239AS - Special Topics in Signals and Systems
Project 2 - Classification Analysis

Mansee Jadhav - 204567818
Mauli Shah - 004567942
Ronak Sumbaly - 604591897

February 21, 2016

INTRODUCTION

In this project we have done data analysis on the **20 Newsgroups dataset** which comprises of 20,000 newsgroup documents that are partitioned evenly across 20 newsgroups categories. The dataset is analyzed by using the data mining approach of classification. Classification is an approach for identifying to which of a set of categories does a new observation belongs, on the basis of a training set of data containing instances whose category are known.

The 20 Newsgroup Dataset is used for the classification problem and is modeled and tested against various classifiers like Naive Bayes, Support Vector Machines and Logistic Regression. Observations obtained for each of the questions is presented below in this report along with the method employed.

Dataset & Problem Statement

Ques (a) Histogram of the number of documents per topic

For any classification problem unbalanced datasets should be handled properly. We plotted a histogram of the number of documents versus topics to ensure if documents are evenly distributed. As seen below in the histogram, number of documents (≈ 600) are almost evenly distributed across each of the topics.

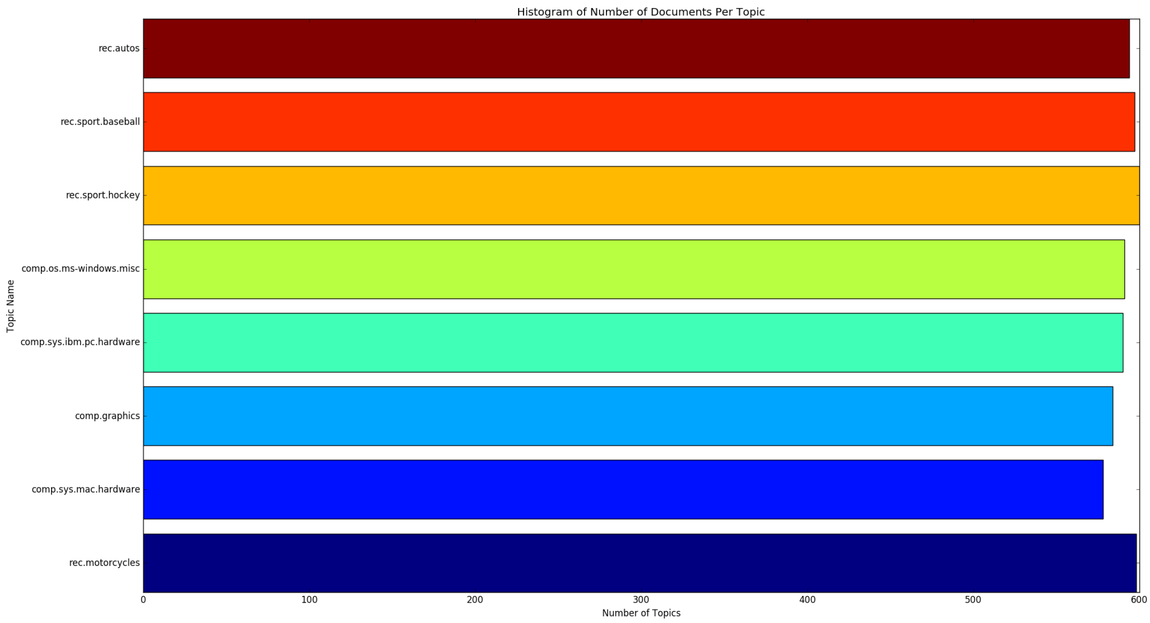

Figure 1: Histogram: Number of Documents vs. Topic Name

All the above categories were processed to two classes **Computer Technology & Recreational Activity**. Further we computed the number of documents in the two classifying groups
Computer Technology and Recreational Activity

TRAINING DATASET

1. Number of Documents in Computer Technology : 2343
2. Number of Documents in Recreational Activity : 2389

TESTING DATASET

1. Number of Documents in Computer Technology : 1560
2. Number of Documents in Recreational Activity : 1590

Modeling Text Data & Feature Extraction

Ques (b) Pre-processing and TFxIDF Representations

Since there are lot of common words in each document we need to preprocess the data so that we can find significant terms in the dataset. For this, we first remove punctuations, common stop words and finding which words share the same stem so that they can be counted together while finding their TFXIDF. In order to do the latter we used a SnowBall stemmer (nlkt) to achieve this. In addition to removing the above mentioned terms, we also removed non ASCII characters. This helped in increasing the over-all accuracy when we tired training the classifier.

