Shashank Agarwal
Agarwal.202@osu.edu

Anurag Kalra
Kalra.25@osu.edu

Data mining lab: Report 2

Objectives:

- Build 2 classifiers (we chose Naive Bayes and K-Nearest neighbour) based on the feature vectors produced in Assignment 1 of lab. Predict the Topic labels for articles in documents.
- Find the accuracy of these classifiers. Determine time to build classifier model -Offline costs and time to classify new tuples -online costs. Measuring the parameters for 80/20 and 60/40 split of training and testing data.

We used the feature vector (where each row is a document and the value is a series of 0 and 1. “1” represents that the document has the given keyword and 0 represents the absence of a keyword) (Filename: feature_matrix.pytext) which we produced after Assignment 1 of lab in classifiers. All records were cleaned and saved in ‘out_file.csv’ with each field separated by commas.

Naive Bayes classifier

We used scikit-learn library to help in classifying Topic labels. We are able to directly leverage our feature vector. We used 3 different implementations of Naive Bayes algorithm: Gaussian Naive Bayes, Multinomial Naive Bayes and Bernoulli Naive Bayes.

Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The parameters σ_y and μ_y are estimated using maximum likelihood [1]. As the Gaussian model is best used for continuous data, we predicted it to have least accuracy for our data classification. After testing we can see that this model gave the least accuracy compared to next 2 models.

Bernoulli Naive Bayes:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

[1]. The Bernoulli classifier model is good for classifying documents where the input features are boolean(binary) values. We were expecting this model to give the best accuracy. However in practice, the multinomial naive bayes based model gave slightly better accuracy.

Multinomial Naive Bayes:

The multinomial classification model typically expects word frequency vector as input. Although tf-idf vector also works in practice. Our feature vector of word document binary value also worked well in this model and we got the best accuracy from this model.

Shashank Agarwal
Agarwal.202@osu.edu

Anurag Kalra
Kalra.25@osu.edu

Results:

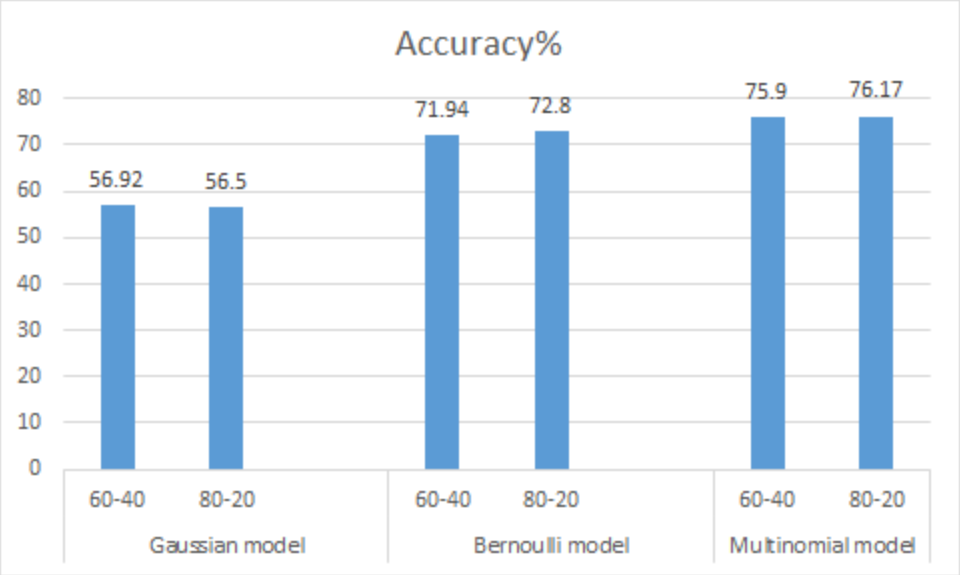
Total documents: 21,578
For 80:20 skew, number of training articles: 8514 (after removing all documents without topic)
For 60:40 skew, number of training articles: 5695 (after removing all documents without topic)

	80:20	60:40
Gaussian model		
Offline cost	0:00:27.493419	0:00:20.454081
Total number of predictions	2258	4506
Correct predictions	1277	2565
Accuracy	56.5%	56.92%
Precision	26.0273972603	32.8947368421
Online cost	0:01:00.233989	0:01:46.488136
Bernoulli model		
Offline cost	0:00:28.234636	0:00:21.522949
Total number of predictions	2258	4506
Correct predictions	1644	3242
Accuracy	72.8%	71.94%
Precision	65.95	56.5217391304
Online cost	0:00:56.450237	0:01:29.240347
Multinomial model		
Offline cost	0:00:28.950202	0:00:20.573384
Total number of predictions	2258	4506
Correct predictions	1720	3424
Accuracy	76.17%	75.9%

