Analysis on Public Perception Towards LSE on Reddit

Team member:

- Hsia-Ying(Cherrie) Chen H.Chen68@lse.ac.uk (mailto:H.Chen68@lse.ac.uk)
- Lina Ou l.ou2@lse.ac.uk (mailto:l.ou2@lse.ac.uk)
- Jinhao(Jaden) Wu j.wu88@lse.ac.uk (mailto:j.wu88@lse.ac.uk)

Github repository: link (https://github.com/st101cc/reddit.git)

Table of Contents

Introduction

Motivation

As undergraduate students at the London School of Economics and Political Science (LSE), we want to know what is the public’s general perception about the university we are currently pursuing our degrees in. Through this project, we hope to understand of how reputable LSE is in the public's eyes, and how a degree from LSE could be beneficial in the workforce.

Goal of the study

We hope to answer the following research questions:

1. What are some common traits that the public associate LSE with?
2. What do people on Reddit want to know more about LSE?

Data Acquisition

Reddit posts

To obtain the data required for our analysis, we used reddit API, which allows us to scrape up to 1000 posts from a subreddit. We collected our data on **15 April 2023**, where we managed to scrape 851 new posts from the subreddit "TheLse". For each post, we collected information about its title, content, and score (the number of upvotes minus the number of downvotes).

The code to acquire the data is stored in another notebook `data_acquisition.ipynb` , and the data is stored in the file `data.csv` inside the [github repository (https://github.com/st101cc/reddit.git)](https://github.com/st101cc/reddit.git). The data we got is shown below:

In [10]:

```
import pandas as pd

df = pd.read_csv('data.csv')
df
```

Out[10]:

	title	content	score
0	LSE accomodation for couples	Hi! \r\n\r\nI will be moving with my partner t...	3
1	which accommodation do rich international stud...	where do rich students live during their studi...	0
2	Bankside or Garden Hall	Im a general course student and can't choose b...	2
3	Is the reserve list separate from an ordinary ...	I got placed on a reserve list for the Masters...	2
4	Postgraduate housing	Hello! I was just accepted into LSE for my mas...	2
...	...	...	...
846	Update to waiting for decades + advice for fut...	Hi! I made [a post](https://www.reddit.com/r/T...	12
847	Advice on Teachers Reference for Undergrad App...	As part of my undergrad application to LSE for...	2
848	international undergrad applicants???	a bunch of people i know still haven’t receive...	3
849	Winter/March holidays	So am I wrong or are the winter and March/Apri...	1
850	Differemt price for the same room?	Hi. Was checking room availability at Passfiel...	2

851 rows × 3 columns

Data Cleaning

We performed the following steps to prepare our data for analysis:

1. Handle missing values
2. Convert dataframe column types
3. Lower all cases
4. Remove stop words and punctuations for the title and content of the post
5. Convert words into its dictionary form
6. Found 100 most common words in the title and content of the posts
7. Found 100 most common words in the title and content of the 50 top scoring posts

In [11]:

```
# remove empty values
df.dropna(inplace = True)
df.shape
```

Out[11]:

```
(703, 3)
```

After removing the empty values from the dataframe, there are only 703 posts left to be analyzed.

In [12]:

```
#check dataframe type
df.dtypes
```

Out[12]:

```
title      object
content    object
score       int64
dtype: object
```

In [13]:

```
#make sure the title and content have the correct type
df['title'] = df['title'].astype(str)
df['content'] = df['content'].astype(str)
```

In [14]:

```
#lower all cases
df['title'] = df['title'].apply(str.lower)
df['content'] = df['content'].apply(str.lower)
```

In [15]:

```
# remove stopwords
import nltk
from nltk.corpus import stopwords
sw = stopwords.words('English')

stop_words = set(stopwords.words('english'))
df['title'] = df['title'].apply(lambda x: ' '.join([word for word in x.split() if word.lower()
df['content'] = df['content'].apply(lambda x: ' '.join([word for word in x.split() if word.lowe
```

