PROJECT REPORT

IRWA - PART 1 - G102-2

Clàudia Morales - 199906
Roger Sola - 199780
Queralt Zamora - 199903
G102-2

Universitat Pompeu Fabra Barcelona

1. Introduction

This document is the Part 1 Project Report, prepared by team G102-B. Here, it is described all the necessary information about the elaboration of the code for this first part, including the assumptions, methods, and algorithms employed. Each subsection presents the reasoning behind the decisions made and is classified into two main parts.

The team developed the code using a Google Colaboratory file, which required importing libraries and the provided dataset. The project aims to preprocess a dataset of tweets related to the 2021 Farmers Protests in India. This event was widely discussed on social media platforms like Twitter, making it an ideal source for analysis.

The repository for the deliveries has been shared with the labs teacher Francielle Do Nascimento, but since it is a public repository, the link to get in is the following:

https://github.com/QuerZamora/IRWA_G102_2

Clàudia Morales - 199906
Roger Sola - 199780
Queralt Zamora - 199903
G102-2

Universitat Pompeu Fabra Barcelona

2. Preprocessing Data

In this section, we explain the preprocessing steps applied to the dataset of tweets related to the 2021 Farmers Protests. This part of the project is important to ensure that the data is clean, consistent and perfect to analyse.

2.1. Lowercase conversion

Standardizing the text is a crucial preprocessing step. All tweets are converted to lowercase to ensure that words like "Protest" and "protest" are treated equally during analysis.

```
# preprocessing function
def preprocess_text(text):
    text = text.lower() # convert text to lowercase
```

2.2. URL removal

URLs in tweets sometimes provide context but do not add value to the content analysis itself. So we removed URLs using regular expressions. This step helps reduce noise.

```
text = re.sub(r'http\S+|www\S+', '', text) # remove URLs
```

2.3. Hashtag handling

According to Hint 2, we decided to extract the hashtags separately for further analysis. Hashtags can carry important information, as they are often used to categorize or emphasize topics in tweets. After extraction, hashtags are removed from the main tweets to reduce noise and ensure that the remaining is clean for analysis. Hashtags are stored separately for future tasks.

```
# hashtags = re.findall(r'#\w+', text)
hashtags = re.findall(r'#\S+', text) # extract hashtags separately (HINT 2)
text = re.sub(r'#\w+', '', text) # delete hastags from main tweet (HINT 2)
```

2.4. Space removal

Tweets often contain extra spaces, which can produce inconsistency. As part of the preprocessing, we removed the initial and ending spaces, and double spaces are reduced to a single space. This step ensures that tweets are uniformly formatted and ready for tokenization and further analysis.alysis.

```
# more preprocess methods
text = text.strip() # removing spaces at the beginning and spaces at the end
text = re.sub(r'\s+', ' ', text) # changing double spaces to single spaces.
```

2.5. Stemming and Stop-word removal

To reduce words to their root form, we applied stemming to the tokens in each tweet. This helps in unifying variations of the same word as a single entity (e.g., "protesting," "protesters," and "protests" are all reduced to "protest"). Stop-word removal is also performed. Stop-words are common words that don't add significant meaning (e.g., "and," "the," "is") and we removed them to focus on the more meaningful terms. In this step, we also make sure that punctuation marks are filtered out to ensure only relevant words remain for future analysis.

```
tokens = word_tokenize(text) # tokenize text
filtered_tokens = [stemmer.stem(word) for word in tokens if word not in stop_words and word not in punctuation ]
return filtered_tokens, hashtags
```

2.6. Preprocessed Tweets storage

Once the preprocessing steps are applied, the resulting data for each tweet is stored in a structured format. This ensures that the tweet data is ready for further analysis.

```
# read and preprocess the data line by line from .json file
preprocessed_tweets = []
with open(docs_path, 'r') as file:
    for line in file:
        try:
            tweet = json.loads(line)

            # extract relevant information
            tweet_id = tweet.get('id')
            content = tweet.get('content')
            preprocessed_content,hashtags = preprocess_text(content)
            preprocessed_tweets.append({
                'id': tweet_id,
                'preprocessed_content': preprocessed_content,
                'date': tweet.get('date'),
                'hashtags': hashtags,
                'likes': tweet.get('likeCount'),
                'retweets': tweet.get('retweetCount'),
                'url': tweet.get('url')
            })
        except json.JSONDecodeError as e:
            print(f"Error parsing line: {e}")
```

