University of Glasgow | School of Computing Science

Assessed Coursework

COMPSCI5078 / COMPSCI5107: Web Science MSc - 2023/24

Individual Coursework: Geo-localisation

Name: Ganesh Sankar
GUID: 2935548S

Coursework Part 1: Geo-localization

1. Introduction:

This report explores the process of converting the given Geo-tagged Twitter dataset and distributing them in a grid over the city of London. The given grid is then populated with the frequency of the tweets that belong to that particular cell. The resulting grid data is further analyzed to retrieve insights on many aspects of Geo-localization techniques such as the total number of tweets, geographical distribution, average tweet per km^2, etc. This analysis is then interpreted to gain information along with potential challenges associated with such Geo-tagged datasets.

2. Pseudo Code:

Haversine distance method:

```
def computeDistance(self, lat1, lon1, lat2, lon2):
     # Method accepts 4 parameters that contains latitude and
     # longitude values of two coordinates.

     # Returns the haversine distance of the coordinates.
```

Method to create a grid based on two points of a rectangular area.

```
def createGrid():
     self.rows = computeDistance(coord1,coord2) for row distance
     self.columns = computeDistance(coord1,coord2) for column distance
     self.noofGrids = rows * columns

     # TO GET OFFSET COORDINATES FOR ROW AND COL POINTS
     for each i in range(rows):
          rowPoints.append(row_lat_coord + i * latOffset)
     for each j in range(columns):
          colPoints.append(col_lon_coord + j * lonOffset)
```

Algorithm to create grid data of the number of tweets:

To convert the given tweets data to their respective grids based on the location, all the JSON data are concatenated into list and stored in a variable called JSON_DATA.

This list is then iterated and for each tweet, the latitude and longitude values are extracted. These coordinates are then compared to the established grid with the createGrid() function mentioned above. If the location of the tweet falls under a particular grid, an np array of the grid representation value is incremented.

Finally, the compiled grid representation of all the incremented values is returned as an np array, which is then used for the representation and visualization of the data.

The Pseudo code for this algorithm to convert the raw tweet data into a meaningful grid representation of geo-data is given below.

```
def createTweetGrid():
    locations = np.zeros([len(row), len(column)])

    for tweet in JSON_DATA:
        tweet_lon = tweet['coordinates'][0] # GET LONGITUDE OF TWEET
        tweet_lat = tweet['coordinates'][1] # GET LATITUDE OF TWEET

        # INITIALIZE EMPTY ARRAY TO STORE COORDINATES
        np_lat = np.empty([grid_creator.rows])
        np_lon = np.empty([grid_creator.columns])
        lat_index = 0
        lon_index = 0

        # LOOPING THROUGH ROW DATA TO FIND TWEET GRID LOCATION
        for i in range(len(rowPoints)):
            if rowPoints[i] >= tweet_lat:
                lat_index = i-1
                break

        # LOOPING THROUGH COLUMN DATA TO FIND TWEET GRID LOCATION
        for i in range(len(colPoints)):
            if colPoints[i] >= tweet_lon:
                lon_index = i-1
                break

        locations[lat_index, lon_index] += 1

     return locations
```

3. Statistical Interpretation of the given data:

I will now discuss the different statistical metrics of the dataset and explore different perspectives of the dataset.

After applying the grid algorithm using the haversine distance formula we are presented with a resulting 2D grid of shape (59, 48). This grid contains the total number of tweets in each cell based on the provided grid algorithm. This representation of the given data reveals critical information on the data distribution, total number of data points, and geographical distribution of the data.

This grid representation is then used to convert the given tweets and their location to be inserted as part of the grid resulting in a grid of size (59, 48) with each cell containing data on the number of tweets that belong to that particular cell.

Given below are some statistics on the resulting data and my interpretation of it:
Total tweets across the entire geographical grid: **13192**

Maximum number of tweets in a cell: **4326**

Non-zero value density: **0.07062146892655367 (~7.06%)**

The average number of tweets per grid including zero values: **4.66**
The average number of tweets per grid with non-zero values: **56.84**

Standard Deviation of the number of tweets across all the non-zero grids: **322.56**
The top 5 cells with the highest number of tweets and their coordinates:

Rank	No. of Tweets	Coordinates
1	4326	51.51326638983051, -0.12374999999999997
2	1176	51.50606786440678, -0.14132
3	607	51.470075237288135, -0.08860999999999997
4	427	51.49886933898305, -0.14132
5	406	51.50606786440678, -0.10618

Interpretation:
The data above shows that the most concentrated cell on the grid with 4326 tweets. This grid is located at the center of the London city.

A more in-depth examination of the top 5 most concentrated grid cells reveals their collective representation, amounting to a substantial 60% of the entire geo-tagged dataset. This is again confirmed by the result of non-zero value density. This suggests that only around 7.06% of cells in the given grid contain a value other than zero.

4. Visualization:

The 2D grid data with the number of tweets per grid can be effectively visualized using a heatmap as shown in Fig 1.1. The heatmap is log normalized since the data is extremely concentrated in just a couple of cells making the rest of the data not be displayed on the chart.

