https://colab.research.google.com/drive/1U08_kOqOobGDXRw6-gzoqnCx16k7yAwe?usp=sharing

Answers to Assignment4 by Chan Yu and Naihe Xiao

Work breakdown: Part I is done jointly, coding of Part II is mostly done by Chan, while presentation and data report is mostly done by Naihe.

Set up library

Show code

Part I: Sentiment Analysis with a Twitter Dataset

Read data

Show code

	Unnamed: 0	OriginalTweet	Sentiment
0	0	TRENDING: New Yorkers encounter empty supermar...	0
1	1	When I couldn't find hand sanitizer at Fred Me...	2
2	2	Find out how you can protect yourself and love...	2
3	3	#Panic buying hits #NewYork City as anxious sh...	0
4	4	#toiletpaper #dunnypaper #coronavirus #coronav...	1

A. Balance of training data

Compute proportion of observations

Show code

```
Proportion of sentiment == 0: 37.4 %
Proportion of sentiment == 1: 18.7 %
Proportion of sentiment == 2: 43.8 %
```

B.Tokenize

Split by white space

Show code

	Unnamed: 0	OriginalTweet	Sentiment	tokens
0	0	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	1	[@MeNyrbie, @Phil_Gahan, @Chrisitv, https://t....
1	1	advice Talk to your neighbours family to excha...	2	[advice, Talk, to, your, neighbours, family, t...
2	2	Coronavirus Australia: Woolworths to give elde...	2	[Coronavirus, Australia:, Woolworths, to, give...

C.Remove URL tokens

Remove URL by RE 'http.+'

Show code

	Unnamed: 0	OriginalTweet	Sentiment	tokens
0	0	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	1	[@MeNyrbie, @Phil_Gahan, @Chrisitv, and, and]
1	1	advice Talk to your neighbours family to excha...	2	[advice, Talk, to, your, neighbours, family, t...
2	2	Coronavirus Australia: Woolworths to give elde...	2	[Coronavirus, Australia:, Woolworths, to, give...

D. Remove punctuation and turn to lowercase

Remove punctuation and convert to lowercase

Show code

	Unnamed: 0	OriginalTweet	Sentiment	tokens	lowercase_tokens	tokens_no_punct
0	0	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	1	[@MeNyrbie, @Phil_Gahan, @Chrisitv, and, and]	[@menyrbie, @phil_gahan, @chrisitv, and, and]	[menyrbie, phil_gahan, chrisitv, and, and]

I might want to keep the punctuation @ if I want to analyze the frequency of the appearance of the tags.

E. Stemming

Woolworths to		Woolworths,	woolworths, to,	woolworths, to,	

Stem with PorterStemmer

Show code

	Unnamed: 0	OriginalTweet	Sentiment	tokens	lowercase_tokens	tokens_no_punct
0	0	TRENDING: New Yorkers encounter empty supermar...	0	[TRENDING:, New, Yorkers, encounter, empty, su...	[trending:, new, yorkers, encounter, empty, su...	[trending, new, yorkers, encounter, empty, sup...
1	1	When I couldn't find hand sanitizer at Fred Me...	2	[When, I, couldn't, find, hand, sanitizer, at,...	[when, i, couldn't, find, hand, sanitizer, at,...	[when, i, couldnt, find, hand, sanitizer, at, ...
		Find out how you can protect		[Find, out, how, you	[find, out, how, you,	[find, out, how,

F. Remove stopwords

Remove sw

Show code

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

	Unnamed: 0	OriginalTweet	Sentiment	tokens	lowercase_tokens	tokens_no_punct
0	0	TRENDING: New Yorkers encounter empty supermar...	0	[TRENDING:, New, Yorkers, encounter, empty, su...	[trending:, new, yorkers, encounter, empty, su...	[trending, new, yorkers, encounter, empty, sup...
1	1	When I couldn't find hand sanitizer at Fred Me...	2	[When, I, couldn't, find, hand, sanitizer, at,...	[when, i, couldn't, find, hand, sanitizer, at,...	[when, i, couldnt, find, hand, sanitizer, at, ...
2	2	yourself and	2	can protect	can, protect,	you, can, protect,

G. Convert to vectors of word counts

Vectorization with CountVectorizer

Show code

```
Share of X_train (41155, 1000)
```

H. Naive Bayes

Convert data type to numeric

Show code

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  after removing the cwd from sys.path.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  """
```

Build model

Show code

