



7BUIS025W Coursework 2023-24

Web and Social Media Analytics

Yahya Miah

W2036476

Word count: 3080

Introduction

In this report, fully electric vehicle experiences will be investigated. The experiences of electric vehicle owners in different settings will be investigated, using the social media modelling process. Limited data will be collected from social media sources. This electric vehicle industry is becoming more popular due to the sustainability of the transportation, though each experience isn't trouble-free. Fully electric vehicles offer zero-emission transportation, which is good for the environment. Investigating fully electric vehicles will allow for effective analysis as vehicles such as cars are the most popular. Any findings established can be compared to their alternatives from different settings to then allow the ability to suggest actions for the sake of improving the experience of owners of electric vehicles.

The electric vehicle market is projected to become valued at £714.9 bn by 2030 from £291.5 bn in 2023 (Ukpanah, 2024). More than 10 million electric vehicles are on the road as of 2023, with Tesla having the largest market share (14.55%) selling more than all other manufacturers.

The electric vehicle industry in general is positively growing and succeeding as the electric vehicles sold among all vehicles increased from 5% in 2020 to 14% in 2022, where it is also predicted that the increase will be up to 18%. In the US alone, the fully electric vehicles increase in sales reached 7.9% from 6.1% the previous year. In 2022, China's share of electric car total sales was 22%. The US and China are some of the world's largest car markets, with China being the largest (Carlierm, 2024). The purchase and use of these vehicles differ significantly across the world, with some countries predominantly using electric cars, for example, Norway with an 80% proportion of electric cars sold in 2022. Ultimately, the industry is growing but the proportion of electric cars is region-dependent.

Looking further into the industry statistics you can learn that Europe has the largest market share of 40% in 2023 projected to increase by 5% in 2024 (Ukpanah, 2024). It is very clear that Europe in comparison to other continents contains more electric vehicles. This is confirmed by the Norway statistics in combination with Iceland and Sweden having more than 30% in sales in 2022.

Analysing the technology statistics it can be inferred that the batteries used within these vehicles will potentially fall by 40% by 2025 from 2022. Multiple manufacturers plan to make their products cost-effective, for example, General Motors wants to lower the costs of electric cars by researching the lithium metal-batter chemistry and Gotion High-Tech plans to mass produce their batteries made by lithium-manganese iron phosphate by 2024 (TheEnergyMix, 2023). Offering batteries with a large range by km could lead to a significant increase in the use of EVs, with Goldman Sachs's estimation that Full Electric vehicle sales could account for 47% of global vehicle sales by 2030.

The top-selling electric full-vehicle manufacturer was Tesla (Statista, 2024). The Tesla Model Y had the highest number of sales which was 772,364 as of August 2023, and the Tesla Model 3 had the second highest with 364,403 sales. The Tesla Model 3 in 2024 has a range in miles of 358, reaching 60mph in 3.1 seconds, and costing £42,990. Alternatively, the Rivian R1T has a range in miles of 314, 3.0 seconds acceleration to 60 mph, and costs £61,300. Comparing two

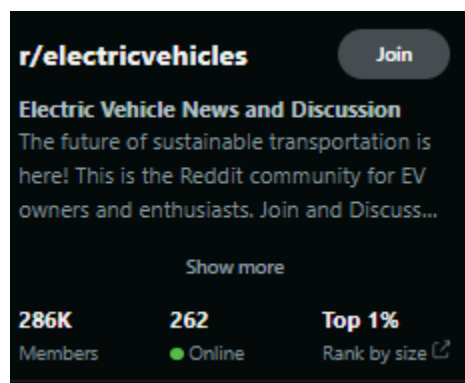
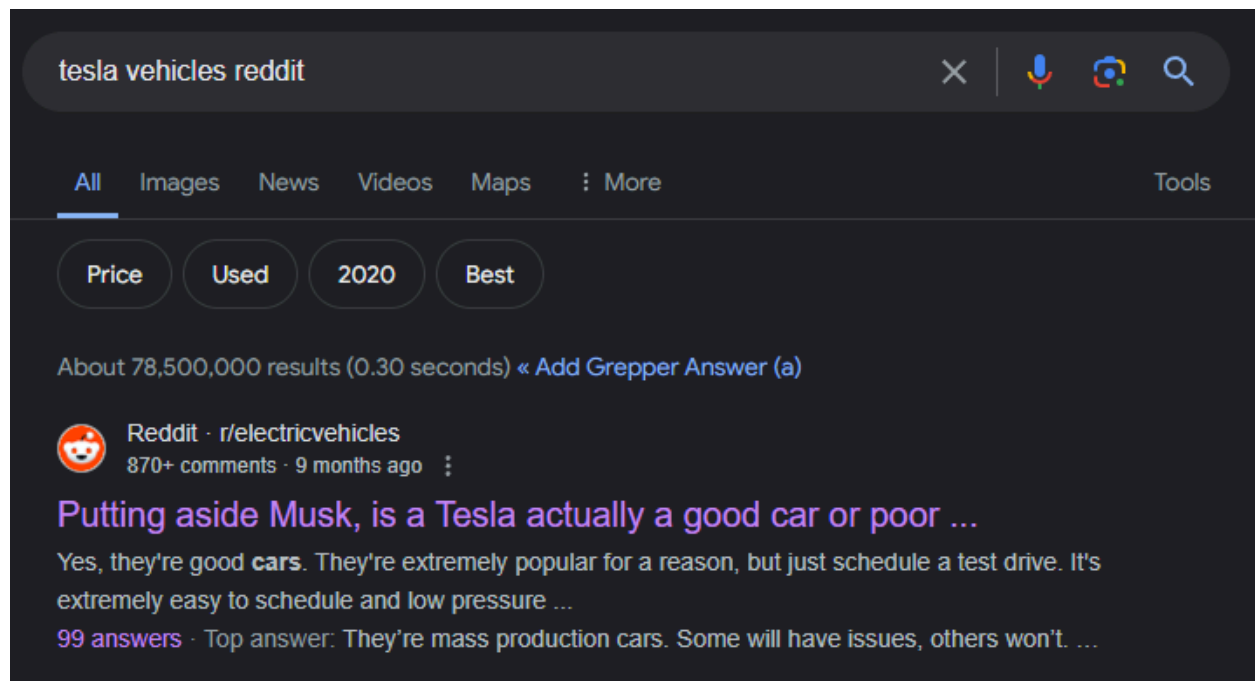
of the leading electric cars establishes that Tesla in general offers a better deal when factoring in the cost and performance, with the Rivian model only having a small advantage with acceleration. In addition, Tesla offers semi-autonomous driving capabilities.

