EE219 Project-1

Classification Analysis on Textual Data

Sneha Shankar – 404946026

Devanshi Patel – 504945601

Contents

[1. Introduction 3](#_Toc504944199)

[2. Dataset and Problem Statement 3](#_Toc504944200)

[Question (a) 3](#_Toc504944201)

[3. Modeling Text Data and Feature Extraction 4](#_Toc504944202)

[Question (b) 4](#_Toc504944203)

[Question (c) 4](#_Toc504944204)

[4. Feature Selection 5](#_Toc504944205)

[Question (d) Decomposition 5](#_Toc504944206)

[5. Learning Algorithms 6](#_Toc504944207)

[Question (e) Linear Support Vector Machines 6](#_Toc504944208)

[Question (f) 5-fold cross validation 12](#_Toc504944209)

[Question (g) Naive Bayes 15](#_Toc504944210)

[Question (h) Logistic Regression 18](#_Toc504944211)

[Question (i1) Logistic Regression with Regularization 25](#_Toc504944212)

[6. Multiclass Classification 42](#_Toc504944213)

[Question (i2) Naive Bayes and SVM 42](#_Toc504944214)

# 1. Introduction

This project focusses on statistical classification of textual data. The term classification involves categorizing a data set into its respective class based on the known categories. We have performed various classification techniques on the same dataset to draw a definitive comparison about their performances. We analyzed classifiers like Linear Support Vector Machines, Naïve Bayes and Logistic Regression to gain better insight into their working.

# 2. Dataset and Problem Statement

Here, we use the ‘20 Newsgroups’ dataset provided by the scikit-learn package in Python. In this dataset, the documents are categorized into 20 different classes or newsgroups. Some of the classes are closely related to each other and can also be grouped together as a super-category (e.g. comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware can be grouped together into a category of ‘computer technology’).

## Question (a)

In any classification problem, it is very important to have a balanced relative size of the data sets corresponding to different categories. In case of an imbalance, we can either reduce the number of samples in the majority classes to match those in the minority ones or use an appropriate penalty function to assign more weights to the errors of minority classes. Here we have plotted a histogram of the number of documents obtained in each category for 8 such categories. The observations from this histogram eliminates the possibility of an imbalance in the dataset.

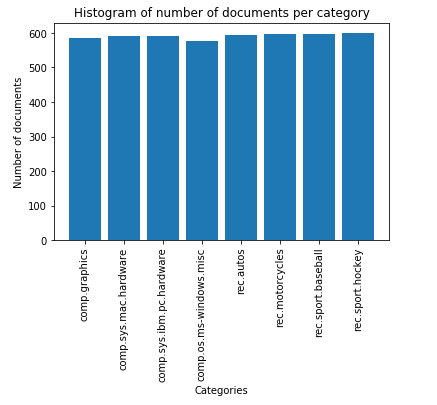


Fig.1 Histogram of Number of Documents per category

# 3. Modeling Text Data and Feature Extraction

## Question (b)

To classify a document, it is very important to represent it properly with its appropriate set of features. This representation should contain only as much relevant information which will avoid the problem of overfitting. The problem of overfitting can be avoided by removing frequently occurring stop words, stemming and also eliminating highly in-frequent words. This also ensures that we do not have to deal with extremely large feature vector which may contain irrelevant information. For this purpose we have tokenized the documents into words and then used the Porter Stemmer to stem the words to their base words and finally we excluded the stop words and punctuations. We have used Python’s NLTK library for the same.

After this pre-processing, it is important to determine the relevance of a word within a document. We have used the TFIDF measure to capture it. Here, TF stands for the ‘Term Frequency’ which denotes the number of times a particular term occurs in a given document. This is used to determine the frequent words. In our project, we set the threshold for words using the min\_df parameter. IDF stands for ‘Inverse Document Frequency’ which is used to denote the relevance of rare words. TFIDF of a particular term is the multiplication of the TF and IDF of that term.

After applying TFIDF transformation on the training dataset, we obtain a document – term matrix in which each row represents a document and each column a term. We have made the following observations by setting min\_df as 2 and 5 respectively.

|  |  |
| --- | --- |
| **Minimum Document Frequency** | **Number of Terms Extracted** |
| 2 | 25342 |
| 5 | 10726 |

This was observed for a set of 4732 documents and with default value of max\_df in the TFIDF transformer.

## Question (c)

Like TFIDF measures the relevance of a word to a particular document, TFICF is a measure which denotes how relevant a word is to a particular class. It is computed by multiplying frequency of a term in a particular class with its inverse class frequency. Here we use all 20 classes of the dataset to determine the 10 most significant terms with respect to the TFICF measure in each of the following classes:

comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, misc.forsale, soc.religion.christian.

The 10 most significant terms observed as tabulated as follows:

1. **For minimum document frequency = 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **comp.sys.ibm.pc.hardware** | **comp.sys.mac.hardware** | **misc.forsale** | **soc.religion.christian** |
| bio | monitor | printer | faith |
| ide | duo | pc | christian |
| monitor | quadra | manual | bibl |
| motherboard | centri | sale | jesu |
| pc | nubu | cd | atho |
| floppi | fpu | packag | truth |
| scsi | mac | forsal | church |
| disk | scsi | ship | scriptur |
| isa | simm | disk | sin |
| jumper | lc | wolverin | christ |

1. **For minimum document frequency = 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **comp.sys.ibm.pc.hardware** | **comp.sys.mac.hardware** | **misc.forsale** | **soc.religion.christian** |
| bio | machin | printer | faith |
| ide | lc | pc | rutger |
| monitor | ram | manual | christian |
| motherboard | fpu | sale | bibl |
| pc | mac | cd | jesu |
| floppi | se | packag | truth |
| scsi | scsi | window | church |
| disk | vram | forsal | scriptur |
| isa | simm | ship | sin |
| jumper | monitor | disk | christ |

# 4. Feature Selection

## Question (d) Decomposition

The extracted TFIDF matrix is a high dimensional matrix which comprises of documents and terms (i.e. rows and columns) in the order of thousands. Such a high dimensional matrix should not be used for classification as learning algorithms generally perform poorly in such a scenario. As a result, LSI dimensionality reduction method is used to get the best rank-k approximation of the original matrix by minimizing the sum of squared errors. This is known as feature selection. The steps followed are:

* Pre-process training and testing datasets
* Apply TFIDF transformation
* Select best features using LSI Decomposition with k=50

The above steps were performed for min\_df=2 and min\_df=5 respectively.

Alternatively, it is also possible to reduce the dimensionality of the original TFIDF matrix using NMF (Non-negative matrix factorization).

We will compare the results of the following sections using both the aforementioned methods.

# 5. Learning Algorithms

From this section onwards, we will be classifying the documents into two broad categories namely ‘Computer Technology’ and ‘Recreational Activity’. Therefore, we have divided the 8 categories into two groups which belong to the previously mentioned categories.

## Question (e) Linear Support Vector Machines

SVM classifies documents based on the sign of vector representation of the document multiplied by the weights that are learnt by the classifier. A positive sign signifies that the document belongs to one class while the one with negative sign belongs to the other. The amount of error the classifier permits for a given data sample is controlled by gamma. There are two types of SVM based on the values of gamma fed to the classifier:

* Hard Margin SVM with large values of 𝛾 (i.e. 𝛾 >> 1) highly penalizes the incorrect classification of documents.
* Soft Margin SVM (when 𝛾 << 1) is very lenient towards incorrect classification of some documents provided that the majority of them are well separated.

