AG News Topic Classification

CSML 1010 - Winter 2020 - Group 20

Tony Lee, Viswesh Krishnamurthy

PROBLEM SELECTION & DEFINITION

- A text classification problem was chosen from the website https://datasets.quantumstat.com/. We chose the AG News corpus dataset

- The goal of this project is to develop a text classifier model that can accept a news 'headline' and 'content' of the news to classify the news article into one of the 4 following categories
  - World (coded as 1)
  - Sports (2)
  - Business (3)
  - Sci/Tech (4)

BACKGROUND OF THE DATASET

- AG is a collection of more than 1 million news articles. News articles have been gathered from more than 2000 news sources by ComeToMyHead in more than 1 year of activity. ComeToMyHead is an academic news search engine which has been running since July, 2004. The dataset is provided by the academic community for research purposes in data mining (clustering, classification, etc), information retrieval (ranking, search, etc), xml, data compression, data streaming, and any other non-commercial activity. For more information, please refer to the link http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

- The AG's news topic classification dataset is constructed by Xiang Zhang (xiang.zhang@nyu.edu) from the dataset above. It is used as a text classification benchmark in the following paper: Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level Convolutional Networks for Text Classification. Advances in Neural Information Processing Systems 28 (NIPS 2015).

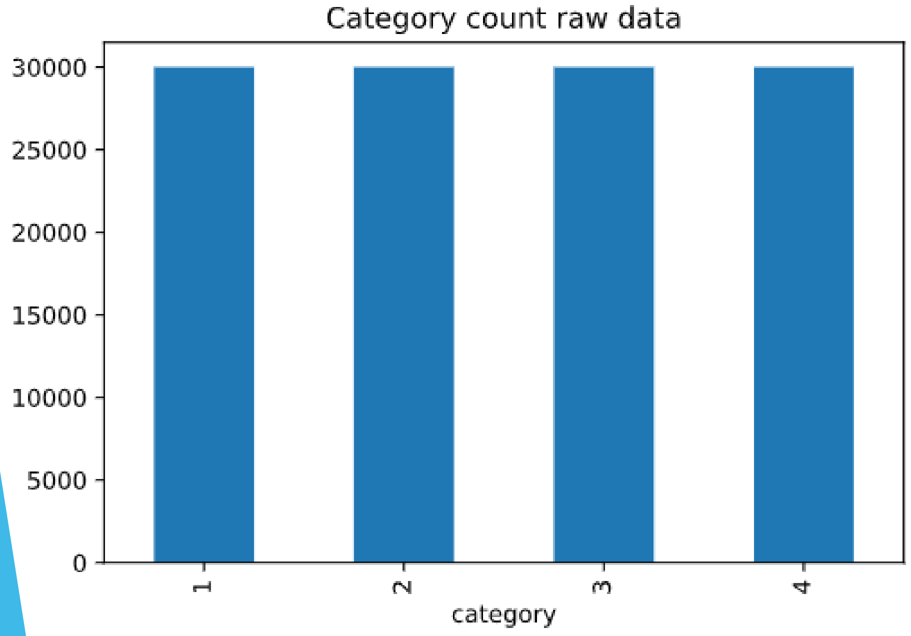
FEATURE ENGINEERING & MODELS - PLAN

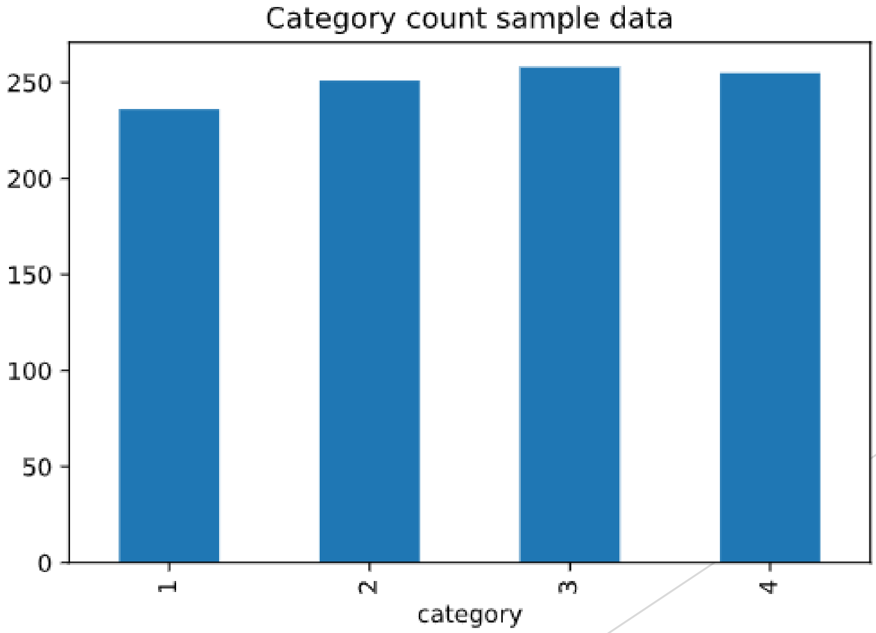
- We will perform feature engineering using the following methods
  - Bag of words
  - Bag of n-grams
  - Tfidf
  - Glove

