ABHIJIT NAYAK (nayakab), SPRING 2022, Paper 2 – Sentiment Analysis

WHAT IS THE PERCEPTION OF PEOPLE TOWARDS WORKING FROM HOME BASED ON THEIR TWEETS?(1500 words)

Introduction

Covid-19 pandemic has engulfed the world in a lot of problems and new changes have come up due to the existing problems at hand. Employees expressed conflicting emotions towards working from home as all the offices were closed due to the lockdown [1]. As a result of government mandates during the pandemic, people had to stick to remote work which means a sort of arrangement where people carry out their job responsibilities from the place where they live [2]. Businessmen, entrepreneurs, students, office workers, government employees and many others were affected because of the pandemic [3]. So, a lot of things changed during these difficult times and but only one thing remained constant which is work from home. Here I will be analyzing the sentiments of working people towards working from home in the corona virus pandemic using data scraped from twitter.

Research Question

What is the perception of people towards working from home based on their tweets?

Method

To gather data on this topic, I scraped tweets from Twitter by searching via different keywords in Python. Data collected for this paper was focused on tweets related to remote work, work from home in the corona virus pandemic for a certain timeframe. Then I did sentiment analysis, and I coded these sentiments as output labels for this data. After that I built machine learning (multi-class) classification models trained on this data. Finally, I used these trained models to perform classification on some test data (new tweets scraped for a different timeframe).

Data Collection:

I searched for all relevant tweets on remote work or work form home or telework during the Corona Virus pandemic based on different keywords like *"work from home", "remote work", "working remote", "work remote", "worked from home", "working from home", "wfh"*. My choice of keywords aren't case-sensitive. Scraping tweets based on these keywords from Twitter gave me the most precise information related to the objective of this paper. For instance, the collected tweets as mentioned below:

1. Katie, @reelslimkatiee. (2021 March 11). The nice things about working from home is when something at work pisses you off, you can go do something. Twitter. https://twitter.com/reelslimkatiee/status/1370118702847336450
2. Cohen, @lindsaycohen. (2021 March 23). When everyone else stayed home frontline journalists went to news conferences asked questions informed the public it is infuriating to. Twitter. https://twitter.com/lindsaycohen/status/1374564951331794944

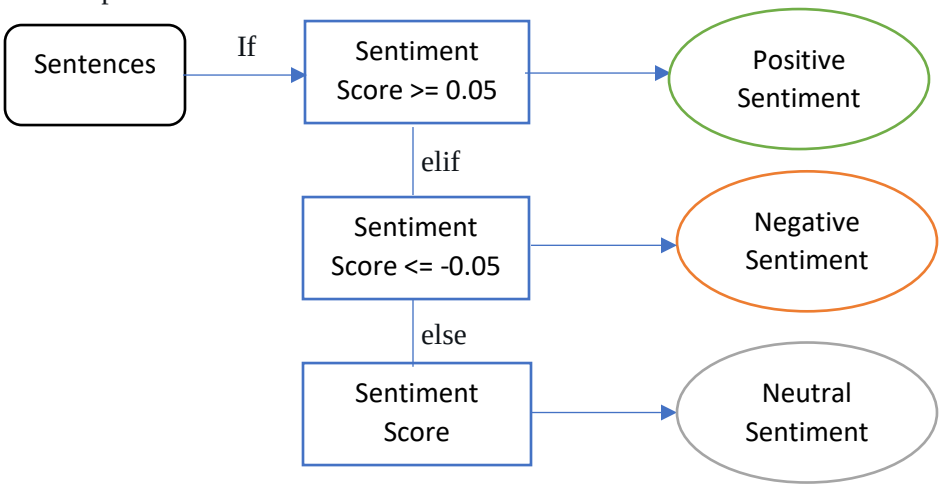
The first tweet shows the positive attitude of her towards the work from home initiative. The context of the tweet implies that she is able to take a break if something goes wrong at work and clear her mind the return back to work easily. Second tweet clearly states a negative sentiment towards remote work as she says in her post that frontline workers like journalists can't work from home. As their job is a field job where they need to go to news conferences and ask questions to inform the public.

