Assignment 3

Xiaohan Li 1001357989

Work flow

Step 1: Data Cleaning
Step 2: Exploratory Analysis
Step 3: Model Preparation
Step 4: Model Implementation
Step 5: Discussion

Research Question: *"What can public opinion on Twitter tell us about the Canadian political landscape in 2019?"*

2019 Canadian Election Data Set

	sentiment	negative_reason	text
0	negative	Women Reproductive right and Racism	b"@RosieBarton So instead of your suggestion, ...
1	positive	NaN	b"#AllWomanSpacewalk it's real!\n@Space_Statio...
2	negative	Economy	b"#Brantford It's going to cost YOU $94 BILLIO...
3	positive	NaN	b"#Canada #CanadaElection2019 #CanadaVotes \n#...
4	negative	Economy	b"#Canada #taxpayers are sick & tired of h...

Sentiment Analysis Data Set

	text	label
0	Josh Jenkins is looking forward to TAB Breeder...	1
1	RT @MianUsmanJaved: Congratulations Pakistan o...	1
2	RT @PEPalerts: This September, @YESmag is taki...	1
3	RT @david_gaibis: Newly painted walls, thanks ...	1
4	RT @CedricFeschotte: Excited to announce: as o...	1

Step 1: Data Cleanning

The first step of processing text value is to make all the words in lowercase. Later, the following requirement is applied to the text value:

o All html tags and attributes (i.e., /<[^>]+>/) are removed.

o Html character codes (i.e., &...;) are replaced with an ASCII equivalent.

o All URLs are removed.

o All characters in the text are in lowercase.

o All stop words are removed. Be clear in what you consider as a stop word.

o If a tweet is empty after pre-processing, it should be preserved as such.

The coding snipping figure on the right present how these requirements are conducted.

The processed Sentiment Analysis datasets:

	text	label	clean_text
0	josh jenkins is looking forward to tab breeder...	1	bjosh jenkins looking forward tab breeders cro...
1	rt @mianusmanjaved: congratulations pakistan o...	1	mianusmanjaved congratulations pakistan becomi...
2	rt @pepalerts: this september, @yesmag is taki...	1	pepalerts september yesmag taking maine mendoz...
3	rt @david_gaibis: newly painted walls, thanks ...	1	davidgaibis newly painted walls thanks million...
4	rt @cedricfeschotte: excited to announce: as o...	1	cedricfeschotte excited announce july 2017 fes...

url remove

```
def remove_url(text):
    """
    This function is used to remove url portion of a text.
    reference: https://stackoverflow.com/questions/11331982/how-to-remove-any-url-within-a-string-in-python/40823105#40823105
    """
    text = re.sub(r'(https|http)?:\/\/(\w|\.|\/|\?|\=|\&|\%)*\b', '', text, flags=re.MULTILINE)
    return text
```

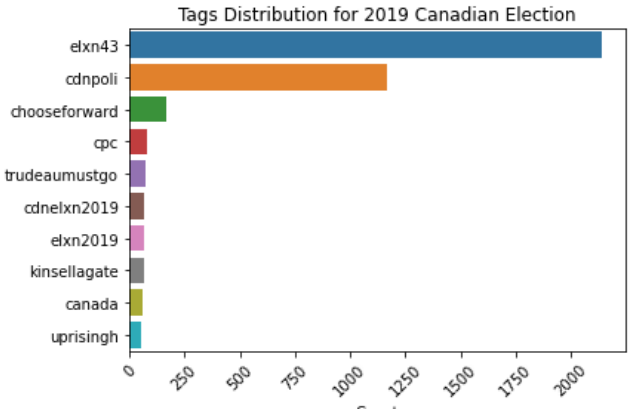
```
def remove_punc(text):
    """
    This function is used to remove those punctuation in the text while the punctuation list is from string.punctuation
    reference: https://stackoverflow.com/questions/265960/best-way-to-strip-punctuation-from-a-string
    """
    regex = re.compile('[%s]' % re.escape(punctuation))
    return regex.sub('', text)

def remove_stop(text):
    """
    This function is used to remove those stop words, and the stop words are from nltk.stipwords.
    """
    wordlist = text.split()
    word_temp = stopwords.words('english')
    word_temp = word_temp + [ 'rt','brt','http','https']
    cleanedwordlist = [word for word in wordlist if word not in word_temp]
    return " ".join(cleanedwordlist).strip()
```

