Final Project Report

IST736 Text Mining

Farahin Choudhury

**Introduction:**

The vast domain of sports analytics lends itself to a lot of possibilities for exploration and extracting even more information about sports themselves. Commentary on sites social media sites like Facebook and YouTube reveal even more about user behavior when interactive with content related to sports. YouTube is a site where so many different channels exist of different areas of interest and entertainment, one of which is sports. There are a whole host of sports-related channels on YouTube where users can interact via commentary on any given video, which will be the focus for this analysis. Text mining with comments on YouTube is a fascinating avenue for exploration and building models on, specifically as classification tasks. Using Python and its various libraries within the sci-kit learn module is a great way to find meaning in YouTube comments and find trends in the kinds of comments users make. Sports analytics is not only interesting; this domain makes text mining possible and applying algorithms necessary to extract and retrieve meaning and context

**Project idea:**

Having a dataset with over 37,000 observations across more than 10,000 YouTube channels could make the analysis less robust since the channels are all not sports related. While developing the idea for this project, exploring the dataset lead to discovering the occurrences of each individual channel, the top 5 all being sports channels. This concluded in narrowing down the data to just the data for those 5 channels. The channels with the data for this analysis are as follows: Sky Sports Football, The United Stand, BT Sport, NBA and NFL.

**Objectives:**

Now that the data has been narrowed down to only containing observations for the aforementioned YouTube channels, the goals and objective can be defined. This analysis can go in many different directions since sports analytics is so vast. In the interest of simplifying the analysis, the objectives will be achieving text classification. Unsupervised and supervised learning methods will be applied to uncover themes, by way of Latent Dirichlet Allocation (or LDA for short) or clustering. Term frequency vectorization is a necessary aspect of text mining and sentiment analysis to find common terms. An attempt will be made to find common tags and words in comments to achieve these objectives.

**Data collection and description:**

The data from this project was acquired from Kaggle and it was downloaded as a .csv file. The original dataset had over 37,000 observations but after refining the objectives and areas of analysis, the data was pared down to include only data from the sports channels. The number of observations dropped to about 1,200 which is a more manageable yet adequate amount of data needed for this analysis. The data in this dataset was quite detailed, especially considering that one of the columns in the dataset included the number of dislikes on a given video. This is an important detail because YouTube removed the dislike count feature on its platform, so having this additional information is valuable and uncovers trends about what viewers enjoy watching on YouTube.

The focus will primarily be on the tags and the comments. That is where the content for these videos lie and those data values will be the source for the vectorization. Simply reading through the comments and associated tags will provide an idea of the content of each video, but it should be noted that, according to the description of the data in Kaggle, the data was extracted via an API and there are 20 comments provided per video. These comments are lumped together in the ‘comments’ column of the data set but that will not lessen the data exploration and vectorization tasks for this analysis.

**Modules and algorithms:**

This section will go into detail about the process of text classification and which algorithms were eventually applied to achieve the objectives. This will also cover the steps taken to arrive at those algorithms. Some python code will be provided to further explain certain methods for data preparation and how the algorithms were built.

It was imperative to incorporate the NLTK and sci-kit modules as there is a lot of text to analyze and extract meaning from. In order to test out and further explore text data, there are libraries within each of these python modules that make classification and categorization possible. The regular expression module ‘re’ was also consulted to list out words from the ‘comments’ field in the data set. The final consensus was to use the following libraries within sci-kit learn: CountVectorizer, TfidfVectorizer, train\_test\_split and LatentDirichletAllocation.

An important note to make is that this data set did not come with data regarding sentiment or polarity. While that data could have been manually entered in, that process would not be informed since the polarity can be subjective due to how the comments can be perceived, and no two individuals would assign the same polarity to a given comment (especially in this data set). In the interest of time and maintaining a neutral data set, which might be tricky considering sports can sometimes be a polarizing area of interest, the topic modeling algorithm LDA was applied. This lends itself to be an ideal model for any type of corpus; it is evident that the model can work on different file types, as the data set is not a traditional corpus.

Along with the LDA algorithm, a function was created to show a list of the topics given the specific model that was created with this algorithm. The results will be explained in the following section on what parameters were used and how the data was vectorized. This is critical to point out because different vectorizers will yield different results.

