

PORSCHE SENTIMENT ANALYSIS

EXPLORING COMMUNITY THOUGHTS AND PERCEPTIONS FROM REDDIT

⌚ Data Collection interval :
from January 1, 2025 – June 30, 2025 #reddit platform



- Basir Ahmad Mehrzad (74089841)

Instructor :

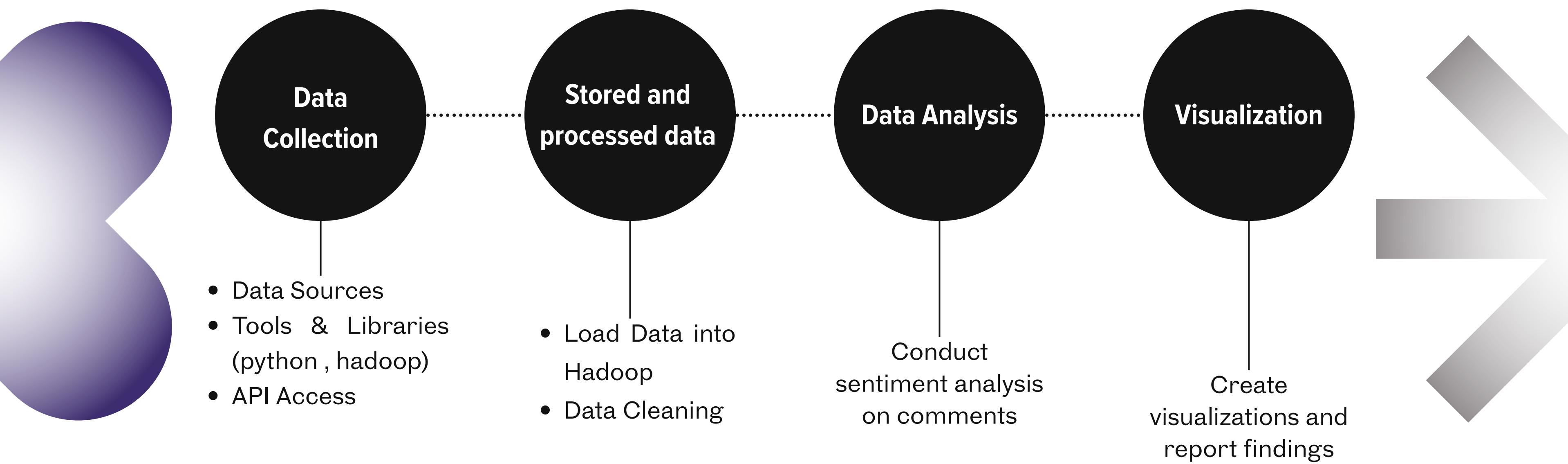
- Professor Ali Vaisifard

Course:

- Big Data & Analytics

Project phases

Key steps in our data analysis



Data Collection

JustPaste.it - Share Text × JustPaste.it - Share Text × preferences (reddit.com) × + - ×

https://www.reddit.com/prefs/apps

GET NEW REDDIT MY SUBREDDITS HOME - POPULAR - ALL - USERS | ASKREDDIT - PICS - FUNNY - MOVIES - GAMING - WORLDNEWS - NEWS - TODAYILEARNED - NOTTHEC EDIT ▾

reddit PREFERENCES options apps RSS feeds friends blocked password/email Consistent_Ask_7870 (1) | 🔍 | 📧 | 🗑️ | preferences | logout

developed applications

PorscheSentiment collecting Porsche posts by Basir Ahmad Mehrzad
personal use script cZ-gDTfE_SRbK5uv3dDHsg

change icon

secret 7SF38U7KrZ5sMlsU-600c2fJ2PU78w

developers Consistent_Ask_7870 (that's you!) remove
add developer:

name PorscheSentiment

description collecting Porsche posts by Basir Ahmad Mehrzad

about url

redirect url http://localhost:8080

update app delete app

create application

By creating an app, you agree to [Reddit's Developer Terms](#) and [Data API Terms](#). You must also [register](#) to use the API.

