PERFORMANCE COMPARISON OF VARIOUS CLASSIFICATION ALGORITHMS FOR TEXT CLASSIFICATION

CSC 565 Data Mining - Final Project Report

Due on April 29, 2016 at 11:59pm

Professor Predrag Radivojac CSC 565

INDIANA UNIVERSITY

Anirudh Kamalapuram Muralidhar | Praneet Vizzapu | Santhosh Soundararajan

.

PERFORMANCE COMPARISON OF VARIOUS CLASSIFICATION ALGORITHMS FOR TEXT CLASSIFICATION

TEAM MEMBERS:

Anirudh Kamalapuram Muralidhar[1] anikamal@umail.iu.edu

Santhosh Soundararajan[2] soundars@umail.iu.edu

Praneet Vizzapu[3] pranvizz@umail.iu.edu

GUIDANCE UNDER FACULTY:

Johan Bollen

Associate Professor

SOIC, Indiana University

jbollen < at > indiana < dot > edu

1. PROJECT SUMMARY & MOTIVATION

1.1 Objectives:

Our primary objective in this project is to implement and explore various data mining approaches to perform **Text Classification** on a real-world datasets which is why we chose the **"20 Newsgroups Data"** and with this data, we intend to perform the following:

i. Implement the following text classification algorithms on the data:

  (a) Decision Tree classifiers

  (b) Random Forest Classifier

  (c) Naive Bayes Classifier Multinomial

  (d) Logistic Regression

  (e) SVM - SGD classifier

ii. Results with Cross-Validation and Without Cross-Validation

iii. Performance measured with precision, recall, F-1 and average

1.2 Significance of the Problem:

Automatic text classification or document classification, which is the task of assigning unlabeled text documents to pre-assigned classes of documents, is an important task that can help both in organizing data as well as in the text based information retrieval.

Automatic text classification has been an important application and research topic since the dawn of digital documents. Today, text classification is a necessity due to the very large amount of text documents that we are generating on a daily basis along with Business Intelligence that depend on textual data such as Product Reviews, Tweets etc.
So, our goal is to automatically classify large volumes of text data via a number of data mining algorithms to suggest the best possible approach for the text classification problem.

2. BACKGROUND

In this section, we will introduce the concepts of each of the five algorithms we will be using for classifying 20,000 newsgroup documents and later the numerical statistic TF-IDF, that is intended to reflect how important a word is to a document in a collection or corpus:

a)**Decision Tree classifiers:** A Decision Tree is a classifier in the form of a tree structure, where each node is either a: **Leaf node -** indicates the value of the target attribute (class) of examples, or **Decision node -** specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.
A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance.

b)**Random Forest Classifier:** This algorithm is designed for scrutinizing very high dimensional data with numerous classes whose representative data is text corpus. A novel feature weighting method and tree selection method are developed for making random forest framework well suited to categorize text documents with a lot of topics.
In our project, we introduce a feature weighting method for subspace sampling, then a tree selection method is presented. By integrating these two methods, a novel improved random forest algorithm is proposed.

i. Feature Weighting Method: In this subsection, we will give details of the feature weighting method for subspace sampling in random forests. Consider an M-dimensional feature space A1,A2, ,AM. We present how to compute the weights w1,w2,,wM for every feature in the space. These weights are then used in improved algorithm to grow each decision tree in random forest.

ii. Feature Weight Computation: To compute the feature weight, we measure the informativeness of each input feature A as its correlation to the class feature Y. A large weight indicates that the class labels of objects in training data are correlated with the values of feature A. Therefore, A is informative to the class label of objects and has a strong power in prediction of class labels of new objects.

c)**Naive Bayes Classifier Multinomial::** While this is a simple approach which assumes conditional independence of the features, it still provides good results in the context of text classification with the least **Prior Data** in the form of training data comparing all other learning models. Our intention is to compute the Posterior probability and evaluate the quality of results with the help of Prior and Likelihood probabilities in the Training Data. And, we will use the Full Bayesian Network to construct a Gibbs Sampler which will sample from the posterior probability $P_{posterior}(Topic \mid Word)$ in order to predict the probability of topic for a given text documet.

e.)Logistic Regression:

Logistic regression is a regression model that is popularly used for classification tasks. In logistic regression, the probability that a binary target is True is modeled as a logistic function of a linear combination of features.

The following sections illustrates how logistic regression is used to train a multi-dimensional(n dimentions) classifier for our case of text classification. The training data consists of 20000 news group data. The decision boundaries separate out the data into n classes.

e.)SVM - SGD classifier:

SVM has SVC(classification),SVR(Regression) and SGD(Stochastic Gradient Descent) algorithms to do class classification and prediction. But it has bad performance in text classification, as it has high demands for good tokenizers (filters) and hence we thought that the accuracy could be very bad. Hence We are going for SGD.

2.1 TF-IDF based text processing:

We will give a quick informal explanation of TF-IDF before proceeding. Essentially, TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. Words that are common in a single or a small group of documents tend to have higher TFIDF numbers than common words such as articles and prepositions. The formal procedure for implementing TF-IDF has some minor differences over all its applications, but the overall approach works as follows. Given a document collection D, a word w, and an individual document $d \in D$, we calculate:

$$w_d = f_{w,d} * log(|D|/f_{w,d}) \qquad (i)$$

where $f_{w,d}$ equals the number of times w appears in $d, |D|$ is the size of the corpus, and $f_{w,D}$ equals the number of documents in which w appears in D.

2.2 Word2Vec based text processing: Why we stay away from this model?

The bag-of-words model is one of the most popular representation methods for object categorization. The key idea is to quantize each extracted key point into one of visual words, and then represent each image by a histogram of the visual words. For this purpose, a clustering algorithm (e.g., K-means), is generally used for generating the visual words. Although a number of studies have shown encouraging results of the bag-of-words representation for object categorization, theoretical studies on properties of the bag-of-words model is almost untouched, possibly due to the difficulty introduced by using a heuristic clustering process.

2.2 Pervious Work on the problem:

There are many papers available online which tend to show the past work done on text classification and these are some problems they are trying to solve. This paper Text Classification Using Machine Learning Techniques by KOTSIANTIS, IKONOMAKIS of University of Patras Greece talk about a few problems they wanted to talk about. Make the dimensionality reduction more efficient over large corpus. Would combining uncorrelated, but well performing methods yield a performance increase? And they generated a accuracy of 57.0 for Naive Bayes achieves a performance and with 65.9 (polynomial SVM) and 66.0 (RBF SVM) the SVMs perform substantially better than all conventional methods.

Another paper A Review of Machine Learning Algorithms for Text-Documents Classification by Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee*, Khairullah K from the Universiti Teknologi PETRONAS, Tronoh, Malaysia talk about several algorithms or combination of algorithms as hybrid approaches was proposed for the automatic classification of documents, among these algorithms, SVM, NB and kNN classifiers. SVM classifier has been recognized as one of the most effective text classification method in the comparisons of supervised machine learning algorithms [74]. SVM capture the inherent characteristics of the data better and embedding the Structural Risk Minimization.

- Mining trend, i.e. marketing, business, and financial trend (stock exchange trend) form e-documents (Online news, stories, views and events).

- For Spam filtering and e-mail categorization the user may have folders like electronic bills, e-mail from family, friends and so on, and may want a classifier to classify each incoming e-mail thats automatically move it to the appropriate folder.

- Which feature selection methods are both computationally scalable and high-performing across classifiers and collections? How Lexicon generation can be made. Lexicon is the set of words the document contains. These words are used as the features of the learning model (the classifier). Once the Lexicon is generated then, based on it feature vectors of the documents need to be generated.

3. METHODS

3.1 Description Of The Data

We have used the famous 20 newsgroup dataset[1] for this analysis. This data set is a collection of 18,846 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. This data was collected by Ken Lang, probably for his Newsweeder: Learning to filter netnews paper. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other while others are highly unrelated. The 20 categories present in this data set are given in the table next page.

	$Category$	$Sub - Category$	$Number of Documents$
1	comp	comp.graphics	973
2	comp	comp.os.ms-windows.misc	985
3	comp	comp.sys.ibm.pc.hardware	982
4	comp	comp.sys.mac.hardware	963
5	comp	comp.windows.x	988
6	rec	rec.autos	990
7	rec	rec.motorcycles	996
8	rec	rec.sport.baseball	994
9	rec	rec.sport.hockey	999
10	sci	sci.crypt	991
11	sci	sci.electronics	984
12	sci	sci.med	990
13	sci	sci.space	987
14	misc	misc.forsale	975
15	talk	talk.politics.misc	774
16	talk	talk.politics.guns	910
17	talk	talk.politics.mideast	940
18	talk	talk.religion.misc	628
19	alt	alt.atheism	799
20	soc	soc.religion.christian	997

Of these 18846 documents we have used 11314 documents to train the model and get the prior probabilities of words for each category. Then the rest 7532 documents were used to test the performance of the model, that is, the model will then use the prior information which it observed from the train data and use it to predict the categories of these 7532 documents.

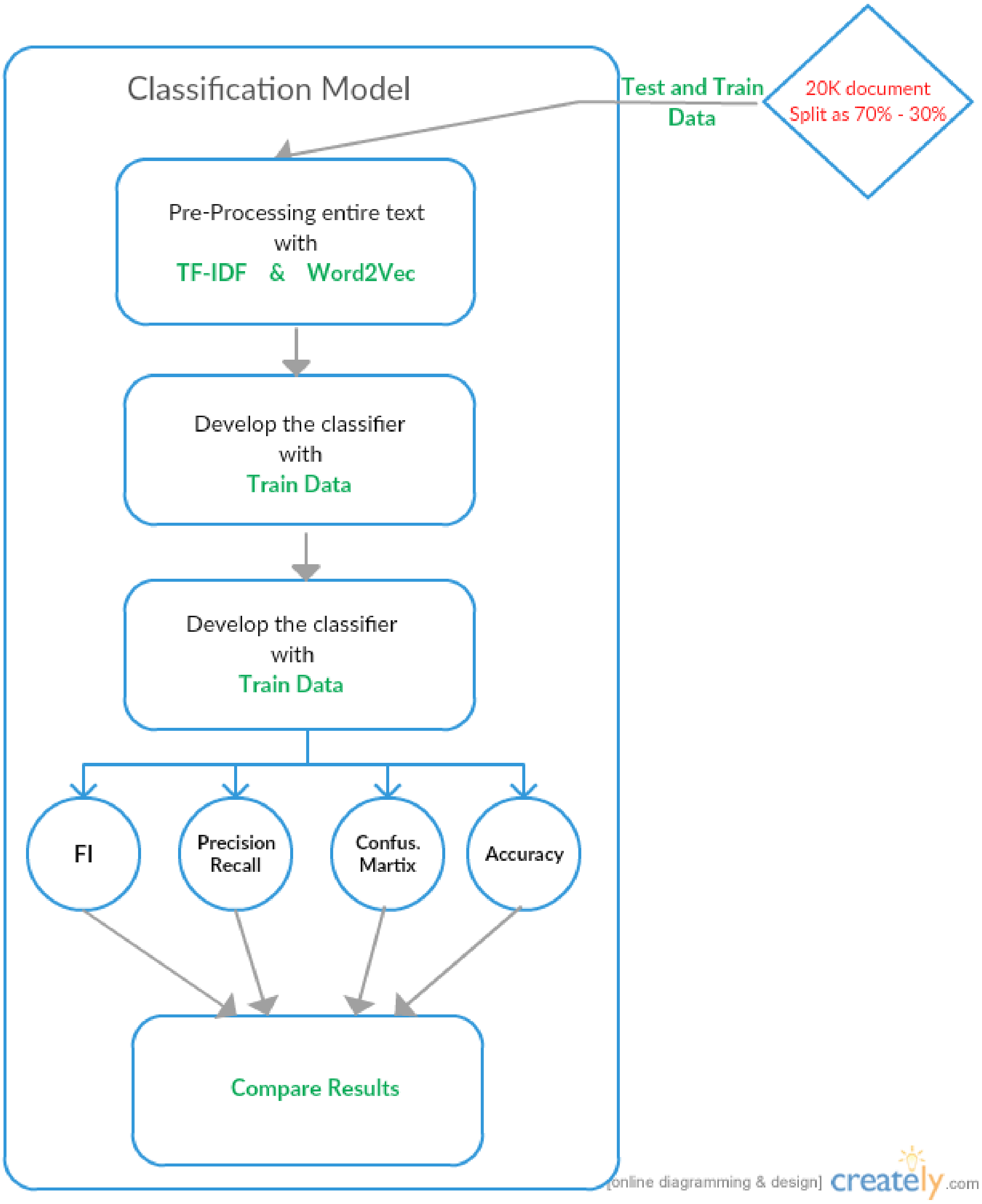
Organization Governing Data

The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale / soc.religion.christian).

3.2 Methodology

3.2.1 Our Project Flowcharts:

The given flow chart was created specifically for this project and the link for the online content will be shared in a metadata file.