```
Train accuracy: 0.68
Train error: 0.32
```

```
Test accuracy: 0.44
Test error: 0.56
```

5 most probable words in each class

Show code

```
Most probable words of class positive: [('coronaviru', 6703), ('covid19', 4865), ('price
Most probable words of class neutral: [('coronaviru', 3792), ('covid19', 2751), ('store
Most probable words of class negative: [('coronaviru', 7466), ('covid19', 6001), ('store
```

I. ROC fitting

It is appropriate to fit a ROC curve in this scenario, since the data is not severely imbalanced, hence each data point will not make a big difference, and therefore ROC curve can capture valuable information including the behavior of the model at different thresholds.

J. TF-IDF

Vectorization with TF-IDF

Show code

```
Train accuracy with TF-IDF: 0.66
Test accuracy with TF-IDF: 0.45
```

The train accuracy decrease and test accuracy increases compared to the previous approach.

K.Redo

Build model with lemmatizer and TF-IDF

Show code

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Train accuracy with lemmatization & TF-IDF: 0.66
Test accuracy with lemmatization & TF-IDF: 0.42
```

The test accuracy decreases compared to the previous two attempts.

Part II: Having fun with NLP using the Twitter API

(1) Extracting data

Setup Tweepy token

Show code

Download data through Tweepy API

Show code

Pull data that stroed in Github

Show code

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   OriginalTweet  700 non-null    object
 1   Attitude       700 non-null    int64 
 2   Emotion        700 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 16.5+ KB
None
```

(2) Exploratory Data Analysis

Compute proportion of observations

Show code

```
Proportion of Attitude == 0: 53.9 %
Proportion of Attitude == 1: 46.1 %
```

Display data

Show code

```
(700, 3)
```

	OriginalTweet	Attitude	Emotion
0	b"@FLbeachrealtor @johnpilger NATO doesn't wan...	0	0
1	b'@kfc Financing the Russian market means supp...	1	1
2	b'These men may be "heroes" in Russia, but the...	0	0
3	b'@RussianEmbassy @mfa_russia @RusembUkraine @...	1	0
4	b'#NGO = #DARKNESS! #RUSSIA/#PUTIN is fighting...	0	0

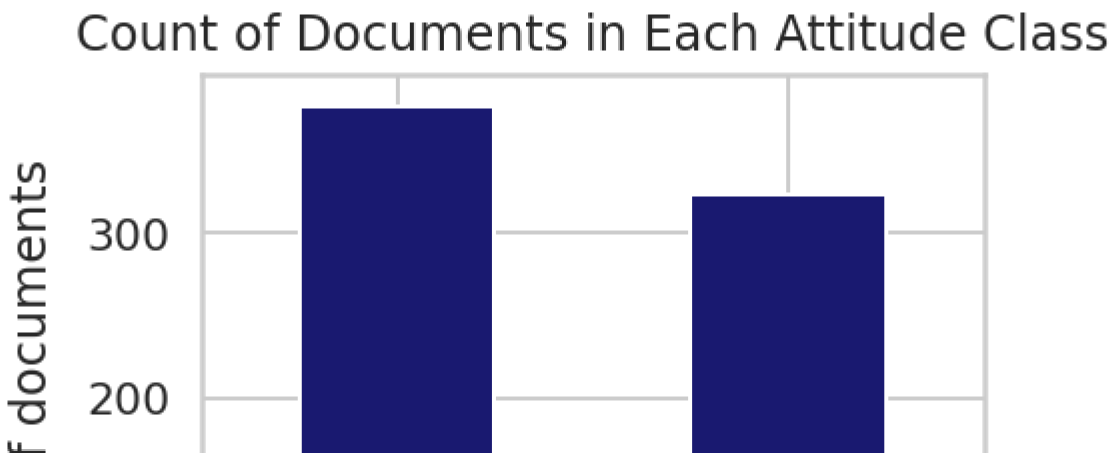
Comparison of emotion scores in two groups

Show code

	Attitude								Emotion			
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min
attitude_cat												
Neutral, unsupportive, unclear	377.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	377.0	0.196286	0.397716	0.0

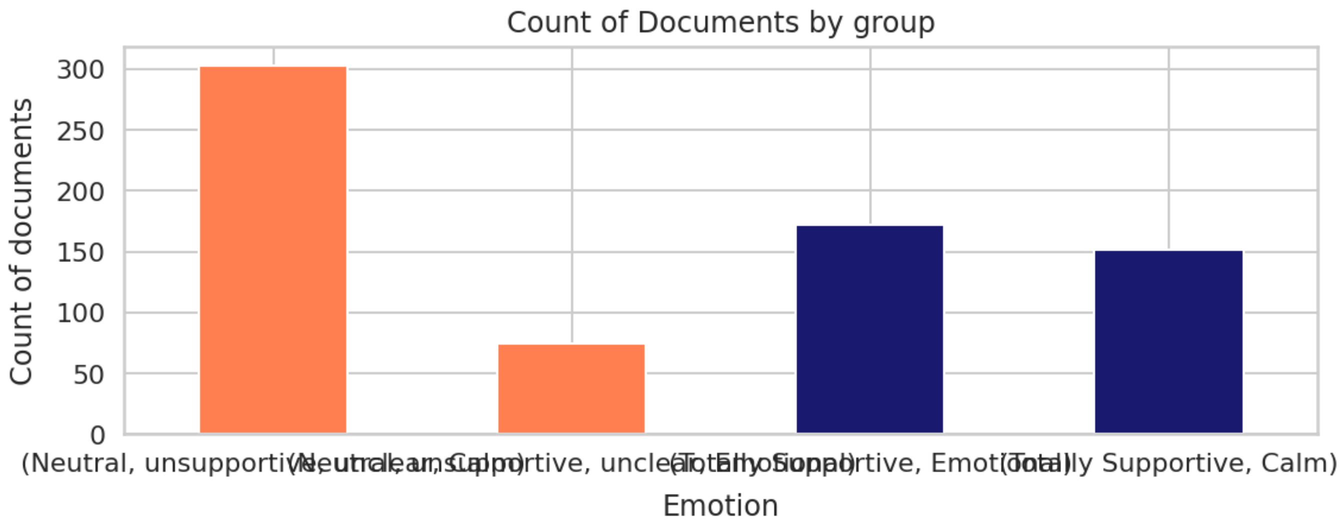
Basic Plot

Show code



Basic Plot

Show code



Value counts

Show code