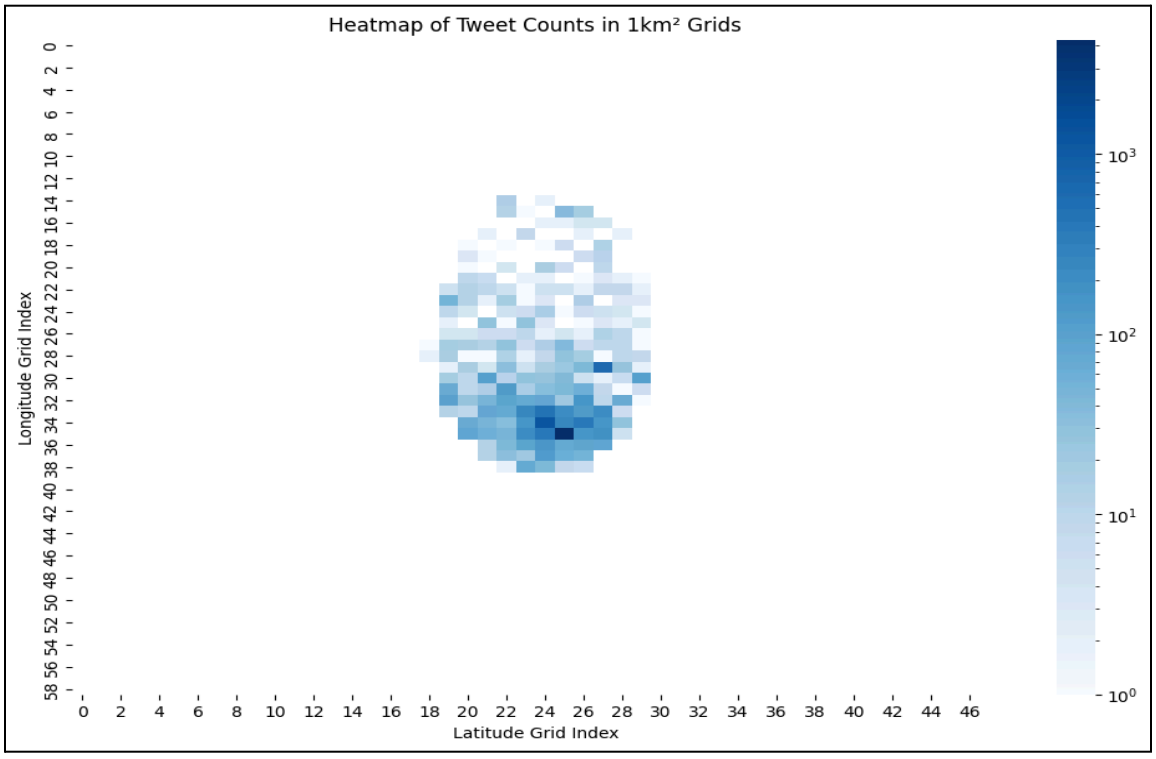

Fig 1.1: Heatmap of given tweet data in a 1 km^2 Grid

This representation of the grids can further be improved by overlaying an actual map of London on the displayed heatmap as shown below in Fig 1.2.

Note: The data looks inverted as the geo location is synced with the actual output. This is needed as matplotlib starts the grid from the top left corner but our data requires it to be from the bottom left corner.

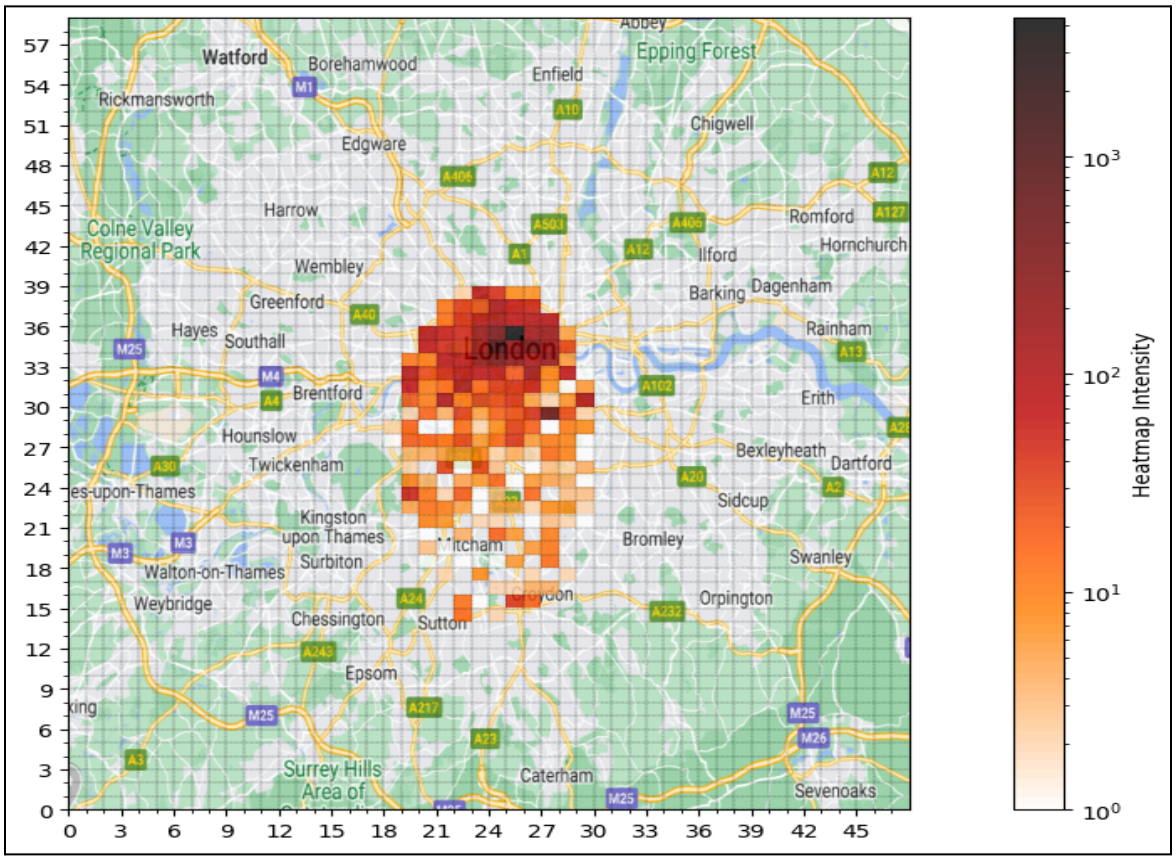

Fig 1.2: Grid heatmap with Google Map overlay

The location data clearly displayed the distribution of the tweets across the city of London. The image in the background is cropped in a way for proper alignment with the starting coordinates of the given grid data. The concentration of tweets in and around the city center becomes evident along with the increasing sparsity of data when moved away from the center of the city.

To understand the latitude and longitudinal distribution of the dataset, the row-wise and column-wise distribution is displayed in the figure Fig 1.3.

This shows that the number of tweets peaks around [35, 25], and most of the data is concentrated in just a couple of latitude and longitude values around that area.