In [16]:

```
# remove the puntuations
import string

def remove_punctuations(text):
    if isinstance(text, str):
        translator = str.maketrans('', '', string.punctuation)
        return text.translate(translator)
    else:
        return text

df = df.fillna('')
df = df.applymap(remove_punctuations)
```

In [17]:

```
#convert words to its lemma
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

def lemmatize_text(text):
    return ' '.join([lemmatizer.lemmatize(w) for w in word_tokenize(text)])

df['title'] = df['title'].apply(lemmatize_text)
df['content'] = df['content'].apply(lemmatize_text)
```

In [18]:

```
#look at the cleaned data
df = df.reset_index(drop=True)
df
```

Out[18]:

	title	content	score
0	lse accomodation couple	hi moving partner london looking couple accomm...	3
1	accommodation rich international student stay in	rich student live study arab chinese etc	0
2	bankside garden hall	im general course student cant choose two advice	2
3	reserve list separate ordinary waitlist	got placed reserve list master environmental p...	2
4	postgraduate housing	hello accepted lse master wondering accommodat...	2
...	...	...	...
698	update waiting decade advice future applicant lse	hi made a posthttpswwwredditcomrthelsecomments...	12
699	advice teacher reference undergrad application	part undergrad application lse bachelor histor...	2
700	international undergrad applicant	bunch people know still haven ' t received res...	3
701	wintermarch holiday	wrong winter marchapril holiday essentially st...	1
702	differemt price room	hi checking room availability passfield two op...	2

703 rows × 3 columns

Prepare data for analysis

In [19]:

```
# Find 50 most common words for post titles
import collections
all_text_title = ' '.join(df['title'])
title_counts = collections.Counter(all_text_title.split())
common_titles = title_counts.most_common(50)
print(common_titles)
```

```
[('lse', 193), ('msc', 82), ('student', 70), ('application', 44), ('accommodatio
n', 43), ('economics', 38), ('graduate', 36), ('course', 35), ('v', 35), ('bsc',
34), ('summer', 28), ('undergrad', 26), ('offer', 26), ('international', 25), ('a
dmission', 25), ('finance', 25), ('year', 25), ('hall', 24), ('question', 22),
('master', 22), ('help', 21), ('advice', 20), ('support', 19), ('school', 18),
('get', 17), ('undergraduate', 17), ('political', 16), ('general', 16), ('anyon
e', 16), ('program', 16), ('chance', 15), ('online', 15), ('housing', 14), ('grou
p', 14), ('history', 14), ('applying', 14), ('apply', 13), ('financial', 13), ('a
mp', 13), ('social', 13), ('looking', 13), ('bankside', 12), ('g', 12), ('scienc
e', 12), ('urbanest', 11), ('economy', 11), ('economic', 11), ('late', 11), ('man
agement', 11), ('degree', 11)]
```

In [20]:

```
# Find 50 most common words for post contents
all_text_content = ' '.join(df['content'])
content_counts = collections.Counter(all_text_content.split())
common_contents = content_counts.most_common(50)
print(common_contents)
```

```
[('lse', 571), ('’', 344), ('student', 310), ('would', 300), ('im', 270), ('yea
r', 240), ('course', 219), ('get', 199), ('i', 191), ('anyone', 180), ('know', 17
9), ('offer', 174), ('application', 173), ('hi', 170), ('msc', 145), ('like', 13
8), ('program', 135), ('m', 122), ('economics', 120), ('thanks', 116), ('internat
ional', 116), ('also', 113), ('school', 107), ('one', 106), ('got', 103), ('exper
ience', 99), ('looking', 98), ('degree', 98), ('wondering', 94), ('master', 92),
('applied', 92), ('want', 91), ('need', 90), ('finance', 90), ('time', 89), ('uni
versity', 89), ('apply', 89), ('study', 88), ('it', 84), ('hello', 84), ('help',
84), ('math', 83), ('first', 83), ('thank', 81), ('s', 80), ('currently', 79),
('really', 77), ('please', 77), ('good', 76), ('people', 75)]
```

