REDDIT REPORT

Nihal Parchand **Rohit Kunjilikattil**

30.04.2019

FOUNDATIONS OF INTELLIGENT SYSTEMS

INTRODUCTION

Reddit, also known as the front page of the internet, is home to thousands of communities with a never ending source of information. Whatever you interest, be it soccer or video games or pottery, there is a reddit thread (subreddit) for it where people with the same interests can converse freely and endlessly.

One of the integral part of reddit are the reddit post in which people can share text, images, videos, gif, etc. A reddit post typically belongs to a reddit thread which is known as a subreddit. The goal of this project was to develop classifiers that can correctly predict the subreddit of an unlabelled post.

Let me give you a walkthrough of the project. We have downloaded the data and have divided it into training,development and test. After we have developed a classifier, we train it on the training set of the data. After the training is completed, we run it on a separate set of data known as the test data and we analyse how well was the classification algorithm able to predict the subreddits correctly. Lastly we have compared the results of the different approaches that we have undertaken.

Approach

This project was basically a data classification challenge as the classifier model had to predict the subreddit based on the text in the post. So we used the basic approaches which help us develop such a classifier.

1. Support Vector Classifier:-

   A Support Vector Classifier (SVC) is a discriminative classifier formally defined by a separating hyperplane. In other words, SVC basically divides the plane of the current inputs into optimal planes which can categorize new examples. This imaginary divide between the planes is called a hyperplane. For a 2d input plane, the hyperplane is a line through the 2d space that divides it.

2. Random Forest Classifier:-

   A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. It basically creates a collection of decision trees and trains them to increase the overall accuracy. In simple words, it creates a bunch of decision trees randomly, trains the model on these trees and then merges them together to get a more stable prediction

3. Logistic Regression :-

   Logistic Regression is a type of predictive analysis that used to define the relationship between one dependant binary variable and one or more independent variables. Unlike normal regression, instead of predicting numeric values, logistic regression predicts the probability of a certain input belonging to a class.

4. LSTM :-

   Long Short Term Memory : - LSTM is an artificial recurrent neural network. It is different than normal feedforward neural network because LSTM has feedback connections. LSTM can process sequences of data, while retaining information about the data over the sequence. LSTM cell consists of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time and the three gates are used to manage the flow of the information in and out of the cell.

Installation

For running this project you will need to install scikit-learn which needs some prerequisites:

Scikit-learn requires:

Python (>= 2.7 or >= 3.4),

NumPy (>= 1.8.2),

SciPy (>= 0.13.3).

Run this command on your terminal to install scikit-learn

```
pip install -U scikit-learn
```

We have used anaconda python version 2.7 and jupyter notebooks for our project.

Anaconda python version 2.7 download link:

https://www.anaconda.com/distribution/

You also need to install PRAW which is a reddit API for python. PRAW, an acronym for "Python Reddit API Wrapper", is a python package that allows for simple access to Reddit's API. PRAW aims to be easy to use and internally follows all of Reddit's API rules. With PRAW there's no need to introduce sleep calls in your code. Give your client an appropriate user agent and you're set.

PRAW is supported on python 2.7, 3.3, 3.4, 3.5 and 3.6. The recommended way to install PRAW is via pip.

Run this command on your terminal to install scikit-learn

pip install praw

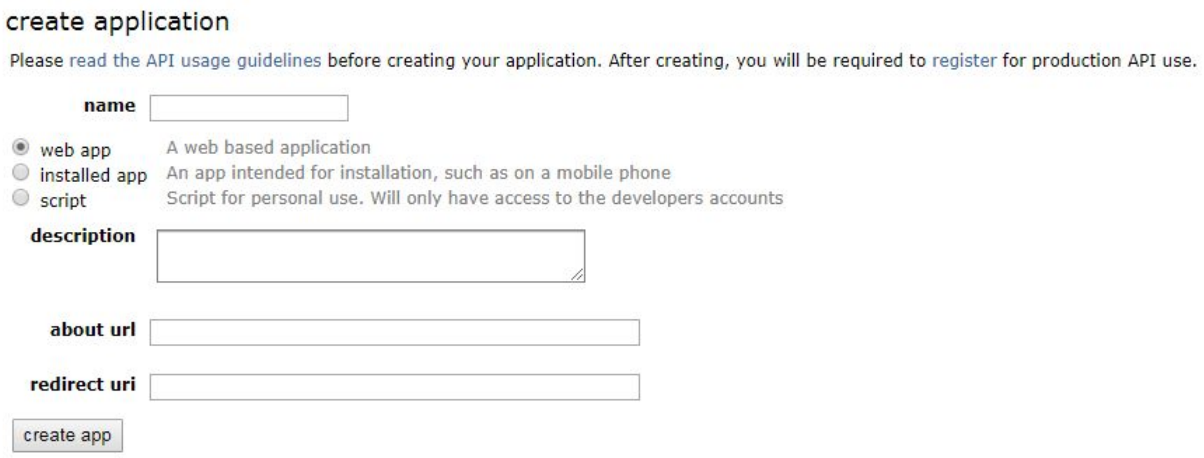
To install the latest development version of PRAW run the following instead:

pip install --upgrade https://github.com/praw-dev/praw/archive/master.zip

Setting up the REDDIT Account and APP

First create an account on Reddit.com and then go to this link for creating your reddit app which will be needed for connection purposes

https://www.reddit.com/prefs/apps



create application

Please read the API usage guidelines before creating your application. After creating, you will be required to register for production API use.

name

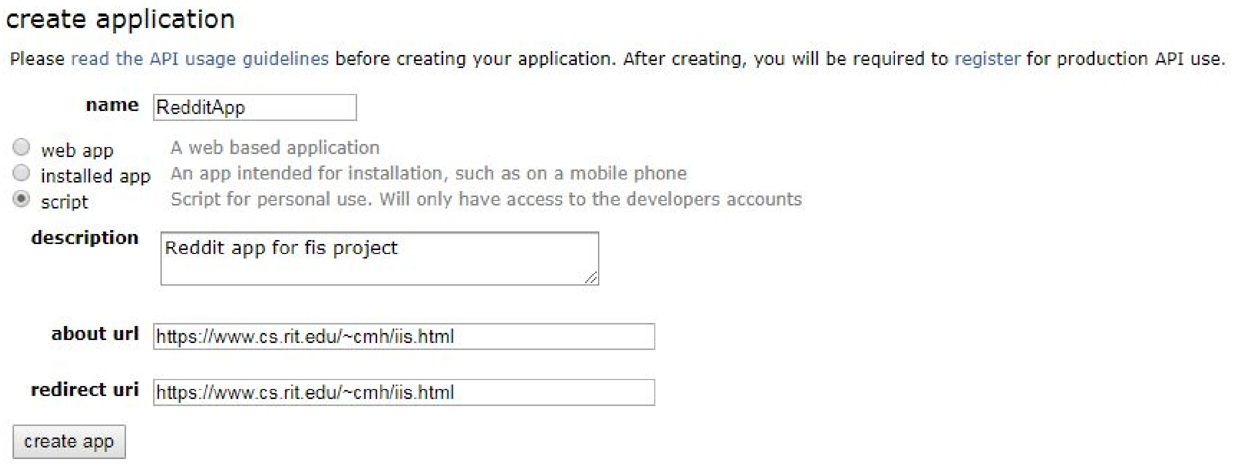
● web app — A web based application
○ installed app — An app intended for installation, such as on a mobile phone
○ script — Script for personal use. Will only have access to the developers accounts

description

about url

redirect uri

Enter your app name and select script option. Then enter details about the description about your app and enter https://www.cs.rit.edu/~cmh/iis.html for the last two fields.



create application

Please read the API usage guidelines before creating your application. After creating, you will be required to register for production API use.