2.7. Map IDs

For further analysis in the evaluation stage, it was necessary to map the tweets' Ids with the document ids, as the document Ids will be considered in next parts. It was necessary to upload the .csv file (as

tweet_ids_map) that content the relation between the docsId and the Tweets ID, and then produce a left join with the actual dataset that contains all the information of the tweets (tweet_info_df).

```
ids_path = '/content/drive/MyDrive/1st TERM/RIAW/LABS/PART 1/data/tweet_document_ids_map.csv'

# ids csv loaded
tweet_ids_map = pd.read_csv(ids_path)
# display(tweet_ids_map)
```

After this, the data frame of tweet_id_info contains all the necessary information. It is also important to note that, since the number of entries from the new .csv file is much less than the total number of tweets, those tweets that don't have any document ID relation have the value of NaN in that column. In the following picture it can be observed the necessary code for the merge and the display of the first rows of the updated data frame.

```
# merged the tweet_info_df with the tweet_ids_map based on the 'Tweet ID' column
# 'id' in tweet_ids_map corresponds to 'Tweet ID' in tweet_info_df
tweet_info_df = tweet_info_df.merge(tweet_ids_map, left_on='Tweet ID', right_on='id', how='left')

# renamed 'docid' column from tweet_ids_map to 'Document ID' in the merged DataFrame
tweet_info_df = tweet_info_df.rename(columns={'docid': 'Document ID'})

# drop the 'id' column that was used for merging if no longer needed
tweet_info_df = tweet_info_df.drop(columns=['id'])

display(tweet_info_df.head())
```

	Tweet ID	Tweet	Date	Hashtags	Likes	Retweets	Url	docId
0	1364506249291784198	world progress indian polic govt still tri tak...	2021-02-24T09:23:35+00:00	#modidontsellfarmers, #farmersprotest, #freeno...	0	0	https://twitter.com/ArjunSinghPanam/status/136...	doc_0
1	1364506237451313155	kisanektamorcha farmer constantli distroy crop...	2021-02-24T09:23:32+00:00	#farmersprotest, #modiignoringfarmersdeaths, #...	0	0	https://twitter.com/PrdeepNain/status/13645062...	doc_1
2	1364506195453767680	ਪੈਟਰੋਲ ਦੀਆਂ ਕੀਮਤਾਂ ਨੂੰ ਮੱਦੇਨਜ਼ਰ ਰੱਖਦੇ ਹੋਏ ਮੇਰੇ...	2021-02-24T09:23:22+00:00	#farmersprotest	0	0	https://twitter.com/parmarmaninder/status/1364...	NaN
3	1364506167226032128	reallyswara rohini_sgh watch full video	2021-02-24T09:23:16+00:00	#farmersprotest, #nofarmersnofood	0	0	https://twitter.com/anmoldhaliwal/status/13645...	doc_2

2.8. Output display

To ensure the preprocessing is successful, the tweet information is displayed in the proper format.

```
# printing method as demanded (HINT 1)
print("Tweet | Date | Hashtags | Likes | Retweets | Url")
print()
for index, row in tweet_info_df.head().iterrows():
    print(index, f"| {row['Tweet']} | {row['Date']} | {row['Hashtags']} | {row['Likes']} | {row['Retweets']} | {row['Url']}")
```