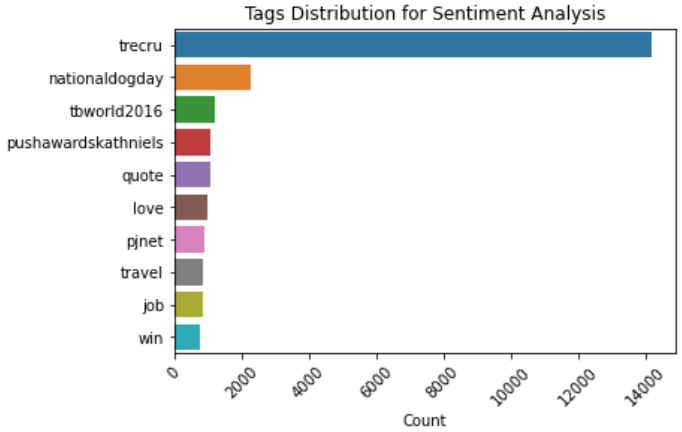
```
def clean_html(text):
    """
    This function is used to remove those html tags (such as <h1>Title</h1>).
    reference: https://stackoverflow.com/questions/9662346/python-code-to-remove-html-tags-from-a-string
    """
    clean_rule= re.compile('<.*?>')
    clean_text = re.sub(clean_rule, '', text)
    return clean_text
```

```
def replace_htmlcharac(text):
    """
    This function is used to replace html character codes (i.e., &...;) with an ASCII equivalent
    reference: https://www.geeksforgeeks.org/html-unescape-in-python/
    https://stackoverflow.com/questions/3194516/replace-special-characters-with-ascii-equivalent
    """
    text = html.unescape(text)
    text = unicodedata.normalize('NFD', text).encode('ascii', 'ignore')
    return text
```

Step 2: EDA

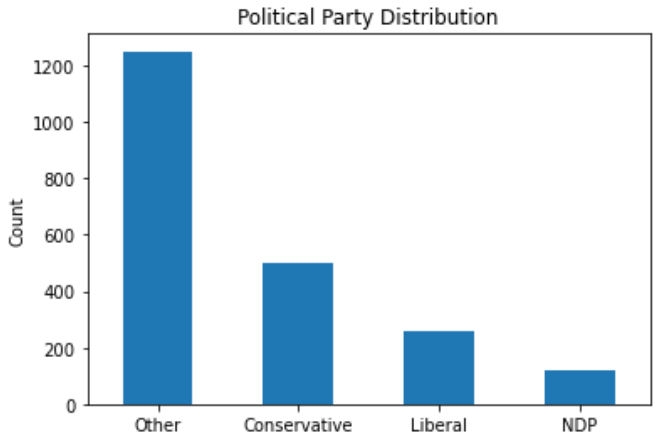
Topic 1: Tags Distribution for both data

