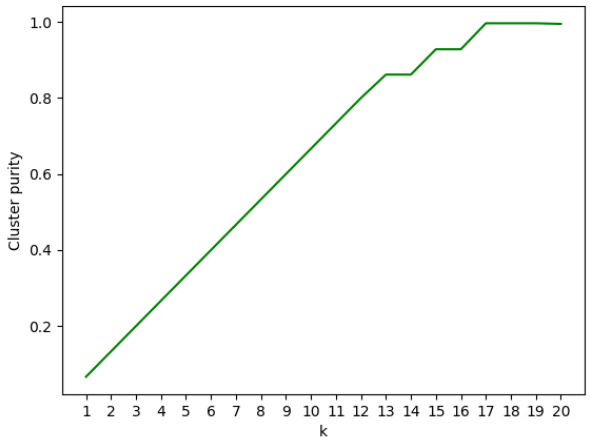

Figure 9: Purity as a function of k on R31 Dataset

DBSCAN on Jain Dataset

DBSCAN was run on the Jain dataset and parameters *minpoints* and *eps* were varied in a grid search manner.
For larger values of parameters like ϵ=0.9 and *minpoints* = 8 we get the following cluster with purity 0.74

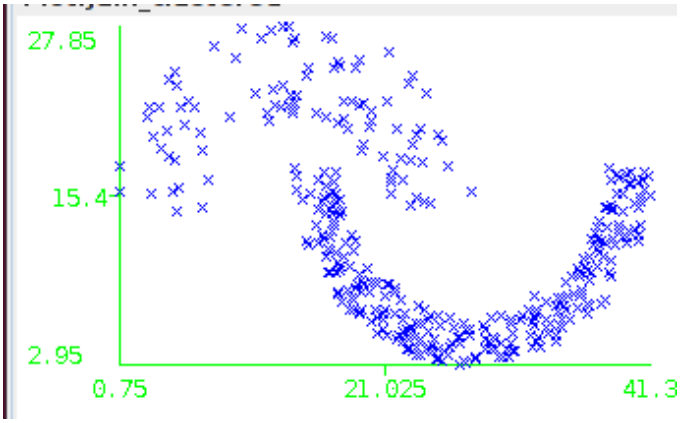

Figure 10: DBSCAN on Jain for large values of parameters

When smaller values of ϵ and *minpoints* were used such as ϵ=0.08 and *minpoints* = 4 three cluster were obtained and purity was much higher than the previous mentioned case and tending to 1, and a couple of points were treated as OUTLIERS and very few points were misclassified as a third cluster due to small values of parameters.

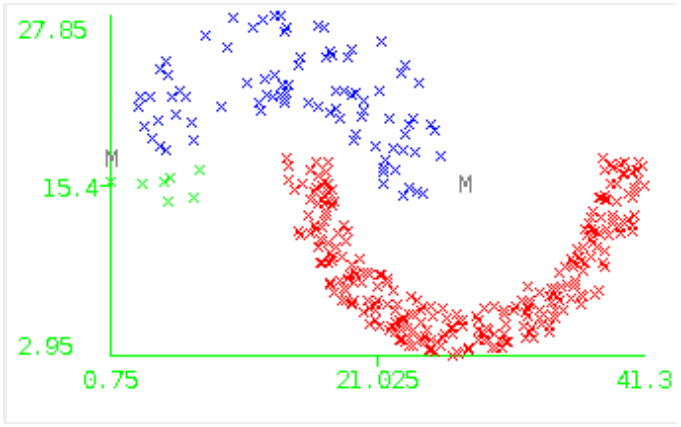

Figure 11: DBSCAN on Jain for large values of parameters

Comparison of clustering algorithms on different datasets

Various clustering algorithms were run on Spiral,Flames and Path-Based datasets and best was chosen for Hierarchical Clustering algorithm from different types of linkages.
For Path-Based,single Hierarchical did poorly and produced just one cluster. Hierarchical Ward performed the best. The below figures clearly show the difference between DBSCAN with certain chosen parameters and Hierarchical which can result in completely different clusters based on *minpoints* and *eps*