Classification Model
Test and Train Data
20K document Split as 70% - 30%
Pre-Processing entire text with TF-IDF & Word2Vec
Develop the classifier with Train Data
Develop the classifier with Train Data
FI
Precision Recall
Confus. Martix
Accuracy
Compare Results
[online diagramming & design] creately.com

This flow diagram depicts the overall structure of the model that we propose that can compare different classification algorithms with various accuracy measures. And as mentioned we have implemented algorithms with the accuracy measures like F1 measure, Precision Recall, Confusion matrix and overall Accuracy and we finally compare all the results and make our ouw recommendation on which algorithm to use at wat instance the user is at a particular point.

3.2.2 Our Algorithms and Associated Formulas:

In this section we define the problem in a programatic way to continue on to establish the algorithms

GIVEN: A description of an instance, $x \in X$, where X is the instance language or instance space. And the problem we are trying to solve here is about how to represent text documents and as mentioned earlier, we followed two approaches that is TFIDF text transformation and Word2Vec text transformation.

A fixed set of categories: $C = \{c_1, c_2, , c_n\}$

TO DETERMINE: The category of $x : c(x) \in C$, where $c(x)$ is a categorization function whose domain is X and whose range is C. We want to know how to build categorization functions (classifiers). So the folowing points summarises the implementation details:

a)Decision Tree classifiers: As mentioned in the previous section, our Decision Tree classifier makes use of **Leaf node** and **Decision node** to compute the relative weights of the nodes and eventually end up having a decision of which class the particular Newsgroup belongs.

b)Random Forest Classifier: In Random forest implementation, our feature weights are normalized for feature subspace sampling. Supposing the correlation between a feature A_i and the class label feature Y is $corr(A_i, Y)$ for {i=1,,N}. We can then define the normalized weight as :

$$w_i = \frac{\sqrt{corr(A_i, Y)}}{\sum_{i=1}^{N} \sqrt{corr(A_i, Y)}} \qquad (ii)$$

The extraction of square root of the correlation is a common technique for smoothing. It can be easily seen that the normalized weight w_i measures the relative informativeness of feature A_i. This weight information will be used in feature subspace sampling when designing our algorithm.

So we came up with the algorithm:

Random Forest: Pseudocode - Function createTree()

```
create a new node $n$;
if stopping criteria is met then
  return $n$ as a leaf node;
else
  for j=1 to j=M do
    compute the informativeness measure corr(Aj,Y) by $Equation(2)$;
  end for
compute feature weights $w1, w_2, ..., w_M$ by Equation (ii);
use the feature weighting method to randomly select m features;
use these m feature as candidates to generate the best split for the node to be partitioned;
call createTree() for each split;
end if
```

```
13: return $n$;
```

c)Naive Bayes Classifier Multinomial: For the Naive Bayes Classifier, we will use the Simplified Bayesian Network defined earlier (fig a). The Classifier will be trained first on the basis of the Training data and will then Predict the Topic Class of all words in the Test data. Following is the overview of both these phases :

Training Phase:

Firstly, using the Training data we will learn Likelihood and Prior beliefs. We can also model our Prior beliefs as being Uninformed.

- **Prior** $P(s_i)$: This is the global Prior Probability of Topic Class. We will use both Informed and Uninformed priors. The informed prior will comprise values of $P(s_i), s_i \in S$ learned from the Training data while the uninformed prior will comprise values of $P(s_i), s_i \in S$ where every Tag is equally likely.
- **Likelihood** $P(w_i|s_i)$**:** We will compute the value of Likelihood from the Training data.

Testing Phase:

Given the values of Prior and Likelihood, the classifiers task is to compute the Posterior probability $P(s|w_i)$, and predict the Unobserved Variable (Topic Class) s_i for input w_i.

- **Posterior** $P(s_i|w_i)$**:** The classifier predicts the most probable Topic Class s_i from the posterior in the following manner:

$$s_i = argmax_{s_i \in S}\ P(s_i|w_i) \qquad (iii)$$

In Equation-(iii), we can write $P(s_i|w_i)$ using Bayes Theorem as follows:

$$P(s_i|w_i) = \frac{P(s_i) * P(w_i|s_i)}{P(w_i)}$$

Since $P(w_i)$ will remain constant, we can rewrite the above equation as:

$$P(s_i|w_i) = P(s_i) * P(w_i|s_i) \qquad (iv)$$

i.e. $Posterior \approx Prior * Likelihood$

Rewriting Equation - (iii) using Equation - (iv), we get:

$$s_i = argmax_{s_i \in S}\ P(s_i) * P(w_i|s_i) \qquad (v)$$

Therefore, using the Naive Bayes Classifier, we can compute the most probable Topic Class $s_i \in S$ given a word w_i in Test Data by formulating the Posterior $P(s_i|w_i)$ on the basis of the Prior and Likelihood probabilities per Equation - (v).

d.)Logistic Regression:

We want to compute probability of Topic Class and we use a framework called logistic regression where **Logistic** refers to a special mathematical function it uses and **Regression** combines a weight vector with observations to create an answer or we can see it as a more general cookbook for building conditional probability distributions. and **Nave Bayes** is a special case of logistic regression.

Notations:

-Weight vector β_i

-Observations X_i

We then define **Bias** as β_0 (like intercept in linear regression)

$$P(Y=0|X) = \frac{1}{1 \ + \ exp[\beta_0 + \sum_i \beta_i X_i]} \qquad (vi)$$

$$P(Y=1|X) = \frac{exp[\beta_0 + \sum_i \beta_i X_i]}{1 \ + \ exp[\beta_0 + \sum_i \beta_i X_i]} \qquad (vii)$$

we can reduce it to,

$$P(Y=0|X) = \alpha(-(\beta_0 + \sum_i \beta_i X_i))$$

$$P(Y=1|X) = 1 \ - \ \alpha(-(\beta_0 + \sum_i \beta_i X_i)) \qquad where \ \alpha = \frac{1}{1 + exp[-Z]}$$

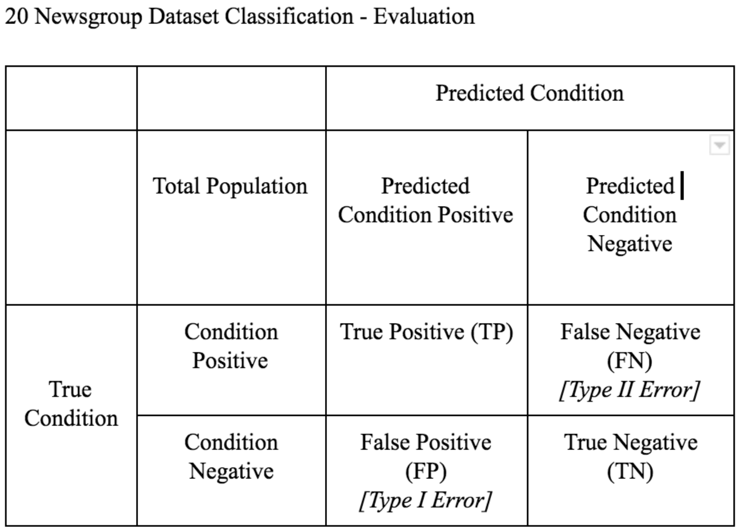
The the above equations (vi) ans (vii) illustrates how logistic regression is used to train a multi-dimensional(n dimentions) classifier for our case of text classification. The training data consists of 20000 news group data. The decision boundaries separate out the data into n classes.

e.)SVM - SGD classifier:

In machine learning, support vector machines (SVMs, also support vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines.

3.2.3 Evaluation Strategy:



20 Newsgroup Dataset Classification - Evaluation

		Predicted Condition	
	Total Population	Predicted Condition Positive	Predicted Condition Negative
True Condition	Condition Positive	True Positive (TP)	False Negative (FN) *[Type II Error]*
	Condition Negative	False Positive (FP) *[Type I Error]*	True Negative (TN)

Performance evaluation

Performance evaluation is one of the main portion of any project. So to test our models developed through various algorithms we use the following ways to evaluate its performance.

1. Simple accuracy. 2. Precision measure. 3. Recall measure. 4. F-1 measure. 5. Confusion matrix.

Simple accuracy: According to ISO 5725-1, accuracy is defined as the closeness of measure to its true value. Mathematically accuracy is defined as follows:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

2. Precision measure:

Precision which is also known as positive predictive value is defined as the probability that a randomly selected document is relevant.

Precision = tp/(tp+fp)

3. Recall Measure: Recall also known as sensitivity is defined as the probability that a randomly selected relevant document is retrieved in the search.

Recall = (tp) / (tp+fn)

4. F-1 measure:

F-1 measure takes into account of both the precision and the recall values by taking their harmonic mean value.

F = 2*(Precision * Recall) / (Precision + Recall)

5. Confusion matrix:

Confusion matrix or error matrix is defined as the table in which portrays the performance of the algorithm. Each column in the matrix represent the instance in the predicted class and each row in the matrix represent the instance in the actual class. Each value in the matrix represents the relation between predicted class and actual class.

4. RESULTS

The Accuracy table shows the primary comparison between the accuracies we obtained from the various algorithms:

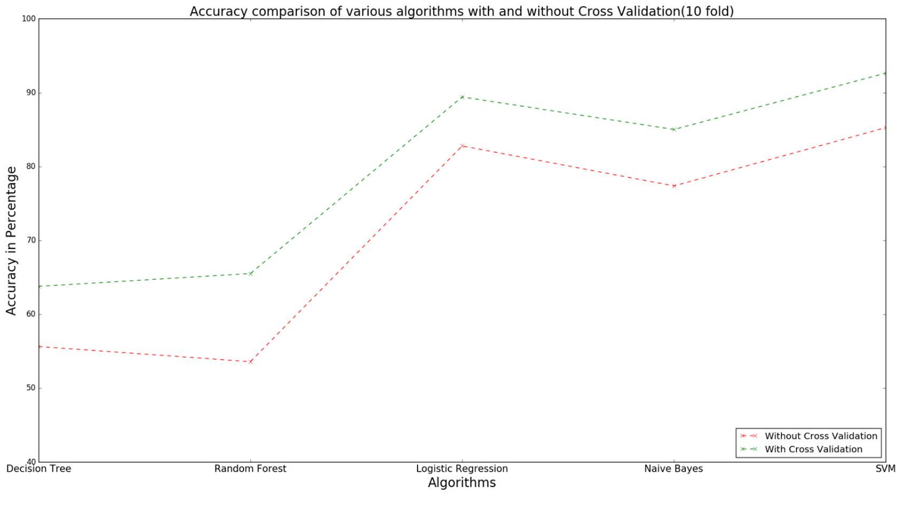
$[CrossValidation - CV]$

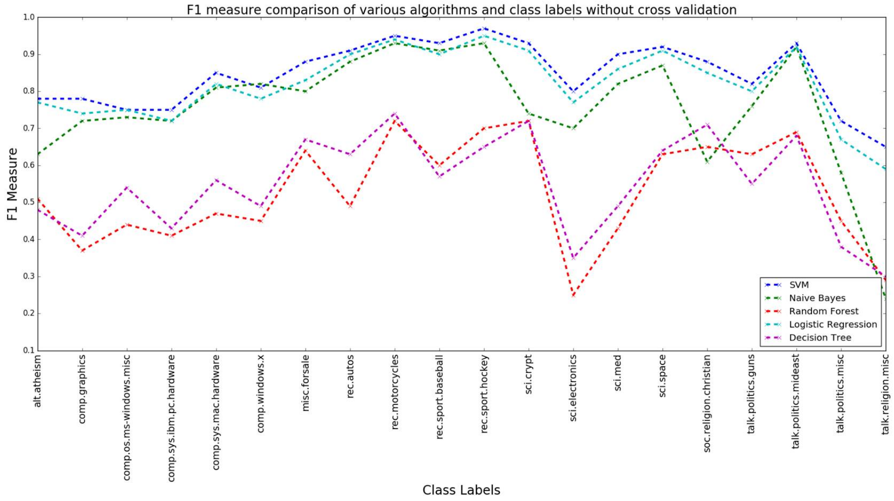
	$Algorithms$	$Accuracy(withCV)$	$Accuracy(withoutCV)$
1	SVM	92.64	85.26
2	Naive Bayes	85.02	77.38
3	Logistic	89.42	82.79
4	Random Forest	65.52	53.59
5	Decision Tree	63.78	55.64