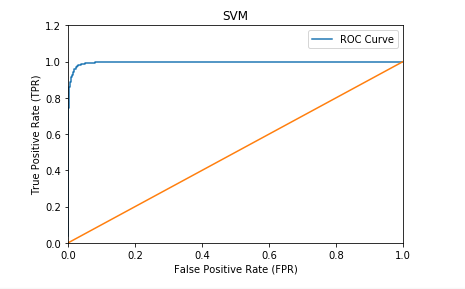
When we use the Linear Kernel to train our classifier on the training dataset, we get the following results.

|  |  |  |  |
| --- | --- | --- | --- |
| **Minimum Document frequency** | 𝛾 | **Type of decomposition** | **Accuracy** |
| 2 | 1000 | SVD | 97.3015873016 |
| NMF | 96.1587301587 |
| 5 | 1000 | SVD | 97.1428571429 |
| NMF | 96.2222222222 |
| 2 | 0.001 | SVD | 50.4761904762 |
| NMF | 50.4761904762 |
| 5 | 0.001 | SVD | 50.4761904762 |
| NMF | 50.4761904762 |

* **min\_df=2;** 𝛾**=1000; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3015873016 |
| Precision | 97.3252190977 |
| Recall | 97.2925737784 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1503 | 57 |
| **Actual: Recreational Activity** | 28 | 1562 |

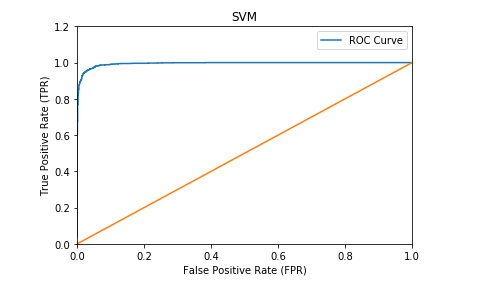


The ROC curve is created by plotting True Positive Rate(TPR) against False Positive Rate(FPR) for different values of threshold. The ideal ROC curve has an area of 1 hence, if a ROC curve shows classes having area of approximately 1, it indicates that the test data is correctly classified.

* **min\_df=2;** 𝛾=**1000; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 96.1587301587 |
| Precision | 96.1719683027 |
| Recall | 96.1520319303 |

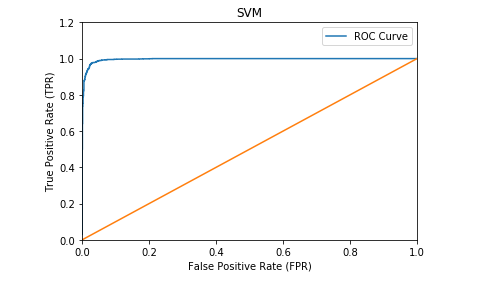
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1489 | 71 |
| **Actual: Recreational Activity** | 50 | 1540 |



* **min\_df=5;** 𝛾**=1000; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.1428571429 |
| Precision | 97.1576956958 |
| Recall | 97.1359458152 |

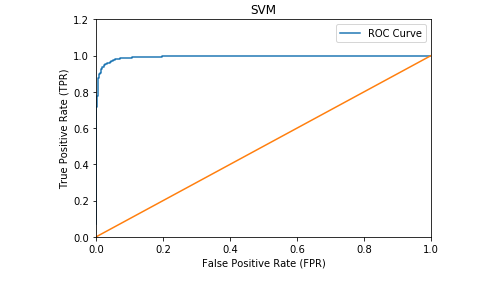
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1504 | 56 |
| **Actual: Recreational Activity** | 34 | 1556 |



* **min\_df=5;** 𝛾**=1000; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| **Accuracy** | 96.2222222222 |
| **Precision** | 96.2283670606 |
| **Recall** | 96.2179487179 |

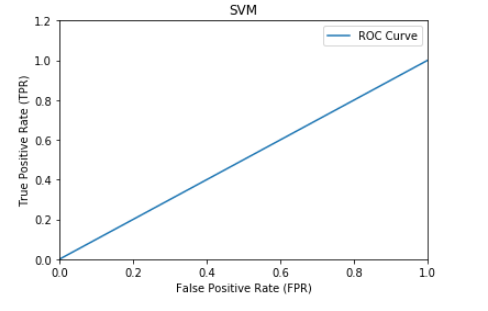
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1494 | 66 |
| **Actual: Recreational Activity** | 53 | 1537 |



* **min\_df=2;** 𝛾**=0.001; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 50.4761904762 |
| Precision | 25.2380952381 |
| Recall | 50.0 |

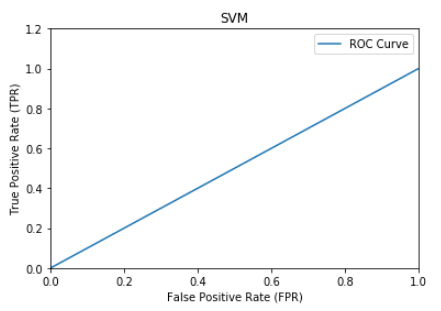
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 0 | 1560 |
| **Actual: Recreational Activity** | 0 | 1590 |



* **min\_df=2;** 𝛾**=0.001; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 50.4761904762 |
| Precision | 25.2380952381 |
| Recall | 50.0 |

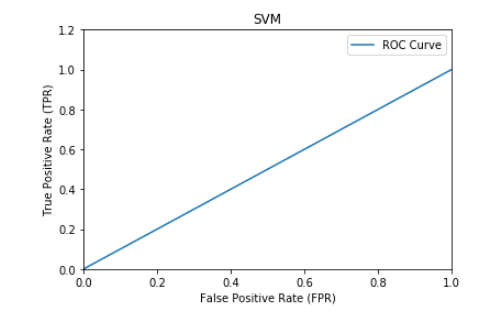
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 0 | 1560 |
| **Actual: Recreational Activity** | 0 | 1590 |



* **min\_df=5;** 𝛾**=0.001; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 50.4761904762 |
| Precision | 25.2380952381 |
| Recall | 50.0 |

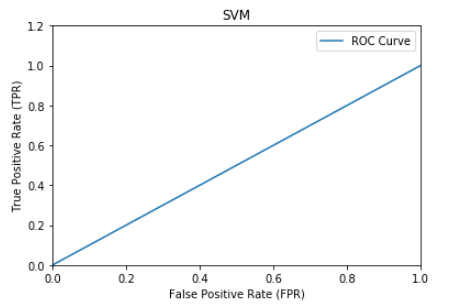
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 0 | 1560 |
| **Actual: Recreational Activity** | 0 | 1590 |



* **min\_df=5;** 𝛾**=0.001; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 50.4761904762 |
| Precision | 25.2380952381 |
| Recall | 50.0 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 0 | 1560 |
| **Actual: Recreational Activity** | 0 | 1590 |



**Observations:**

By observing the above statistics, we can say that we get the best accuracy using SVD decomposition with min\_df=2. Overall, decomposition by SVD gives better accuracy than that of NMF for same values of min\_df and 𝛾.

Hence it is proved that Linear Support Vector Machines are efficient when dealing with textual data which involves high dimensionality and sparsity.

## Question (f) 5-fold cross validation

Cross-validation is a useful technique that helps in assessing and improving the performance of a model. It also helps to check for overfitting and to determine whether a model will generalize well to unseen data.