name: RedditApp

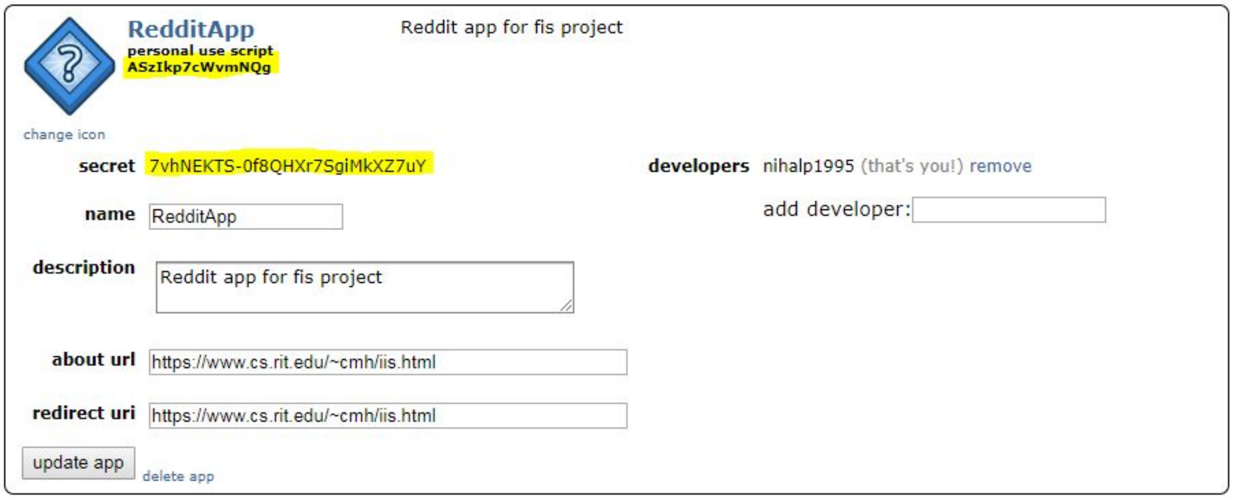
○ web app — A web based application
○ installed app — An app intended for installation, such as on a mobile phone
● script — Script for personal use. Will only have access to the developers accounts

description: Reddit app for fis project

about url: https://www.cs.rit.edu/~cmh/iis.html

redirect uri: https://www.cs.rit.edu/~cmh/iis.html

You need to save the highlighted information from the app details shown below



RedditApp
personal use script
ASzIkp7cWvmNQg

Reddit app for fis project

secret 7vhNEKTS-0f8QHXr7SgiMkXZ7uY

developers nihalp1995 (that's you!) remove

add developer:

name RedditApp

description Reddit app for fis project

about url https://www.cs.rit.edu/~cmh/iis.html

redirect uri https://www.cs.rit.edu/~cmh/iis.html

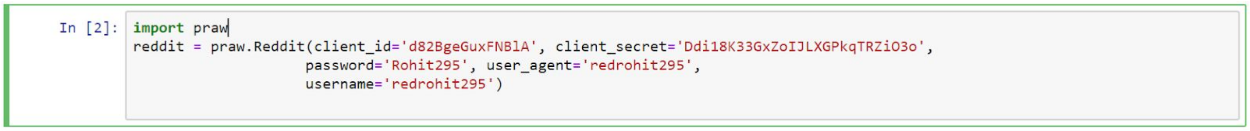
The first highlighted text is the client_id

The second highlighted text is the client_secret

Implementation

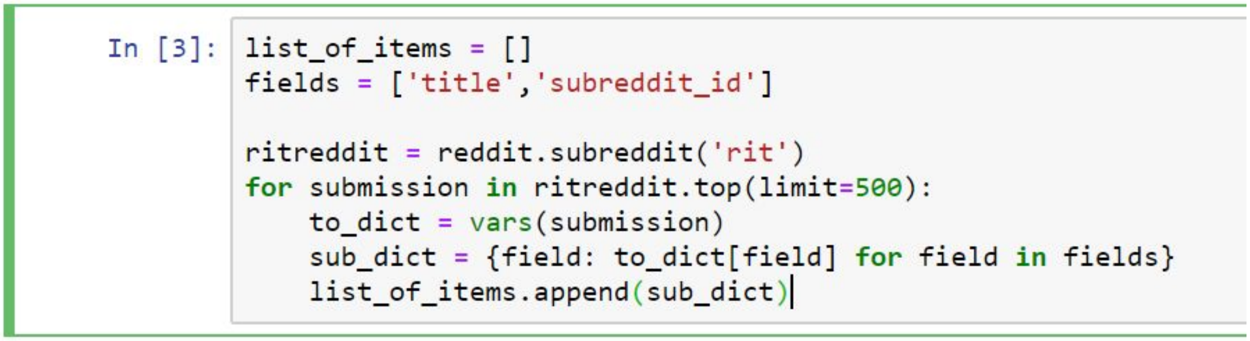
Step 1: Instantiating a PRAW instance for interacting with Reddit.

```
import praw

reddit = praw.Reddit(client_id='CLIENT_ID', client_secret="CLIENT_SECRET",

                password='PASSWORD', user_agent='USERAGENT',

                username='USERNAME')
```



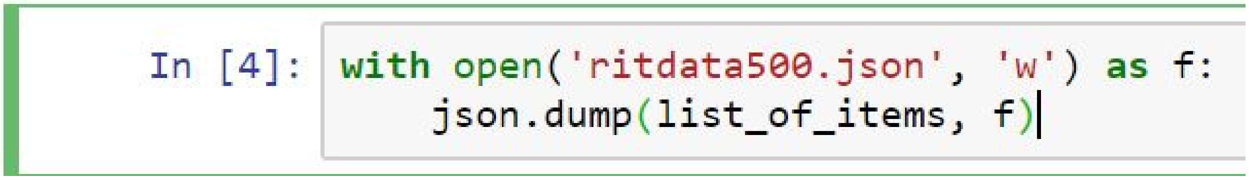
```
In [2]: import praw
        reddit = praw.Reddit(client_id='d82BgeGuxFNBlA', client_secret='Ddi18K33GxZoIJLXGPkqTRZiO3o',
                             password='Rohit295', user_agent='redrohit295',
                             username='redrohit295')
```

For our project we used the subreddit rit and we extracted the top 500 title and subreddit_id and stored it in a dictionary.



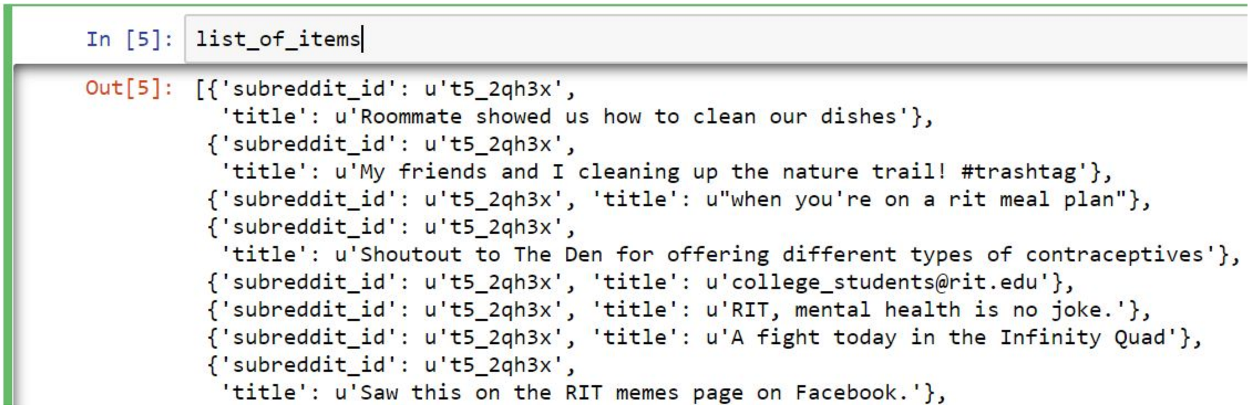
```
In [3]: list_of_items = []
        fields = ['title','subreddit_id']

        ritreddit = reddit.subreddit('rit')
        for submission in ritreddit.top(limit=500):
            to_dict = vars(submission)
            sub_dict = {field: to_dict[field] for field in fields}
            list_of_items.append(sub_dict)
```

After this we stored the data in a json file format.