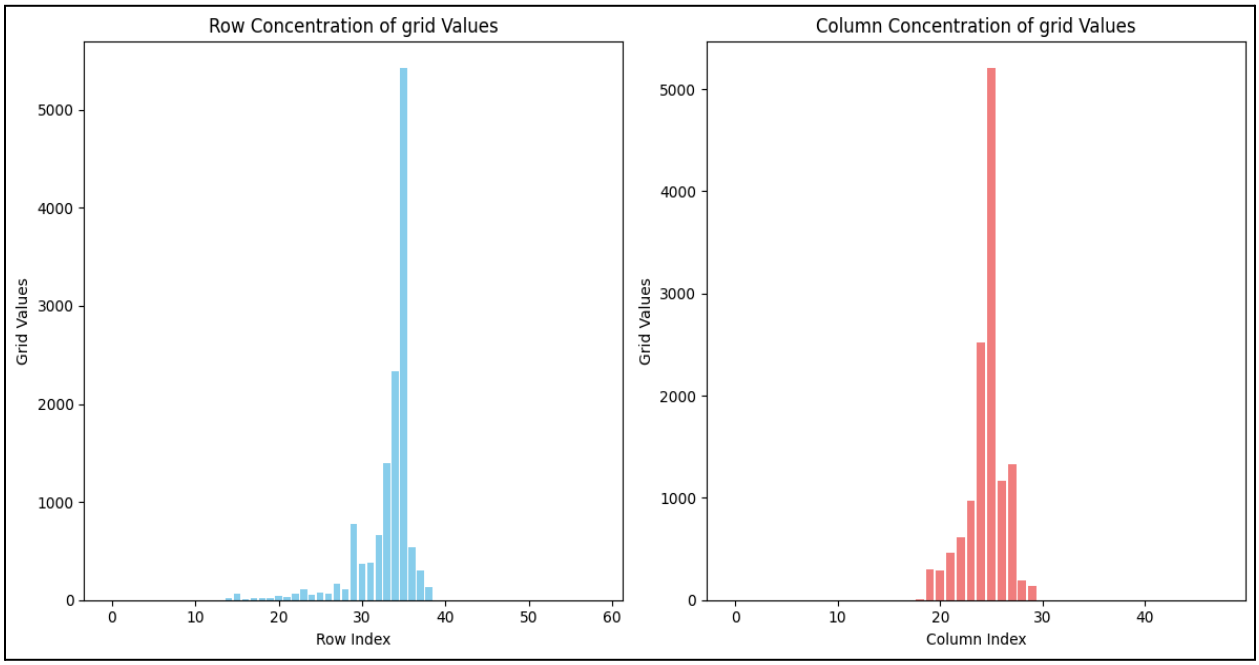

Fig 1.3: Row and Column Concentration of Tweets

5. Data Interpretation:

From both the raw data statistics and the visualizations, the following can be observed in the given dataset and grid mapping.

- The concentration of 4362 tweets within a single grid cell denotes the increased Twitter activity in that region.
- This highly active area might not be entirely accurate, as this could be a default coordinate set by Twitter when the user sets the location as London, UK.
- The top 5 most concentrated grid cells collectively contributing to a substantial 60% of the grid data suggest the domination of this location hot-spot compared to other locations.
- The density analysis and average tweet per grid metrics reveal the localized nature of the tweets in the grid, which is confirmed by the visualizations.
- The high median value suggests the data is varied and skewed in an extensive manner.
- Since the concentration of tweets is localized to popular places in the city center, there is a high chance of those places being tourist locations. This can be confirmed by the tweets posted by users experiencing cafes and tourist spots in and around the city center.

Though the information revealed by the data seems to be accurate for the most part, there are some valid geo-localization issues in processing such data.

- Users manually set location coordinates when posting tweets, often leading to a tendency to generalize the tweet's location to central London. This practice introduces data disparity, increasing the chance of data hot spots.
- Using geo-tagged data requires a delicate balance between getting a highly accurate location and generalizing it to a broader range. This process is crucial to mitigate privacy concerns associated with revealing precise user locations while still preserving the integrity of the data accuracy.
- The data may exhibit spatial bias, where certain regions or urban centers are overrepresented due to a higher concentration of Twitter users. This can lead to an inaccurate representation of social media activity across diverse geographical areas.
- The presence of spam or bot accounts posting geo-tagged tweets could result in a biased dataset.
- Based on the use case, external events such as concerts or other gatherings might affect the quality of the data.

6. Conclusion:

The given Twitter data was analyzed across the given grid coordinates of the city of London with statistical data and representation of crucial geo-localization information. The data distribution was documented along with geo-tagged metrics using a heatmap. Further data interpretation was done along with potential issues concerning Geo-tagged datasets.

Coursework Part 2: Newsworthiness Scoring

1. Introduction:

In this report, I will outline the process of developing a newsworthy scoring method based on the given dataset of high-quality, low-quality, and background tweets. The given data consists of tweet details scrapped from Twitter. The parameters of interest are the id, text, and qualityScore of a particular tweet. Both the high-quality and low-quality tweets are classified based on having a quality score greater than 0.6.

2. Algorithm/ Pseudo Code:

The initial step is to convert the given raw JSON data into a Python JSON object and store them as a list for each dataset (HQ, LQ, and BG tweets).

To tokenize the given tweets in both high-quality and low-quality data, I have employed spacy. spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. The below pseudo code defines a method `text_pipeline_spacy_special()` where it tokenizes the word and includes or excludes stop words based on the parameters.

```
def text_pipeline_spacy_special(text, include_stopwords=False):
    tokens = []
    doc = nlp(text)
    for t in doc:
        if not t.is_punct and not t.is_space
           and (include_stopwords or not t.is_stop):
            tokens.append(t.text.lower())
    return tokens
```

Now using the above-defined tokenizer, the term frequencies are calculated for the HQ, LQ, and BG tweet datasets as follows.

```
# Calculating Term Frequency for the Background Quality Model
BGterms = []
for tweet in tqdm(BGdata):
    BGterms += tweet['text']
bgTF = dict(Counter(BGterms))

# Calculating Term Frequency for the High Quality Model
HQterms = []
```

```
for tweet in tqdm(HQdata):
    terms_local = text_pipeline_spacy_special(tweet['text'])
    HQterms += terms_local
    tweet['terms'] = terms_local
hqTF = dict(Counter(HQterms))

# Calculating Term Frequency for the Low Quality Model
LQterms = []
for tweet in tqdm(LQdata):
    terms_local = text_pipeline_spacy_special(tweet['text'])
    LQterms += terms_local
    tweet['terms'] = terms_local
lqTF = dict(Counter(LQterms))
```

The term frequencies of HQ, LQ, and BG data are stored in variables hqTF, lqTF, and bgTF respectively. Using the term frequencies, the total number of terms is also calculated as Fhq, Flq, and Fbg.

