EE-219 Project 1

Shrey Agarwal (004943082)

Varun Saboo (505028591)

January 29, 2018

Project 1: Classification Analysis on Textual Data

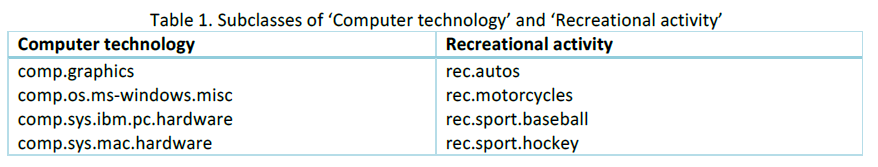
**Objective –** To analyze the performance of various classifiers for the task of document classification.

**Data –** The dataset used is the ‘20 Newsgroup’ from scikit-learn which contains 20 different categories of documents.

* Training data: 80% of the data is used for training.
* Testing data: 20% of the data used for testing.

Note that a subset of the dataset is used based on the requirements for each task.

For tasks (e) to (i), the following categories are considered.



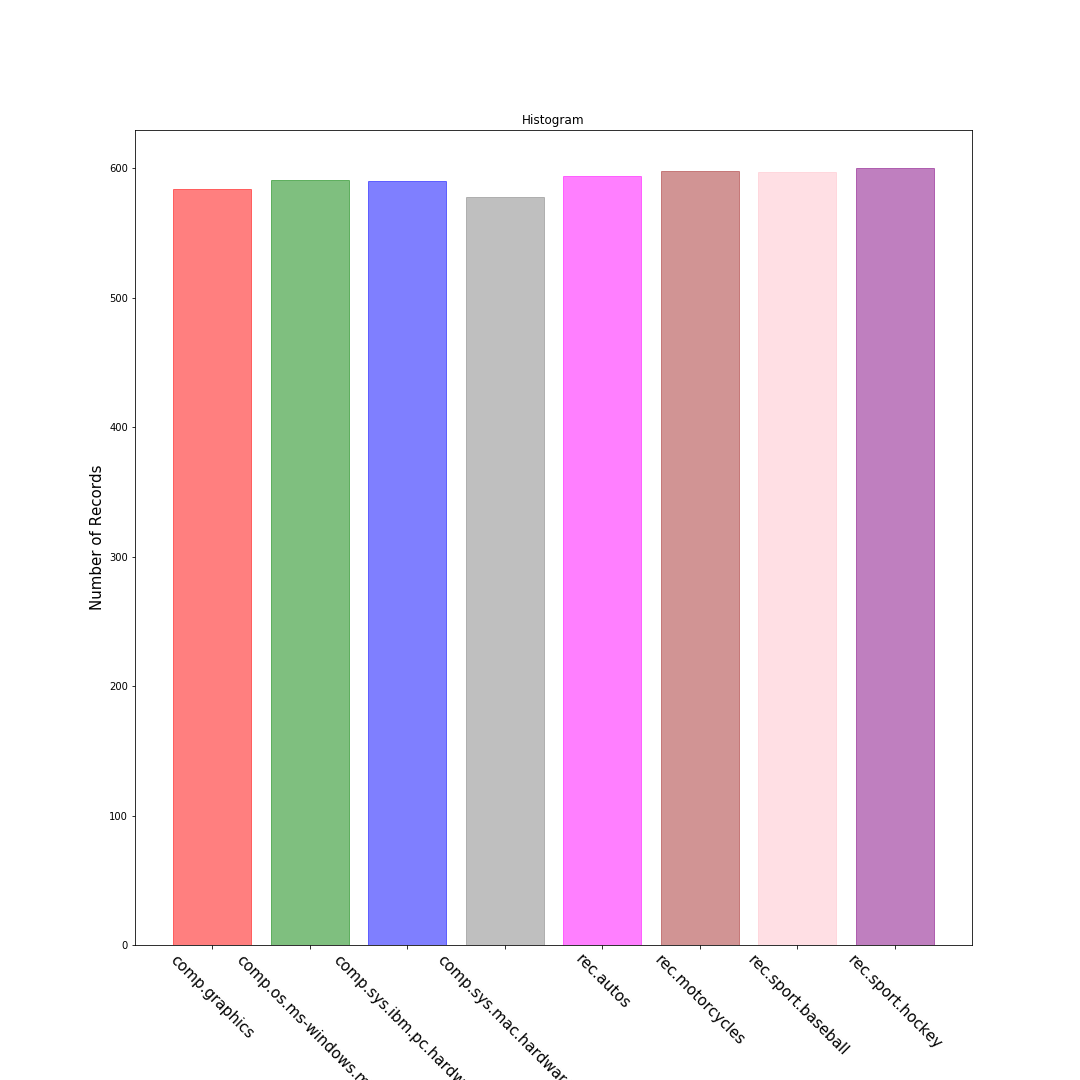
For task (j) which deals with multiclass classification, the following categories are considered:

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

# Task (a) – Visualization

We observe the distribution of data for the following 8 classes –

['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'rec.autos','rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']



As we can observe the distribution of data is uniform. All classes have approximately 600 samples in the training set.

# Task (b) – Preprocessing & TF-IDF Vector Representation

In the preprocessing stage, we did the following operations on the data.

* Removal of quotes, headers and footers while reading the dataset.
* Word Tokenizing
* Stopword Removal
* Stemming using Snowball Stemmer

After this, we created TF-IDF vector representations for each word. To create this representation, we used a parameter called min\_df which is the minimum count of each word in the document. This value was varied between 2 and 5 for all the subsequent tasks.

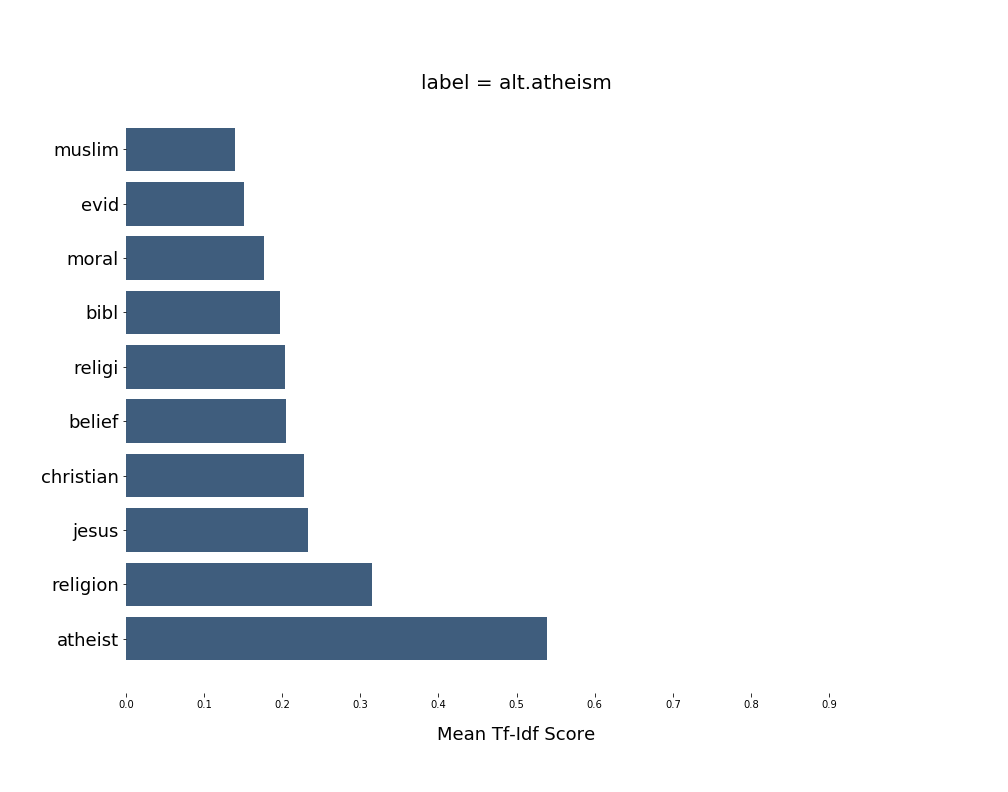
For min\_df = 5, the number of terms extracted = 7058.

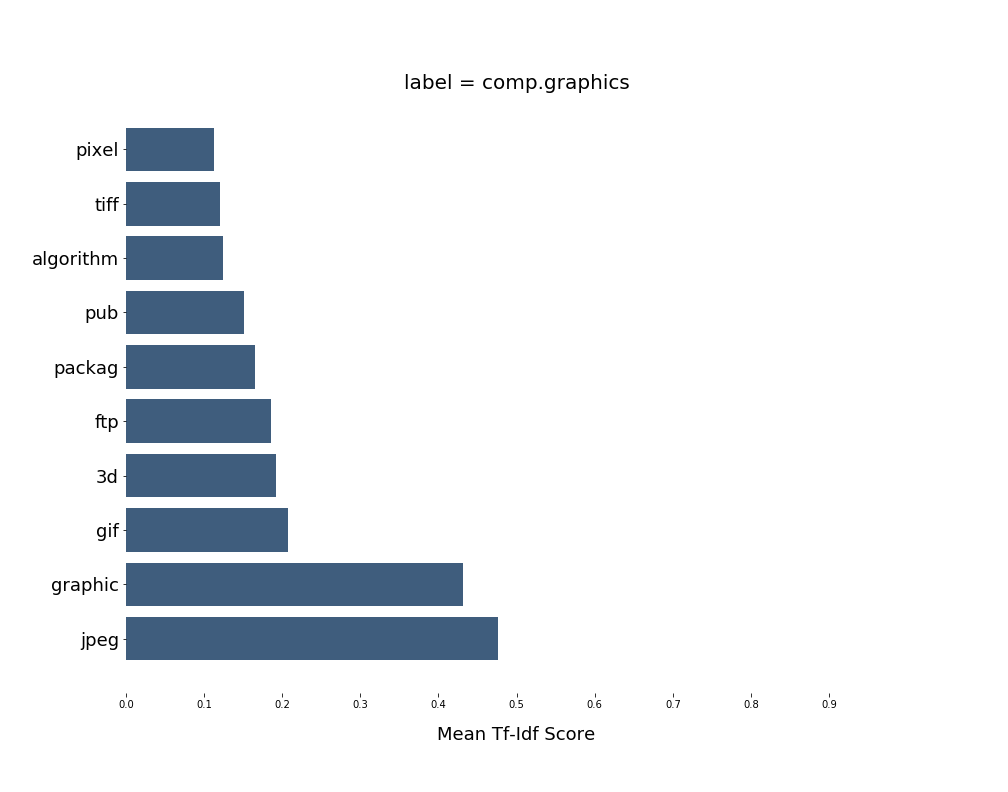
For min\_df = 2, the number of terms extracted = 17020.