In [21]:

```
# Find 50 most common words for the titles of the highest scoring 50 posts
highest_scoring = df.sort_values(['score'], ascending = False).groupby('score').head(50)
top_text_title = ' '.join(highest_scoring['title'])
top_title_counts = collections.Counter(top_text_title.split())
top_common_titles = top_title_counts.most_common(50)
print(top_common_titles)
```

```
[('lse', 113), ('msc', 45), ('student', 33), ('v', 26), ('application', 25), ('ac
commodation', 22), ('graduate', 21), ('undergrad', 16), ('course', 16), ('offer',
15), ('economics', 15), ('help', 13), ('summer', 13), ('master', 13), ('advice',
12), ('international', 12), ('political', 12), ('question', 12), ('school', 11),
('online', 11), ('get', 11), ('undergraduate', 11), ('bsc', 11), ('support', 10),
('history', 10), ('year', 10), ('applying', 10), ('admission', 10), ('group', 9),
('program', 9), ('got', 8), ('economy', 8), ('g', 8), ('looking', 8), ('hall',
8), ('economic', 8), ('ucl', 7), ('london', 7), ('entry', 7), ('accomodation',
7), ('science', 7), ('housing', 7), ('scheme', 7), ('urbanest', 7), ('uk', 7),
('social', 7), ('society', 7), ('degree', 7), ('apply', 7), ('anyone', 7)]
```

In [22]:

```
# Find 50 most common words for the content of the highest scoring 50 posts
top_text_content = ' '.join(highest_scoring['content'])
top_content_counts = collections.Counter(top_text_content.split())
top_common_contents = top_content_counts.most_common(50)
print(top_common_contents)
```

```
[('lse', 330), ('’', 204), ('student', 183), ('im', 157), ('would', 142), ('yea
r', 135), ('get', 118), ('course', 109), ('offer', 104), ('i', 103), ('know', 9
8), ('hi', 90), ('application', 87), ('anyone', 86), ('like', 83), ('program', 7
9), ('msc', 78), ('m', 71), ('school', 71), ('thanks', 71), ('also', 66), ('inter
national', 65), ('economics', 64), ('looking', 61), ('one', 59), ('got', 57),
('s', 55), ('need', 54), ('study', 53), ('help', 52), ('degree', 51), ('universit
y', 51), ('want', 50), ('finance', 50), ('experience', 50), ('wondering', 47),
('it', 47), ('math', 47), ('master', 46), ('good', 46), ('really', 45), ('receive
d', 45), ('2', 43), ('ive', 43), ('first', 43), ('applied', 42), ('hello', 42),
('thank', 42), ('please', 42), ('grade', 42)]
```

In [23]:

```
import pickle

# Save the list to a file for data analysis
with open('words.pkl', 'wb') as file:
    pickle.dump(common_titles, file)
    pickle.dump(common_contents, file)
    pickle.dump(top_common_titles, file)
    pickle.dump(top_common_contents, file)
```

Data Analysis

Exploratory Data Analysis (EDA)

To perform the exploratoty data analysis, we load the list obtained from the previous section.

In [24]:

```
import pickle

# Load the list from the file
with open('words.pkl', 'rb') as file:
    titles = pickle.load(file)
    contents = pickle.load(file)
    top_titles = pickle.load(file)
    top_contents = pickle.load(file)

    titles_d = dict(titles)
    contents_d = dict(contents)
    top_titles_d = dict(top_titles)
    top_contents_d = dict(top_contents)
```