Along with the term frequencies and total terms in each dataset, the relevance of a term is calculated by the probability of the term belonging to the high-quality dataset terms and low-quality dataset terms. This is defined in the functions `computeRhqt(term)` and `computeLhqt(term)`.

```
def computeRhqt(term):
    return (hqTF.get(term, 0) / Fhq) / (bgTF.get(term, 1) / Fbg)

def computeRlqt(term):
    return (lqTF.get(term, 0) / Flq) / (bgTF.get(term, 1) / Fbg)
```

To calculate the newsworthiness scores for high and low-quality tweets, we iterate through the tweet, evaluating the significance of each term using the functions `computeRhqt(term)` and `computeLhqt(term)`. The scores are then aggregated for terms greater than the threshold of 2, and finally, the newsworthiness score is calculated using a log transformation given below.

```
HQdata_nScores.append(math.log2((1+Shqt) / (1+Slqt)))
```

```
LQdata_nScores.append(math.log2((1+Shqt) / (1+Slqt)))
```

This is done for both the HQ and LQ data as shown in the pseudo-code below:

```
HQdata_nScores = []
for tweet in tqdm(HQdata):
    Shqt = Slqt = 0
    for term in tweet['terms']:
        Rhqt = computeRhqt(term)
        Rlqt = computeRlqt(term)
        Shqt += Rhqt if Rhqt > 2 else 0
        Slqt += Rlqt if Rlqt > 2 else 0
    HQdata_nScores.append(math.log2((1+Shqt) / (1+Slqt)))

LQdata_nScores = []
for tweet in tqdm(LQdata):
    Shqt = Slqt = 0
    for term in tweet['terms']:
        Rhqt = computeRhqt(term)
        Rlqt = computeRlqt(term)
        Shqt += Rhqt if Rhqt > 2 else 0
        Slqt += Rlqt if Rlqt > 2 else 0
    LQdata_nScores.append(math.log2((1+Shqt) / (1+Slqt)))
```

The resulting newsworthiness scores of both HQ and LQ dataset is combined and stored as pandas dataframe with their respective newsworthiness score for further analysis and visualizations.

3. Data Analysis:

For the initial round of analysis, a threshold metric is implemented for calculating Shqt and Slqt. This value is used to bin together newsworthy scores based on the threshold, or else it returns zero. By fine-tuning this threshold, we are able to analyze the overlapping of low-quality and high-quality data and their newsworthiness as shown in the below diagrams.

These diagrams indicate that a balance needs to be established in choosing a threshold so that not a lot of outlying newsworthy tweets get termed as non-newsworthy. As we increase the threshold, we can clearly see the number of tweets with a newsworthy score of zero increases, along with that, we also get a clear distinction between HQ and LQ data. Based on the charts, a threshold of 4-5 seems to be a good value to classify the newsworthy score calculation.

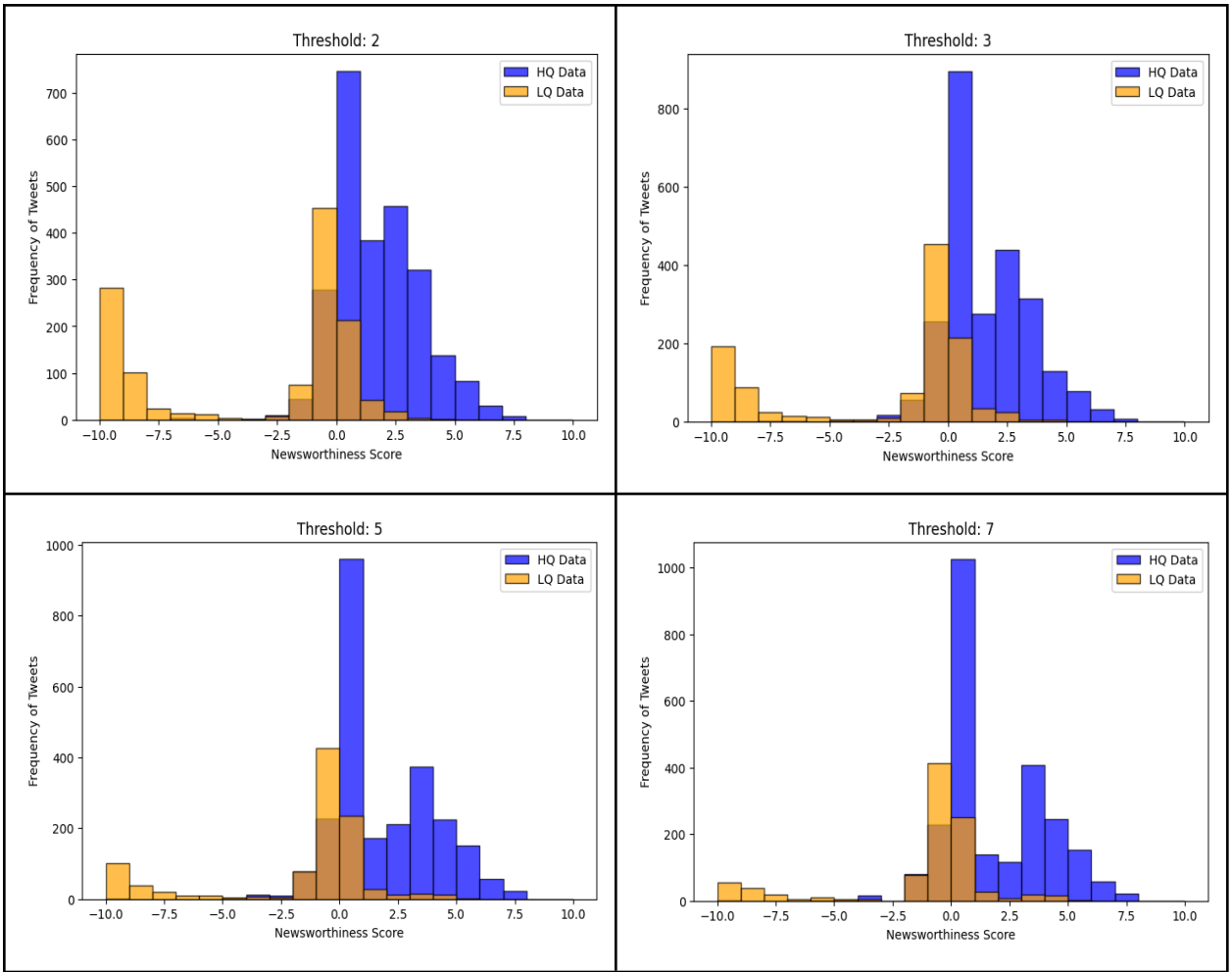

Fig 2.1: Comparison of HQ and LQ newsworthy scores on different thresholds for the scoring algorithm

By having a threshold of 5, the HQ data and LQ data are combined along with their corresponding newsworthy scores to print out the following statistics defining the data distribution.