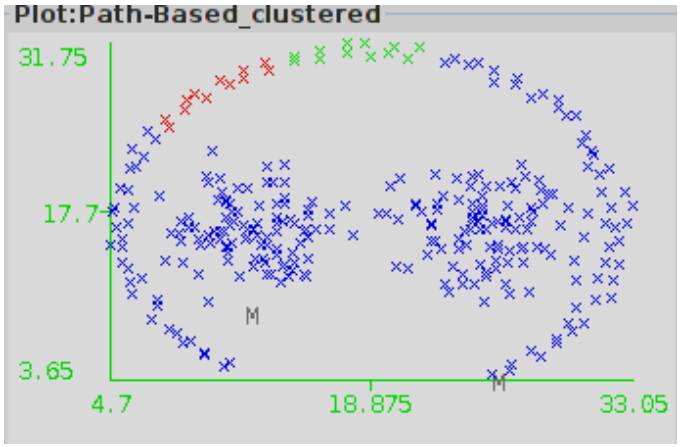

Figure 12: DBSCAN on Path-Based

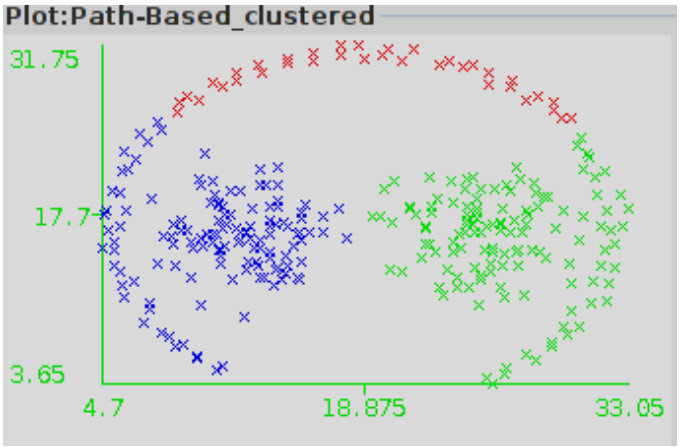

Figure 13: Hierarchical (Ward) on Path-Based

For Flames, DBSCAN resulted in classifying some points(here ϵ=0.08 and *minpoints*= 8 and for larger values resulted in a single cluster) as outliers and so does Hierarchical Ward linkage clustering