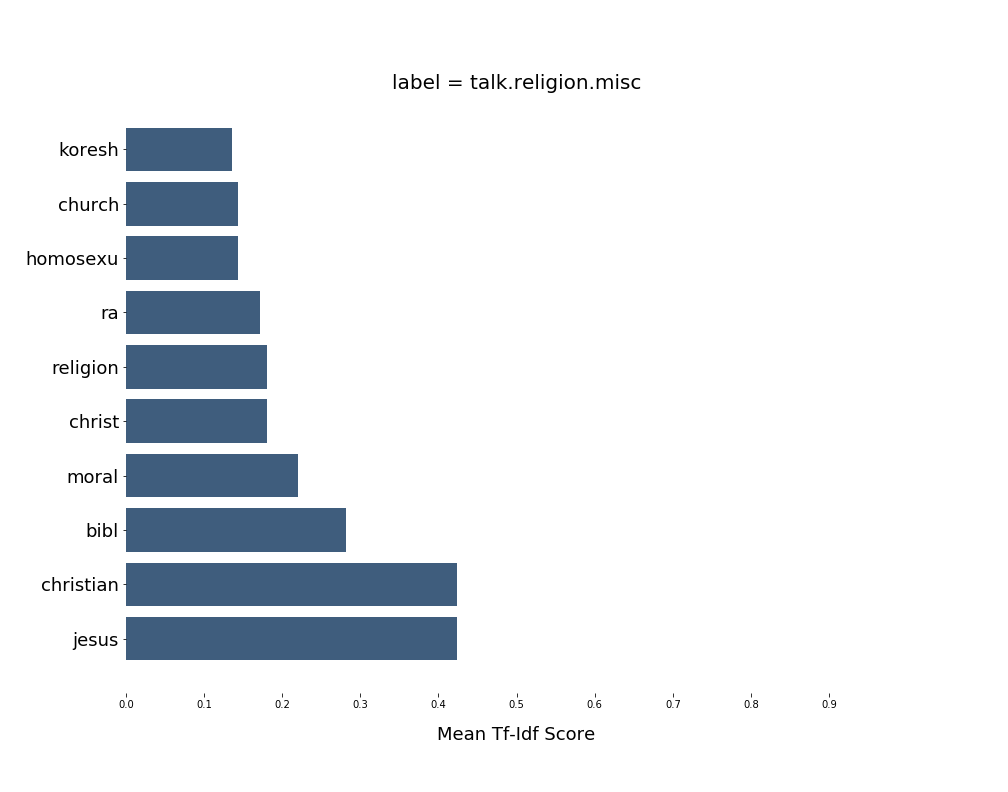
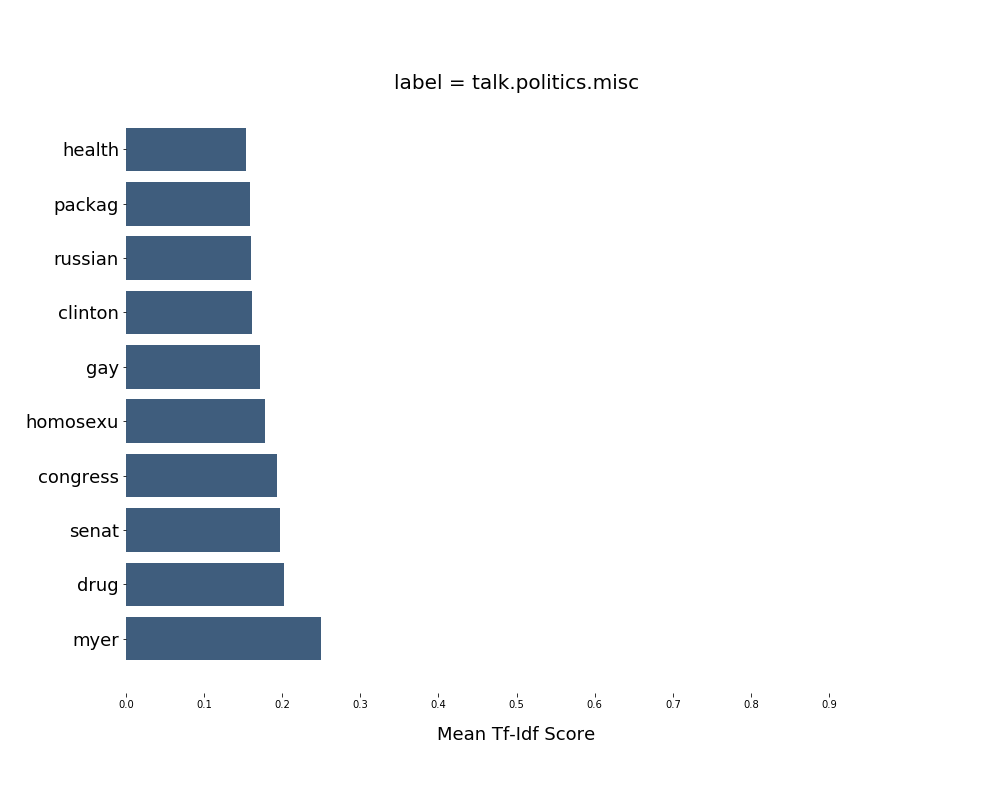
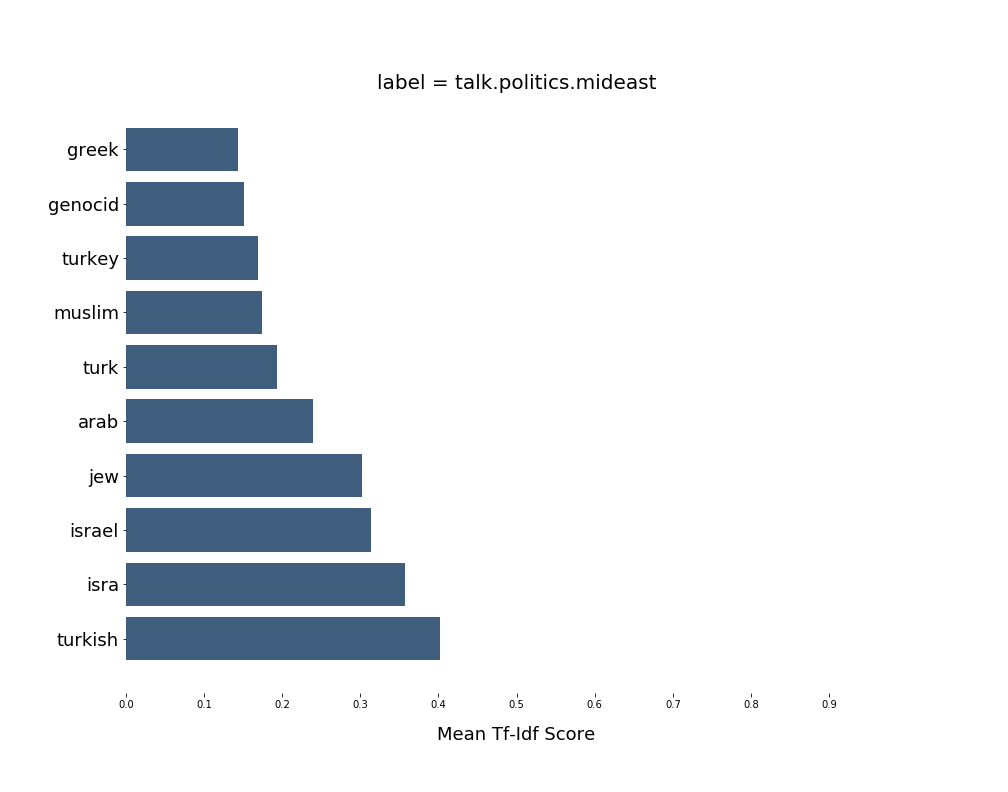
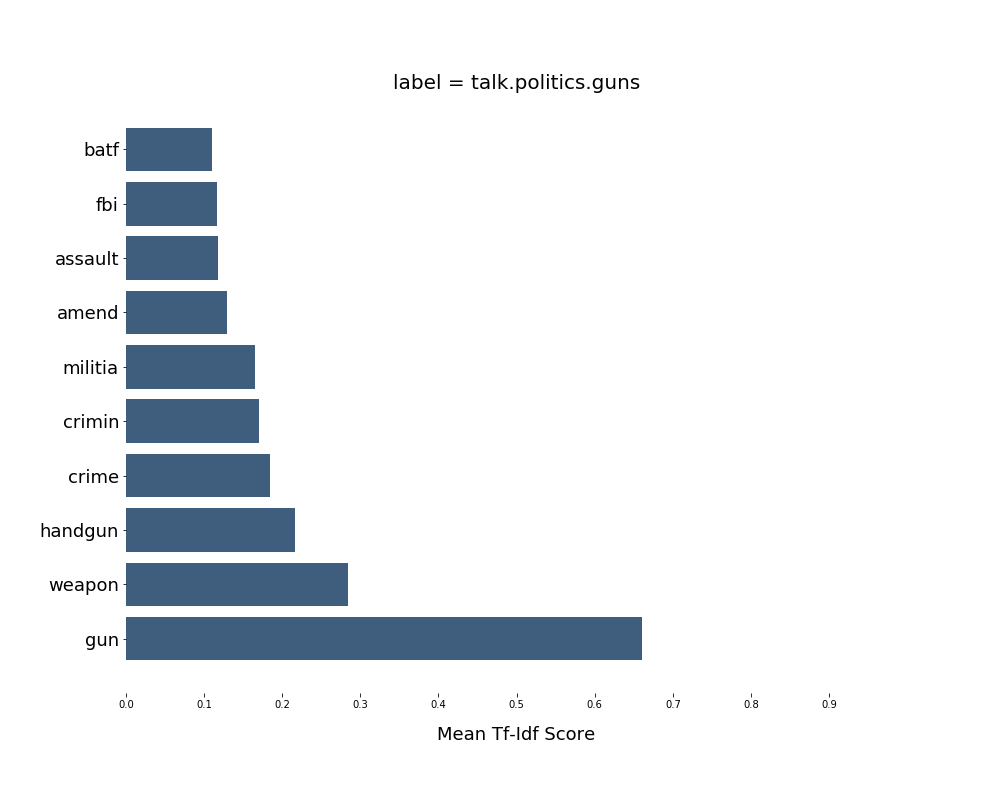
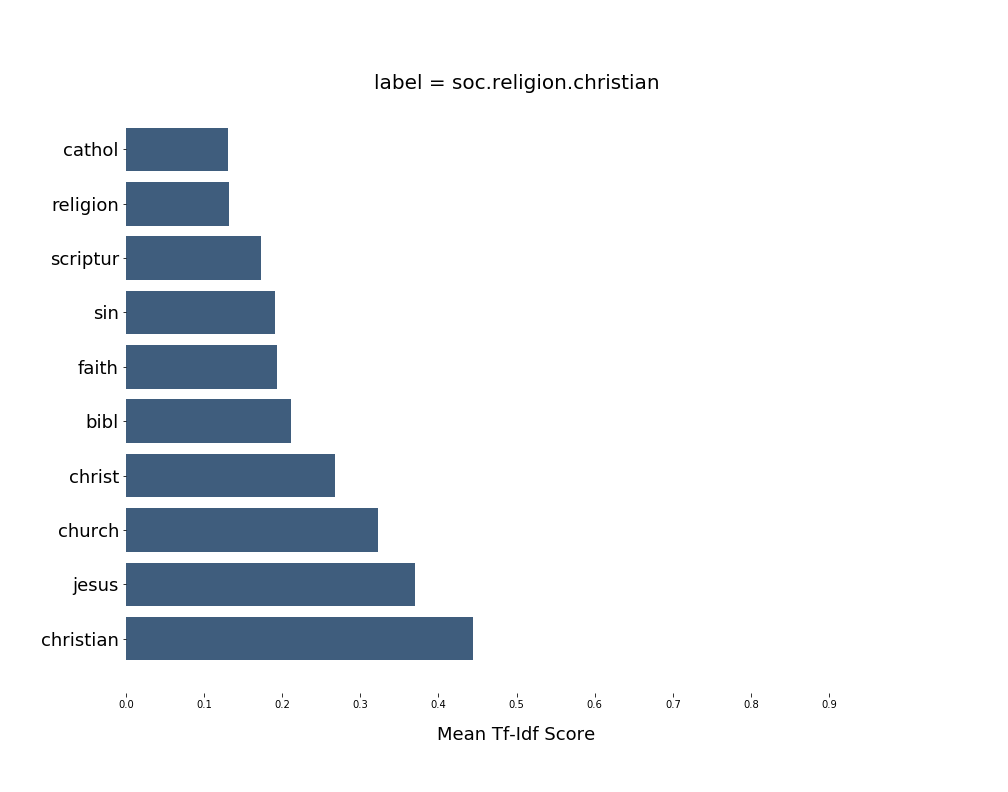
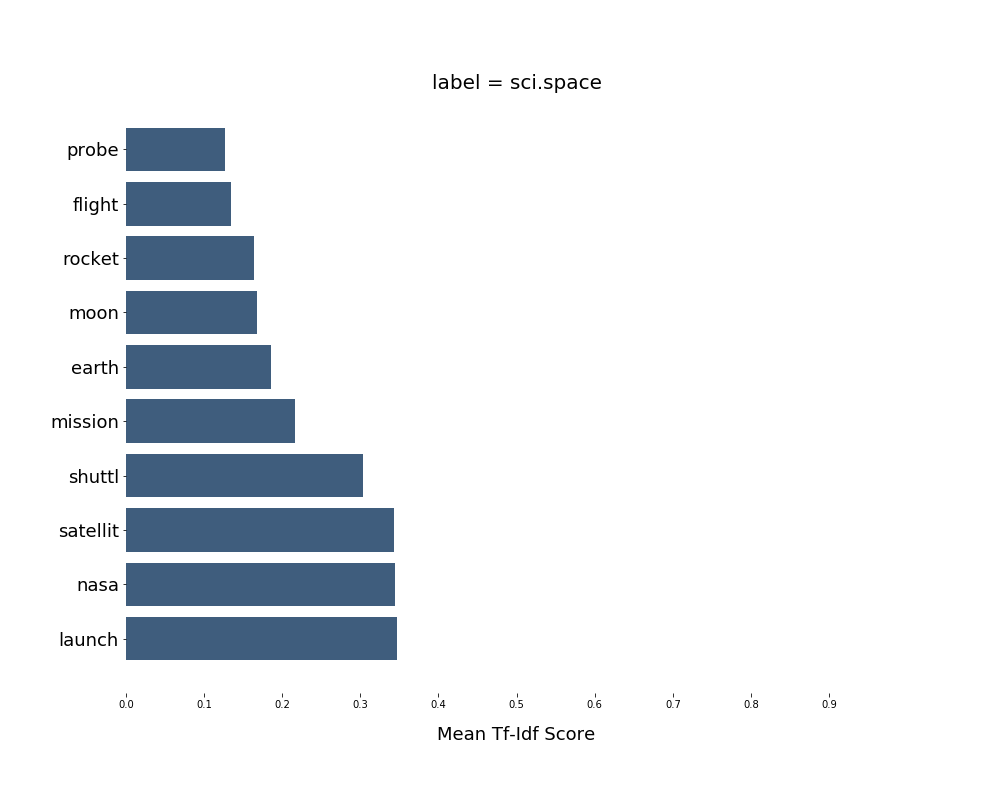
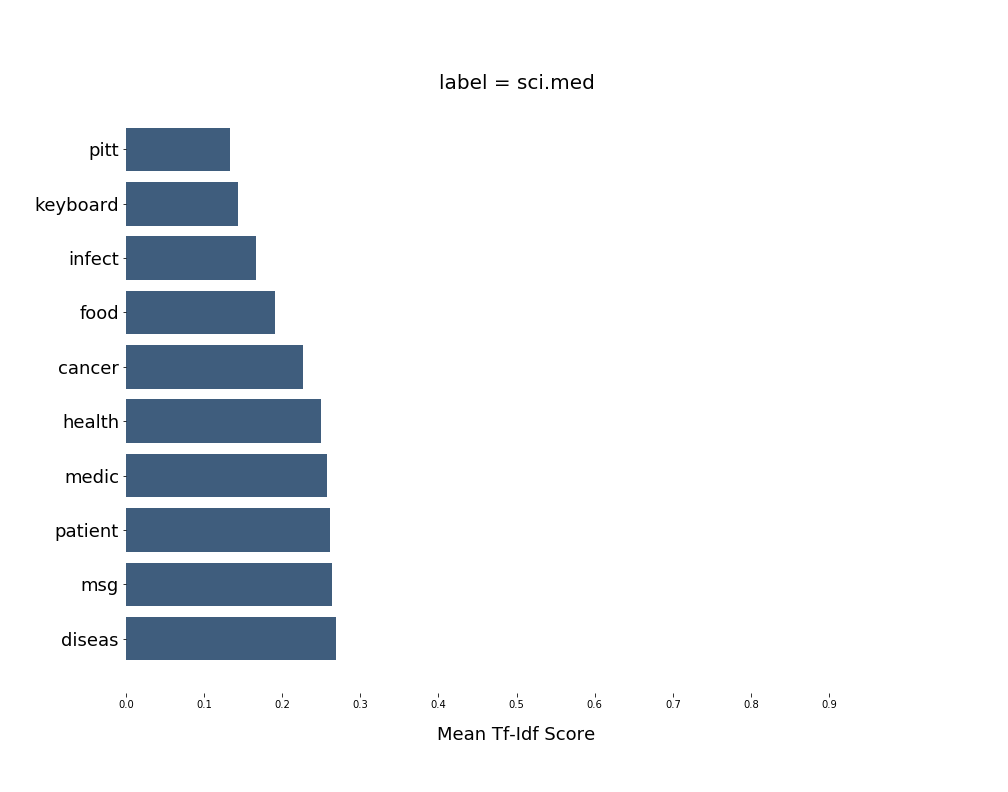