The scraping of the pertinent information regarding these tweets was completely done by Python script. I use snscrape a library in Python to scrape data from Twitter. I didn't choose Tweepy or twint because I needed data from March 2020 onwards which wasn't possible via Tweepy and twint packages. With snscrape, I was able to scrape tweets more than 1500 tweets from 2020, March 1st to 2021, June 30th for different locations in USA, India, Japan, and Germany. I concatenated the data collected for each location into one final dataframe as my train set for further analysis. I collected my test data set from 2021, August 1st to 2022, January 1st consisting of 500 tweets. So, my train dataset has a timeframe of 15 months from March 2020 to June 2021 and test set has a timeframe of 5 months from August 2021 to January 2022.

Analysis:

Sentiment Analysis

The total number of tweets collected in the train dataset was 1710. It had few other language codes apart from English i.e., 'Tamil', 'Telugu', 'Kannada', 'Deutsche', 'Japanese', 'Undefined'. So, for simplicity I removed all the rows from the train data.frame whose language code was not 'en' (English). I removed these instead of using a paid translation API because the tweets containing different language code were less. So, a data.frame of all English tweets were left at hand of length 1410. I used Vader (Valence Aware Dictionary and Sentiment Reasoner) available in vaderSentiment package in python. I chose Vader because it is a lexicon and rule-based sentiment analysis tool that is specifically accustomed to the sentiments expressed in social media.



Then, I encoded positive sentiment as '1', negative sentiment as '-1', neutral sentiment as '0' in a column "sentiments" in the train dataframe which would serve as my target column for machine learning model training in the further steps.

	content	sentiments
0	walking home from work my eyeballs are actuall...	0
1	question for my work from home peeps how produ...	0
2	shannonodkomo yeah when i got home from work m...	1
3	angryblacklady our old keurig broke early in t...	-1
4	bonnevivante ill probably water at am when i g...	0
...	...	...
1405	bianchicarole itsmarkhamill hamillhimself laur...	1
1406	miloshthemedic i would love to do that but ple...	-1
1407	in these times im very happy to have the right...	1
1408	im really thankful i have my patreons and that...	1
1409	taraustralis drcaplin skyfire so is our gym an...	1