Data collection and pre-processing


The approach taken to find and collect the data:

To investigate the experiences of electric vehicle owners in different settings, the target social media post would be related to Tesla, due to their popularity in the electric vehicle industry. This is because as established in the introduction Tesla dominates the industry, so the reviews on their fully electric vehicles would give an understanding of why their vehicles are chosen over others worldwide. The Tesla Model 3 is the all-time bestselling plug-in electric car worldwide (Shahan, 2021). Tesla's use of sustainable energy for their vehicles is the reason for them to become the front runner in the industry as environmentally conscious consumers are willing to purchase their vehicles. It would make sense to find data from a social media site related to fully electric vehicles, to investigate the experiences of electric vehicle owners in different settings who use or don't use Tesla vehicles.


Google search was used to find social media sites that show how customers and potential customers feel about Tesla vehicles. Google is a better search engine to use, as it has a larger search index in comparison to Bing when it comes to pulling results (Magne, 2024). The social media site to be used is Reddit, as it can be used to find the opinions on the choice of Tesla's fully electric vehicles from subreddits related to electric vehicles, to understand why it was chosen over others from different settings. The choices are justified as the first web page to be shown is an electric vehicles Reddit thread with over 870 comments about opinions on Tesla cars. The electric vehicles subreddit has over 286k members with hundreds of people online, making it a reliable site to gain insight on fully electric vehicle experiences, with the thread itself having lots of comments.



When it comes to collecting social media data posts the Reddit API is used. This process needs a developer account to set up, a script application to be developed, and a client secret and an app ID unique identifier to be generated. The client secret and app ID authenticate application requests to Reddit's servers.

 **reddit** [PREFERENCES](#) [options](#) [apps](#) [RSS feeds](#) [friends](#) [blocked](#) [password/email](#) [delete](#)

developed applications


change icon

w2036475_FEV
personal use script
XLNA_xSQJDdsJ4eRspez5w

for my social media analytics assignment

secret igW00sftHLYwZ6S3sB8gwY66xtPcAA

developers 1ggl3p1ggl3_777 (that's you!) [remove](#)

name

add developer:

description

about url

redirect uri

[delete app](#)

Data collection procedure:

Using the Reddit API with the Praw and Pprint libraries it is possible to explore the social media post, but first, the right framework must be developed, within Google Collab. In Google Collab using the Python Reddit API Wrapper library, it is possible to interact with thread components such as comments, as API is used to connect with Reddit. The data collection procedure starts with installing Praw and importing the necessary libraries. Using Python code a client secret and API ID must be assigned to a variable.

The variables are used within the initiation of the Praw Reddit instance. For the client_id and client_secret praw.Reddit pre-built variables in combination with the user_agent argument "Comment Extraction" are used to make sure there is a good relationship with the Reddit API by following the API's guidelines when it comes to identification. For the data collection procedure after the initialization of the Praw Reddit instance, a Reddit submission is put into a variable to then be used as a value for the URL parameter within the submission function from the praw package to ultimately create the submission object. The submission object will be accessed during the data exploration.

!pip install praw

```
import praw
import pprint
```



Collecting praw

Downloading praw-7.7.1-py3-none-any.whl (191 kB)

191.0/191.0 kB 1.5 MB/s eta 0:00:00

Collecting prawcore<3,>=2.1 (from praw)

Downloading prawcore-2.4.0-py3-none-any.whl (17 kB)

Collecting update-checker>=0.18 (from praw)

Downloading update_checker-0.18.0-py3-none-any.whl (7.0 kB)

Requirement already satisfied: websocket-client>=0.54.0 in /usr/local/lib/python3.10/

Requirement already satisfied: requests<3.0,>=2.6.0 in /usr/local/lib/python3.10/dist-

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-package

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-p

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-p

Installing collected packages: update-checker, prawcore, praw

Successfully installed praw-7.7.1 prawcore-2.4.0 update-checker-0.18.0



SECRET="igW00sftHLYwZ6S3sB8gwY66xtPcAA"

APP_ID="XLNA_xSQJDdsJ4eRspez5w"

```
reddit = praw.Reddit(
    client_id=APP_ID,
    client_secret=SECRET,
    user_agent="Comment Extraction"
)
```

```
SUBMISSION_URL="https://www.reddit.com/r/electricveh
submission = reddit.submission(url=SUBMISSION_URL)
```

```
✓ 2s ▶ for top_level_comment in submission.comments:
        print(type(top_level_comment))

⏏ WARNING:praw:It appears that you are using PRAW
It is strongly recommended to use Async PRAW: !
See https://praw.readthedocs.io/en/latest/gett

<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
<class 'praw.models.reddit.comment.Comment'>
```

Exploratory phase

To explore the Reddit social media dataset for the sake of the investigation, it is best practice to analyse the most popular words and phrases.

Most popular words:

For analysing the dataset to find the most popular words, when using Google Collab with Praw, it is best practice to use Python libraries such as collections and nltk. The specific classes used to count the words and remove stopwords are called Counter and stopwords from collections and nltk.corpus. Counter will be used to count the words in the social media data post and stopwords will be used to put English stopwords into variables to be removed during the analysis process.

After importing and installing the necessary libraries, the process starts with downloading stopwords to then be put into a variable. Using a for loop with comments within the submission.comments object, an if statement is used to filter out the stop words. The filtered words/tokens are then counted using the Counter() function. Using the most_common() function from the Counter class the most common words based on the counter of each word will be found.

```
[5] from collections import Counter
    from nltk.corpus import stopwords
    import string
    import nltk
    from nltk.util import ngrams

    #removal of stopwords
    nltk.download('stopwords')
    stop_words = set(stopwords.words('english'))

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

```
words = []

for top_level_comment in submission.comments:
    if isinstance(top_level_comment, praw.models.Comment):
        comment_text = top_level_comment.body

        tokens = comment_text.lower().split()
        table = str.maketrans('', '', string.punctuation)
        tokens = [word.translate(table) for word in tokens]
        filtered_tokens = [word for word in tokens if word not in stop_words]

        words.extend(filtered_tokens)

word_counts = Counter(words)

top_words = word_counts.most_common(15)
print("Top words: ", top_words)
```

Most popular phrases:

For analysing the Reddit post to find the most popular words, when using Google Collab with Praw, the use of the Python libraries aforementioned are necessary. The specific classes similar to those used in the analysis of the most popular words are used, however, the n-grams class from the nltk collection is used. The n-grams class is used to extract pairs of words from the comments which will be useful when trying to find the top phrases rather than words.

The process runs very similarly to the looping process of the top word finder. A combination of the Counter and most_common function in combination with the ngrams() function is used. The ngrams() function is used to create a variable that holds the bi_gram combination of words, based on the set parameter 2.