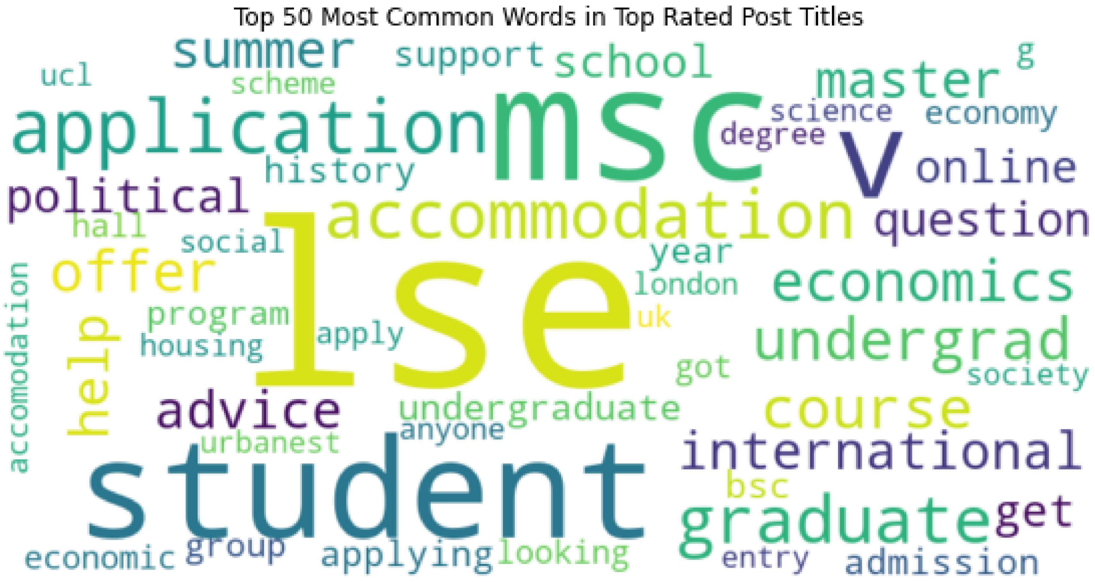
Firstly, we loaded the top 50 common words in top rated post titles and plot a word cloud to visualize the size of the words in proportion with its frequency.

In [25]:

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud


# Create a word cloud for 50 most common words in the title of the highest scoring 50 posts
wordcloud = WordCloud(width=800, height=400, background_color="white").generate_from_frequencies