In this part, we use 5-fold cross validation to obtain the best value for the parameter 𝛾. **The best value** obtained for min\_df=2 and 5 withboth SVD and NMF decomposition is **100**. We get the following results (approximated to 2 digits) by trying different values of 𝛾 and clearly we observe that the **best accuracy was obtained for 𝛾=100 (i.e. k=2)**

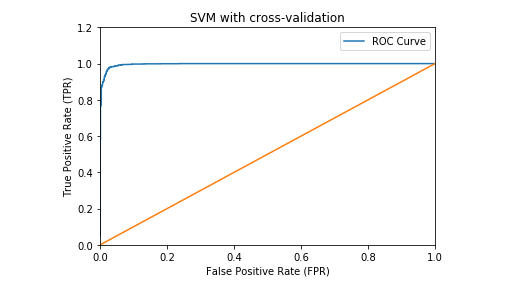
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Minimum Document Frequency** | **Type of Decomposition** | **Accuracy** | | | | | | |
| 𝛾**=0.001** | 𝛾**=0.01** | 𝛾 **=0.1** | 𝛾 **=0** | 𝛾**=10** | 𝛾 **=100** | 𝛾 **=1000** |
| 2 | SVD | 50.48 | 51.39 | 96.06 | 97.01 | 97.3 | 97.43 | 97.30 |
| NMF | 50.48 | 50.48 | 50.47 | 93.52 | 95.4 | 96.10 | 96.15 |
| 5 | SVD | 50.48 | 53.20 | 95.96 | 97.11 | 97.4 | 97.27 | 97.14 |
| NMF | 50.48 | 50.48 | 61.61 | 94.28 | 95.3 | 95.78 | 96.22 |

Following are the results obtained for the best value of 𝛾 chosen i.e. 100

* **min\_df=2; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.4285714286 |
| Precision | 97.4447004878 |
| Recall | 97.4213836478 |

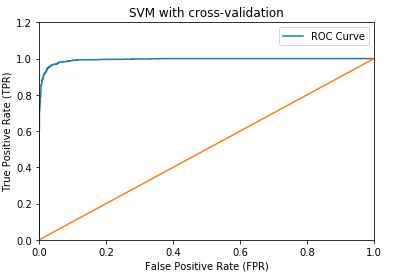
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1508 | 52 |
| **Actual: Recreational Activity** | 29 | 1561 |



* **min\_df=2; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 96.0634920635 |
| Precision | 96.0848418736 |
| Recall | 96.0546686018 |

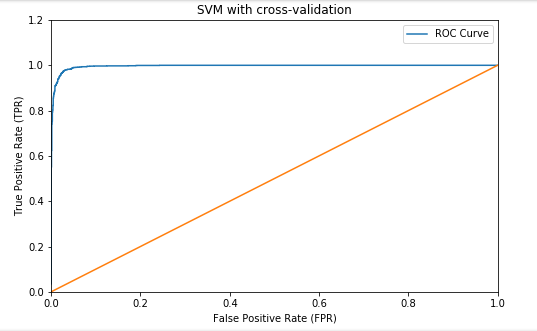
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1484 | 76 |
| **Actual: Recreational Activity** | 48 | 1542 |



* **min\_df=5; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.2698412698 |
| Precision | 97.2870440288 |
| Recall | 97.2623367199 |

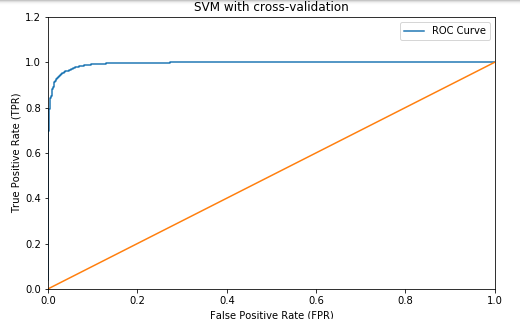
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1505 | 55 |
| **Actual: Recreational Activity** | 31 | 1559 |



* **min\_df=5; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 95.7777777778 |
| Precision | 95.7952219118 |
| Recall | 95.7698355104 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1481 | 79 |
| **Actual: Recreational Activity** | 54 | 1536 |



**Observations:**

Thus, we observe that the best accuracy is obtained for min\_df=2 with SVD decomposition and 𝛾=100.

## Question (g) Naive Bayes

Next, we perform the same classification using Naive Bayes algorithm. This algorithm uses Bayes rule to estimate the maximum likelihood probability of a class based on given feature set of a document. Here we are assuming that the features are statistically independent given the class.

We get the following results upon using Multinomial Naive Bayes classifier. Since Multinomial NB does not work with SVD decomposition because of the presence of negative values, we have used two approaches for this problem:

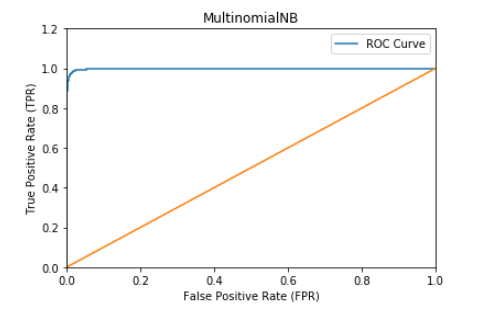
* Without decomposition (using the original TFIDF matrix to predict)
* With decomposition by NMF

|  |  |  |
| --- | --- | --- |
| **Minimum Document Frequency** | **Type of Decomposition** | **Accuracy** |
| 2 | None | 98.1587301587 |
| NMF | 93.5873015873 |
| 5 | None | 98.3174603175 |
| NMF | 94.380952381 |

* **min\_df=2; without decomposition**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 98.1587301587 |
| Precision | 98.1874310043 |
| Recall | 98.1488872762 |

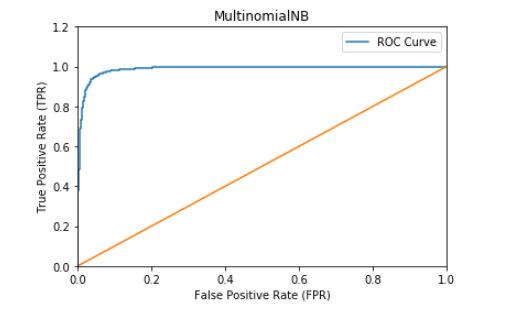
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1515 | 45 |
| **Actual: Recreational Activity** | 13 | 1577 |



* **min\_df=2; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 93.5873015873 |
| Precision | 94.02054292 |
| Recall | 93.5413642961 |

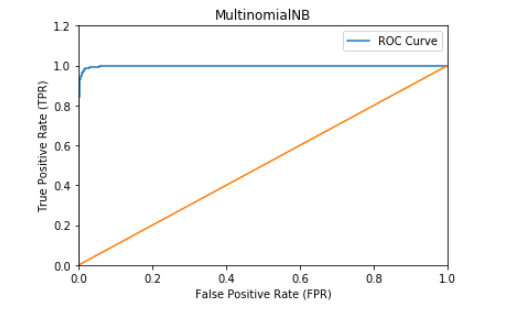
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1384 | 176 |
| **Actual: Recreational Activity** | 26 | 1564 |



* **min\_df=5; without decomposition**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 98.3174603175 |
| Precision | 98.3192049108 |
| Recall | 98.3157958394 |

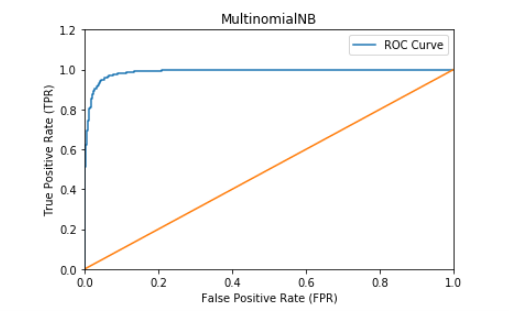
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1531 | 29 |
| **Actual: Recreational Activity** | 24 | 1566 |



* **min\_df=5; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 94.380952381 |
| Precision | 94.637070806 |
| Recall | 94.3462747944 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1415 | 145 |
| **Actual: Recreational Activity** | 32 | 1558 |