1410 rows × 2 columns

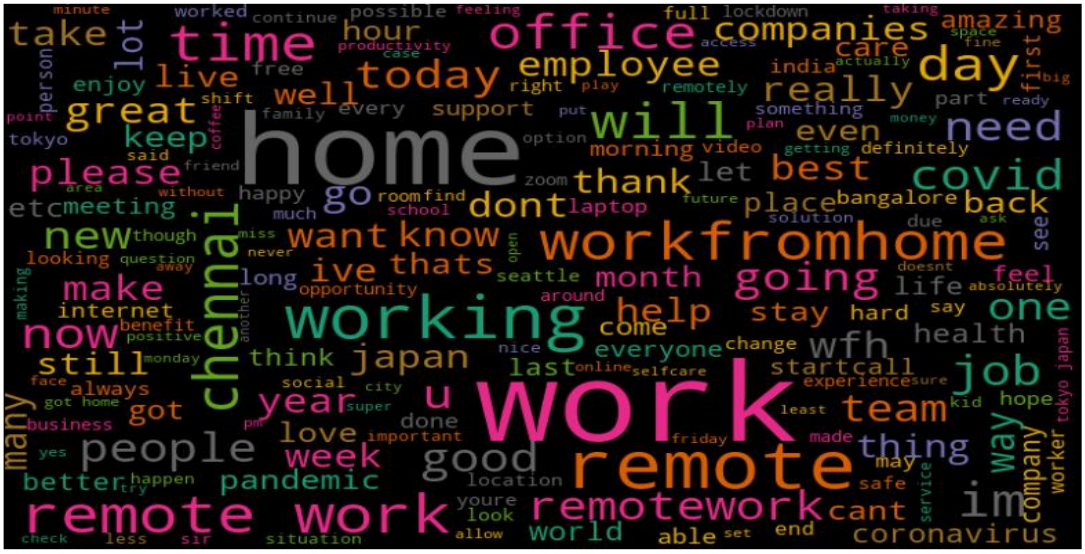
Pre-Processing Tweets

I made a copy of the dataframe to keep the original dataframe intact for backup. Changed the text of the tweet in the dataframe to lowercase. Removed the URL links and HTML reference characters with the help of regular expressions. Then reset the index of the dataframe and dropped the old indices. Also, removed the non-letter characters with the help of regular expressions and finally removed punctuations, numbers, and special characters. Now, I have a clean dataframe ready containing all the tweets in English language.

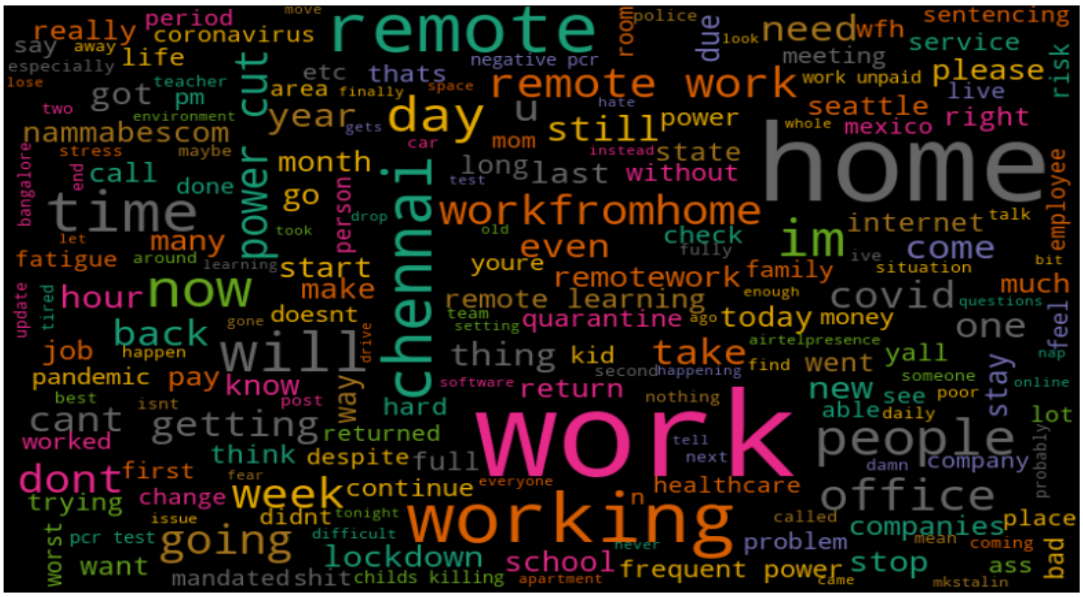
url	date	content	renderedContent	id	user	replyCount
https://twitter.com/Dan8arton/status/140966639...	2021-06-29 00:13:54+00:00	walking home from work my eyeballs are actuall...	Walking home from work. My eyeballs are actual...	1409666398998056962	{'_type': 'snscrape.modules.twitter.User', 'us...	0
https://twitter.com/chickey253/status/14095032...	2021-06-28 13:25:32+00:00	question for my work from home peeps how produ...	Question for my work from home peeps... how prod...	1409503232074813440	{'_type': 'snscrape.modules.twitter.User', 'us...	0
https://twitter.com/thunderstorm106/status/140...	2021-06-28 07:56:43+00:00	shannonodkomo yeah when i got home from work m...	@ShannonODKOMO Yeah when I got home from work ...	1409420483653619713	{'_type': 'snscrape.modules.twitter.User', 'us...	0
https://twitter.com/riotheatherrr/status/14092...	2021-06-27 19:37:47+00:00	angryblacklady our old keurig broke early in t...	@AngryBlackLady Our old Keurig broke early in ...	1409234522432962561	{'_type': 'snscrape.modules.twitter.User', 'us...	0
https://twitter.com/MAndersson1968/status/1408...	2021-06-27 03:13:39+00:00	bonnevivante ill probably water at am when i g...	@bonnevivante I'll probably water at 1am when ...	1408986858097360896	{'_type': 'snscrape.modules.twitter.User', 'us...	1