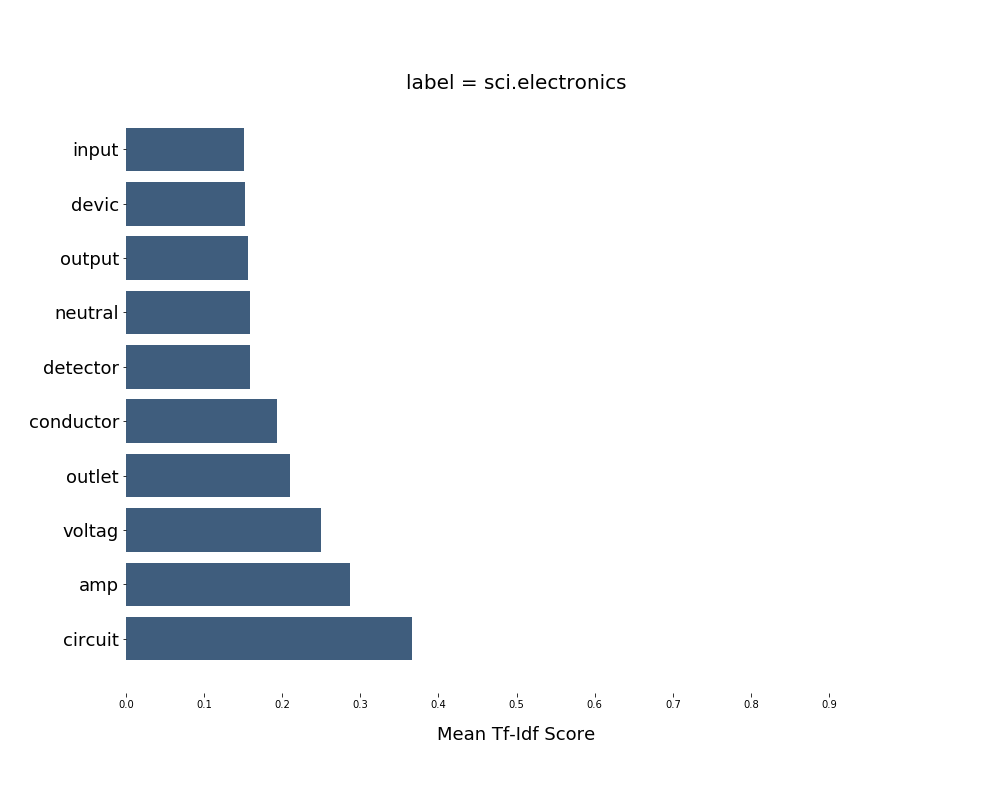
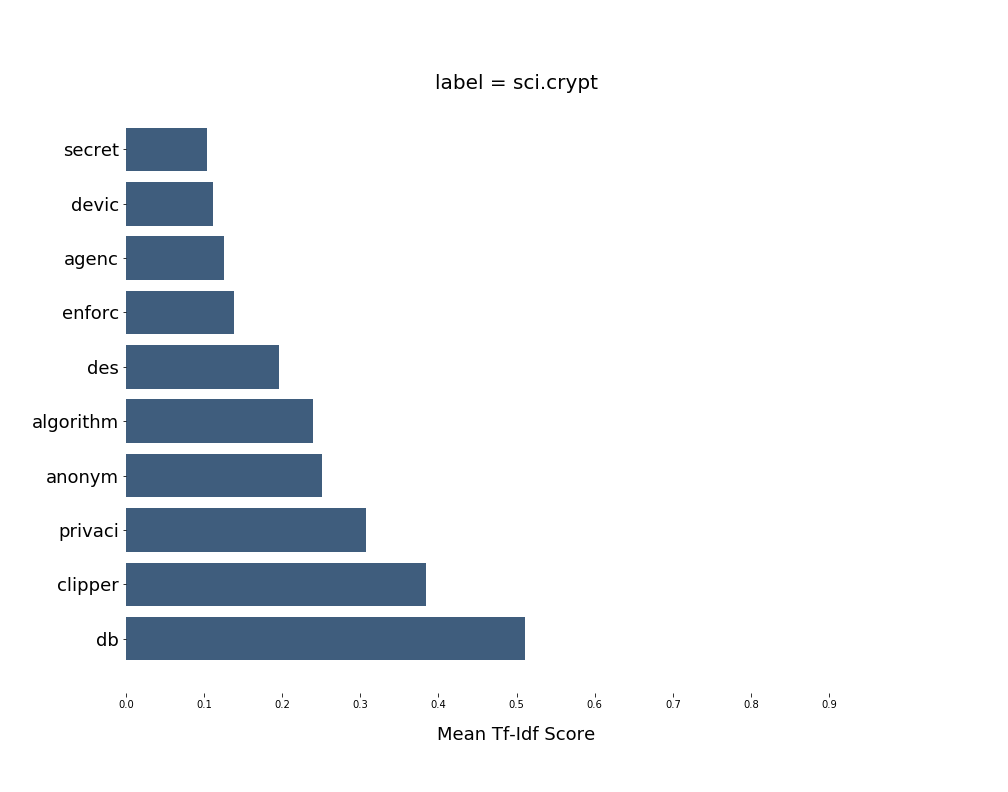
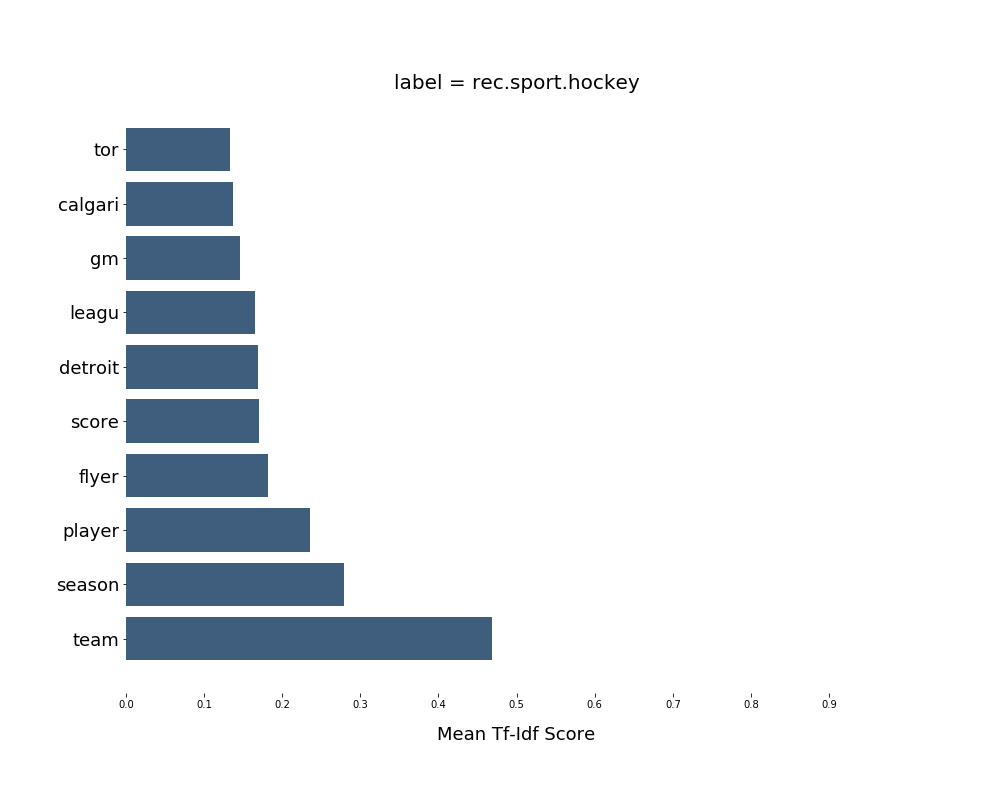
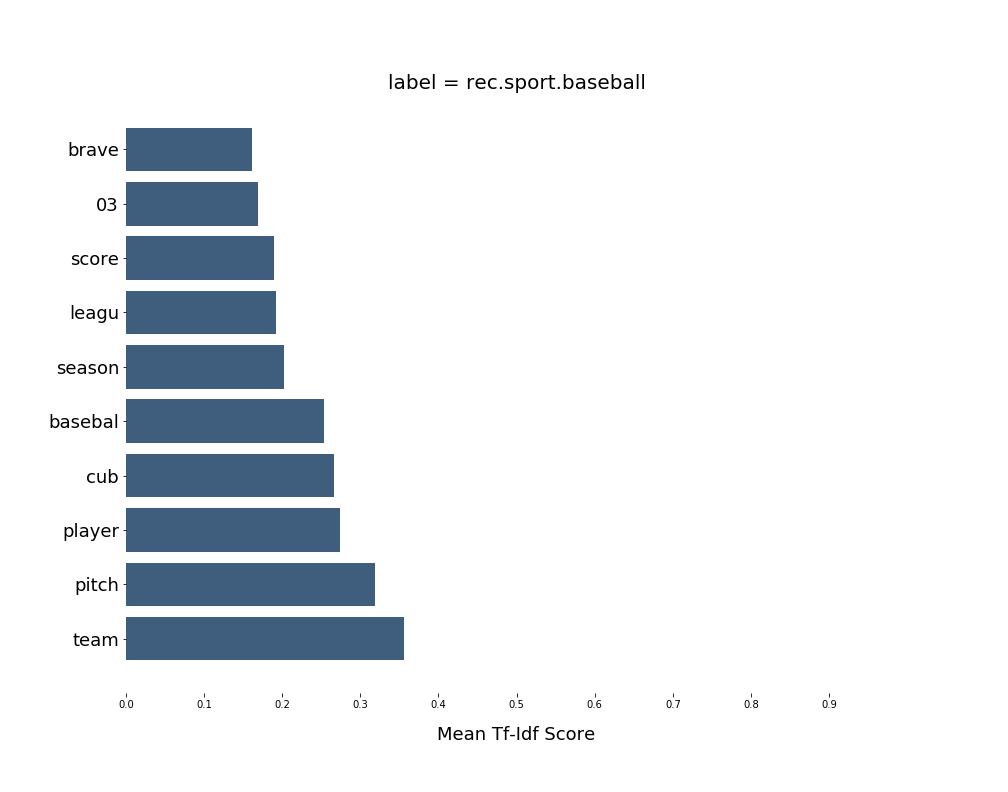
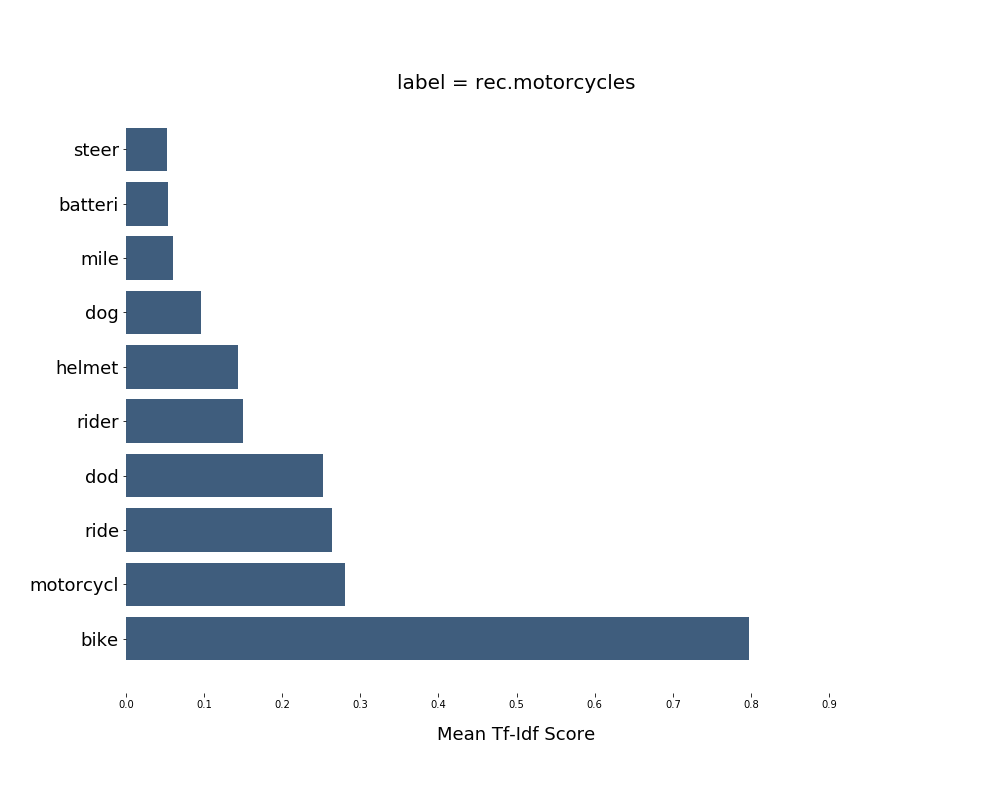
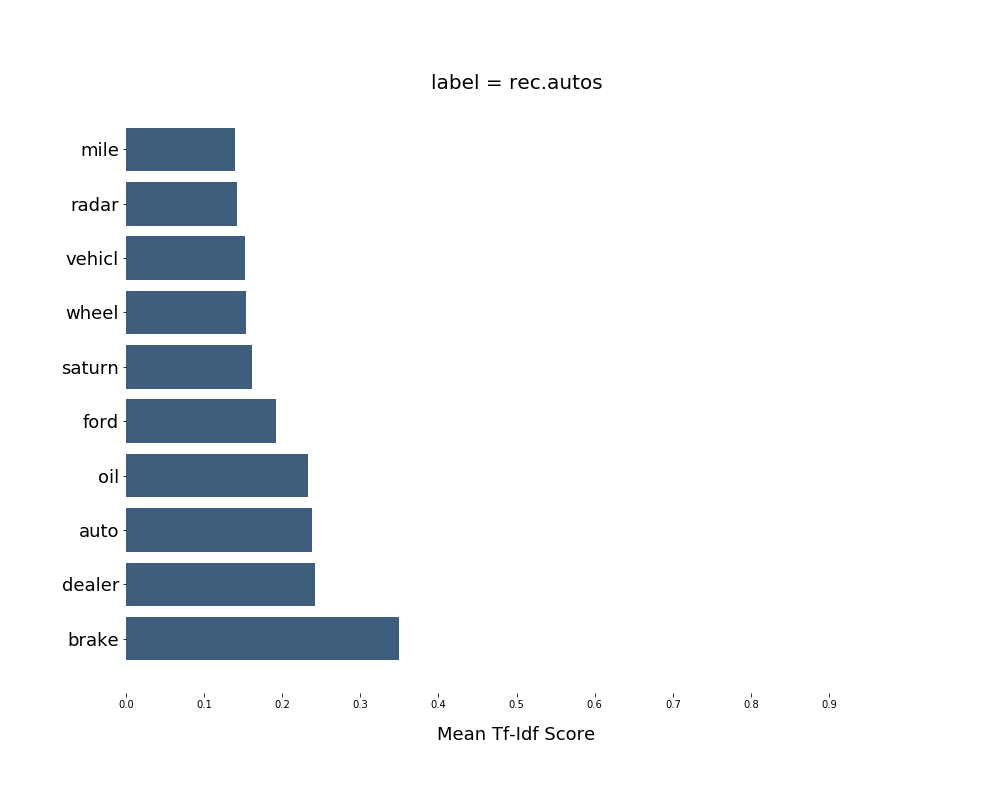