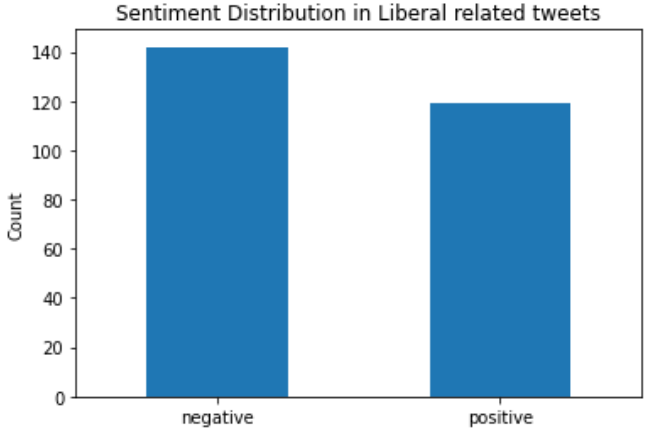




Topic 2: Political Tweets Distinguish

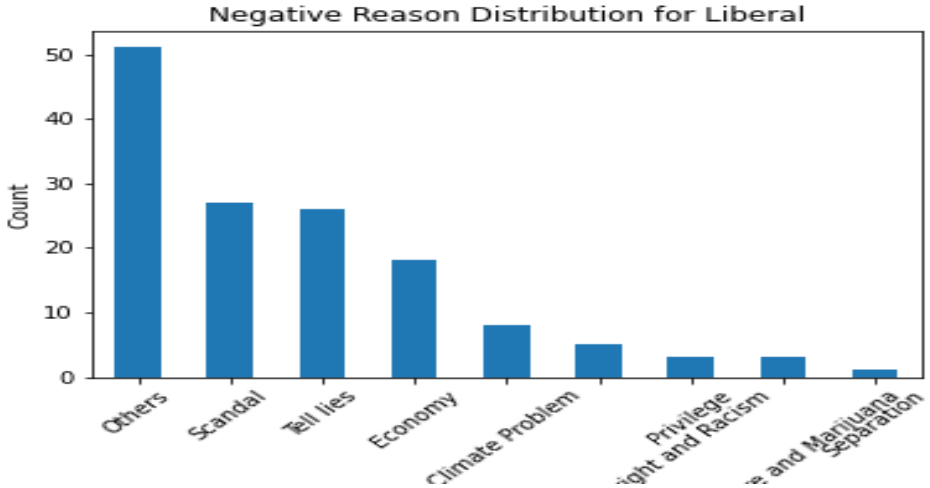
Use different Key Words to determine the political trend of a specific tweet, and find the sentiment distribution in each pollical class.





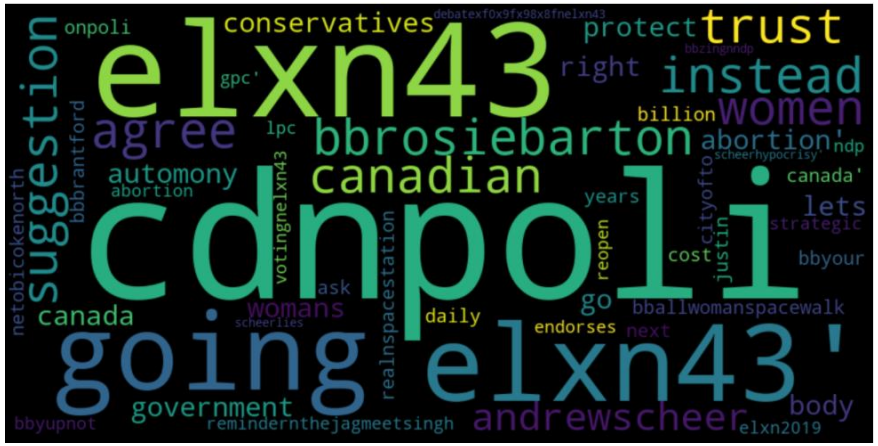
Topic 3: Negative Reason Distribution

Investigate more on the reason why user think negatively on the specific party



Topic 4: Word Cloud

Find the most mentioned word in the data



Step 3: Model Preparation and Implementation

1) Use Count and TFIDF Vectorizer to create WF and TFIDF dataset

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
#obtain our dataset
data = sentiment_analysis[['clean_text','label']]
```