Story Generation and Visualization

Plotted a wordcloud showing the most common words in positive tweets:

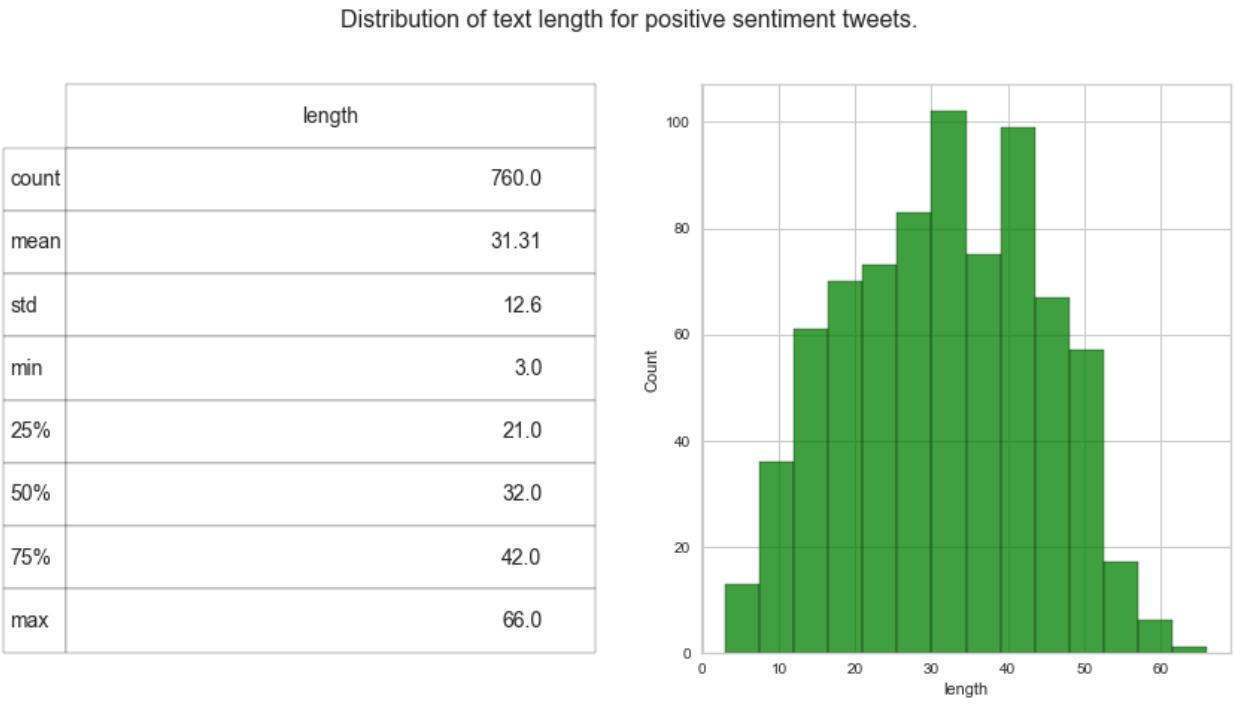


Plotted a wordcloud showing the most common words in negative tweets:



Plotted a wordcloud showing the most common words in neutral tweets:

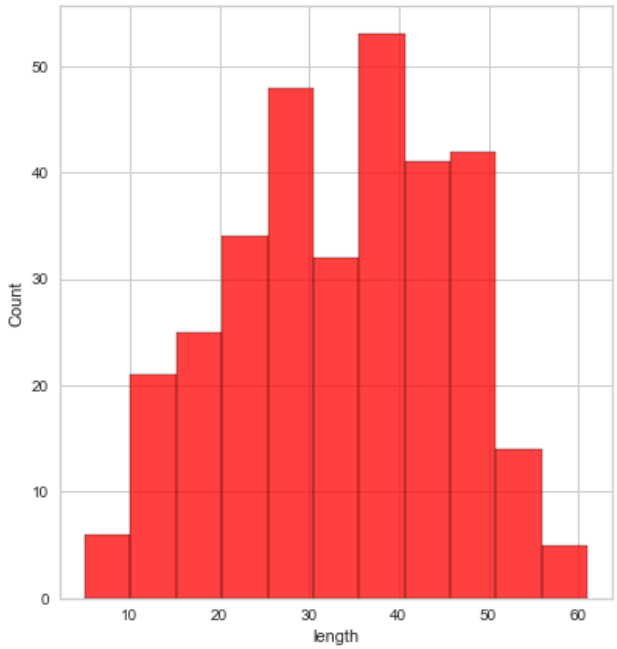
Distribution of text length for positive sentiment tweets:



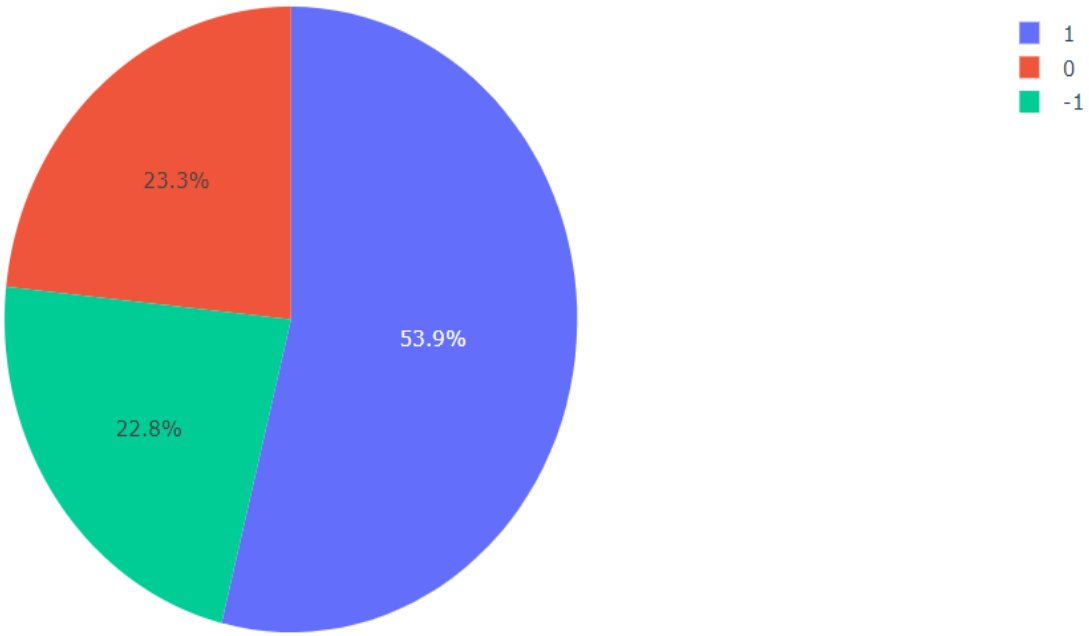
Distribution of text length for negative sentiment tweets:

Distribution of text length for positive sentiment tweets.

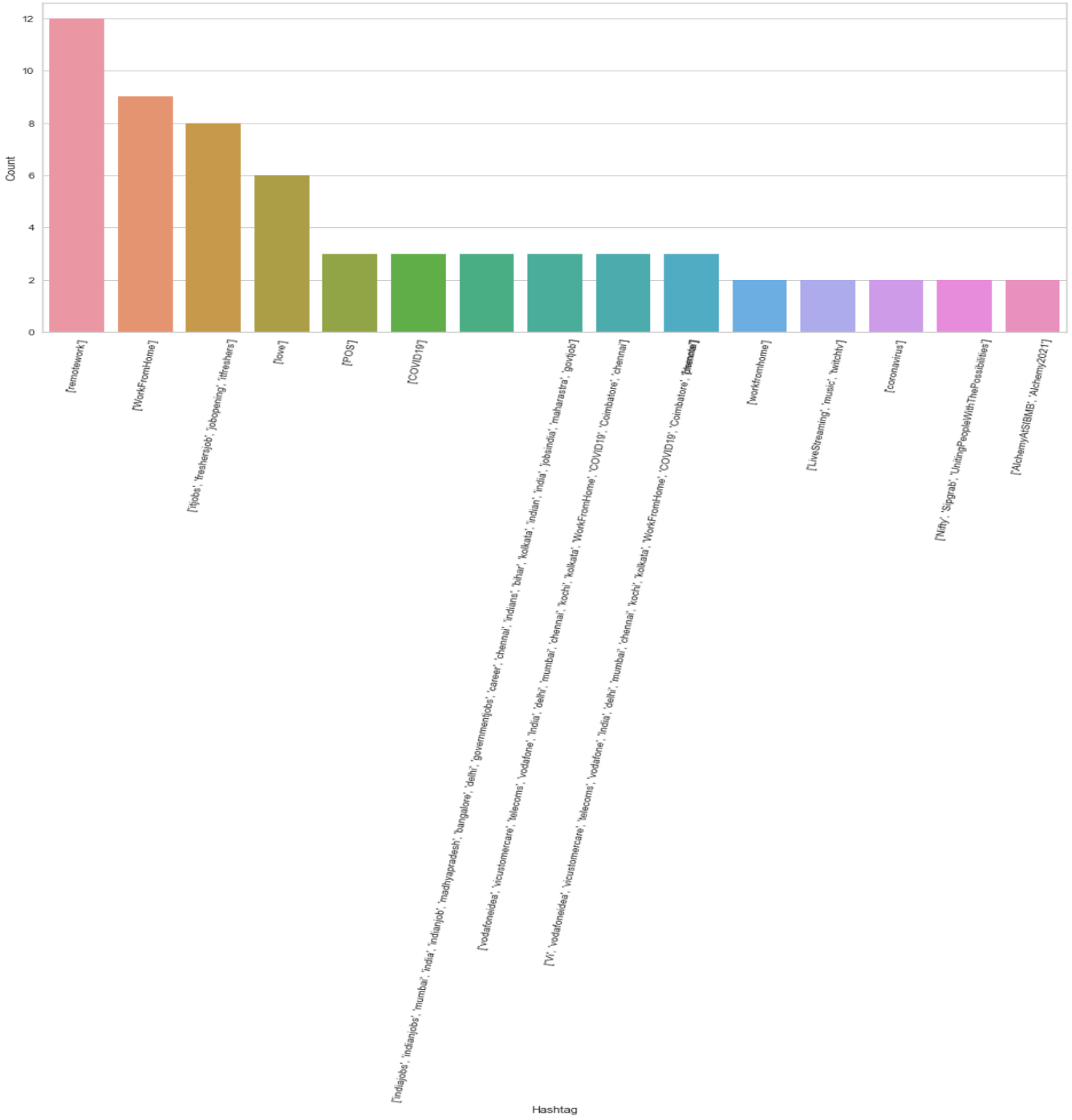
	length
count	321.0
mean	33.49
std	11.96
min	5.0
25%	24.0
50%	35.0
75%	43.0
max	61.0



Pie chart of different sentiments of tweets:



Most Commonly used hashtags:



For visualization and story generation, I used matplotlib (plotting bar plots), plotly (potting pie charts), seaborn (histplot) and Wordcloud (using wordcloud) packages respectively.