Metrics	**Value**
Total Count	4028
Mean	-0.9439
Standard Deviation	5.327
Median	0.1712

This metric reveals that the data showcases a wide range of values with notable spread among the data.

Additionally, analyzing the effect of using stopwords during the tokenization process will also play a major role in affecting the desired results. To achieve this, the newsworthy scores are calculated for both types of data (with and without stop words) and compared using visualization techniques to find insightful information.

This comparison is shown in Figure 2.2. Here, the charts are printed side by side for both HQ and LQ tweets data with stop words and without stop words.

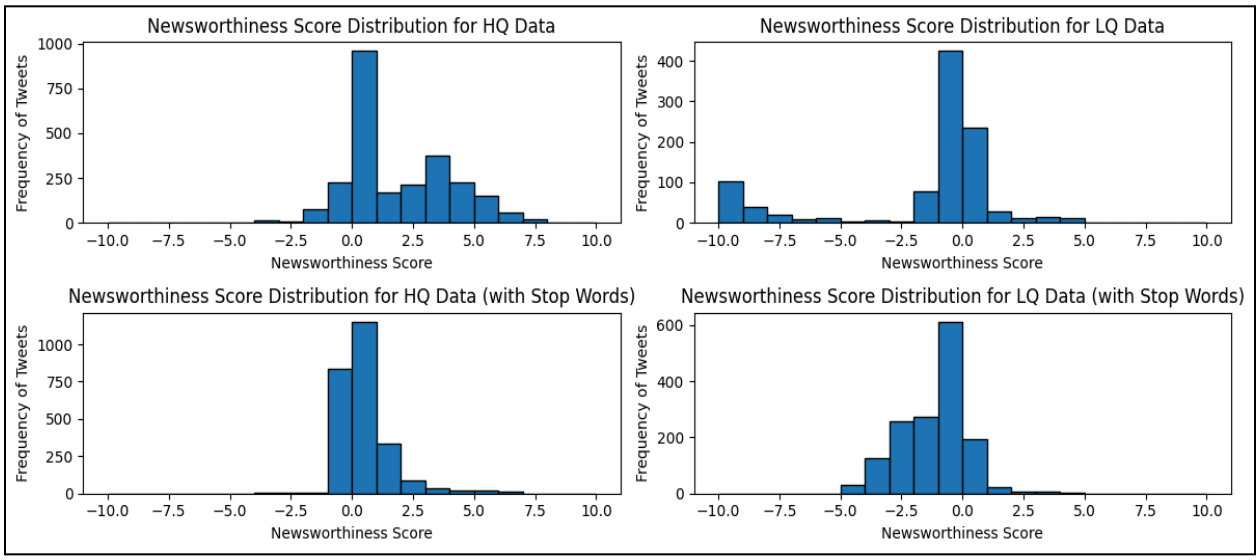

Fig 2.2: Comparison of HQ and LQ data newsworthiness with and without stop words

The comparison reveals the data distribution when stop words are included. Including stop words makes the distribution smoother and removes outlier data from the LQ data set with a newsworthy score of < -5. This is further confirmed by the box plot representation of the same data shown in Fig 2.3.

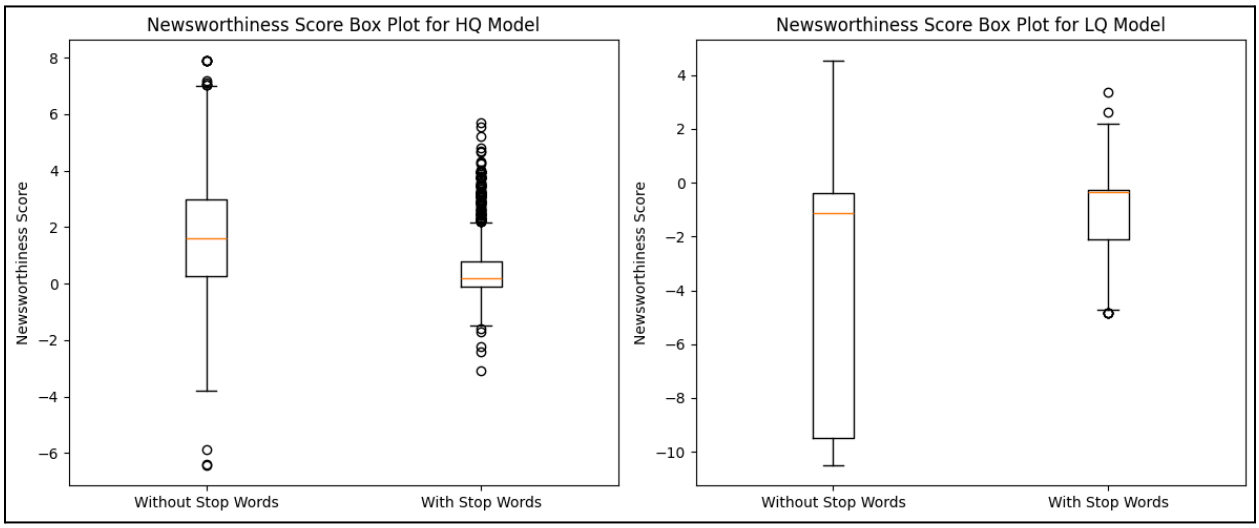

Fig 2.3: Box plot representation of newsworthiness score of HQ and LQ data with and without stop words.

4. Conclusion:

The analysis of newsworthiness score of high and low-quality data revealed crucial information on the newsworthiness distribution. This distribution was further studied by adjusting and fine-tuning thresholds in the scoring algorithm. Additionally, the data was calculated including stop words for tokenization which resulted in a less varied and closely distributed dataset compared to the tokenizer that excluded stop words. This suggests that the choice of tokenization strategy, particularly the use of stop words, plays a significant role in shaping the distribution and characteristics of the newsworthiness scores.