name PorscheSentiment

web app A web based application

installed app An app intended for installation, such as on a mobile phone

script Script for personal use. Will only have access to the developers accounts

description collecting Porsche posts

about url

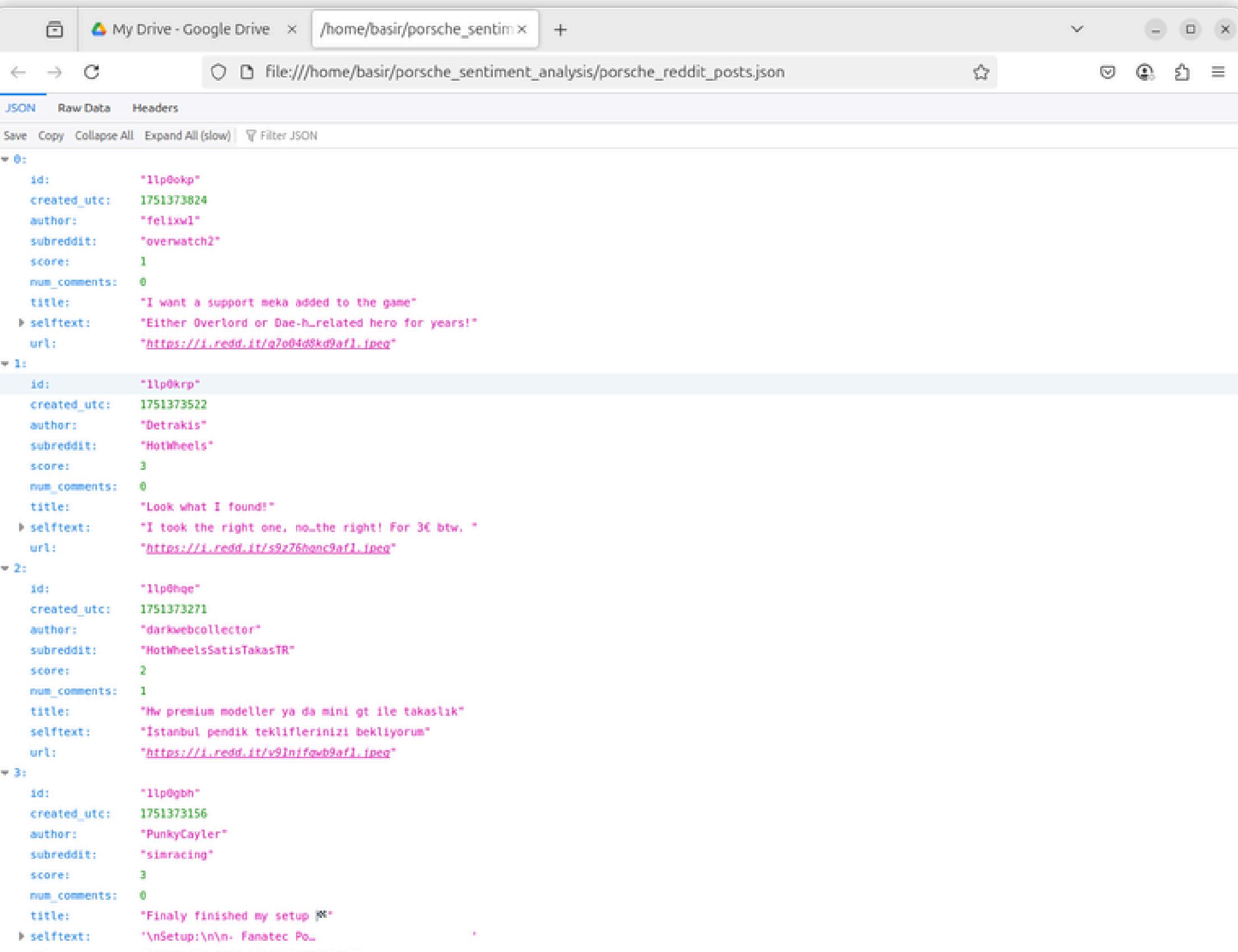
redirect url http://localhost:8080



.json file

This is the raw data file
that was collected from
Reddit. It contains
unstructured comments.

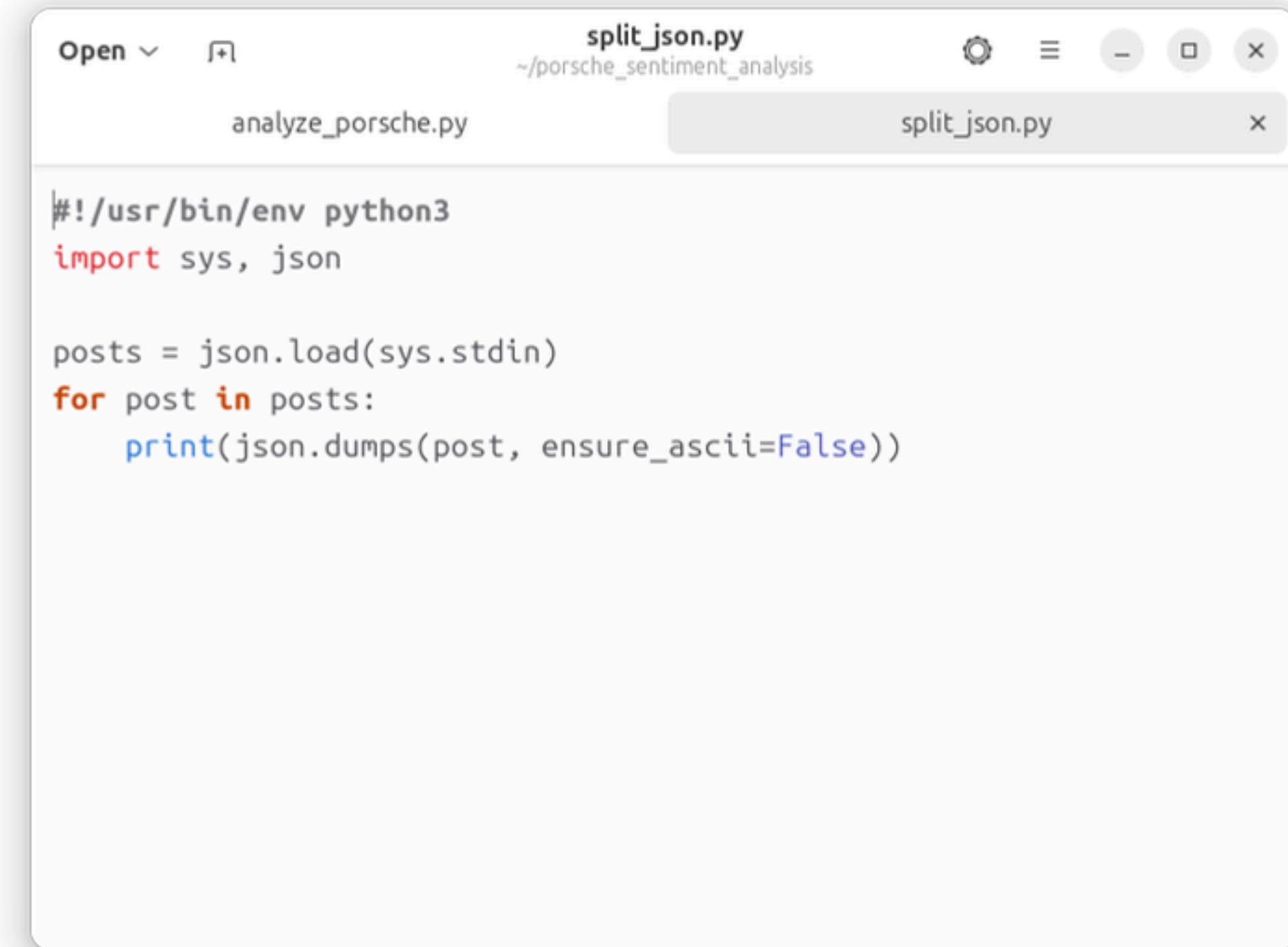
It is later cleaned and
processed for sentiment
analysis and visualization.



```
My Drive - Google Drive /home/basir/porsche_sentim +  
file:///home/basir/porsche_sentiment_analysis/porsche_reddit_posts.json  
JSON Raw Data Headers  
Save Copy Collapse All Expand All (slow) Filter JSON  
+ 0:  
id: "1lp0okp"  
created_utc: 1751373824  
author: "felixwl"  
subreddit: "overwatch2"  
score: 1  
num_comments: 0  
title: "I want a support meka added to the game"  
selftext: "Either Overlord or Dae-h-related hero for years!"  
url: "https://i.redd.it/q7e04d8kd9af1.jpeg"  
+ 1:  
id: "1lp0krp"  
created_utc: 1751373522  
author: "Detrakis"  
subreddit: "HotWheels"  
score: 3  
num_comments: 0  
title: "Look what I found!"  
selftext: "I took the right one, now the right! For 3€ btw. "  
url: "https://i.redd.it/s9z76hanc9af1.jpeg"  
+ 2:  
id: "1lp0hqe"  
created_utc: 1751373271  
author: "darkwebcollector"  
subreddit: "HotWheelsSatisTakasTR"  
score: 2  
num_comments: 1  
title: "Hw premium modeller ya da mini gt ile takaslik"  
selftext: "İstanbul pendik tekliflerinizi bekliyorum"  
url: "https://i.redd.it/v9lnifawb9af1.jpeg"  
+ 3:  
id: "1lp0gbh"  
created_utc: 1751373156  
author: "Punkycayler"  
subreddit: "simracing"  
score: 3  
num_comments: 0  
title: "Finally finished my setup 🎉"  
selftext: "\nSetup:\n\n- Fanatec Po...  
url: "https://i.redd.it/8kxwv7fjw9af1.jpeg"
```