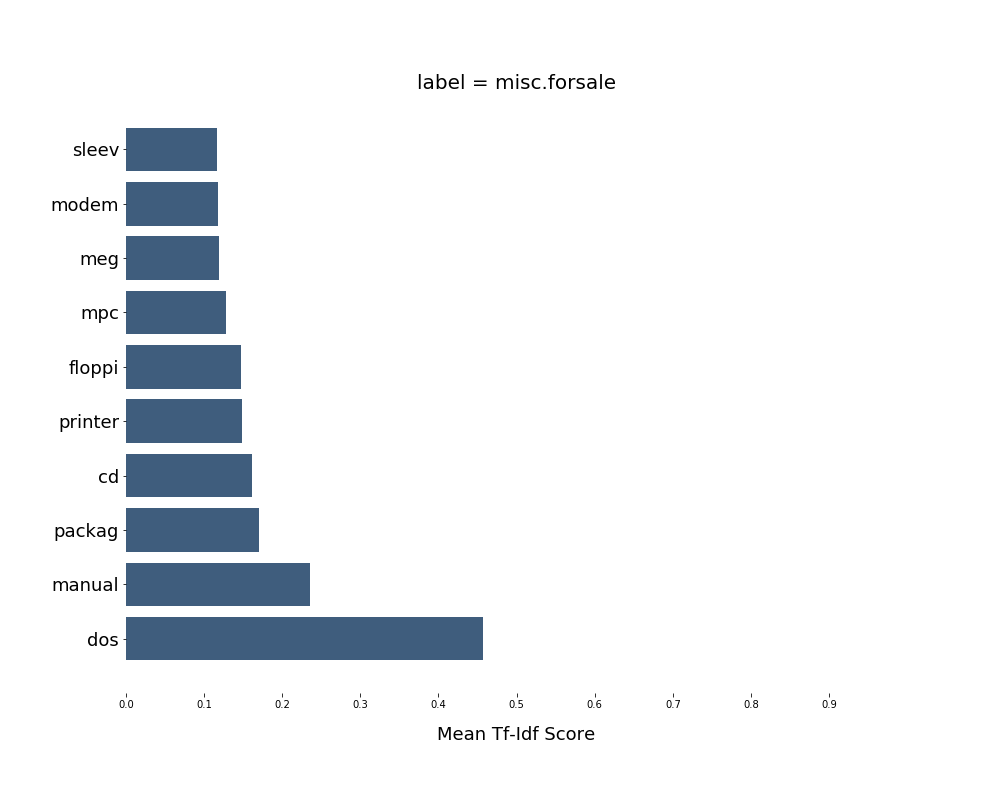
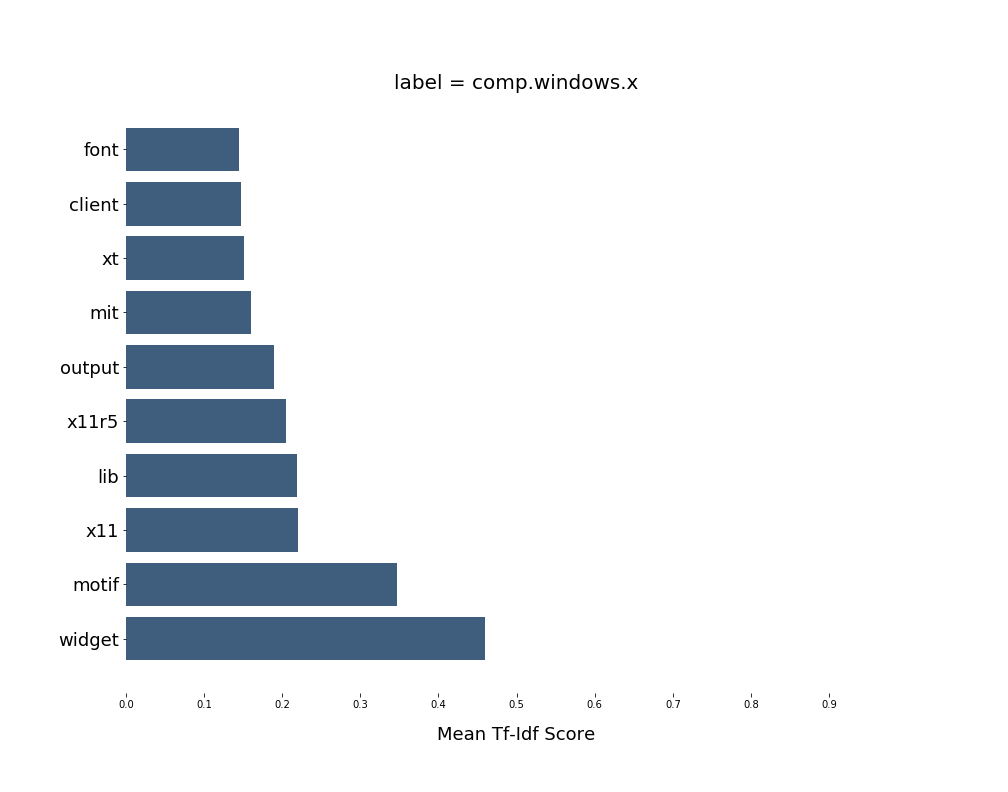
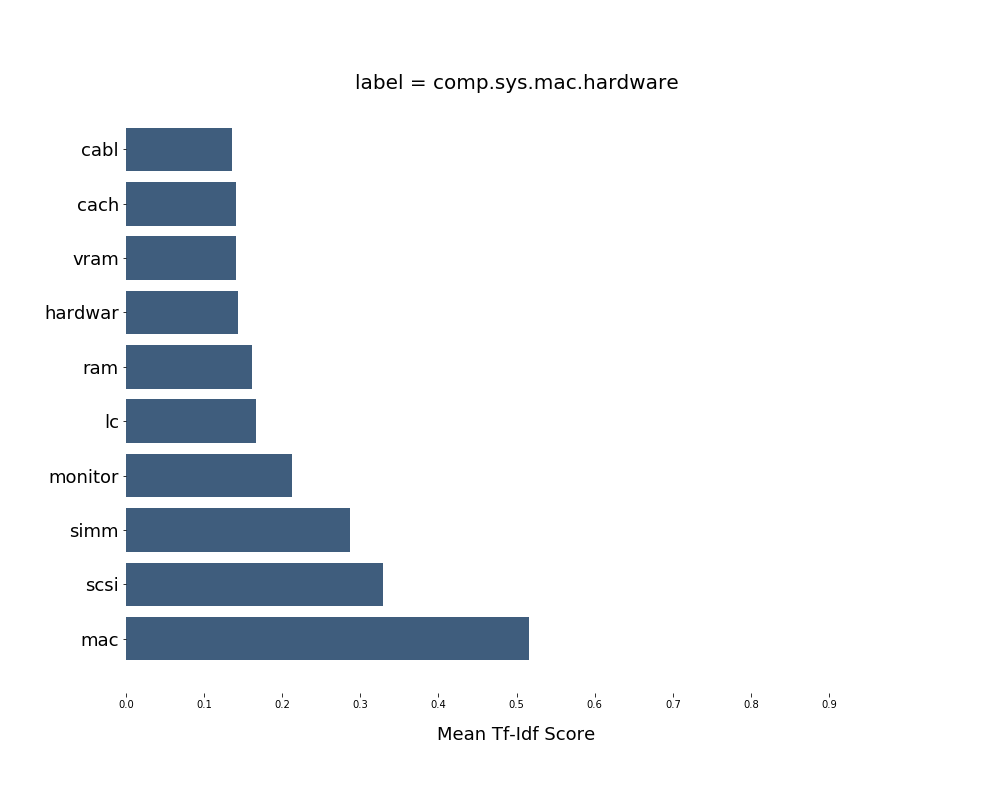
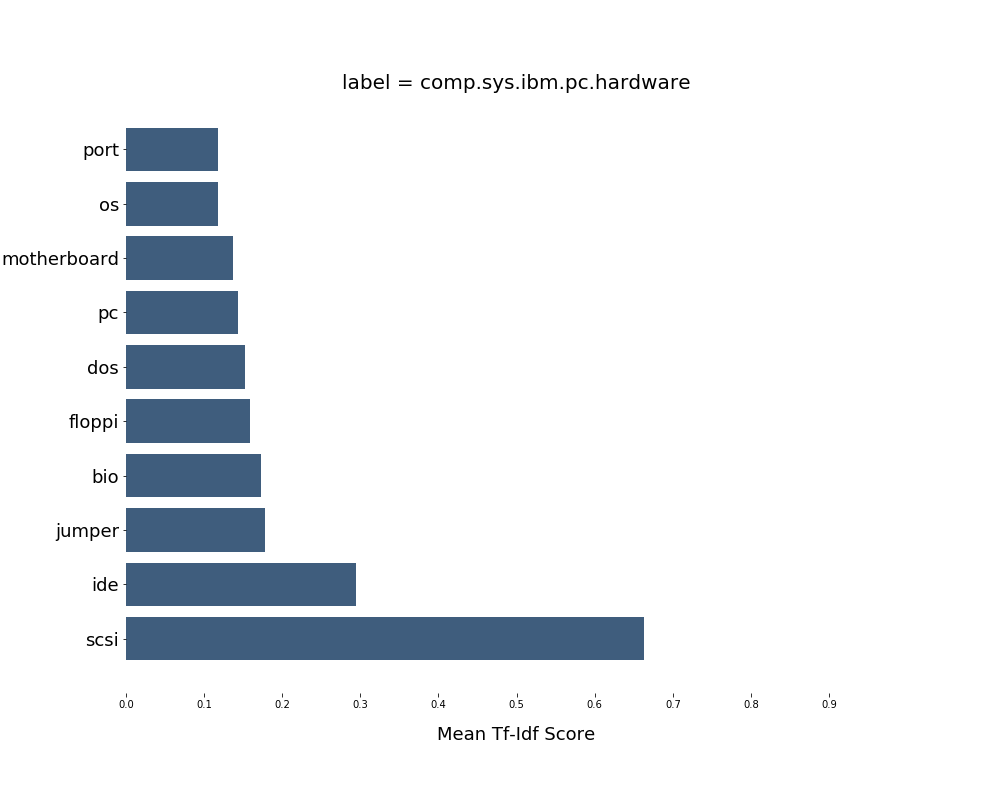
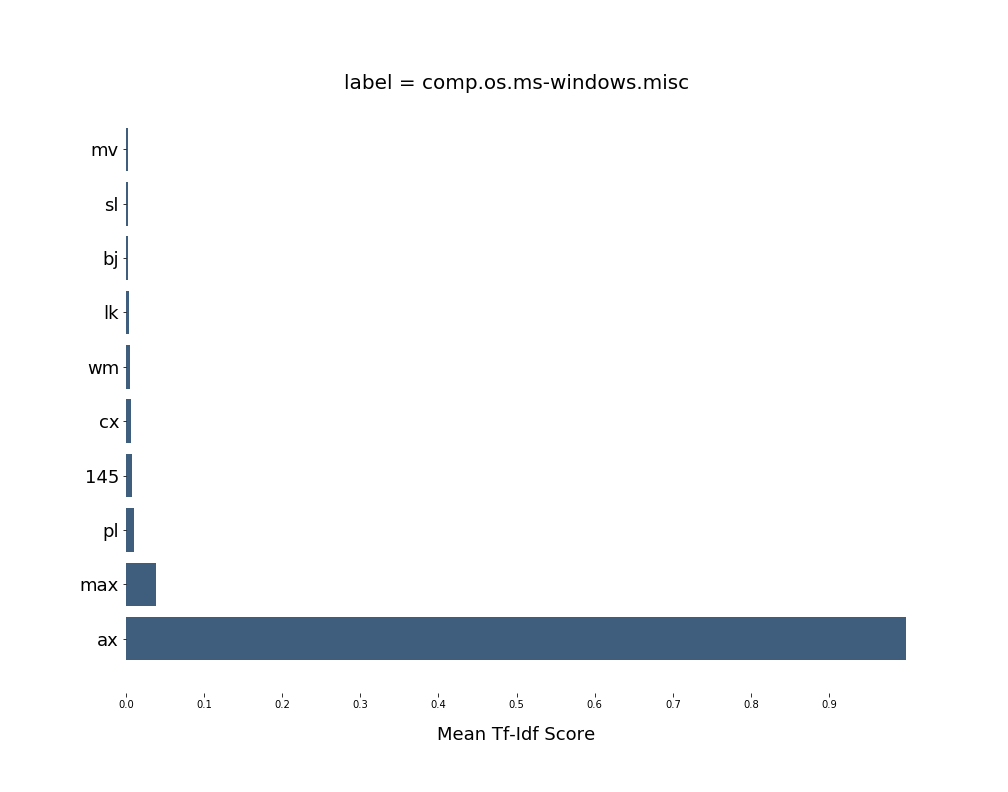
As expected, we got large number of terms with lower min\_df value.