Sentence Bert- Word Embedding of the pre-processed Tweets

I used transfer learning here to perform the word embeddings of the text or pre-processed tweets which would be in turn fed as input to machine learning models. So, I used "paraphrase-MiniLM-L6-v2" pretrained model to perform embeddings on the contents.

```
0       [ 2.94951051e-01 -2.15740845e-01  3.56519222e-...
1       [ 9.74271074e-02 -2.47539520e-01  5.57080925e-...
2       [ 7.56199509e-02  2.91682273e-01  1.19519241e-...
3       [-2.73747951e-01  1.16695531e-01  2.23515946e-...
4       [ 3.81139874e-01  2.05805540e-01  4.71804857e-...
                                ...
1405    [ 7.28416145e-02  2.66791612e-01 -2.32497901e-...
1406    [ 0.14342813  0.21905158  0.00822043 -0.078754...
1407    [-0.07896378 -0.09476763 -0.06954506 -0.230162...
1408    [-0.00367064 -0.02306548  0.07028096 -0.124976...
1409    [ 4.01110381e-01 -4.97798979e-01  8.31544623e-...
Name: content_embeddings, Length: 1410, dtype: object
```

Classification-Machine Learning Modelling

Training

I built 4 models in total: Random Forest, XGBoost, Support Vector Machines and Stacking CV Classifier. I passed the sentence embeddings (generated in the previous step using Sentence Bert) as input to the models above and trained them on it. I kept the target column as 'sentiments' generated after the sentiment analysis step. I also performed hyperparameter-tuning for all the 4 models. For the stacking classifier, I used SVM as my meta learning model because it had the best results amongst the rest and used XGBoost, Random Forest as second level of learning. And I observed that the ROCAUC scores increased drastically in the stacking classifier compared to the rest 3.

Testing

I generated another 500 rows of tweets from March 2020 to June 2021 and performed all the pre-processing steps on this dataset. Then I did predictions on this data using each of the 4 models. Since this is the test data with no true labels, I applied Vader sentiment analysis on this dataset to generate sentiments and coded them as labels for test predictions performance comparison. I compared the classifier model predictions with Vader sentiments and found that SVM had the best results. Stacking classifier results drastically fell compared to the train predictions suggesting that this model is over-fitting.

Results

Training Set

Metrics/ Model	Random Forest	XGBoost	SVM	Stacking CV Classifier
Accuracy	0.67	0.66	0.72	0.94
ROCAUC	0.78	0.81	0.85	0.99

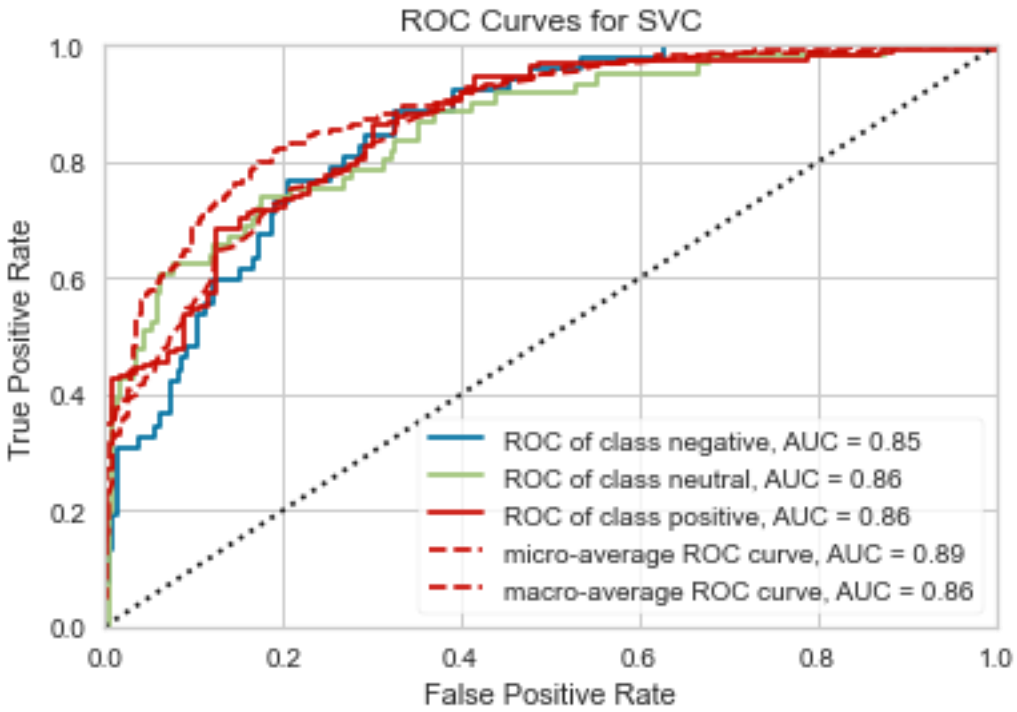
For SVM

```
Confusion Matrix :
  [[ 30   7  15]
 [  8  38  15]
 [ 17  15 137]]

 Classification Report :
               precision    recall  f1-score   support

           -1       0.55      0.58      0.56        52
            0       0.63      0.62      0.63        61
            1       0.82      0.81      0.82       169

    accuracy                           0.73       282
   macro avg       0.67      0.67      0.67       282
weighted avg       0.73      0.73      0.73       282
```