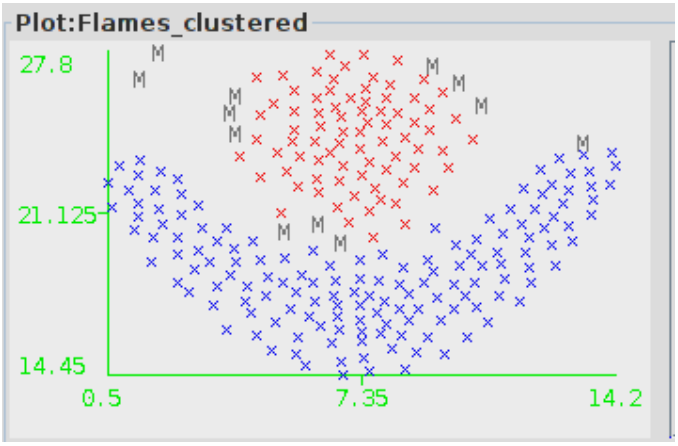

Figure 14: DBSCAN on Flames

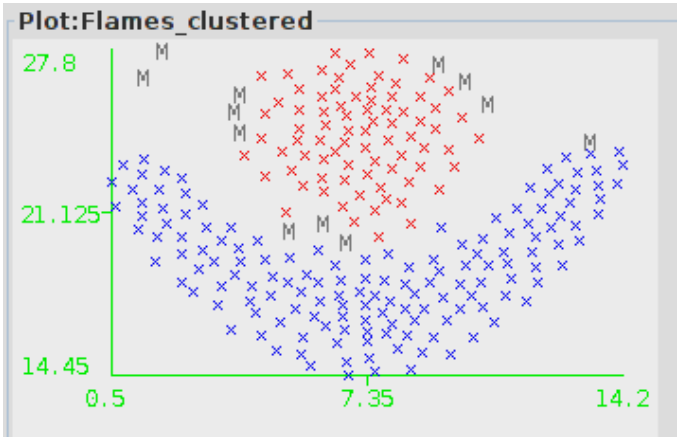

Figure 15: Hierarchical (Ward) on Flames

For Spiral, DBSCAN and Hierarchical Clustering performed really well as expected

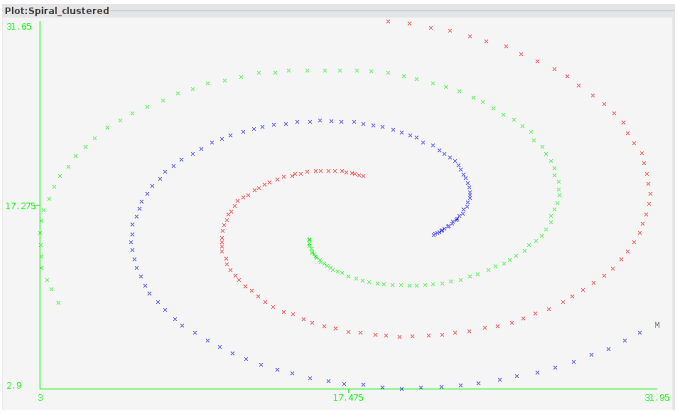

Figure 16: DBSCAN on Spiral

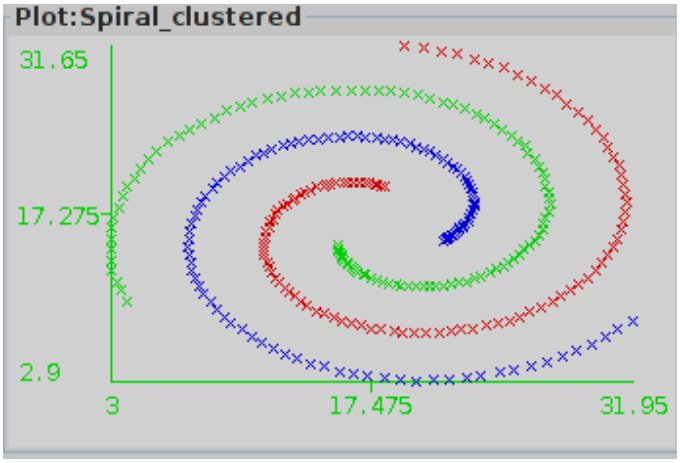

Figure 17: Hierarchical (Ward) on Spiral

$K-$means on D31 Dataset

K-means clustering algorithm was run on D31 dataset for k=31 and all clusters were succesfully retrieved

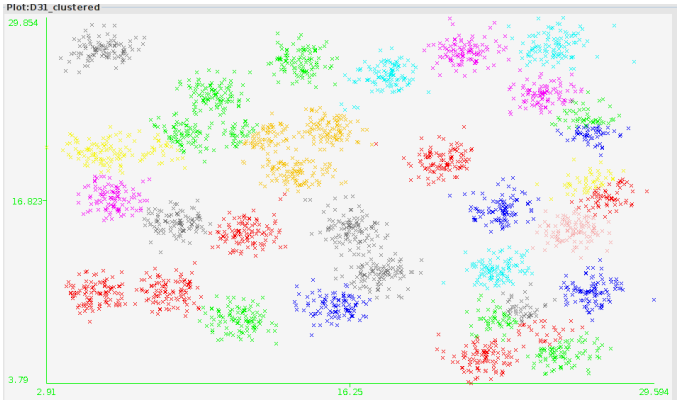

Figure 18: K means on D31 Dataset

DBSCAN too gave a similar performance, except in a very few places where it took portions of the

other cluster as part of it due to not clearly separated clusters, as the value of *minpoints* wasnt kept too high (6 in this case)