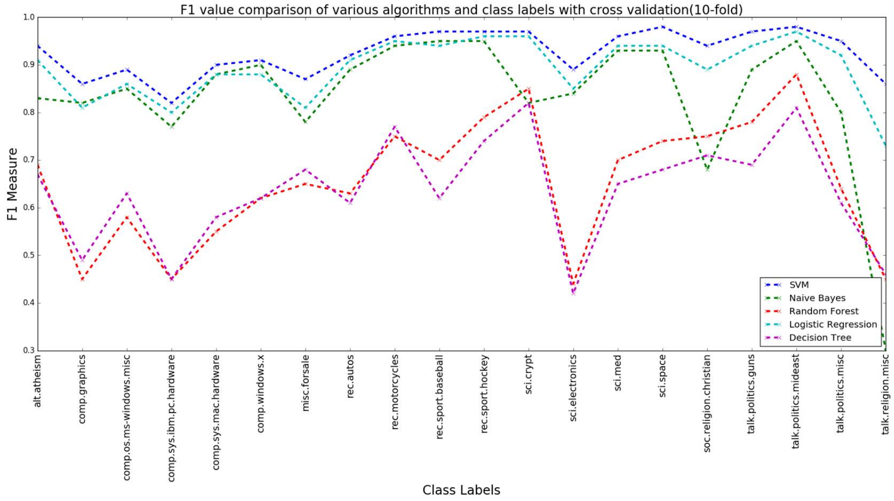
Notations for Results:

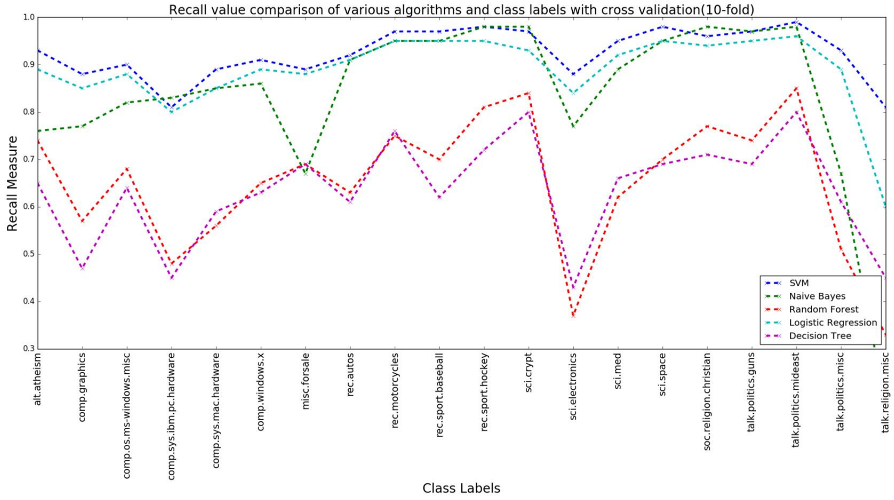
With 10 fold CV: From this table we can observe that algorithms such as SVM, Naive Bayes and logistic regression performs really well on text data compared to algorithms such as decision tree and random forest.

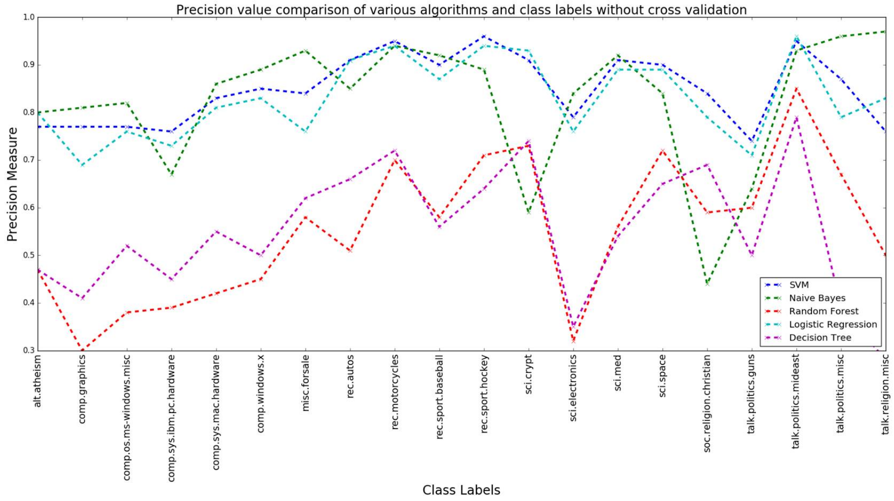
Without CV: We can also see similar pattern in results when we run the model on test data, but we see that cross validation yields more accuracy compared to without CV for all the algorithms.

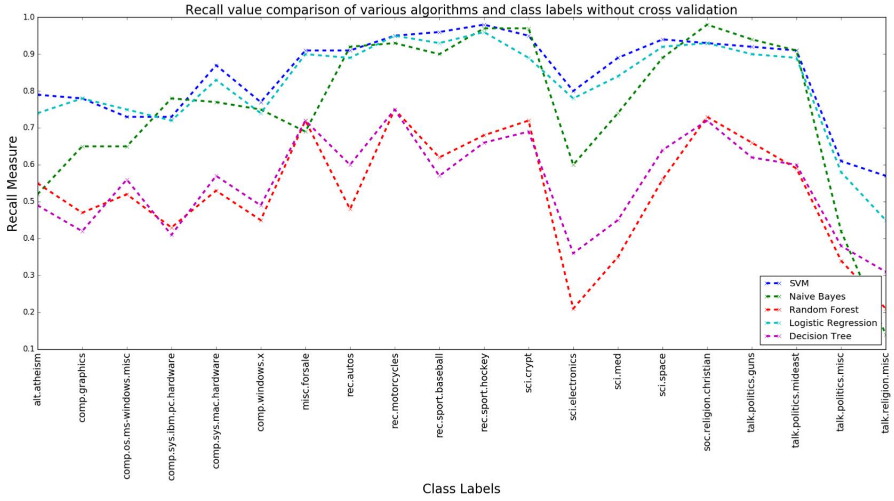


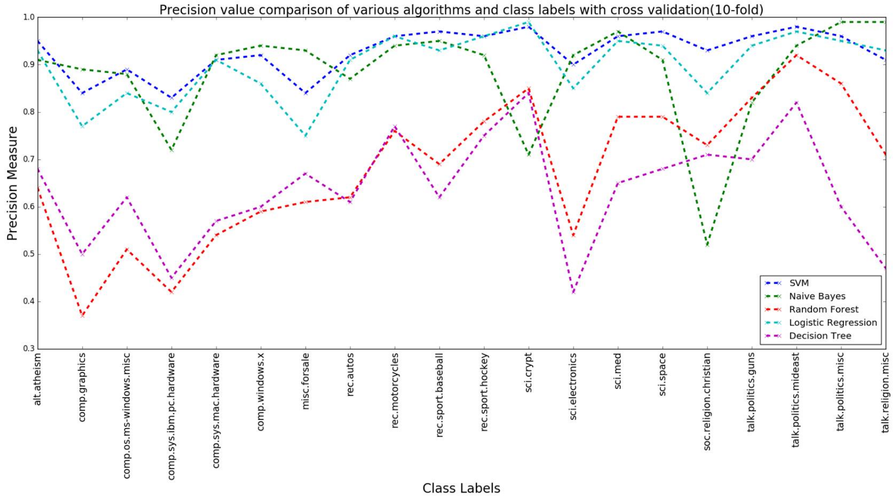

F1 measure comparison of various algorithms and class labels without cross validation


F1 value comparison of various algorithms and class labels with cross validation(10-fold)


Recall value comparison of various algorithms and class labels with cross validation(10-fold)


Precision value comparison of various algorithms and class labels without cross validation


Recall value comparison of various algorithms and class labels without cross validation


Precision value comparison of various algorithms and class labels with cross validation(10-fold)

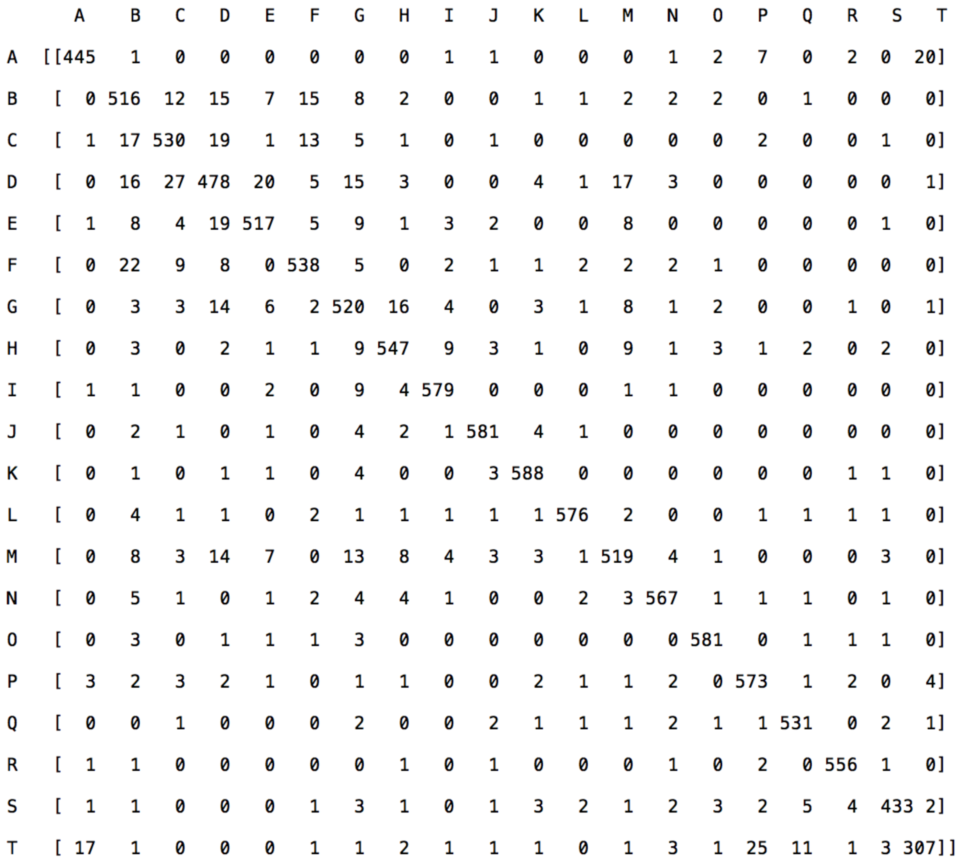
4.1 Analysis of SVM algorithm

With 10-fold CV

```
Accuracy:  92.6462789464
                          precision    recall  f1-score   support


             alt.atheism       0.95      0.93      0.94       480
           comp.graphics       0.84      0.88      0.86       584
 comp.os.ms-windows.misc       0.89      0.90      0.89       591
comp.sys.ibm.pc.hardware       0.83      0.81      0.82       590
   comp.sys.mac.hardware       0.91      0.89      0.90       578
          comp.windows.x       0.92      0.91      0.91       593
            misc.forsale       0.84      0.89      0.87       585
               rec.autos       0.92      0.92      0.92       594
         rec.motorcycles       0.96      0.97      0.96       598
      rec.sport.baseball       0.97      0.97      0.97       597
        rec.sport.hockey       0.96      0.98      0.97       600
               sci.crypt       0.98      0.97      0.97       595
         sci.electronics       0.90      0.88      0.89       591
                 sci.med       0.96      0.95      0.96       594
               sci.space       0.97      0.98      0.98       593
  soc.religion.christian       0.93      0.96      0.94       599
      talk.politics.guns       0.96      0.97      0.97       546
   talk.politics.mideast       0.98      0.99      0.98       564
      talk.politics.misc       0.96      0.93      0.95       465
      talk.religion.misc       0.91      0.81      0.86       377

             avg / total       0.93      0.93      0.93     11314
```

Inference: From this table we can see that SVM has high value for precision, recall and F-1 measure, this shows how good a SVM classifier for text classification.

Confusion Matrix:



```
      A    B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
A  [[445   1   0   0   0   0   0   0   1   1   0   0   0   1   2   7   0   2   0  20]
B   [  0 516  12  15   7  15   8   2   0   0   1   1   2   2   2   0   1   0   0   0]
C   [  1  17 530  19   1  13   5   1   0   1   0   0   0   0   0   2   0   0   1   0]
D   [  0  16  27 478  20   5  15   3   0   0   4   1  17   3   0   0   0   0   0   1]
E   [  1   8   4  19 517   5   9   1   3   2   0   0   8   0   0   0   0   0   1   0]
F   [  0  22   9   8   0 538   5   0   2   1   1   2   2   2   1   0   0   0   0   0]
G   [  0   3   3  14   6   2 520  16   4   0   3   1   8   1   2   0   0   1   0   1]
H   [  0   3   0   2   1   1   9 547   9   3   1   0   9   1   3   1   2   0   2   0]
I   [  1   1   0   0   2   0   9   4 579   0   0   0   1   1   0   0   0   0   0   0]
J   [  0   2   1   0   1   0   4   2   1 581   4   1   0   0   0   0   0   0   0   0]
K   [  0   1   0   1   1   0   4   0   0   3 588   0   0   0   0   0   0   1   1   0]
L   [  0   4   1   1   0   2   1   1   1   1   1 576   2   0   0   1   1   1   1   0]
M   [  0   8   3  14   7   0  13   8   4   3   3   1 519   4   1   0   0   0   3   0]
N   [  0   5   1   0   1   2   4   4   1   0   0   2   3 567   1   1   1   0   1   0]
O   [  0   3   0   1   1   1   3   0   0   0   0   0   0   0 581   0   1   1   1   0]
P   [  3   2   3   2   1   0   1   1   0   0   2   1   1   2   0 573   1   2   0   4]
Q   [  0   0   1   0   0   0   2   0   0   2   1   1   1   2   1   1 531   0   2   1]
R   [  1   1   0   0   0   0   0   1   0   1   0   0   0   1   0   2   0 556   1   0]
S   [  1   1   0   0   0   1   3   1   0   1   3   2   1   2   3   2   5   4  433 2]
T   [ 17   1   0   0   0   1   1   2   1   1   1   0   1   3   1  25  11   1   3 307]]
```

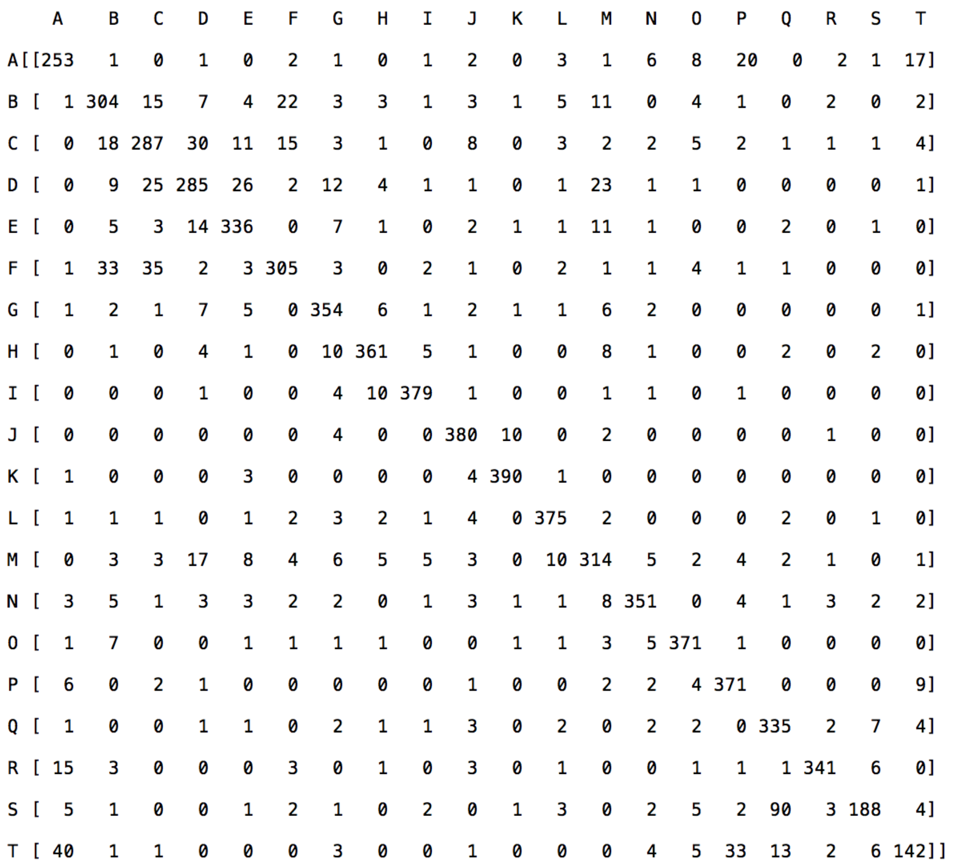
Inference form Confusion Matrix: From the confusion matrix also we can see that the diagonal values are really high and other values are really low for a SVM classifier, this again shows why SVM is one of the best classifiers for text classification.

Results Without CV:

```
Accuracy: 85.2628783856
                          precision    recall  f1-score   support

             alt.atheism       0.77      0.79      0.78       319
           comp.graphics       0.77      0.78      0.78       389
 comp.os.ms-windows.misc       0.77      0.73      0.75       394
comp.sys.ibm.pc.hardware       0.76      0.73      0.75       392
   comp.sys.mac.hardware       0.83      0.87      0.85       385
          comp.windows.x       0.85      0.77      0.81       395
            misc.forsale       0.84      0.91      0.88       390
               rec.autos       0.91      0.91      0.91       396
         rec.motorcycles       0.95      0.95      0.95       398
      rec.sport.baseball       0.90      0.96      0.93       397
        rec.sport.hockey       0.96      0.98      0.97       399
               sci.crypt       0.91      0.95      0.93       396
         sci.electronics       0.79      0.80      0.80       393
                 sci.med       0.91      0.89      0.90       396
               sci.space       0.90      0.94      0.92       394
  soc.religion.christian       0.84      0.93      0.88       398
      talk.politics.guns       0.74      0.92      0.82       364
   talk.politics.mideast       0.95      0.91      0.93       376
      talk.politics.misc       0.87      0.61      0.72       310
      talk.religion.misc       0.76      0.57      0.65       251

             avg / total       0.85      0.85      0.85      7532
```

Inference: From this table also, we can see that SVM performs well on test data as well, this can be concluded from the high values of precision, recall and F-1 measure observed.

Confusion Matrix:



```
      A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
A[[253   1   0   1   0   2   1   0   1   2   0   3   1   6   8  20   0   2   1  17]
B [  1 304  15   7   4  22   3   3   1   3   1   5  11   0   4   1   0   2   0   2]
C [  0  18 287  30  11  15   3   1   0   8   0   3   2   2   5   2   1   1   1   4]
D [  0   9  25 285  26   2  12   4   1   1   0   1  23   1   1   0   0   0   0   1]
E [  0   5   3  14 336   0   7   1   0   2   1   1  11   1   0   0   2   0   1   0]
F [  1  33  35   2   3 305   3   0   2   1   0   2   1   1   4   1   1   0   0   0]
G [  1   2   1   7   5   0 354   6   1   2   1   1   6   2   0   0   0   0   0   1]
H [  0   1   0   4   1   0  10 361   5   1   0   0   8   1   0   0   2   0   2   0]
I [  0   0   0   1   0   0   4  10 379   1   0   0   1   1   0   1   0   0   0   0]
J [  0   0   0   0   0   0   4   0   0 380  10   0   2   0   0   0   0   1   0   0]
K [  1   0   0   0   3   0   0   0   0   4 390   1   0   0   0   0   0   0   0   0]
L [  1   1   1   0   1   2   3   2   1   4   0 375   2   0   0   0   2   0   1   0]
M [  0   3   3  17   8   4   6   5   5   3   0  10 314   5   2   4   2   1   0   1]
N [  3   5   1   3   3   2   2   0   1   3   1   1   8 351   0   4   1   3   2   2]
O [  1   7   0   0   1   1   1   1   0   0   1   1   3   5 371   1   0   0   0   0]
P [  6   0   2   1   0   0   0   0   0   1   0   0   2   2   4 371   0   0   0   9]
Q [  1   0   0   1   1   0   2   1   1   3   0   2   0   2   2   0 335   2   7   4]
R [ 15   3   0   0   0   3   0   1   0   3   0   1   0   0   1   1   1 341   6   0]
S [  5   1   0   0   1   2   1   0   2   0   1   3   0   2   5   2  90   3 188   4]
T [ 40   1   1   0   0   0   3   0   0   1   0   0   0   4   5  33  13   2   6 142]]
```

Inference form Confusion Matrix: From the confusion matrix also we can see that the diagonal values are high and other values are low for a SVM classifier, this again shows why SVM is one of the best classifiers for text classification.

Overview on SVM:

From these results we can say SVM is one of the best candidate for text classification algorithms.

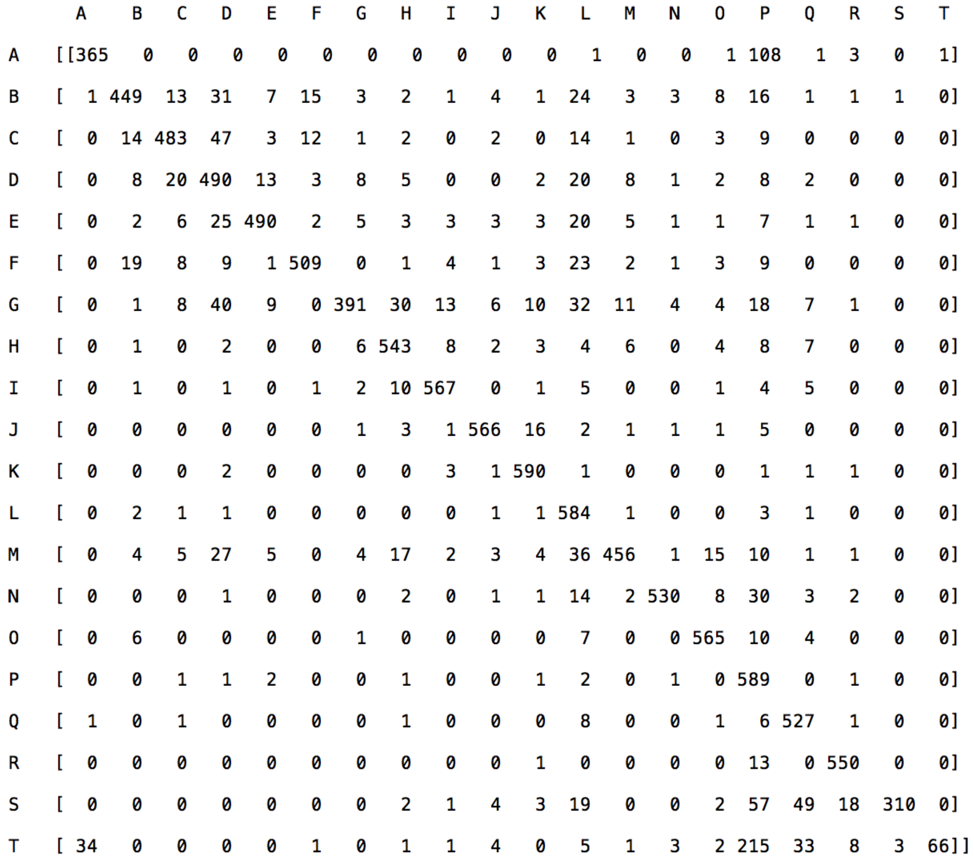
4.2 Analysis of multinomial naive Bayes.

With 10-fold CV

```
Accuracy: 85.0273996818
                          precision    recall  f1-score   support

             alt.atheism       0.91      0.76      0.83       480
           comp.graphics       0.89      0.77      0.82       584
 comp.os.ms-windows.misc       0.88      0.82      0.85       591
comp.sys.ibm.pc.hardware       0.72      0.83      0.77       590
   comp.sys.mac.hardware       0.92      0.85      0.88       578
          comp.windows.x       0.94      0.86      0.90       593
            misc.forsale       0.93      0.67      0.78       585
               rec.autos       0.87      0.91      0.89       594
         rec.motorcycles       0.94      0.95      0.94       598
      rec.sport.baseball       0.95      0.95      0.95       597
        rec.sport.hockey       0.92      0.98      0.95       600
               sci.crypt       0.71      0.98      0.82       595
         sci.electronics       0.92      0.77      0.84       591
                 sci.med       0.97      0.89      0.93       594
               sci.space       0.91      0.95      0.93       593
  soc.religion.christian       0.52      0.98      0.68       599
      talk.politics.guns       0.82      0.97      0.89       546
   talk.politics.mideast       0.94      0.98      0.95       564
      talk.politics.misc       0.99      0.67      0.80       465
      talk.religion.misc       0.99      0.18      0.30       377

             avg / total       0.88      0.85      0.85     11314
```

Inference: From this table we can see that a multinomial naive Bayes classifier performs well on text classification task, the results shows high value of precision for all class labels and low value of recall for talk.religion.misc alone. Overall Naive Bayes proves to be a good classifier.