# Task (c) – Significant Terms (TF-ICF)

The following images represent the 10 most significant words in for each class. The graph also shows the relative importance between them.







# Task (e) – (j)



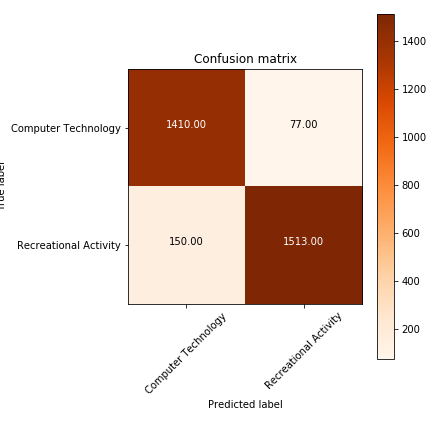
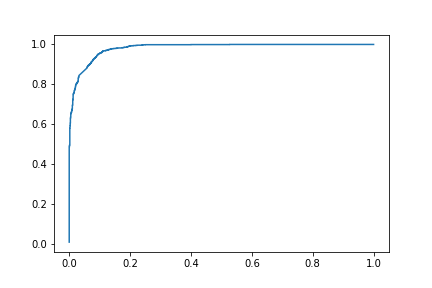


# Task (e) – Binary Classification using SVM

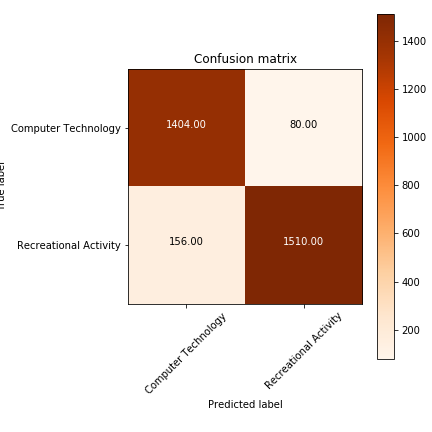
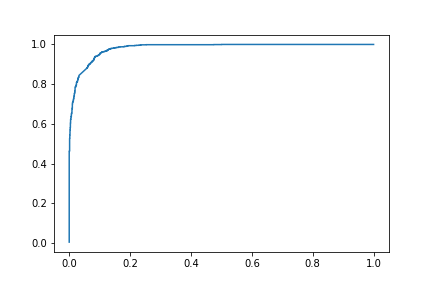
Hard SVM classified the data into two classes with good accuracy.

## Hard SVM

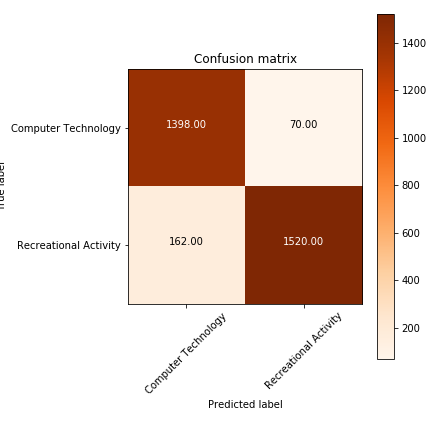
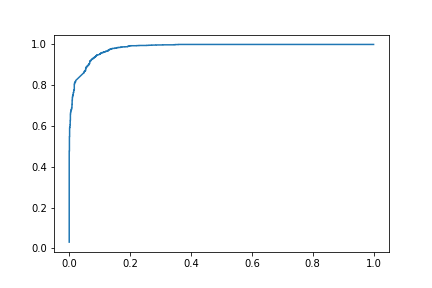
* + LSI (min\_df = 5)



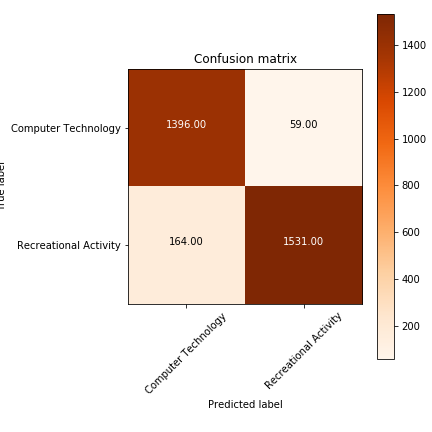
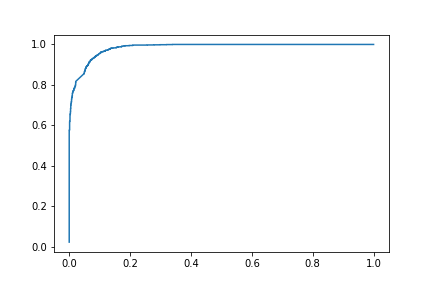
* + LSI (min\_df = 2)



* + NMF (min\_df = 5)



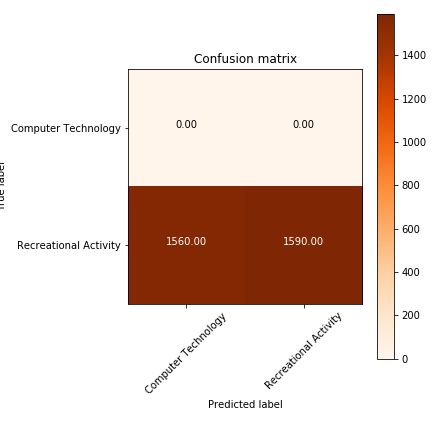
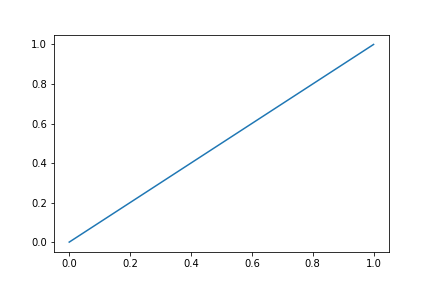
* + NMF (min\_df = 2)



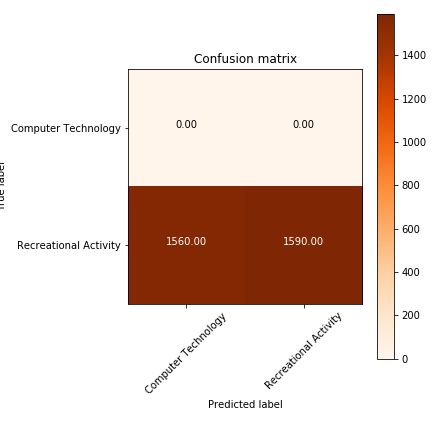
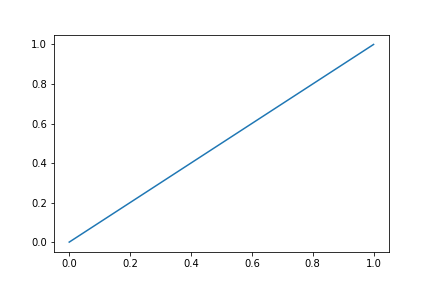
## SOFT SVM

Soft SVM could not classify the data and always gave the same output for all samples. As we can observe, it predicts everything as Recreational Activity. Since the gamma value is so low, the penalization is bare minimum and hence everything gets misclassified to one class.