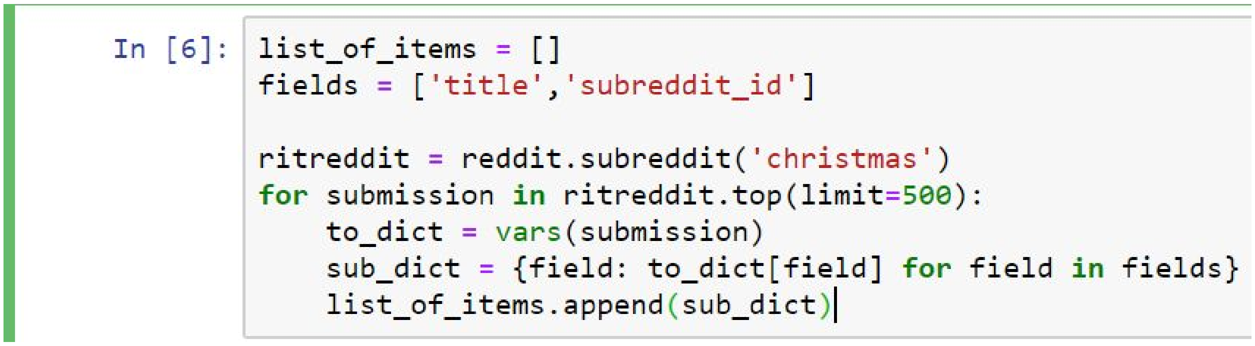
```
In [4]: with open('ritdata500.json', 'w') as f:
            json.dump(list_of_items, f)
```

A snapshot of how the data looks like in the dictionary.



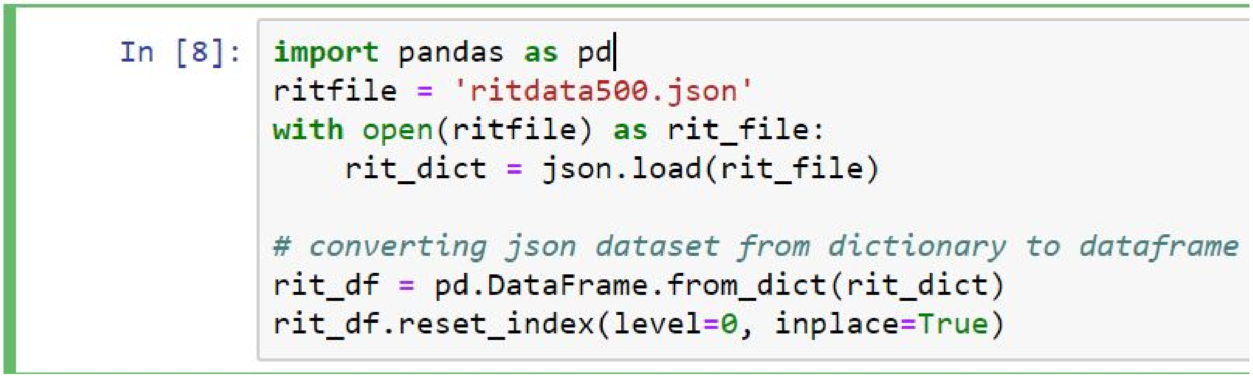
```
In [5]: list_of_items

Out[5]: [{'subreddit_id': u't5_2qh3x',
          'title': u'Roommate showed us how to clean our dishes'},
         {'subreddit_id': u't5_2qh3x',
          'title': u'My friends and I cleaning up the nature trail! #trashtag'},
         {'subreddit_id': u't5_2qh3x', 'title': u"when you're on a rit meal plan"},
         {'subreddit_id': u't5_2qh3x',
          'title': u'Shoutout to The Den for offering different types of contraceptives'},
         {'subreddit_id': u't5_2qh3x', 'title': u'college_students@rit.edu'},
         {'subreddit_id': u't5_2qh3x', 'title': u'RIT, mental health is no joke.'},
         {'subreddit_id': u't5_2qh3x', 'title': u'A fight today in the Infinity Quad'},
         {'subreddit_id': u't5_2qh3x',
          'title': u'Saw this on the RIT memes page on Facebook.'},
```

Similarly we extract the subreddit id and title from another subreddit christmas



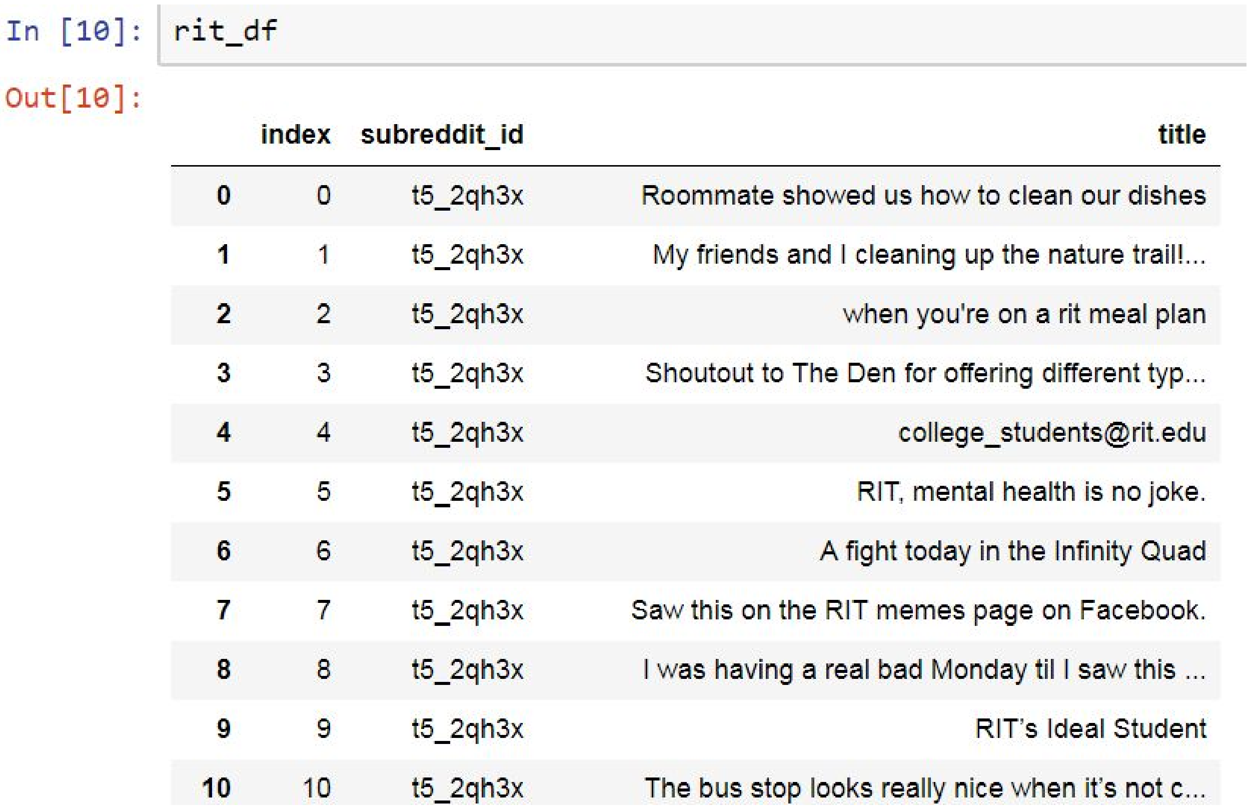
```
In [6]: list_of_items = []
        fields = ['title','subreddit_id']

        ritreddit = reddit.subreddit('christmas')
        for submission in ritreddit.top(limit=500):
            to_dict = vars(submission)
            sub_dict = {field: to_dict[field] for field in fields}
            list_of_items.append(sub_dict)
```

Now we read the json file and store the data in a pandas dataframe.



```
In [8]: import pandas as pd
        ritfile = 'ritdata500.json'
        with open(ritfile) as rit_file:
            rit_dict = json.load(rit_file)

        # converting json dataset from dictionary to dataframe
        rit_df = pd.DataFrame.from_dict(rit_dict)
        rit_df.reset_index(level=0, inplace=True)
```