```

phrases = []

for top_level_comment in submission.comments:
    if isinstance(top_level_comment, praw.models.Comment):
        comment_text = top_level_comment.body.lower()

        comment_text = comment_text.translate(str.maketrans('', '', string.punctuation))
        bi_grams = ngrams(comment_text.split(), 2)

        phrases.extend([' '.join(gram) for gram in bi_grams if gram[0] not in stop_words and gram[1] not in stop_words])

phrase_counts = Counter(phrases)

top_phrases = phrase_counts.most_common(15)
print("Top phrases: ", top_phrases)

```

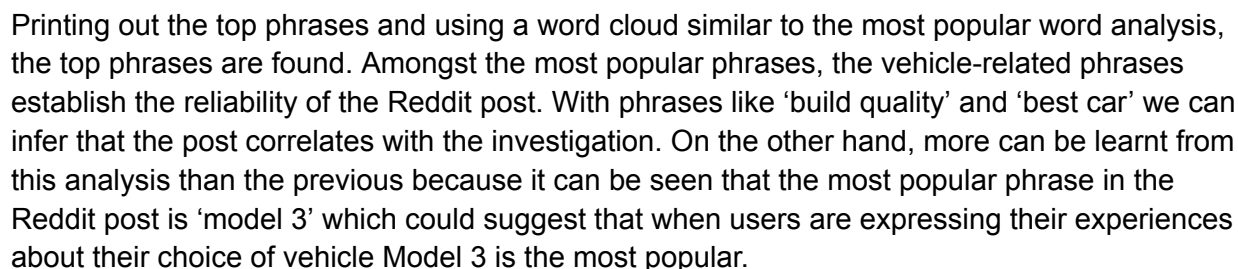
Results and word associations/correlations:

Printing out the top words and using a word cloud, the top results are then found. Amongst the most popular words, the vehicle-related words establish the reliability of the social media datapost. With the investigation being about fully electric vehicles the most popular words are as expected. In addition to this the words 'quality' and 'issues' correlate with the investigation of the different experiences within different settings due to the comparisons of vehicles.

```

Top words:  [('car', 174), ('tesla', 153), ('cars', 99), ('model', 87), ('quality', 73), ('issues', 67),

```



Time-series analysis of the top comments:

For the time-series analysis of the top comments in the post, a function is created to extract comments from the Reddit submission to be counted based on the number of comments per day. In summary, the extracted comments are then organised into a Pandas DataFrame where the timestamps are converted to datetime format.

When exploring the data valuable insights can be found by analysing the timeline of comments within a post, to understand if the opinions of the users are coming from discussion or random one-off comments. These insights are valuable because themes and topics found in discussion rather than one-off comments can be analysed for the investigation. When plotted it can be seen the majority were in June 2023 with occasional comments onwards, especially in April 2024. The Tesla Model 3 with a revamped interior was released in late 2023, which led to lots of discussion on the worth of fully electric vehicles.

```

[13] import pandas as pd


comments = []

# Recursively fetch comments due to too many comments
def extract_comments(comments):
    all_comments = []
    for comment in comments:
        if isinstance(comment, praw.models.Comment):
            all_comments.append({
                'created_utc': comment.created_utc,
                'body': comment.body
            })
        elif isinstance(comment, praw.models.MoreComments):
            all_comments.extend(extract_comments(comment.comments()))
    return all_comments

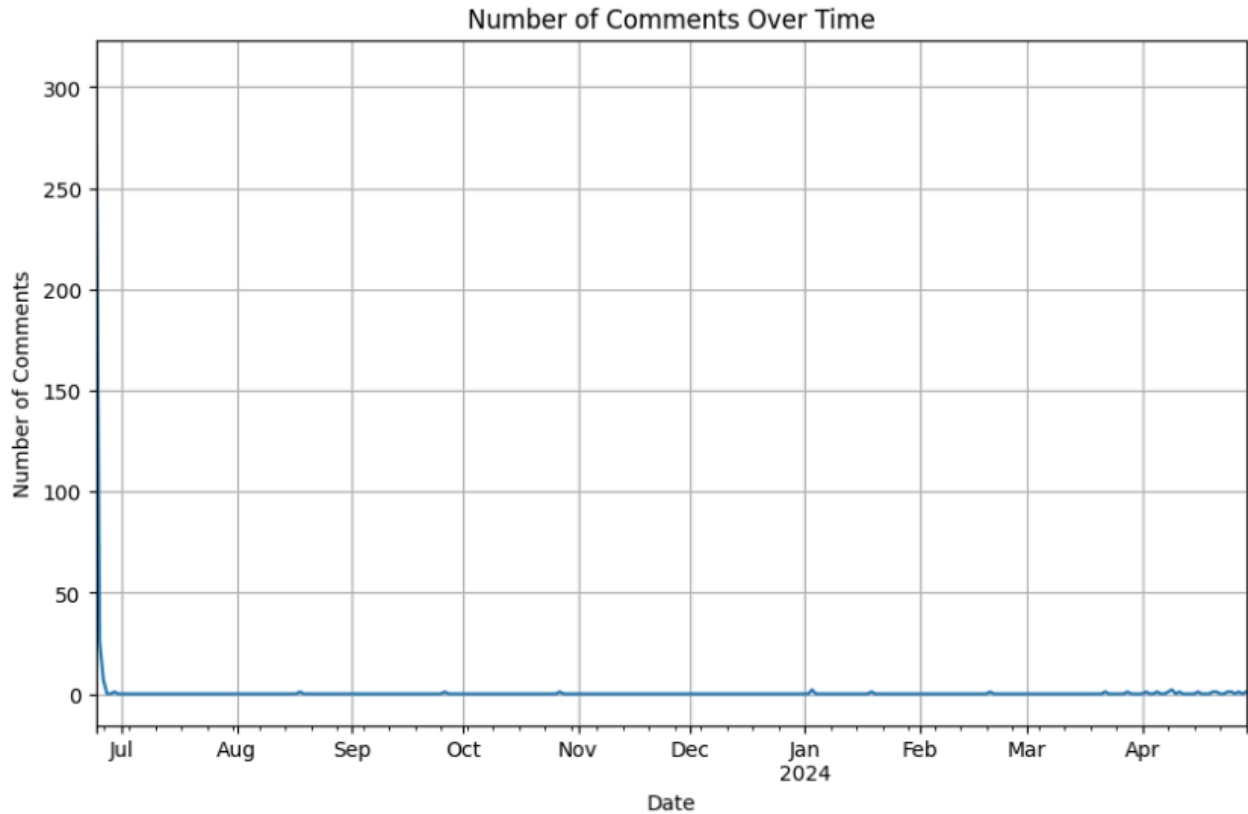
comments = extract_comments(submission.comments)