- We will use the following models to perform classification
  - Logistic regression
  - Decision trees
  - SVM
  - Naive Bayes

DATA CLEANING

- The raw data was explored to identify whether it required balancing. It was found that there were equal number of 'headlines' and 'content' for each category. (Each category has 30,000)
- We started with a sample of 1000 rows from the raw data. The sample data was pretty balanced as well





DATA CLEANING

- Header: The raw data did not contain headers for the columns. The columns were appropriately named

```
train_data.columns = ['category', 'headline', 'content']
```

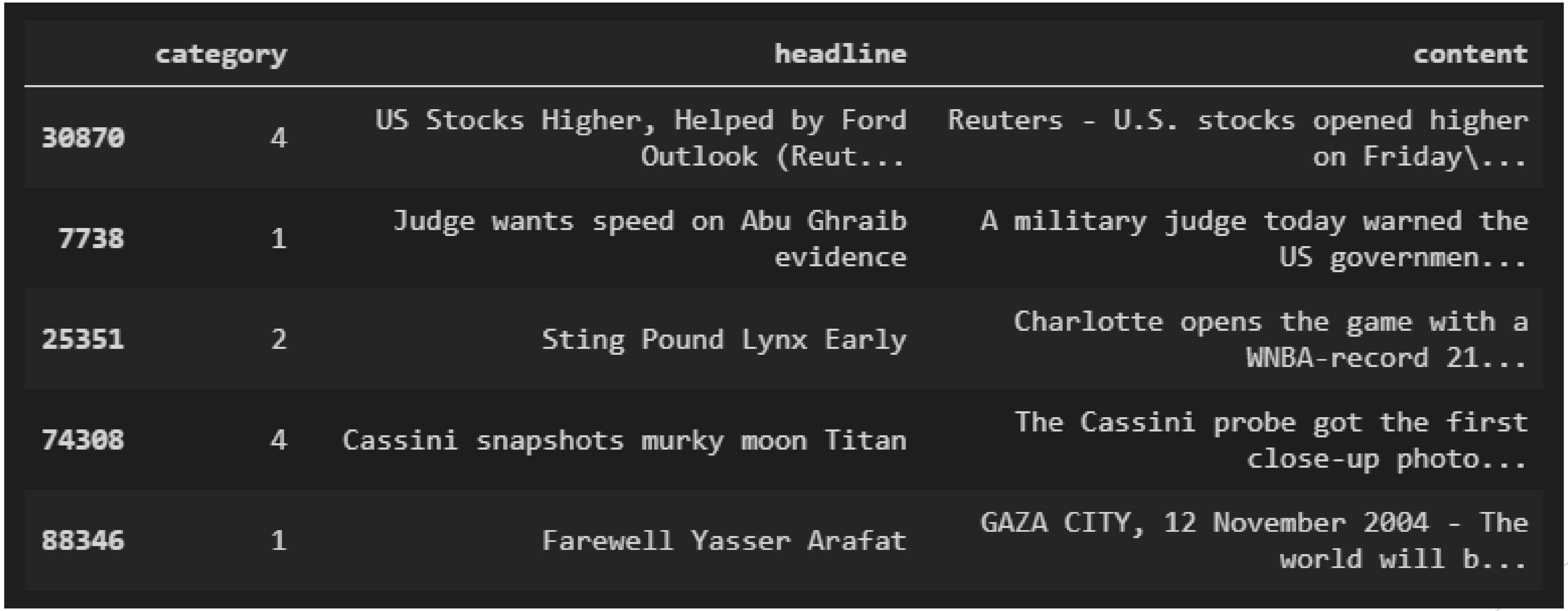
- HTML Code: There were some HTML code found in the raw data, like “&gt” and “&lt” and were removed
- News Sources: Some of the rows in the ‘headline’ column contained the news source like (Reuters), (AP) etc. These were removed as they don’t contribute much to the overall classification problem

```
import re
def clean(x):
    x = re.sub(r'(&[A-Za-z]+)|\(.*\)', '', x)
    return str(x)
for i, row in train_data_sample.iterrows():
    train_data_sample.at[i, "headline"] = clean(row.headline)
```