Testing Set

Metrics/ Model	Random Forest	XGBoost	SVM	Stacking CV Classifier
Accuracy	0.55	0.61	0.62	0.62
ROCAUC	0.76	0.79	0.81	0.82

For SVM

```
Accuracy Score :
  0.628

 AUC Score :
 0.8122466743309698

 Confusion Matrix :
  [[ 60  11  50]
 [ 20  73  51]
 [ 32  22 181]]

 Classification Report :
              precision    recall  f1-score   support

          -1       0.54      0.50      0.52       121
           0       0.69      0.51      0.58       144
           1       0.64      0.77      0.70       235

    accuracy                           0.63       500
   macro avg       0.62      0.59      0.60       500
weighted avg       0.63      0.63      0.62       500
```

The results above state that the stacking CV classifier is over-fitting and SVM has the best performance compared to the rest.

From the value counts obtained on test set below and the count of sentiments in the train set shown below suggests that the general population has a positive sentiment towards working from home.

```
Test Data Sentiments from Vader:
 1    235
 0    144
-1    121
Name: content, dtype: int64

Test Data predictions by the classifier:
 1    282
-1    112
 0    106
dtype: int64
```

```
Train data sentiments from Vader
```

Out[60]:

```
 1    760
 0    329
-1    321
Name: sentiments, dtype: int64
```

Conclusion

In this paper, the sentiment analysis was conducted using around 2000 tweets from March 2020 to January 2022 from Twitter to study people's approach towards work from home during corona virus pandemic. The positive sentiments (56%) towards remote work during the pandemic are more compared to the negative sentiments (24%). Rest 20% of the sentiments were neutral towards this approach. There was majority of positive sentiments compared to the negative and neutral in all the countries: USA, India, Germany, and Japan. This was because people were more productive working from home, they were able to spend more time with family saving a lot of transportation costs and bringing the AQI down as there were less vehicles running and so less pollution. But there were few concerns, some people were becoming unfit because of the unhealthy regime from working from home and frontline workers like journalists, daily wage laborers, doctors had to be in the office or field to carry out their roles and responsibilities. Even though people were being productive, developing new hobbies there was one major concern of health issues arising due to lack of physical work. I also observed some limitations while working on this paper was there were tweets in regional languages and the amount of data being scraped was very less. So, to translate the regional language tweets we would need to use paid language translation APIs and then the amount of data would increase which could permit the usage of neural networks into the picture. These could be the future aspects to be looked into.

References

[1] Nagaratna Parameshwar Hegde, Sreesha Vikkurty, Gnyanee Kandukuri, Sriya Musunuru, Ganapatikrishna Parameshwar Hegde. (2021, October 4). Employee sentiment analysis towards remote work during COVID-19 using twitter data.

[2] Charlene Zhang, Martin C. Yu, Sebastian Marin. (2021, May 3) Exploring public sentiment on enforced remote work during COVID-19.

[3] Twinkle Goyal, Nikhil Malhotra. (2021). Sentiment analysis using twitter information flow about the work from home culture that is widely adopted due to covid pandemic.