# Plot the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Top 50 Most Common Words in Top Rated Post Titles')
plt.axis('off')
plt.show()
```



- The plot above shows that 'lse', 'application', 'student', 'msc', and 'accomodation' are some of the most frequently appearing words in titles of 50 highest scoring posts.
- Most of the words that appear in the word cloud have neutral meanings and are associated with general information about LSE.

Secondly, we plot a line graph to visualize the top 50 words in all post titles.

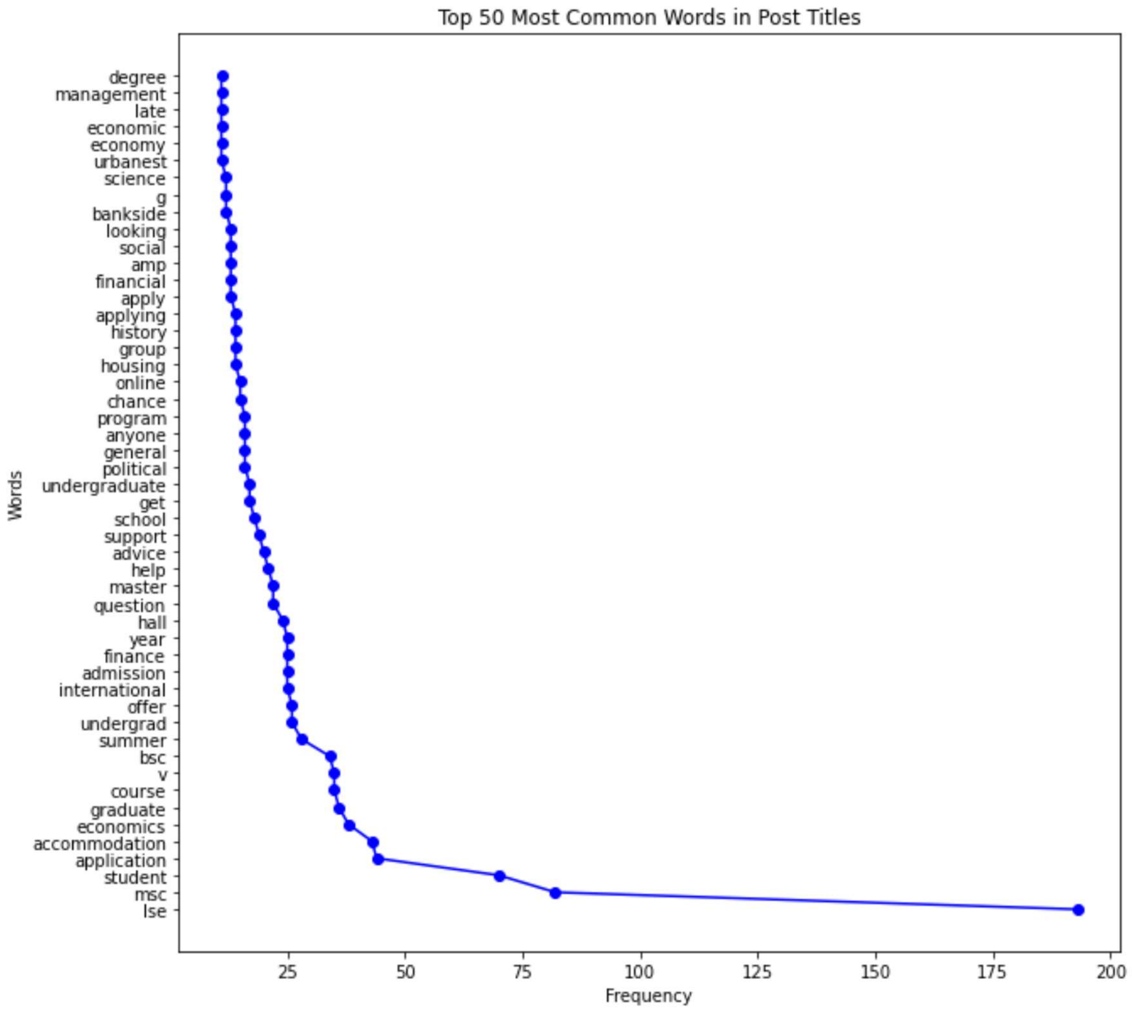
In [26]:

```
# Extracting keys and values from dictionary
names = list(titles_d.keys())
nums_t = list(titles_d.values())

# Creating a line graph
plt.figure(figsize=(10, 10))
plt.plot(nums_t, names, marker='o', color='blue')

# Adding labels and title
plt.ylabel('Words')
plt.xlabel('Frequency')
plt.title('Top 50 Most Common Words in Post Titles')