split_json.py

This script formats the large JSON file into one comment per line. This structure is easier to process with tools like Hadoop and Python. The output is used for text cleaning



The screenshot shows a terminal window with the following details:

- File path: ~/porsche_sentiment_analysis
- File name: split_json.py
- Content of the file:

```
#!/usr/bin/env python3
import sys, json

posts = json.load(sys.stdin)
for post in posts:
    print(json.dumps(post, ensure_ascii=False))
```

analyze_porsche.py

This Python file cleans the Reddit comments — removing noise, emojis, links, and irrelevant characters. It also applies sentiment scoring using VADER. The result is saved in cleaned_posts.txt, which becomes the core for all future analysis.

```
analyze_porsche.py
~/porsche_sentiment_analysis

import pandas as pd
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import matplotlib.pyplot as plt

# 1. Load your cleaned data
#   Make sure 'cleaned_posts.txt' exists in this folder:
#   It should have lines like: post_id<TAB>cleaned_text
df = pd.read_csv('cleaned_posts.txt', sep='\t', names=['post_id','text'], dtype=str)

# 2. Compute sentiment scores with VADER
analyzer = SentimentIntensityAnalyzer()
df['sentiment'] = df['text'].apply(lambda t: analyzer.polarity_scores(t)['compound'])

# 3. Bucket into Negative, Neutral, Positive
bins = [-1.0, -0.05, 0.05, 1.0]
labels = ['Negative', 'Neutral', 'Positive']
df['category'] = pd.cut(df['sentiment'], bins=bins, labels=labels)

# 4. Count each category
counts = df['category'].value_counts().reindex(labels)

# 5. Print the distribution
print("Sentiment distribution:")
for cat, cnt in zip(labels, counts):
    print(f" {cat}: {cnt} posts ({(cnt/len(df))*100:.1f}%)")

# 6. Plot & save a pie chart
plt.figure(figsize=(6,6))
plt.pie(
    counts.values,
    labels=counts.index,
    autopct='%1.1f%%',
    startangle=90,
    wedgeprops={'edgecolor':'white', 'linewidth':1}
)
plt.title('Porsche Reddit Sentiment Distribution')
plt.tight_layout()
plt.savefig('sentiment_pie.png')
plt.close()

print("Pie chart saved to sentiment_pie.png")
```

cleaned_posts.txt

This file contains clean, processed text for each comment. It's used for both the word cloud and sentiment analysis.

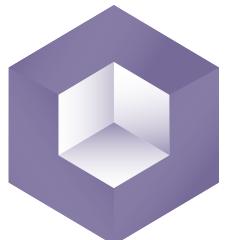
Open cleaned_posts.txt
~/porsche_sentiment_analysis

analyze_porsche.py cleaned_posts.txt

```
1lnz0ex cars want see underground get hands ultra rare hypercars one prototypes surely get hands mercedes amg one high hyper
porsche 911 gt1 stra&enversion mid hyper mid high super top sports looking clk lancia 037 stradale high c sports mazda 787b mid
super low hyper top c sports jaguar xj13 mid high c sports mid sports
1lnzozy automotive technician programs currently live las vegas nv 33 years old interested cars whole life recently decided get
automotive mechanics hobby currently pursuing bs applied mathematics minor physics looked community college aas automotive
technology master technician program motorsport technology certificate achievement meeting advisor ensure assumption motorsport
technology ca think eliminate confusion given credentials would allow work master technician motorsport technician would able
work euro cars background given get additional training dealerships training programs heard ptap porsche ferrari technician
program audi aep program would love apart experience skill set facilitate creation environment work high end automobiles heard
programs difficult get young probability accepted low perspective possible hire specific brand technician shadow would possible
still get accepted programs going uti requirement enrollment programs options aside path laid career hobby mind spending time
money set finances music production data science etc money motivating factor education skill development personal garage would
look pretty awesome even though hobby still see value acquiring appropriate certifications solidify experience reputation
capable technician insight appreciated thank
1lnzp9o taiwan property market banking industry perspective plenty talk taiwan housing bubble almost none actually happens
behind scenes banking side thought might interest anyone equally baffled madness taiwan housing market background work adjacent
taiwan financial industry personally worked multiple major banks taiwan course career either product team senior management
level accountant commentary analysis purely reading insanity taiwan banking sector enjoy context first taiwan one developed
economies escaped 2008 financial crisis relatively unscathed also means missed wake call major reforms systemic audit risk
exposure real restrictions housing finance operates whilst us europe got meaningful regulation carried crazy dialled 11 past 25
years property prices skyrocketed wages barely moved mortgages gotten bigger longer somehow interest rates bar mortgage approval
stayed low even though household incomes budged two decades holdings liquidity taiwanese banks keep scale prices market demanded
get increasingly creative leveraging regulatory loopholes self assessed risk models capital rules allow lend many multiples
actual holdings mortgages actually work taiwan moment sign mortgage agreement taiwan bank generates digital credit account books
mortgage asset balance sheet money come bank deposits holdings created thin air system works called capital adequacy ratios car
rules dictate much bank lend based capital actually hold taiwan globally basel iii banks required maintain something like 8
capital total risk weighted assets weird part mortgages considered low risk taiwan often assigned risk weight 40 even lower
means bank issue 25 times real holdings mortgages basically debt leverage steroids regulation oversight approves risk models
banks taiwan banking sector regulatory capture complete banks design risk models audit exposure simply report every 24 months
operate capital requirements let leverage 25 1 mortgages like letting casinos set odds audit books government may review
periodically entire system relies heavily self reporting including default rate closed book internal modelling assumption house
prices always go tl dr taiwan banks issue 25 times holdings debt forms loans mortgages new loan mortgage allows bank lend even
imaginary new money created buy house considered asset backed value property mean real world sellers walk away real liquid money
sellers flex neighbours department store shoppers buying porsche suv expensive repair difficult drive safely taiwan narrow
streets buyers get 30 40 years debt often retirement age assuming job done ai next year bank collects decades interest buyer
defaults takes back house something real exchange money never actually existed whilst behind scenes every new mortgage creates
fresh interest bearing asset bank new assets improve bank financial standing basel iii rules turn justifies issuing even loans
using capital base actually increased material way would banks operate way pure profit maximisation socialised risk profitable
```

Additional Processing Using Hadoop

To demonstrate scalability and distributed processing, we performed additional sentiment analysis using the Hadoop framework. A JSON file containing Reddit comments was uploaded to HDFS. We used a mapper.py and reducer.py script to classify sentiments as Positive, Negative, or Neutral using the TextBlob library. The output was a frequency count of each sentiment category.



Hadoop HDFS – Web UI Overview

This interface shows the Hadoop cluster running in our local environment. It provides real-time monitoring of resource usage, job tracking, and data blocks. This web-based tool is crucial to verify the health and execution of jobs across the Hadoop Distributed File System (HDFS).

The screenshot shows a web browser window titled "Social Media Sentiment A X" with the tab "Namenode information" selected. The URL is "localhost:9870/dfshealth.html#tab-overview". The page title is "Hadoop" and the navigation bar includes "Overview", "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities".

Overview 'localhost:9000' (✓active)

Started:	Tue Jul 01 12:55:12 +0000 2025
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaeecc760
Compiled:	Mon Mar 04 06:35:00 +0000 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-b76d4f87-8849-4f89-9249-001e18a1d876
Block Pool ID:	BP-510352726-10.0.2.15-1751228231136

Summary

Security is off.

Safemode is off.

5 files and directories, 1 blocks (1 replicated blocks, 0 erasure coded block groups) = 6 total filesystem object(s).

Heap Memory used 216.53 MB of 276.5 MB Heap Memory. Max Heap Memory is 1.26 GB.

Non Heap Memory used 53.76 MB of 55.09 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Uploading JSON to HDFS with put

To begin Hadoop processing, we first uploaded the JSON file containing Reddit comments to HDFS using the hdfs dfs -put command. This made the data accessible across the Hadoop cluster for MapReduce processing.

```
basir@ubuntu-hadoop: ~/porsche_sentiment_analysis
basir@ubuntu-hadoop: ~/porsche_sentiment_analysis
basir@ubuntu-hadoop: ~/porsche_sentiment_analysis

7818 NodeManager
7246 DataNode
7118 NameNode
(venv) basir@ubuntu-hadoop:~/hadoop$ hdfs dfs -mkdir -p /data/porsche
(venv) basir@ubuntu-hadoop:~/hadoop$ hdfs dfs -put -f porsche_reddit_posts.json /data/porsche/
put: 'porsche_reddit_posts.json': No such file or directory
(venv) basir@ubuntu-hadoop:~/hadoop$ hdfs dfs -put -f porsche_reddit_posts.json /data.porsche/
put: '/data.porsche/': No such file or directory: 'hdfs://localhost:9000/data.porsche'
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ hdfs dfs -put -f porsche_reddit_posts.json /data/porsche/
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ hdfs dfs -ls /data/porsche
Found 1 items
-rw-r--r--  1 basir supergroup  291513 2025-07-01 13:03 /data/porsche/porsche_reddit_posts.json
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ nano split_json.py
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ nano split_json.py
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ chmod +x split_json.py
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ cat > split_json.py << 'EOF'
>#!/usr/bin/env python3
import sys, json

posts = json.load(sys.stdin)
for post in posts:
    print(json.dumps(post, ensure_ascii=False))
> EOF
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ chmod +x split_json
chmod: cannot access 'split_json': No such file or directory
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ chmod +x split_json.py
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ ./split_json.py < porsche_reddit_posts.json > posts_per_line.json
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ hdfs dfs -mkdir -p /data/porsche
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ hdfs dfs -put -f posts_per_line.json /data/porsche/
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$ hdfs dfs -ls /data/porsche
Found 2 items
-rw-r--r--  1 basir supergroup  291513 2025-07-01 13:03 /data/porsche/porsche_reddit_posts.json
-rw-r--r--  1 basir supergroup  280804 2025-07-01 13:17 /data/porsche/posts_per_line.json
(venv) basir@ubuntu-hadoop:~/porsche_sentiment_analysis$
```

Hadoop MapReduce Workflow

This slide shows how Hadoop's Mapper and Reducer scripts were used to clean and prepare Reddit comment data.

mapper.py cleaned each line of JSON input by removing emojis, links, and stopwords, and emitted cleaned text.

reducer.py ensured unique post IDs by removing duplicates.

The final output, cleaned_posts.txt, became the core dataset used for sentiment analysis and word cloud generation later in the project.