## Question (h) Logistic Regression

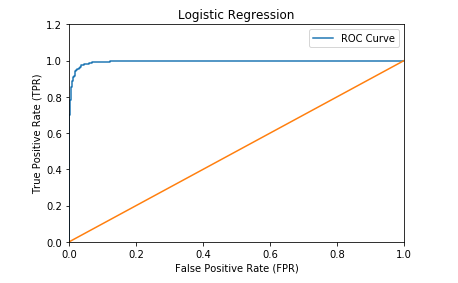
Next we use Logistic Regression classifier for the same task. Logistic Regression is used to obtain the best fitting model that describes the relationship between the categorical outcome variable and one or more independent variables by using a logistic function to estimate the probabilities. We get the following results when using a LR classifier (default). Since the default LR classifier object operates with L2 regularization, following results are tabulated for Logistic Regression with L2 regularization.

|  |  |  |
| --- | --- | --- |
| **Minimum Document Frequency** | **Type of Decomposition** | **Accuracy** |
| 2 | SVD | 96.6666666667 |
| NMF | 93.9365079365 |
| 5 | SVD | 96.7936507937 |
| NMF | 93.7777777778 |

* **min\_df=2; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 96.6666666667 |
| Precision | 96.6955249138 |
| Recall | 96.6563860668 |

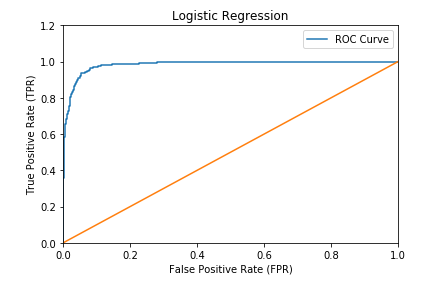
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| Actual: Computer Technology | 1491 | 69 |
| Actual: Recreational Activity | 36 | 1554 |



* **min\_df=2; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 93.9365079365 |
| Precision | 94.0650476234 |
| Recall | 93.9120706338 |

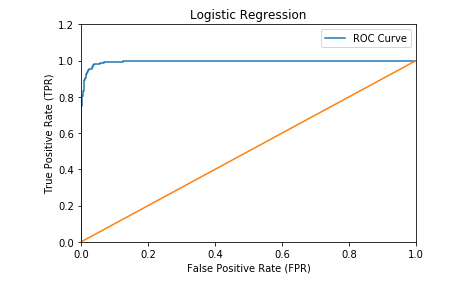
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1425 | 135 |
| **Actual: Recreational Activity** | 56 | 1534 |



* **min\_df=5; with SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 96.7936507937 |
| Precision | 96.8289747736 |
| Recall | 96.7821722303 |

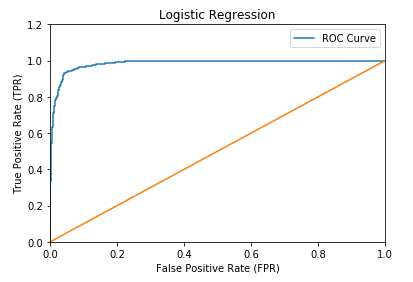
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1491 | 69 |
| **Actual: Recreational Activity** | 32 | 1558 |



* **min\_df=5; with NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 93.7777777778 |
| Precision | 93.9088040136 |
| Recall | 93.7530237059 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1422 | 138 |
| **Actual: Recreational Activity** | 58 | 1532 |



**Observations:**

Here, we observe that SVD decomposition performs better as compared to NMF and the best accuracy is obtained with min\_df=5 using SVD decomposition. We also observe from the above ROC curves that the area is similar to the one in SVM.

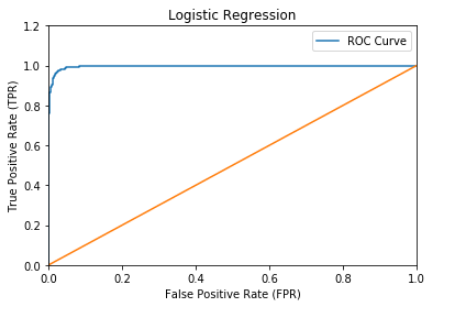
Now, since the default classifier works with L2 regularization, we changed the C to a very large value (C=10000) and recorded the following observations:

|  |  |  |
| --- | --- | --- |
| **Minimum Document Frequency** | **Type of Decomposition** | **Accuracy** |
| 2 | SVD | 97.3968253968 |
| NMF | 96.2857142857 |
| 5 | SVD | 97.3333333333 |
| NMF | 96.0000000000 |

* **min\_df=2; SVD decomposition**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3968253968 |
| Precision | 97.4141060641 |
| Recall | 97.3893323657 |

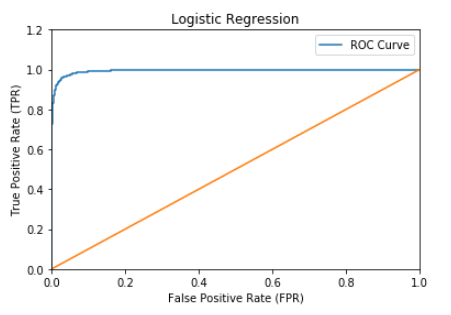
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1507 | 53 |
| **Actual: Recreational Activity** | 29 | 1561 |



* **min\_df=2; NMF decomposition**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 96.2857142857 |
| Precision | 96.2970095613 |
| Recall | 96.2796323174 |

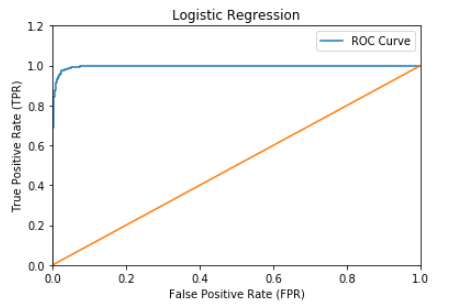
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1492 | 68 |
| **Actual: Recreational Activity** | 49 | 1541 |



* **min\_df=5; SVD decomposition**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3333333333 |
| Precision | 97.3461321287 |
| Recall | 97.3270440252 |

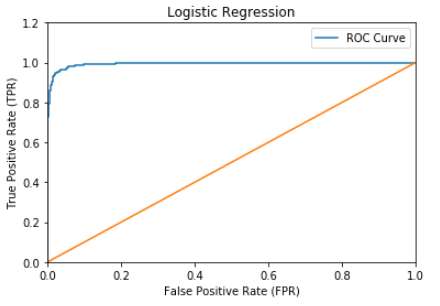
|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1508 | 52 |
| **Actual: Recreational Activity** | 32 | 1558 |



* **min\_df=5; NMF decomposition**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 96.0 |
| Precision | 96.0068296271 |
| Recall | 95.9954039671 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted: Computer Technology** | **Predicted: Recreational Activity** |
| **Actual: Computer Technology** | 1490 | 70 |
| **Actual: Recreational Activity** | 56 | 1534 |



**Observations:**

By overcoming the effect of L2 regularization (by setting C to very large value), we find that SVD decomposition still gives better results than NMF. The best value of accuracy is found for min\_df=2 using SVD decomposition.

## Question (i1) Logistic Regression with Regularization

In this part, we add a regularization term to optimize the LR classifier and to improve generalization performance. We use both L1 and L2 norm regularizations and vary the values of regularization coefficients from 0.001 to 1000 to obtain the results as shown below.