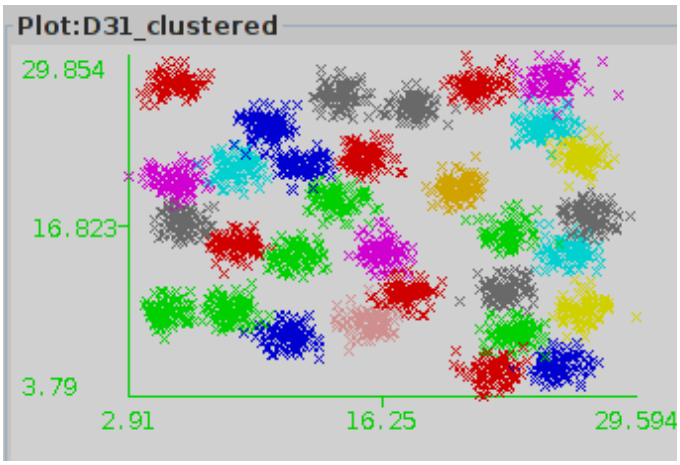

Figure 19: DBSCAN on D31 Dataset

2 Naive Bayes

Naive Bayes classification is one where each feature is assumed independant of the others given the class. It is a really fast and efficient algorithm and performance really well on datasets with large features and hence is one of the best algorithms when it comes to text classification as vocabulary size can be really large resulting in datasets with extremely large number of features.
The given dataset consisted of many parts with each part containing many text files for emails where each word was represented by a number and each mail was classified as being spam or not.
As such this text file cannot be fed directly to the Naive Bayes and some preprocessing was performed where each text file was represented as a frequency vector where i^{th} term in vector was the number of times the i^{th} word of the vocabulary (all unique words in all training files) occured.
Each of the parts was then put into 5 folds for a $80-20$ split.

Multinomial Naive Bayes

Multinomial Naive Bayes was then implemented (using the assumption that the likelihood was a multinomial distribution) ,from scratch and applied on the dataset and the average metrics after 5 fold were obtained as:

Metric	Average over the 5 folds
Accuracy	0.965425
Precision	0.949397
Recall	0.972960
F1-Score	0.960957

A precision vs recall curve was then plotted for the 5 folds and the best graph based on average precision score is s hown below

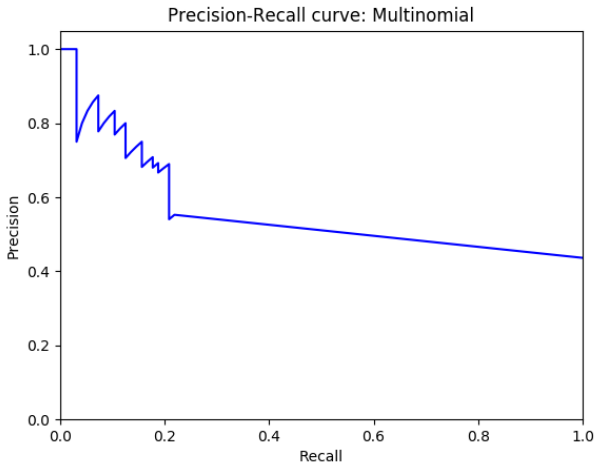

Figure 20: PR Curve for Multinomial Naive Bayes

Bernoulli Naive Bayes

Bernoulli Naive Bayes was then implemented (using the assumption that the likelihood was a bernoulli distribution for each word in the vocabulary) ,from scratch and applied on the dataset and the average metrics after 5 fold were obtained as:

Metric	Average over the 5 folds
Accuracy	0.921767
Precision	0.985487
Recall	0.833806
F1-Score	0.902546