**Results:**

After initializing and refining the algorithms for this analysis, it became clear that the algorithm that would work best for the given data was the Latent Dirichlet Allocation algorithm (LDA). Topic modeling in LDA is a viable method in learning about the data and as previously mentioned, there is a lot within sports analytics that can reveal what users are having conversations about (regardless of athletes, sports, statistics, etc.). In addition, the preferred vectorization for this data was to use the term-frequency vectorizer ‘tf-idf’ in keeping honest with the objective of finding common terms in the tags and comments.

The train and test data comprised of the comments and tags, respectively. While 1,200 is an adequate number of observations, the test size was kept small at 20% (0.2). The majority of the data analyzed was in the X\_train split, which contained the comments. The train\_test\_split function in sci-kit learn makes splitting the data easy with minimal code.

A close-up of a map

Description automatically generated with low confidence

The above image is an example of a word cloud that was produced using the WordCloud module in Python. This is to show the terms that occur within the data, and in this example there was a subset of the training data used as the input to produce this graphic. The coloring of the words is arbitrary, but the size of the words actually has a purpose. The bigger the word in the word cloud, the more often the word occurs in the text. Two of the biggest words that appear are “Chelsea”, which is the name of a soccer club originated in the UK and “Ziyech”, who is a soccer player. This indicates that several of the comments contain these two words, which explains why the channels like BT Sports and Sky Sports Football have so many observations in the data set. Using a different subset of the training data would yield a different result with different words appearing in the word cloud, but this is a good demonstration of term frequency and will help to uncover themes in sports.

Having used the tf-idf vectorizer with this data, it is imperative to report on some of the vocabulary in the output of the vectorization. Some of these keep with the soccer theme, such as ‘englishman’ and ‘penalty’. The term ‘englishman’ could pertain to any sport, but it is largely associated with European topics, soccer being a big one and anecdotally a large part of life throughout the UK (where an ‘englishman’ would hail from). The vectorized term frequency is given as 1,302 for ‘englishman’ while the frequency for ‘penalty’ is 2,761. These are interesting metrics that lend themselves as terms that might appear in the LDA model.

Text

Description automatically generated 

The list of words in the above screenshots are the ranked terms from the vectorized data. More specifically, the feature names were extracted from the tf-idf vectorizer. The numbers provided next to each feature name shows the topic distribution. They seem like small figures but they are demonstrating the feature rankings. In the vectorizer, it’s important to include the stop words parameter in order to exclude those words from the analysis, as they provide no context. It is not a coincidence that the first term in this list is ‘game’, which is a term associated with playing any sport. The terms span across different sports but they are all indicative of the overarching theme of sports.

Text

Description automatically generated

The above is a culmination of the topic modeling task in LDA. The LatentDirichletAllocation command takes in the vectorized data along with other parameters, such as the number of components (which is the number of topics) and the number of iterations. There is a display\_topics function that was created from scratch that takes in the model, feature names and the total number of words per topic as the parameters. The LDA algorithm displays the terms in each of the above topics to decipher one topic from another. Although it may seem that the words do not provide meaning that makes sense to the average person, understanding the data and having done this analysis will show that this algorithm does work.

One of the above topics to note is ‘Topic 11’ which names a few athletes who both play soccer. Other words that would indicate that this topic pertains to soccer are ‘united’ and ‘scotland’ as they are clubs and locations where soccer is played competitively. The same can be said for ‘topic 10’, which pertains to basketball since it provides the names of three different basketball players in different NBA teams: ‘giannis’, ‘lebron’ and ‘tatum’. The other terms in this specific topic are in fact basketball related, but they don’t specifically pertain to the different players mentioned in that topic.

Lastly, the LDA model is fit and transformed using the vectorized data. This provides another distribution of the topics, with the LDA model as the only parameter. With this data, most of the topics showed a distribution of 0.00560355. This number was calculated in the backend of the LDA algorithm, which is a strength in machine learning and text mining.

**Conclusion:**

Classification in text mining with python is a valuable skill which takes time to fine-tune. While LDA ended up being the model for this analysis, the model is only as good as the data being provided to do the analysis on. The topics definitely could have been more cohesive and perhaps the vectorization could have included more comments in order to extract more meaning and thus overall better topic modeling. If sentiment were assigned to each individual comment, that would have added an additional layer of complexity to this analysis, but sentiment is not a necessary asset for topic modeling. For further exploration, the topic modeling could be explored using the genism module and compared to LDA. Sports analytics leaves much room for exploration and makes the process challenging but promising.