Once the data has been pre-processed, the next step is to find the TFXIDF of each term. For this we convert the document into a set of numerical features. This is done using CountVectorizer. Next we get the TFxIDF Representations using a Transformer. A matrix is obtained with the number of documents (records) as the row and the number of terms obtained as the number of columns.

Number of Terms Extracted – 57088

This result is the number of terms extracted on the classes which come under computer technology and recreational activity.

Ques (c) TFxICF - 10 most significant terms

To find the top 10 significant terms for classes **comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, misc.forsale, and soc.religion.christian** the following steps were followed:

1. Each document is cleaned to remove stop words, punctuations and stems. The words are stored along with their count in a list where each index of the list represents the class number. Since the given dataset has 20 classes, a list of size 20 is obtained in the end of this step.

2. All the unique terms in each class and their corresponding count are found.

3. To find the significance of each term a metric called TFxICF is computed using the following formula.

4. With all the details available, TFxICF can be calculated for each term. These details are stored in a dictionary and sorted to find the 10 most significant words

Results

comp.sys.ibm.pc.hardware	comp.sys.mac.hardware	misc.forsale	soc.religion.christian
adaptec	iisi	obo	liturgi
motherboard	duo	hobgoblin	kulikauska
irq	quadra	spiderman	clh
vlb	centri	liefeld	christ
aspi	powerbook	hiram	atheist
dx	nubus	xforc	cathol
floppi	fpu	hulk	atho
scsi	scsi	sabretooth	sabbath
ide	lciii	wolverin	resurrect
jumper	simm	forsal	scriptur

Please Note : Calculating TFxICF for each term is a time consuming process. There are many terms in each class. Hence the total time to get the top 10 significant terms is proportional to the time taken to calculate TFXICF for each term within the class.

Feature Selection

Ques (d) LSI Decomposition of TFxIDF Matrix

Here we deal with feature selection. We select a subset of more relevant features to improve the performance measure. **Latent semantic indexing** (LSI) is an indexing and retrieval method that uses a mathematical techniques to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text.

LSI is based on the principle that words that are used in the same contexts tend to have similar meanings. We reduce the features to lower dimensional space by representing data in term document matrix , with columns of TFxIDF representation of documents obtained above. Steps followed.

1. Preprocess the datasets - Both Training Testing.

2. Create TFxIDF Vector Representation.

3. Apply LSI Decomposition to return feature space to 50 terms.

Learning Algorithms

Ques (e) Linear Support Vector Machines

Linear SVM classifier is capable of doing multi-class classification on the dataset. We used a **linear kernel** to train out classifier and then it on our training dataset. The statistics obtained for the classifier are as follows.

Table 1: SVM Statistics

Statistic	**Result**
Accuracy	97.174
Precision	96.413
Recall	98.050

Table 2: SVM Confusion Matrix

	Predicted: Computer	**Predicted: Recreational**
Actual: Computer	1502	58
Actual: Recreational	31	1559

In order to characterize the trade-off between the two quantities we plot the **receiver operating characteristic (ROC) curve**. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings. An area of 1 signifies perfect classification. As seen from below obtained ROC all the classes have area ≈ 1. Hence all our test cases are classified correctly.