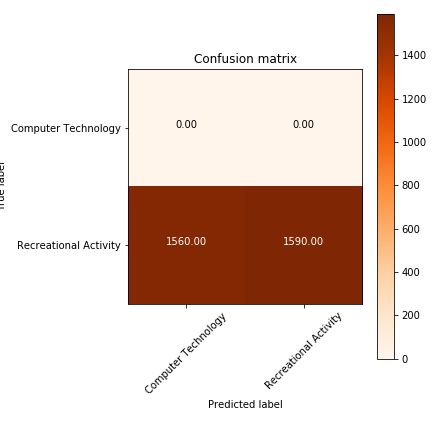
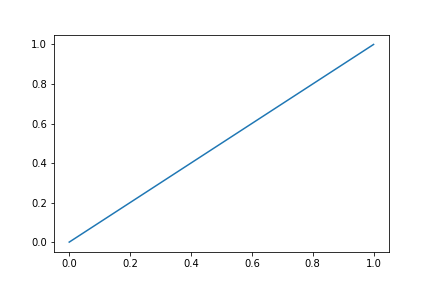
* + LSI (min\_df = 5)



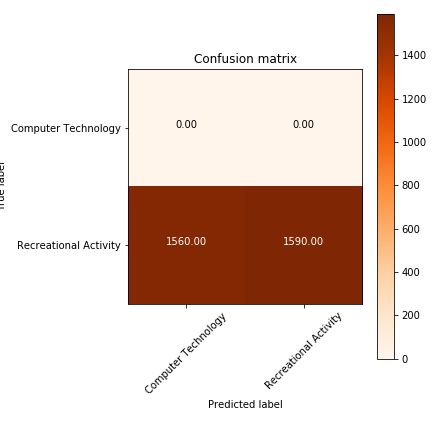
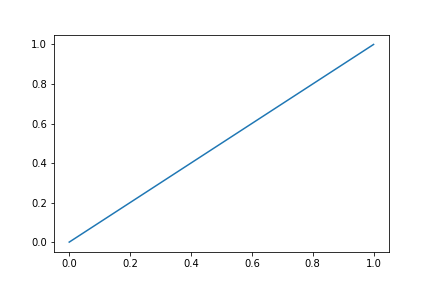
* + LSI (min\_df = 2)



* + NMF (min\_df = 5)



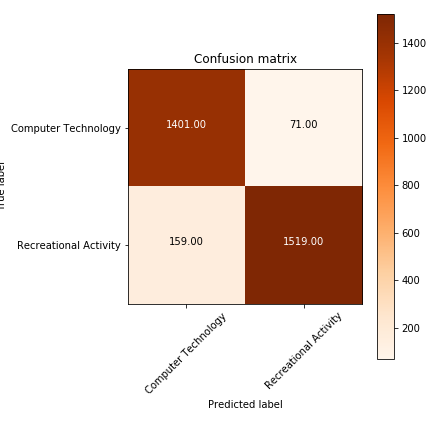
* + NMF (min\_df = 2)

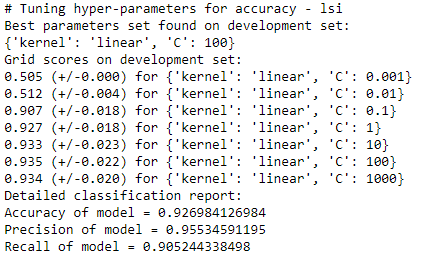


# Task (f) – 5 Fold Cross Validation

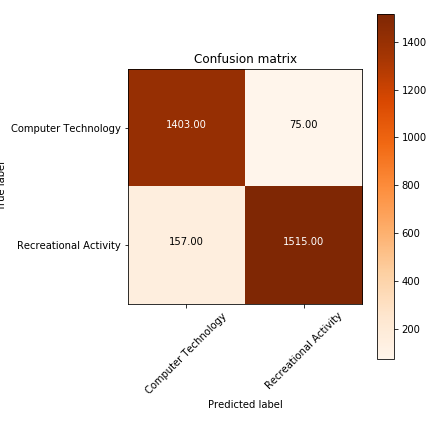
With the help of grid search CV function in sklearn, we were able to obtain the best parameter values with 5 fold cross validation.

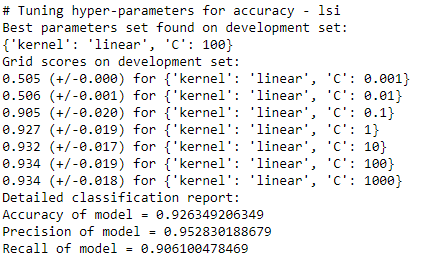
## LSI C=100 (min\_df = 5)



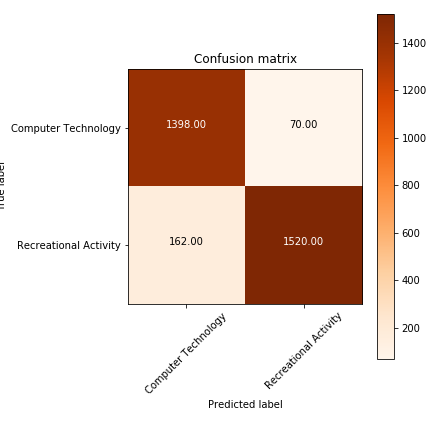


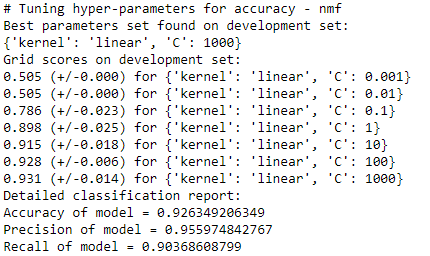
## LSI C=100 (min\_df = 2)



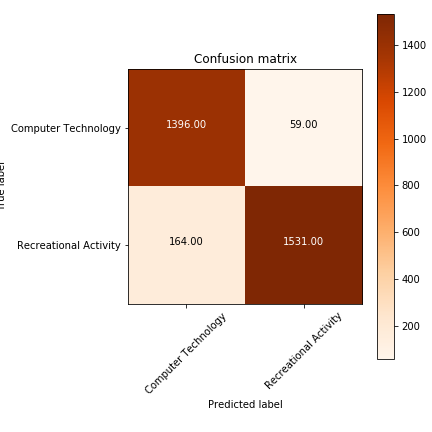


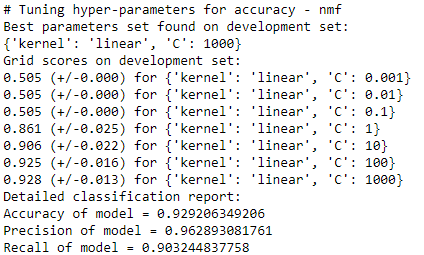
## NMF C=100 (min\_df = 5)





## NMF C=100 (min\_df = 2)



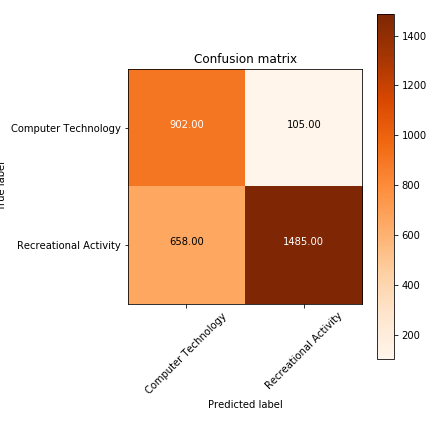
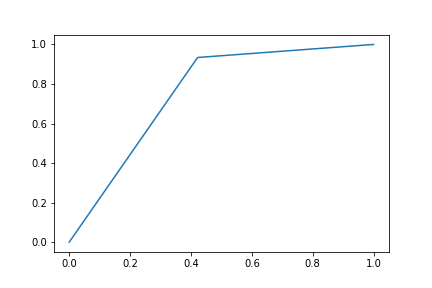


## 

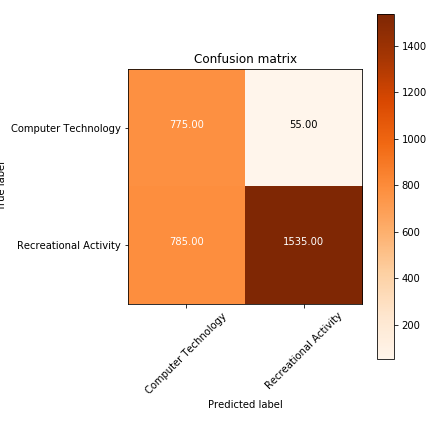
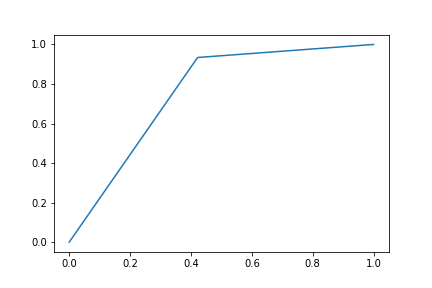
# Task (g) – Naïve Bayes

Instead of just trying multinomial Naïve Bayes classifier, we also experimented with Gaussian and Bernoulli Naïve Bayes classifier. We observe that all variants of Naïve Bayes have same results. The results have been recorded in the table above and the confusion matrix along with the ROC curve has been plotted below.

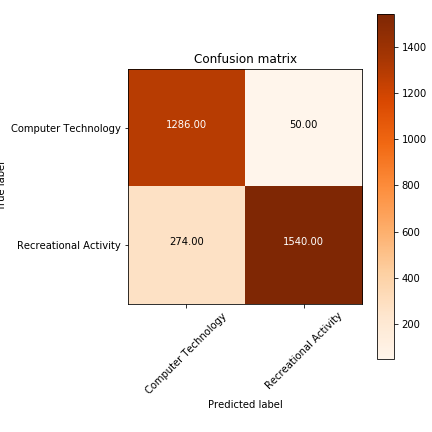
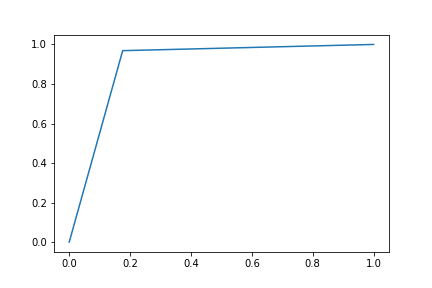
## Gaussian Naïve Bayes (LSI) (min\_df = 5)



## Gaussian Naïve Bayes (LSI) (min\_df = 2)

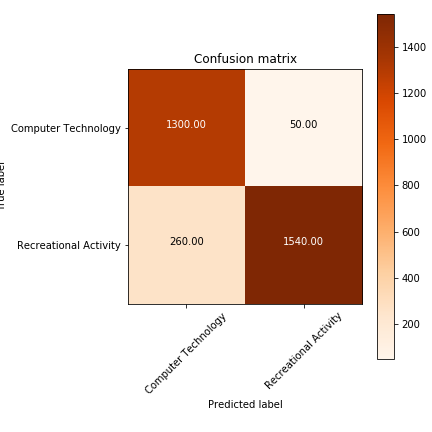
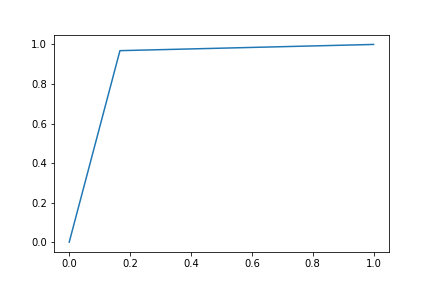


## Gaussian Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes (NMF) (min\_df = 5)



All versions of Naïve Bayes resulted in the same confusion report and RoC curve.