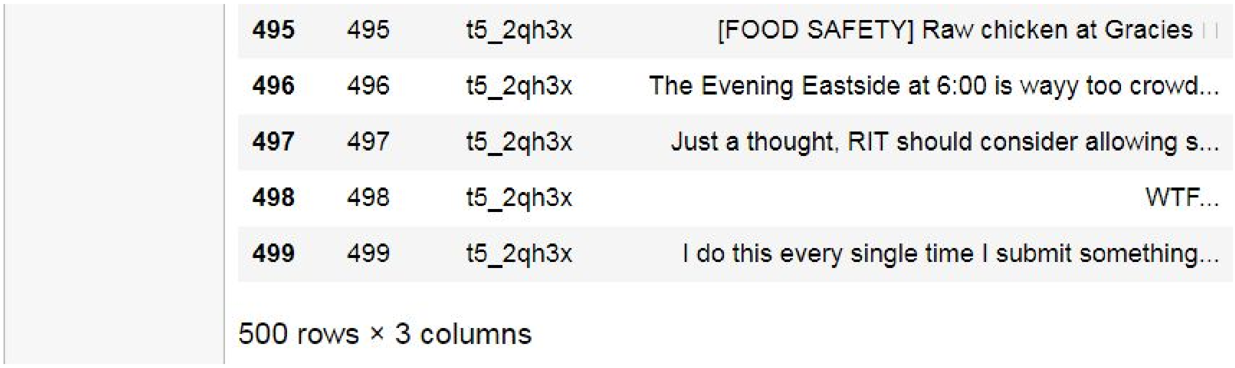
This is how the data looks in dataframe



```
In [10]: rit_df
```

Out[10]:

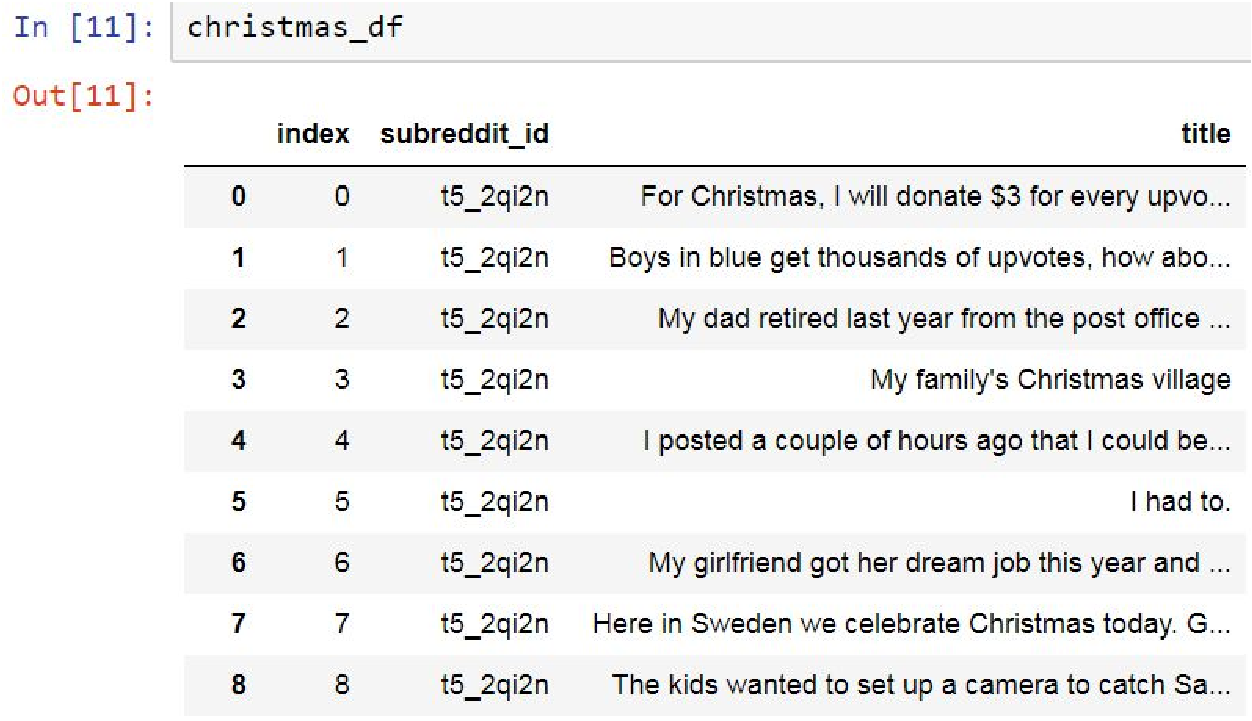
	index	subreddit_id	title
0	0	t5_2qh3x	Roommate showed us how to clean our dishes
1	1	t5_2qh3x	My friends and I cleaning up the nature trail!...
2	2	t5_2qh3x	when you're on a rit meal plan
3	3	t5_2qh3x	Shoutout to The Den for offering different typ...
4	4	t5_2qh3x	college_students@rit.edu
5	5	t5_2qh3x	RIT, mental health is no joke.
6	6	t5_2qh3x	A fight today in the Infinity Quad
7	7	t5_2qh3x	Saw this on the RIT memes page on Facebook.
8	8	t5_2qh3x	I was having a real bad Monday til I saw this ...
9	9	t5_2qh3x	RIT’s Ideal Student
10	10	t5_2qh3x	The bus stop looks really nice when it’s not c...



495	495	t5_2qh3x	[FOOD SAFETY] Raw chicken at Gracies □□
496	496	t5_2qh3x	The Evening Eastside at 6:00 is wayy too crowd...
497	497	t5_2qh3x	Just a thought, RIT should consider allowing s...
498	498	t5_2qh3x	WTF...
499	499	t5_2qh3x	I do this every single time I submit something...

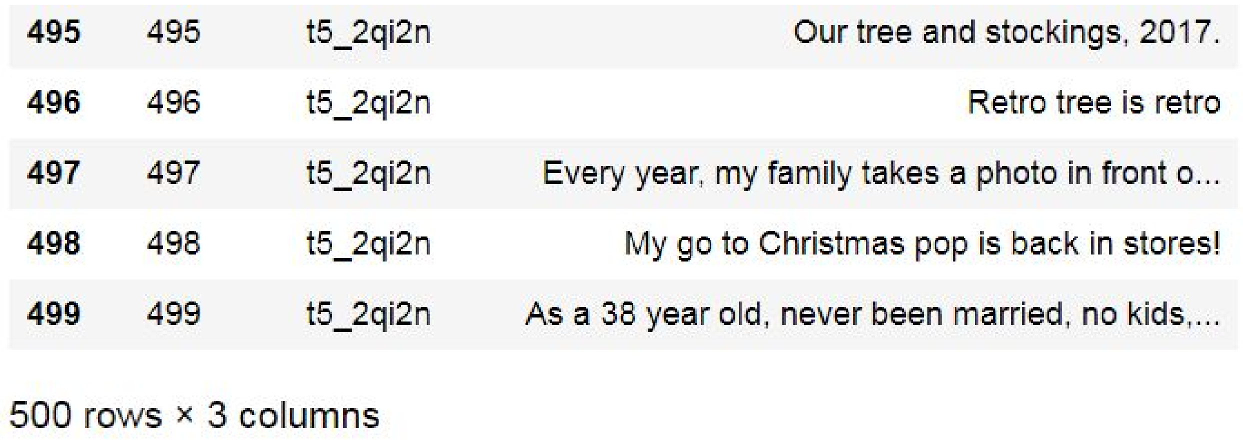
500 rows × 3 columns

Similarly for the other subreddit christmas.



```
In [11]: christmas_df
```

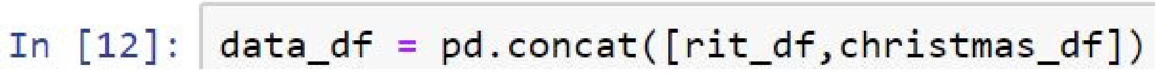
```
Out[11]:
```



	index	subreddit_id	title
0	0	t5_2qi2n	For Christmas, I will donate $3 for every upvo...
1	1	t5_2qi2n	Boys in blue get thousands of upvotes, how abo...
2	2	t5_2qi2n	My dad retired last year from the post office ...
3	3	t5_2qi2n	My family's Christmas village
4	4	t5_2qi2n	I posted a couple of hours ago that I could be...
5	5	t5_2qi2n	I had to.
6	6	t5_2qi2n	My girlfriend got her dream job this year and ...
7	7	t5_2qi2n	Here in Sweden we celebrate Christmas today. G...
8	8	t5_2qi2n	The kids wanted to set up a camera to catch Sa...
495	495	t5_2qi2n	Our tree and stockings, 2017.
496	496	t5_2qi2n	Retro tree is retro
497	497	t5_2qi2n	Every year, my family takes a photo in front o...
498	498	t5_2qi2n	My go to Christmas pop is back in stores!
499	499	t5_2qi2n	As a 38 year old, never been married, no kids,...

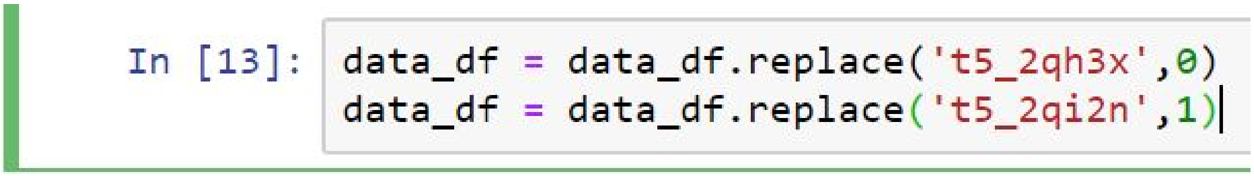
500 rows × 3 columns

After this we concat the data into one single dataframe.



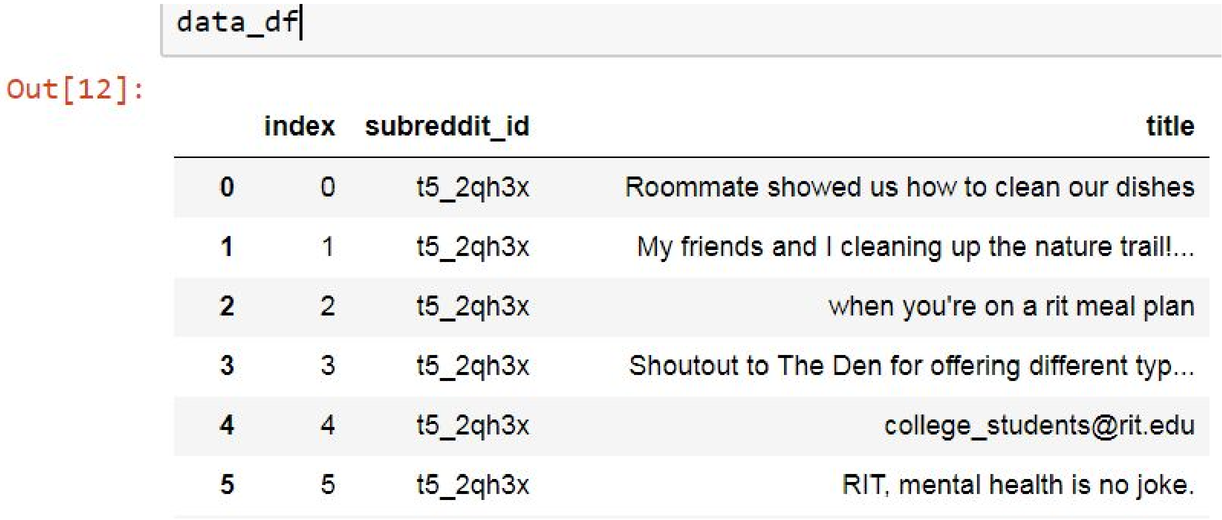
```
In [12]: data_df = pd.concat([rit_df,christmas_df])
```

Replacing the subreddit_id with 0 and 1 as it will be our target attribute.



```
In [13]: data_df = data_df.replace('t5_2qh3x',0)
         data_df = data_df.replace('t5_2qi2n',1)
```

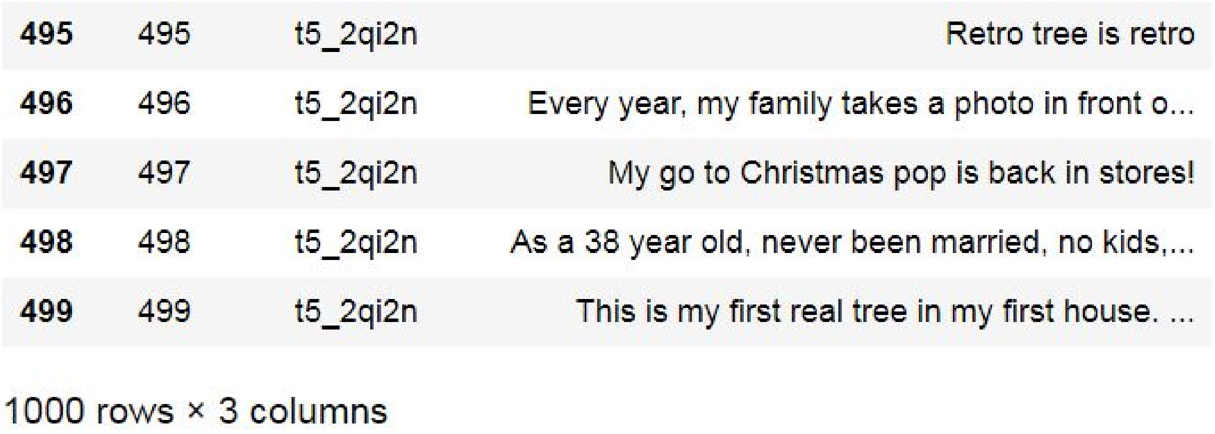
After this the data frame looks like this



```
data_df
```

Out[12]:

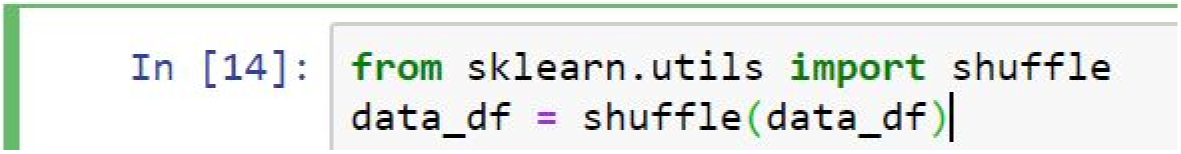
	index	subreddit_id	title
0	0	t5_2qh3x	Roommate showed us how to clean our dishes
1	1	t5_2qh3x	My friends and I cleaning up the nature traill!...
2	2	t5_2qh3x	when you're on a rit meal plan
3	3	t5_2qh3x	Shoutout to The Den for offering different typ...
4	4	t5_2qh3x	college_students@rit.edu
5	5	t5_2qh3x	RIT, mental health is no joke.



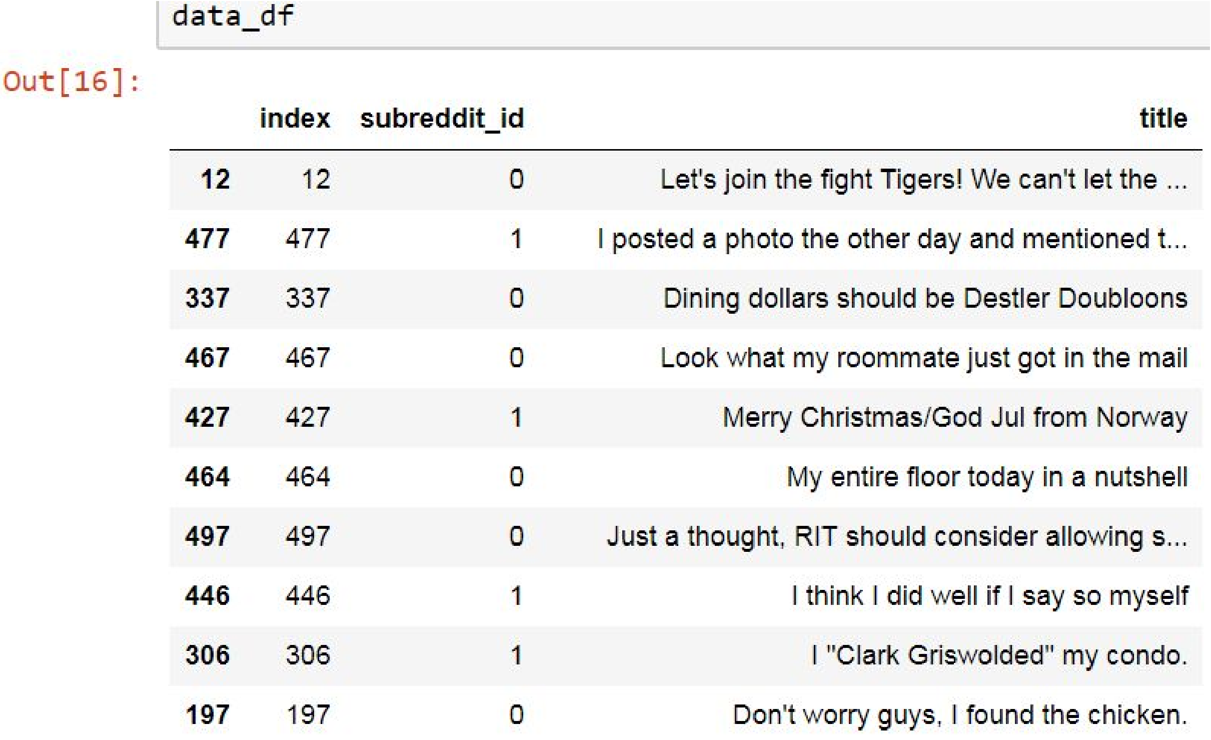
495	495	t5_2qi2n	Retro tree is retro
496	496	t5_2qi2n	Every year, my family takes a photo in front o...
497	497	t5_2qi2n	My go to Christmas pop is back in stores!
498	498	t5_2qi2n	As a 38 year old, never been married, no kids,...
499	499	t5_2qi2n	This is my first real tree in my first house. ...

1000 rows × 3 columns

We shuffle the dataset every time before we split it into training, development and test sets.



```
In [14]: from sklearn.utils import shuffle
         data_df = shuffle(data_df)
```

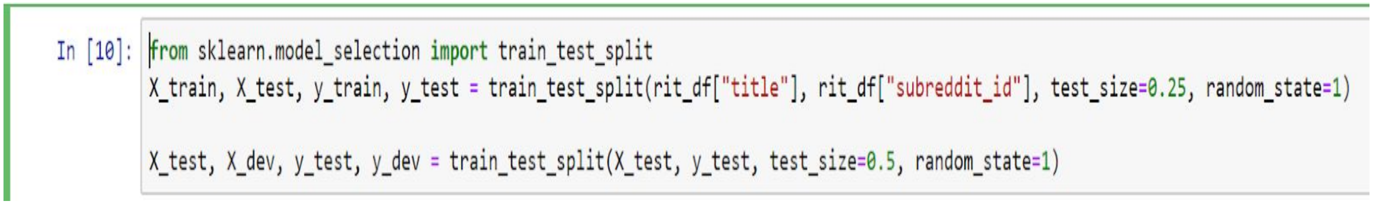


```
data_df
```

Out[16]:

	index	subreddit_id	title
12	12	0	Let's join the fight Tigers! We can't let the ...
477	477	1	I posted a photo the other day and mentioned t...
337	337	0	Dining dollars should be Destler Doubloons
467	467	0	Look what my roommate just got in the mail
427	427	1	Merry Christmas/God Jul from Norway
464	464	0	My entire floor today in a nutshell
497	497	0	Just a thought, RIT should consider allowing s...
446	446	1	I think I did well if I say so myself
306	306	1	I "Clark Griswolded" my condo.
197	197	0	Don't worry guys, I found the chicken.

After this step we split the data into training, development and test data sets. For our project we split the data in 50% / 25% / 25% ratio.



```
In [10]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(rit_df["title"], rit_df["subreddit_id"], test_size=0.25, random_state=1)

         X_test, X_dev, y_test, y_dev = train_test_split(X_test, y_test, test_size=0.5, random_state=1)
```

Pseudo code for SVM :-

```
candidateSV = { nearest pair from opposite classes }
while there are conflicting points, do the following
    Find a conflicting point
    candidateSV = candidateSV ∪ violator
    if any $\alpha$p < 0 due to addition of c to S then
        candidateSV = candidateSV / p
        repeat till all such points are pruned
    end if
end while
```

Pseudo code for Random Forest :-

The random Forest pseudocode has two parts:- creation and prediction

Pseudo code for creation of Random Forest :-

1. Randomly select ‘k’ features from total of ‘m’ features
2. Out of all the ‘k’ features, calculate a node which will be the best split point and label it ‘d’.
3. Split ‘d’ into daughter nodes by splitting at the best split points again and again
4. Repeat steps 1 to 3 until a particular number of nodes has been reached.
5. Build the forest by repeating steps 1 to 4 ‘n’ times.

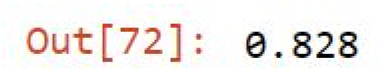
Pseudo code for Random Forest prediction:-

1. Takes the test features and use the rules of each decision tree to predict an outcome and store it.
2. Then we calculate vote for each predicted target.
3. Take the prediction with highest votes as the final prediction of the algorithm

RESULTS

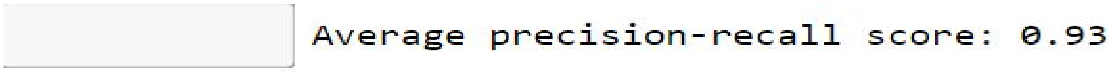
1. SVM

Accuracy for SVM:

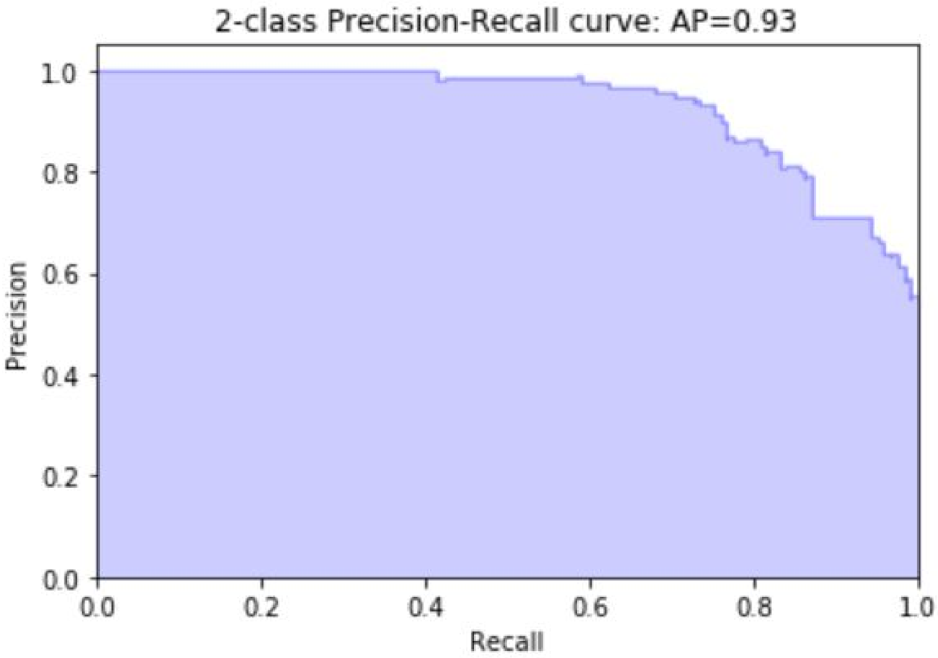


```
Out[72]:  0.828
```

Average Precision-Recall



```
Average precision-recall score: 0.93
```



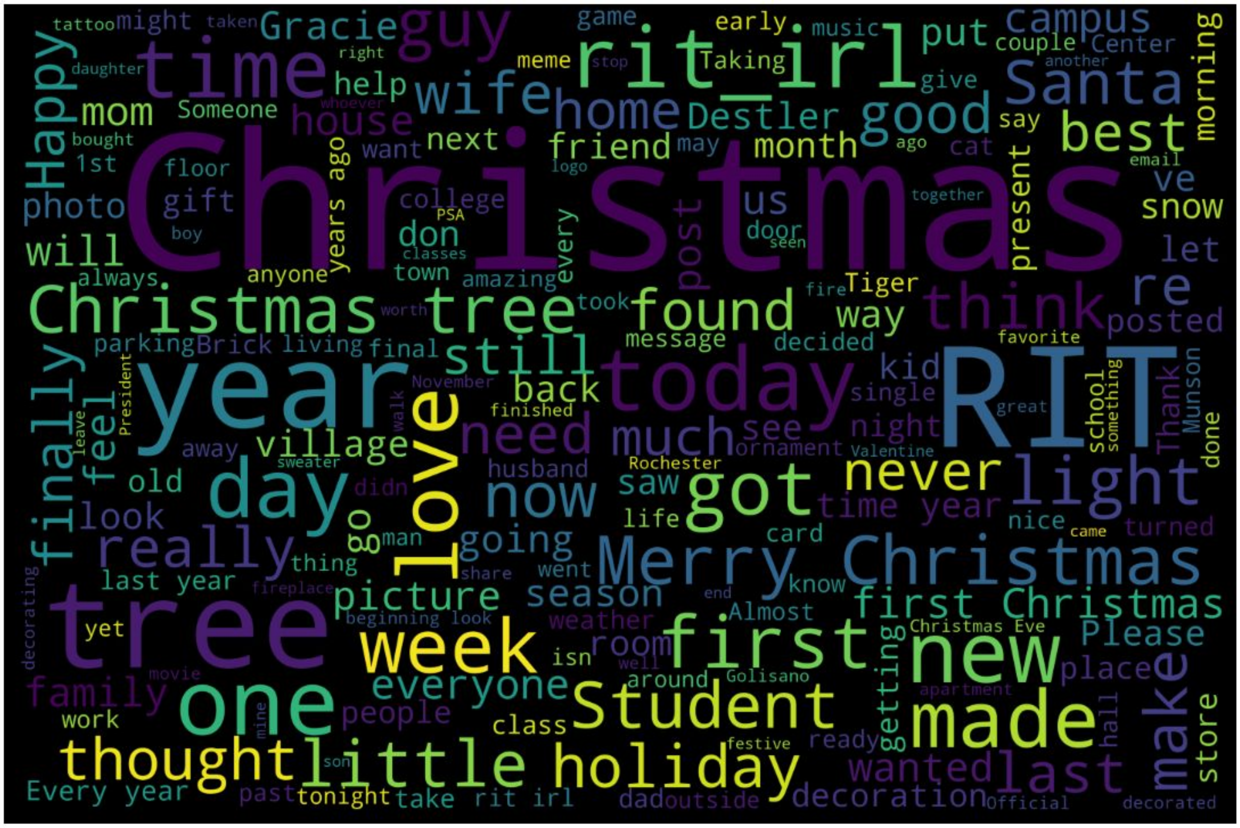
Precision-Recall curve :- Precision recall curve, similar to the ROC curve, is a plotting of the precision and recall for different threshold values. Precision is the ratio of true positives (positive outputs predicted correctly) to the to the sum of true positives and false positives(positive outputs predicted incorrectly). Recall is the ratio of true positives to the sum of true positives and false negatives(negative outcomes predicted incorrectly).

Precision = True Positives / (True Positives + False Positives)

Recall = True Positives / (True Positives + False Negatives)

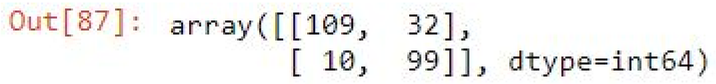
Wordcloud:

Word Cloud, also known as Tag Cloud, is a representation of text data in a visual way. A word cloud displays words, with the importance of each word being shown via the size of the word. A word cloud helps in quickly identifying the most important and prominent words at a glance. We have displayed the word clouds for our dataset. As we can see from the word cloud, ‘Christmas’ is the most prominent word in the dataset, followed by RIT, year and others.



Confusion Matrix

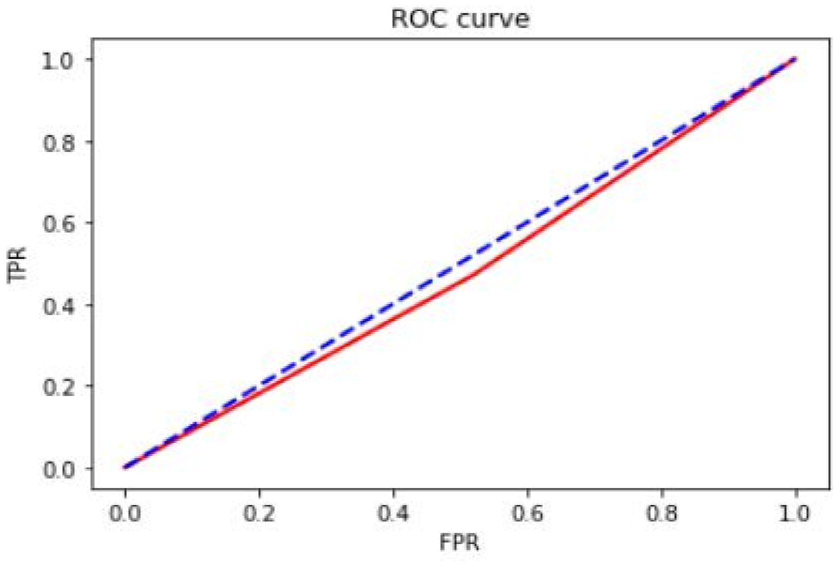
A confusion matrix is a table that is used in machine learning to describe the performance of a classifier model. We usually calculate it on data for which the true values are known. It allows to clear confusion between classes, e.g : identifying mislabeled classes. A confusion matrix is basically the number of correct and incorrect results summarized by count values and distributed by class.



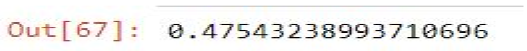
```
Out[87]:  array([[109,  32],
                 [ 10,  99]], dtype=int64)
```

ROC/AUC Curve

ROC (Receiver Operating Characteristics) curve and AUC (Area Under The Curve) are an important evaluation metric for judging the performance of a classifier ROC is a curve that represents probability whereas AUC represents degree of separability. So generally, higher the AUC, better is the model at distinguishing between models and thus the better is the model.



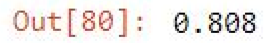
ROC-AUC score



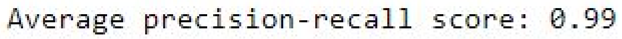
```
Out[67]:  0.47543238993710696
```

2. RANDOM FOREST CLASSIFIER

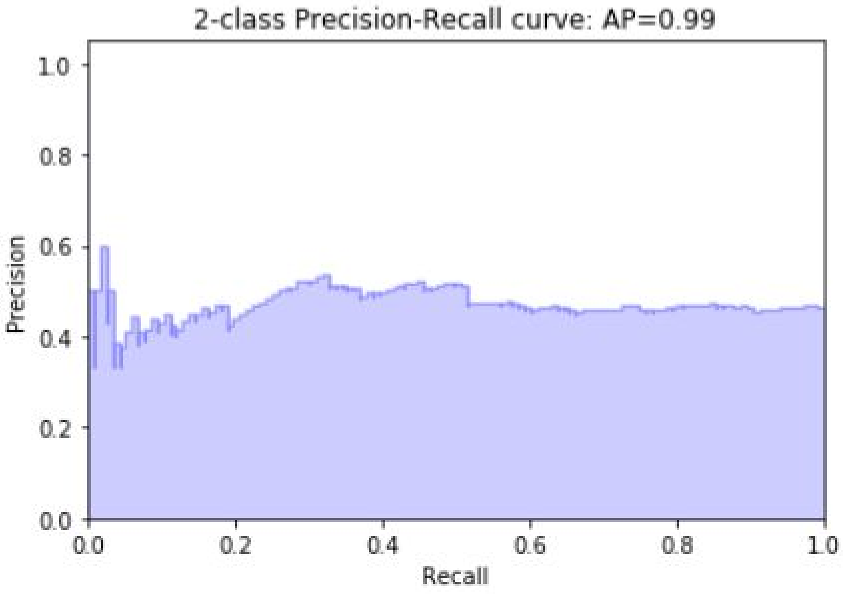
Accuracy



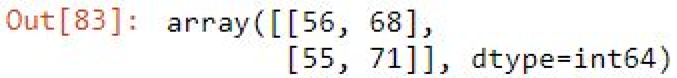
```
Out[80]:  0.808
```



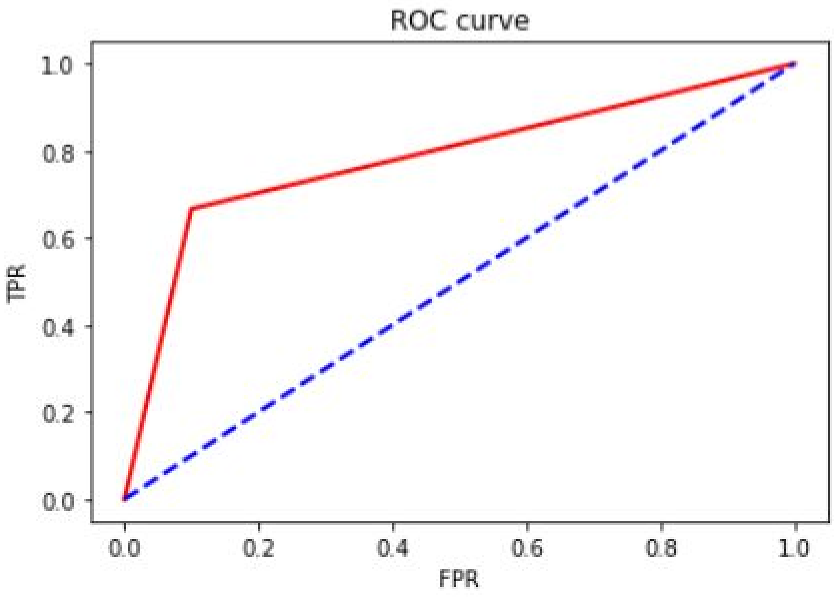
```
Average precision-recall score: 0.99
```

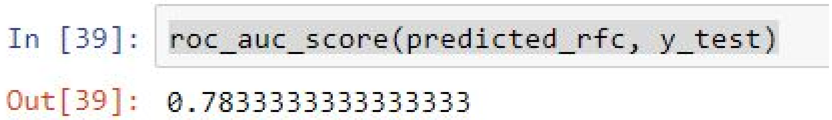


2-class Precision-Recall curve: AP=0.99



```
Out[83]: array([[56, 68],
                [55, 71]], dtype=int64)
```

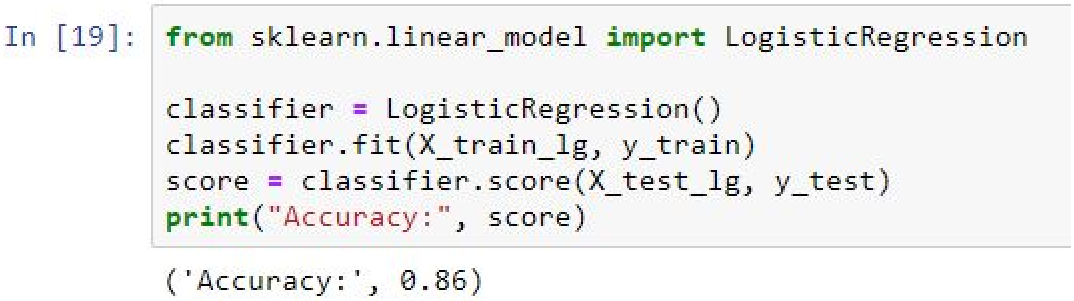




```
In [39]: roc_auc_score(predicted_rfc, y_test)

Out[39]: 0.7833333333333333
```

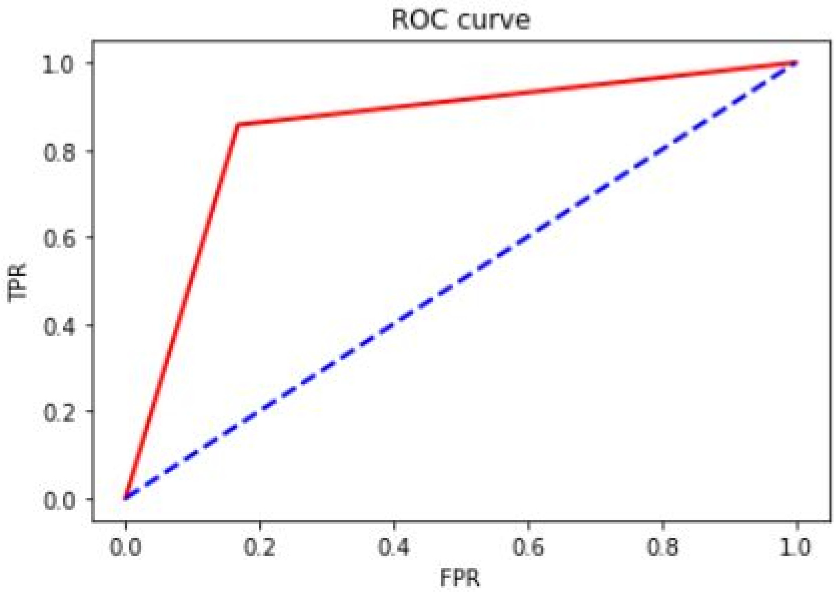
Logistic Regression:

Accuracy

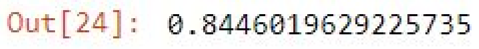


```
In [19]: from sklearn.linear_model import LogisticRegression

         classifier = LogisticRegression()
         classifier.fit(X_train_lg, y_train)
         score = classifier.score(X_test_lg, y_test)
         print("Accuracy:", score)

         ('Accuracy:', 0.86)
```

ROC CURVE



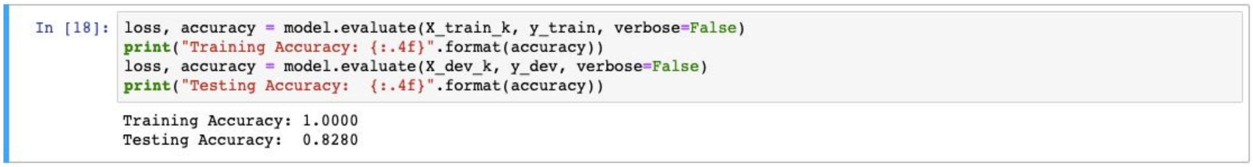
ROC-AUC score



```
Out[24]:  0.8446019629225735
```

LSTM

Accuracy



```
In [18]: loss, accuracy = model.evaluate(X_train_k, y_train, verbose=False)
         print("Training Accuracy: {:.4f}".format(accuracy))
         loss, accuracy = model.evaluate(X_dev_k, y_dev, verbose=False)
         print("Testing Accuracy:  {:.4f}".format(accuracy))

         Training Accuracy: 1.0000
         Testing Accuracy:  0.8280
```

COMPARISON OF THE ALGORITHMS:-

Algorithm	Accuracy	Precision-Recall	ROC-AUC score
Support Vector Machine	82.8	0.93	0.475
Random Forest Classifier	80.8	0.99	0.7833
Logistic Regression	86		0.8446
LSTM	82.8		

INSIGHTS FROM RESULTS:-

We were hoping that LSTM would be our best performing algorithm but as we can see from the results that is not the case. The algorithm didn't work as expected. We think that there may have been an issue with the way in which we provided the input. LSTM requires a 3d array and our input was a 2d array and we think there may have been an error during this conversion. One way to improve the algorithm would be make sure the input is correct and in proper format. Another point of improvement could have been removing the stop words and stemming the input data. Also running it on different dataset might give us a better idea of its performance. Also, we were able to verify through our algorithms the fact that the more different the dataset, the higher the accuracy of the classifier. Overall , we learnt the working and implementation of the various machine algorithm that be used to developer a pretty accurate text classifier.

CONCLUSION

So according to our observations and analysis we found that the Logistic Regression predicted the output for test set with highest accuracy among all the 4 algorithms which we have implemented. Whereas, the random forest gave the minimum accuracy. From the insights we found that all algorithms can classify the unlabelled posts to the correct subreddit with a relatively high accuracy unless the two subreddits are very similar to each other.