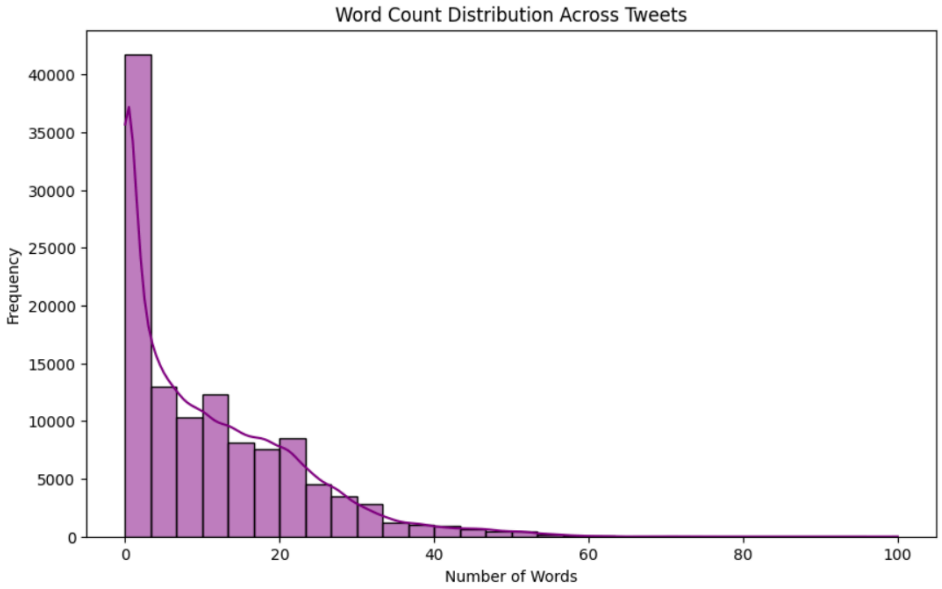
```
Tweet | Date | Hashtags | Likes | Retweets | Url

0 | world progress indian polic govt still tri take india back horrif past tyranni narendramodi delhipolic shame | 2021-02-24T09:23:35+00:00 | #modidontsellfarmers, #farmersprotest,
1 | kisanektamorcha farmer constantli distroy crop throughout india realli 's heart break ... care crop like children govt agricultur minist laugh us🚜🏌w win💪 | 2021-02-24T09:23:
2 | ਪੈਟਰੋਲ ਦੀਆਂ ਕੀਮਤਾਂ ਨੂੰ ਮੱਦੇਨਜ਼ਰ ਰੱਖਦੇ ਹੋਏ ਮੇਰੇ ਹਿਸਾਬ ਨਾਲ ਪ੍ਰਦੇਸ਼ ਜਾਣ ਨਾਲੋਂ ਬਿਹਤਰ ਆ ਭਾਰਤ ਚ ਪੈਟਰੋਲ ਪੰਪ ਪਾ ਲਈਏ। 😂😂🤔🤔 | 2021-02-24T09:23:22+00:00 | #farmersprotest | 0 | 0 | https://twitt
3 | reallyswara rohini_sgh watch full video | 2021-02-24T09:23:16+00:00 | #farmersprotest, #nofarmersnofood | 0 | 0 | https://twitter.com/anmoldhaliwal/status/1364506167226032128
4 |  | 2021-02-24T09:23:10+00:00 | #kisanektamorcha, #farmersprotest, #nofarmersnofood | 0 | 0 | https://twitter.com/KotiaPreet/status/1364506144002088963
111329 | मध्यप्रदेश में निजी व्यापारी 200 करोड़ का धान खरीदकर भाग गया। किसान पैसे के लिए धरना दे रहे है। अब भी बताना पड़ेगा काला क्या है?सान_एकता_जिंदाबाद | 2021-02-12T10:10:56+00:00 | #farmlaws, #farmer
```

Clàudia Morales - 199906
Roger Sola - 199780
Queralt Zamora - 199903
G102-2

3. Data Analysis

In this section, we present the data analysis applied to the dataset of tweets related to the 2021 Farmers' Protests. The analysis provides information about the word count distribution, vocabulary size, most frequent words, hashtag analysis and the relationship between likes and hashtags. In this part, we are going to show only the graphics of the results and give a brief explanation.