The combined results obtained by performing **SVD decomposition** are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Min\_df** | **k** | **L1 Regularization** | | **L2 Regularization** | |
| **Testing Error** | **Mean of Coeff** | **Testing Errors** | **Mean of Coeff** |
| 2 | 0.001 | 50.48 | 0.0 | 31.36 | -0.0 |
| 0.01 | 10.57 | -0.1 | 5.9 | -0.0 |
| 0.1 | 5.23 | -0.35 | 4.12 | 0.01 |
| 10 | 2.67 | -0.41 | 2.73 | 0.14 |
| 100 | 2.60 | -0.39 | 2.47 | -0.06 |
| 1000 | 2.60 | -0.39 | 2.60 | -0.31 |
| 5 | 0.001 | 50.47 | 0.0 | 26.82 | -0.00 |
| 0.01 | 9.77 | -0.12 | 5.74 | -0.02 |
| 0.1 | 5.27 | -0.94 | 4.12 | -0.13 |
| 10 | 2.60 | -1.32 | 2.60 | -0.59 |
| 100 | 2.67 | -1.60 | 2.57 | -1.04 |
| 1000 | 2.67 | -1.5 | 2.69 | -1.5 |

Thus, we can see from above table that for k=100, the test error stabilizes for min\_df=2 and min\_df=5 both. The best value of test error (i.e. least) using l1 regularization is observed for k=100 in case of min\_df=2 and for k=10 for min\_df=5. While, the best value of test error (i.e. least) using l2 regularization is observed for k=100 for both values of min\_df.

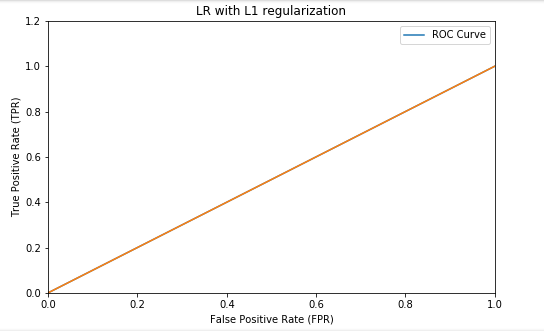
The results obtained by doing **NMF decomposition** were as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Min\_df** | **k** | **L1 Regularization** | | **L2 Regularization** | |
| Testing Error | Mean of Coeff | Testing Errors | Mean of Coeff |
| 2 | 0.001 | 50.48 | 0.0 | 49.52 | -0.0 |
| 0.01 | 50.47 | 0.0 | 48.98 | -0.0 |
| 0.1 | 31.3 | -0.07 | 12.22 | -0.02 |
| 10 | 3.59 | -6.9 | 4.92 | -0.78 |
| 100 | 3.71 | -13.63 | 4.16 | -1.98 |
| 1000 | 3.74 | -16.2 | 3.68 | -5.14 |
| 5 | 0.001 | 50.47 | 0.0 | 49.52 | -0.0 |
| 0.01 | 50.47 | 0.0 | 38.06 | -0.0 |
| 0.1 | 18.16 | 0.001 | 9.26 | -0.05 |
| 10 | 3.94 | -5.99 | 5.14 | -1.27 |
| 100 | 3.96 | -8.88 | 4.38 | -3.19 |
| 1000 | 3.93 | -9.7 | 3.97 | -5.97 |

The following ROC curves were obtained using SVD decomposition with min\_df=2 and 5 by varying k.

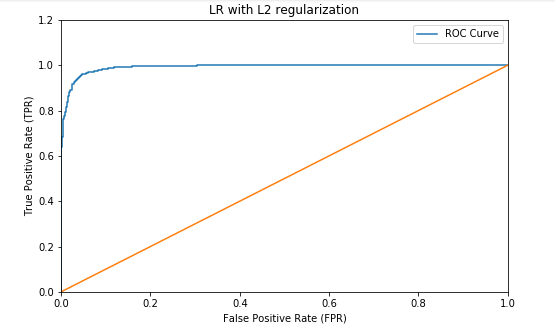
* **min\_df=2; k=0.001; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 49.5238095238 |
| Precision | 24.7619047619 |
| Recall | 50.0 |



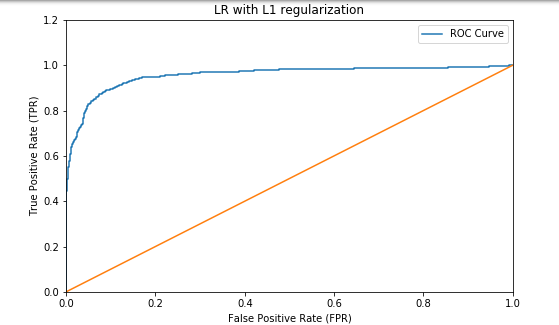
* **min\_df=2; k=0.001; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 68.6349206349 |
| Precision | 80.8378588053 |
| Recall | 68.3333333333 |



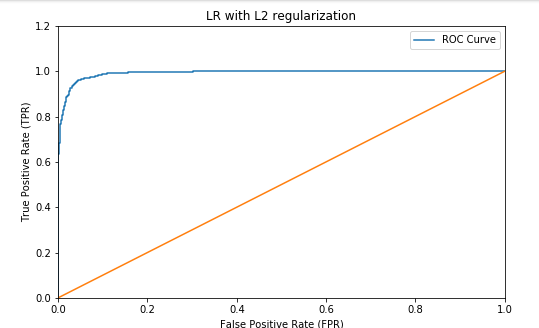
* **min\_df=2; k=0.01; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 89.4285714286 |
| Precision | 89.6047355982 |
| Recall | 89.4611756168 |



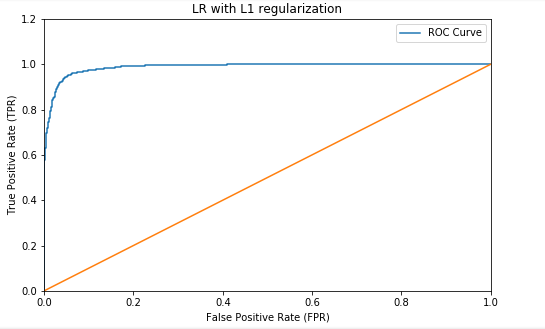
* **min\_df=2; k=0.01; l2**

|  |  |
| --- | --- |
| Statistic | Result |
| Accuracy | 94.0952380952 |
| Precision | 94.5455605495 |
| Recall | 94.0487421384 |



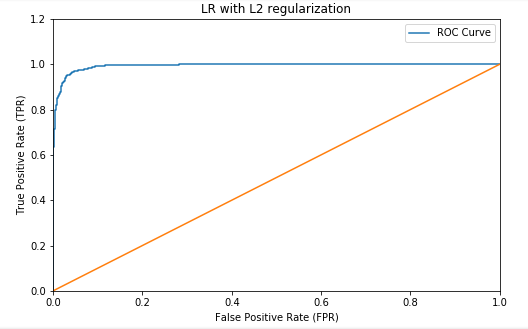
* **min\_df=2; k=0.1; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 94.7619047619 |
| Precision | 94.7724895935 |
| Recall | 94.755684567 |



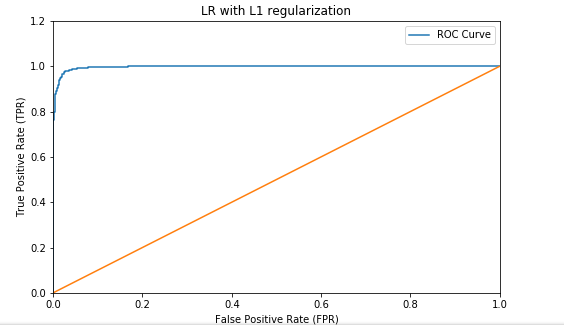
* **min\_df=2; k=0.1; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 95.873015873 |
| Precision | 95.9123748789 |
| Recall | 95.860546686 |