## Gaussian Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes (NMF) (min\_df = 2)

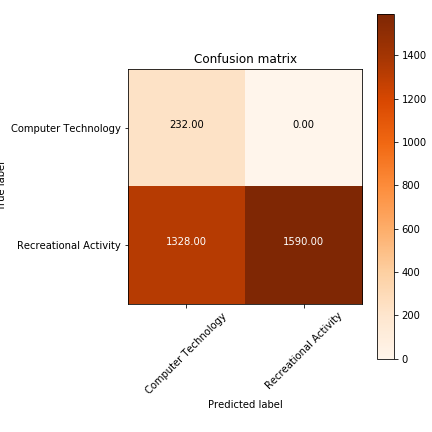
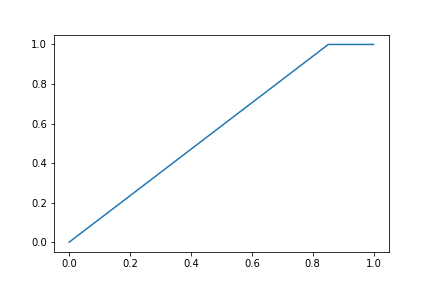


All versions of Naïve Bayes resulted in the same confusion report and RoC curve.

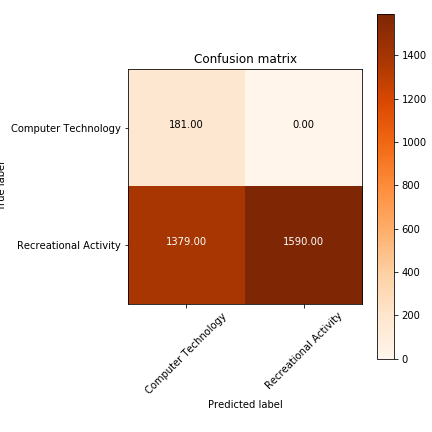
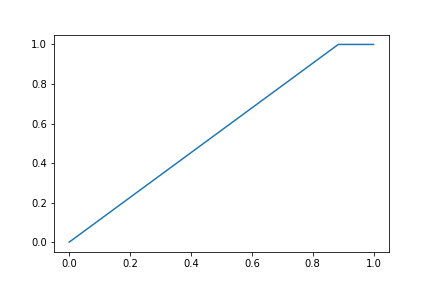
# Task (h,i) – Logistic Regression

## Without regularization

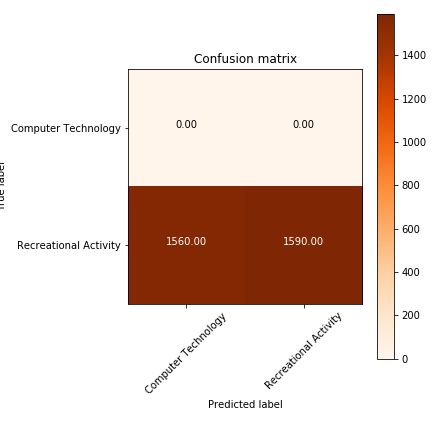
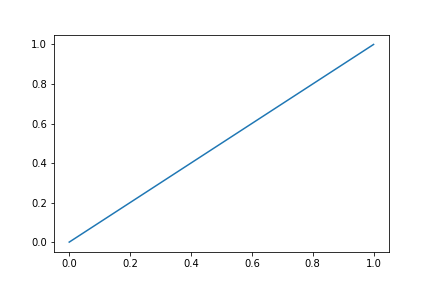
### LSI (min\_df = 5)



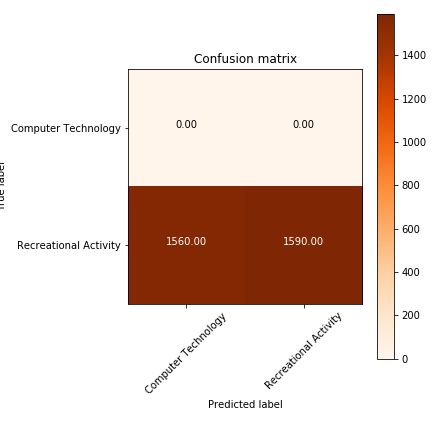
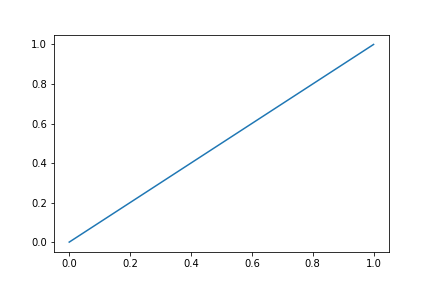
### LSI (min\_df = 2)



### NMF (min\_df = 5)

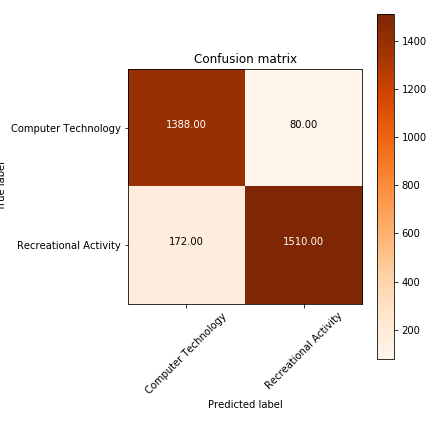
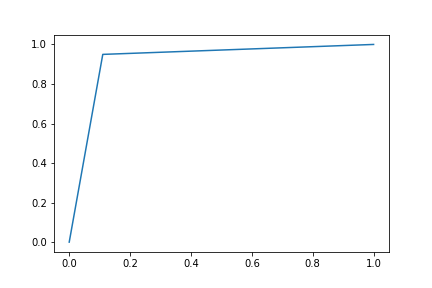


### NMF (min\_df = 2)

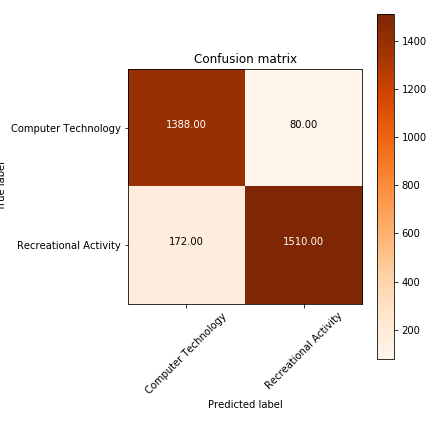
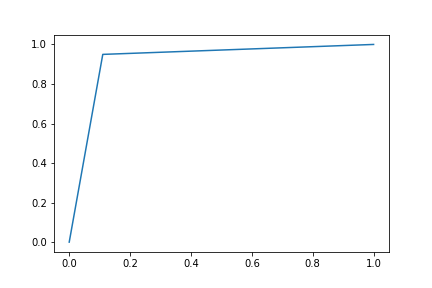


## L1 Regularization

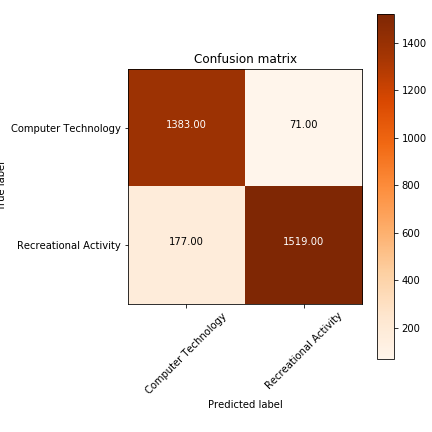
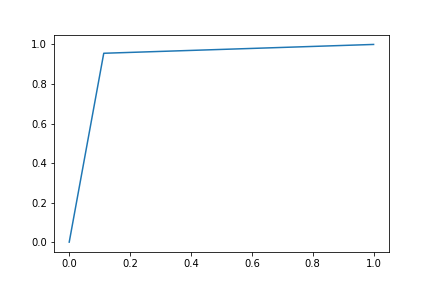
### LSI (min\_df = 5)



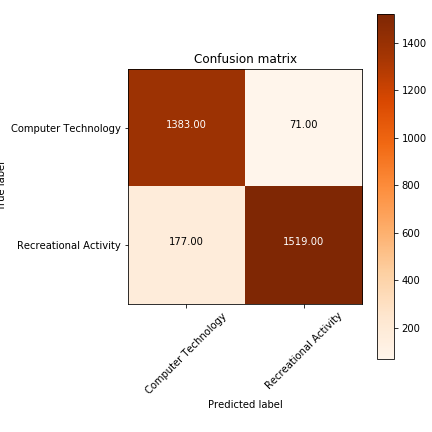
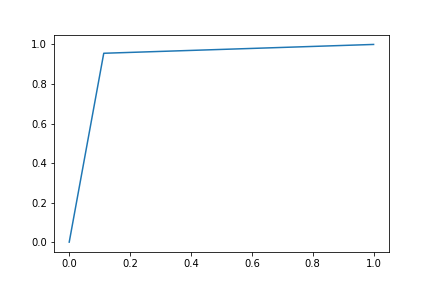
### LSI (min\_df = 2)



### NMF (min\_df = 5)

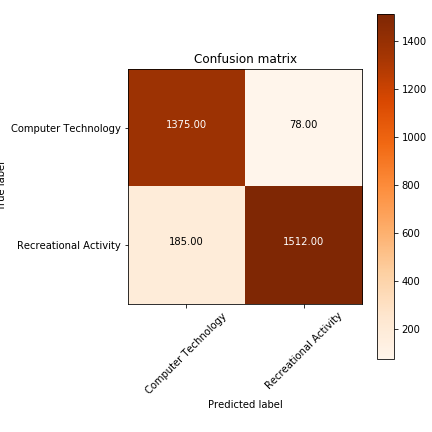
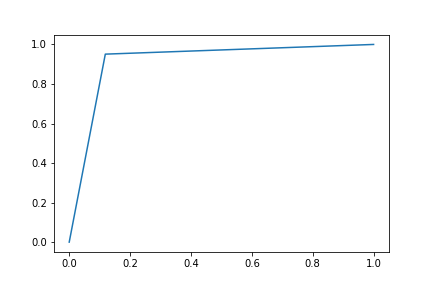


### NMF (min\_df = 2)

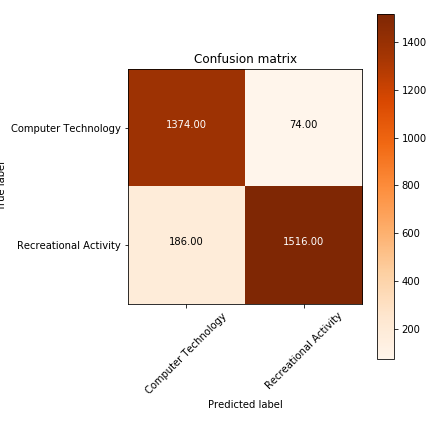
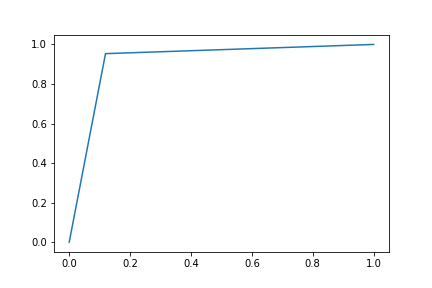


## L2 Regularization

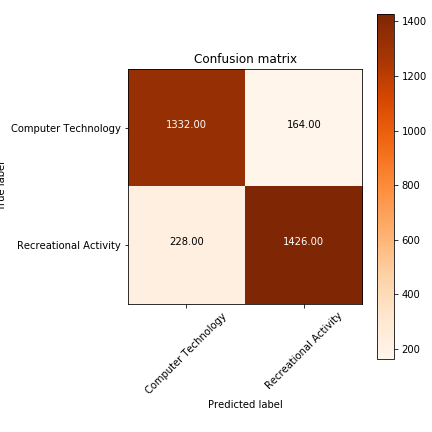
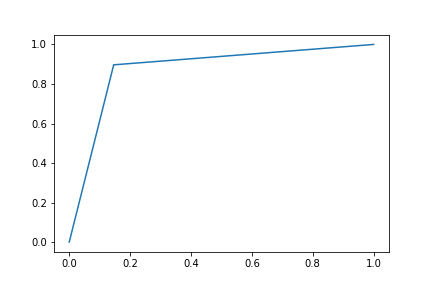
### LSI (min\_df = 5)



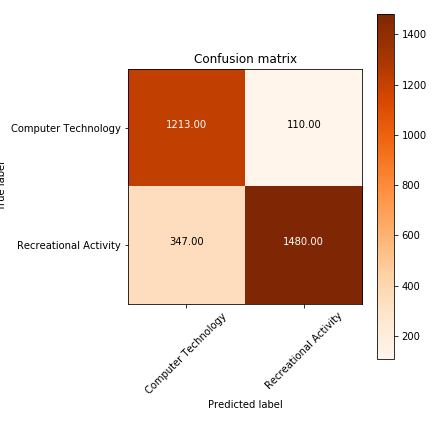
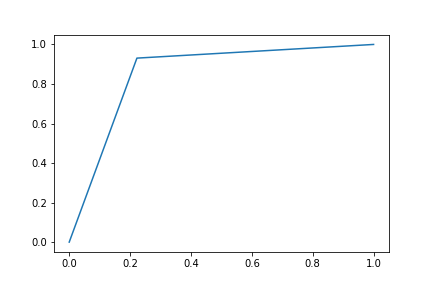
### LSI (min\_df = 2)



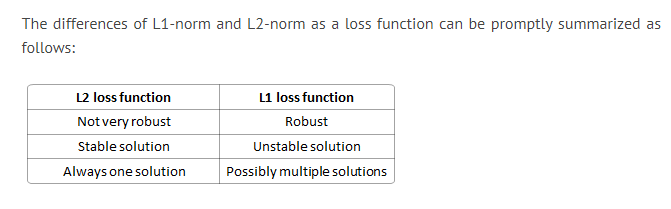
### NMF (min\_df = 5)



### NMF (min\_df = 2)



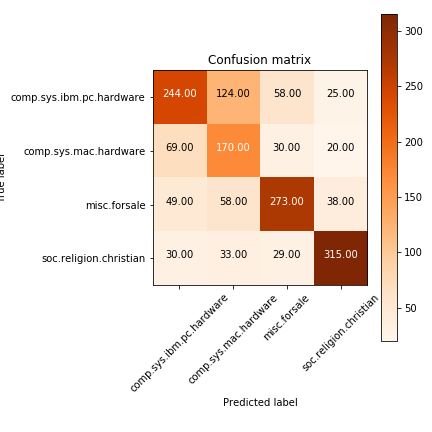
As we can observe, the error reduces as we increase the regularization parameter. Note that in sklearn, varying the regularization parameter is done by varying the penalty (C). Thus, in this graph, higher regularization parameter implies lower C. Higher C implies less impact of regularization. Therefore, as we increase the inverse of regularization, the error reduces.