3.1. Word Count distribution



The number of words in tweets seems to follow a geometric (or exponential) distribution. Therefore, we can say that tweets tend to be short texts. In fact, there is a peak at the beginning, indicating that most of them do not exceed 5 to 10 words.

3.2. Average sentence length

```
# 2. Average sentence length

#create_markdown("2. Average sentence length")
average_sentence_length = tweet_info_df['word_count'].mean()
print(f"Average number of words per tweet: {average_sentence_length:.2f}")
```

```
Average number of words per tweet: 10.87
```

We calculated the average sentence length across all tweets to provide how much the users tend to write in their tweets. In the result, we can see that the average tweet length contains around 10 words which makes sense observing the previous section. This tells us that most users are concise at the moment to write their tweets.

3.3. Vocabulary size

```
# 3. Vocabulary size

#create_markdown("3. Vocabulary size")
all_words = ' '.join(tweet_info_df['Tweet']).split() # combine all words from the tweets
vocabulary = set(all_words)
vocab_size = len(vocabulary)# Count unique words (vocabulary size)
print(f"Vocabulary size (unique words): {vocab_size}")
```

```
Vocabulary size (unique words): 104894
```

To observe the diversity of the language used in the dataset we calculated the vocabulary size, the number of unique words across all tweets. We get as a result a total of 104894 unique words, indicating a rich and diverse use of the language.

3.4. Ranking of Tweets based on Likes and Retweets

Top 10 Most Retweeted Tweets:

	Tweet	Retweets	Url
111329	मध्यप्रदेश में निजी व्यापारी 200 करोड़ का धान ...	7723	https://twitter.com/RakeshTikaitBKU/status/136...
7645	's happen germani german govt ' block path bar...	6164	https://twitter.com/dhruv_rathee/status/136414...
89780	disha ravi 21-year-old climat activist arrest ...	4673	https://twitter.com/rupikaur_/status/136088206...
88911	disha ravi broke court room told judg mere edi...	3742	https://twitter.com/amaanbali/status/136090860...
111556	farmer sweet ' see amandacerni rihanna 🖤 🥺	3332	https://twitter.com/jedijasmin_/status/1360162...
64492	india target young women silenc dissent amp mu...	3230	https://twitter.com/rupikaur_/status/136179092...
108072	bollywood betray panjab amp farmer india hero ...	3182	https://twitter.com/RaviSinghKA/status/1360260...
60721	लहरों को ख़ामोश देख कर ये ना समझना कि समंदर मे...	3057	https://twitter.com/sherryontopp/status/136189...
29510	हाँ मैं जानता हूँ कि मैं शायर नहीं और ज़ुल्म क...	3040	https://twitter.com/sherryontopp/status/136309...
24160	कलियुग है साहब यहाँ झूठे को स्वीकार किया जाता ...	2622	https://twitter.com/sherryontopp/status/136337...

Top 10 Liked Tweets:

	Tweet	Likes	Url
7645	's happen germani german govt ' block path bar...	27888	https://twitter.com/dhruv_rathee/status/136414...
111329	मध्यप्रदेश में निजी व्यापारी 200 करोड़ का धान ...	25824	https://twitter.com/RakeshTikaitBKU/status/136...
60721	लहरों को ख़ामोश देख कर ये ना समझना कि समंदर मे...	19284	https://twitter.com/sherryontopp/status/136189...
29510	हाँ मैं जानता हूँ कि मैं शायर नहीं और ज़ुल्म क...	19198	https://twitter.com/sherryontopp/status/136309...
111556	farmer sweet ' see amandacerni rihanna 🖤 🥺	17325	https://twitter.com/jedijasmin_/status/1360162...
24160	कलियुग है साहब यहाँ झूठे को स्वीकार किया जाता ...	15582	https://twitter.com/sherryontopp/status/136337...
108072	bollywood betray panjab amp farmer india hero ...	12949	https://twitter.com/RaviSinghKA/status/1360260...
83016	मेरी दोस्ती का फ़ायदा उठा लेना क्योंकि मेरे वि...	12782	https://twitter.com/sherryontopp/status/136113...
69436	गढ़ी सांपला की पंचायत में पधारे सभी किसानों का...	12317	https://twitter.com/RakeshTikaitBKU/status/136...
89423	wish fli delhi border look tough time farmer f...	12273	https://twitter.com/avinashkalla/status/136089...