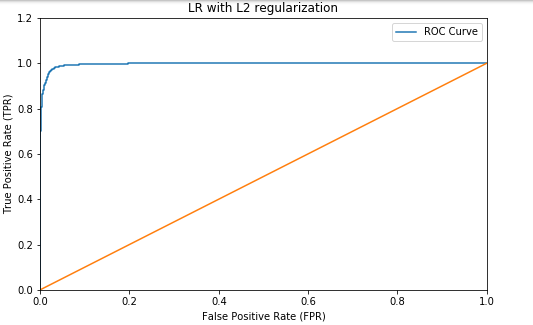
* **min\_df=2; k=10; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3333333333 |
| Precision | 97.3556303595 |
| Recall | 97.3246250605 |



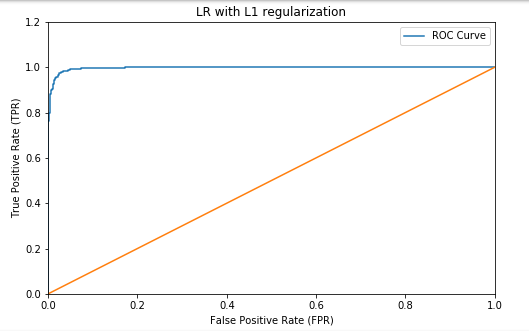
* **min\_df=2; k=10; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.2698412698 |
| Precision | 97.2977498531 |
| Recall | 97.2599177552 |



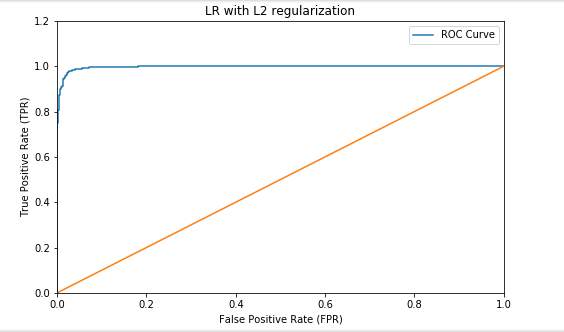
* **min\_df=2; k=100; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3968253968 |
| Precision | 97.4141060641 |
| Recall | 97.3893323657 |



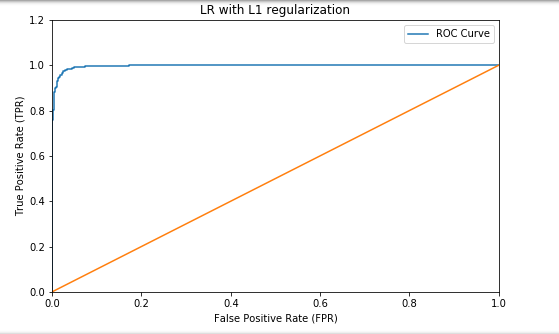
* **min\_df=2; k=100; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.5238095238 |
| Precision | 97.5411680994 |
| Recall | 97.5163280116 |



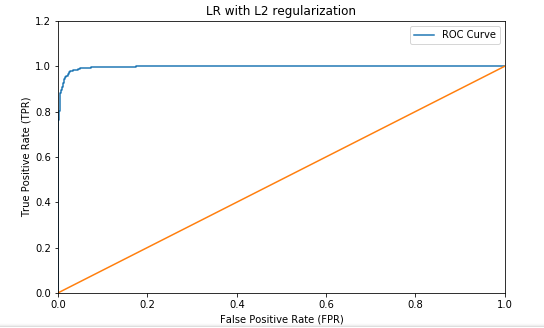
* **min\_df=2; k=1000; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3968253968 |
| Precision | 97.4141060641 |
| Recall | 97.3893323657 |



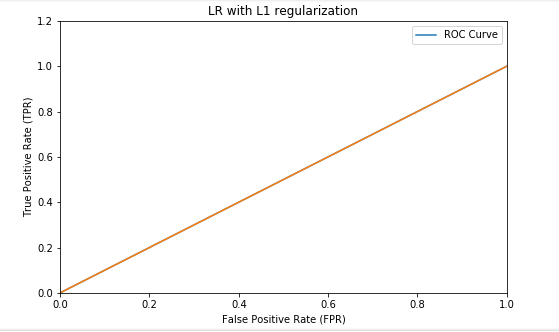
* **min\_df=2; k=1000; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3968253968 |
| Precision | 97.4141060641 |
| Recall | 97.3893323657 |



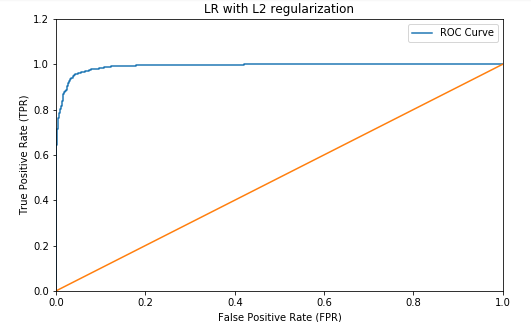
* **min\_df=5; k=0.001; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 49.5238095238 |
| Precision | 24.7619047619 |
| Recall | 50.0 |



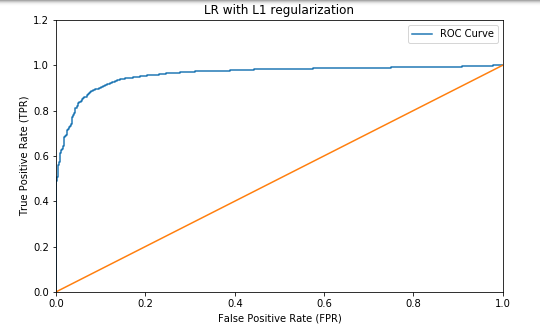
* **min\_df=5; k=0.001; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 73.1746031746 |
| Precision | 82.6488706366 |
| Recall | 72.9166666667 |



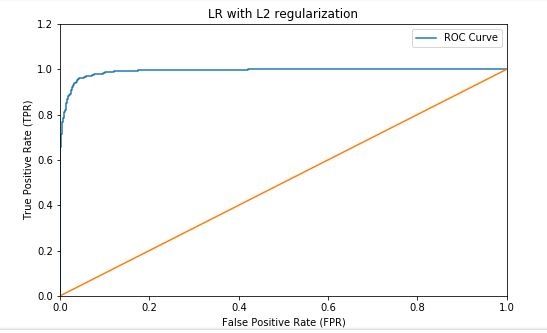
* **min\_df=5; k=0.01; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 90.2222222222 |
| Precision | 90.2922285515 |
| Recall | 90.2431059507 |



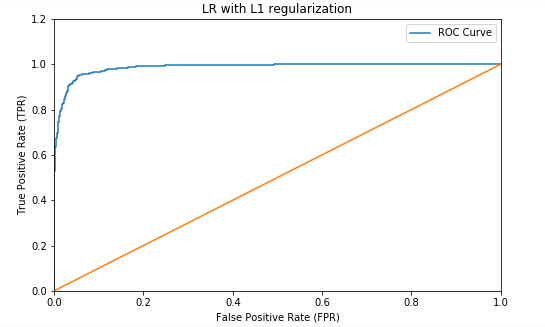
* **min\_df=5; k=0.01; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 94.253968254 |
| Precision | 94.6132723112 |
| Recall | 94.2126269956 |