```
#obtain text and target array
X = data.iloc[:,[0]].values.flatten()
y = data.iloc[:,1].values.flatten().astype('int')
```

2) First 100 words in the Bag of Words

Let's print first 100 features.

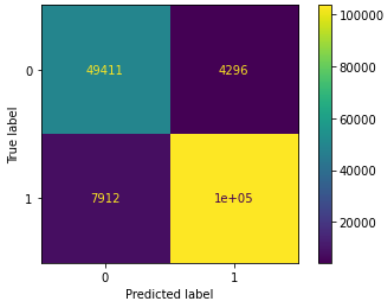
```
# print first 50 features
print(wf_vec.get_feature_names()[:100])
```

['10', '100', '11', '12', '15', '16', '1st', '20', '2016', '2017', '25', '30', '50', 'absolutely', 'abuse', 'account', 'action', 'actually', 'ad', 'adorable', 'af', 'afternoon', 'ago', 'ahead', 'aint', 'album', 'almost', 'alone', 'already', 'also', 'always', 'amazing', 'america', 'american', 'angel', 'animals', 'anniversary', 'annoying', 'another', 'anymore', 'anyone', 'anything', 'app', 'apple', 'appreciate', 'appreciated', 'arent', 'around', 'art', 'artist', 'ask', 'ass', 'attack', 'available', 'award', 'away', 'awesome', 'b10', 'b5', 'ba', 'babe', 'babies', 'baby', 'back', 'bad', 'bag', 'ball', 'bamazing', 'ban', 'band', 'banother', 'bare', 'bawesome', 'bbe', 'bbeautiful', 'bbest', 'bc', 'bcan', 'bcant', 'bcheck', 'bcome', 'bcongrats', 'bcongratulations', 'bday', 'bdo', 'bdont', 'beach', 'beat', 'beautiful', 'beauty', 'become', 'bed', 'behind', 'believe', 'benjoy', 'best', 'bet', 'better', 'bfor', 'bfound']

3) After training 7 models, Logistic Regression is chosen for the Sentiment prediction

	precision	recall	f1-score	support
0	0.86	0.92	0.89	53707
1	0.96	0.93	0.94	111411
accuracy			0.93	165118
macro avg	0.91	0.92	0.92	165118
weighted avg	0.93	0.93	0.93	165118

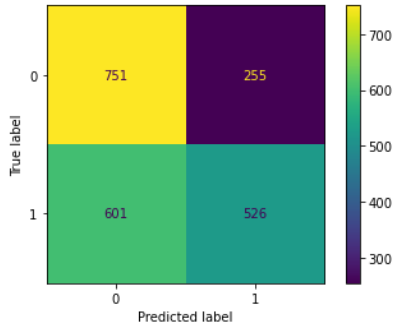
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2da54a3c248>

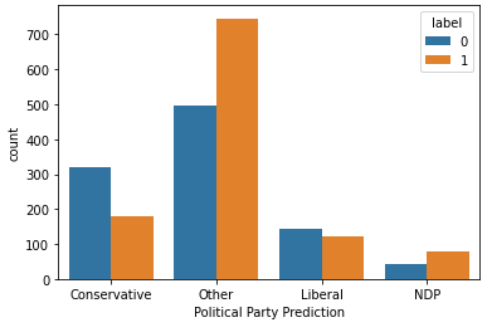


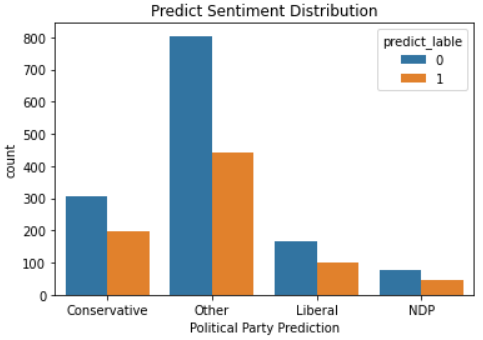
4) The model does not perform well for the 2019 Canadian Elections Dataset

	precision	recall	f1-score	support
0	0.56	0.75	0.64	1006
1	0.67	0.47	0.55	1127
accuracy			0.60	2133
macro avg	0.61	0.61	0.59	2133
weighted avg	0.62	0.60	0.59	2133

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2da58531e88>







5) Discussion

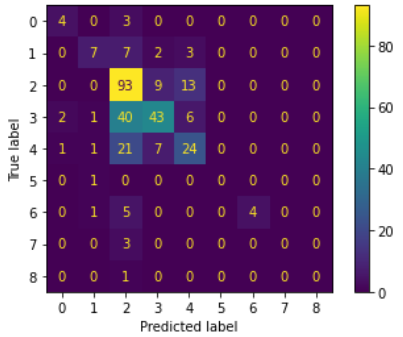
One major reason lead to the bad performance would be that tweeter users usually have specific words to describe their negative thoughts, but these specific words were not captured clearly by the sentiment analysis model, while the words used and vectorized in the model were all from sentiment analysis dataset. This problem would lead to a bad performance.

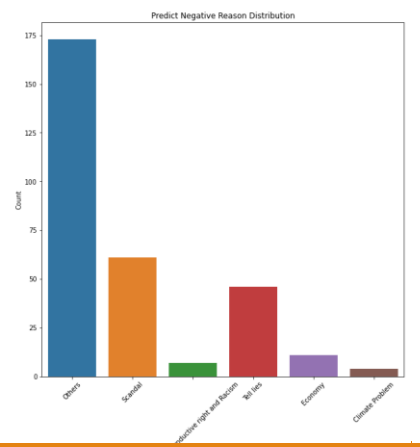
Step 5: Summary and Discussion

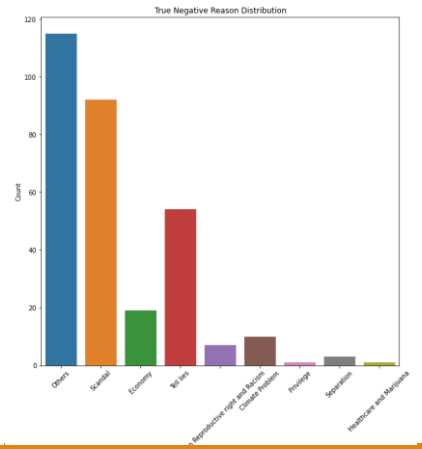
Random Forest

	precision	recall	f1-score	support
0	0.57	0.57	0.57	7
1	0.64	0.37	0.47	19
2	0.54	0.81	0.65	115
3	0.70	0.47	0.56	92
4	0.52	0.44	0.48	54
5	0.00	0.00	0.00	1
6	1.00	0.40	0.57	10
8	0.00	0.00	0.00	3
9	0.00	0.00	0.00	1
accuracy			0.58	302
macro avg	0.44	0.34	0.37	302
weighted avg	0.60	0.58	0.56	302

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2da59388948>







After using the first NLP sentiment analysis model, we can see that the model did not perform well on predicting the sentiments of users on 2019 Elections topic, while the performance of the model on the sentiment analysis dataset is pretty good. However, the model did summarize that tweeter users have more negative sentiments on a specific political party especially for "Conservative" and "Liberal“ respectively, which are the biggest 2 competitive political parties in Canada.

Secondly, a random forest model was applied to predict the negative reason regarding the 2019 elections. Although the model did not predict the number of each reason accurately, it describe the reason distribution well. Others, Scandal and Telling Lies are top 3 reasons that people have negative sentiments.

Improvement Suggestion:

For the first model, the model performance can be improved by adding more Canadian Election related tweets to the sentiment analysis dataset. In this case, more election related words can be added into bag of words for analysis. Therefore, the accuracy of the model when predicting the sentiments of users can be increased.

For the second model, one optional method to increase the performance is to increase the training data. Since we only have 2133 rows of tweets, the prediction on each negative reason would be difficult. After adding the data, the variation of each word in the bag can be increased, In this case, the model would perform better.