```
Neutral Attitude: #of calm documents 303
Neutral Attitude: #of emotional documents 74
Supportive Attitude: #of calm documents 151
Supportive Attitude: #of emotional documents 172
```

Set up library

Show code

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Functions

Show code

Preprocessing

Show code

(3) Building model

Model

Show code

```
Training accuracy:  0.89
Testing accuracy: 0.6
```

Comparison observation counts in two groups

Show code

```
Train (total 560 ): counts for Supportive = 268 vs. Neutral/Unclear = 292
Test (total 140 ): counts for Supportive = 55 vs. Neutral/Unclear = 85
```

(4) Evaluate model

Predict probability

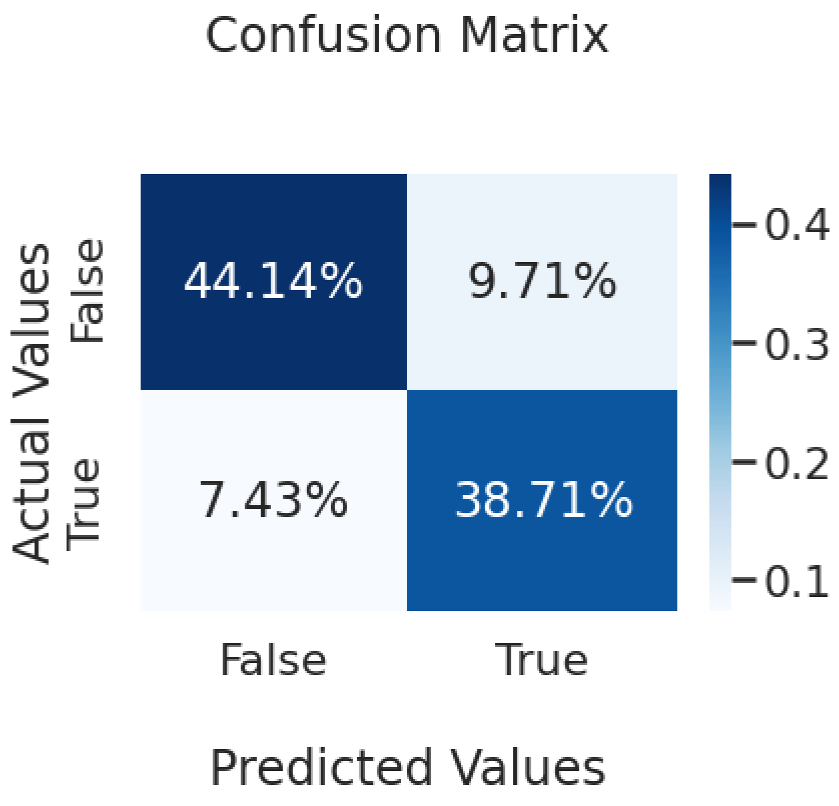
Show code

	OriginalTweet	Attitude	Emotion	attitude_cat	emo_cat	token	att
116	b'Gruesome evidence points to #WarCrimes made...	0	0	Neutral, unsupportive, unclear	Calm	[bgruesom, evid, point, warcrim, made, russia,...	
147	b'@RussianEmbassy @mfa_russia @RusembUkraine @...	1	1	Totally Supportive	Emotional	[brussianembassi, mfa_russia, rusembukrain, ru...	
	Ukraine war: Russia			Neutral.		[ukrain, war,	

Confution matrix

Show code

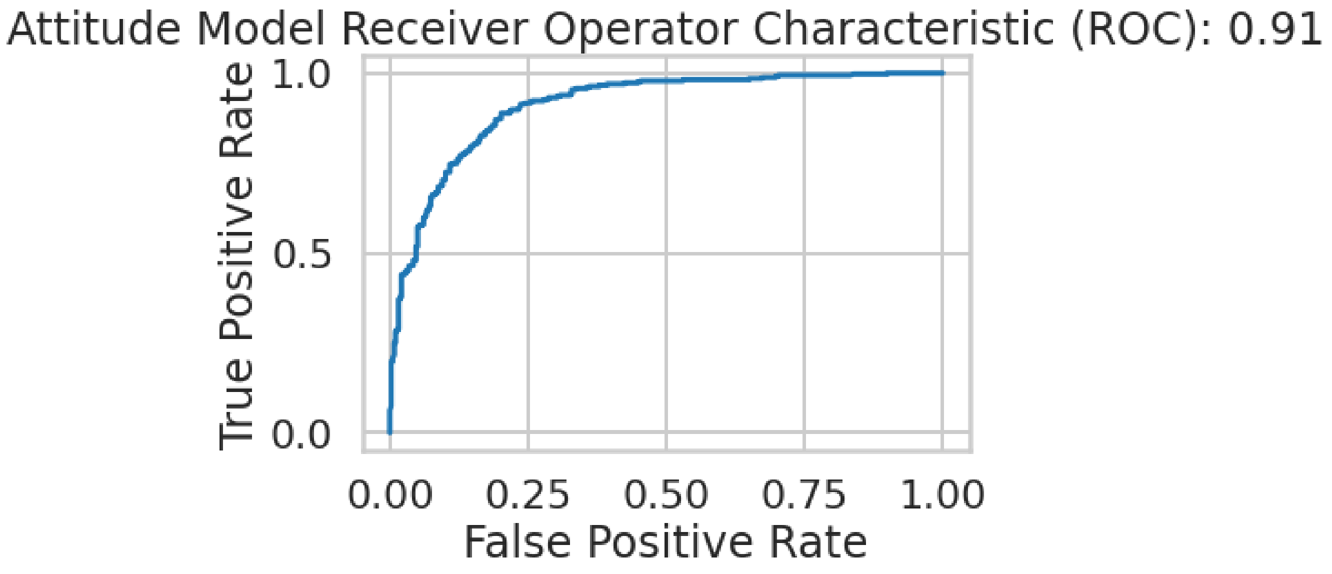
```
Attitude model
Accuracy: 0.83
True positives: 271
True negatives: 309
False positives: 68
False negatives: 52
Confusion matrix
[[309  68]
 [ 52 271]]
```