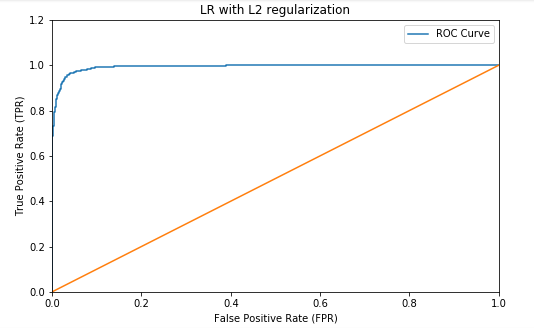
* **min\_df=5; k=0.1; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 94.7301587302 |
| Precision | 94.7416713721 |
| Recall | 94.723633285 |



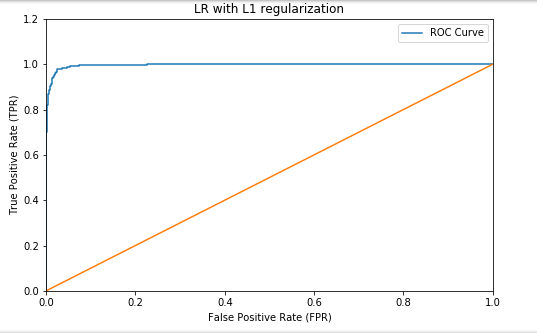
* **min\_df=5; k=0.1; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 95.873015873 |
| Precision | 95.9196154927 |
| Recall | 95.8593372037 |



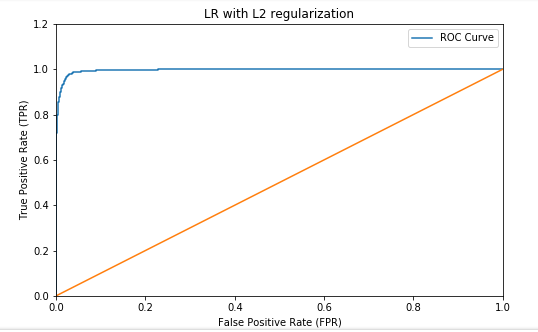
* **min\_df=5; k=10; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3968253968 |
| Precision | 97.4165612454 |
| Recall | 97.3887276246 |



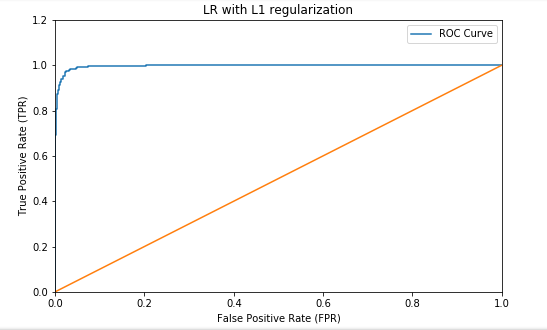
* **min\_df=5; k=10; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3968253968 |
| Precision | 97.4279161206 |
| Recall | 97.3863086599 |



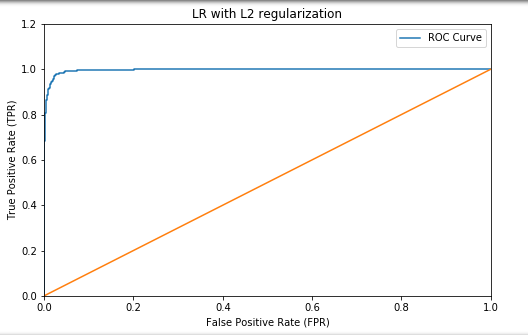
* **min\_df=5; k=100; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3333333333 |
| Precision | 97.3461321287 |
| Recall | 97.3270440252 |



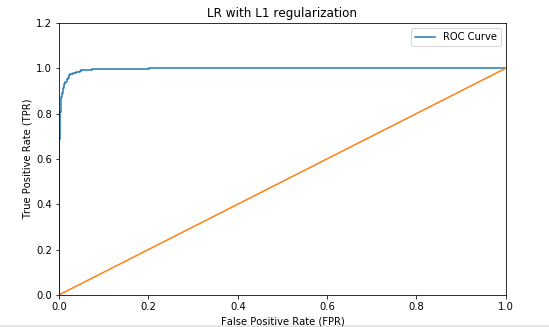
* **min\_df=5; k=100; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.4285714286 |
| Precision | 97.4496149643 |
| Recall | 97.4201741655 |



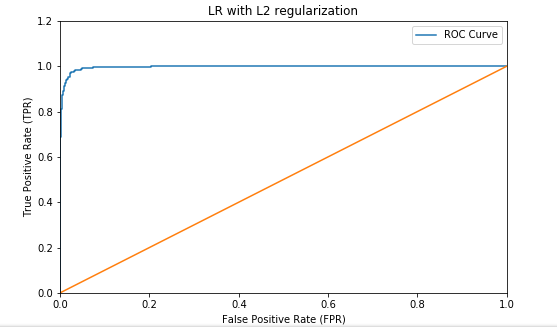
* **min\_df=5; k=1000; l1**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3333333333 |
| Precision | 97.3461321287 |
| Recall | 97.3270440252 |



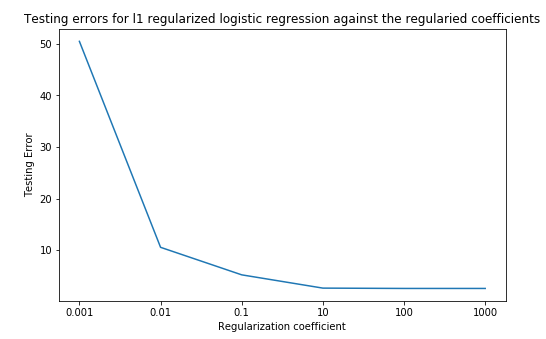
* **min\_df=5; k=1000; l2**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 97.3015873016 |
| Precision | 97.3154228422 |
| Recall | 97.2949927431 |

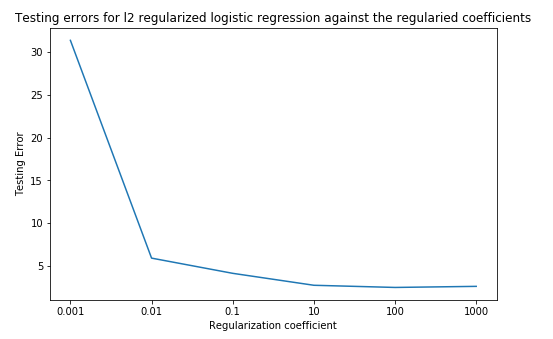


**Testing Errors:**

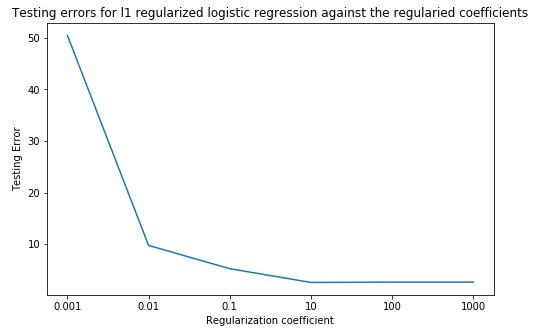
* **min\_df=2; l1 regularization**



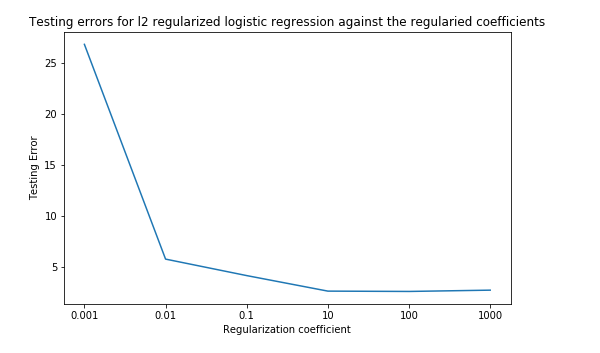
* **min\_df=2; l2 regularization**



* **min\_df=5; l1 regularization**



* **min\_df=5; l2 regularization**



**Observations:**

We observe that for smaller values of regularization parameter, the testing error is high due to excessive regularization. However, the error decreases steadily on increasing regularization up to a certain point. As we increase the regularization parameter, the fitted hyperplane moves away from the origin.

