Machine learning for text categorization: Experiments using clustering and classification

by

Poojitha Bikki

B.Tech., Shiv Nadar University, India, 2016

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2018

Approved by:

Major Professor
Dr. William Hsu

Copyright

Abstract

This work describes a comparative study of empirical methods for categorization of new articles within text corpora: unsupervised learning for an unlabeled corpus of text documents and supervised learning for hand-labeled corpus. The goal of text categorization is to organize natural language (i.e. human language) documents into categories that are either predefined or that are inherently grouped by similar meaning. The first approach, automatic classification of texts, can be handy when handling massive amounts of data and has many applications such as automated indexing of scientific articles, spam filtering, classification of news articles etc. Classification using supervised or semi-supervised inductive learning involves labeled data, which can be expensive to acquire and may require semantically deep understanding of the meaning of texts. The second approach falls under the general rubric of document clustering, based on the statistical distribution and co-occurrence of words in a full-text document. Developing a full pipeline for document categorization draws on methods from information retrieval (IR), natural language processing (NLP), and machine learning (ML).

In this project, experiments are conducted on two text corpora: news aggregator data, which contains news headlines collected from a web aggregator and a news data set consisting of original news articles from the British Broadcasting Corporation (BBC). First, the training data is developed from these corpora. Next, common types of supervised classifiers, such as linear, Bayesian, ensemble models and support vector machines (SVM) are trained, on the labelled data and the trained classification models are used to predict the category of an article, given the related text. The results obtained are analyzed and compared to determine the best performing model. Then, two unsupervised learning techniques – *k*-means and Latent Dirichlet Allocation (LDA) are applied to obtain clusters of data points. *k*-means separates the documents into disjoint clusters of

similar news. Additionally, LDA was used, which treats documents as a mixture of topics, to find latent topics in text. Finally, visualizations of the results are produced for evaluation: to allow qualitative assessment of cluster separation in the case of unsupervised learning, or to understand the confusion matrix for the supervised classification task by heat map visualization as well as precision, recall, and other holistic metrics. From an application standpoint, the unsupervised techniques applied can be used to find news that are similar in content and can be categorized under a specific topic.

Table of Contents

List of Figures

List of Tables

Acknowledgements

This work would not have been possible without the guidance of Dr. William H. Hsu, my major professor. His belief in my abilities to finish this project and his constant support throughout my graduate studies have played a significant role in what I have achieved over these years at K-State. I am also grateful to Dr. Daniel Andersen and Dr. Torben Amtoft for serving on my graduate committee and helping me finish this project successfully.

I am especially indebted to Jason Bengtson and Julie Bell from Administrative and IT services at Hale Library who have been very supportive throughout this project phase. They have been super cool and amazing supervisors. It has been a wonderful experience working as a graduate assistant under their supervision.

I am also grateful to Dr. Roman Ganta, with whom I have had the pleasure to work during my first semester. I cannot thank him and his wife Suhasini enough for their support and guidance ever since my first day at K-State. I am also thankful to Dr. Lindsey Hulbert, from Animal Sciences Department for her valuable guidance and support on one of the projects that I have worked with her on.

Nobody has been more important to me in whatever I have done so far, than my parents and my younger brother Manoj. Unconditional love and inspiration from my Mom and Dad are always with me in whatever I pursue. They have always given their best, to make me whatever I am as a person today. My Dad's guidance, especially, has a leading role in my every step towards success. Thanks to them for everything.

Most importantly, I would like to thank my friends Nithin, Pavan, Pruthvi, Raja, Sharmila, Sindhu, Sneha, and Sravani for their constant love and support. They have been my second family and past two years would not have been so amazing without all of them. I hope this goes far.

Chapter 1 - Introduction

1.1 Problem Definition

Massive volumes of text are available to users online in the form of scientific articles, news, product reviews, social media content, etc. It is more practical for users to look for information by browsing through categories rather than searching the whole information space. Google Directory (Wikipedia, 2018) was one such web classification system that existed in 2011. Many such systems, however are handled by human experts and do not scale up well with the increased number of web pages available. Real-world applications of text categorization often require a system to deal with tens of thousands of categories defined over a large taxonomy. Since building these text classifiers by hand is time-consuming and costly, automated text categorization has gained importance over the years. Classification methods developed based on machine learning deal with text classification problem in a fantastic way. Text classification or categorization is assigning documents to a predefined category.

As we know, news is vital source of information that keeps a person informed about what's happening around the world. News on the web, is updated on a frequent basis and a lot of online news related apps are competitively providing users with lots of choices. There is hence a need for text categorization in this domain. This work aims to address the abovementioned need by using machine learning techniques.

1.2 Objective

I chose the domain of news in order to apply machine learning techniques that allow for automatic categorization. My aim is to classify news into various categories, such as – business,

technology, sports, entertainment and politics. For this purpose, supervised machine learning techniques are applied on news data sets. I also use unsupervised machine learning techniques to allow for text classification when labelled data is not present. The overall goal is to predict the category of news article, present results obtained on evaluation of different classification models. In addition, I also show how clustering can be used to group news articles that are similar in content and can be categorized under a topic given the content of the article.

1.3 Overview

In this project, I used news data sets (Dheeru & Karra Taniskidou, 2017) (Greene & Cunningham, 2006) available online to assign category label to new documents based on likelihood suggested by classifiers trained on the set of labelled documents. The data set contains news documents that fall into 5 distinct categories. The data is labelled by human experts from the domain and these values are used as ground truth to validate the results obtained. I also, trained my classifiers on new aggregator data set which contains about 400,000 news headlines that belong to 4 distinct categories. The results obtained using both data sets are shown and compared using various performance metrics. Unsupervised machine learning techniques like K-means and Latent Dirichlet Allocation (LDA) are used to cluster news which help with text categorization when no labelled data is present. Scikit-learn (Pedregosa, Varoquaux, Gramfort, & Michel, 2011), a machine learning package of python is used to implement the above-mentioned techniques. Bokeh (Team, 2014), an interactive visualization library is also used to show the cluster outputs (Besbes, 2017). Other python visualization libraries like *Matplotlib* (Hunter, Dale, Dorettboom, & Team, 2012) and *Seaborn* (Waskom, 2012-2017) were also used.

Chapter 2 - Background and Related Work

This chapter introduces text classification and clustering techniques used in the project. I also described text extraction methods that were used to prepare the data for modelling using machine learning.

2.1 Literature Survey

Recently, the role of text mining has become crucial due to the availability of the increasing number of electronic documents from a variety of sources which often include unstructured or semi-structured data. The main goal of text mining is to enable users to extract information from text. The task of automatically classifying and discover patterns from different type of documents, falls on the cross roads of Information Retrieval (IR), Natural Language Processing (NLP), Machine Learning (ML). Text Classification (TC) is thus an important part of text mining. Automatic TC systems can be built by means Knowledge-engineering techniques i.e. defining a set of logical rules that convert expert knowledge on how to classify documents under the given set of categories (Korde, 2012).

One such technique would be to automatically label each incoming news story with a topic like “sports”, “politics”, or “art”. This classification task starts with a training set $D = (d_1, d_{2, \ldots,} d_n)$ of documents that are already labelled with a classes C_1, C_2 ,.. (e.g., sport, politics). Then, a classification model which is able to assign the correct class ‘C_j’ to a new document ‘d_i’ (Korde, 2012). Text classification is of two types: single label and multi-label. A single label document belongs to only one class and a multi-label document may be belong to more than one class. In this project, we only consider single label document classification.

2.1.1 Text Feature Extraction

Raw data is a sequence of symbols and cannot be fed directly to the algorithms themselves as most of them expect numerical feature vectors with a fixed size, rather than the raw text documents with variable length. The most common steps to extract numerical features from text are as follows:

- **Tokenizing** strings and giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators.
- **Counting** the occurrences of tokens in each document.
- **Normalizing** and weighting with diminishing importance tokens that occur in most documents.

Bag of Words representation:

In this representation, each individual token occurrence frequency (normalized or not) is treated as a **feature**. The vector of all the token frequencies for a given document is considered a multivariate sample. A corpus of documents can thus be represented by a matrix with one row per document and one column per token (e.g. word) occurring in the corpus.

Vectorization the general process of turning a collection of text documents into numerical feature vectors. This specific strategy (tokenization, counting and normalization) is called the **Bag of Words** or "Bag of n-grams" representation. Using this representation, documents are described by word occurrences while completely ignoring the relative position information of the words in the document (Lars, et al., 2013).

In text classification, a text document may partially match many categories. We need to find the best matching category for the text document. The term frequency/inverse document frequency (TF-IDF) approach is commonly used to weight each word in the text document depending on how

unique the term is. In other words, the TF-IDF approach captures the relevancy among words, text documents and particular categories. It is further elaborated in the section below.

2.1.2 TF-IDF term weighting

In large text corpus, often words like 'the', 'an', 'is' called the stop words carry very little importance in terms of classifying them into a category. If the classifier uses the frequencies of these terms to train, the significance of most important terms is not recognized. So, we often use TF-IDF approach to reweight the count features (Lars, et al., 2013).

$$tf\_idf(t,d) = tf(t,d) * idf(t)$$

tf (t, d) is the frequency of term *t* in a document *d*

$$idf(t) = \log\frac{1+\mathrm{n}}{1+\mathrm{df(d,t)}} + 1$$

n is the total number of documents

$df(d,t)$ is the number of documents that contain term t.

The resulting TF-IDF vectors are then normalized by the Euclidean norm:

$$v_{norm} = \frac{v}{\sqrt{v_1^2 + v_2^2 + v_3^2 + .. + v_n^2}}$$

2.2 Established Methods

2.2.1 Classification Methods

Given a data set $D = \{x_i , y_i\}^{m}1$, where $x_i \in R_n$, $y_i \in \{1, 2, ..., k\}$, the task of multiclass classification is to learn a model that outputs a single class label '*y'* given an example '*x'*

2.2.1.1 Naïve Bayes Text Classifier

It is a simple probabilistic classifier based on applying Bayes' Theorem with strong independence assumptions. It treats text as bag of words and the occurrence of terms and their

positions are independent of the probabilities. Hence, underlying probability model is an independent feature model. Depending on the precise nature of the probability model, the naïve Bayes classifiers can be trained very efficiently by requiring a relatively small amount of training data to estimate the parameters necessary for classification.

(Khan, Baharudin, & Lee, 2010) The building model for Naïve Bayes would be as follows:

For each document text in class *ci*, build a probabilistic model $P(T: t_1, t_2, t_3, .. t_n | c_i)$ where

- T: text in class *ci*
- n: size of the vocabulary

Text is classified as follows:

The category c_i with the highest score among all categories C is the one that is most probable to generate the text d_j and is calculated as $c_{\max a\ posteriori} = \ arg_{c_i} \max\ p(c_i)P(d_j|c_i)$

$$P(c_i|d_j) = p(c_i)P(d_j|c_i)$$

$P(c_i|d_j)$ is posterior probability of class C_i

$p(c_i)$ is prior probability of class C_i

$P(d_j|c_i)$ is posterior probability of d

The multinomial Naïve Bayes model is as follows:

$$P\left(c_i\middle|d_j\right) = p(c_i)\ P\left(\overrightarrow{d_J}\ \middle|c_i\right)$$

where each $P\left(\overrightarrow{d_j}\ \middle|c_i\right)$ is calculated as

$$\prod_{k=1}^{|T|} P\left(t_{kj}\ \middle|\ c_i\right) = \sum_{i=1}^{|T|} \log P\left(t_{kj}\ \middle|\ c_i\right)$$

where each $P\left(t_{kj}\ \middle|\ c_i\right)$is calculated as

$$P(t_{kj} \mid ci) = \frac{\sum tf(t_k, j \in ci) + \alpha}{\sum Ndj \in ci + \alpha \cdot n}$$

(Raschka, 2014)

- t_k: A word from the feature vector *j* of a sample.
- $\sum tf(t_k, j \in ci)$: The sum of raw term frequencies of word t_k from all documents in the training sample that belong to class ci.
- $\sum N_{dj \in cj}$ The sum of all term frequencies in the training data set for class cj.
- α: An additive smoothing parameter (α =1 for Laplace smoothing).
- n: The size of the vocabulary (number of different words in the training set).

The term frequency is multiplied with inverse document frequency as explained in TF_IDF weighting section, since we are characterizing the documents with TF-IDF weighting.

2.1.1.2 Logistic Regression

Logistic Regression is a classification method suitable to multi class problem with more than two or more possible outcomes (Contributors, 2017). It is a model that can be used to predict the probabilities of different possible outcomes of a categorically distributed dependent variable given a set of independent variables. Here, it is assumed that we have a series of N observed data points. Each data point *'i'* has 'M' features $X_{1,i \ldots} X_{M,i}$. Each data point is associated with a categorical outcome Y_i (dependent variable) which takes one of the *1…K* class labels. Similar to the linear model, multiclass logistic regression uses linear prediction function *f (k, i)* to predict the probability of observation *'i'* having outcome '*k'*

$$f(k, i) = \beta_k \, . x_i$$

Where β_k is the set of regression coefficients associated with outcome *k*, and x_i (row vector) is the set of explanatory variables associated with variable associated with observation '*i'.* In

logistic regression with K classes, K-1 binary regression models are executed where once class is chosen as a reference. Then, the K-1 outcomes are separately regressed against the chosen reference and the resultant probabilities of categorical outcomes after mathematical calculation are as follows:

$$\Pr(Y_i = 1) = \frac{e^{\beta_1 . X_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k . X_i}}$$

$$\Pr(Y_i = 2) = \frac{e^{\beta_2 . X_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k . X_i}}$$

Similarly,

$$\Pr(Y_i = K - 1) = \frac{e^{\beta_{k-1} . X_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k . X_i}}$$

Different regression estimation coefficients exist and the Scikit-learn provides parameters that can be applied according to the specific problem. In this project, logistic regression is implemented using the one vs rest classifier approach, treating multiclass problem as binary class problem, that is run several times on each outcome chosen as the reference.

2.1.1.3 Support Vector Machines

Support Vector Machines (SVM) were originally designed for binary classification. But, they can effectively be extended to multiclass classification. Multiclass SVM methods either use multiple binary classifiers or a larger optimization problem is required. There are two popular methods in SVM multiclass classification: One-Against-All, One-Against-One.

The one-against-all method constructs '*k*' SVM models where '*k*' is the number of classes. The m^{th} SVM is trained with samples that belong class '*m'* with positive data labels and all other samples that are not in class *m* with negative labels. Classification occurs with

$f(x) = arg\ max_i\ f_i(x)$, where $f(x)$ is the class x belong to. The class for test data is the class which classifies test data with greatest margin.

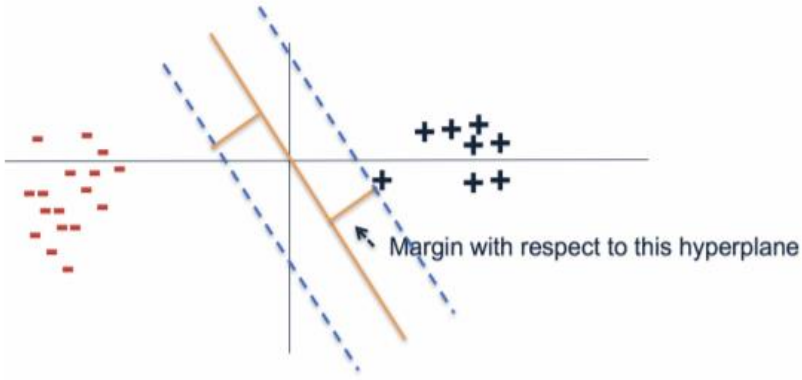

Figure 2.1 *Margin for binary classifier* **(Roth, 2016)**

In binary classifier, margin can be considered as the smallest distance between positive sample and negative sample. For multiclass case, margin is defined as score difference between highest and second highest scoring labels as shown below.

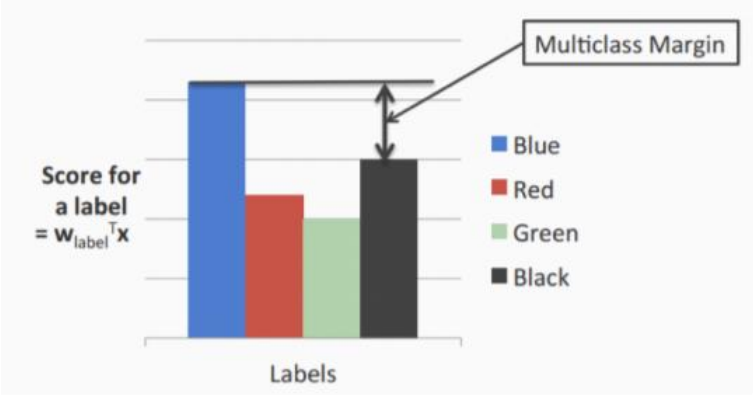

Figure 2.2 *Multiclass margin* **(Roth, 2016)**

The scores are calculated by taking the dot product of the weight vector '*w'* for each label and '*x'*. The labels with low scores apart from the top two are discarded. The goal in multi-class SVM is to maximize the margin or equivalently minimize the total norm of weights such that the true label has score at-least one more than the second-best label (Roth, 2016).

In one-against-one strategy, the strategy is to build a set of classifiers and choose the class that is selected by most of the classifiers. It involves building $\frac{k(k-1)}{2}$ classifiers but the training time required may be low as the training data set for each classifier is much smaller. In this project, I used one against all approach while implementing SVM with Linear SVC (kernel=linear) because when we have large number of categories that the documents should be divided into, building large number of classifiers may not be optimal.

2.2.2 Clustering Methods

Clustering is an unsupervised learning technique which is used to find similarities in the data and group them together. It plays a significant role when used to draw inferences from data sets consisting of input data without labeled responses. Below, I will discuss briefly about the clustering techniques I used in my project.

2.2.2.1 k-means Clustering

It takes all the documents as input and then partitions them and separates them it into *k* clusters. Partition is done in such a way that inter cluster distance between data points in a cluster is very less. In k-means clustering, K seeds are formed first, and then the observations are grouped into K clusters based on the distance with each of K seeds (often Euclidean distance) (MacQueen, 1967). The observation is included in the n^{th} seed or cluster, if the distance between the observation and the n^{th} cluster is minimum, when compared to other clusters. Once all data points are assigned to their respective clusters, new position of each centroid is calculated as the average position of all the points in its cluster. These steps keep repeating until the centroid stops moving a lot from each iteration to iteration.

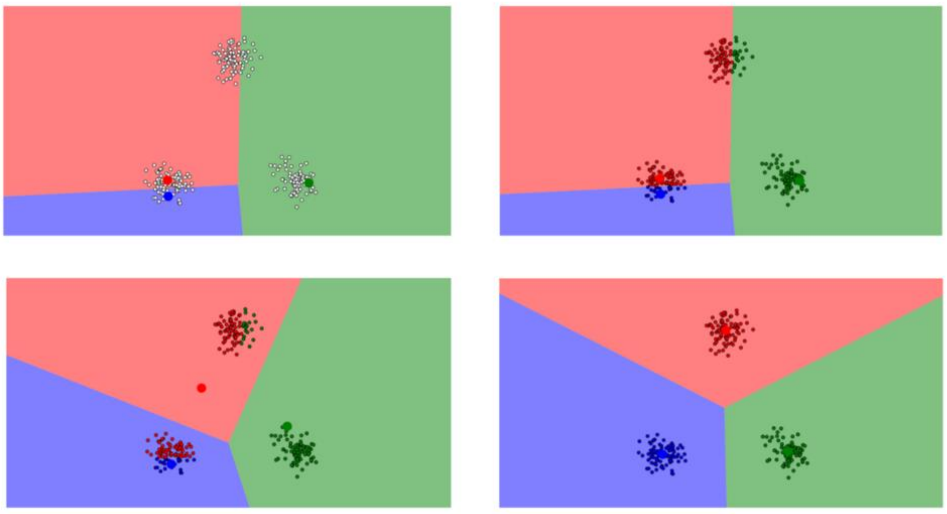

Figure 2.3 *Formation of clusters using k-means* **(Miani, 2017)**

In the above figure, we can see 3 randomly picked seeds each in different colors (Red, Green, Blue). Each point in the plane is colored according the centroid that it is closest to at each moment. The centroids (the larger blue, red, and green circles) start randomly and then quickly adjust to capture their respective clusters. We can observe that the centroids are recomputed in each iteration to make clustering process better, as we keep adding the documents.

2.2.2.2 Latent Dirichlet Allocation (LDA)

Topic modeling is one of the most powerful techniques in text mining for data mining, latent data discovery, and finding relationships among text documents. LDA is one of the most popular methods in this field. It is an unsupervised generative probabilistic model. The basic idea of LDA is that, the documents are represented as random mixtures over latent topics, where a topic is characterized by a distribution over words. (Waldron, 2015) The following steps are followed while modelling topics using LDA.

- **Number of topics** is fixed and given as an input to the algorithm
- Next, algorithm assigns temporary topic to each word in each document according to **Dirichlet distribution**.

Then, Collapsed Gibbs Sampling (CGS) algorithm can work from this first random guess and over many iterations to discover the topics. The following is simple description of next steps (Waldron, 2015):

- For each word in each document the following steps are followed:
    - For each of the K number of topics,
        - First find the percentage of words in the document that were generated from this topic. This will give us an indication of how important the topic is to a document.
        - Next, find the percentage of the topic that came from this word across all documents. This will give us an indication of how important the word is to the topic.
        - Multiply the two percentages together, this will give an indication of how likely it is that the topic in question generated the word under consideration.
- The products obtained from each topic are compared and the topic with the highest value is chosen as the topic for that word.
- This process is repeated until a topic assignment to a word changes no further which means that the topics will have converged into K distinct topics.

So, in perspective of LDA documents are created by choosing the topics from which the document will be generated, and the proportion of each document to come from each topic. Then, it generates appropriate words from the topics chosen in the proportions specified.

Chapter 3 - Implementation

This chapter discusses about the data set used, data-preprocessing, implementation steps of this project.

3.1 Data Sets

- News Aggregator Data set (Dheeru & Karra Taniskidou, 2017) contains 422937 news headlines collected from a web aggregator from 10-March-2014 to 10-August-2014. The data is available directly through a csv file.
- British Broadcasting Corporation (BBC) news data set (Greene & Cunningham, 2006) is in the form of raw text documents and contains 2225 text files from the BBC news website corresponding to stories in five topical areas from 2004-2005. The text documents are arranged into 5 folders named with the class label (business, entertainment, politics, sport, tech) and each of them contains new articles related to that class label.

3.1.1 Data Preparation

Since the news aggregator data set is available as a csv, I import Pandas, which is a python library often used for data manipulation and analysis. Before importing the csv into Pandas data frame for further analysis, it is cleaned for rows that are not properly delimited. After data cleaning (removing duplicates rows and null values), 406916 news headlines are retained. Now, we have 9 attributes as shown below.

```
Data columns (total 9 columns):
id            406916 non-null int64
headline      406916 non-null object
source_url    406916 non-null object
publisher     406914 non-null object
category      406916 non-null object
story_id      406916 non-null object
host_url      406916 non-null object
timestamp     406916 non-null float64
date          406916 non-null object
dtypes: float64(1), int64(1), object(7)
```

The main attributes that are used are ‘headline’ and ‘category’. The distribution of various categories in the data set is as follows:

e = entertainment	147178
b = business	111666
t = science and technology	104439
m = health	43633

For the BBC data, I parsed through folders of each category, read content from each file and then store into the Pandas data frame in a specific format, which is further elaborated in the below section. After deleting the duplicate records using *pandas.DataFrame.drop_duplicates* function, we are left with 2060 news documents. The raw data set has documents in which first line of every document is the title of the article and all the other lines are content of the article. While loading it into the data frame, all the necessary attributes are separated as columns. We now have the following data columns in the data frame.

```
Data columns (total 5 columns):
content          2060 non-null object
title            2060 non-null object
category_name    2060 non-null object
category_id      2060 non-null int64
raw_content      2060 non-null object
dtypes: int64(1), object(4)
```

The ‘*title’* attribute is the first line of every document. ‘*category_name’* contains class labels corresponding to each article stored from the directory names of the data set we have (The articles related to business are inside the folder name ‘business’). ‘*category_id’* is the integer id we assign to each class label ('business': 1, 'entertainment': 2, 'politics': 3,'sport': 4, 'tech': 5).

The distribution of categories in this data set is as follows:

business	498
sport	489
politics	388
entertainment	362
tech	323

3.1.2 Data Preprocessing

Data preprocessing is an important step in text classification. The raw content should be processed to eliminate unnecessary data and avoid misleading outcomes. Duplicates and null values are removed before loading it to the data frame as explained above. Before we pass the data to the classifier, we need to provide high quality representation of text and I used NLTK's advanced text processing mechanisms for this purpose. Tokenizer, WordNetLemmatizer and stop-word removal techniques have been used. To further analyze our data set, we need to transform each article's text to a feature vector, a list of numerical values representing the text's characteristics. This is because most ML models cannot process raw text, instead only deal with numerical values. We call this process feature extraction and I used the most common Bag of words TF-IDF approach which has been elaborated in section 2.1.2. This is implemented using *sklearn.feature_extraction.text.TfidfVectorizer* in this project. After preprocessing and feature extraction, number of features (Vocabulary size) are as follows:

BBC news articles – 6065

News Aggregator data set – 45816

3.2 Implementation Steps

Below, the process flow along with implementation steps is shown. Figure 3.1 shows the process, starting from collecting the news headlines from news aggregator data set and news articles from BBC data set.

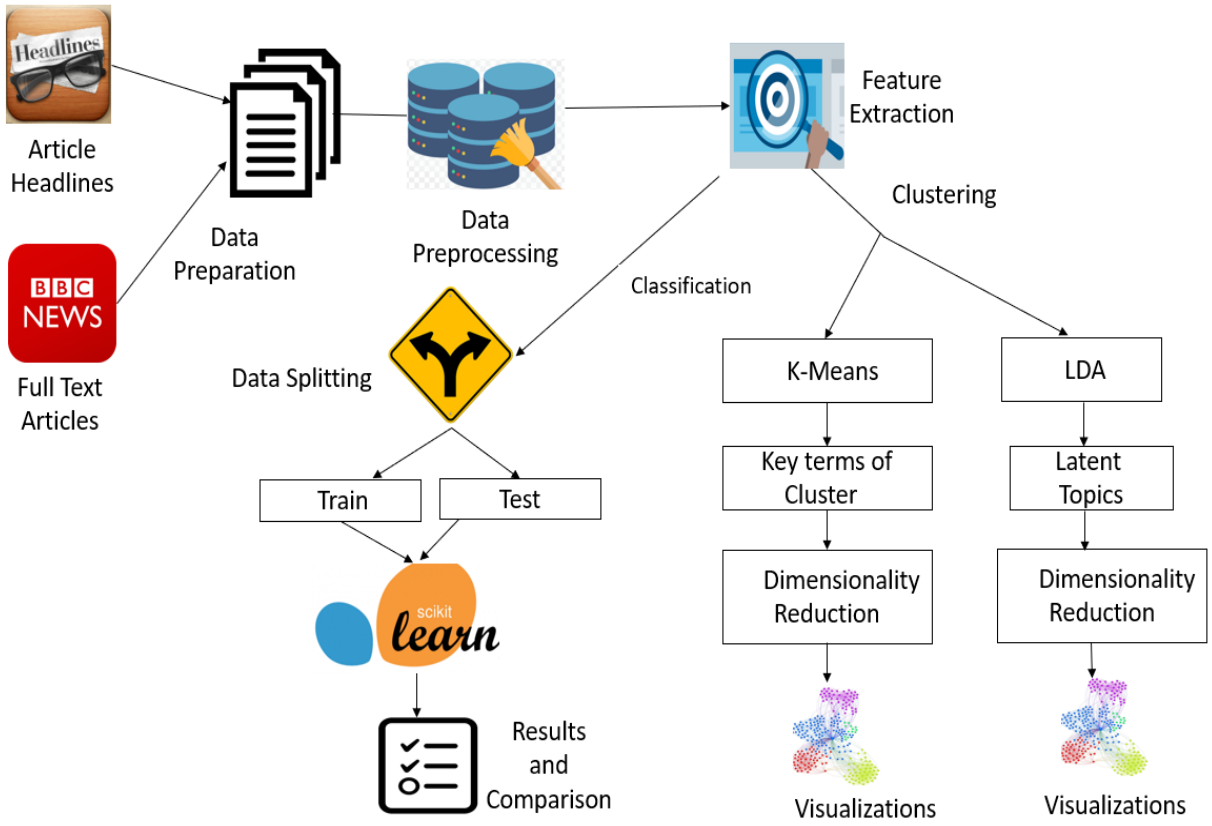

Figure 3.1 *Workflow diagram of project showing implementation steps*

Chapter 4 - Experiments

This chapter explains the experimental design used for implementing this project. Different experiments that are conducted on the data set using the various approaches are mentioned in sections 2.2.1 and 2.2.2.

4.1 Training and Test Data Sets

Separating data into training and testing sets is an important part of evaluating classification models. The training set is used to build the model (determine its parameters) and the test set is used to measure its performance (holding the parameters constant). I divided both the data sets into 70% − 30%, where 70% is training data and 30% is test data.

News Aggregator Data Set	
Total Records	*406916*
Training Records	*284841*
Testing Records	*122075*

Table 4.1 *Training and test data splits for headlines*

BBC news Data Set	
Total documents	*2060*
Training documents	*1442*
Testing documents	*618*

Table 4.2 *Training and test data splits for full text articles*

4.2 Experiment Design

In this section, the approach used to conduct the experiments on the data set using several machine learning algorithms mentioned in the section 2.2.1 and 2.2.2 is discussed. All the algorithms were implemented using Scikit-learn: A machine learning package in Python.

First, I experimented on BBC news data set, both on headlines and full text articles. Various classification algorithms are implemented to predict the category, given a headline text or article content. Due to the insufficiency of data, performance is quite low when classifiers are trained using just headlines from BBC data set. For this reason, I experimented on a larger data set (news aggregator data set) with more number of headlines to see if classification task works well with headlines as well. Results of the above-mentioned experiments are shown in section 5.2.1. Then, I experimented in depth on full text BBC articles to evaluate the models and results, by doing a k-fold cross validation and plotting ROC's.

In *k-fold* cross validation (Pedregosa, Varoquaux, Gramfort, & Michel, 2011), the training set is split into *k* smaller sets and for each of the *k* folds, a model is trained using *k-1* of the folds as training data and the resulting model is validated on the remaining part of the data. (i.e., it is used as a test set to compute a performance measure such as accuracy). The mean cross validation scores, and various metrics like accuracy, precision, recall, and F1-score to evaluate my models are shown in sections 5.2.1, 5.2.2.

Finally, I experimented with two clustering algorithms K- means and LDA from section 2.2.2. The clustering algorithms for headlines (news aggregator data set) are applied on a subset of 10,000 headlines separated from the original set. The cluster models are fit to the document vectors using the *fit* and *fit_transform* methods. Then, the dimensions of matrices obtained as cluster outputs are reduced first using *TruncatedSVD* to 50, and then using *TSNE* to 2 to

visualize the clusters on a 2 D plane. I then, showed the visualizations of clusters and keywords that form the clusters as a qualitative analysis.

4.2.1 Support Vector Machines

Support Vector Machines is implemented using the Scikit-learn SVM class component LinearSVC (*sklearn.svm.LinearSVC*). LinearSVC works best when compared to nonlinear methods. Multi class mode of SVC is implemented using one vs one scheme whereas LinearSVC uses one vs the rest. SVC gives very poor results when experimented with default parameters. Therefore, LinearSVC is chosen and experimented with multi-class strategy: *'ovr'* trains n_classes one-vs-rest classifiers, while *'crammer_singer'* optimizes a joint objective over all classes. There is no difference in the accuracies using both but, *'crammer_singer'* is more expensive to compute. Therefore, default '*ovr*' is chosen. The results obtained show great accuracy with this classifier.

4.2.2 Logistic Regression

Logistic Regression in Scikit-learn is implemented using the *sklearn.linear model.LogisticRegression* class. Different 'multi_class' parameters are experimented. For multi_class='multinomial', liblinear solver cannot be used. Therefore, Iexperimented with default 'ovr' multi class option. The results obtained with this model are best compared to other classifier models that I have used.

4.2.3 Random Forests

Random forest algorithm is implemented by importing *sklearn.ensemble.RandomForestClassifier* from Scikit-learn. The performance of classifier is evaluated by tuning the number of trees in ensemble, maximum features, and the depth of tree.

Increasing the depth of the tree increased the accuracy. Limiting the tree depth typically will make the ensemble converge a little earlier. Also, our experiments show that the depth of tree 10 achieves good accuracy when applied on BBC data set. The same depth of tree is used for the latter data set to present the results, because kernel cannot handle very large trees. Depth of tree can be increased further but, computation takes more time and consumes lot of memory. The parameter *class_weight = 'balanced'* is used to automatically adjust weights inversely proportional to the class frequencies from input data.

4.2.4 Multinomial Naïve Bayes

Similar to the above classifiers, I implemented Naïve Bayes by importing *sklearn.naive_bayes.MultinomialNB.*

4.2.5 k-means

An alternative implementation of K-means called MiniBatchkMeans is used. It is imported from Scikit-learn using *sklearn.cluster import MiniBatchKMeans*. Parameters like n_clusters, init='k-means++', n_init, init_size, batch_size, max_iter are tweaked and the final values used are shown in section 5.3.2, 5.3.3.

n_clusters (int): The number of clusters to form as well as the number of centroids to generate.

init: {'k-means++', 'random'}, default: 'k-means++'

Method for initialization, defaults to 'k-means++':

'k-means++': selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.

max_iter (int): Maximum number of iterations over the complete data set before stopping independently of any early stopping criterion heuristics.

batch_size (int**)**: Size of the mini batches. (default=100)

init_size (int): Number of samples to randomly sample for speeding up the initialization

4.2.6 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is implemented by importing *lda.LDA* class from python packages. LDA, being a probabilistic graphical model (i.e. dealing with probabilities) requires raw counts, therefore, a CountVectorizer() is used with parameters min_df = 4, max_df = 0.5, ngram_range = (1,2). These parameters are chosen based on some research along with a series of experiments. Then, I used parameters n_topics=30, n_iter =2000 to show the results of LDA.

4.2.7 Dimensionality Reduction

- The class *sklearn.decomposition.TruncatedSVD* is imported from Scikit-learn. It was used in the experiments to reduce the dimensions linearly. The *n-components* parameter is used to set the number of dimensions desired and it was set to 50 in this experiment.
- The class *sklearn.manifold.TSNE* is imported from Scikit-learn. t-SNE helps visualize high dimensional data. It converts the similarities between the data points to joint probabilities and tries to minimize divergence between the joint probabilities of high dimensional data. For this experiment, it is used to reduce the dimensions to 2, so that resulting clusters of k-means and LDA can be visualized on a 2D plane.

Chapter 5 - Experimental Results

This chapter explains the results of several experiments described in the section 4.2 and evaluation metrics used to evaluate the models.

5.2 Classification Models - Evaluation

5.2.1 Performance Metrics

5.2.1.1 Accuracy

Accuracy is the number of correct predictions made compared to the total number of predictions.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ Predictions}$$

5.2.1.2 Precision and Recall

Precision is the fraction of retrieved instances that are relevant. Precision is the ratio of True Positives to the sum of True Positive (TP) and False Positive (FP) (Joshi, 2016).

$$Precision = \frac{TP}{TP + FP}$$

Recall is the fraction of relevant instances that are retrieved. Recall, also called true positive rate or sensitivity is the ratio of True Positive to the sum of True Positive and False Negative (FN) (Joshi, 2016).

$$Recall = \frac{TP}{TP + FN}$$

5.2.1.3 F1-score

F1-score is a measure of the test's accuracy. It is calculated using the precision and recall of the test data. F1-score is also defined as the harmonic mean of precision and recall (Joshi, 2016).

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Performance Metrics				
Classifier Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.9741	0.9719	0.9727	0.9722
Linear SVC	0.9692	0.9663	0.9705	0.9680
Multinomial NB	0.9676	0.9654	0.9663	0.9656
Random Forest	0.9498	0.9489	0.9480	0.9482

Table 5.1 *Performance metrics of classification algorithms - BBC articles*

Performance Metrics				
Classifier Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.6197	0.6764	0.6125	0.6225
Linear SVC	0.6165	0.6596	0.6096	0.6196
Multinomial NB	0.6165	0.6839	0.6085	0.6196
Random Forest	0.5194	0.73055	0.51136	0.5288

Table 5.2 *Performance metrics of classification algorithms - BBC headlines*

Performance Metrics				
Classifier Model	**Accuracy**	**Precision**	**Recall**	**F1 score**
Logistic Regression	0.9458	0.9459	0.9374	0.9415
Linear SVC	0.9532	0.9517	0.9487	0.9502
Multinomial NB	0.9388	0.9397	0.9296	0.9344
Random Forest	0.8015	0.8021	0.7931	0.7918

Table 5.3 *Performance metrics of classification algorithms – news aggregator headlines*

5.2.2 Cross validation

To evaluate each model and high accuracies obtained as results for full text articles, I used the *k*-fold cross-validation technique: iteratively training the model on different subsets of the data and testing against the held-out data. Scikit-learn provides a utility

function, *cross_val_score*, that allows us to run this operation in a single line of code. Here, I used *k=10* to evaluate the model's performance.

Figure 5.1 below shows the box plot of the results and Table 5.4 shows mean accuracies of each model.

Model name	**Mean accuracy**
Logistic Regression	0.9718
Linear SVC	0.9708
Multinomial NB	0.9684
Random Forest Classifier	0.9486

Table 5.4 *Cross validation accuracies - BBC articles*

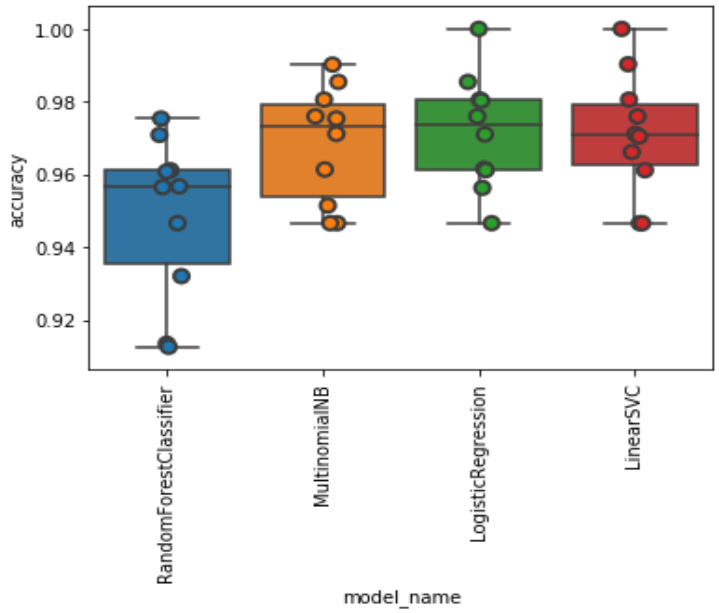

Figure 5.1 *Tukey Box plot of cross validation scores- BBC articles* (Weisstein, 2018)

From the above box plot, we can observe that the results of Random Forests, Naïve Bayes show some variance compared to other models. The accuracies of LogisticRegression, LinearSVC, Multinomial NB are high with about 97% accuracy. However, there is variance between the results of Naïve Bayes and random Forests compared to the other two. The observed data points

are plotted for each fold denoting accuracies for each one of them. The points are obtained by plotting a strip plot along with a combination of box plot and both are imported form the Seaborn library. A horizontal offset of data points is observed with parameter called *'jitter' = true* which drifts the data points along the categorical axis by default to observe the distribution clearly in-case of overlapping points (Waskom, 2012-2017).

5.2.3 Confusion Matrix

A *confusion matrix* for a model, is a matrix which has the true positive, true negative, false positive and false negative values for the given test data. It helps to show the discrepancies between the predicted and actual labels to understand main sources of misclassification in our test set. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions. Heat maps are used to represent matrix values as colors to better visualize the level of classification or misclassification with the hue ranges. In the below results, since the misclassified outputs are very low, the color differentiation cannot be noticed. And the color variation across the diagonal are just numbers of correctly classified records. The shades in color are different as they show the variation of number of samples for each category in the test set but do not represent the level of accuracy of classification for each class separately. The color bar (legend) is automatically obtained with default suitable divisions according to the distribution of data using the *seaborn.heatmap* function from Seaborn library.

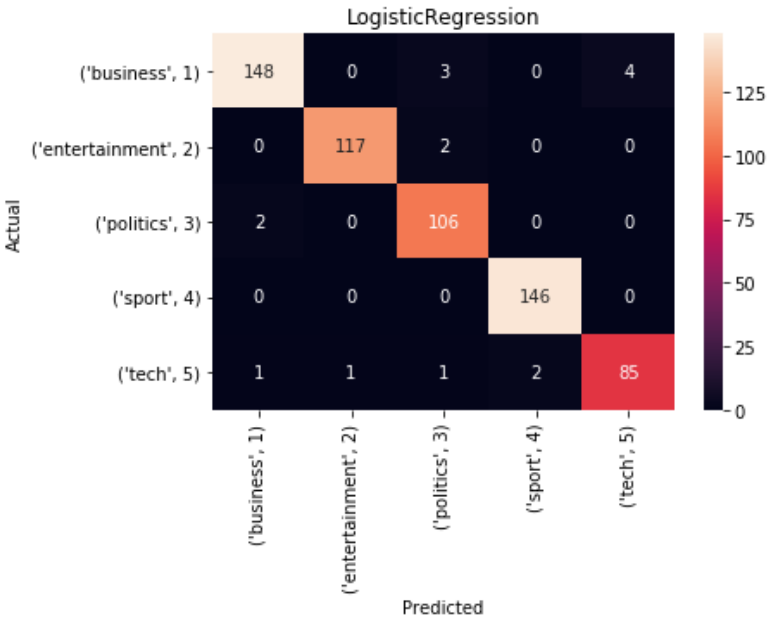

Figure 5.2 *Heat map for Logistic Regression*

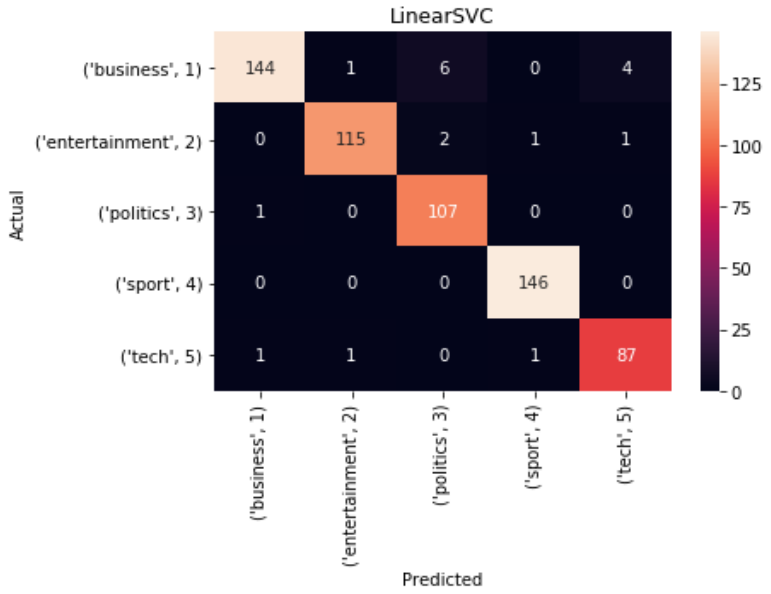

Figure 5.3 *Heat map for Linear SVC*

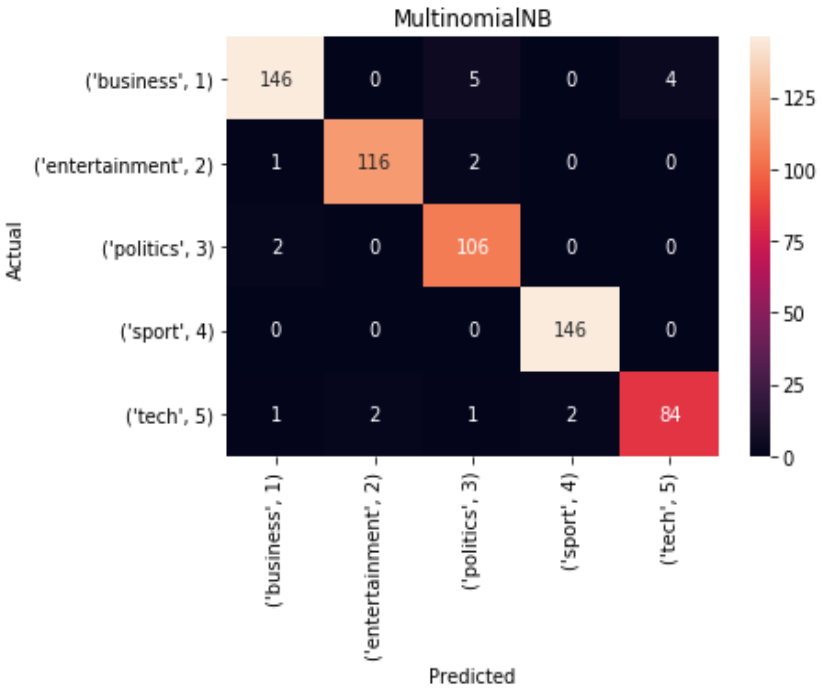

Figure 5.4 *Heat map for multinomial Naive Bayes*

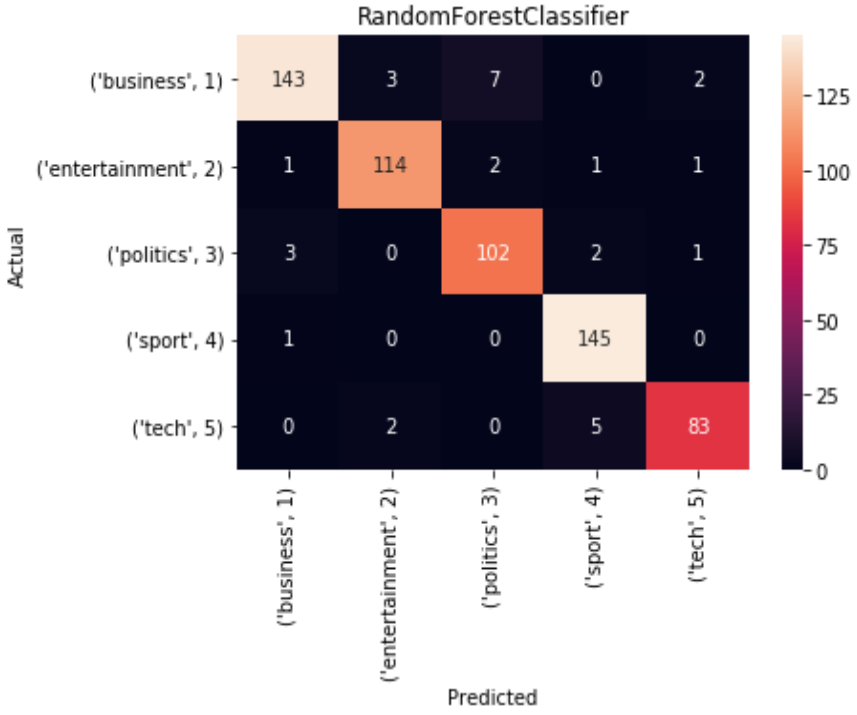

Figure 5.5 *Heat map for Random Forests*

From the above matrices, majority of the predicted outcomes are on the diagonal which are true positives. These matrices clearly reflect the high accuracies of Logistic regression, LinearSVC, multinomial NB as observed in the cross-validation results.

5.2.4 Receiver Operating Characteristic curve (ROC), Area Under the Curve (AUC)

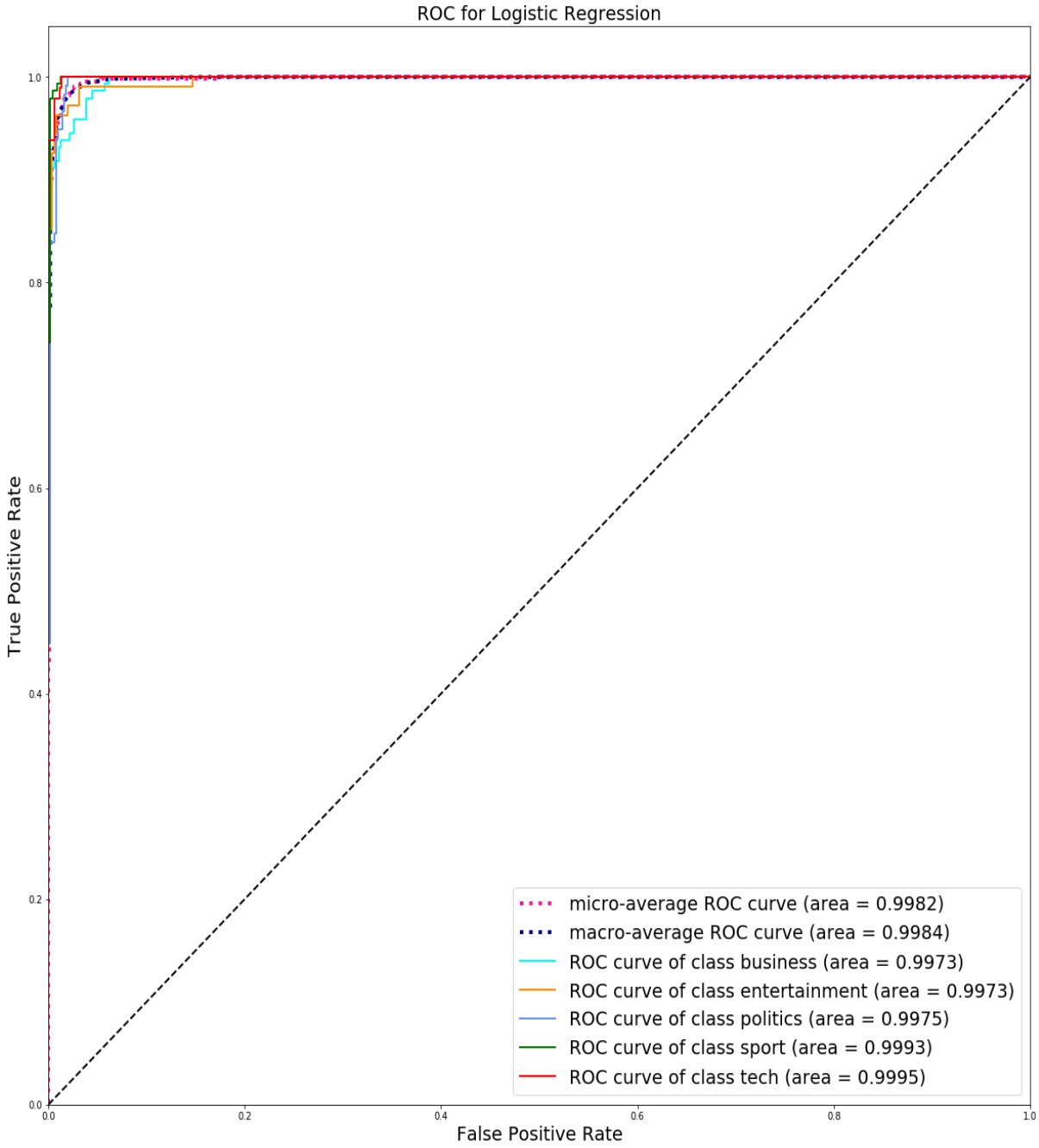

Figure 5.6 *ROC curve for logistic Regression*

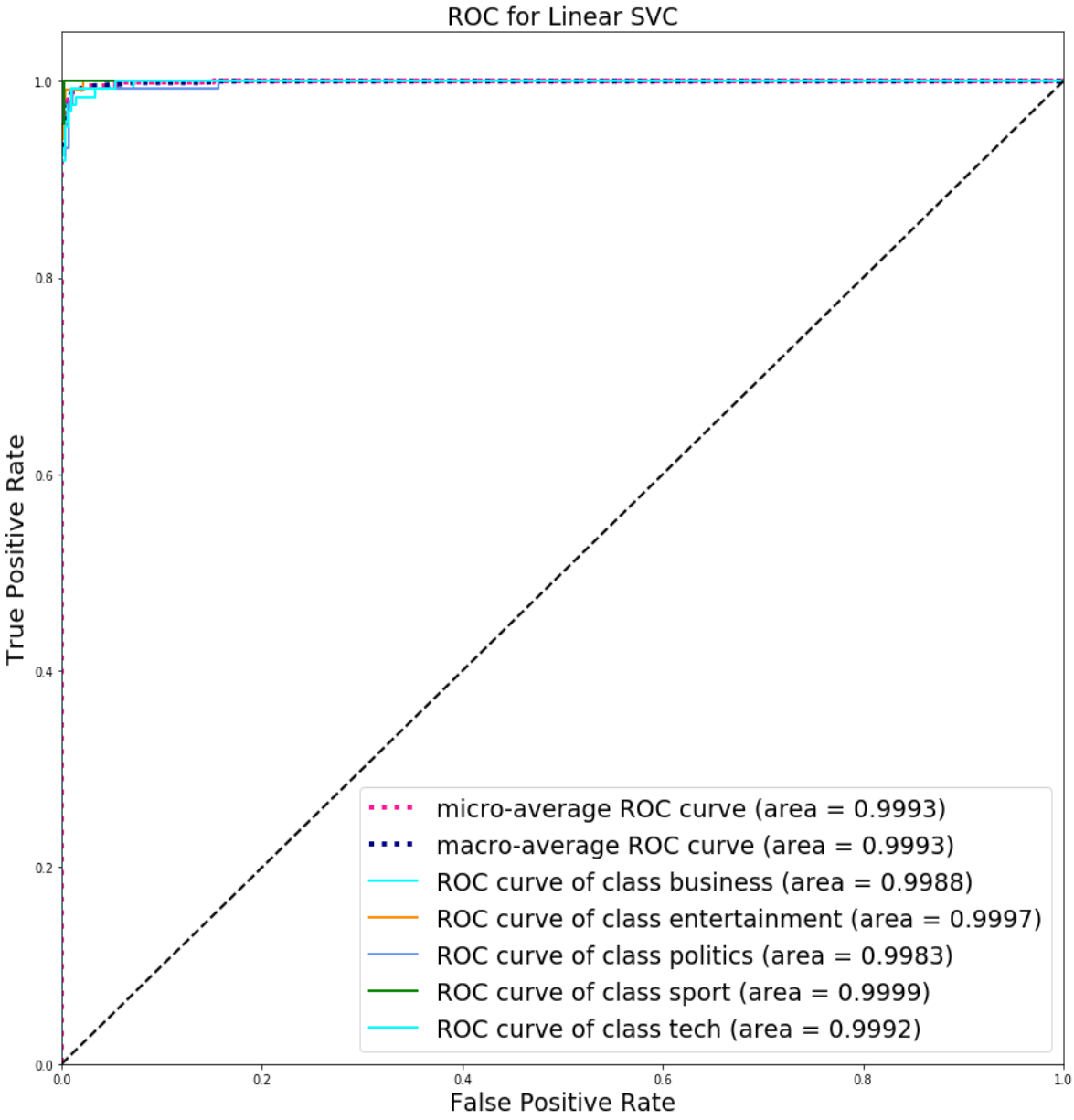

Figure 5.7 *ROC curve for Linear SVC*

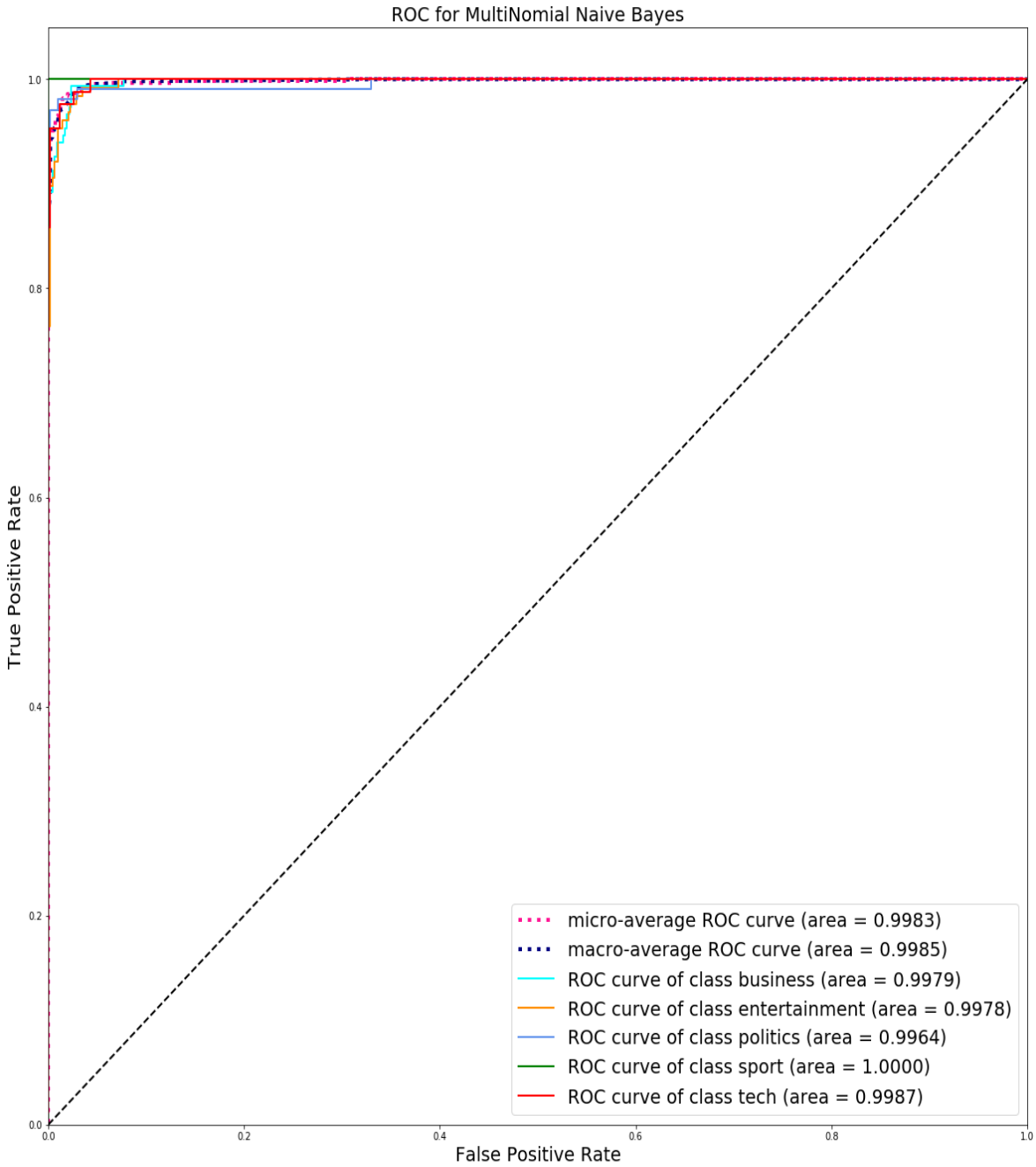

Figure 5.8 *ROC curve for Naive Bayes*

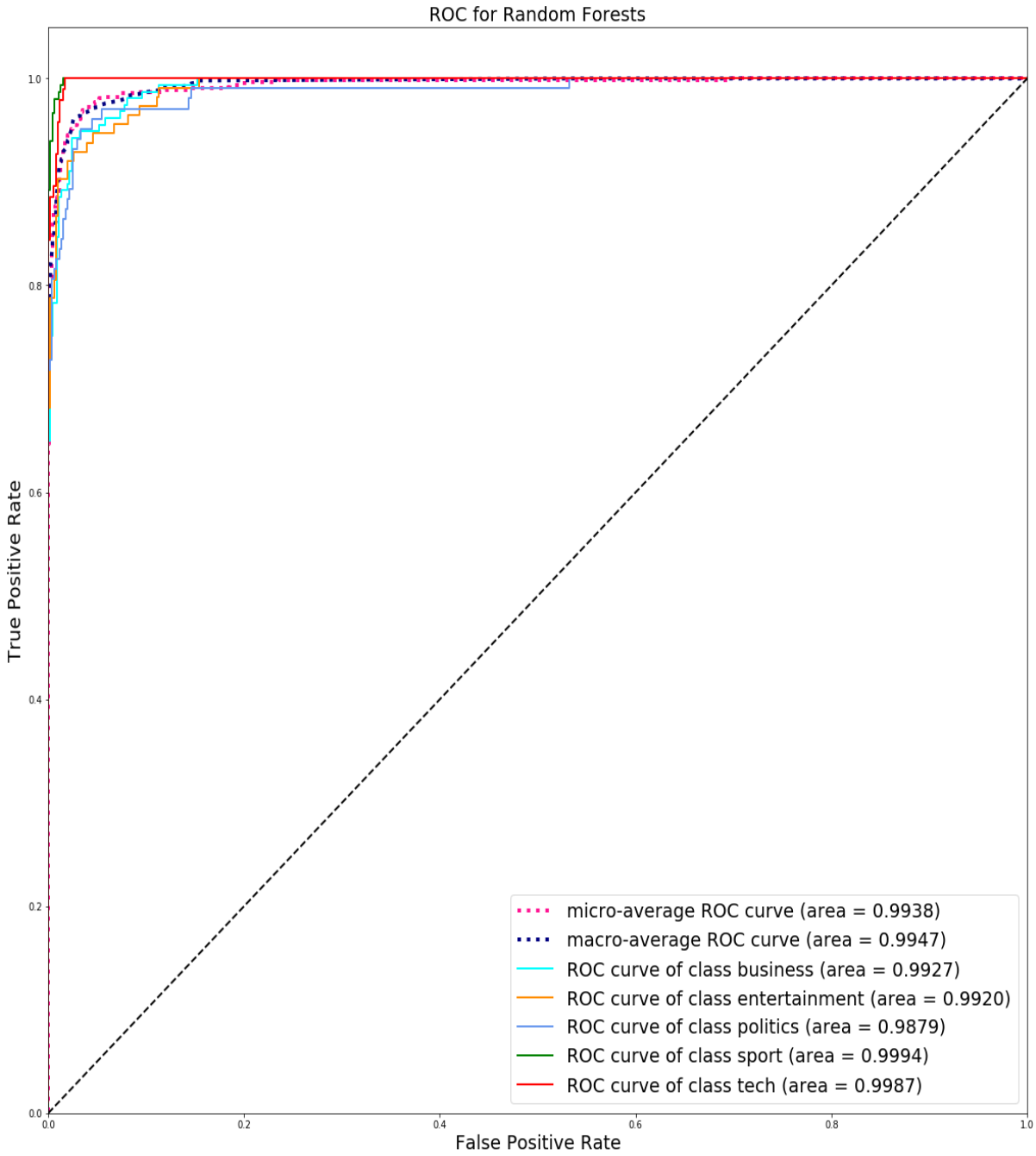

Figure 5.9 *ROC curve for Random Forests*

5.3 Clustering Results and Analysis

5.3.1 k-means (K=5)

MiniBatckKMeans is implemented as an alternative to k-means. The advantage of this algorithm is to reduce the computational cost by not using all the data set each iteration but a subsample of a fixed size. The parameters used for this algorithm are:

n_clusters= 5, init='k-means++', n_init=1, init_size=1000, batch_size=1000, max_iter=1000

Using *K=5*, which are the number of distinct categories of articles, we obtain the results shown in Figure 5.11.

The ten most significant keywords that are mainly responsible for formation of each cluster as shown in Figure 5.10. The keywords reflect the categories that the data points belong to. In this case, according to the ground truth values available they can be labelled as: Cluster 0 – Business, Cluster 1 - Entertainment, Cluster 2 - Politics, Cluster 3 – Sports, Cluster 4 – Technology.

```
Cluster 0:
growth | bank | sales | economy | shares | oil | economic | prices | china | analysts |

Cluster 1:
film | music | awards | award | band | star | actor | album | festival | films |

Cluster 2:
labour | election | blair | party | brown | mr blair | mr brown | howard | prime | prime minister |

Cluster 3:
england | cup | match | team | side | club | season | injury | players | final |

Cluster 4:
mobile | technology | users | software | games | computer | phone | net | microsoft | digital |
```

Figure 5.10 *Top 10 keywords of clusters*

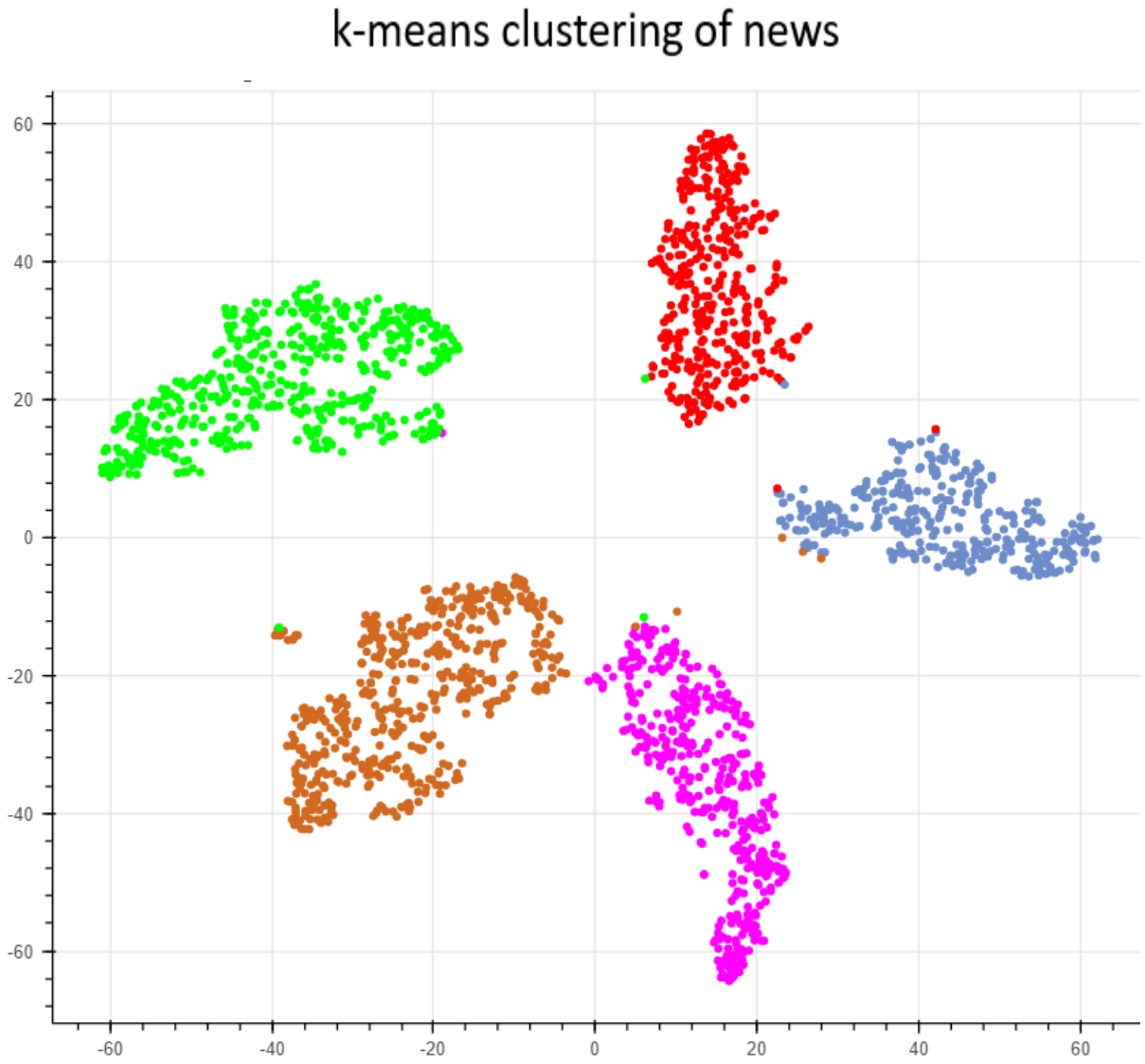

Figure 5.11 *k-means (K=5) clusters visualized using Bokeh*

From the above results, it can be deduced that given sufficient data and predefined number of clusters, categorization of text documents into superficial categories (like the ones above) can be performed using K-Means without the need for labelled data.

5.3.2 Visualization on full text – BBC data set

5.3.2.1 k-means (K=30)

From figure below Fig 5.14, the separated clusters can be observed. In the demo, I show the interactive plot by hovering on data point, to see only a part of article content, category and

the cluster it belongs to. Then it is observed data points that are relevant, are grouped as clusters.

Significant words from each cluster are shown in Fig 5.12 and Fig 5.13.

The words include both unigrams and bigrams separated by comma delimiter.

```
Keywords of Cluster 0:
lords , lord , human rights , rights , human , law , government , house , bill , police ,

Keywords of Cluster 1:
wales , scotland , williams , rugby , italy , six nations , ireland , nations , france , coach ,

Keywords of Cluster 2:
sales , bn , profits , car , bmw , profit , company , quarter , gm , us ,

Keywords of Cluster 3:
party , labour , election , mr , howard , tax , tories , mr howard , tory , lib ,

Keywords of Cluster 4:
people , broadband , tv , show , technology , net , digital , bt , users , online ,

Keywords of Cluster 5:
film , films , festival , movie , actor , star , comedy , box office , hollywood , director ,

Keywords of Cluster 6:
band , album , song , best , rock , music , number , chart , singer , single ,

Keywords of Cluster 7:
mobile , phone , phones , mobile phone , mobiles , technology , people , handsets , music , mobile phones ,

Keywords of Cluster 8:
shares , bn , fiat , stock , firm , company , euros , lse , market , boerse ,

Keywords of Cluster 9:
dollar , china , bn , deficit , us , mci , euro , trade , budget , bush ,

Keywords of Cluster 10:
open , seed , australian , roddick , match , australian open , tennis , win , federer , set ,

Keywords of Cluster 11:
apple , tobacco , smoking , information , court , journalists , lawsuit , case , ruling , government ,

Keywords of Cluster 12:
mr , blair , mr blair , brown , labour , election , prime , minister , prime minister , mr brown ,

Keywords of Cluster 13:
yukos , russian , gazprom , oil , russia , rosneft , yugansk , bn , khodorkovsky , auction ,

Keywords of Cluster 14:
games , game , nintendo , gaming , gamers , ds , xbox , video , sony , ea ,
```

Figure 5.12 *Top 10 keywords of clusters 0-14*

```
Keywords of Cluster 15:
poverty , report , university , government , children , aid , peace , education , poor , countries ,

Keywords of Cluster 16:
best , film , awards , award , actress , actor , oscar , aviator , nominated , nominations ,

Keywords of Cluster 17:
england , robinson , rugby , ireland , six nations , nations , six , game , france , wales ,

Keywords of Cluster 18:
software , mac , bittorrent , computer , patents , mac mini , apple , mini , ibm , patent ,

Keywords of Cluster 19:
virus , microsoft , mail , security , spam , software , users , program , windows , spyware ,

Keywords of Cluster 20:
government , mr , bn , bank , india , us , country , new , could , minister ,

Keywords of Cluster 21:
turkey , eu , turkish , bn , citigroup , talks , country , cyprus , membership , deal ,

Keywords of Cluster 22:
club , rangers , bellamy , celtic , souness , villa , manager , game , everton , juninho ,

Keywords of Cluster 23:
growth , rates , rate , economy , figures , prices , bank , rise , market , december ,

Keywords of Cluster 24:
olympic , athens , kenteris , athletics , drugs , doping , iaaf , world , champion , thanou ,

Keywords of Cluster 25:
music , industry , digital , musicians , music industry , piracy , sales , peer , digital music , copyright ,

Keywords of Cluster 26:
economy , growth , oil , prices , demand , crude , japan , exports , economic , domestic ,

Keywords of Cluster 27:
chelsea , liverpool , league , game , cup , arsenal , season , club , win , goal ,

Keywords of Cluster 28:
eu , commission , lisbon , european , straw , countries , anil , reliance , mr , mr straw ,

Keywords of Cluster 29:
award , prize , radio , band , concert , book , winner , winners , music , show ,
```

Figure 5.13 *Top 10 keywords of clusters 15-29*

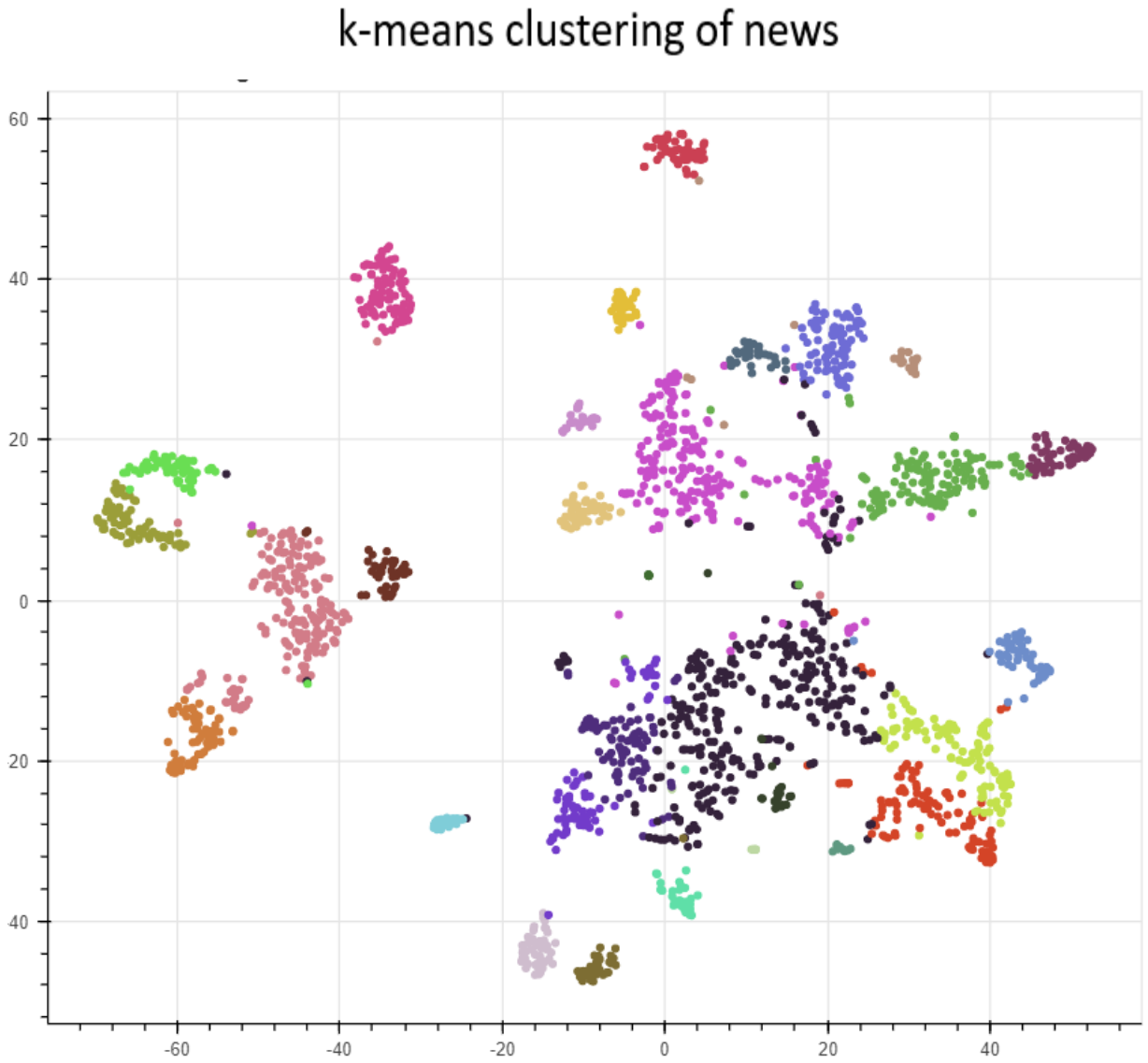

Figure 5.14 *k - means clusters (K=30) visualized with Bokeh applied on full text*

However, k-means works based on the assumption that each cluster is attributed to a single topic and gives disjoint clusters. Some overlapping clusters are seen in the above picture. But, documents in general are characterized by a mixture of topics. LDA is useful in such cases. Reasonably accurate mixtures of topics in a document set can be captured. Better results can be seen using LDA. The results of LDA algorithm are shown in the below section.

5.3.2.2 LDA

LDA, being a probabilistic graphical model (i.e. dealing with probabilities) requires raw counts, so a CountVectorizer() is used with parameters min_df=4, max_df=0.5, ngram_range=(1,2). These parameters are chosen based on some research along with a series of experiments. Then, we choose the parameters for LDA **n_topics = 30**, n_iter = 2000. The number of topics for LDA are chosen to be 30 because it seemed to be a realistic value to nicely divide the data set into latent topics. The topics formed as a result of LDA are shown in Figure 5.15 and the visualization of topics is shown in Figure 5.16. The underlined words denote bigrams in the topics. Since both unigrams and bigrams were considered, both appear as our topic words.

Topic 0: court legal fraud case action bank charges financial
Topic 1: china oil dollar india foreign japan state trade
Topic 2: mobile phone digital broadband music service phones services
Topic 3: deal bid offer club united board reports takeover
Topic 4: spokesman london saying ms campaign anti statement community
Topic 5: tax budget spending brown pay increase services taxes
Topic 6: england wales rugby france nations six nations ireland robinson
Topic 7: games gaming sony video dvd players nintendo online
Topic 8: search online web information apple google survey found
Topic 9: growth economy figures economic rise rate bank prices
Topic 10: open final champion olympic race title roddick match
Topic 11: european eu europe germany airline german deutsche air
Topic 12: yukos russian ukip drugs russia gazprom silk kilroy
Topic 13: club chelsea league united arsenal football manager liverpool
Topic 14: countries aid president international bush tsunami nations economic
Topic 15: police law rights bill human human rights without house
Topic 16: lord secretary minister blunkett straw foreign campbell mr blunkett
Topic 17: technology computer software around power system dr project
Topic 18: tv show television radio series programmes audience channel
Topic 19: half minutes goal ball points left lead break
Topic 20: scotland ireland williams irish jones italy scottish thomas
Topic 21: security users software mail net microsoft virus system
Topic 22: labour election blair party mr blair prime howard prime minister
Topic 23: music band album song chart record rock single
Topic 24: children report health local women education system ms
Topic 25: got ve really we know lot re great
Topic 26: life star london man career book musical later
Topic 27: sales shares profits euros business stock share executive
Topic 28: film award awards actor director films actress oscar
Topic 29: move need problem far called making rather give

Figure 5.15 *Topics extracted using LDA*

Topics 2, 13, 22 can be noticed as appropriately modelled. They are related to topics: mobiles and broadband, football leagues, Prime Minister Mr. Blair respectively. While Topic 25 doesn't give any specific meaning or a topic.

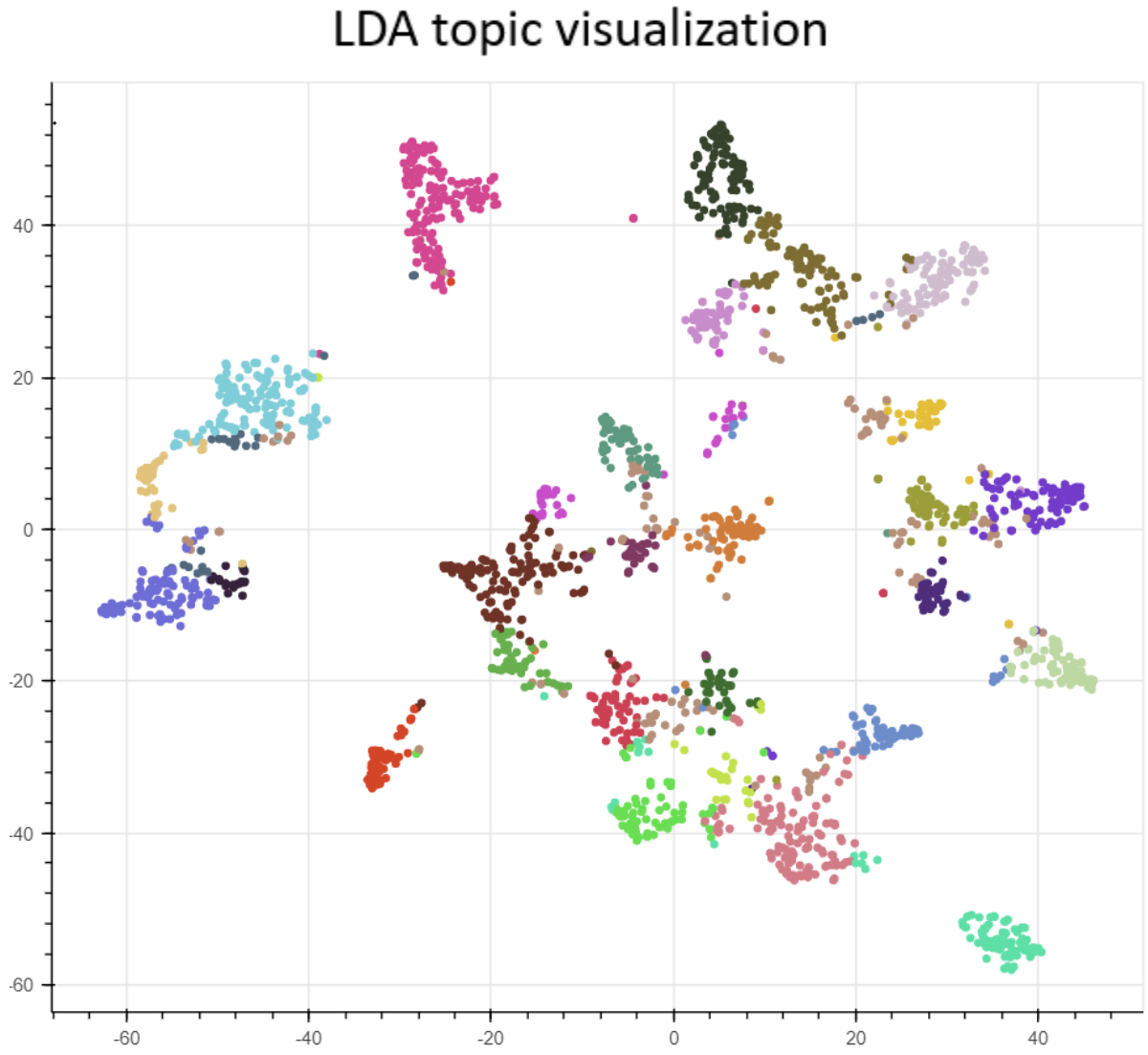

Figure 5.16 *LDA topic visualization using Bokeh*

5.3.3 Visualization on Headlines – News Aggregator Data

5.3.3.1 k-means

For this data set, a subset of 10,000 headlines is separated and clustering is performed on that subset. k-means is applied on this subset with K=30. It is observed from the clusters shown in Fig 5.19 that, dispersed clusters are observed even though the cluster keywords show some resemblance. All the parameters are same as of those used for the previous described data set.

```
Key words of Cluster 0:
true detective , detective , true , swift , taylor , taylor swift , keibler , stacy , stacy keibler , finale ,

Key words of Cluster 1:
stocks , us stocks , us , china , weak , lower , asia , data , ukraine , start ,

Key words of Cluster 2:
neil , neil young , young , music , pono , kickstarter , player , ponomusic , music player , neil degrasse ,

Key words of Cluster 3:
test , blood , blood test , alzheimer , predict , predict alzheimer , disease , alzheimer disease , test predict , predicts

Key words of Cluster 4:
web , bieber , justin bieber , justin , deposition , wide web , world wide , wide , world , ban ,

Key words of Cluster 5:
xbox , titanfall , xbox one , one , microsoft , titanfall xbox , launch , review , live , twitch ,

Key words of Cluster 6:
new , sxsw , us , mars , veronica , veronica mars , gas , says , google , ios ,

Key words of Cluster 7:
gomez , selena , selena gomez , bieber , justin bieber , justin , bieber selena , dance , dedicates , song ,

Key words of Cluster 8:
may like , skimming , app may , like skimming , reading app , speed reading , speed , reading , like , app ,

Key words of Cluster 9:
sneaks dressing , sneaks , fan sneaks , dressing room , dressing , room , cyrus fan , fan , miley cyrus , cyrus ,

Key words of Cluster 10:
candy , crush , candy crush , ipo , maker , crush maker , valuation , billion , king , ipo valuation ,

Key words of Cluster 11:
strangers , first , kiss , gm , first kiss , cars , owners , ignition , first time , switch ,

Key words of Cluster 12:
lindsay , lohan , lindsay lohan , list , lovers , reality , show , lohan reality , reality show , oprah ,

Key words of Cluster 13:
wearhouse , jos bank , jos , men wearhouse , men , bank , buy jos , billion , buy , deal ,

Key words of Cluster 14:
mortgage , rates , mortgage rates , bank , interest , mar , interest rates , mortgage interest , march , today ,
```

Figure 5.17 *Top 10 keywords of clusters 0-14 (headlines)*

```
Key words of Cluster 15:
gox , mt gox , mt , bankruptcy , files , bitcoin , sbarro , files bankruptcy , gox files , us ,

Key words of Cluster 16:
dunham , lena , lena dunham , snl , girls , saturday , molestation , saturday night , acting , night live ,

Key words of Cluster 17:
juan , juan pablo , pablo , bachelor , galavis , pablo galavis , bachelor juan , nikki , nikki ferrell , ferrell ,

Key words of Cluster 18:
malaysia , missing , plane , missing plane , malaysia airlines , airlines , mh370 , flight , search , report ,

Key words of Cluster 19:
finale , bachelor , bachelor finale , juan , pablo , juan pablo , finale juan , season finale , final rose , recap ,

Key words of Cluster 20:
miley , cyrus , miley cyrus , underwear , performs , cyrus performs , performs underwear , fan , change , cyrus fan ,

Key words of Cluster 21:
american , idol , american idol , top , eagle , american eagle , idol top , eagle outfitters , recap , outfitters ,

Key words of Cluster 22:
gold , ukraine , high , china , demand , safehaven , ukraine china , high ukraine , gains , 6month ,

Key words of Cluster 23:
snowden , edward snowden , edward , nsa , sxsw , leaks , speaks , snowden speaks , snowden sxsw , nsa leaker ,

Key words of Cluster 24:
urges gm , group urges , 1b victims , victims fund , set 1b , safety group , gm set , 1b , auto safety , victims ,

Key words of Cluster 25:
recall , gm , colorado , gm recall , taxes , pot , probe , million , marijuana , recreational ,

Key words of Cluster 26:
sdk wearable , wearable devices , release sdk , google release , devices , wearable , sdk , release , google , android ,

Key words of Cluster 27:
rises fivemonth , crimea tensions , fivemonth , fivemonth high , crimea , fears crimea , china fears , high china , gold rises , tensions

Key words of Cluster 28:
trailer , thrones , game thrones , game , season , new , thrones season , season trailer , new game , thrones trailer ,

Key words of Cluster 29:
china , data , china data , shares , weak , output , industrial , asian , wall , china industrial ,
```

Figure 5.18 *Top 10 keywords of clusters 15-29 (headlines)*

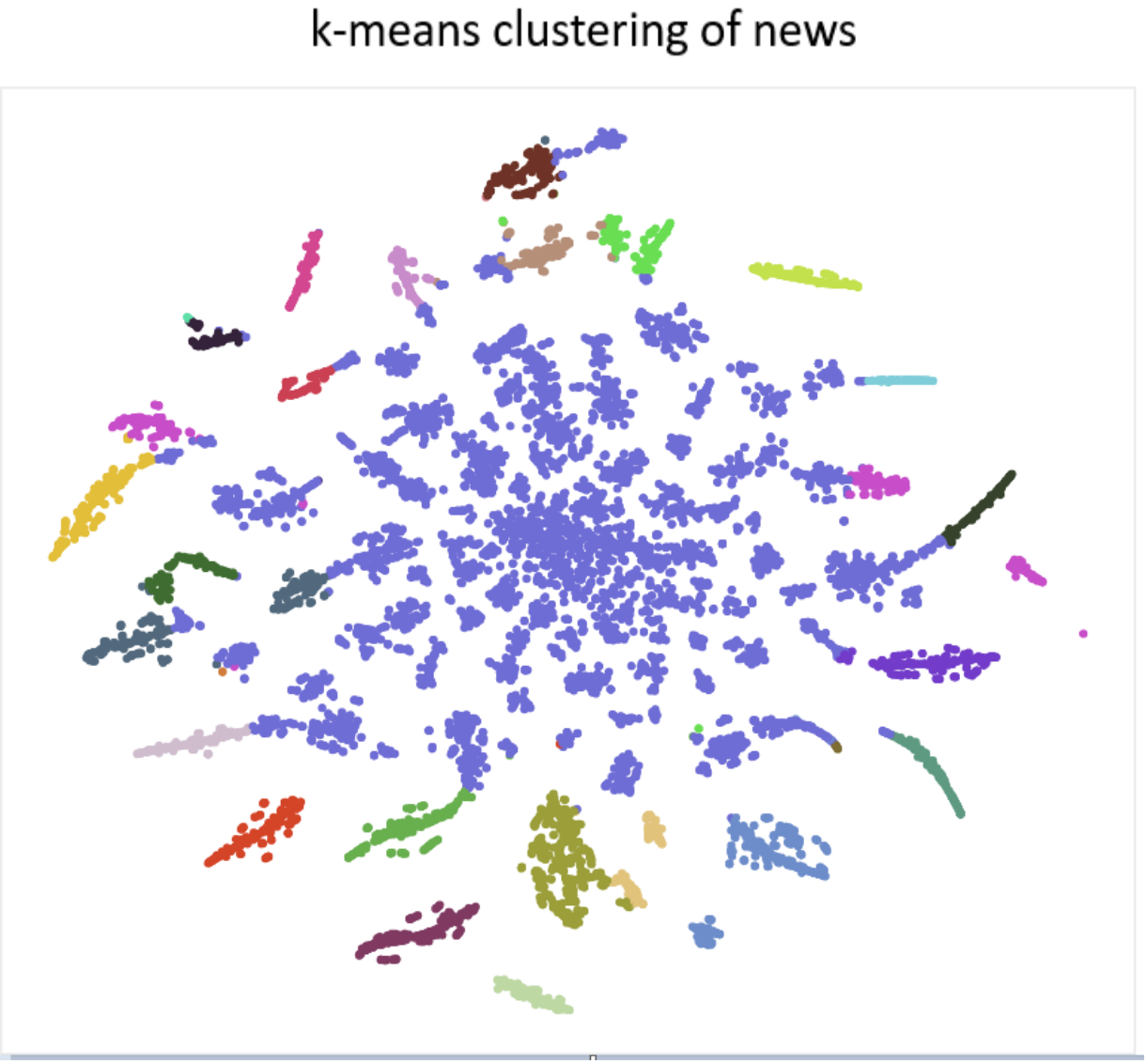

Figure 5.19 *K- Means clusters (K=30) visualized with Bokeh applied on headlines*

5.3.3.2 Latent Dirichlet Allocation (LDA)

The results shown below are results of LDA applied on the subset of 10,000 headlines. There is no dominance of a topic as there was in k-means. The underlined words denote bigrams in the topics. Since both unigrams and bigrams were considered, both appear as our topic words. The topics formed using LDA are shown in Figure 5.20 and the visualization of topics is shown in Figure 5.21.

Topic 0: new study genghis khan genghis khan app know good
Topic 1: titanfall one xbox xbox one microsoft windows live titanfall
Topic 2: new google nsa apps twitter business real new york
Topic 3: bossy ban starbucks campaign app beyonce sandberg oil
Topic 4: young neil neil young music sxsw kickstarter pono player
Topic 5: lindsay lohan lindsay lohan idol show american list american idol
Topic 6: web world woodley shailene shailene woodley wide world wide wide web
Topic 7: flash full costume first look revealed gustin grant gustin
Topic 8: public earth street cell transit stem cell record water
Topic 9: ios apple update chiquita banana new cosmos fyffes
Topic 10: book keith richards keith richards children children book gold rolling
Topic 11: herbalife mt gox gox mt bitcoin drug us company
Topic 12: first strangers kiss video wars star new watch
Topic 13: stocks china data us weak bird flappy lower
Topic 14: test alzheimer blood blood test google predict android new
Topic 15: mars colorado veronica veronica mars million movie veronica marijuana
Topic 16: juan pablo juan pablo bachelor the bachelor galavis pablo galavis
Topic 17: missing carney plane malaysia report boe bank says
Topic 18: snowden sxsw mortgage rates edward edward snowden nsa march
Topic 19: season thrones men big bank trailer jos wearhouse
Topic 20: dunham lena lena dunham day st patrick live night
Topic 21: sprint t mobile softbank us war son use rooney
Topic 22: gold ukraine china high us premier demand tinder
Topic 23: miley cyrus miley cyrus zac efron zac efron chu arthur chu
Topic 24: crush detective candy candy crush ipo maker finale true
Topic 25: tv kim kardashian kim kardashian girls shows sweeney disney
Topic 26: gas prices gas prices cancer us week market rise
Topic 27: gm recall gm recall probe ignition general motors general motors
Topic 28: swift taylor taylor swift keibler stacy stacy keibler billboard money
Topic 29: bieber justin justin bieber selena gomez selena gomez video deposition

Figure 5.20 *Topics extracted using LDA applied on headlines*

Topic 1, Topic 25 and Topic 29 can be noticed as appropriately modelled. They denote news about topics: Microsoft's Titanfall released for Xbox, Kim Kardashian show, Video of Justin Bieber and Selena Gomez respectively. While Topic 13 doesn't give any specific meaning or a topic. It is more generalized.

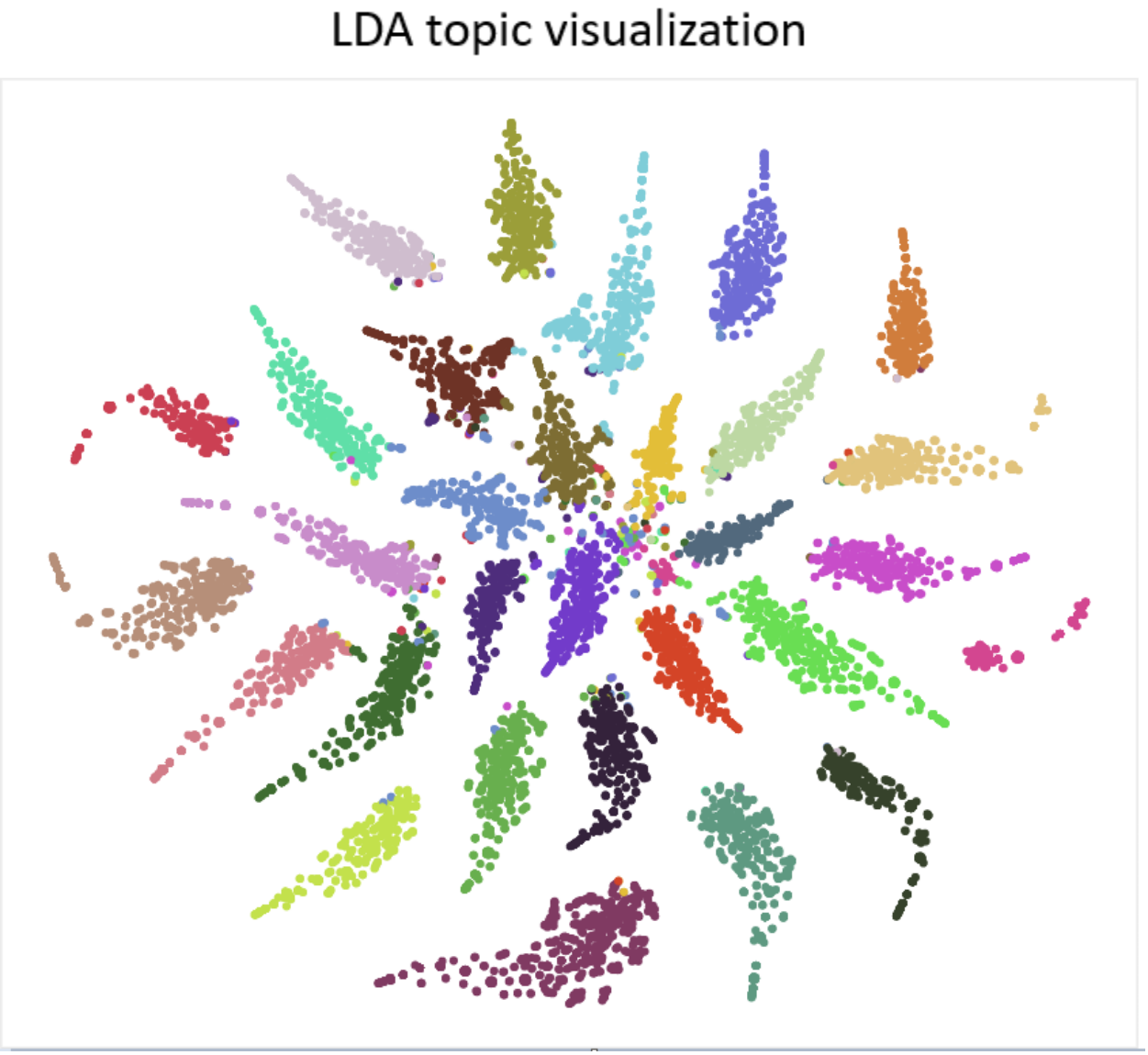

Figure 5.21 *LDA applied on headlines - Topic visualization using Bokeh*

Chapter 6 - Summary and Future Work

6.1 Summary of Results

6.1.1 Classification Results Summary

From several classification experiments conducted on different data sets (headlines and full articles), it can be concluded that with sufficient amount of data and decent number of features(words), high accuracies can be achieved for text classification tasks.

- From the results obtained, Logistic Regression, Linear SVC, multinomial Naïve Bayes performed slightly better than Random Forests with F1-scores of about 0.97, 0.97, 0.97 and 0.95 respectively.
- The order almost pertains similarly for experiments conducted on headlines where, Logistic Regression, Linear SVC, multinomial Naïve Bayes, and Random Forests result in F1-scores of 0.94,0.95,0.93,0.79 respectively. A small margin of 0.01 F1-score higher can be observed for Linear SVC over Logistic regression.
- K-fold Cross validation (K =10) scores are used to reconcile the accuracies achieved for the test set (BBC news articles).
- Results obtained for Random Forests are highly dependent on the parameters considered. For the above reasons, categorization of news articles using Logistic Regression, Linear SVM and multinomial Naïve Bayes, given a proper training set can be a good classification problem. Considering average length of a headline as compared to that of a full document, the accuracies obtained for classification using headline texts are also good.

6.1.2 Clustering Results Summary

Unsupervised clustering technique - k-means performs well for dividing news articles into broad categories and thus can be used to superficially categorize data sets without labels. This is observed from the visualizations plotted using Bokeh library. The keywords with high TF-IDF weights for each cluster are used to recognize different clusters formed. However, clusters formed using LDA(n_topics=30) seem far better than those formed using k-means (K=30). From the visualizations, it is observed that LDA shows better separation of topics and hence can be used as a topic modelling approach in this domain. Headlines in the news aggregator data set are collected from various sources, and hence are more in similar nature (same news can be published by different publishers). This might be one of the reasons for LDA results to show more clear separation of topics, when applied on news aggregator data set. However, quantitative performance measures (which are out of the scope of this project) are probably required before we decide the ideal number of topics for a given set of documents. In this task, the number of topics is assumed to be 30 for clustering tasks based on trial and error methods.

6.2 Future Work

One topic for continued research is to perform better analysis on the outputs of clusters and measuring performance of clustering techniques, to evaluate the obtained outputs. In addition to the current work, Named-Entity Recognition (NER), a subtask of information extraction can be used to classify text into pre-defined categories based on entities associated with the text. This project can be extended to implement NER using NLP libraries so that given a news article as input, entities (persons, places and organizations) that are related to the article are obtained as the output. All the techniques: Supervised classification methods, clustering (topic

modelling) and NER, can be combined to form an application that serves automated text categorization, finding latent topics from documents and retrieve articles related to a person, place or organization. The application can then be good search engine without the need to have a predefined database for all the entities.

Chapter 7 - References

Besbes, A. (2017, March 15). *How to mine newsfeed data and extract interactive insights in Python*. Retrieved from https://ahmedbesbes.com/how-to-mine-newsfeed-data-and-extract-interactive-insights-in-python.html

Contributors, W. (2017, October 31). *Mutinomial Logistic Regression*. (Wikipedia, The free Encyclopedia) Retrieved from https://en.wikipedia.org/w/index.php?title=Multinomial_logistic_regression&oldid=808086228

Dheeru, D., & Karra Taniskidou, E. (2017). *University of California, Irvine Machine Learning Repository.* Retrieved from https://archive.ics.uci.edu/ml/citation_policy.html

Greene, D., & Cunningham, P. (2006). *Insight - BBC Datasets.* Retrieved from http://mlg.ucd.ie/datasets/bbc.html

Hunter, J., Dale, D., Dorettboom, M., & Team, M. D. (2012). Retrieved from https://matplotlib.org/

Joshi, R. (2016, September 9). *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures*. Retrieved from http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

Khan, A., Baharudin, B., & Lee, L. H. (2010). A Review of Machine Learning Algorithms for Text Documents Classification. *JOURNAL OF ADVANCES IN INFORMATION TECHNOLOGY, 1*(1).

Korde, V. (2012). Text Classificaton and Classifiers. *International Journal of Artificial Intelligence & Applications (IJAIA), 3*(2).

Lars, B., Gilles, L., Mathieu, B., Pedregosa, F., Mueller, A., Grisel, O., . . . Holt, B. (2013). *Scikit Learn*. Retrieved from http://scikit-learn.org/stable/

MacQueen, J. (1967). SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS. *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, 1*, 281-297.

Miani, V. (2017, August 19). *Machine Learning for Humans: Unsupervised Learning*. (Medium Corporation) Retrieved from https://medium.com/machine-learning-for-humans/unsupervised-learning-f45587588294

Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning, 12*, 2825-2830.

Raschka, S. (2014, October 4). *Naive Bayes Text Classification*. Retrieved from http://sebastianraschka.com/Articles/2014_naive_bayes_1.html

Roth, D. (2016, October 27). *Multiclass Classification.* Retrieved from http://l2r.cs.uiuc.edu/Teaching/CS446-17/LectureNotesNew/multiclass/main.pdf

Team, B. D. (2014). *Bokeh: Python library for interactive visualization*. Retrieved from http://www.bokeh.pydata.org

Waldron, M. (2015, January 20). *Text Analysis 101; A basic understanding for Business Users: Topic Modelling*. (AYLIEN) Retrieved from http://blog.aylien.com/text-analysis-101-a-basic-understanding-for/

Waskom, M. (2012-2017). *Seaborn: Statistical Data Visualization*. Retrieved from https://seaborn.pydata.org/

Weisstein, E. (2018, April 18). *Box-and-Whisker Plot*. Retrieved from Wolfram MathWorld: http://mathworld.wolfram.com/Box-and-WhiskerPlot.html

Wikipedia. (2018, March 12). *Google Directory*. Retrieved from Wikipedia, the free encyclopedia: https://en.wikipedia.org/wiki/Google_Directory