comments_df = pd.DataFrame(comments)
comments_df['created_utc'] = pd.to_datetime(comments_df['created_utc'], unit = 's')
comments_df.set_index('created_utc', inplace = True)
comment_counts = comments_df.resample('D').size()

```

 comments_df.describe()

	body
count	368
unique	368
top	They're mass production cars. Some will have i...
freq	1



Text mining

Text mining approach:

For finding topics and themes within the Reddit datapost, a text mining approach can be used. For the modelling and analysis a Latent Dirichlet Allocation model is developed, to find topics within the comment data. The purpose of this model is essentially to identify themes based on the comments established within the topics. The model identifies topics without prior knowledge of the domain topics, so themes are established based on the results of the modelling.

To create an LDA model using the Reddit submission in Google Collab with Python the corpora class from the gensim library is installed and imported. The corpora sub-module provides the tools and functions for developing the LDA model that is used to analyse collections of Reddit comments. The process starts with tokenizing the comments to then be used by a genesis model that is looking for several topics. The number of topics 4 is chosen based on the discussion of the Reddit post that is about fully electric vehicles, model type, technologies within said vehicles, and costs.

```
#Text Mining - LDA

!pip install gensim
import gensim
import gensim.corpora as corpora

# Tokenize comments
documents = [comment.split() for comment in words]
vocab = corpora.Dictionary(documents)
corpus = [vocab.doc2bow(text) for text in documents]

num_topics = 4
lda = gensim.models.LdaMulticore(corpus=corpus, id2word=vocab, num_topics=num_topics)
print(lda.print_topics())

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async`
and should_run_async(code)
Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (4.3.2)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.1
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim)
WARNING:gensim.models.ldamulticore:too few updates, training might not converge; consider increasing the
[(0, '0.036*"one" + 0.026*"best" + 0.023*"like" + 0.020*"get" + 0.019*"3" + 0.015*"issues" + 0.013*"build
```

Evaluation of model:

With the printed-out topics and the use of a Coherence Model, the model is evaluated. With 4 topics established from the LDA model, the coherence of the model to determine the goodness of fit must be found. The coherence score determines how meaningful and interpretable topics are. The coherence model used in the investigation uses the LDA model, the extracted comments, the dictionary of the comments, and the 'c_v' coherence measure. The 'c_v' coherence measure calculates the coherence based on the similarity between pairs. The coherence score is found by measuring the cosine similarity between the comments on a topic. A coherence score closer to 1 means that the topics are more coherent, so with a coherent score of 0.81 the model is a good fit.

```
from gensim.models.coherencemodel import CoherenceModel

#Evaluation of topic coherence
coherence_model = CoherenceModel(model=lda, texts=documents, dictionary=vocab, coherence='c_v')
coherence_score = coherence_model.get_coherence()
print(f"Topic Coherence Score: {coherence_score}")

for topic_id, topic in lda.print_topics():
    print(f"Topic {topic_id + 1}: {topic}")

#very good coherence score
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_` and `should_run_async` are deprecated

Topic Coherence Score: 0.8187233635941433

Topic 1: 0.036*"one" + 0.026*"best" + 0.023*"like" + 0.020*"get" + 0.019*"3" + 0.015*"issues" +

Topic 2: 0.015*"years" + 0.014*"service" + 0.012*"drive" + 0.011*"good" + 0.011*"price" + 0.010*

Topic 3: 0.073*"car" + 0.043*"quality" + 0.027*"better" + 0.024*"model" + 0.021*"tesla" + 0.018*

Topic 4: 0.045*"cars" + 0.035*"tesla" + 0.031*"teslas" + 0.028*"great" + 0.014*"company" + 0.014*

Analysis of topics:

To analyse the topics to find insights it is effective to use visualisation techniques. The pyLDAvis library is used to create an interactive visualisation notebook of the LDA model, using the document-term matrix and the vocabulary.

```
!pip install pyLDAvis
import pyLDAvis.gensim

pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim.prepare(lda, corpus, vocab)
pyLDAvis.display(vis)
```

Using the intertopic distance map that uses multidimensional scaling, within the visualisation notebook, 4 topics are displayed. From this, it can be seen that all the topics are separate which means that the topics aren't necessarily similar which is effective in investigating how the author of comments feels about Tesla's fully electric vehicles in comparison to others in different settings as different themes can be found based on the discussions that the topics are derived from. Regarding the most relevant terms in each topic, Topic 1 is mostly about durability, Topic 2 is mostly about the company, Topic 3 is mostly about value, and Topic 4 is about comparing quality. Themes that can be derived could be longevity, brand, price, and model types.

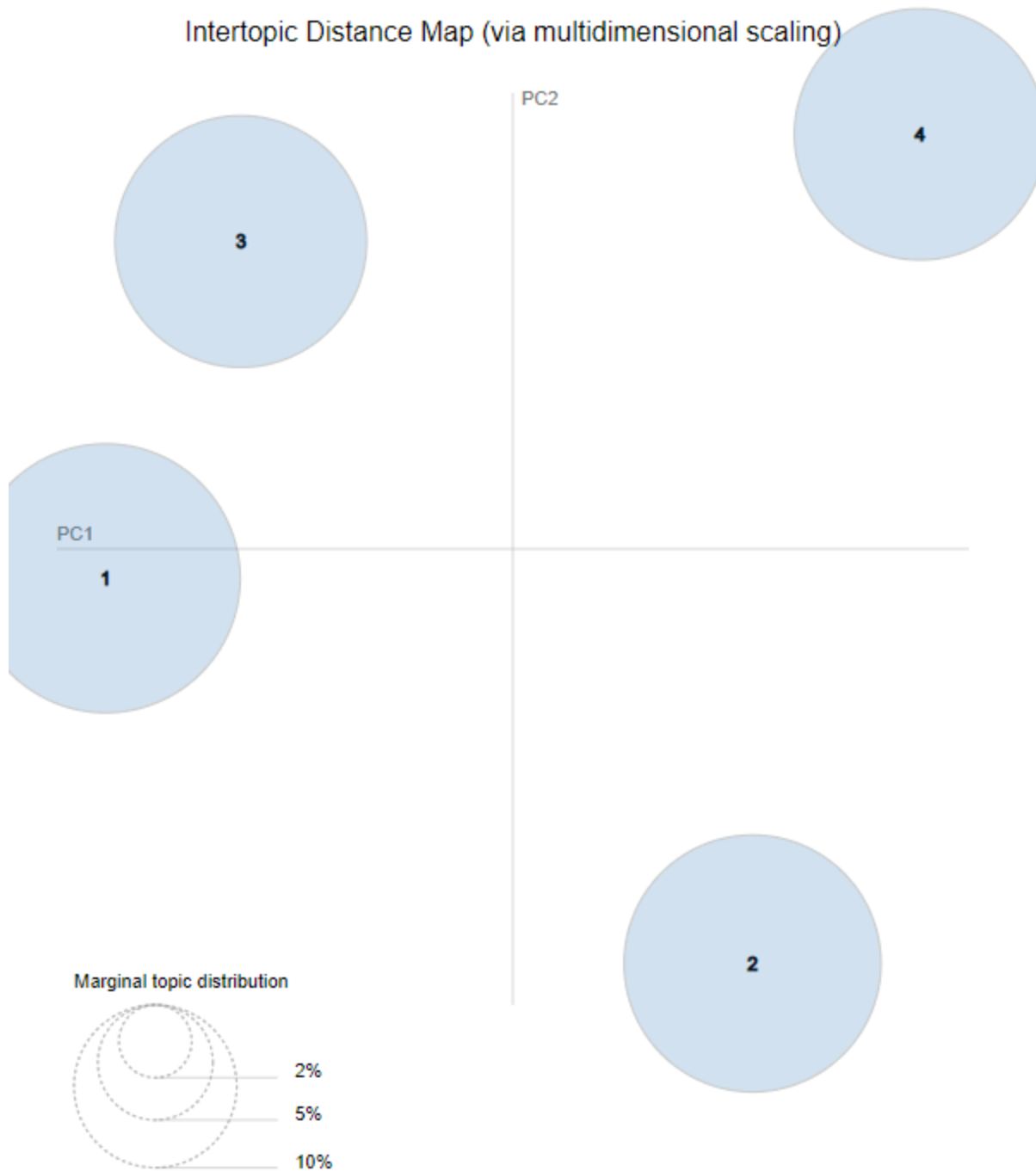
Selected Topic: 0

[Previous Topic](#)

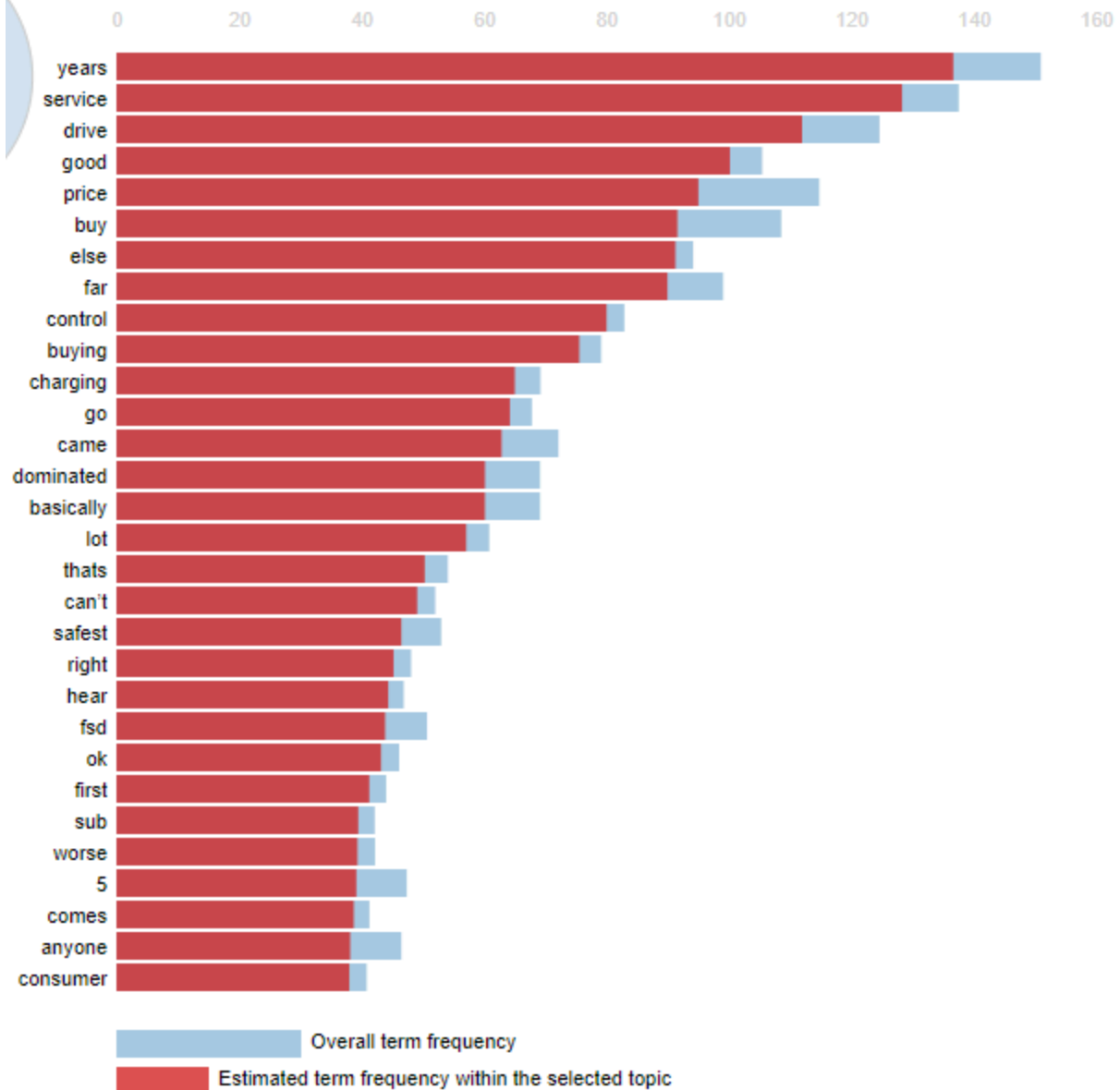
[Next Topic](#)

[Clear Topic](#)

Intertopic Distance Map (via multidimensional scaling)



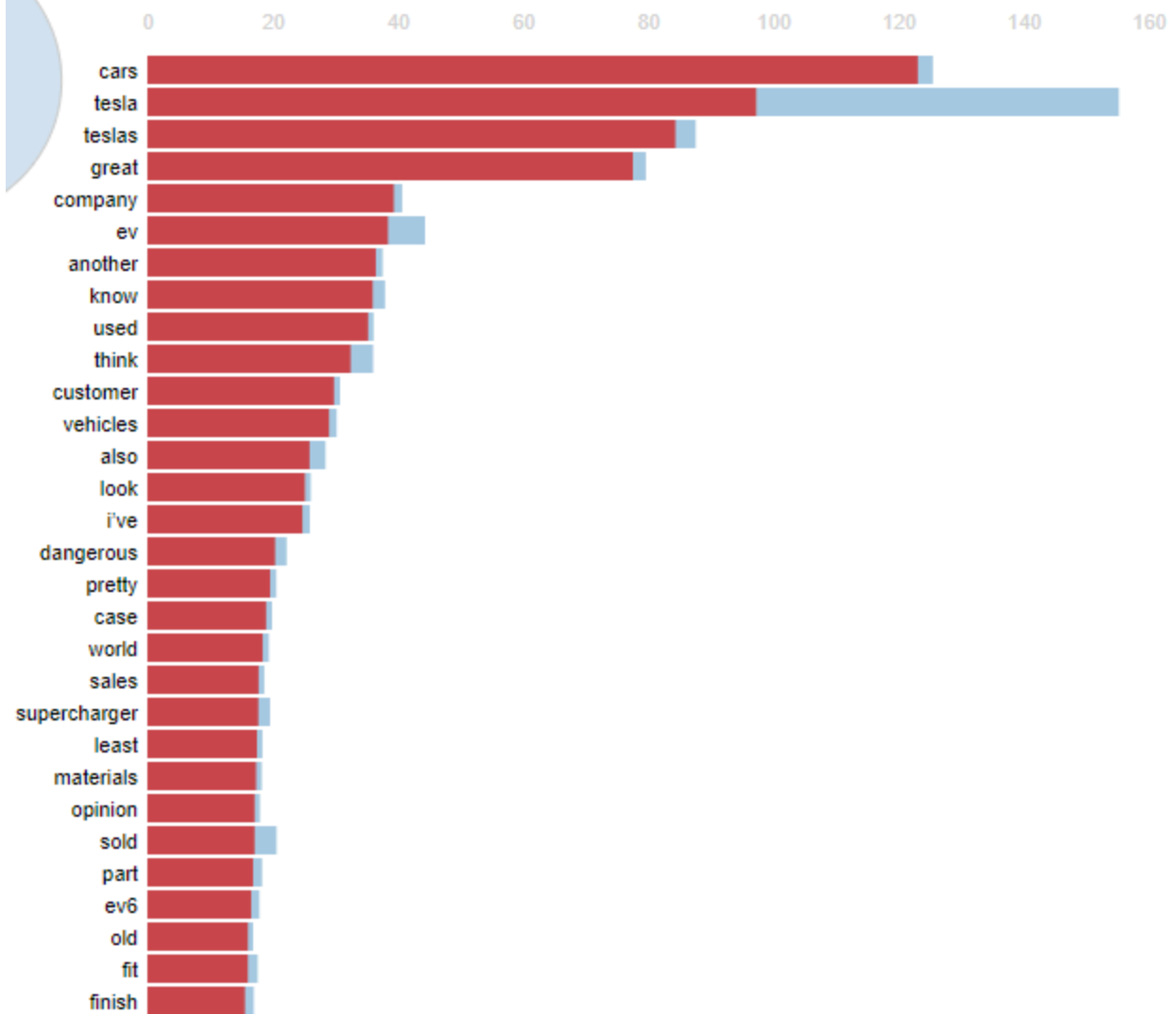
Top-30 Most Relevant Terms for Topic 1 (27.3% of tokens)



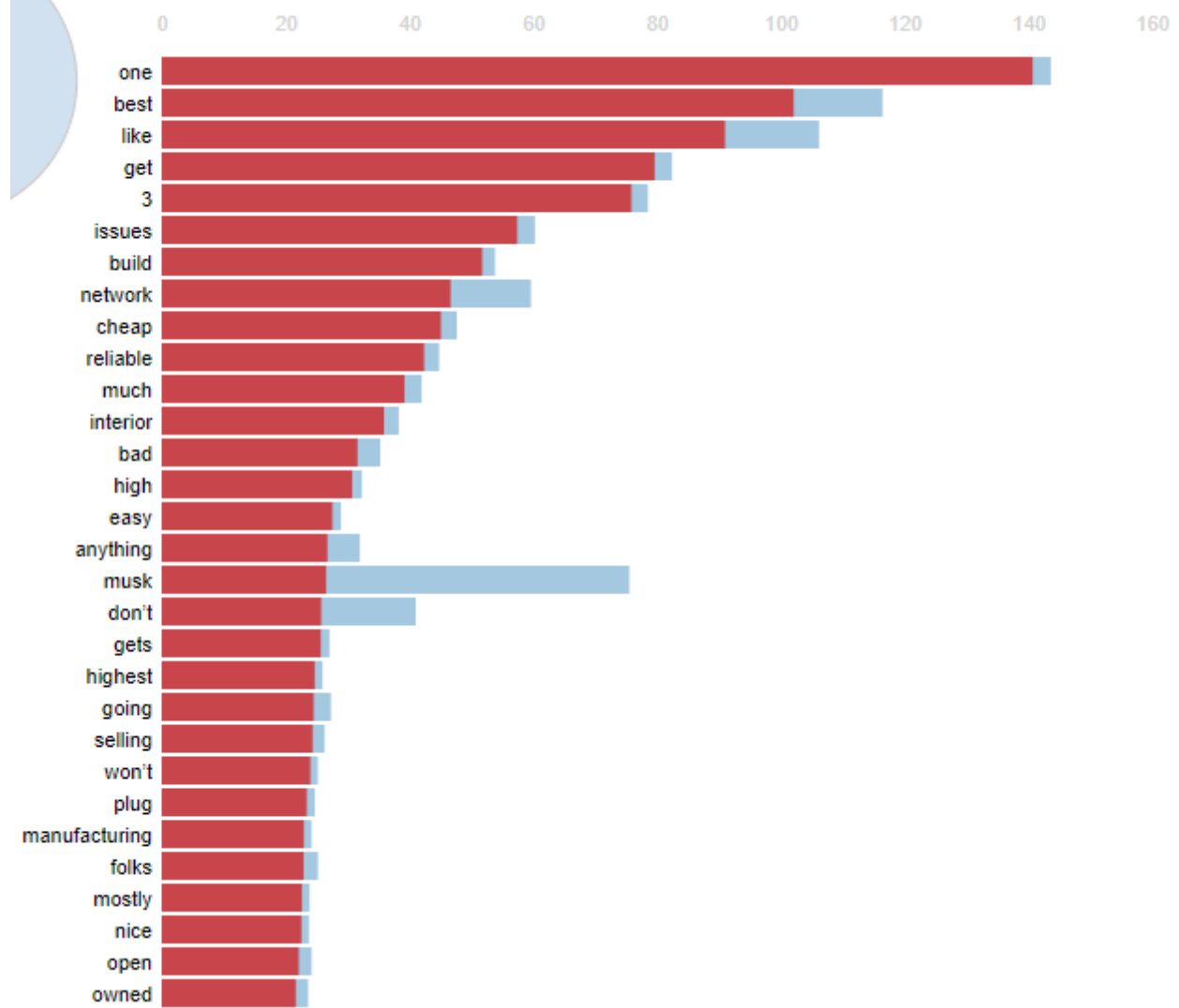
1. $\text{saliency}(\text{term } w) = \text{frequency}(w) * [\sum_t p(t | w) * \log(p(t | w)/p(t))]$ for topics t ; see Chuang et. al (2012)

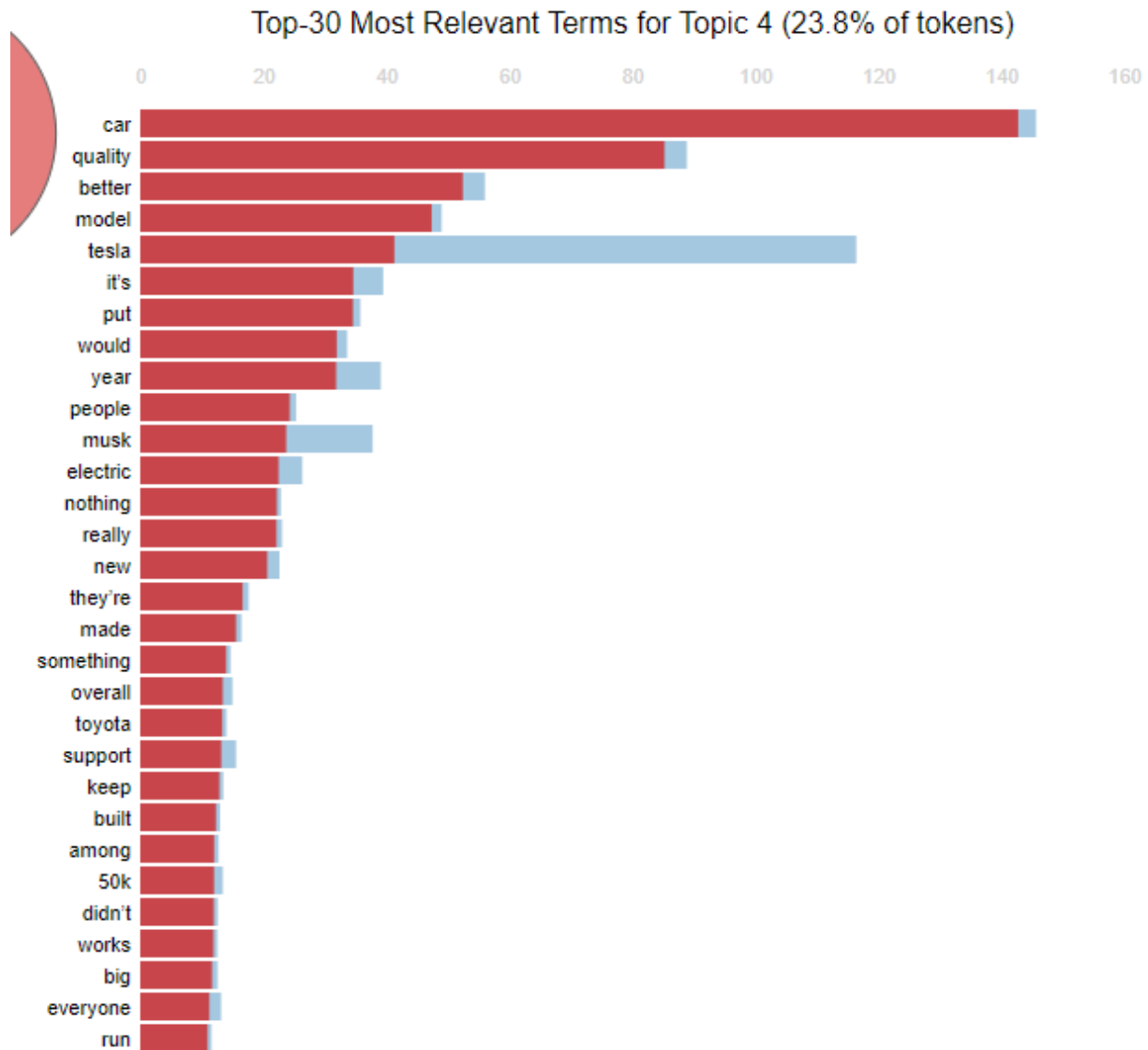
2. $\text{relevance}(\text{term } w | \text{topic } t) = \lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

Top-30 Most Relevant Terms for Topic 2 (24.9% of tokens)



Top-30 Most Relevant Terms for Topic 3 (23.9% of tokens)





Sentiment modelling and analysis:

Sentiment modelling is used to understand the subjective information such as opinions, emotions and attitudes, conveyed in the Reddit post. Sentiment modelling can be used to understand the owner's sentiment as well as understand the opinions of the authors of the comments.

The SentimentIntensityAnalyzer submodule from the nltk.sentiment.vader library is what provides the tools to create the analyzer. The sentiment model is used on the comments of the post to generate sentiment scores. To understand the owner sentiment (post author) the average sentiment score is found.

```

nltk.download("vader_lexicon")
from nltk.sentiment.vader import SentimentIntensityAnalyzer

sid = SentimentIntensityAnalyzer()

#Analyze the sentiment for each comment
sentiments = []
for comment in submission.comments.list():
    #Filter out the deleted comments
    if not hasattr(comment, 'body'):
        continue

    #Sentiment analysis
    sentiment_scores = sid.polarity_scores(comment.body)
    sentiments.append(sentiment_scores)

#overall sentiment scores
compound_scores = [score['compound'] for score in sentiments]
avg_sentiment = sum(compound_scores) / len(compound_scores)

```

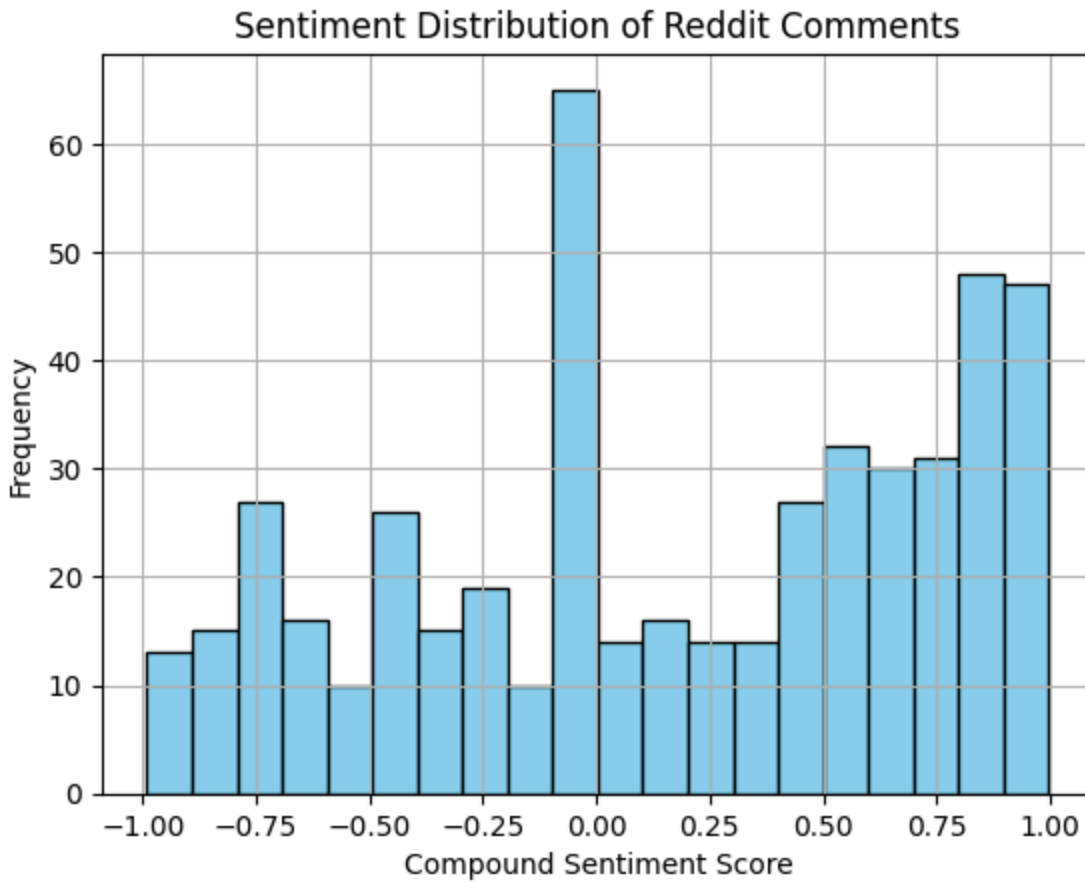
The sentiment score of 0.17 suggests that there is a positive sentiment, though slightly. Because the owner's Reddit thread is made up of comments from various authors the average sentiment score based on the comments is a good representation of the owner's sentiment. So it can be interpreted that the owner's sentiment leans towards lightly positive or neutral. Further investigation with plots using the sentiments it can be found that Reddit comments with a neutral sentiment had the highest frequency but there were also a lot of comments with a more positive frequency, which explains the average sentiment score. Lastly, it can be seen that there is a spread of different scores when looking at the plot and a sorted data frame, which means that the sentiment score can be used to generalise feelings on electric cars in general as not all Reddit users in the post are a fan of Tesla electric cars..

```

print("Average Sentiment Score:", avg_sentiment)

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
and should_run_async(code)
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data] Package vader_lexicon is already up-to-date!
Average Sentiment Score: 0.1738306748466255

```



	Author	Sentiment Score
80	Shootels	0.9976
351	iamozymandiusking	0.9972
36	cantanko	0.9911
236	tankerdudeucsc	0.9897
87	omgitsme17	0.9895
...
57	dawnsearlylight	-0.9625
185	jaymansi	-0.9625
23	03Void	-0.9670
143	Snoo93079	-0.9703
350	curious_astronauts	-0.9909

Conclusion

In conclusion, the investigation successfully used social media analytical techniques to understand the experiences of electric vehicle owners in different settings by using Tesla vehicles as an example for comparison. The results of the exploratory phase found that the model type, quality, other cars, and charging capabilities were the variables that affected electric vehicle owner experiences when comparing vehicles with one another. This conclusion in the context of the original investigation comes from the most popular phrases such as 'build quality', 'best car', and 'charging network'. The results of the topic analysis found that the topics of discussion were related to the durability of electric vehicles, the company Tesla, the price of the electric vehicles, and the comparisons of electric vehicles. In the context of the original investigation, the results establish that when purchasing a fully electric vehicle it is important for owners or future owners to understand the capabilities of the vehicle. The results of the sentiment modelling found that when generalising all the 800+ opinions of the post, it is found that the experiences are positively neutral. However, there is a large spread of different feelings amongst the authors, which establishes that the Reddit post used for the original investigation allows for fair insights.

Using statistics and trends found from the research on the current state of the industry and existing models, the first part of the approach taken for the sake of the investigation was effective. This is further proven in the data gathering part of the process, where a social media data post was found that provides 800+ comments that could be used for data exploration and text mining. Using the text mining techniques on the data the themes found from the distributed topics gave further insight on what is important to electric vehicle owners, further justifying the approach of the investigation. The sentiment analysis process confirms that the approach used a variety of different experiences to derive insights based on the results. For future work to improve the investigation process sentiment analysis could be used before the exploratory phase to determine if a data post will give fair results, multiple data posts could be used for less bias, and lastly, the use of a data post related to a less popular fully electric vehicle company used for understanding owner comparisons can be used to understand experiences.

References:

Ukpanah, I. (2024). *Electric Vehicles: A Deep Dive into the Statistics and Trends for 2024*.

GreenMatch. Available at: <https://www.greenmatch.co.uk/electric-vehicles>

Carlier, M. (2024). *Largest automobile markets based on new car registrations 2023*. Statista. Available at:

<https://www.statista.com/statistics/269872/largest-automobile-markets-worldwide-based-on-new-car-registrations/#:~:text=China%20is%20the%20largest%20automobile,rise%20of%20around%2011%20percent>.

The Energymix. (2023). *Battery Maker Claims Breakthrough, Pledges to Double EV Range in 2024*. Available at:

<https://www.theenergymix.com/battery-breakthrough-poised-to-double-ev-range-in-2024/>

Magne, C. (2024). *Bing vs. Google: A Comprehensive Comparison*. Upgrow. Available at:

<https://www.upgrow.io/blog/bing-vs-google#:~:text=Both%20search%20engines%20deliver%20high.with%20an%20accompanying%20mobile%20app>

Shahan, Z. (2021). *Tesla Model 3 Has Passed 1 Million Sales*. CleanTechnica. Available at:

<https://cleantechnica.com/2021/08/26/tesla-model-3-has-passed-1-million-sales/>