We ranked the tweets based on the number of likes and retweets. The analysis shows a high correlation between the two rankings, with 7 out of the top 10 tweets in both categories being the same. Therefore, a tweet with many retweets is likely to have more likes and a tweet with many likes is likely to have more retweets.

3.5. WordCloud of the most frequent words

The most frequent words make quite a lot of sense if we consider the main topic of these tweets, "Farmers Protests of 2021." Therefore, we can see that 'farmer' is the most significant word, apart from 'support', 'protest', and 'india'. Additionally, it could be deduced that these protests took place in India if it was the case that we don’t have this information.

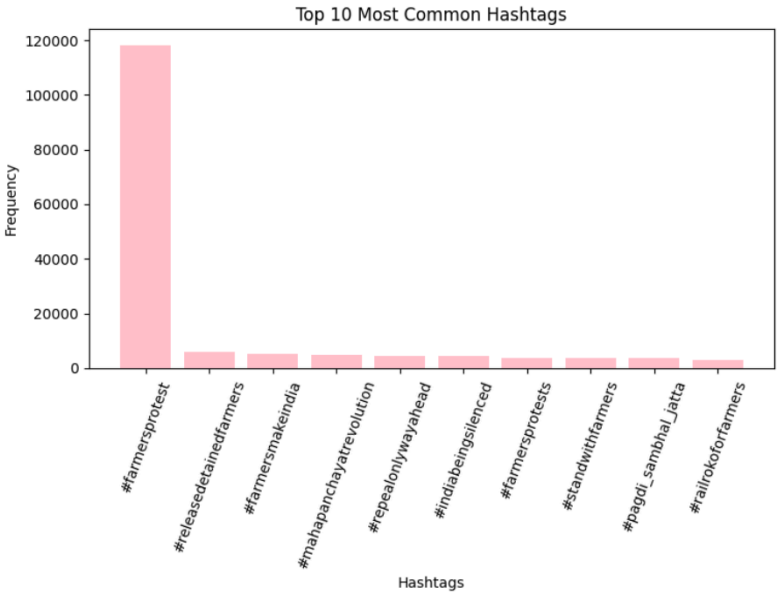
3.6. Hashtags analysis

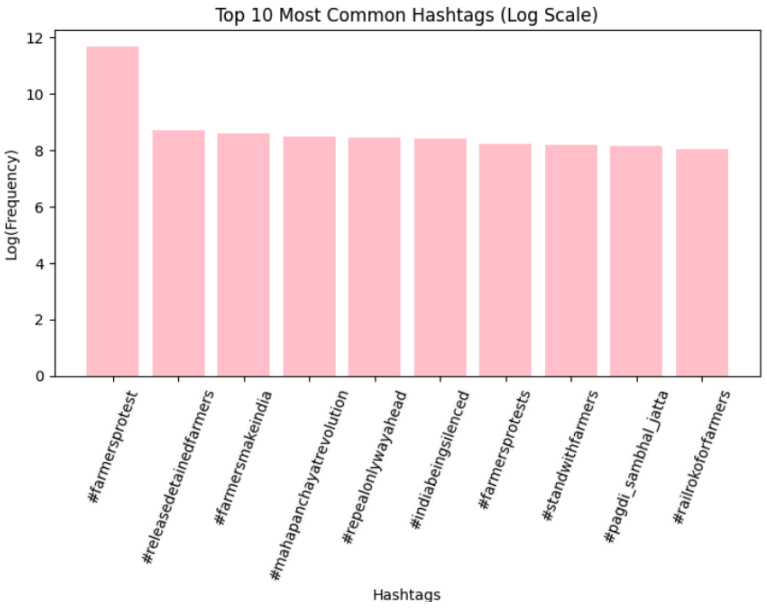
Hashtags play a crucial role in tweets visibility and engagement. Analyzing the most frequent hashtags we saw that #farmersprotest is by far the most used. This confirms the main theme of the dataset.