DATA EXPLORATION

- We will take a look at the top few rows of the dataset

```
train_data_sample.head()
```



	category	headline	content
30870	4	US Stocks Higher, Helped by Ford Outlook (Reut...	Reuters - U.S. stocks opened higher on Friday\...
7738	1	Judge wants speed on Abu Ghraib evidence	A military judge today warned the US governmen...
25351	2	Sting Pound Lynx Early	Charlotte opens the game with a WNBA-record 21...
74308	4	Cassini snapshots murky moon Titan	The Cassini probe got the first close-up photo...
88346	1	Farewell Yasser Arafat	GAZA CITY, 12 November 2004 - The world will b...

DATA EXPLORATION

- To better understand the data, we built a “Countvectorizer” with a minimum document frequency of 2 and included regex commands to discard numbers, symbols & special characters.
- For now, we chose to build a ‘unigram’
- We produced a bag of words dataframe, cv_matrix_df

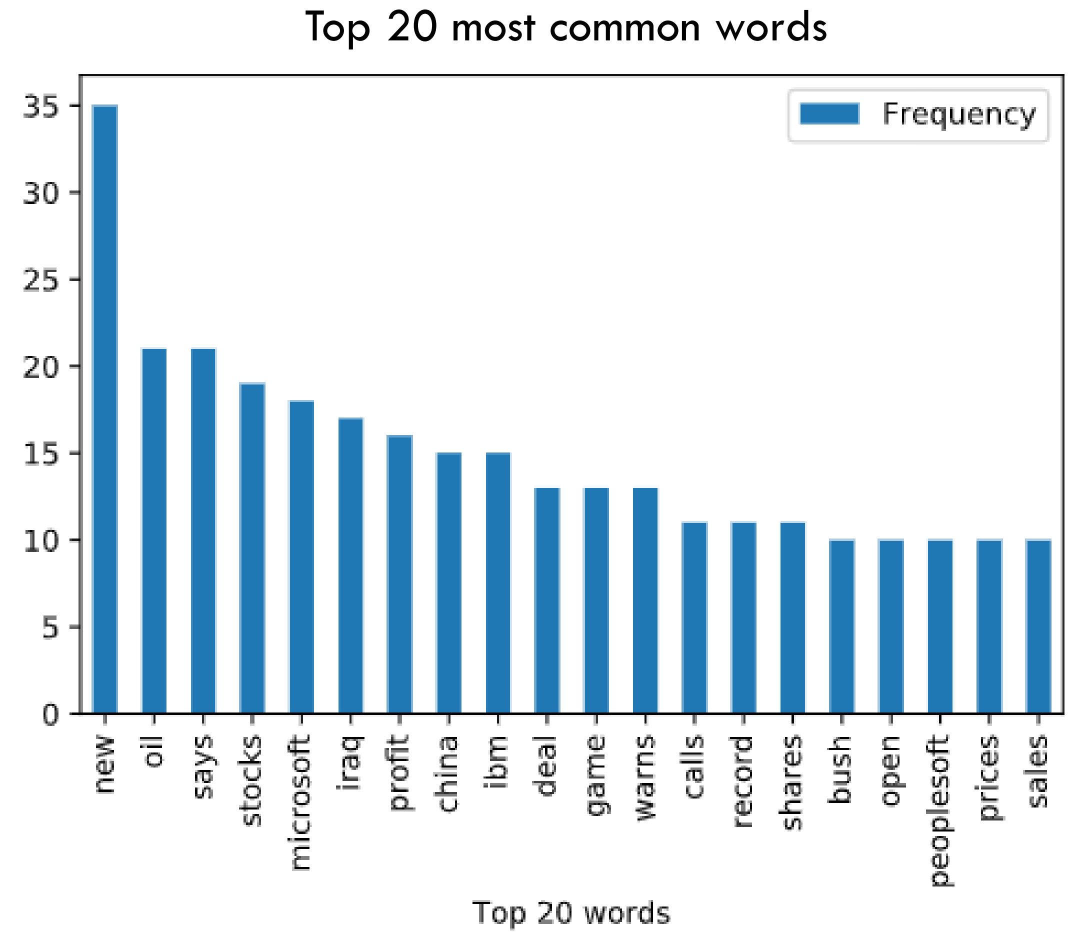
```
# create a CountVectorizer from raw data, with options to clean it
cv = CountVectorizer(min_df = 2, lowercase = True, token_pattern=r'(?u)\b[A-Za-z]{2,}\b',
                     strip_accents = 'ascii', ngram_range = (1, 1),
                     stop_words = 'english')
cv_matrix = cv.fit_transform(train_data_sample.headline).toarray()

# get all unique words in the corpus
vocab = cv.get_feature_names()

# produce a dataframe including the feature names
cv_matrix_df = pandas.DataFrame(cv_matrix, columns=vocab)
```