Coursework Part 3: Newsworthiness Scoring for Geo-tagged Data

1. Introduction:

In this part of the coursework, I will outline the use of the scoring algorithm defined in Coursework 2 to calculate the newsworthiness of the Geo-tagged dataset given in Coursework 1. The objective is to empirically analyze the given tweet data and their newsworthiness score and find an acceptable threshold.

2. Empirical analysis and statistics of data:

In this section, I will discuss the various thresholds that were tested and the resulting tweets that fall under that threshold.

For the given Geo-tagged data, the same newsworthy score was applied, so the results could be compared in each and every step. Here, the threshold was varied from 0 to 5 for the newsworthiness score to analyze the empirical data.

To achieve this, a threshold is chosen and the original data is filtered based on this newsworthiness threshold. From the filtered data, a random amount of data is sampled to check for any potential misfits for the given newsworthiness scores. Given below are a few examples of such analysis.

nScore > -1
This is shown as an example of what a tweet with a newsworthiness score of less than 0 will be. As we can see in the below table, It mostly consists of personal tweets with little to no information, hence the low newsworthiness score.

nScore	Tweet
-0.999335	Just posted a photo @ All About Eve https://t.co/2okWqCWtJq
-0.999335	Just posted a photo @ Steven Hitchcock Savile Row Bespoke https://t.co/e8auyTSGcP
-0.999335	Just posted a photo @ Norbury https://t.co/WvcNykSW33
-0.999335	Just posted a photo @ Inner Temple https://t.co/u9nqznmoZL
-0.999335	Just posted a photo @ Bankside https://t.co/t5sDWdDmLk

nScore > 0.5

With a newsworthiness score of 0.5 or greater, we get to see some meaningful tweets. Even though the newsworthiness score is above 0, we can see some adverts, scams, or other personal tweets. This is caused due to the data terms that were considered high quality or low quality in the given dataset.

nScore	Tweet
0.5	Before food / after food\n\n @gemmaturnbullphotos \n\n#actor #casting #headshot #acting #actorslife #cinema #entertainment #film #london #movie #theatre #tv @ London, United Kingdom https://t.co/kI57TqvOya
0.501380	Party wall @ Barbican, City Of London https://t.co/pFwEoeoq8T
0.503201	Shana tova #ChouekaTheBaker #challah #roshhashanah @ London, United Kingdom https://t.co/xYCyZFC35M
0.503201	#weebri 'Lush As You Fancy' #lushandlovely #oilonboardpainting it's the full #doubledouble #piemashandliquor of #oilpainting satisfaction guaranteed.... @ London, United Kingdom https://t.co/BvjigLNi0M
0.503873	Monday: 19th Sept 2022 - London bustling to send off Queen Elizabeth II #londonlife #ripqueenelizabeth #queenelizabeth #queenelizabethii #britishmonarchy #onthisday @ London, United Kingdom https://t.co/t6N0yDx4Sb

nScore > 0.758

When the threshold is finetuned at around 0.758, we can see that most of the tweet seems to be newsworthy. In this particular example, some tweets are about approval for a construction, a job offer, and a radio station news. All of these can be considered newsworthy. But there can be some edge cases such as tweets 3 and 4 which seem to be a personal tweet wishing someone for their birthday and an advert for shirts and dresses.

nScore	Tweet
0.758581	New Approval of Details planning application at 89 Plender Street London Camden NW1 0JN. Registered on 22 September 2022.\n\nhttps://t.co/nCwYcup3oB https://t.co/Q0NQmn2xRX
0.760523	This job is now open at Starbucks in England. Follow us and turn on mobile alerts to hear about jobs like these as soon as they're posted: Barista - Store# 12385, VICTORIA - PALMER ST #Retail
0.760801	Happy Birthday Renu Hossain Ji best wishes World Tabla Council\nhttps://t.co/caqYmhf0XN @ England, UK https://t.co/F2BR8Y6cgM
0.761214	That perfect TSD..Trans Seasonal Dress..wear it 3 ways...\nGanni X Levi's Shirt Midi Dress..\n\n1. Stand alone\n2. Layered for extra warmth\n3. Open as Coat or Jacket @ London, United Kingdom https://t.co/fku98c2lvg

0.761764	A quick #qso with #M0JXS @icom_uk #ic705 #amateurradio #amateurradiooperator #hamradiooperator #hamradio #London #VHF

Statistical Data

To understand the distribution of the data with respect to the newsworthiness score, a histogram is used to visualize this data as show in Fig 3.1.