Shashank Agarwal
Agarwal.202@osu.edu

Anurag Kalra
Kalra.25@osu.edu

Precision	63.829787234	76.0869565217
Online cost	0:00:10.044041	0:00:17.889725



K-Nearest neighbour

We used scikit-learn library to implement K Nearest Neighbour classifier. We were able to directly use our feature vector as input to the library without doing any transformation on it. We used Ball tree data structure from library. The reason we picked it was because it is specially good for high dimension data.

[2] A ball tree recursively divides the data into nodes defined by a centroid C and radius r, such that each point in the node lies within the hyper-sphere defined by r and C. The number of candidate points for a neighbor search is reduced through use of the *triangle inequality*:

$$|x + y| \leq |x| + |y|$$

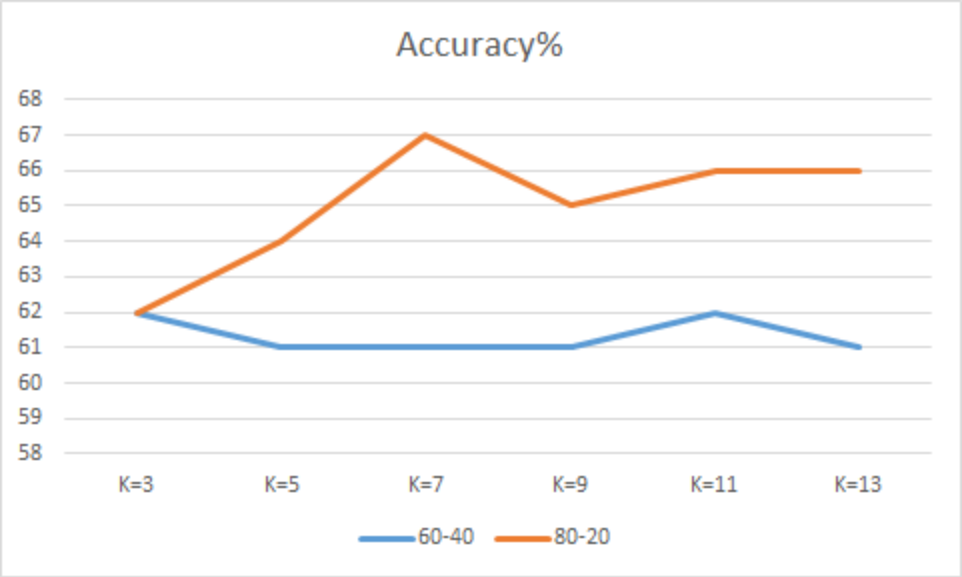
Results:

Total documents: 21,578
For 80:20 skew, number of training articles: 8514 (after removing all documents with no topic)
For 60:40 skew, number of training articles: 5695 (after removing all documents with no topic)

Shashank Agarwal
Agarwal.202@osu.edu

Anurag Kalra
Kalra.25@osu.edu

	K=3	K=5	K=7	K=9	K=11	K=13
80-20						
Accuracy%	62	64	67	65	66	66
Total execution time (sec)	0:02:02.703219	0:02:16.599799	0:02:00.656269	0:01:56.165304	0:01:58.765790	0:02:03.281071
60-40						
Accuracy%	62	61	61	61	62	61
Total execution time (sec)	0:02:02.70480	0:02:16.799349	0:02:01.636243	0:01:56.765264	0:01:58.790130	0:02:03.372712



Special handling for topics. For both the classifiers:

1. We do not consider the articles which don't have TOPIC in our classifier.
2. When creating classifier, for documents having multiple topics, we treat the single document as separate multiple documents. So for each topic value we consider the article as a new article instance.

Shashank Agarwal
Agarwal.202@osu.edu

Anurag Kalra
Kalra.25@osu.edu

Measuring accuracy

For single topic articles, we match if the predicted value matches exactly with the actual topic value. For multiple topic articles, we try to match the predicted topic with any of the actual topic values. If any topic matches with the predicted one, we consider that as success.

Work division

Shashank Agarwal: Naive Bayes classifier
Anurag Kalra: KNN classifier

References

[1] : http://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes
[2] :http://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbor-algorithms