# Displaying the graph
plt.show()
```



- By visualizing the frequency of the word appearance, we can see that the most frequently appearing word is 'lse', followed by 'msc', 'student', and 'application'.
- There is a large gap between the first and second most common word, and the appearance of words after the top ten common words does not vary a lot.
- Other than providing numerical count to each word's appearance, the result from this plot is actually similar to the previous plot.

Thirdly, we plot two bar charts to show the top 50 words in post contents and in top rated post contents respectively.

In [27]:

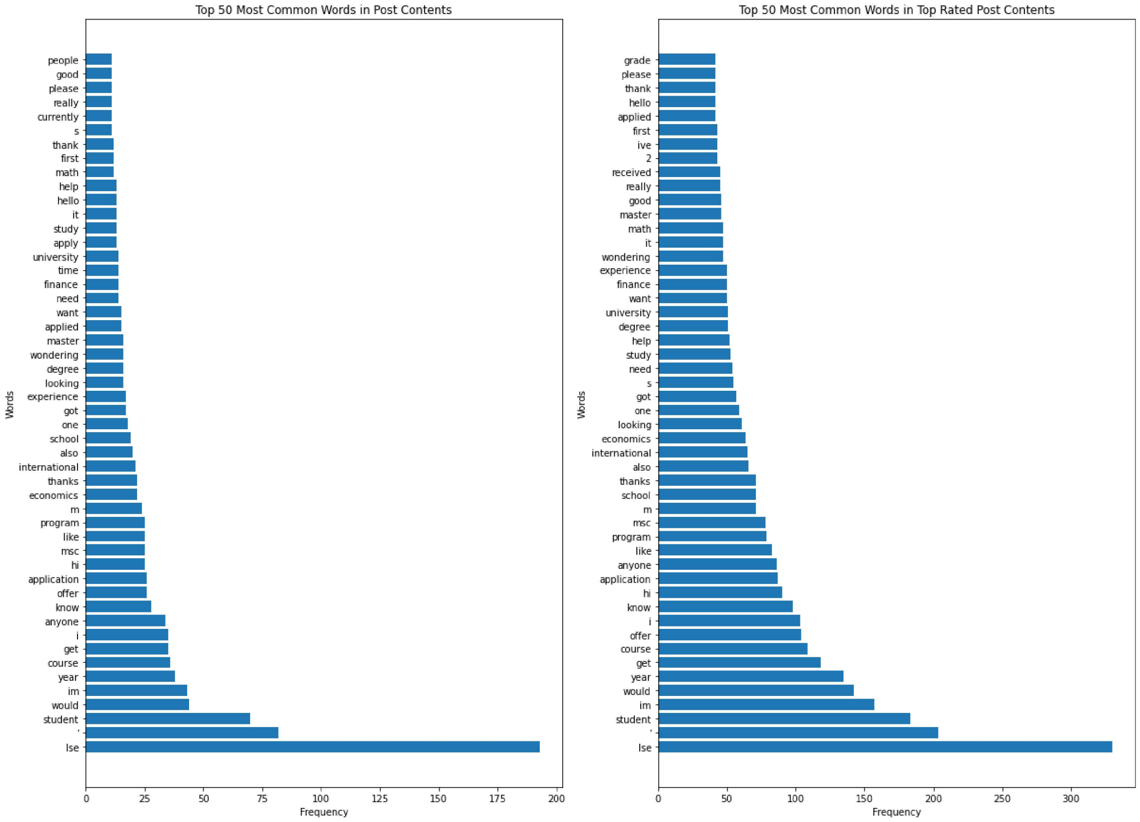
```
# Extracting keys and values from dictionary
words = list(contents_d.keys())
nums_c = list(contents_d.values())
top_words = list(top_contents_d.keys())
top_nums = list(top_contents_d.values())

fig, ax = plt.subplots(1,2, figsize = (20,15))
# Create bar chart for the top 50 words in post contents.
ax[0].barh(words,nums_t)
ax[0].set_xlabel('Frequency')
ax[0].set_ylabel('Words')
ax[0].set_title('Top 50 Most Common Words in Post Contents')

# Create bar chart for the top 50 words in top rated post contents.
ax[1].barh(top_words,top_nums)
ax[1].set_xlabel('Frequency')
ax[1].set_ylabel('Words')
ax[1].set_title('Top 50 Most Common Words in Top Rated Post Contents')
```

Out[27]:

```
Text(0.5, 1.0, 'Top 50 Most Common Words in Top Rated Post Contents')
```



- Comparing the two graphs above, we can see that the most common words in all posts and top rated posts are very similar although the number of appearances differ.
- The most frequently appearing words are 'lse', 'student', 'im', 'would', and 'year', where the pronouns and auxiliary verbs presented here are not really helpful in analyzing public perception of LSE.

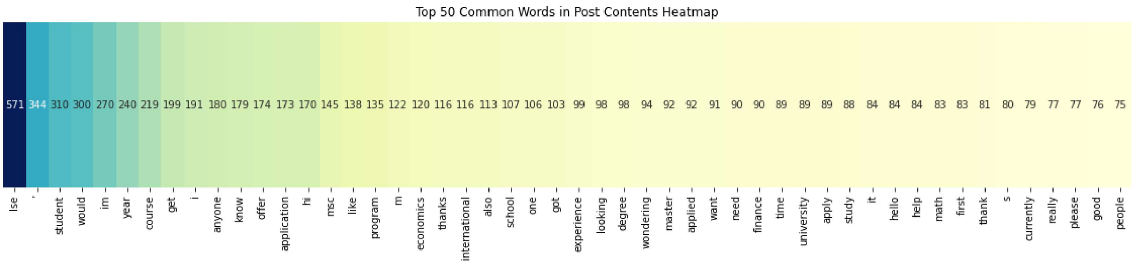
- Other words that are associated with LSE include 'economics', 'international', 'finance', 'good', and 'experience', which are some features that LSE is known for.

Finally, we plot a heat map to visualize the top 50 words in all post contents.

In [28]:

```
import seaborn as sns
# Creating a list of frequencies to use as heatmap data
heatmap_data = [nums_c]

# Creating a heatmap
plt.figure(figsize=(20, 3))
sns.heatmap(heatmap_data, annot=True, fmt="d", cmap="YlGnBu", cbar=False,
            xticklabels=words, yticklabels=False)
plt.title('Top 50 Common Words in Post Contents Heatmap')
plt.show()
```



- The heatmap here shows similar results to the previous graphs.

Q1: What are some common traits that the public associate LSE with?

It can be shown that the common traits that the public associates LSE with are being international and strong in economics. This is because these two traits are represented by the most common words we extracted. There are also many keywords associated with MSc and graduates, which can be inferred that the postgraduate education of the LSE may be either famous or controversial.

Q2: What do people on Reddit want to know more about LSE?

People may want to know more about the application process, the academic reputation and the accommodation of the LSE because there are many common keywords (offer, application, applied, accommodation, program, course, degree, personal, etc) related to these topics.

In [29]:

```
highest_scoring.head()
```

Out[29]:

	title	content	score
607	got in 🤩	global medium culture msc applied nov 17 got h...	24
60	got in 🤩	global medium culture msc applied nov 17 got h...	24
370	whats job graduating	might help current prospective student underst...	18
570	lse v ucl v warwick ppe	hi all im fortunate position offer warwick ucl...	16
395	2024 undergrad — wait excruciating	completely understand circumstance excess appl...	16

Moreover, if we look at the top scoring posts, we can see that the top rated post are mostly about people who received offers from LSE, indicating that people on Reddit are mostly interested about whether they can get accepted into the university.

Conclusion

Findings

In conclusion, our report illustrates that many comments on Reddit about the LSE are focused on its academic reputation, its application process and its accommodation. The common traits that the LSE shows to the public are its strength in diversity and economic research. There are also many keywords related to the postgraduate education of the LSE, which indicates that this topic has drawn many heated discussions as well.

Limitations

While we were trying to understand public perception of LSE, we only gathered data from Reddit, which is a relatively small sample of the population to draw general conclusions from our findings.

Furthermore, most of the words we found have neutral meanings and failed to give us an insight of the public's like or dislike toward the university. The tokenized words also contains single words and pronouns that after the data cleaning process and are not insightful for our analysis.

Last but not least, we could only obtain fragmented idea about what the post creator is trying to convey by analyzing the words separately.

Future Improvements

In the future, we could use additional features of the NLTK module that are not currently included to remove auxiliary verbs and pronouns. We could also make use of other Python libraries like PushShift or PRAW to analyse and collect more than 1000 posts. If possible, we would also like to collect different data sources to extend our study to a larger population.

To address the issue of collecting neutral words that are less insightful, we could also look at whether there are more positive or more negative words and analyse the top scored posts in regards to this.

Testing and debugging could also be a potential future improvement. We could invest more time in creating thorough tests and debugging tools to make it easier to catch and debug errors to make sure the code runs smoothly, reliably, and achieve the intended goal.

In [ ]: