FINAL PROJECT

Data Mining: 5243

Rajarshi Biswas, Sayam Ganguly

The Ohio State University

Table of Contents

1. Problem Statement

1.1 Motivation

With the ubiquity of internet, the modern era of social media inundates us every day with a plethora of information. The onus on us to categorize thousands of Tweets, Facebook feeds etc., to consume what relevant to us. Imagine, you are a tech loving person and you are only interested in the tech related Tweets, among hundreds and hundreds of them, from BBC. what if you had to do the same for multiple Twitter handles, Facebook accounts? Filtering the relevant information by yourself is daunting!!

1.2 Our Goal

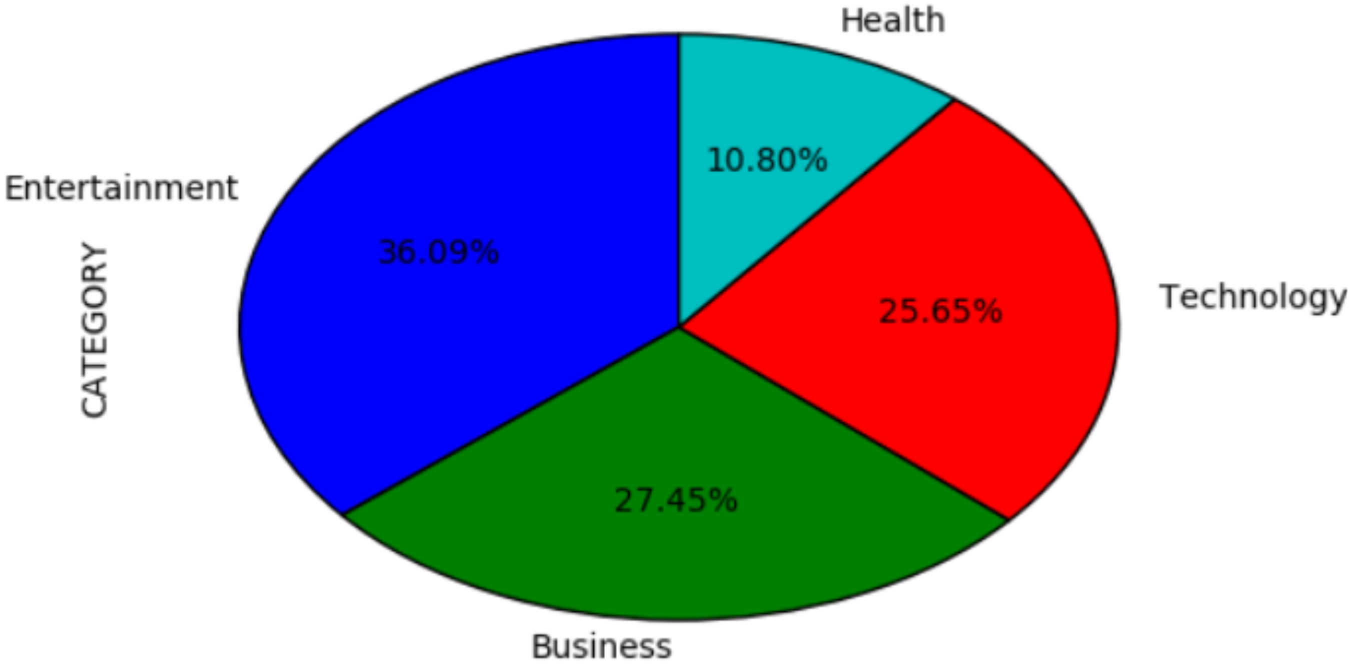
To make our online life easy, we propose to build an automated text stream category classification. Our model will analyze text streams and will classify to its relevant category. As categorizing all types of texts will be outside the scope of the class project, we narrowed it down to only **categorizing the news streams**. With the help of our training data we plan to train our model that will identify and categorize News. To test on live data, we used Twitter streams and evaluate how our model performs. Furthermore, we develop our model in a modular way so that it can be extended to take any other news streams and classify it into categories.

1.3 Brief overview of the test data

According to our research we found, the most relevant data that we could found is **News Aggregator Dataset from UCI Machine Learning Repository.** The dataset has 422937 news pages classified into the 4 different categories. We discuss more about the dataset, exploratory analysis, and preprocessing in the next section. The dataset is public and can be accessed from **https://archive.ics.uci.edu/ml/datasets/News+Aggregator.**

2. The Training Dataset

To train our model, we have used News Aggregator Dataset from UCI Machine Learning Repository. In this dataset news are grouped into clusters that represent pages discussing the same news story. The dataset includes also references to web pages that, at the access time, pointed (has a link to) one of the news page in the collection. Total number of news entries in the dataset is 422419. Furthermore, these different entries are classified in four categories – Business, Health, technology and Entertainment. The below pie chart depicts the distribution of these four categories in the news dataset.



The dataset has the following attributes:

ID: Numeric ID
TITLE: News title.
URL: URL of the news article.
PUBLISHER: Publisher name
CATEGORY: News category (b = business, t = science and technology, e = entertainment, m = health)
STORY: Alphanumeric ID of the cluster that includes news about the same story
HOSTNAME: URL hostname
TIMESTAMP: Approximate time the news was published, as the number of milliseconds since the epoch 00:00:00 GMT, January 1, 1970

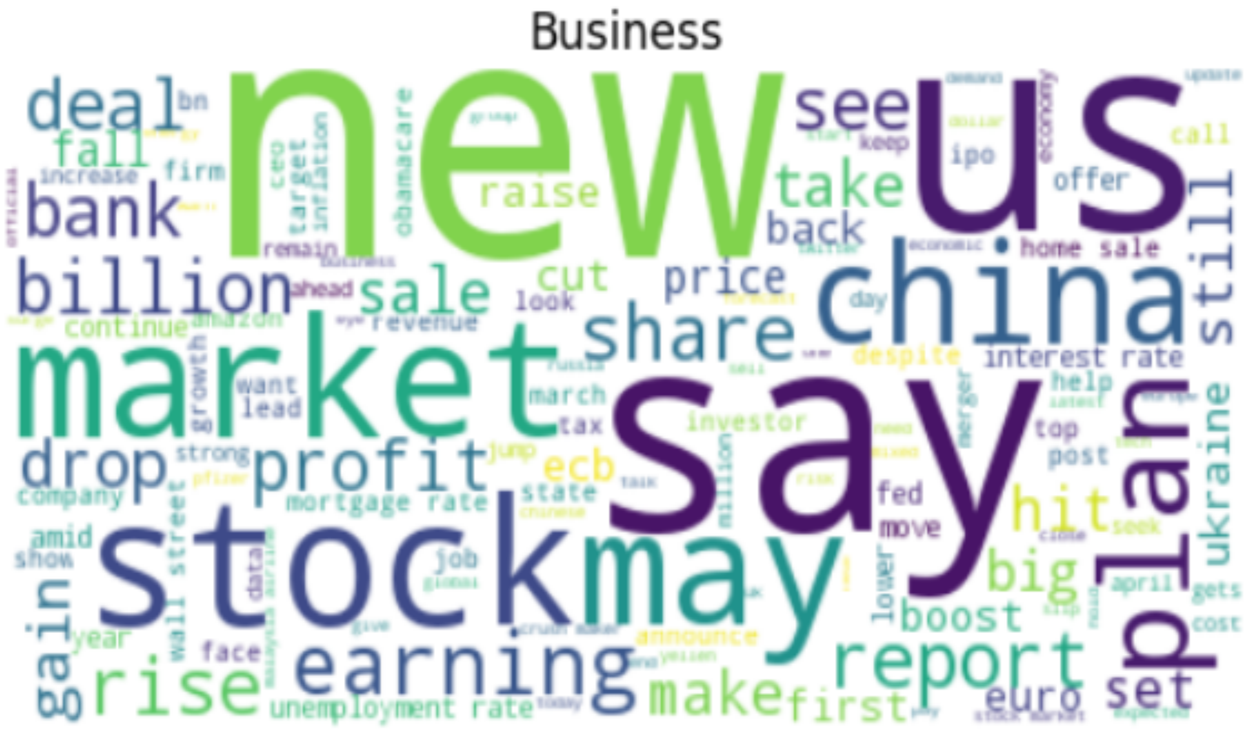
As we wanted to test our news categorization model on Twitter streams, we first try to see if this dataset fits our objective. As per our research we have found that an average Tweet length is 67.9 characters and the median is 60. This was published by a Twitter Employee (Isaac Hepworth) by analyzing 1 million Tweets.

The relevant graph can be found here - https://www.flickr.com/photos/hepwori/6732189421/in/photostream

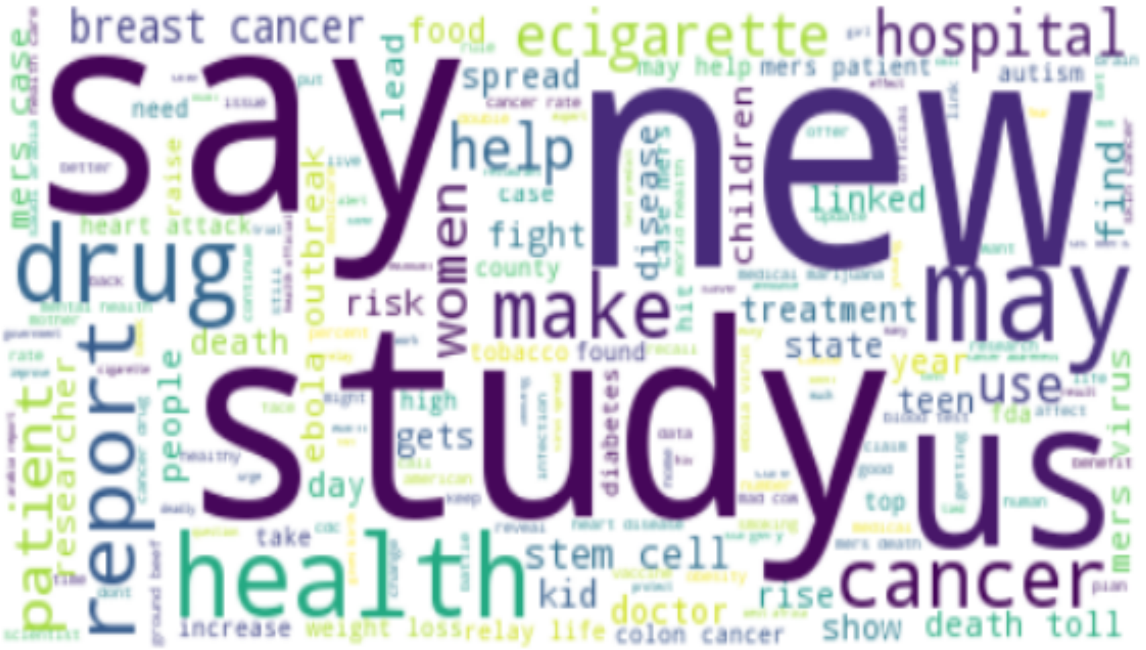
In our dataset, the most relevant attribute that has a similarity with Tweets is the news title. The news TITLE attribute has an average length of 58 characters and the median is 56. Also, our research shows that the news tweets contents are like the news headlines' contents in our dataset, thus we choose it (and the category) to classify our model. Later we show that our prediction is correct and our model predicts Tweets with acceptable accuracy.

2.2 Exploratory Analysis of the Training Dataset

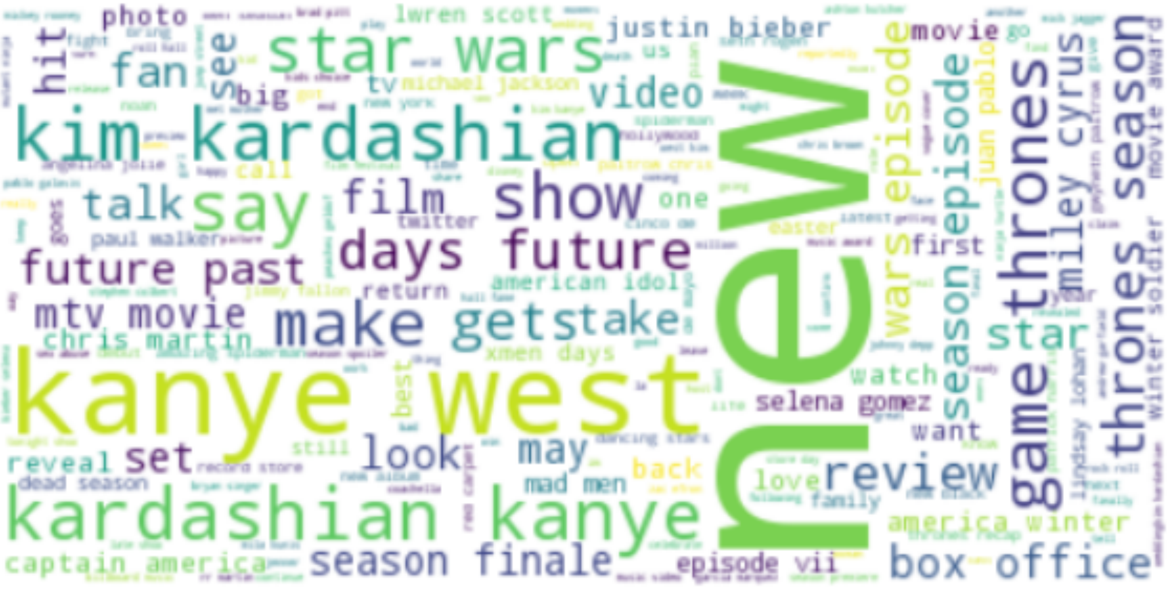
Given the different news categories, we first decide to capture the common words for all the news categories Individually. We did this by plotting word clouds for the news titles, grouped by different news categories. To do this we preprocessed the news title's texts to remove all the common stop words, punctuations, digits and tokenized the words to plot them. The below figures show the word clouds for different news categories –

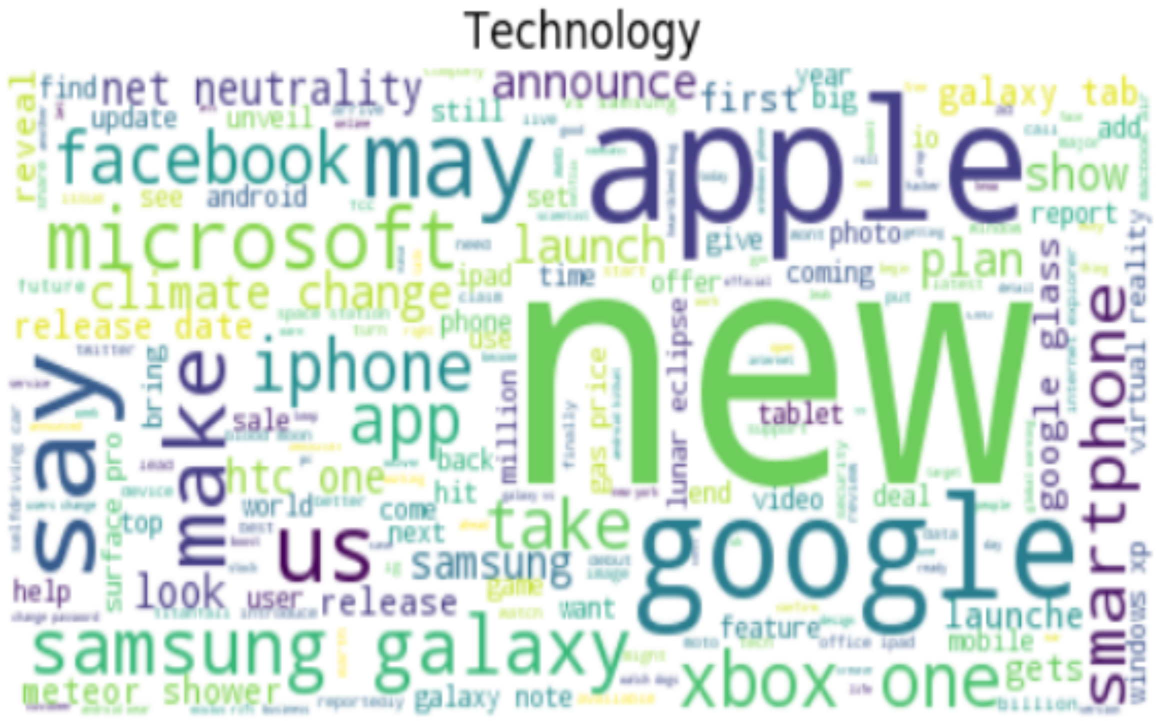


Health



Entertainment

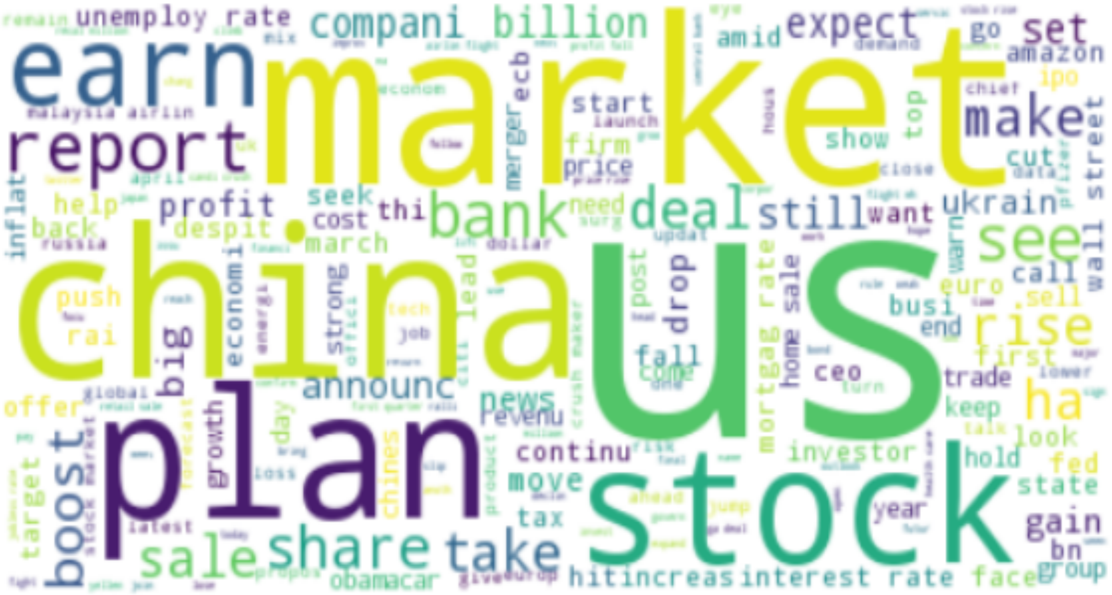




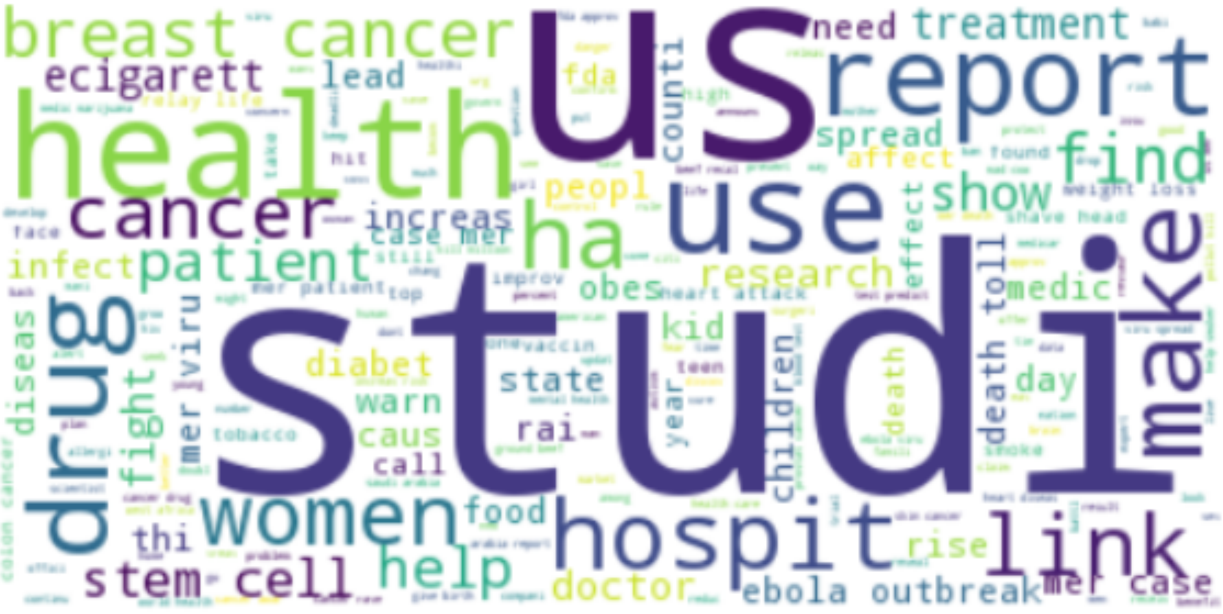
2.2 Preprocessing of the data

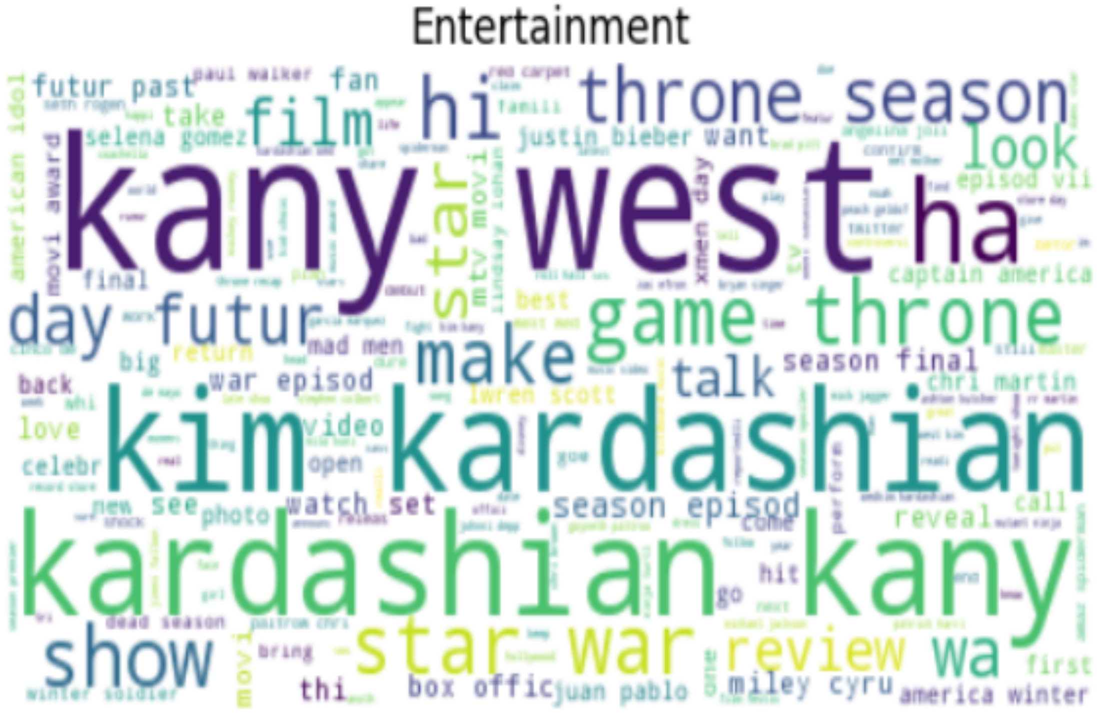
- As we have discussed earlier that in the news dataset only the news title and the category is relevant to classify the news streams from Twitter, we decided to remove the remaining attributes from the dataset.
- Then we remove the punctuation and digits from the title column. Afterwards, we tokenize the words and remove the stop words commonly occurring in the English language.
- As the word clouds from the figures in the above section indicates that in addition to the commonly occurring stop words in English, we observe the high frequency of say, may, new in all news categories. These words may not add any value to train our model. Hence, we have decided to remove these words. Furthermore, we applied stemming on the corpus. Following are the word clouds for different news categories generated after applying these preprocessing:

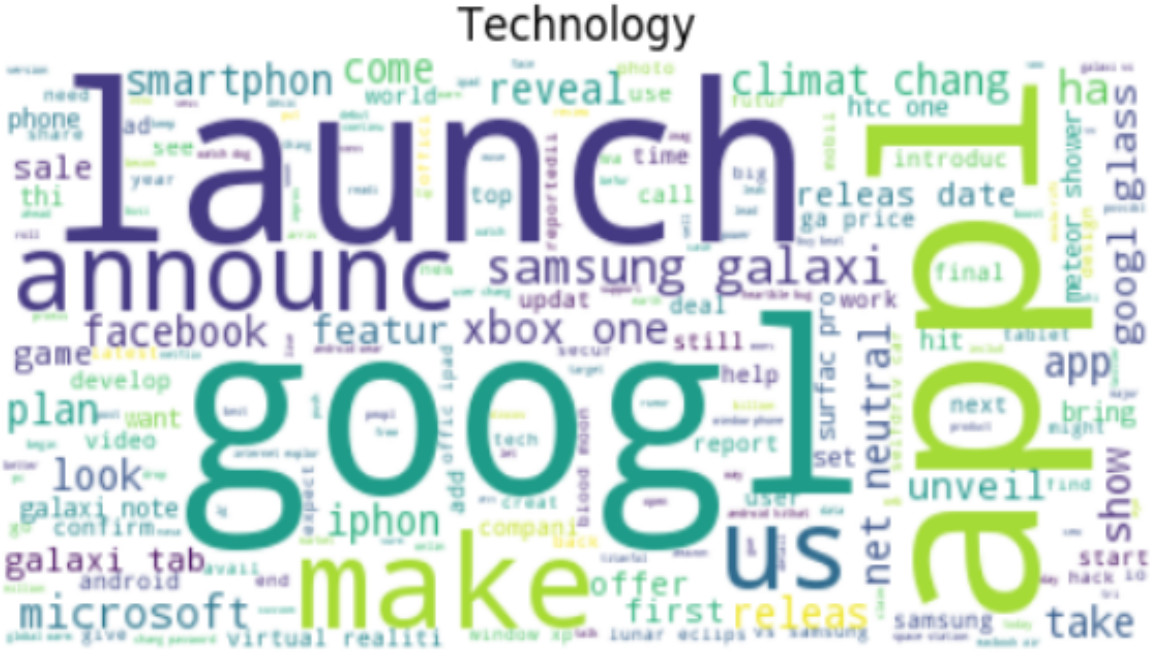
Business



Health




Entertainment
kany west
kim kardashian
kardashian kany
ha
hi
throne season
game throne
film
star
futur
day
make
talk
look
show
star war
review
wa
season episod
season final
war episod
captain america
episod vii
futur past
justin bieber want
selena gomez
mtv movi
movi award
american idol
lindsay lohan
xmen day
mad men
chri martin
lwren scott
video
watch set
reveal
photo
box offic
juan pablo
miley cyru
america winter
winter soldier
dead season
love
celebr
back
big
return
best
final
fan
take
paul walker
hit
come
open
new
see
thi
movi
first
call
perform
go
bring


Technology
launch
appl
googl
announc
make
us
smartphon
come
reveal
climat chang
ha
htc one
samsung galaxi
releas date
ga price
googl glass
meteor shower
facebook
featur
xbox one
net neutral
surfac pro
app
unveil
iphon
galaxi tab
galaxi note
microsoft
virtual realiti
releas
offer
first
samsung
take
show
plan
look
game
develop
sale
phone
world
introduc
final
work
hit
tablet
bring
next
set
report
tech
updat
deal
still
help
user
compani
confirm
android
video
want
lunar eclips
window xp
blood moon
start
hack

- Finally, we split the entire data set into separate training and test data. To achieve this, we split the news data from each category into 70:30 ratio (where 70% data goes to train the model, and the rest 30% kept testing the model later), and then combine them again together.

3 Program Description

The entire source code (spread across different files) is broken down in the following parts

3.1 Pre-processing

The Preprocessing.py, preprocesses the data and segregate the test and train dataset in a manner discussed in later sections.

3.2 Model (Feature extraction and Classification)

We implemented a total of four text feature extraction techniques, and for each of these, we experimented with five different classification algorithms.

Feature Extraction

- Bag of words
- BiGram
- TriGram
- TFIDF

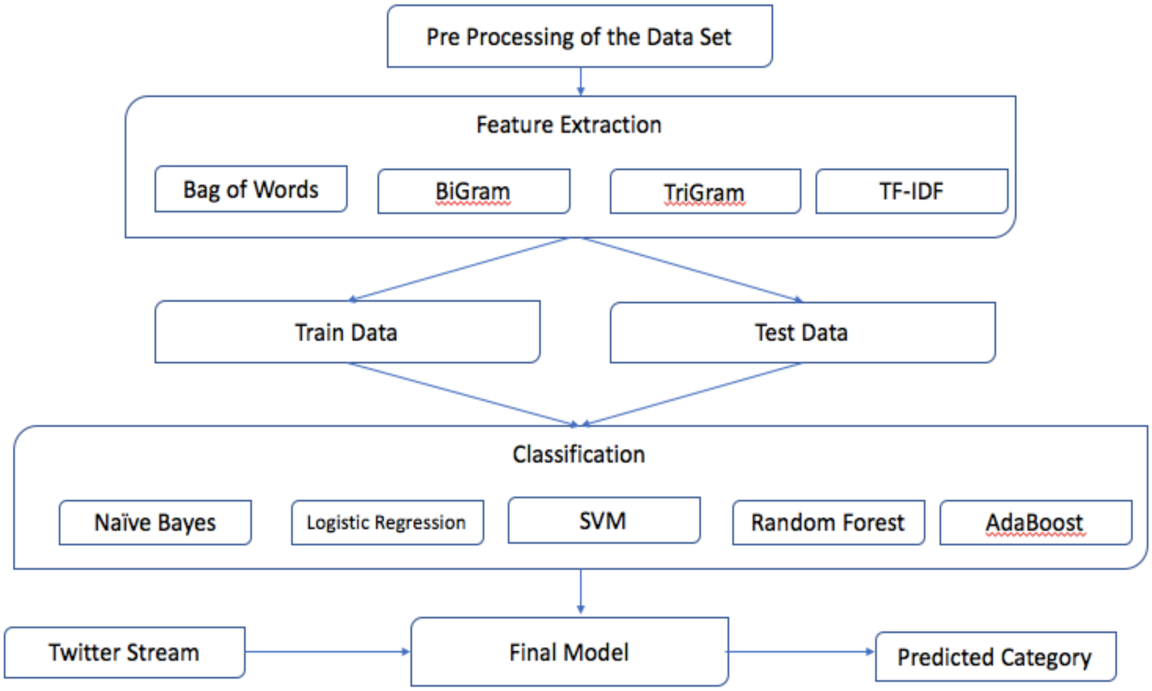
Classification Algorithms

- Multinomial Naïve Bayes
- Logistic Regression
- Linear support Vector Machine
- Random Forest
- AdaBoost

3.3 Twitter News prediction

Finally, we use our trained model to classify twitter streams. In addition, we generate a csv file with the category of each tweets.

we used pickle to generate intermediate datasets. Below is the flowchart of programming flow:



4. Model Description

To develop our model, we have experimented with different feature extraction techniques, and for each of those we have experimented with different classification algorithms. Finally, we choose the feature extraction method and classification algorithm that gives the best performance to classify twitter stream.

4.1 Feature Extraction

Following are the brief description of different feature extraction technique that we have experimented with.

4.1.1 Bag of Words Model

In this model, each of the news title's text is represented as term frequency vectors, disregarding grammar and even word order but keeping multiplicity, where each dimension of the vector represents the count of a word in the text. This model is also known as unigram model. The term frequency vector of the entire training dataset become the features to the classification model.

4.1.2 BiGram Model

A bigram is a sequence of two adjacent elements from string of tokens. Using this approach, the news title's text is broken down into bigrams. In other words, this model, represents the document as term frequency vectors of set of two words occurring consecutively in the text. This term frequency vectors become the features of the various classification algorithms.

4.1.3 TriGram Model

In the Trigram model, the news title text is represented by trigrams. That is set of three words occurring consecutively in the text. The term frequency vectors of these sets become the features of the news title text, where each dimension represents the count of the trigram in the text.

4.1.4 TFIDF Model

Term frequency-inverse document frequency, or TFIDF, is a numerical statistic that is intended to reflect how important a word is to a document in a collection of corpus. The TF-IDF increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, This TFIDF vector representations of the review corpus become the features to the classification model.

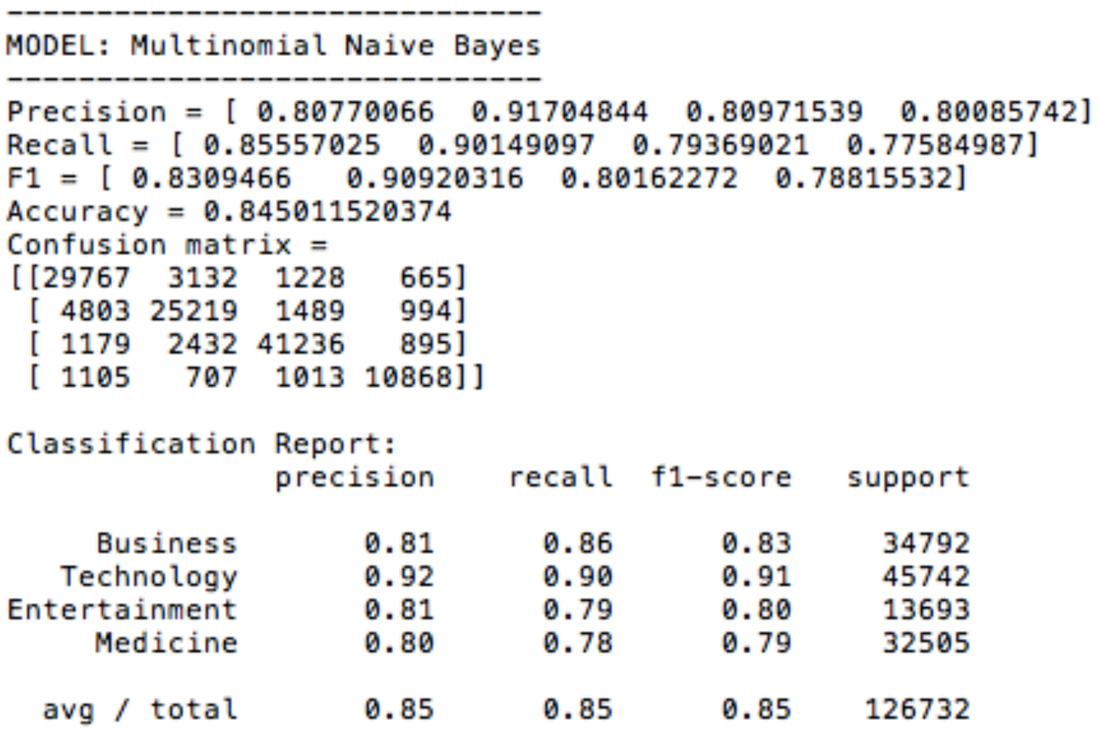
5. Model Evaluation

Below table compares the accuracy of the different models.

	Multinomial Naïve Bayes	Logistic Regression	Linear SVM	Random Forest	Adaboost
Bag Of Words	0.845011520374	0.836639522772	0.818514660859	0.796862670833	0.570905532936
BiGram	0.769718776631	0.700833254427	0.710459868068	0.69056787388	0.399078370104
TriGram	0.531365401004	0.524169112773	0.526504750181	0.51643542678	0.366190070385
TF-IDF	0.840356026891	0.844853707035	0.832915127987	0.808943281886	0.580184957233

5.1 Model Evaluation and Final Model Selection for Tweet categorization

The above model clearly shows that the best performed model is Multinomial Naïve Bayes with Bag of words approach. Also, the TF-DIF feature extraction method worked comparatively well, especially for Logistic Regression. The better performance of Bag of Words and TF-IDF over other approaches is expected. As the news titles or headlines has an average of 59 characters, thus less number of words, counting the term frequency of two or three consecutive words will not be effective. For this same reason, we can see that the trigram model performed worse than the bigram. Thus, we choose Bags of Words with Multinomial Naïve Bayes, and TF-IDF with Logistic Regression as our final Model in attempt for the Tweet categorization. We show below different statistics for these two selected models.



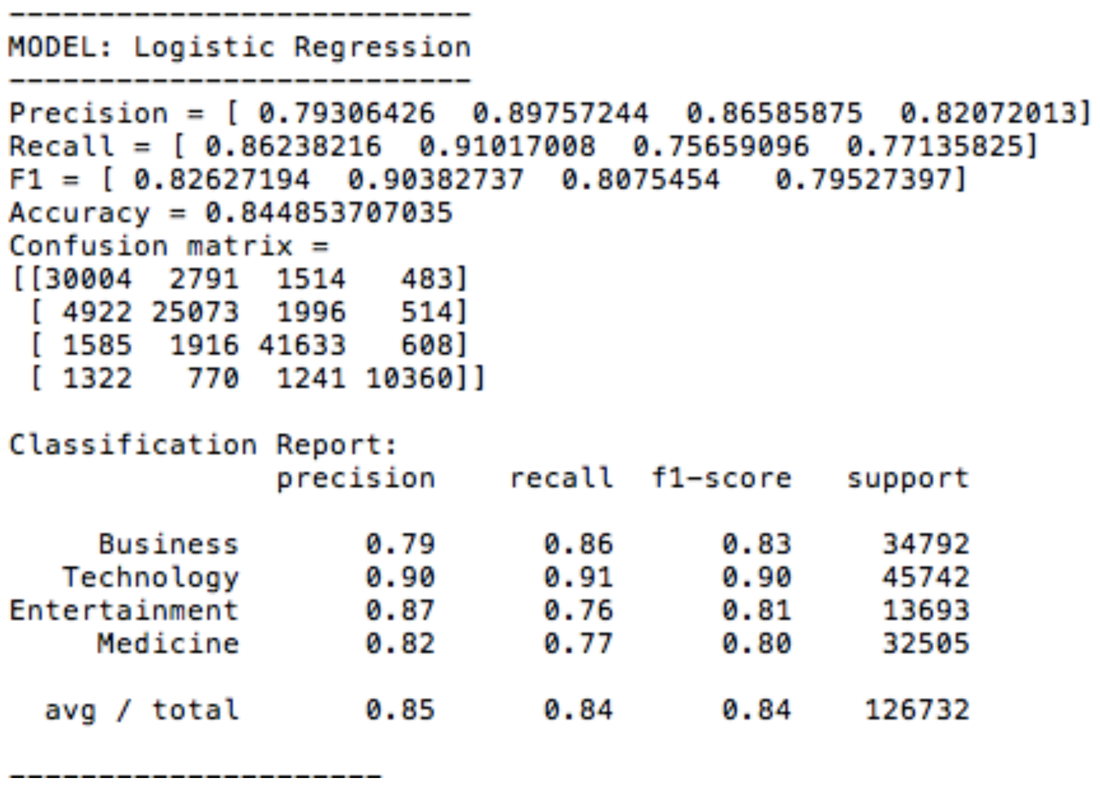
```
------------------------------
MODEL: Multinomial Naive Bayes
------------------------------
Precision = [ 0.80770066  0.91704844  0.80971539  0.80085742]
Recall = [ 0.85557025  0.90149097  0.79369021  0.77584987]
F1 = [ 0.8309466   0.90920316  0.80162272  0.78815532]
Accuracy = 0.845011520374
Confusion matrix =
[[29767  3132  1228   665]
 [ 4803 25219  1489   994]
 [ 1179  2432 41236   895]
 [ 1105   707  1013 10868]]

Classification Report:
               precision    recall  f1-score   support

     Business       0.81      0.86      0.83     34792
   Technology       0.92      0.90      0.91     45742
Entertainment       0.81      0.79      0.80     13693
     Medicine       0.80      0.78      0.79     32505

  avg / total       0.85      0.85      0.85    126732
```

Bags of Words with Multinomial Naïve Bayes



```
--------------------------
MODEL: Logistic Regression
--------------------------
Precision = [ 0.79306426  0.89757244  0.86585875  0.82072013]
Recall = [ 0.86238216  0.91017008  0.75659096  0.77135825]
F1 = [ 0.82627194  0.90382737  0.8075454   0.79527397]
Accuracy = 0.844853707035
Confusion matrix =
[[30004  2791  1514   483]
 [ 4922 25073  1996   514]
 [ 1585  1916 41633   608]
 [ 1322   770  1241 10360]]

Classification Report:
               precision    recall  f1-score   support

     Business       0.79      0.86      0.83     34792
   Technology       0.90      0.91      0.90     45742
Entertainment       0.87      0.76      0.81     13693
     Medicine       0.82      0.77      0.80     32505

  avg / total       0.85      0.84      0.84    126732

---------------------
```

TF-IDF with Logistic Regression

Our program generates the statistics for all the models. We write all the details in specific files (more details about the files can be found on the README). Please look at those for further details of the performance statistics of the other models.

6. Applying our model on Twitter stream

Next, we apply our trained model on the twitter streams. We get the twitter live streams using Tweepy APIs. For every tweet, our model classifies a category among business, health, entertainment and technology. Below we discuss the preprocessing of the Tweets and the prediction performance of our model.

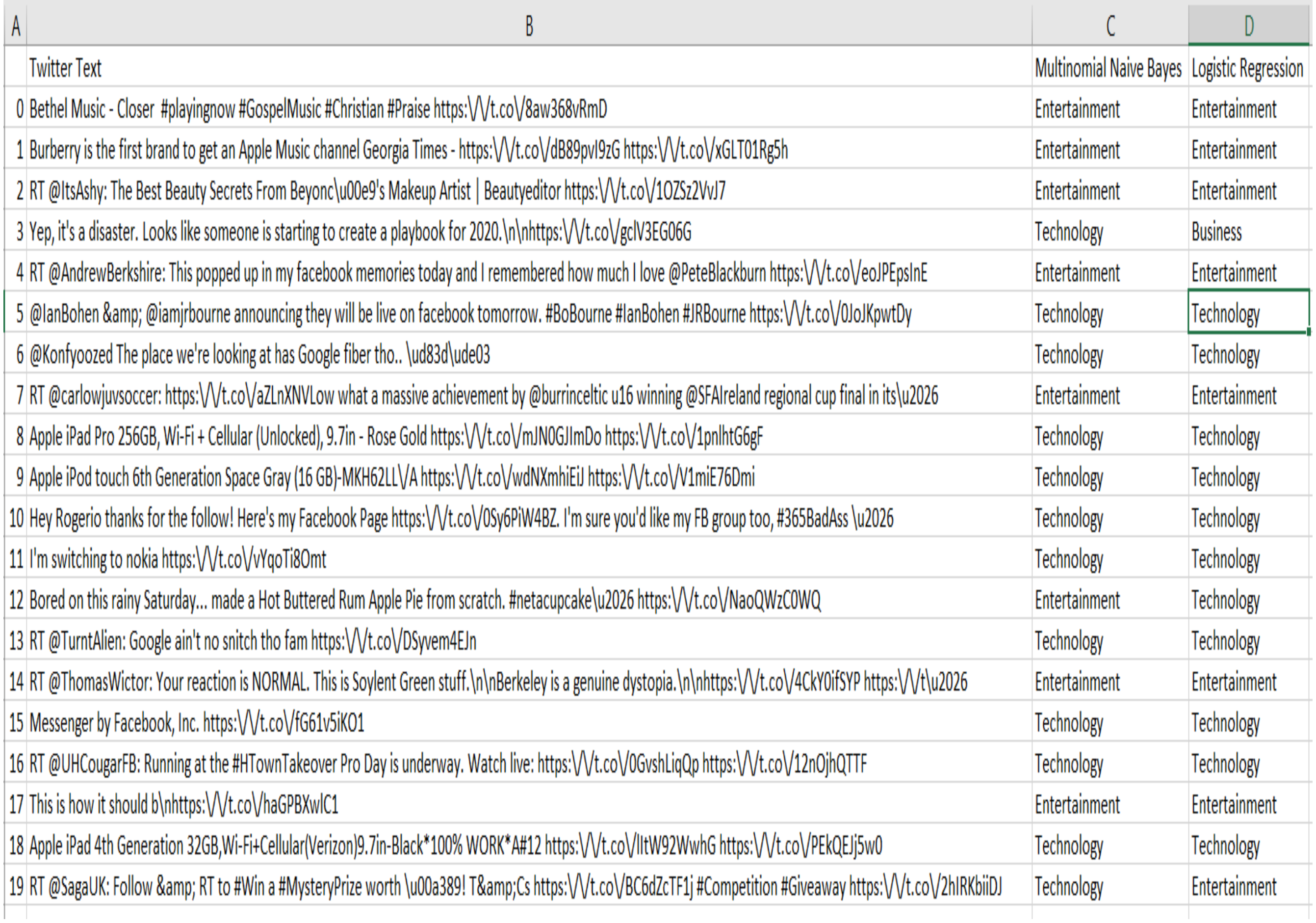
6.1 Preprocessing Twitter data

- We preprocessed each of the tweet's texts to remove all the common stop words, punctuations, digits and tokenized the words to plot them. In addition, we applied stemming on the Twitter data.

- Also, we have removed the text “rt” (short for retweets) from the tweets.
- We have also taken only English tweets.

6.2 Prediction Performance of our model

We have experimented with different genres of tweets to classify them. Following table shows the predicted category of 10 such Tweets. These tweets were taken by filtering different keywords. We can observe that our model performed almost accurately on predicting the categories. The miss predictions are acceptable given the nature of the limited category of the training dataset and the nature of tweets.



A	B	C	D
	Twitter Text	Multinomial Naive Bayes	Logistic Regression
0	Bethel Music - Closer #playingnow #GospelMusic #Christian #Praise https:\/\/t.co\/8aw368vRmD	Entertainment	Entertainment
1	Burberry is the first brand to get an Apple Music channel Georgia Times - https:\/\/t.co\/dB89pvI9zG https:\/\/t.co\/xGLT01Rg5h	Entertainment	Entertainment
2	RT @ItsAshy: The Best Beauty Secrets From Beyonc\u00e9's Makeup Artist | Beautyeditor https:\/\/t.co\/1OZSz2VvJ7	Entertainment	Entertainment
3	Yep, it's a disaster. Looks like someone is starting to create a playbook for 2020.\n\nhttps:\/\/t.co\/gclV3EG06G	Technology	Business
4	RT @AndrewBerkshire: This popped up in my facebook memories today and I remembered how much I love @PeteBlackburn https:\/\/t.co\/eoJPEpsInE	Entertainment	Entertainment
5	@IanBohen &amp; @iamjrbourne announcing they will be live on facebook tomorrow. #BoBourne #IanBohen #JRBourne https:\/\/t.co\/0JoJKpwtDy	Technology	Technology
6	@Konfyoozed The place we're looking at has Google fiber tho.. \ud83d\ude03	Technology	Technology
7	RT @carlowjuvsoccer: https:\/\/t.co\/aZLnXNVLow what a massive achievement by @burrinceltic u16 winning @SFAIreland regional cup final in its\u2026	Entertainment	Entertainment
8	Apple iPad Pro 256GB, Wi-Fi + Cellular (Unlocked), 9.7in - Rose Gold https:\/\/t.co\/mJN0GJImDo https:\/\/t.co\/1pnlhtG6gF	Technology	Technology
9	Apple iPod touch 6th Generation Space Gray (16 GB)-MKH62LL\/A https:\/\/t.co\/wdNXmhiEiJ https:\/\/t.co\/V1miE76Dmi	Technology	Technology
10	Hey Rogerio thanks for the follow! Here's my Facebook Page https:\/\/t.co\/0Sy6PiW4BZ. I'm sure you'd like my FB group too, #365BadAss \u2026	Technology	Technology
11	I'm switching to nokia https:\/\/t.co\/vYqoTi8Omt	Technology	Technology
12	Bored on this rainy Saturday... made a Hot Buttered Rum Apple Pie from scratch. #netacupcake\u2026 https:\/\/t.co\/NaoQWzC0WQ	Entertainment	Technology
13	RT @TurntAlien: Google ain't no snitch tho fam https:\/\/t.co\/DSyvem4EJn	Technology	Technology
14	RT @ThomasWictor: Your reaction is NORMAL. This is Soylent Green stuff.\n\nBerkeley is a genuine dystopia.\n\nhttps:\/\/t.co\/4CkY0ifSYP https:\/\/t\u2026	Entertainment	Entertainment
15	Messenger by Facebook, Inc. https:\/\/t.co\/fG61v5iKO1	Technology	Technology
16	RT @UHCougarFB: Running at the #HTownTakeover Pro Day is underway. Watch live: https:\/\/t.co\/0GvshLiqQp https:\/\/t.co\/12nOjhQTTF	Technology	Technology
17	This is how it should b\nhttps:\/\/t.co\/haGPBXwlC1	Entertainment	Entertainment
18	Apple iPad 4th Generation 32GB,Wi-Fi+Cellular(Verizon)9.7in-Black*100% WORK*A#12 https:\/\/t.co\/lItW92WwhG https:\/\/t.co\/PEkQEJj5w0	Technology	Technology
19	RT @SagaUK: Follow &amp; RT to #Win a #MysteryPrize worth \u00a389! T&amp;Cs https:\/\/t.co\/BC6dZcTF1j #Competition #Giveaway https:\/\/t.co\/2hIRKbiiDJ	Technology	Entertainment

7. Coding Contribution

7.1 Feature Removal

The dataset consists of 422419 records with 8 features for each record. As our goal is to classify news genre we are only using the "TITLE" feature and "CATEGORY" as the labels. Hence, we eliminate rest of the features from our dataset.

7.2 Feature Transformation

The dataset contains real life news title and hence it will contain punctuations, digits and similar words which will hinder classifier's performance. We first converted the entire corpus to lower case and then removed all the digits and punctuations from the corpus. Then we tokenized the corpus using NLTK library functions. We also removed stopwords and applied stemming to our corpus. In addition, we also removed some other words which were not part of NLTK Stopwords.

7.3 Separating Training and Testing Set

We used a 70-30 split for training and testing set. To ensure equal representation from each of the categories we divided the corpus as per the four categories to create 4 different subsets. Then we applied 70-30 split in each of the subsets. After this we combined all the training subsets to generate the full training corpus. Similarly, we also generate testing corpus. We then used pickle to save these corpuses to be later used for EDA and classification.

7.4 Generating output files

During training of each of the model we generated an output file with all the relevant metrics from each of the classifiers which helped us later in identifying our best model across all the transformation.

7.5 Twitter classification with best models

We analyzed our results and identified the models that are performing the best in news genre classification. Then we used pickle to save the corresponding vectorizer and trained model to be used for twitter classification later. We used Tweepy to access live tweet streams. Once we got the tweets we applied similar transformations as explained in section 7.2 to our twitter data. Then we loaded the pretrained models and vectorizers to predict the genre of the tweet. Finally, we saved the result of classifying 50 tweets in a csv.

8. Contribution

All the team members have equally contributed in the project. Below are the key contributions of the team members.

Rajarshi Biswas

- Performed preprocessing of the dataset.
- Implemented Bag of words and Bigram with different classification models.
- Suggested and implemented the idea of using the model to predict Twitter Stream.

Sayam Ganguly

- Performed preprocessing and EDA of the dataset.
- Suggested the idea using pickle to optimize the performance of the program and save the model for later use.
- Implemented TriGram and TF-IDF with different classification models.

9. Tools Used

We used the following tools and packages for this assignment.

- Python
- Pickle
- Sklearn
- Pandas
- numpy
- nltk
- Tweepy
- Twitter
- Jupyter
- Spyder