Raw Reddit Data | (JSON file)

| mapper.py |
| - Cleans text |
| - Removes links |
| - Removes emojis |
| - Removes stopwords |

| reducer.py |
| - Groups by ID |
| - Removes dupes |

cleaned_posts.txt output |
(Used in sentiment analysis) |

Hadoop Mapper: Reddit Comment Cleaner

This Mapper script is the first step of our Hadoop process. It reads each Reddit post, extracts the text, cleans it by removing emojis, links, and common stopwords, and emits the cleaned result. This is essential for preparing data for sentiment analysis.

```
basir@ubuntu-hadoop:~/porsche_sentiment_analysis
```

```
GNU nano 7.2                                mapper.py *
```

```
#!/usr/bin/env python3
import sys, json, re
from nltk.corpus import stopwords

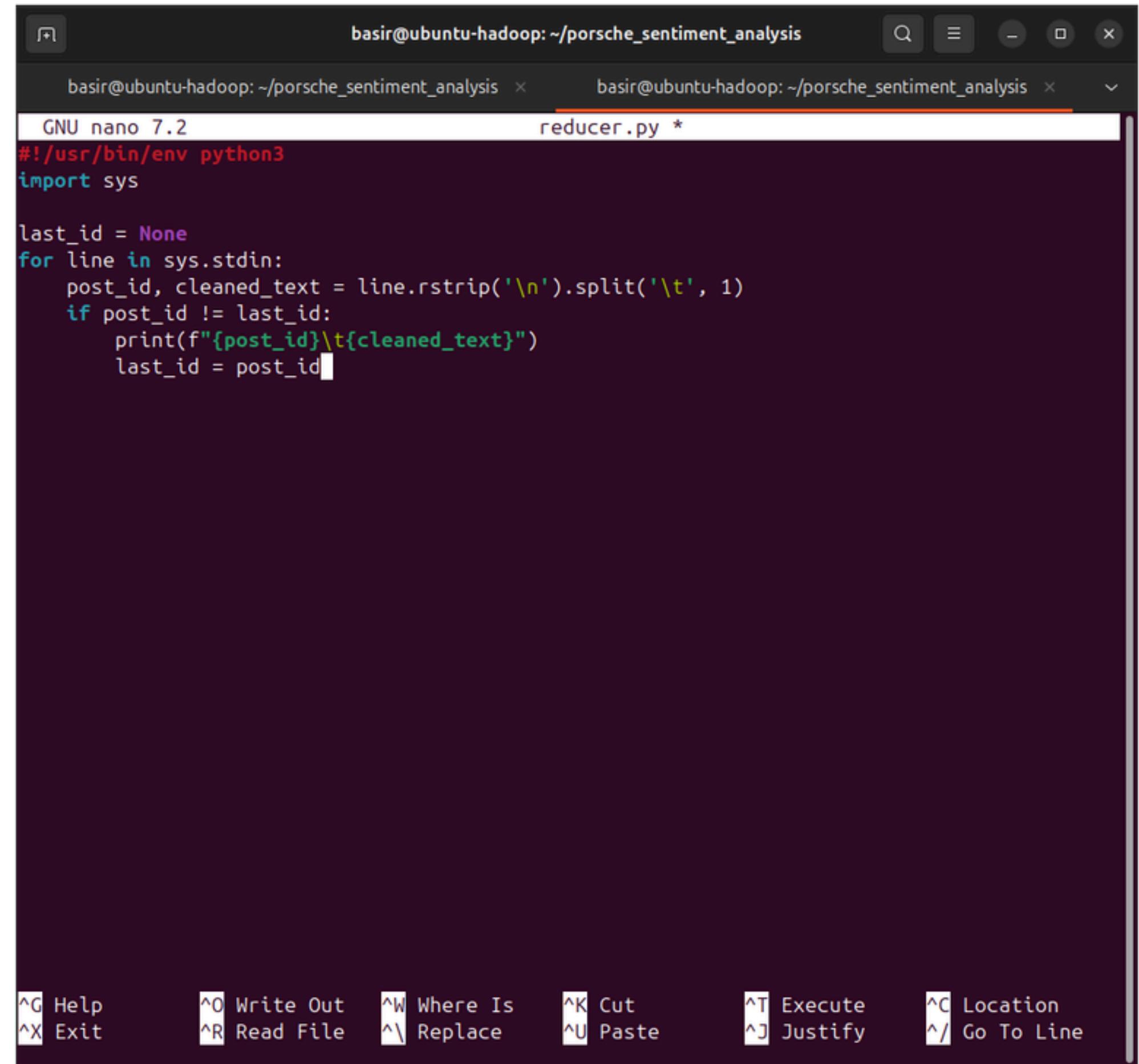
# Patterns
URL_RE    = re.compile(r'https://\S+')
EMOJI_RE = re.compile(
    "["
    u"\U0001F600-\U0001F64F" # emoticons
    u"\U0001F300-\U0001F5FF" # symbols & pictographs
    "]+", flags=re.UNICODE)

stops = set(stopwords.words('english'))

for line in sys.stdin:
    post = json.loads(line)
    text = (post.get('title','') + ' ' + post.get('selftext','')).lower()
    text = URL_RE.sub('', text)
    text = EMOJI_RE.sub('', text)
    tokens = re.findall(r'\b\w+\b', text)
    cleaned = [t for t in tokens if t not in stops]
    if cleaned:
        # emit: postID<TAB>cleaned_text
        print(f"{post['id']}\t{' '.join(cleaned)})
```

Hadoop Reducer: Sentiment Categorizer

This Reducer script eliminates duplicate posts. It ensures that only the first instance of each post_id is kept. This helps us maintain data quality before performing sentiment analysis or word frequency visualization.



The screenshot shows a terminal window titled "reducer.py *". The window has three tabs at the top: "basir@ubuntu-hadoop: ~/porsche_sentiment_analysis", "GNU nano 7.2", and "reducer.py *". The main area of the terminal displays the Python code for a reducer script:

```
GNU nano 7.2
#!/usr/bin/env python3
import sys

last_id = None
for line in sys.stdin:
    post_id, cleaned_text = line.rstrip('\n').split('\t', 1)
    if post_id != last_id:
        print(f"{post_id}\t{cleaned_text}")
    last_id = post_id
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts:

- ^G Help
- ^X Exit
- ^O Write Out
- ^R Read File
- ^W Where Is
- ^Y Replace
- ^K Cut
- ^U Paste
- ^T Execute
- ^J Justify
- ^C Location
- ^/ Go To Line

generate_counts_tsv.py

This script takes the cleaned text and uses the VADER sentiment tool to classify comments as Positive, Neutral, or Negative. It counts how many of each type and saves them in a .tsv file.