Clàudia Morales - 199906
Roger Sola - 199780
Queralt Zamora - 199903
G102-2

Top 10 Most common hashtags

	Hashtag	Count
0	#farmersprotest	118114
1	#releasedetainedfarmers	5998
2	#farmersmakeindia	5346
3	#mahapanchayatrevolution	4863
4	#repealonlywayahead	4617
5	#indiabeingsilenced	4506
6	#farmersprotests	3747
7	#standwithfarmers	3652
8	#pagdi_sambhal_jatta	3550
9	#railrokoforfarmers	3102

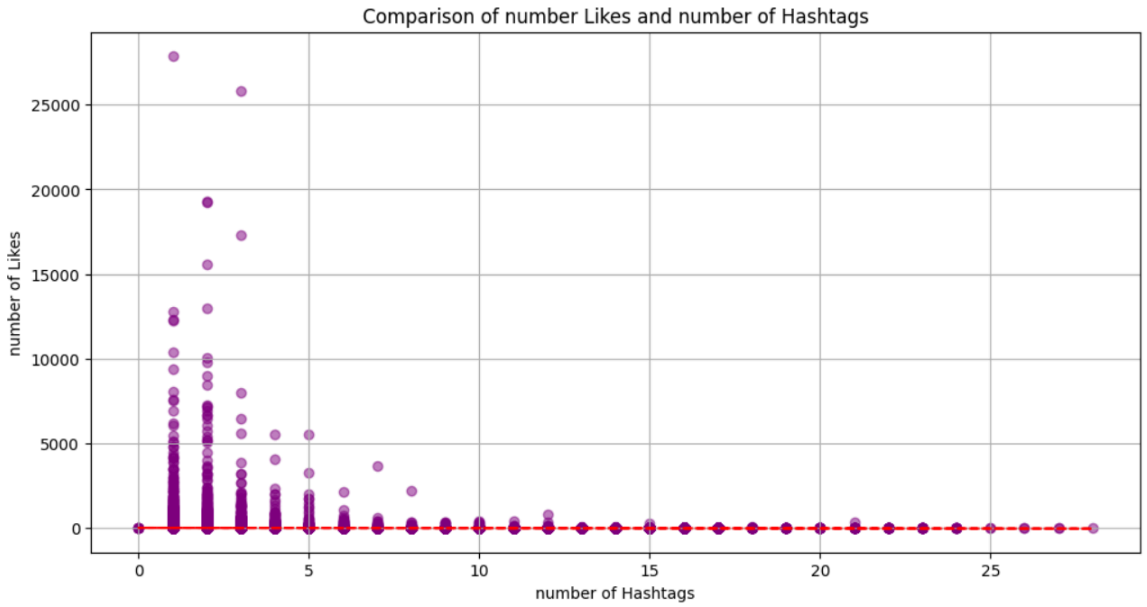




Moreover, due to the scale used in the plot, it is really difficult to see the differences in popularity among the remaining hashtags. So we applied a logarithmic scale to enhance visual clarity and make the differences clearer.

3.7. Relation Between number of Likes and number of Hashtags

It is believed that tweets with more hashtags tend to receive more likes. In other words, a higher number of hashtags is associated with a greater number of likes.



As observed, our initial expectation was incorrect. Tweets with more likes tend to have fewer hashtags, and the number of likes decreases as the number of hashtags increases.

4. Conclusions

In Part 1 of the project, we successfully preprocessed a dataset of tweets related to the 2021 Farmers' Protests, ensuring the data was clean and ready for analysis. Through exploring the data, we discovered that most tweets were short, with a diverse vocabulary, and that tweets with more likes often had more retweets, and vice versa. The hashtag #farmersprotest was the most common and contrary to our expectations, tweets with more hashtags tended to have fewer likes. These observations provide a clear foundation for future analyses of the dataset.