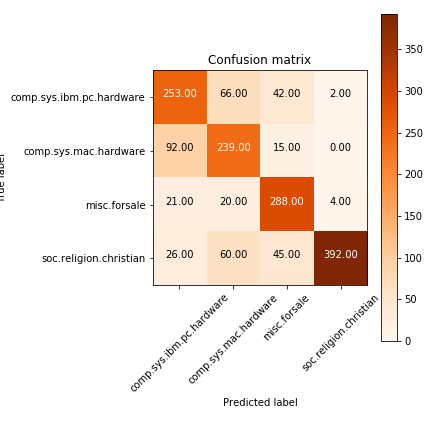
# Task (j) – Multiclass Classification

## Naïve Bayes: (min\_df = 5)

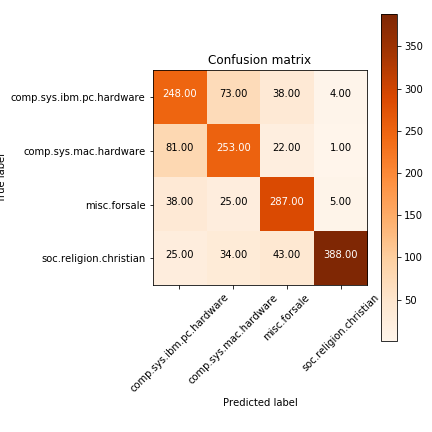
* + Bernoulli Naïve Bayes:
    - Accuracy of model = 0.64
    - Precision of model = [ 0.62 0.44 0.7 0.79]
    - Recall of model = [ 0.54 0.59 0.65 0.77]



* + Multinomial Naïve Bayes:
    - Accuracy of model = 0.74
    - Precision of model = [ 0.64 0.62 0.74 0.98]
    - Recall of model = [ 0.70 0.69 0.86 0.75]

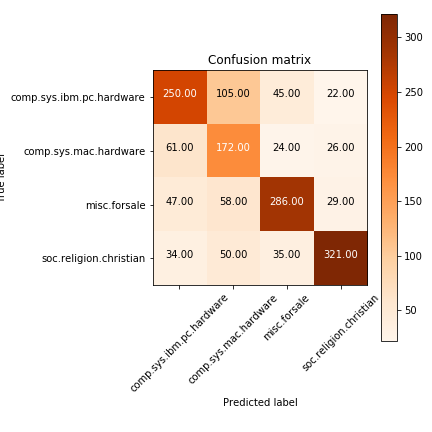


* + Gaussian Naïve Bayes:
    - Accuracy of model = 0.75
    - Precision of model = [ 0.63 0.66 0.74 0.98]
    - Recall of model = [ 0.68 0.71 0.81 0.79]

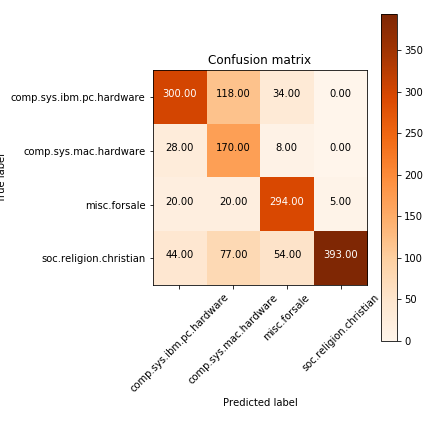


## Naïve Bayes: (min\_df = 2)

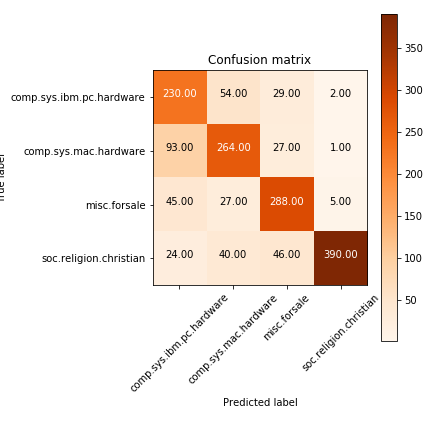
* + Bernoulli Naïve Bayes:
    - Accuracy of model = 0.66
    - Precision of model = [ 0.64 0.45 0.73 0.80]
    - Recall of model = [ 0.59 0.61 0.68 0.73]



* + Multinomial Naïve Bayes:
    - Accuracy of model = 0.74
    - Precision of model = [ 0.74 0.44 0.75 0.99]
    - Recall of model = [ 0.66 0.83 0.87 0.69]

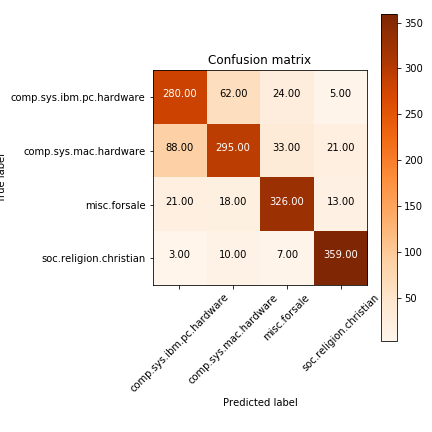


* + Gaussian Naïve Bayes:
    - Accuracy of model = 0.75
    - Precision of model = [ 0.59 0.69 0.74 0.98]
    - Recall of model = [ 0.73 0.69 0.79 0.78]

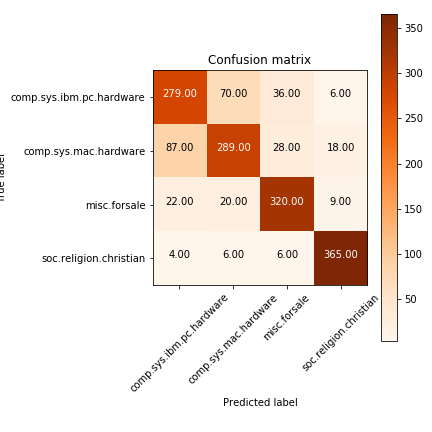


## SVM – One vs One (min\_df = 5)

* + LSI
    - Accuracy of model = 0.81
    - Precision of model = [ 0.71 0.77 0.84 0.90]
    - Recall of model = [ 0.75 0.68 0.86 0.95]

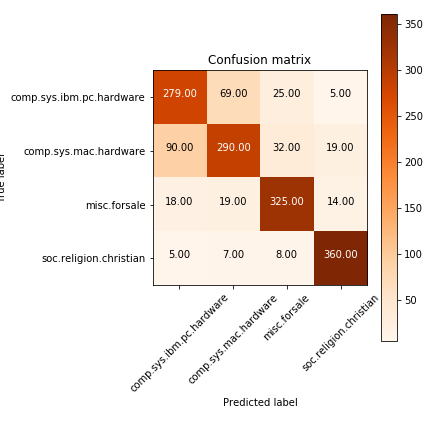


* + NMF
    - Accuracy of model = 0.80
    - Precision of model = [ 0.71 0.75 0.82 0.92]
    - Recall of model = [ 0.71 0.68 0.86 0.96]

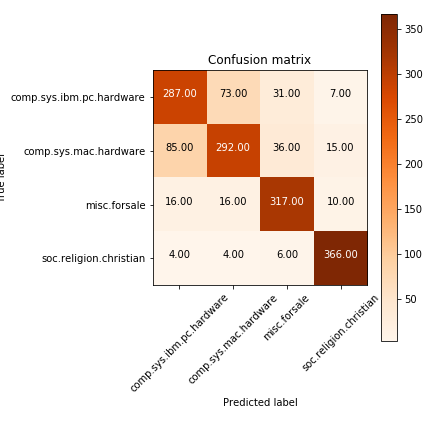


## SVM – One vs One (min\_df = 2)

* + LSI
    - Accuracy of model = 0.80
    - Precision of model = [ 0.71 0.75 0.83 0.90]
    - Recall of model = [ 0.74 0.67 0.86 0.95]

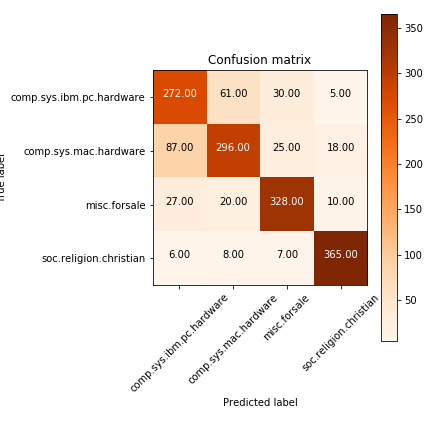


* + NMF
    - Accuracy of model = 0.81
    - Precision of model = [ 0.73 0.76 0.81 0.92]
    - Recall of model = [ 0.72 0.68 0.88 0.96]

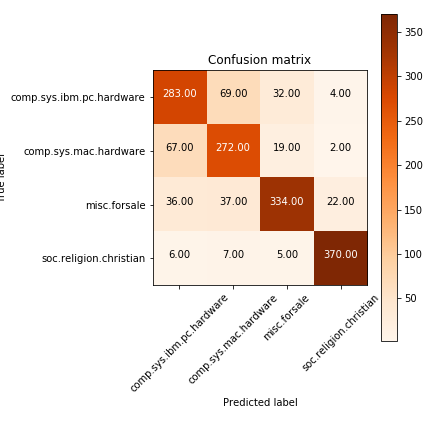


## SVM – One vs Rest (min\_df = 5)

* + LSI
    - Accuracy of model = 0.81
    - Precision of model = [ 0.69 0.77 0.84 0.92]
    - Recall of model = [ 0.74 0.69 0.85 0.95]

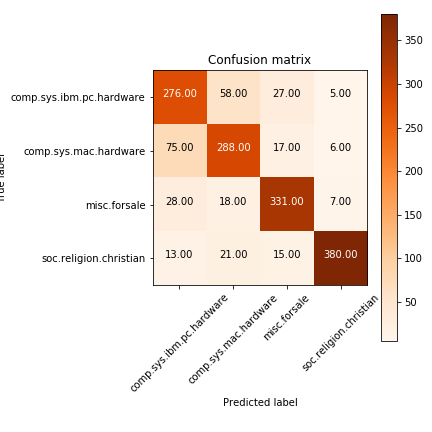


* + NMF
    - Accuracy of model = 0.80
    - Precision of model = [ 0.72 0.70 0.86 0.93]
    - Recall of model = [ 0.73 0.76 0.78 0.95]



## SVM – One vs Rest (min\_df = 2)

* + LSI
    - Accuracy of model = 0.81
    - Precision of model = [ 0.70 0.75 0.85 0.95]
    - Recall of model = [ 0.75 0.75 0.86 0.89]



* + NMF
    - Accuracy of model = 0.80
    - Precision of model = [ 0.76 0.69 0.83 0.92]
    - Recall of model = [ 0.68 0.77 0.85 0.94]