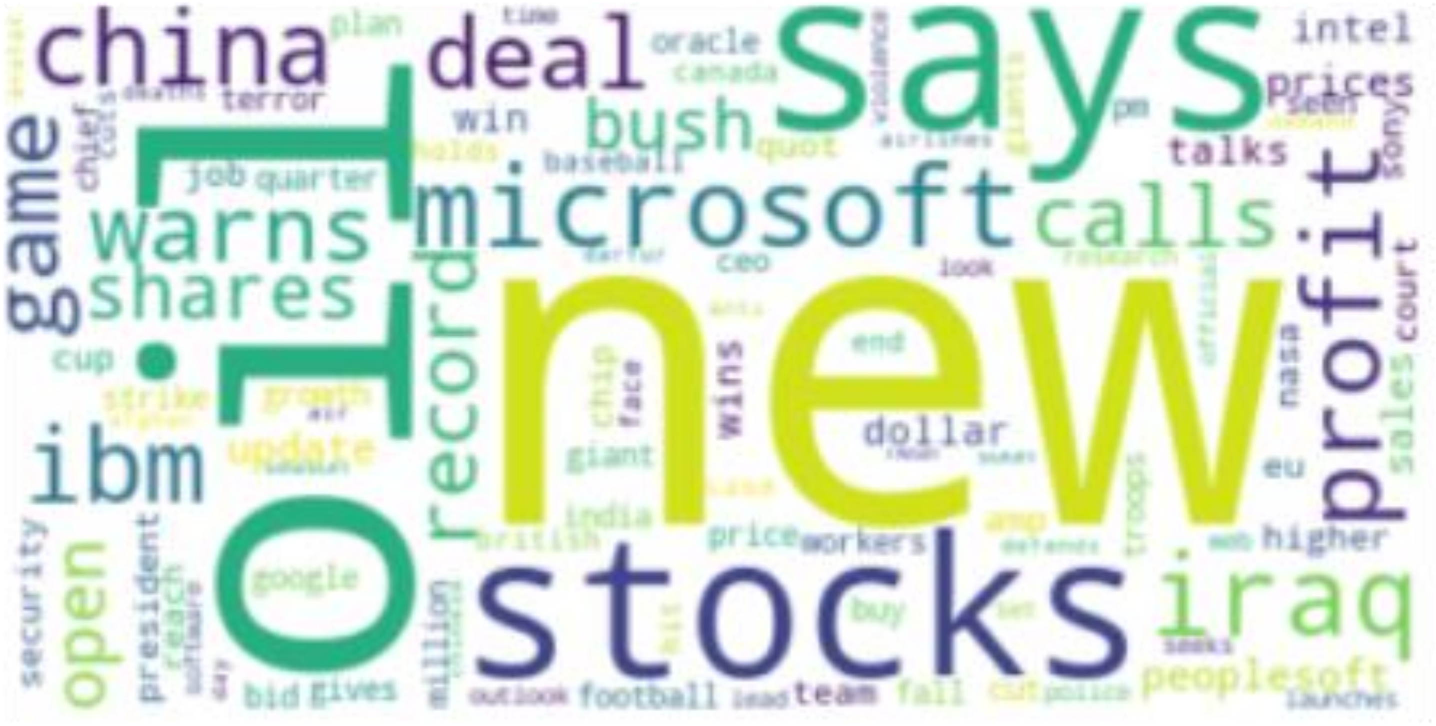
DATA EXPLORATION

- Further manipulating the data, we produced a bar chart of the Top-20 most common words of the headlines of the news articles



DATA EXPLORATION

- We also produced a word cloud of the headlines



This marks the initial proposal stage of the project. As we progress, we will iterate through the ML lifecycle and the feature engineering and data exploration steps may accordingly change