```
Sensitivity (Recall): 0.84
Specificity: 0.82
Precision: 0.8
```

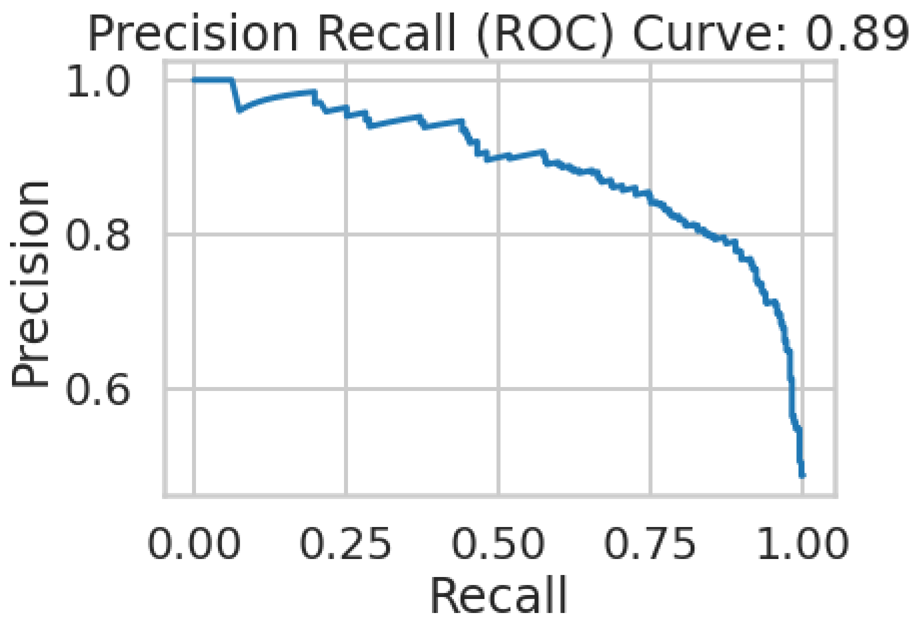
AUROC

Show code



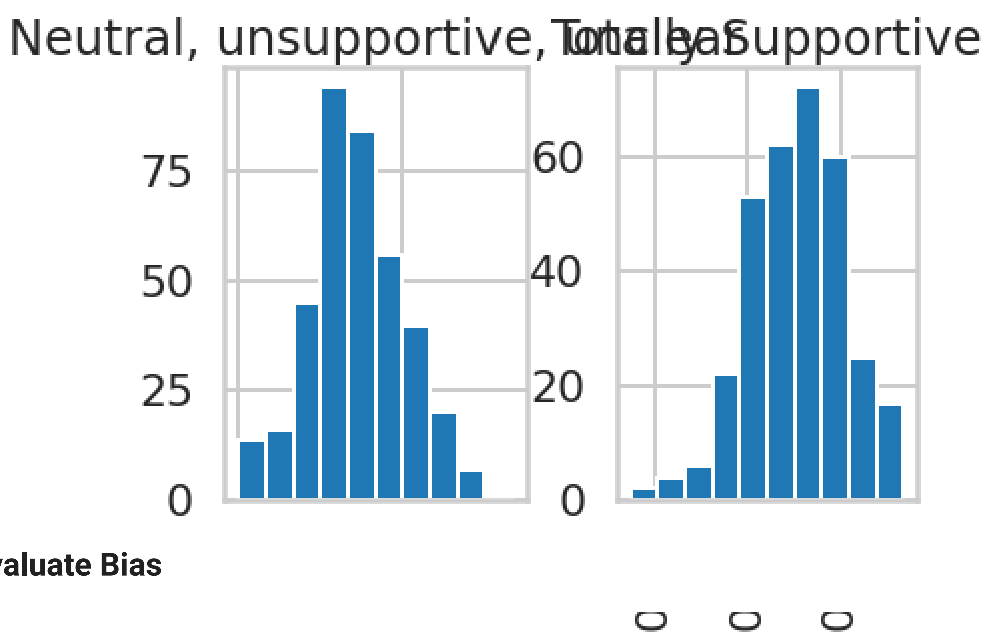
AUPRC

Show code



Plot the predictions

Show code



(5) Evaluate Bias

Evaluate Bias

Show code

```
{0: {'FNR': 0.1390728476821192,
  'lower confidence level': 0.09564723320158103,
  'upper confidence level': 0.18536488655023137},
 1: {'FNR': 0.06976744186046512,
  'lower confidence level': 0.05158627884143348,
  'upper confidence level': 0.17695473251028807}}
```

Report of Attitude Analysis towards Russian-Ukraine conflict on Twitter

by Chan Yu and Naihe Xiao

Work breakdown: Part I is done jointly, coding of Part II is mostly done by Chan, while presentation and data report is mostly done by Naihe.

1. Problem and Motivation

The Russian-Ukraine conflict has been a worldwide concentration for almost a decade, especially after the outburst of the invasive war on February 24th. Since this is the largest-scale war in the Western world after World War II, it's meaningful to attain a basic overview of the public's attitude towards it. Therefore, we are interested in investigating and predicting people's attitudes toward this topic on Twitter by using Natural Language Processing. This is a fascinating assignment, but not effortless in any aspect: while some of the tweets have obvious expressions of their opinions, others are rather ambiguous or sarcastic. In addition, since the Russian-Ukraine conflict has lasted for almost 10 years, many of the tweets are unrelated to this war. Therefore we focused on tweets in recent weeks and used specific search words to prevent irrelevance. In general, we are trying to answer the question: is it possible to predict the tweet's attitude towards the Russian-Ukraine war?

2. Data Extraction

We used tweepy to access the Twitter API for data extraction. To ensure the relevance of our data, we limited the data of the tweets to between March 28th and April 1st, and implemented restrictions so every tweet in our dataset contains both 'Ukraine' and 'Russia'. We extracted 700 observations in total and manually added two columns for each tweet: attitude and emotion.

The feature attitude is our parameter of interest. We classified each observation into two categories: whether it's completely supportive of Ukraine, or it's supportive of Russia, or neutral or unclear. We labeled the former category 1, and the latter category 0.

The feature emotion is labeled simply as a result of our interest: we want to know whether tweets supporting Ukraine are more emotional(furious) than their opponents or not.

Most observations in our data are highly relevant to the topic due to the time and tag restrictions we placed, hence appropriate for analysis. Although 700 is not a big dataset for general data analysis, it's sufficient in this case considering the method of labeling. In addition, we stored the labeled version of our data locally so we will always focus on the same set of samples.

However, the deficient number of features limits the scope of our investigation. Despite this drawback, the dataset we obtained is well-established and ready for analysis.

3. Exploratory Data Analysis

We read the data into a DataFrame, then applied the basic EDA checks to it. There are 700 observations in total and three columns including the original tweet text and the two features. Since the features were labeled manually, there are no NULL values in the dataset. Meanwhile, we converted the data type of the labeled features to integer type which smooths the following text preprocessing.

We plotted the total number of observations in each attitude group as well as emotion group. There are 323 tweet documents that fall in the group of 'Totally Supportive', while the rest of 377 documents fall in the other group. Before plotting, we were worried about whether simple predicting 'Totally Supportive' for all tweets would have reasonable accuracy since it won't be a surprise if the majority stands for Ukraine. However, the plots indicate our data is pretty balanced as 46.1% vs. 53.9% between the two groups, and we realized during the labeling that some tweets are irrelevant to our search words, but their author tagged them to increase public attraction and click-through rate.

Next, we preprocessed the data in a similar methodology that was required in Part 1. We tokenized the tweet text by whitespace and remove the URL links, followed by removing special characters and punctuations, and converting all tokens to lowercase. We applied both Porter stemmer and WordNet Lemmatizer to remove the suffixes and convert the token to its base form. In the end, we removed the stopwords and blank tokens that might be generated during the preprocessing.

4. Model

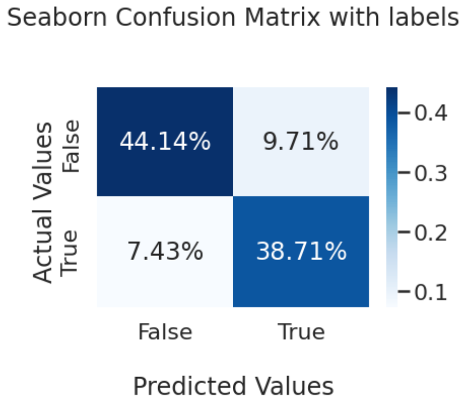
Since our outcome of interest, 'attitude', is a binary variable, hence we are trying to conduct binary classification. We applied the Naive Bayes model, a generative model, to our data for fitting and predicting, due to the following reasons: firstly, our data is fairly balanced (54% vs 46%), which is ideal as balanced data generally generate models with higher accuracy compared to imbalanced ones. Secondly, our machine learning task is a binary-class classification, where Naive Bayes usually illustrates outstanding performances. Finally, we have a small training set(total of 560, as discussed below), so Naive Bayes can yield better testing accuracy. Notice Naive Bayes model is a supervised model since it's trained using labeled data.

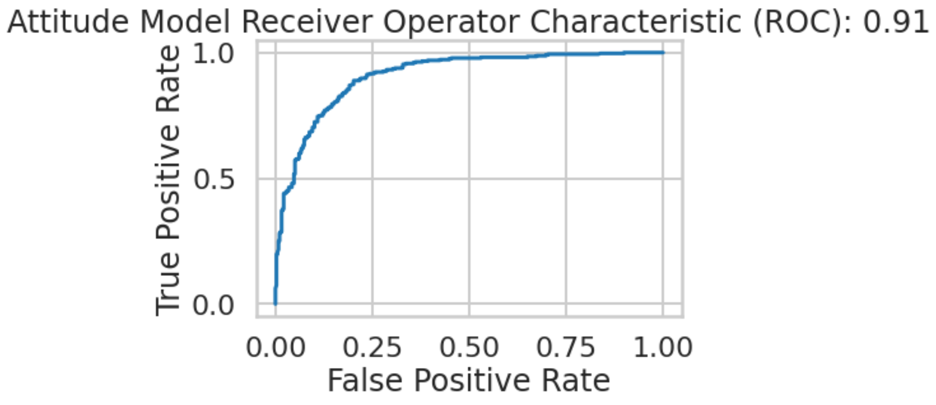
The underlying principle of the Naive Bayes model is the Bayes theorem, which says the posterior distribution(probability of obtaining observation labeled 1 given an observation) is proportional to prior distribution (probability of observing a sample labeled 1) times likelihood function (probability of observing a sample given it's labeled 1). The key assumption, also called the naive assumption, is that the words are independent of each other for a given class. It then estimates the probability of observing each class and the probability of observing each token conditional on class, so it can estimate the posterior probability for each word and label it with the class of higher probability.

Although the Naive Bayes Model is suitable under this circumstance, as mentioned above, the naive assumption that words are independent of each other conditional on class is usually irrational, as most people text based on logistics. Despite this problem, it's a highly applicable tool for us. Indeed, we can calculate the test accuracy and the confusion matrix to verify whether its performance matches our anticipation.

5. Results and Conclusion

The prediction result from our model matches our expectations. The training data accuracy is about 89%, and the testing data accuracy is about 60%, which indicates our model has an overfitting issue. The overfitting was severer with the initial dataset and model setup, where the accuracy was 94% vs. 53%. We have tried some approaches to fix this issue. First, we increased the size of the dataset by labeling more tweets documents. Second, we adjusted the ratio of data splitting for training vs. testing from 70:30 to 80:20, which enlarged the training set. These actions didn't solve the overfitting issue completely but improved the prediction performance slightly.





Both the True Positive Rate and the True Negative Rate are pretty high, and two other statistics in the matrix are fairly low. Meanwhile, we computed the statistics of the model based on the confusion matrix that the sensitivity/recall is about 0.84, specificity is about 0.82, and precision is about 0.80. Then, we plotted the ROC curve which indicates that our model classifier has pretty good performance since the curve is close to the top-left corner, while it should be lying along the diagonal for the random guessing classifier. Similarly, we can conclude the same trend from the PR curve.

Therefore it's appropriate to conclude that we can predict whether a tweet is completely supportive of Ukraine or not based on our model, given its overall performance. This result suggests that we can identify the group of tweets supportive of Ukraine when we are given a new set of tweets with high accuracy, and use the outcome for multiple purposes like deleting certain tweets from one group to manipulate public opinion(while this is not recommended).

In the final part of our analysis, we evaluated the performance of our model across Sentiment. The result shows that less emotional tweets tend to have a higher false-negative rate and a narrower confidence interval. This outcome suggests that the prediction may vary due to other ingredients. If we have more time, we will focus on the elimination of this difference, so that the prediction is not affected by external factors. The size of our data is relatively small(700), so we can not split it into 3 subsets with a validation set, which could have increased our accuracy. If we have collected more data, we will be able to have a validation set, which allows us to perform cross-validation hence training a better model.