The L1-norm should be used when we need a robust solution and can tolerate multiple stable solutions. It is better to go with L2 norm when we need to obtain a single stable solution. Moreover, L1 norm can be used when we have good computational power. But if not, L2 norm is a better option.

# 6. Multiclass Classification

## Question (i2) Naive Bayes and SVM

Here, we perform classification on multiple classes and train classifiers on the documents belonging to the following classes:

A: comp.sys.ibm.pc.hardware

B: comp.sys.mac.hardware

C: misc.forsale

D: soc.religion.christian

We use two classification techniques namely One vs One and One vs Rest. The first technique trains nC2 different classifiers and each classifier trains individual classes against each other. On the contrary, the second technique trains n classifiers and trains each one against the rest. We obtain the following results for Naive Bayes and multiclass SVM classification when performed with both One vs One and One vs Rest methods.

* **min\_df=2; OneVsOne- SVM; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 88.7539936102 |
| Precision | 88.8377177276 |
| Recall | 88.7004676945 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 334 | 40 | 18 | 0 |
| **Actual: B** | 43 | 318 | 23 | 1 |
| **Actual: C** | 22 | 14 | 354 | 0 |
| **Actual: D** | 9 | 2 | 4 | 383 |

* **min\_df=2; OneVsRest- SVM; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 89.0734824281 |
| Precision | 88.985634018 |
| Recall | 89.0188288919 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 319 | 44 | 25 | 4 |
| **Actual: B** | 32 | 325 | 27 | 1 |
| **Actual: C** | 18 | 13 | 357 | 2 |
| **Actual: D** | 3 | 0 | 2 | 393 |

* **min\_df=2; OneVsOne- SVM; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 76.7412140575 |
| Precision | 80.7026663405 |
| Recall | 76.5714274957 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 352 | 33 | 7 | 0 |
| **Actual: B** | 166 | 210 | 9 | 0 |
| **Actual: C** | 78 | 42 | 270 | 0 |
| **Actual: D** | 20 | 7 | 2 | 369 |

* **min\_df=2; OneVsRest- SVM; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 83.2587859425 |
| Precision | 83.0502894791 |
| Recall | 83.1386670879 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 290 | 46 | 39 | 17 |
| **Actual: B** | 58 | 262 | 43 | 22 |
| **Actual: C** | 18 | 8 | 355 | 9 |
| **Actual: D** | 0 | 0 | 2 | 396 |

* **min\_df=2; OneVsOne- NB; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 73.4185303514 |
| Precision | 77.2340704709 |
| Recall | 73.2025930518 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 280 | 10 | 100 | 2 |
| **Actual: B** | 97 | 153 | 131 | 4 |
| **Actual: C** | 35 | 16 | 338 | 1 |
| **Actual: D** | 1 | 0 | 19 | 378 |

* **min\_df=2; OneVsRest- NB; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 73.482428115 |
| Precision | 77.3389900505 |
| Recall | 73.2669134527 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 274 | 10 | 107 | 1 |
| **Actual: B** | 85 | 154 | 139 | 7 |
| **Actual: C** | 29 | 18 | 342 | 1 |
| **Actual: D** | 0 | 0 | 18 | 380 |

* **min\_df=2; OneVsOne- NB; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 78.4664536741 |
| Precision | 78.4683685221 |
| Recall | 78.316198488 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 285 | 48 | 53 | 6 |
| **Actual: B** | 74 | 233 | 74 | 4 |
| **Actual: C** | 38 | 26 | 320 | 6 |
| **Actual: D** | 5 | 1 | 2 | 390 |

* **min\_df=2; OneVsRest- NB; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 80.1277955272 |
| Precision | 80.1200312273 |
| Recall | 79.990297899 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 281 | 52 | 52 | 7 |
| **Actual: B** | 66 | 245 | 70 | 4 |
| **Actual: C** | 35 | 16 | 335 | 4 |
| **Actual: D** | 3 | 1 | 1 | 393 |

* **min\_df=5; OneVsOne- SVM; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 88.8817891374 |
| Precision | 89.0114995819 |
| Recall | 88.8191385069 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 340 | 37 | 15 | 0 |
| **Actual: B** | 51 | 312 | 22 | 0 |
| **Actual: C** | 20 | 16 | 354 | 0 |
| **Actual: D** | 9 | 1 | 3 | 385 |

* **min\_df=5; OneVsRest- SVM; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 89.2651757188 |
| Precision | 89.1745679853 |
| Recall | 89.2062404872 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 321 | 42 | 25 | 4 |
| **Actual: B** | 32 | 323 | 29 | 1 |
| **Actual: C** | 15 | 15 | 358 | 2 |
| **Actual: D** | 2 | 0 | 1 | 395 |

* **min\_df=5; OneVsOne- SVM; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 84.6645367412 |
| Precision | 87.7697522033 |
| Recall | 84.5526627297 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 375 | 9 | 8 | 0 |
| **Actual: B** | 117 | 255 | 13 | 0 |
| **Actual: C** | 49 | 13 | 328 | 0 |
| **Actual: D** | 28 | 1 | 2 | 367 |

* **min\_df=5; OneVsRest- SVM; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 88.3706070288 |
| Precision | 88.2696100182 |
| Recall | 88.2917480342 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 322 | 30 | 30 | 10 |
| **Actual: B** | 39 | 306 | 31 | 9 |
| **Actual: C** | 17 | 13 | 358 | 2 |
| **Actual: D** | 0 | 0 | 1 | 397 |

* **min\_df=5; OneVsOne- NB; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 77.6357827476 |
| Precision | 78.8057048768 |
| Recall | 77.4429645791 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 295 | 18 | 66 | 13 |
| **Actual: B** | 74 | 189 | 109 | 13 |
| **Actual: C** | 23 | 24 | 341 | 2 |
| **Actual: D** | 0 | 1 | 7 | 390 |

* **min\_df=5; OneVsRest- NB; SVD**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 76.357827476 |
| Precision | 77.6406457814 |
| Recall | 76.1585741669 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 279 | 20 | 79 | 14 |
| **Actual: B** | 72 | 183 | 115 | 15 |
| **Actual: C** | 17 | 26 | 341 | 6 |
| **Actual: D** | 0 | 0 | 6 | 392 |

* **min\_df=5; OneVsOne- NB; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 79.8722044728 |
| Precision | 79.9884809317 |
| Recall | 79.7379709606 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 303 | 34 | 50 | 5 |
| **Actual: B** | 55 | 257 | 68 | 5 |
| **Actual: C** | 56 | 28 | 298 | 8 |
| **Actual: D** | 4 | 0 | 2 | 392 |

* **min\_df=5; OneVsRest- NB; NMF**

|  |  |
| --- | --- |
| **Statistic** | **Result** |
| Accuracy | 82.1086261981 |
| Precision | 82.183039153 |
| Recall | 81.9917982831 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted: A** | **Predicted: B** | **Predicted: C** | **Predicted: D** |
| **Actual: A** | 302 | 33 | 52 | 5 |
| **Actual: B** | 53 | 272 | 56 | 4 |
| **Actual: C** | 44 | 23 | 317 | 6 |
| **Actual: D** | 2 | 1 | 1 | 394 |