Confusion Matrix:



```
      A    B    C    D    E    F    G    H    I    J    K    L    M    N    O    P    Q    R    S    T
A  [[365   0    0    0    0    0    0    0    0    0    0    1    0    0    1  108    1    3    0    1]
B  [  1  449   13   31    7   15    3    2    1    4    1   24    3    3    8   16    1    1    1    0]
C  [  0   14  483   47    3   12    1    2    0    2    0   14    1    0    3    9    0    0    0    0]
D  [  0    8   20  490   13    3    8    5    0    0    2   20    8    1    2    8    2    0    0    0]
E  [  0    2    6   25  490    2    5    3    3    3    3   20    5    1    1    7    1    1    0    0]
F  [  0   19    8    9    1  509    0    1    4    1    3   23    2    1    3    9    0    0    0    0]
G  [  0    1    8   40    9    0  391   30   13    6   10   32   11    4    4   18    7    1    0    0]
H  [  0    1    0    2    0    0    6  543    8    2    3    4    6    0    4    8    7    0    0    0]
I  [  0    1    0    1    0    1    2   10  567    0    1    5    0    0    1    4    5    0    0    0]
J  [  0    0    0    0    0    0    1    3    1  566   16    2    1    1    1    5    0    0    0    0]
K  [  0    0    0    2    0    0    0    0    3    1  590    1    0    0    0    1    1    1    0    0]
L  [  0    2    1    1    0    0    0    0    0    1    1  584    1    0    0    3    1    0    0    0]
M  [  0    4    5   27    5    0    4   17    2    3    4   36  456    1   15   10    1    1    0    0]
N  [  0    0    0    1    0    0    0    2    0    1    1   14    2  530    8   30    3    2    0    0]
O  [  0    6    0    0    0    0    1    0    0    0    0    7    0    0  565   10    4    0    0    0]
P  [  0    0    1    1    2    0    0    1    0    0    1    2    0    1    0  589    0    1    0    0]
Q  [  1    0    1    0    0    0    0    1    0    0    0    8    0    0    1    6  527    1    0    0]
R  [  0    0    0    0    0    0    0    0    0    0    1    0    0    0    0   13    0  550    0    0]
S  [  0    0    0    0    0    0    0    2    1    4    3   19    0    0    2   57   49   18  310    0]
T  [ 34    0    0    0    0    1    0    1    1    4    0    5    1    3    2  215   33    8    3   66]]
```

Inference form Confusion Matrix: From the confusion matrix we can see that talk.religion.misc class have been predicted as soc.religion.christian. For this reason we see low value that class alone. This might be because of the high correlation involved between the articles in these classes.

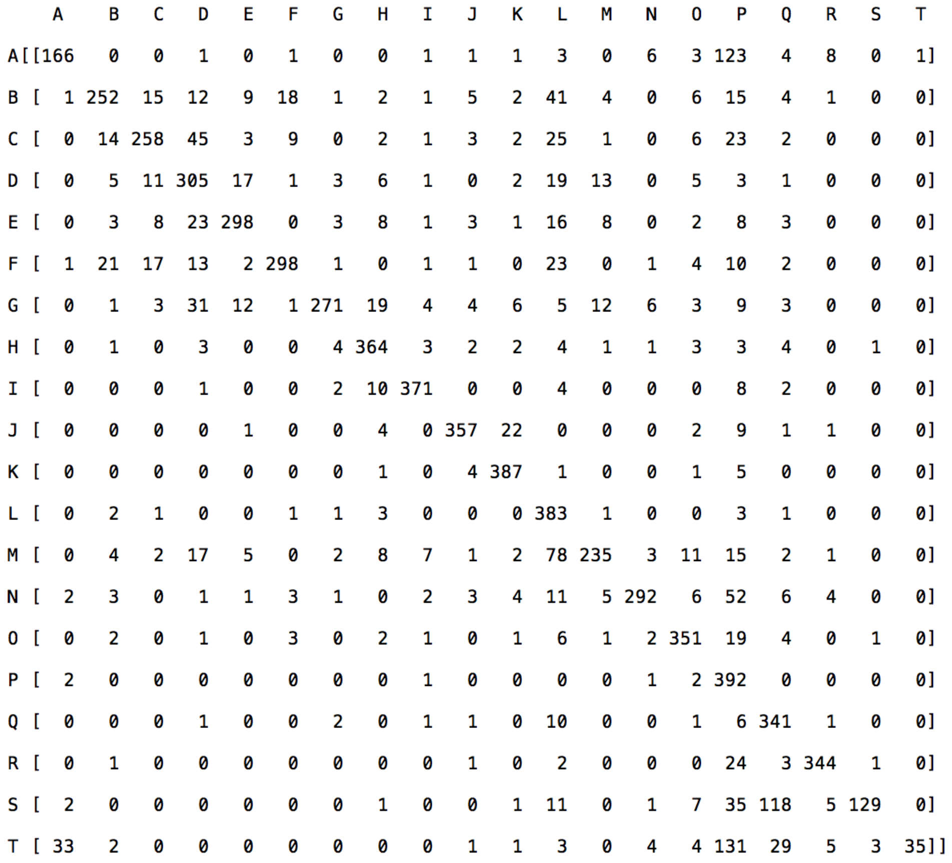
Naive Bayes fails to differentiate these two classes separately, this is where a SVM algorithm stands out.

Results Without CV:

```
Accuracy:  77.389803505
                          precision    recall  f1-score   support

             alt.atheism       0.80      0.52      0.63       319
           comp.graphics       0.81      0.65      0.72       389
 comp.os.ms-windows.misc       0.82      0.65      0.73       394
comp.sys.ibm.pc.hardware       0.67      0.78      0.72       392
   comp.sys.mac.hardware       0.86      0.77      0.81       385
          comp.windows.x       0.89      0.75      0.82       395
            misc.forsale       0.93      0.69      0.80       390
               rec.autos       0.85      0.92      0.88       396
         rec.motorcycles       0.94      0.93      0.93       398
      rec.sport.baseball       0.92      0.90      0.91       397
        rec.sport.hockey       0.89      0.97      0.93       399
               sci.crypt       0.59      0.97      0.74       396
         sci.electronics       0.84      0.60      0.70       393
                 sci.med       0.92      0.74      0.82       396
               sci.space       0.84      0.89      0.87       394
  soc.religion.christian       0.44      0.98      0.61       398
      talk.politics.guns       0.64      0.94      0.76       364
   talk.politics.mideast       0.93      0.91      0.92       376
      talk.politics.misc       0.96      0.42      0.58       310
      talk.religion.misc       0.97      0.14      0.24       251

             avg / total       0.82      0.77      0.77      7532
```

Inference: We see similar kind of results as we observed with cross validation. The class talk.religion.misc is not predicted well by classifier.

Confusion Matrix:



```
    A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
A[[166   0   0   1   0   1   0   0   1   1   1   3   0   6   3 123   4   8   0   1]
B [  1 252  15  12   9  18   1   2   1   5   2  41   4   0   6  15   4   1   0   0]
C [  0  14 258  45   3   9   0   2   1   3   2  25   1   0   6  23   2   0   0   0]
D [  0   5  11 305  17   1   3   6   1   0   2  19  13   0   5   3   1   0   0   0]
E [  0   3   8  23 298   0   3   8   1   3   1  16   8   0   2   8   3   0   0   0]
F [  1  21  17  13   2 298   1   0   1   1   0  23   0   1   4  10   2   0   0   0]
G [  0   1   3  31  12   1 271  19   4   4   6   5  12   6   3   9   3   0   0   0]
H [  0   1   0   3   0   0   4 364   3   2   2   4   1   1   3   3   4   0   1   0]
I [  0   0   0   1   0   0   2  10 371   0   0   4   0   0   0   8   2   0   0   0]
J [  0   0   0   0   1   0   0   4   0 357  22   0   0   0   2   9   1   1   0   0]
K [  0   0   0   0   0   0   0   1   0   4 387   1   0   0   1   5   0   0   0   0]
L [  0   2   1   0   0   1   1   3   0   0   0 383   1   0   0   3   1   0   0   0]
M [  0   4   2  17   5   0   2   8   7   1   2  78 235   3  11  15   2   1   0   0]
N [  2   3   0   1   1   3   1   0   2   3   4  11   5 292   6  52   6   4   0   0]
O [  0   2   0   1   0   3   0   2   1   0   1   6   1   2 351  19   4   0   1   0]
P [  2   0   0   0   0   0   0   0   1   0   0   0   0   1   2 392   0   0   0   0]
Q [  0   0   0   1   0   0   2   0   1   1   0  10   0   0   1   6 341   1   0   0]
R [  0   1   0   0   0   0   0   0   0   1   0   2   0   0   0  24   3 344   1   0]
S [  2   0   0   0   0   0   0   1   0   0   1  11   0   1   7  35 118   5 129   0]
T [ 33   2   0   0   0   0   0   0   0   1   1   3   0   4   4 131  29   5   3  35]]
```

Inference form Confusion Matrix: From the confusion matrix we can see that talk.religion.misc class have been predicted as soc.religion.christian. For this reason we see low value that class alone. This might be because of the high correlation involved between the articles in these classes.

This result is similar to that of with cross validation.

Overview on Naive Bayes classifier:

Overall Naive Bayes classifier performs well on text classification, but it fails sometimes when the two different classes are highly related with each other, like religion as one class and christianity as other one. This is one of the main drawbacks of using Naive Bayes classifier.

4.3 Analysis of Logistic Regression.

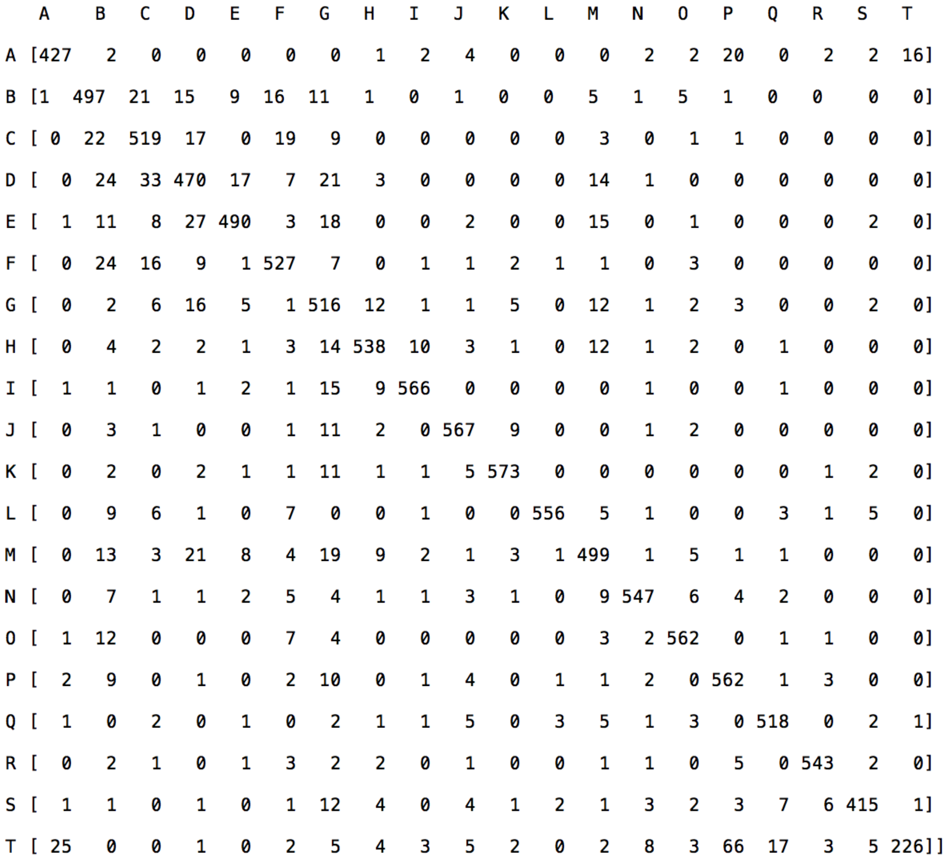
With 10-fold CV

```
Accuracy:  89.4290259855
                          precision    recall  f1-score   support

             alt.atheism       0.93      0.89      0.91       480
           comp.graphics       0.77      0.85      0.81       584
 comp.os.ms-windows.misc       0.84      0.88      0.86       591
comp.sys.ibm.pc.hardware       0.80      0.80      0.80       590
   comp.sys.mac.hardware       0.91      0.85      0.88       578
          comp.windows.x       0.86      0.89      0.88       593
            misc.forsale       0.75      0.88      0.81       585
               rec.autos       0.91      0.91      0.91       594
         rec.motorcycles       0.96      0.95      0.95       598
      rec.sport.baseball       0.93      0.95      0.94       597
        rec.sport.hockey       0.96      0.95      0.96       600
               sci.crypt       0.99      0.93      0.96       595
         sci.electronics       0.85      0.84      0.85       591
                 sci.med       0.95      0.92      0.94       594
               sci.space       0.94      0.95      0.94       593
  soc.religion.christian       0.84      0.94      0.89       599
      talk.politics.guns       0.94      0.95      0.94       546
   talk.politics.mideast       0.97      0.96      0.97       564
      talk.politics.misc       0.95      0.89      0.92       465
      talk.religion.misc       0.93      0.60      0.73       377

             avg / total       0.90      0.89      0.89     11314
```

Inference: From the table we can see that logistic regression performs well when tested with cross validation. Like we saw a problem in predicted talk.religion.misc with Naive Bayes algorithm, we see the same drawback with logistic regression as well.

But logistic regression performs is surely better compared to Naive Bayes algorithm.

Confusion Matrix:



```
   A    B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
A [427   2   0   0   0   0   0   1   2   4   0   0   0   2   2  20   0   2   2  16]
B [1  497  21  15   9  16  11   1   0   1   0   0   5   1   5   1   0   0    0   0]
C [ 0  22  519  17   0  19   9   0   0   0   0   0   3   0   1   1   0   0   0   0]
D [  0  24  33 470  17   7  21   3   0   0   0   0  14   1   0   0   0   0   0   0]
E [  1  11   8  27 490   3  18   0   0   2   0   0  15   0   1   0   0   0   2   0]
F [  0  24  16   9   1 527   7   0   1   1   2   1   1   0   3   0   0   0   0   0]
G [  0   2   6  16   5   1 516  12   1   1   5   0  12   1   2   3   0   0   2   0]
H [  0   4   2   2   1   3  14 538  10   3   1   0  12   1   2   0   1   0   0   0]
I [  1   1   0   1   2   1  15   9 566   0   0   0   0   1   0   0   1   0   0   0]
J [  0   3   1   0   0   1  11   2   0 567   9   0   0   1   2   0   0   0   0   0]
K [  0   2   0   2   1   1  11   1   1   5 573   0   0   0   0   0   0   1   2   0]
L [  0   9   6   1   0   7   0   0   1   0   0 556   5   1   0   0   3   1   5   0]
M [  0  13   3  21   8   4  19   9   2   1   3   1 499   1   5   1   1   0   0   0]
N [  0   7   1   1   2   5   4   1   1   3   1   0   9 547   6   4   2   0   0   0]
O [  1  12   0   0   0   7   4   0   0   0   0   0   3   2 562   0   1   1   0   0]
P [  2   9   0   1   0   2  10   0   1   4   0   1   1   2   0 562   1   3   0   0]
Q [  1   0   2   0   1   0   2   1   1   5   0   3   5   1   3   0 518   0   2   1]
R [  0   2   1   0   1   3   2   2   0   1   0   0   1   1   0   5   0 543   2   0]
S [  1   1   0   1   0   1  12   4   0   4   1   2   1   3   2   3   7   6 415   1]
T [ 25   0   0   1   0   2   5   4   3   5   2   0   2   8   3  66  17   3   5 226]]
```

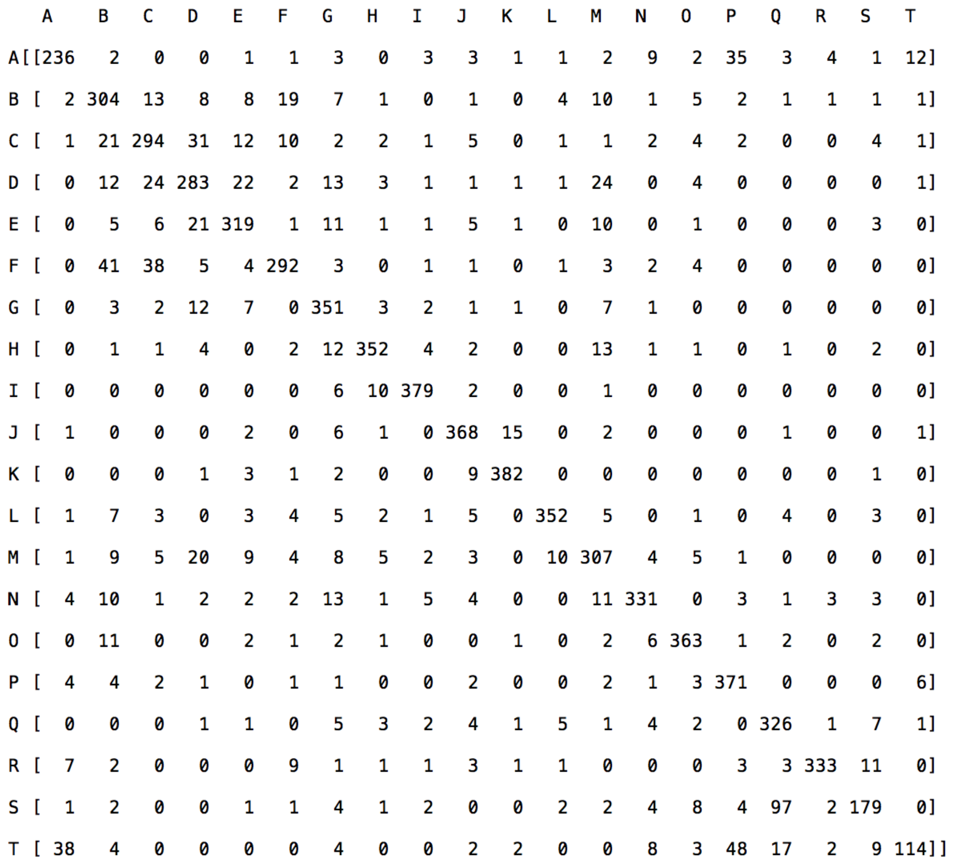
Inference form Confusion Matrix: From the matrix we can see that majority of the classes have been predicted correctly with the logistic regression algorithm.

Results Without CV:

```
Accuracy: 82.7934147637
                          precision    recall  f1-score   support

             alt.atheism       0.80      0.74      0.77       319
           comp.graphics       0.69      0.78      0.74       389
 comp.os.ms-windows.misc       0.76      0.75      0.75       394
comp.sys.ibm.pc.hardware       0.73      0.72      0.72       392
   comp.sys.mac.hardware       0.81      0.83      0.82       385
          comp.windows.x       0.83      0.74      0.78       395
            misc.forsale       0.76      0.90      0.83       390
               rec.autos       0.91      0.89      0.90       396
         rec.motorcycles       0.94      0.95      0.94       398
      rec.sport.baseball       0.87      0.93      0.90       397
        rec.sport.hockey       0.94      0.96      0.95       399
               sci.crypt       0.93      0.89      0.91       396
         sci.electronics       0.76      0.78      0.77       393
                 sci.med       0.89      0.84      0.86       396
               sci.space       0.89      0.92      0.91       394
  soc.religion.christian       0.79      0.93      0.85       398
      talk.politics.guns       0.71      0.90      0.80       364
   talk.politics.mideast       0.96      0.89      0.92       376
      talk.politics.misc       0.79      0.58      0.67       310
      talk.religion.misc       0.83      0.45      0.59       251

             avg / total       0.83      0.83      0.83      7532
```

Inference: From the table we do observe results are similar to that of obtained with cross validation, but with lesser values.

Confusion Matrix:



```
    A    B    C    D    E    F    G    H    I    J    K    L    M    N    O    P    Q    R    S    T
A[[236   2   0   0   1   1   3   0   3   3   1   1   2   9   2  35   3   4   1  12]
B [  2 304  13   8   8  19   7   1   0   1   0   4  10   1   5   2   1   1   1   1]
C [  1  21 294  31  12  10   2   2   1   5   0   1   1   2   4   2   0   0   4   1]
D [  0  12  24 283  22   2  13   3   1   1   1   1  24   0   4   0   0   0   0   1]
E [  0   5   6  21 319   1  11   1   1   5   1   0  10   0   1   0   0   0   3   0]
F [  0  41  38   5   4 292   3   0   1   1   0   1   3   2   4   0   0   0   0   0]
G [  0   3   2  12   7   0 351   3   2   1   1   0   7   1   0   0   0   0   0   0]
H [  0   1   1   4   0   2  12 352   4   2   0   0  13   1   1   0   1   0   2   0]
I [  0   0   0   0   0   0   6  10 379   2   0   0   1   0   0   0   0   0   0   0]
J [  1   0   0   0   2   0   6   1   0 368  15   0   2   0   0   0   1   0   0   1]
K [  0   0   0   1   3   1   2   0   0   9 382   0   0   0   0   0   0   0   1   0]
L [  1   7   3   0   3   4   5   2   1   5   0 352   5   0   1   0   4   0   3   0]
M [  1   9   5  20   9   4   8   5   2   3   0  10 307   4   5   1   0   0   0   0]
N [  4  10   1   2   2   2  13   1   5   4   0   0  11 331   0   3   1   3   3   0]
O [  0  11   0   0   2   1   2   1   0   0   1   0   2   6 363   1   2   0   2   0]
P [  4   4   2   1   0   1   1   0   0   2   0   0   2   1   3 371   0   0   0   6]
Q [  0   0   0   1   1   0   5   3   2   4   1   5   1   4   2   0 326   1   7   1]
R [  7   2   0   0   0   9   1   1   1   3   1   1   0   0   0   3   3 333  11   0]
S [  1   2   0   0   1   1   4   1   2   0   0   2   2   4   8   4  97   2 179   0]
T [ 38   4   0   0   0   0   4   0   0   2   2   0   0   8   3  48  17   2   9 114]]
```

Inference form Confusion Matrix: From the matrix we can see that majority of the classes have been predicted correctly with the logistic regression algorithm.

This result is similar to that of with cross validation.

Overview on Logistic Regression:

Overall we can see that logistic regression algorithm works well for text classification. The performance is not on par with SVM algorithm, but performs well compared to a Naive Bayes classifier.

4.4 Analysis of Random Forest.

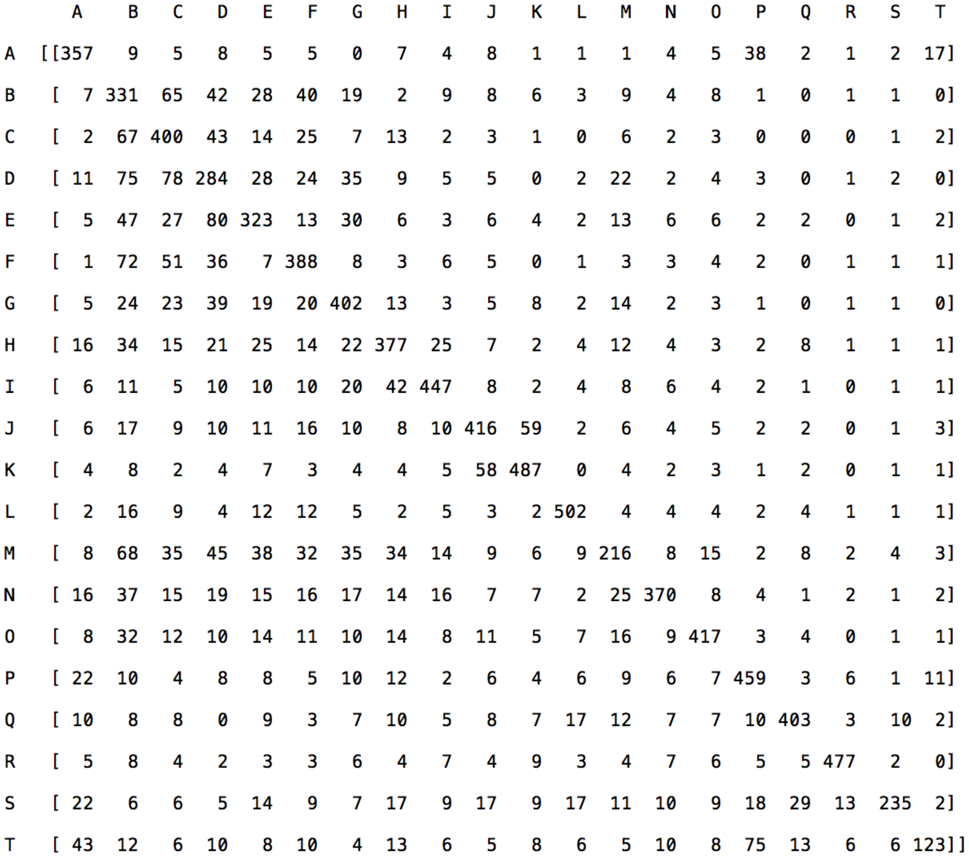
With 10-fold CV

```
Accuracy: 65.5294325614
                          precision    recall  f1-score   support

             alt.atheism       0.64      0.74      0.69       480
           comp.graphics       0.37      0.57      0.45       584
 comp.os.ms-windows.misc       0.51      0.68      0.58       591
comp.sys.ibm.pc.hardware       0.42      0.48      0.45       590
   comp.sys.mac.hardware       0.54      0.56      0.55       578
          comp.windows.x       0.59      0.65      0.62       593
            misc.forsale       0.61      0.69      0.65       585
               rec.autos       0.62      0.63      0.63       594
         rec.motorcycles       0.76      0.75      0.75       598
      rec.sport.baseball       0.69      0.70      0.70       597
        rec.sport.hockey       0.78      0.81      0.79       600
               sci.crypt       0.85      0.84      0.85       595
         sci.electronics       0.54      0.37      0.44       591
                 sci.med       0.79      0.62      0.70       594
               sci.space       0.79      0.70      0.74       593
  soc.religion.christian       0.73      0.77      0.75       599
      talk.politics.guns       0.83      0.74      0.78       546
   talk.politics.mideast       0.92      0.85      0.88       564
      talk.politics.misc       0.86      0.51      0.64       465
      talk.religion.misc       0.71      0.33      0.45       377

             avg / total       0.67      0.66      0.66     11314
```

Inference: From the table, we directly see that random forest classifier performs at an average range for text data, but performs poor when compared to the above three algorithms.

But logistic regression performs is surely better compared to Naive Bayes algorithm.

Confusion Matrix:



```
        A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
A   [[357   9   5   8   5   5   0   7   4   8   1   1   1   4   5  38   2   1   2  17]
B    [  7 331  65  42  28  40  19   2   9   8   6   3   9   4   8   1   0   1   1   0]
C    [  2  67 400  43  14  25   7  13   2   3   1   0   6   2   3   0   0   0   1   2]
D    [ 11  75  78 284  28  24  35   9   5   5   0   2  22   2   4   3   0   1   2   0]
E    [  5  47  27  80 323  13  30   6   3   6   4   2  13   6   6   2   2   0   1   2]
F    [  1  72  51  36   7 388   8   3   6   5   0   1   3   3   4   2   0   1   1   1]
G    [  5  24  23  39  19  20 402  13   3   5   8   2  14   2   3   1   0   1   1   0]
H    [ 16  34  15  21  25  14  22 377  25   7   2   4  12   4   3   2   8   1   1   1]
I    [  6  11   5  10  10  10  20  42 447   8   2   4   8   6   4   2   1   0   1   1]
J    [  6  17   9  10  11  16  10   8  10 416  59   2   6   4   5   2   2   0   1   3]
K    [  4   8   2   4   7   3   4   4   5  58 487   0   4   2   3   1   2   0   1   1]
L    [  2  16   9   4  12  12   5   2   5   3   2 502   4   4   4   2   4   1   1   1]
M    [  8  68  35  45  38  32  35  34  14   9   6   9 216   8  15   2   8   2   4   3]
N    [ 16  37  15  19  15  16  17  14  16   7   7   2  25 370   8   4   1   2   1   2]
O    [  8  32  12  10  14  11  10  14   8  11   5   7  16   9 417   3   4   0   1   1]
P    [ 22  10   4   8   8   5  10  12   2   6   4   6   9   6   7 459   3   6   1  11]
Q    [ 10   8   8   0   9   3   7  10   5   8   7  17  12   7   7  10 403   3  10   2]
R    [  5   8   4   2   3   3   6   4   7   4   9   3   4   7   6   5   5 477   2   0]
S    [ 22   6   6   5  14   9   7  17   9  17   9  17  11  10   9  18  29  13 235   2]
T    [ 43  12   6  10   8  10   4  13   6   5   8   6   5  10   8  75  13   6   6 123]]
```

Inference form Confusion Matrix: From the matrix we can see that random forest does not perform particularly for classes sci.electronics and talk.religion.misc.

We see that talk.religion.misc is predicted as soc.religion.christian because of the high correlation between the articles in them.

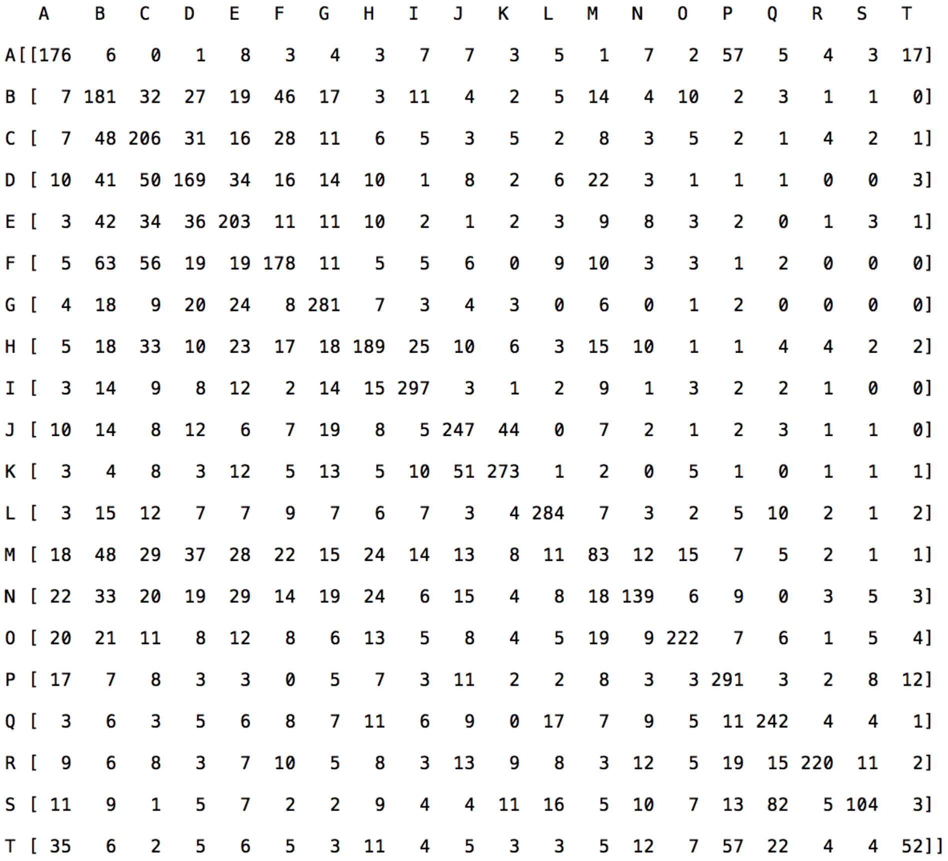
Also the class sci.electronics is predicted as comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware and comp.windows.x. This is again because of the close relations between these classes.

Results Without CV:

```
Accuracy:  53.5979819437
                          precision    recall  f1-score   support

             alt.atheism       0.47      0.55      0.51       319
           comp.graphics       0.30      0.47      0.37       389
 comp.os.ms-windows.misc       0.38      0.52      0.44       394
comp.sys.ibm.pc.hardware       0.39      0.43      0.41       392
   comp.sys.mac.hardware       0.42      0.53      0.47       385
          comp.windows.x       0.45      0.45      0.45       395
            misc.forsale       0.58      0.72      0.64       390
               rec.autos       0.51      0.48      0.49       396
         rec.motorcycles       0.70      0.75      0.72       398
      rec.sport.baseball       0.58      0.62      0.60       397
        rec.sport.hockey       0.71      0.68      0.70       399
               sci.crypt       0.73      0.72      0.72       396
         sci.electronics       0.32      0.21      0.25       393
                 sci.med       0.56      0.35      0.43       396
               sci.space       0.72      0.56      0.63       394
  soc.religion.christian       0.59      0.73      0.65       398
      talk.politics.guns       0.60      0.66      0.63       364
   talk.politics.mideast       0.85      0.59      0.69       376
      talk.politics.misc       0.67      0.34      0.45       310
      talk.religion.misc       0.50      0.21      0.29       251

             avg / total       0.55      0.54      0.53      7532
```

Inference: From the table, we can see that Random Forest algorithms performs poorly. We observe really low values of precision, recall and F-1 measure. From this we can see that Random Forest is one of the poor algorithms for text classification.

Confusion Matrix:



```
    A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
A[[176   6   0   1   8   3   4   3   7   7   3   5   1   7   2  57   5   4   3  17]
B [  7 181  32  27  19  46  17   3  11   4   2   5  14   4  10   2   3   1   1   0]
C [  7  48 206  31  16  28  11   6   5   3   5   2   8   3   5   2   1   4   2   1]
D [ 10  41  50 169  34  16  14  10   1   8   2   6  22   3   1   1   1   0   0   3]
E [  3  42  34  36 203  11  11  10   2   1   2   3   9   8   3   2   0   1   3   1]
F [  5  63  56  19  19 178  11   5   5   6   0   9  10   3   3   1   2   0   0   0]
G [  4  18   9  20  24   8 281   7   3   4   3   0   6   0   1   2   0   0   0   0]
H [  5  18  33  10  23  17  18 189  25  10   6   3  15  10   1   1   4   4   2   2]
I [  3  14   9   8  12   2  14  15 297   3   1   2   9   1   3   2   2   1   0   0]
J [ 10  14   8  12   6   7  19   8   5 247  44   0   7   2   1   2   3   1   1   0]
K [  3   4   8   3  12   5  13   5  10  51 273   1   2   0   5   1   0   1   1   1]
L [  3  15  12   7   7   9   7   6   7   3   4 284   7   3   2   5  10   2   1   2]
M [ 18  48  29  37  28  22  15  24  14  13   8  11  83  12  15   7   5   2   1   1]
N [ 22  33  20  19  29  14  19  24   6  15   4   8  18 139   6   9   0   3   5   3]
O [ 20  21  11   8  12   8   6  13   5   8   4   5  19   9 222   7   6   1   5   4]
P [ 17   7   8   3   3   0   5   7   3  11   2   2   8   3   3 291   3   2   8  12]
Q [  3   6   3   5   6   8   7  11   6   9   0  17   7   9   5  11 242   4   4   1]
R [  9   6   8   3   7  10   5   8   3  13   9   8   3  12   5  19  15 220  11   2]
S [ 11   9   1   5   7   2   2   9   4   4  11  16   5  10   7  13  82   5 104   3]
T [ 35   6   2   5   6   5   3  11   4   5   3   3   5  12   7  57  22   4   4  52]]
```

Inference form Confusion Matrix: From the confusion matrix we can see again why random forest performs poorly. Like we observed with cross validation, we see that classes sci.electronics and talk.religion.misc has the most poor prediction.

Overview on Random Forest:

The random forest performs at a below average level for text classification tasks and performs poorly compared to algorithms such as SVM, Naive Bayes and Logistic Regression.

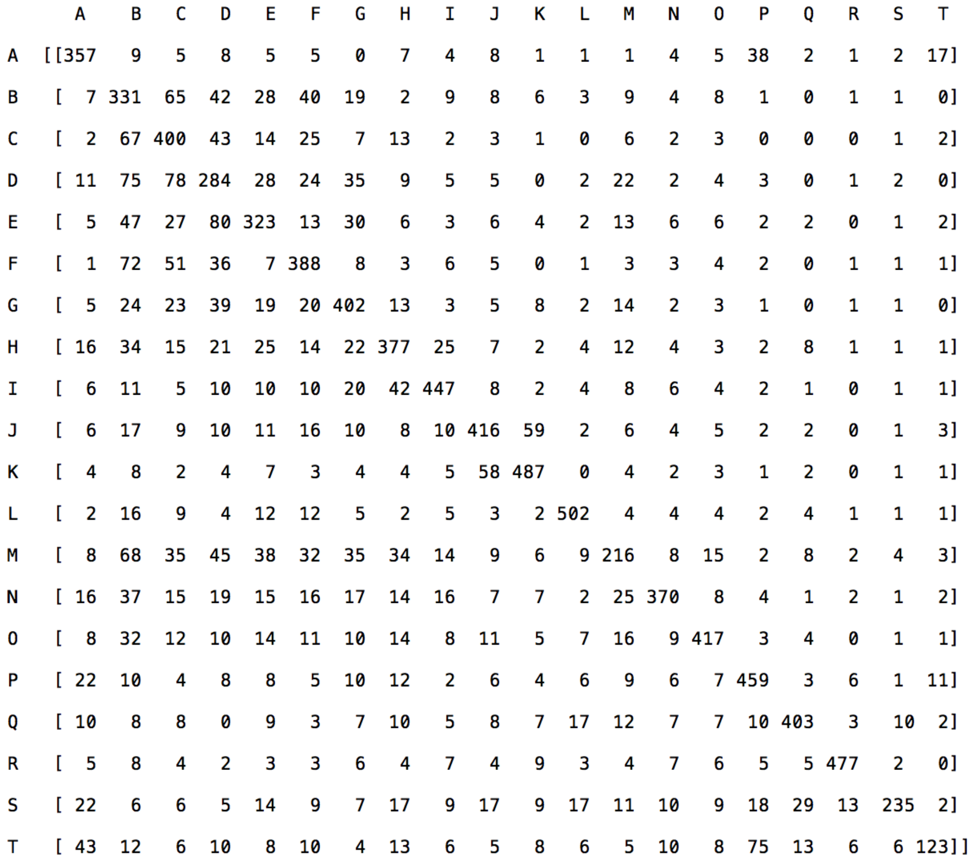
4.5 Analysis of Decision Tree.

With 10-fold CV

```
Accuracy:  65.5294325614
                          precision    recall  f1-score   support


             alt.atheism       0.64      0.74      0.69       480
           comp.graphics       0.37      0.57      0.45       584
 comp.os.ms-windows.misc       0.51      0.68      0.58       591
comp.sys.ibm.pc.hardware       0.42      0.48      0.45       590
   comp.sys.mac.hardware       0.54      0.56      0.55       578
          comp.windows.x       0.59      0.65      0.62       593
            misc.forsale       0.61      0.69      0.65       585
               rec.autos       0.62      0.63      0.63       594
         rec.motorcycles       0.76      0.75      0.75       598
      rec.sport.baseball       0.69      0.70      0.70       597
        rec.sport.hockey       0.78      0.81      0.79       600
               sci.crypt       0.85      0.84      0.85       595
         sci.electronics       0.54      0.37      0.44       591
                 sci.med       0.79      0.62      0.70       594
               sci.space       0.79      0.70      0.74       593
  soc.religion.christian       0.73      0.77      0.75       599
      talk.politics.guns       0.83      0.74      0.78       546
   talk.politics.mideast       0.92      0.85      0.88       564
      talk.politics.misc       0.86      0.51      0.64       465
      talk.religion.misc       0.71      0.33      0.45       377


             avg / total       0.67      0.66      0.66     11314
```

Inference: From the table, we can see that decision tree performs very poorly compared to all the other algorithms, the values of precision, recall and F-1 measure are slightly lower compared to random forest algorithms.