```
Open ⌂ generate_counts_tsv.py
analyze_porsche.py cleaned_posts.txt generate_counts_tsv.py x
~/porsche_sentiment_analysis

# generate_counts_tsv.py
import pandas as pd
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

df = pd.read_csv('cleaned_posts.txt', sep='\t', names=['post_id', 'text'], dtype=str)
analyzer = SentimentIntensityAnalyzer()
df['sentiment'] = df['text'].apply(lambda t: analyzer.polarity_scores(t)['compound'])

bins = [-1.0, -0.05, 0.05, 1.0]
labels = ['Negative', 'Neutral', 'Positive']
df['category'] = pd.cut(df['sentiment'], bins=bins, labels=labels)
counts = df['category'].value_counts().reindex(labels)

# ✓ Export to counts.tsv
counts.to_csv('counts.tsv', sep='\t', header=False)
print("✓ counts.tsv file generated.")
```

Pie Chart Generation

A Python script (generate_sentiment_pie.py) was used to visualize the sentiment distribution in the form of a pie chart.

To clearly communicate the proportion of positive, negative, and neutral sentiments in the collected Reddit comments.

The script reads a .tsv file (counts.tsv) containing sentiment labels and their counts, then creates a pie chart using matplotlib.

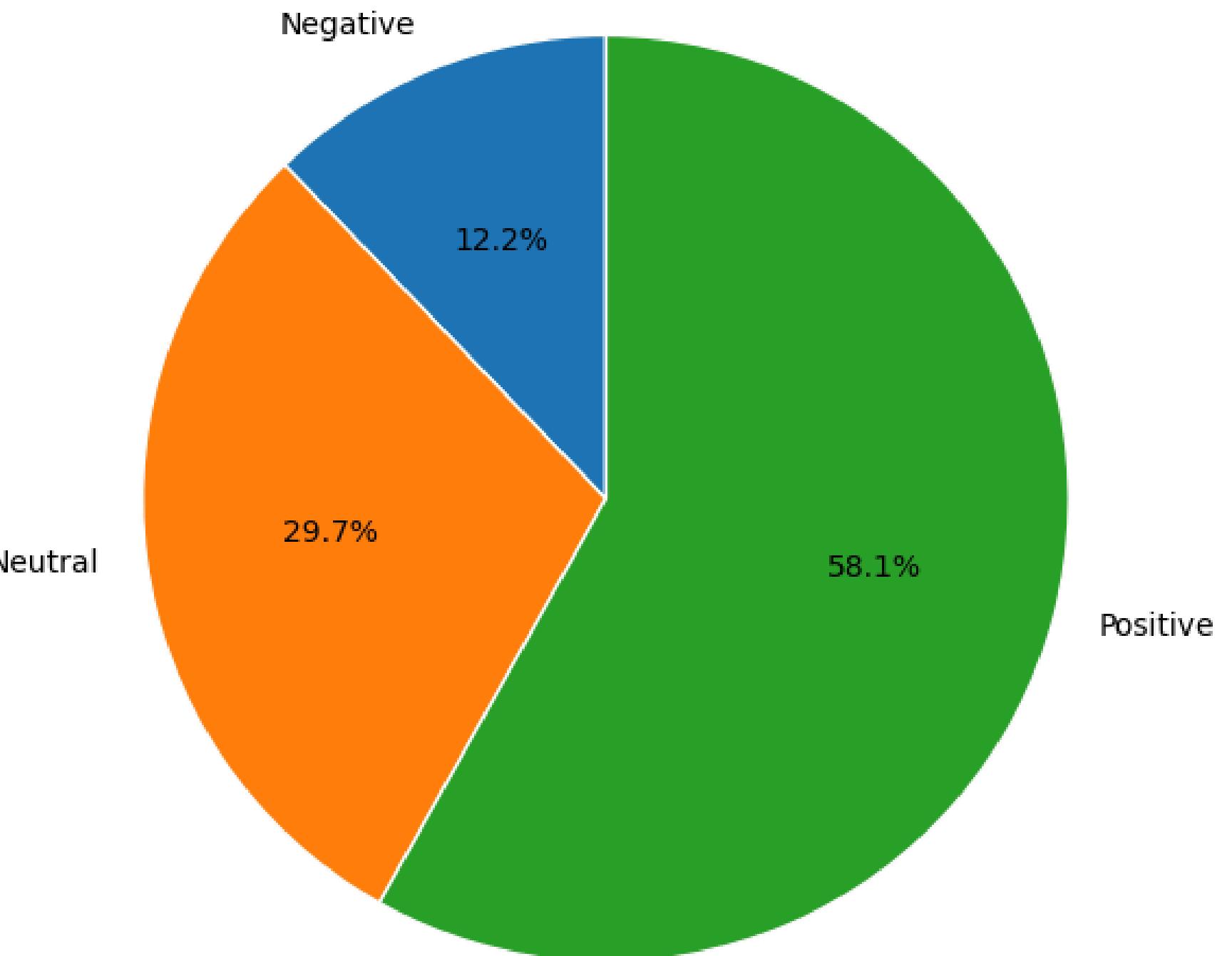
📦 Libraries Used:

matplotlib.pyplot: for creating and saving the pie chart

csv: for reading the .tsv file (tab-separated values)

sys: for command-line arguments (file input/output)

Porsche Reddit Sentiment Distribution



"you can find the .py file in Appendices"

58.1%

Positive Sentiment

58.1 % of comments reflect user satisfaction with the car brand.