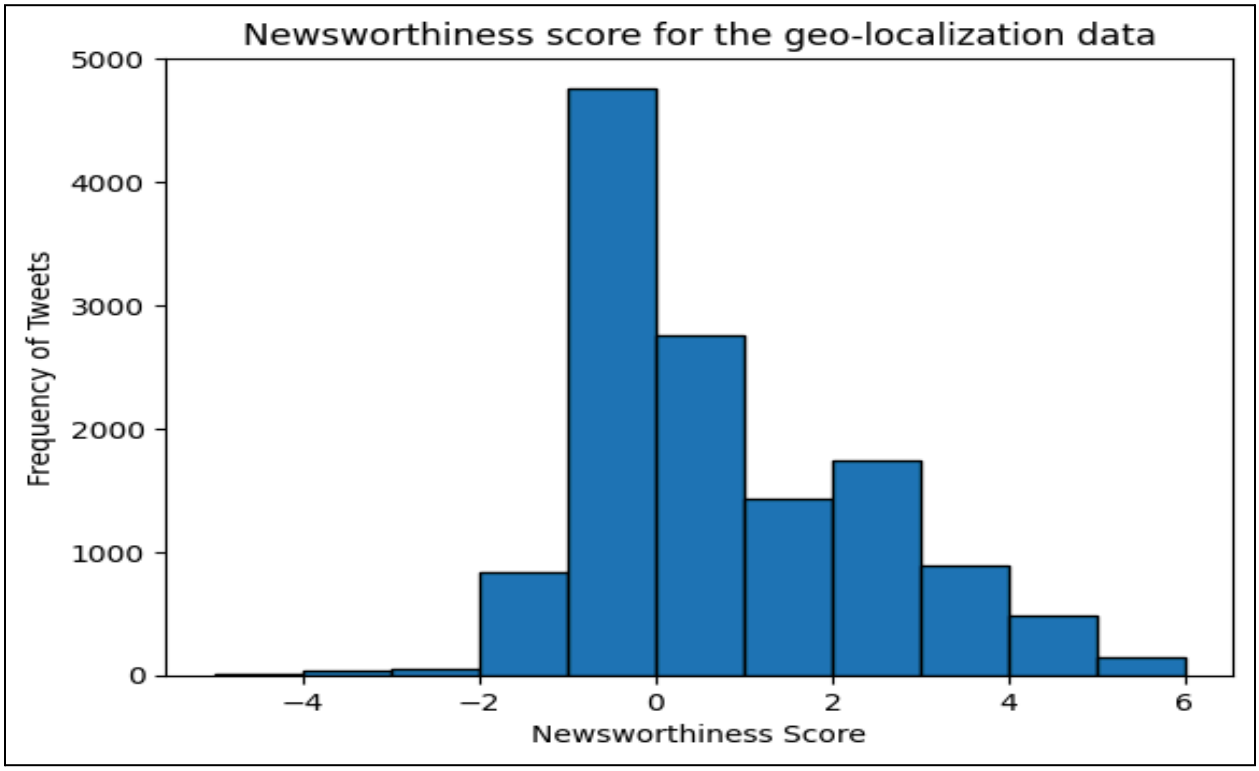

Fig 3.1: Frequency of tweets against Newsworthiness Score

As seen from the above chart, approximately more than 4800 tweets have a newsworthiness score of less than 0. This also confirms the conclusion made in Coursework 1 as most of the data were localized to central London. These tweets might be tourists posting pictures on Twitter which will get filtered out by the newsworthiness scoring.

With the applied threshold of 0.758 to categorize tweets as newsworthy, The statistical analysis of the corresponding data is shown below.

Metrics	Values
Total tweets with a threshold of 0.758	5102
Maximum number of tweets in a single cell	714
Density of non-zero values in the grid	0.0614
The average number of tweets per grid including zero values	1.801
The average number of tweets per grid excluding zero values	26.918
Standard deviations of no. of tweets (non-zero)	70.97

3. Visualization Comparison:

To compare the Geo-tagged data with the newsworthiness filtered data, I have implemented a heatmap to illustrate the variation in data after filtering using the threshold. Additionally, the statistical data is also compared to the original Geo-tagged data.

Initially, the below figure (Fig something) shows the comparison of the original Geo-tagged data against the newsworthiness threshold filtered data with a threshold of **1.**

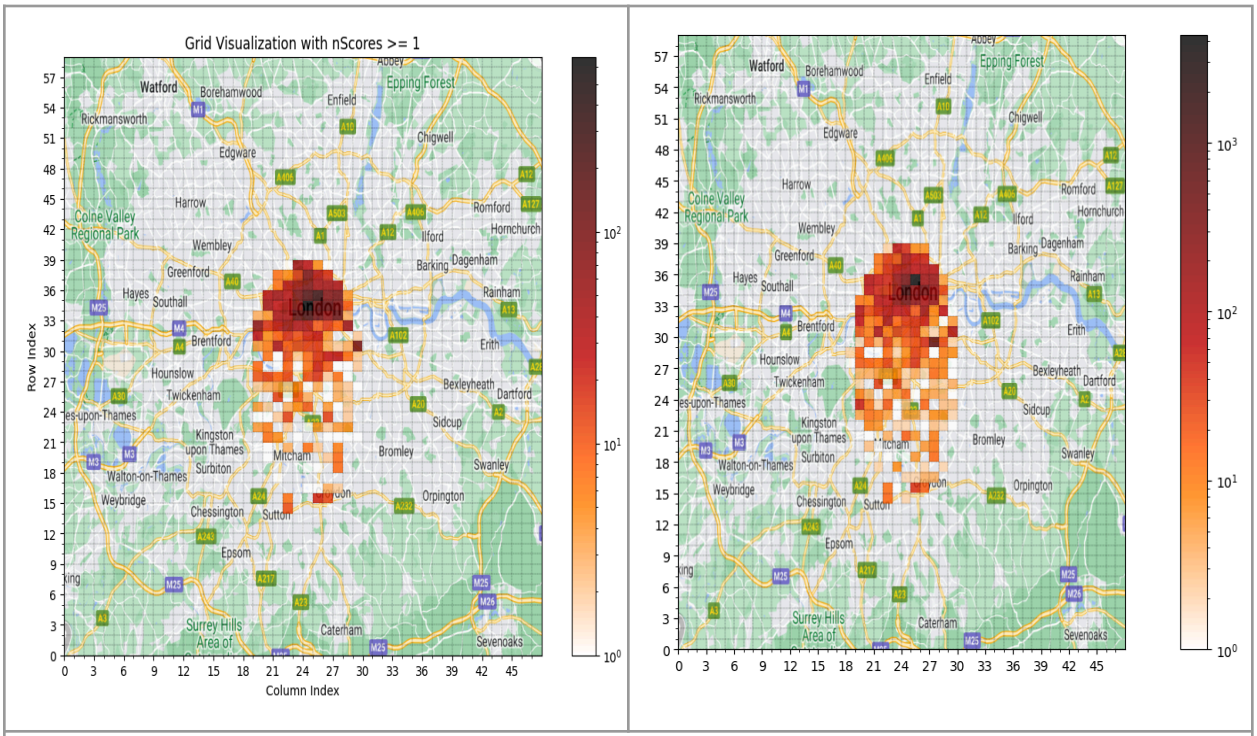

Fig 3.2: Comparison of a heatmap of Geo-tagged data with newsworthiness threshold >= 1 (left) against the original dataset (right).

This comparison reveals that there is a much lower concentration of data overall in the grid as shown in the LogNorm color bar. Though there are less number of data points, the basic distribution remains the same.

To see the variation involved in both the dataset, I will compare the statistical data as shown below.

Metrics	**Values (threshold = 1)**	**Values (Original Data)**
Total tweets across the grid	4736	13192
Maximum number of tweets in a single cell	671	4326
The density of non-zero values in the grid	0.0607	0.0706
The average number of tweets per grid including zero values	1.672	4.658
The average number of tweets per grid excluding zero values	25.737	56.838
Standard deviations of no. of tweets (non-zero)	65.003	322.55

The variation in the data is obvious when the statistical data is compared. Here, the number of tweets filtered out is more than half compared to the original data. A change in frequency variation was also noticed resulting in a smooth spread of data with much less variation compared to the original data denoted by the standard deviation comparison.