A precision vs recall curve was then plotted for the 5 folds and the best graph based on average precision score is s hown below

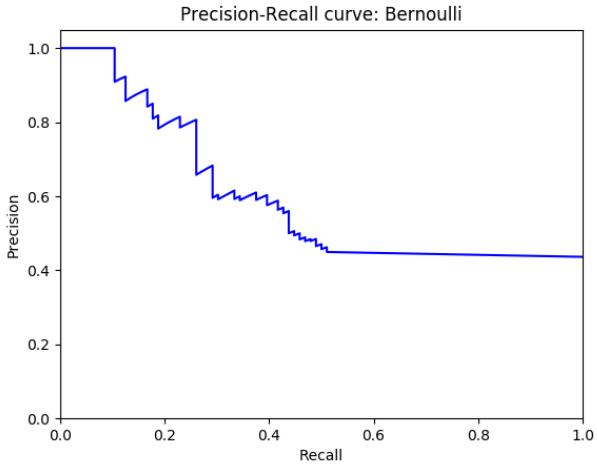

Figure 21: PR Curve for Bernoulli Naive Bayes

Bernoulli is different from Multinomial in the sense that it only takes into account occurences but completely ignores the number of occurences.

Beta Naive Bayes

Beta Naive Bayes was then implemented (using the assumption that the likelihood was a bernoulli distribution for each word in the vocabulary and prior was from Beta(α, β)) ,from scratch and applied on the dataset and the average metrics after 5 fold were obtained as:

A precision vs recall curve was then plotted for the 5 folds and the best graph based on average precision score is s hown below

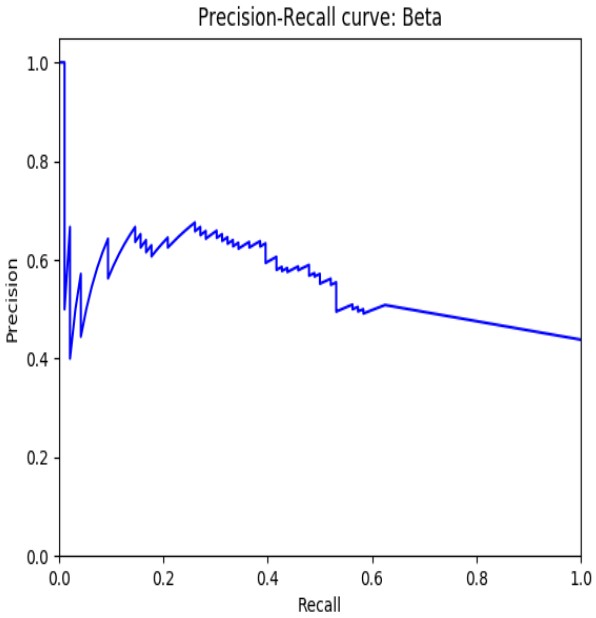

Figure 22: One PR curve for $\alpha = 2, \beta = 10$

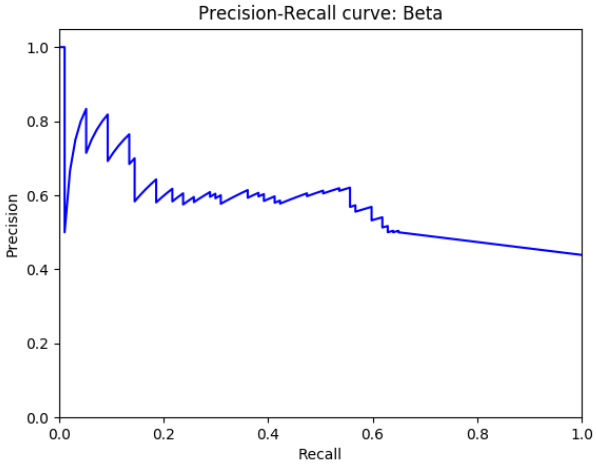

Figure 23: One PR curve for $\alpha = 5, \beta = 5$