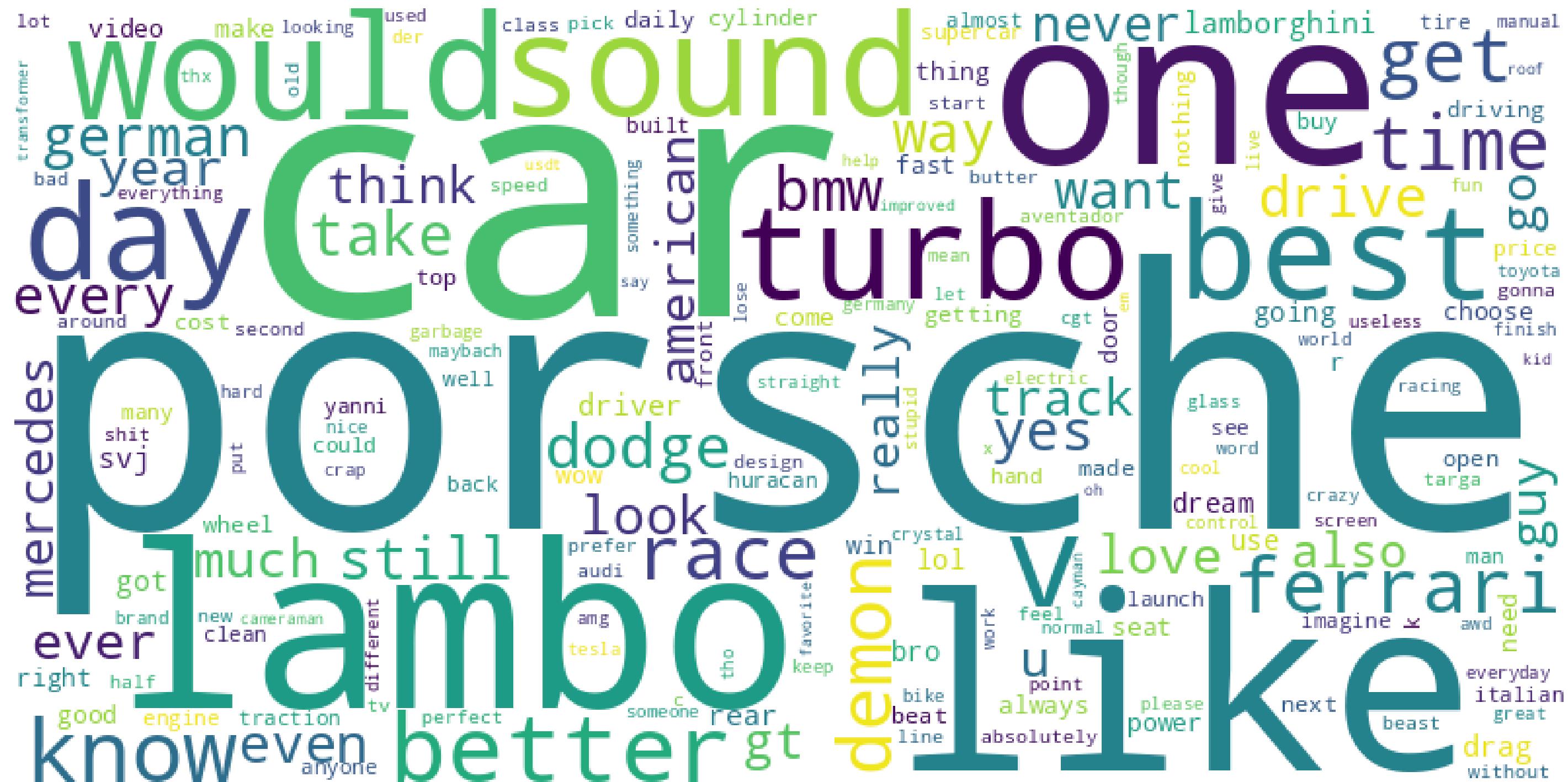
Neutral Sentiment

29.7 % of comments show neutral or informational feedback about the car brand.

Negative Sentiment

12.2 % of comments express user dissatisfaction with the car brand.

Word Cloud – Most Common Words in Porsche Reddit Comments

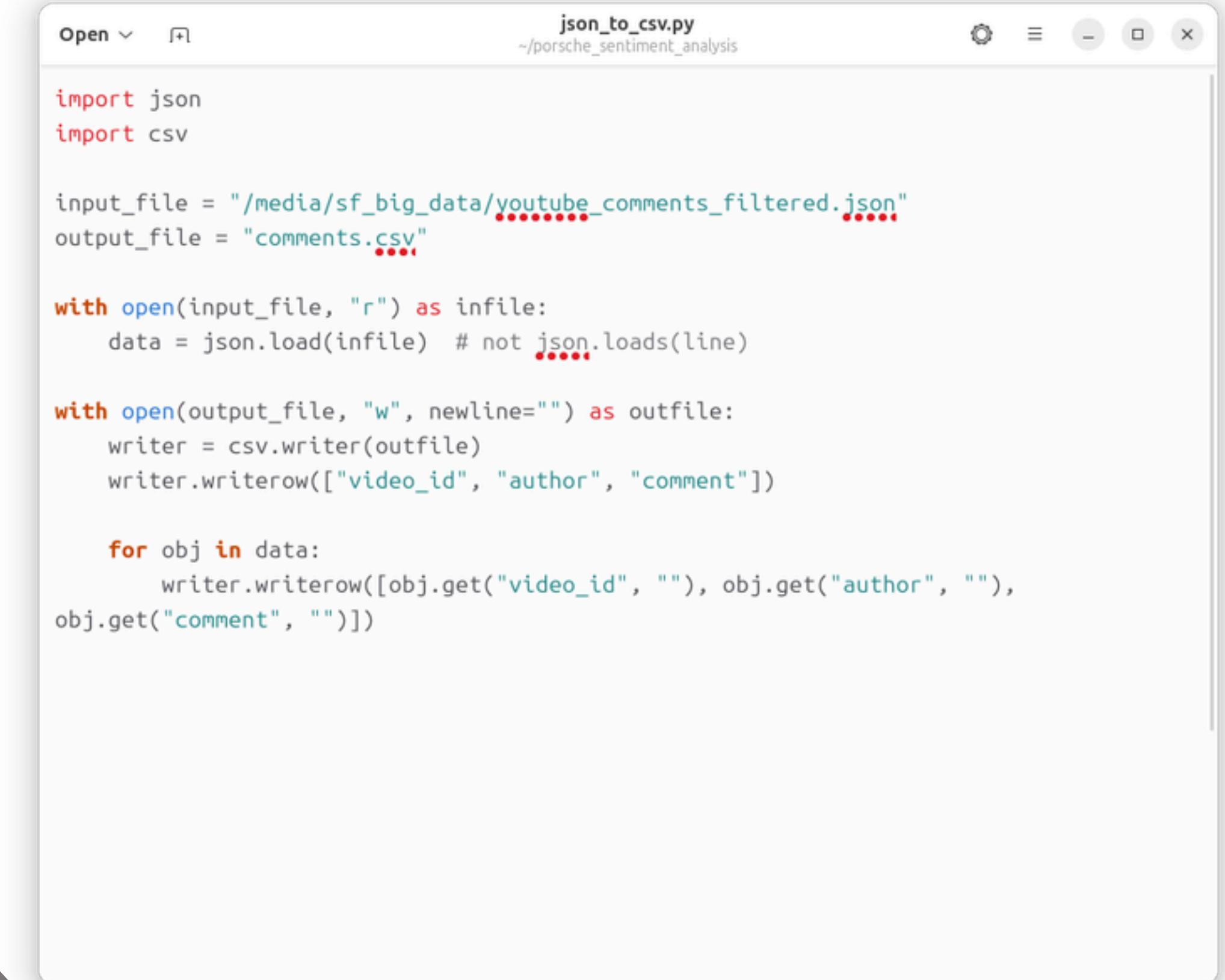


A Python script (generate_wordcloud.py) was used to create this image. "you can find the .py file in Appendices" Libraries used:

- wordcloud – generate the visual
 - matplotlib – save/display the image
 - nltk – clean and remove common stopwords
 - re – clean text with regular expressions

json_to_csv.py

This script transforms the Reddit data into a structured .csv file with columns like author, video ID, and comment. This file is required for visual analysis in Tableau.



The screenshot shows a code editor window titled "json_to_csv.py" with the file path "~/porsche_sentiment_analysis". The code is written in Python and performs the following tasks:

- Imports the json and csv modules.
- Sets the input_file to "/media/sf_big_data/youtube_comments_filtered.json" and the output_file to "comments.csv".
- Opens the input file in read mode ("r") and loads the data into a variable named data.
- Creates a writer object using csv.writer for the output file.
- Writes the header row ["video_id", "author", "comment"] to the output file.
- Iterates over each object in the data list, writing a row for each with values from obj.get("video_id", ""), obj.get("author", ""), and obj.get("comment", "") respectively.

