Spectrum Labs Project

Task 1: Extraction/Cleaning Data

For this task, I request the .json file from the 4chan site through https GET Requests. The fetched .json string is converted to .json python object type for further pre-processing steps. For this project, I fetch the top 100 threads and the comments for those of the four channels:
1)Literature
2)Technology
3)Food & Cooking
4)TV & Film.
Pre-processing tasks involved:

1) Removal of HTML Tags
2) Removing post number from each comment
3) Removal of Encoding formatting texts
4) Collectively store all comments from multiple posts in a list for further analysis

Task 2: Exploratory Data Analysis

This task involves the analysis of comments in each channel which shall help us understand the kind of data we are handling. I perform 4 tasks for this purpose:

1. Average Posts per Thread for each channel
2. Average Length of each post for each channel
3. Top 100 Frequent Words for each channel
4. Sentiment Analysis for Each channel

Results:

1. Highest Average number of posts was achieved by the Technology channel which suggests that most users in 4chan channel would like to engage more in content regarding technology than the rest of the channel
2. Highest Average Length of posts was achieved by Literature Channel. This implies users of the literature channel would like to express their opinion in a more elaborate manner than the rest of the channel.
3. Most of the frequent words in all channels were common words. I observe that users of the 4chan website use a lot of cuss words to express their opinion. It would be interesting to see the Sentiment of posts for each channel.
4. All the channels achieved a very low average sentiment polarity scores. The highest polarity score achieved was 0.0506 by the technology channel and the lowest was achieved by the Food and Cooking channel with a score of 0.029. The scale of polarity scores range from -1 to 1. This concludes that users on average are neutral in

expressing their opinion. However, It is highly that this case is due to the extreme polarity of positive and negative posts.

Task 3: Discriminative Analysis

This task is performed to understand the key features of each post that makes it unique for each particular channel. For this task, I explored the keywords that were more frequent in posts of each channel. For this purpose, I pick the most frequent words of each channel that were not present in the most frequent list of other channels. By this method, We can specifically identify the words that are unique features of that channel.

Results:

1. Some of the most frequent words used in the Literature channel were [Literature, existence, philosophy, argument, God]. These words are inferred to be the core essence of the information in the literature channel. This finding shall help us better build models in our future tasks.
2. Some of the most frequent words used in the Technology channel were [Windows, Computer, Linux, Learn, CPU, Desktop]. These words infer that most of the discussion on technology in the 4chan website were based on operating system and computers.
3. Some of the most frequent words used in the Food & Cooking channel were ['food', 'eat', 'beer', 'water', 'tea', 'meat', 'cheese', 'drink', 'chicken']. An interesting further exploration on this channel would be to analyze the meal of discussion. Given the most frequent word list of this channel. I hypothesize that the discussion on this channel is mainly on Dinner food,
4. Some of the most frequent words used in the TV & Film channel were 'movie', 'film', 'show', 'watch', 'movies', 'episode', 'scene', 'character']. This channel did not have specific words within the film or TV. This suggests that most discussions on this channel were more general than other channels.

Task 4: Modeling

For this task we use bag-of-words for feature selection and further build a multinomial Naive Bayes classifier to classify the posts into channels. We split the train and test set into 7:3 ratio. We use 1-gram models for bag-of-words selection. Later, I would iterate the model for multigram models and further use sophisticated word-to-vector models that capture the essence in the texts better.

Results:

This primitive model achieved an accuracy of 68% on the test set. This accuracy metrics convey that usage of words as features does pretty ok job in classification. Given that the number of classes for this model is 4. This accuracy percent is not too discouraging. However, This calls for more sophisticated feature engineering methods to capture the essence in the posts. Another trivial reason for this accuracy score is that most comments were short and included common words(words that dont distinguish channels). The next step would be to explore with gradient-descent logistic regression models. My intuition to build the best model for this task would be a model that can capture contexts between two posts within the same channel and build word embedding around this essence.