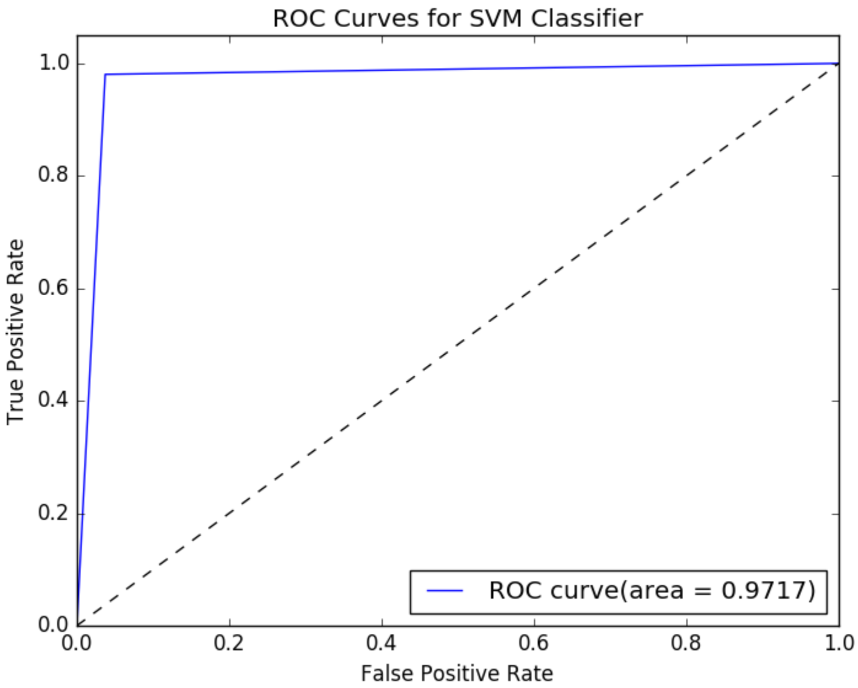

Figure 2: ROC Curve: Linear SVM Classifier

Ques (f) Cross Validated Support Vector Machines

Soft-margin SVM was used to minimize training error. In order to obtain best results we performed a 5 fold cross validation. On careful analysis, we found that the best parameter value i.e value where Soft-SVM gave the best results was at $k = 0$ which is equivalent to Hard-margin SVM. The statistics obtained for the classifier are as follows.

Table 3: Cross Validated Soft-Margin SVM Statistics

Statistic	**Result**
Accuracy	97.174
Precision	96.413
Recall	98.050

Table 4: Cross Validated Soft-Margin SVM Confusion Matrix

	Predicted: Computer	**Predicted: Recreational**
Actual: Computer	1502	58
Actual: Recreational	31	1559

Ques (g) Naive Bayes

We use **Naive Bayes** algorithm for the same classification task as performed earlier. The algorithm estimates the maximum likelihood probability of a class given a document with feature set X, using Bayes rule, based upon the assumption that given the class, the features are statistically independent. We used **Gaussian Naive-based classifier**. The statistics obtained for the classifier are as follows.

Table 5: Naive Bayes Statistics

Statistic	**Result**
Accuracy	93.777
Precision	96.841
Recall	90.629

Table 6: Naive Bayes Confusion Matrix

	Predicted: Computer	**Predicted: Recreational**
Actual: Computer	1513	47
Actual: Recreational	149	1441

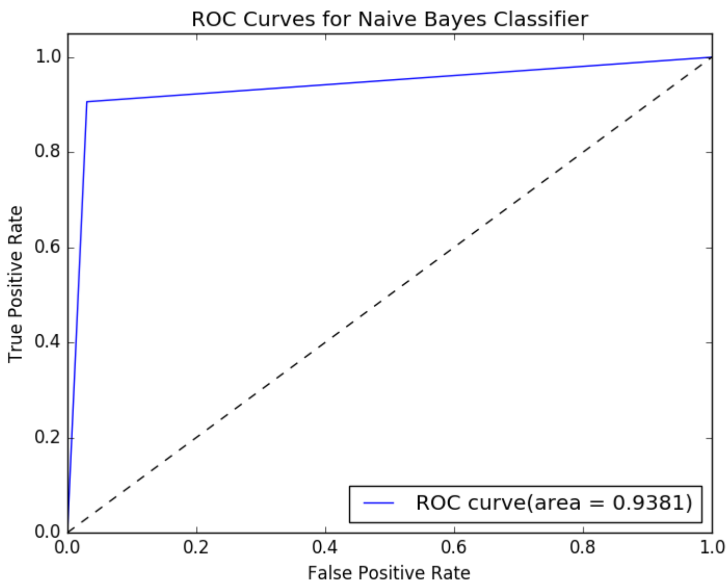

Figure 3: ROC Curve: Naive Bayes Classifier

As seen above Naive Bayes has less area under the ROC curve as compared to SVM classifier signifying that there are some records that were incorrectly classifier.

Ques (h) Logistic Regression

Logistic Regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. It is thus analogous to Regression function. We now apply Logistic regression classification on our data. The statistics obtained for the classifier are as follows.

Table 7: Naive Bayes Statistics

Statistic	**Result**
Accuracy	97.492
Precision	96.780
Recall	98.301

Table 8: Naive Bayes Confusion Matrix

	Predicted: Computer	**Predicted: Recreational**
Actual: Computer	1508	52
Actual: Recreational	27	1563