To experiment with the threshold, and to see a noticeable difference in the visualization comparison, the threshold is increased to **4**. As shown below in the Figure 3.3.

Here, we can see the visual comparison is enough to differentiate the distribution of the Geo-tagged data with a newsworthiness threshold of > 4 against the original data.

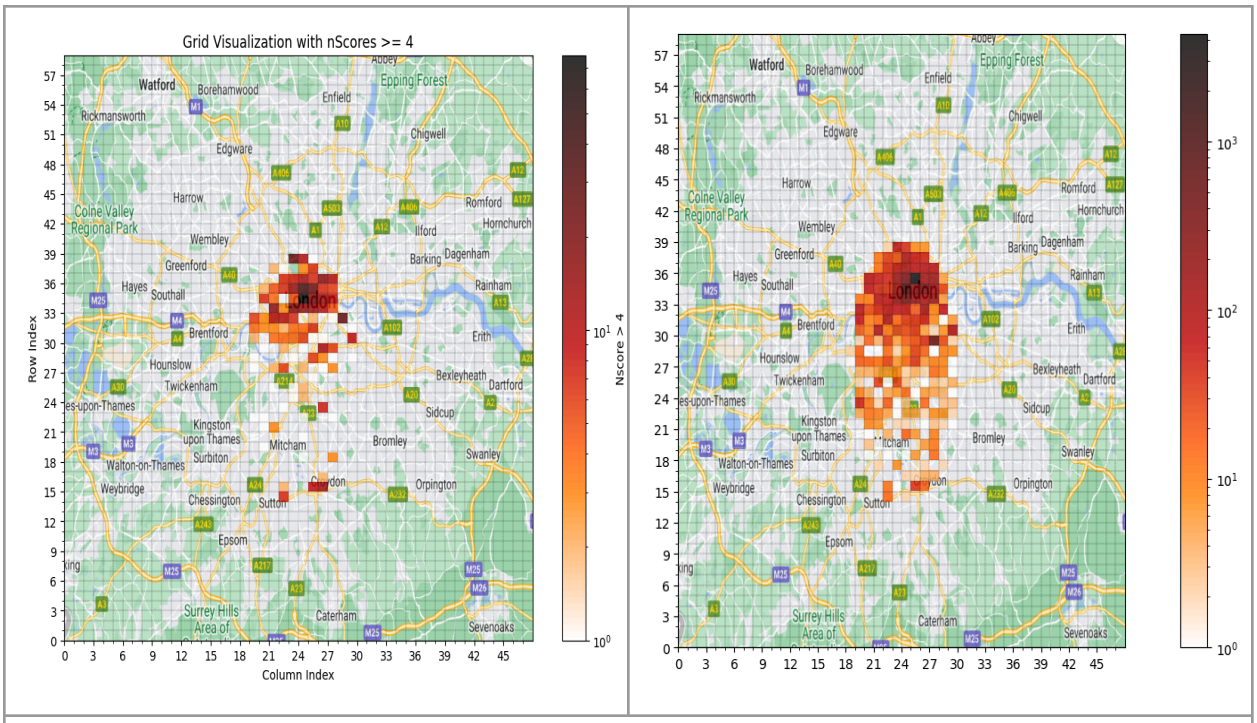

Fig 3.3: Comparison of a heatmap of Geo-tagged data with newsworthiness threshold >= 4 (left) against the original dataset (right).

4. Conclusion:

In conclusion, the analysis of various newsworthiness thresholds revealed distinct patterns in the filtered Geo-tagged dataset. As the threshold increased, the content shifted from personal or advert tweets with minimal information to more meaningful and potentially newsworthy tweets.

A fine-tuned threshold of 0.758 demonstrated a notable improvement in capturing relevant tweets that were newsworthy when analyzed using empirical analysis. Further comparisons were made to compare the filtered data against the original dataset along with statistical metrics. The visual comparison at a threshold of > 4 further emphasized the differences in the distribution of the newsworthy tweets. This calls for a more robust balance of newsworthiness and relevance of the tweet when determining the threshold and their impact on real-world data representations.

Coursework Part 4: Open Tasks

Identify and discuss, with examples, issues for geo-localisation due to the nature of tweets or sources

The analysis of the Geo-tagged dataset involved critical issues that needed to be addressed. Some of the issues and key findings are listed below:

1. User-provided data:
When the location data is collected by Twitter, it gives the option to the user to enter the location manually. This involves searching for a particular location and then selecting it to be reflected as a geo-tagged tweet.

This introduces human error into the system. For example, the user might try to enter a street name that is in London but the same street name is also present in a different country and the user accidentally clicks that.

2. Generalization of the location:
Many a time, the users don't bother with providing an accurate location when posting a tweet. If a user is somewhere in the south of London the tweet can be tagged as London, UK. This will result in assigning a default coordinate of London City which is located in central London.

This issue was very evident in the given dataset as well. The concentration of data at the city center suggests that most of the tweets have defaulted to London, UK with a default coordinate.

3. Presence of spam/ scam accounts:
The data scraped from Twitter is prone to different types of spam/ scam accounts and their tweets. Spamming geo-tagged tweets could result in a skewed dataset. As a real-world example, fake users can choose random location coordinates for their spam tweets. This results in the reliability of the geo-tagged data.

4. External event or gathering:
Similar to scam tweets, any event organized that attracts a lot of crowds could also impact geo-tagged datasets. In this case, many users going to a concert might post tweets that are geo-tagged to that particular location, affecting the overall distribution of the data.

This particular issue goes hand in hand with temporal data. Which affects the type of data you get based on the time of year (Seasonality of data collected).

5. Inconsistent locations data:
When the user is responsible for geo-tagged data, the input may vary according to the format of the locations used. For example, both UK, London and London, UK represent the same

location, but the coordinates might differ for different formats resulting in an inaccurate data aggregation when collecting data into grid squares like it is done in this coursework.

In conclusion, the complexities associated with geo-localization originate from various possible challenges. These challenges are required to be addressed as this might affect the reliability and accuracy of the data and impact the expected results.

These issues can be isolated and solved using different pipelines and require an in-depth understanding of language, temporal variation of data, user behaviors, etc.