Confusion Matrix:



```
      A    B    C    D    E    F    G    H    I    J    K    L    M    N    O    P    Q    R    S    T
A  [[357    9    5    8    5    5    0    7    4    8    1    1    1    4    5   38    2    1    2   17]
B   [  7  331   65   42   28   40   19    2    9    8    6    3    9    4    8    1    0    1    1    0]
C   [  2   67  400   43   14   25    7   13    2    3    1    0    6    2    3    0    0    0    1    2]
D   [ 11   75   78  284   28   24   35    9    5    5    0    2   22    2    4    3    0    1    2    0]
E   [  5   47   27   80  323   13   30    6    3    6    4    2   13    6    6    2    2    0    1    2]
F   [  1   72   51   36    7  388    8    3    6    5    0    1    3    3    4    2    0    1    1    1]
G   [  5   24   23   39   19   20  402   13    3    5    8    2   14    2    3    1    0    1    1    0]
H   [ 16   34   15   21   25   14   22  377   25    7    2    4   12    4    3    2    8    1    1    1]
I   [  6   11    5   10   10   10   20   42  447    8    2    4    8    6    4    2    1    0    1    1]
J   [  6   17    9   10   11   16   10    8   10  416   59    2    6    4    5    2    2    0    1    3]
K   [  4    8    2    4    7    3    4    4    5   58  487    0    4    2    3    1    2    0    1    1]
L   [  2   16    9    4   12   12    5    2    5    3    2  502    4    4    4    2    4    1    1    1]
M   [  8   68   35   45   38   32   35   34   14    9    6    9  216    8   15    2    8    2    4    3]
N   [ 16   37   15   19   15   16   17   14   16    7    7    2   25  370    8    4    1    2    1    2]
O   [  8   32   12   10   14   11   10   14    8   11    5    7   16    9  417    3    4    0    1    1]
P   [ 22   10    4    8    8    5   10   12    2    6    4    6    9    6    7  459    3    6    1   11]
Q   [ 10    8    8    0    9    3    7   10    5    8    7   17   12    7    7   10  403    3   10    2]
R   [  5    8    4    2    3    3    6    4    7    4    9    3    4    7    6    5    5  477    2    0]
S   [ 22    6    6    5   14    9    7   17    9   17    9   17   11   10    9   18   29   13  235    2]
T   [ 43   12    6   10    8   10    4   13    6    5    8    6    5   10    8   75   13    6    6  123]]
```

Inference form Confusion Matrix: From the matrix we can see similar results that we observed with random forest method.

We see that talk.religion.misc is predicted as soc.religion.christian because of the high correlation between the articles in them.

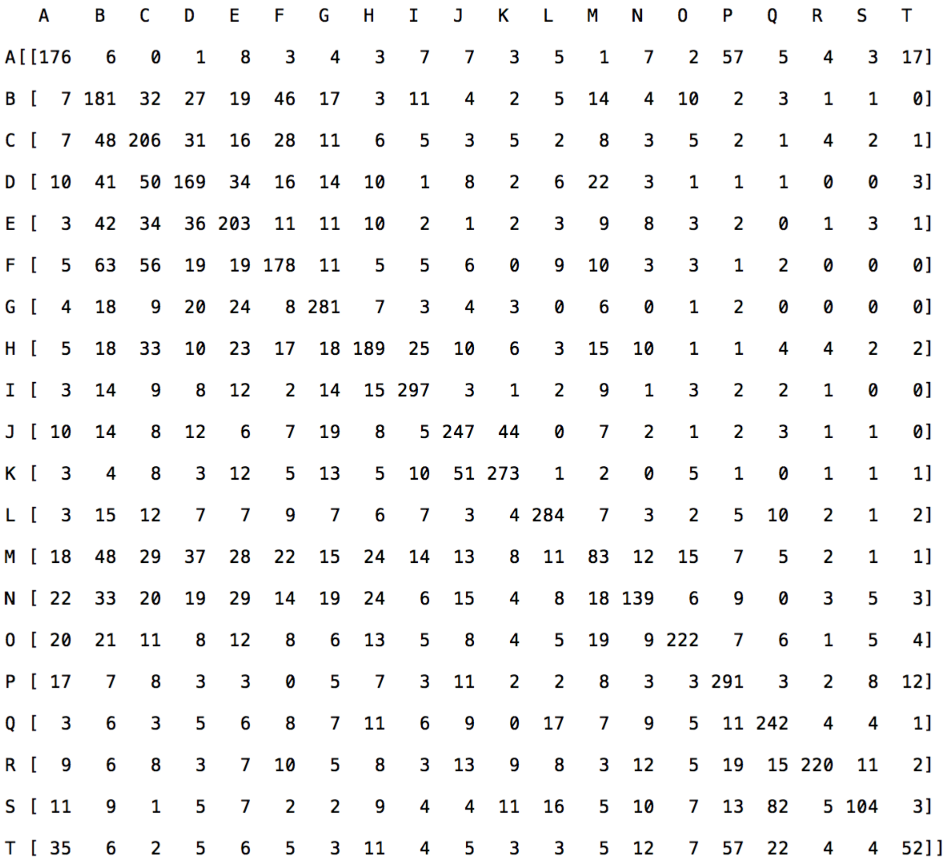
Also the class sci.electronics is predicted as comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware and comp.windows.x. This is again because of the close relations between these classes.

Results Without CV:

```
Accuracy:  53.5979819437
                          precision    recall  f1-score   support

             alt.atheism       0.47      0.55      0.51       319
           comp.graphics       0.30      0.47      0.37       389
 comp.os.ms-windows.misc       0.38      0.52      0.44       394
comp.sys.ibm.pc.hardware       0.39      0.43      0.41       392
   comp.sys.mac.hardware       0.42      0.53      0.47       385
          comp.windows.x       0.45      0.45      0.45       395
            misc.forsale       0.58      0.72      0.64       390
               rec.autos       0.51      0.48      0.49       396
         rec.motorcycles       0.70      0.75      0.72       398
      rec.sport.baseball       0.58      0.62      0.60       397
        rec.sport.hockey       0.71      0.68      0.70       399
               sci.crypt       0.73      0.72      0.72       396
         sci.electronics       0.32      0.21      0.25       393
                 sci.med       0.56      0.35      0.43       396
               sci.space       0.72      0.56      0.63       394
  soc.religion.christian       0.59      0.73      0.65       398
      talk.politics.guns       0.60      0.66      0.63       364
   talk.politics.mideast       0.85      0.59      0.69       376
      talk.politics.misc       0.67      0.34      0.45       310
      talk.religion.misc       0.50      0.21      0.29       251

             avg / total       0.55      0.54      0.53      7532
```

Inference: From the table, with lower values for precision, recall and F-1 measure, we infer that decision tree is not a suitable algorithm for text classification, this can be supported with confusion matrix as well.

Confusion Matrix:



```
     A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
A[[176   6   0   1   8   3   4   3   7   7   3   5   1   7   2  57   5   4   3  17]
B [  7 181  32  27  19  46  17   3  11   4   2   5  14   4  10   2   3   1   1   0]
C [  7  48 206  31  16  28  11   6   5   3   5   2   8   3   5   2   1   4   2   1]
D [ 10  41  50 169  34  16  14  10   1   8   2   6  22   3   1   1   1   0   0   3]
E [  3  42  34  36 203  11  11  10   2   1   2   3   9   8   3   2   0   1   3   1]
F [  5  63  56  19  19 178  11   5   5   6   0   9  10   3   3   1   2   0   0   0]
G [  4  18   9  20  24   8 281   7   3   4   3   0   6   0   1   2   0   0   0   0]
H [  5  18  33  10  23  17  18 189  25  10   6   3  15  10   1   1   4   4   2   2]
I [  3  14   9   8  12   2  14  15 297   3   1   2   9   1   3   2   2   1   0   0]
J [ 10  14   8  12   6   7  19   8   5 247  44   0   7   2   1   2   3   1   1   0]
K [  3   4   8   3  12   5  13   5  10  51 273   1   2   0   5   1   0   1   1   1]
L [  3  15  12   7   7   9   7   6   7   3   4 284   7   3   2   5  10   2   1   2]
M [ 18  48  29  37  28  22  15  24  14  13   8  11  83  12  15   7   5   2   1   1]
N [ 22  33  20  19  29  14  19  24   6  15   4   8  18 139   6   9   0   3   5   3]
O [ 20  21  11   8  12   8   6  13   5   8   4   5  19   9 222   7   6   1   5   4]
P [ 17   7   8   3   3   0   5   7   3  11   2   2   8   3   3 291   3   2   8  12]
Q [  3   6   3   5   6   8   7  11   6   9   0  17   7   9   5  11 242   4   4   1]
R [  9   6   8   3   7  10   5   8   3  13   9   8   3  12   5  19  15 220  11   2]
S [ 11   9   1   5   7   2   2   9   4   4  11  16   5  10   7  13  82   5 104   3]
T [ 35   6   2   5   6   5   3  11   4   5   3   3   5  12   7  57  22   4   4  52]]
```

Overview on Decision Tree:

From these results we can see that decision tree performs similar to random forest, but performs poorly compared to algorithm such as SVM, Naive Bayes and Logistic Regression.

5. Conclusions From the graphs, tables and matrices constructed we can firmly make the following conlusions:

- SVM performs very well for text classification compared to all the other four algorithms.
- Naive Bayes and Logistic regression shows high performance as well, but not as good as SVM classifiers.
- Decision trees and random forests performs similarly and poorly for text classification compared to algorithms SVM, Naive Bayes and Logistic regression.

The reason why SVM works well on text classification task because of the following reasons:

- High dimensional input space: During text classification tasks, we need to handle high number of features, since SVM uses overprotection features, it does not depend on the number of features.
- Document vectors are sparse: Each document when represented as a vector contains only few non-zero entries, it is known that SVM works well with these kind of sparse data.
- Text categorization are linearly separable.

5.1 Further improvements

Building model with neural network, as we know Neural Network is one of the best model, it would be interesting to compare the results of SVM and neural network.

5.2 Future work

- The field of text classification can be particularly used for domain adaptation, that is to train model on one domain and text on others, this is because training models for each data and be really costly.
- Analyze the sentiment of the text, for example given a document X and it is being predicted as Y, we could further analyze about the sentiment of the text such that is the article speaking in favor of class Y or against it.
- Summary extraction: When given an article, we could just extract the summary of the document.

6. Individual Roles:

Anirudh K M: The main take away for me from this project is learning about the various algorithms in details and understanding the various performance measures used. I also learned about the importance of text classification and really inspires me to read more on natural language processing.

Role:

- Literature reading.
- Develop Python scripts to build classifier.
- Evaluated results and inferences for SVM and Logistic Regression
- Some parts of documentation.

Santhosh S: The things that I learnt are about text classification and its wide application. Study of various algorithms and how it works on what data. Also understanding the performance metrics and cross validation.

Role:

- Literature reading.
- Contructing inference from the results obtained.
- Evaluated results and inferences for Naive Bayes and performed bayesian inference from overall probabilities.
- Heavy part of the documentation work.

Praneet Vizappu: The main take away is the understanding of various algorithms such as decision tree, random forest mainly with the cross validation techniques and also understanding the importance of text classification and the challenges involved in that.

Role:

- Literature reading.
- Designing algorithms and their work flows.
- Evaluated results and inferences for Random Forest and Decision Trees.
- Documenation.

References:

1. Text classification and Naive Bayes - https://web.stanford.edu/class/cs124/lec/naivebayes.pdf.

2. A comparison of event models for Naive Bayes Text Classification by Andrew McCallum and Kamal Nigam, Carenegie Mellon University- http://www.cs.cmu.edu/ knigam/papers/multinomial-aaaiws98.pdf

3. Naive Bayes in Nutshell - $https : //www.cs.cmu.edu/\ tom/10701_sp11/slides/GNB_1 - 25 - 2011.pdf$

4. 1. Introduction to Data Mining by Pang-Ning Tan (Author), Michael Steinbach (Author), Vipin Kumar (Author)

5. Scikit learn documentation - http://scikit-learn.org/stable/documentation.html.

6. Text classification with support Vector machines Learning with many relevant features by Thorsten Joachims, Univeristy of Dortmund, $Link : http : //goo.gl/KgJ5Qx$

7. Effective use of word order for text categorization with Random Forest by Rie Johnson and Tong Zhang, Rutgers University, Link: http://goo.gl/S2VlOE

8.High performance feature selection for text classification by Monica Rogati and Yiming Yang, Carnegie Mellon University- $http : //goo.gl/8ON7UL$