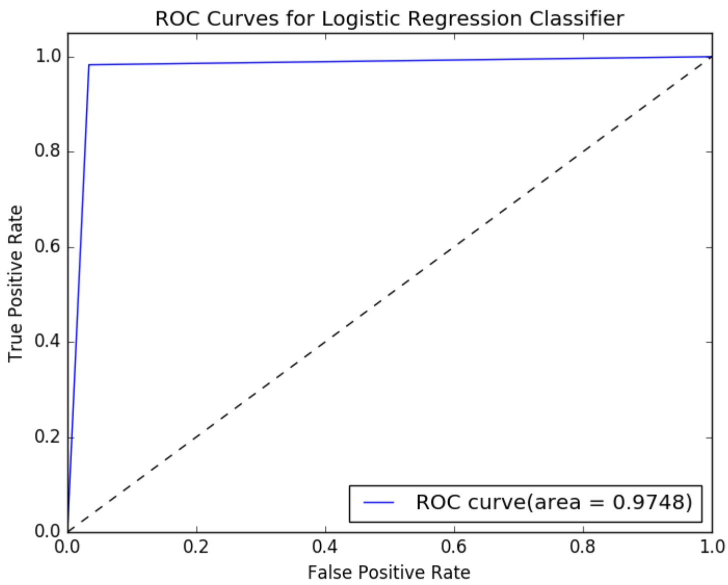

Figure 4: ROC Curve: Logistic Regression Classifier

In order to better visualize the ROC curves of all the classifiers, the ROC curves were combined in the following figure.

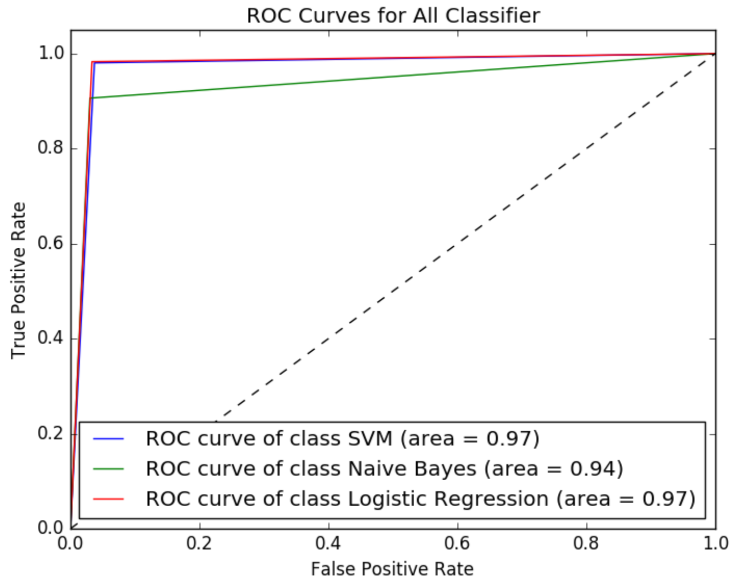

Figure 5: ROC Curve: All Classifiers

As seen Logistic Regression and SVM have almost the same area covered in the ROC signifying that they classify the records correctly while Naive Bayes has the least amongst them all which basically means that there are some records that are incorrectly classified by it.

Multiclass Classification

Ques (i) Multiclass classification - Naive Bayes & SVM

We train classifiers on the documents belonging to the classes **A - comp.sys.ibm.pc.hardware, B -comp.sys.mac.hardware, C - misc.forsale, and D - soc.religion.christian**. Since this is a multi-class problem we use a OneVsOne and OneVsRest classification techniques to train our classifier (Naive Bayes SVM). The steps followed to train the dataset are the same as seen in Ques (b) Ques (c).

Results obtained for OneVsOneClassification
Results : Naive Bayes Classifier

Table 9: Naive Bayes Statistics

Statistic	Result
Accuracy	72.140
Precision	75.274
Recall	71.940

Table 10: Naive Bayes Confusion Matrix

	Predicted: A	Predicted: B	Predicted: C	Predicted: D
Actual: A	287	13	91	1
Actual: B	93	162	126	4
Actual: C	49	28	313	0
Actual: D	1	0	30	367

Results : SVM Classifier

Table 11: SVM Statistics

Statistic	Result
Accuracy	88.945
Precision	89.113
Recall	88.899

Table 12: SVM Confusion Matrix

	Predicted: A	Predicted: B	Predicted: C	Predicted: D
Actual: A	338	38	16	0
Actual: B	43	324	18	0
Actual: C	27	13	349	1
Actual: D	9	2	6	381

Results obtained for OneVsRestClassification

Results : Naive Bayes Classifier

Table 13: Naive Bayes Statistics

Statistic	Result
Accuracy	71.565
Precision	74.889
Recall	71.355

Table 14: Naive Bayes Confusion Matrix

	Predicted: A	Predicted: B	Predicted: C	Predicted: D
Actual: A	273	13	105	1
Actual: B	75	157	148	5
Actual: C	41	30	317	2
Actual: D	0	1	24	373

Results : SVM Classifier

Table 15: SVM Statistics

Statistic	Result
Accuracy	89.584
Precision	89.516
Recall	89.534

Table 16: Naive Bayes Confusion Matrix

	Predicted: A	Predicted: B	Predicted: C	Predicted: D
Actual: A	327	40	21	4
Actual: B	27	329	27	2
Actual: C	21	12	355	2
Actual: D	3	1	3	391