```
import json
import csv

input_file = "/media/sf_big_data/youtube_comments_filtered.json"
output_file = "comments.csv"

with open(input_file, "r") as infile:
    data = json.load(infile) # not json.loads(line)

with open(output_file, "w", newline="") as outfile:
    writer = csv.writer(outfile)
    writer.writerow(["video_id", "author", "comment"])

    for obj in data:
        writer.writerow([obj.get("video_id", ""), obj.get("author", ""),
                        obj.get("comment", "")])
```

comments.csv

This is the final dataset used in Tableau. It connects raw Reddit data to visual tools by putting everything in rows and columns.

Open ▾ comments.csv
~/porsche_sentiment_analysis

json_to_csv.py comments.csv

video_id	author	comment
DAZ_742v37E	@DollyBoy_1923	Is nobody going to mention the pool noodles being used as protection?
DAZ_742v37E	@RickeGriffins,	🧠🚀
DAZ_742v37E	@AsliddinXudoyorov-yv7hh	Inshaalloh anawu mowinadan sotib olaman
DAZ_742v37E	@MiguelBorgesphotography	nice
DAZ_742v37E	@ArcieMayManuel	I wish I have that car❤️❤️😊😊🎉🎊😍😍😊😊😊
DAZ_742v37E	@аўтамабіль	,гэта добры gt3rs!
DAZ_742v37E	@scnniFN	My dream😢
DAZ_742v37E	@domingoferrari5889	I want a 6 speed porsche 911 and thats a beauty but prob a gts so i can have seats for legless passengers in the back
DAZ_742v37E	@Santi_editz23xd	"911:X 992:✓"
DAZ_742v37E	@gavrad7060	closest experience I get is my 43" TV
d3sZdZatcUg	@CesarsChariot	That's fire 🔥 right there
d3sZdZatcUg	@CaptainKedah	"The Best ""i made it Porsche""
d3sZdZatcUg	@SeanF-m4b	It's actually my dream car 🚗
d3sZdZatcUg	@PixelClutch01	Car Lover's Button 👍✓
d3sZdZatcUg	@ErinGoh-x9y	"As a 29year-old kid, I can make 30-40 USD a day, I only invested 50USDT, now I have a lot, I can make 300-800 USD a month, I not only improved my financial situation, but also learned a lot of new knowledge and met friends from all over the world. In the future, I will continue to work hard to help more people realize their dreams"
d3sZdZatcUg	@carlaantonelli8865	"Targa 911 or Cayman gts, I bought the cayman gts 2025, i say from exerience the Cayman wins , what a fun ride 😊"
d3sZdZatcUg	@annablakemore8000	My favorite car
d3sZdZatcUg	@ParentPagsolingan	Be careful your car is gonna broke
d3sZdZatcUg	@ParentPagsolingan	Cool porsche bro i think it cost 100.00 dollars😊
d3sZdZatcUg	@ParentPagsolingan	But its not like titan tv man😊
d3sZdZatcUg	@ParentPagsolingan	bro if a porsche race in a bugati divo into 1967 motor bike😊
d3sZdZatcUg	@MaxGavilanes	Oh My God Bro it's a Transformer
d3sZdZatcUg	@Ospecialman	"And when the roof seals start to leak. which they will. you arrange the

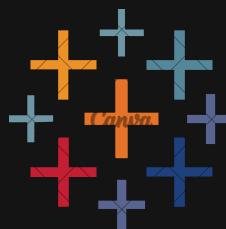


TABLEAU VISUALIZATIONS



1. Bar Chart – Most Active Users (Top Commenters)

- What it shows:

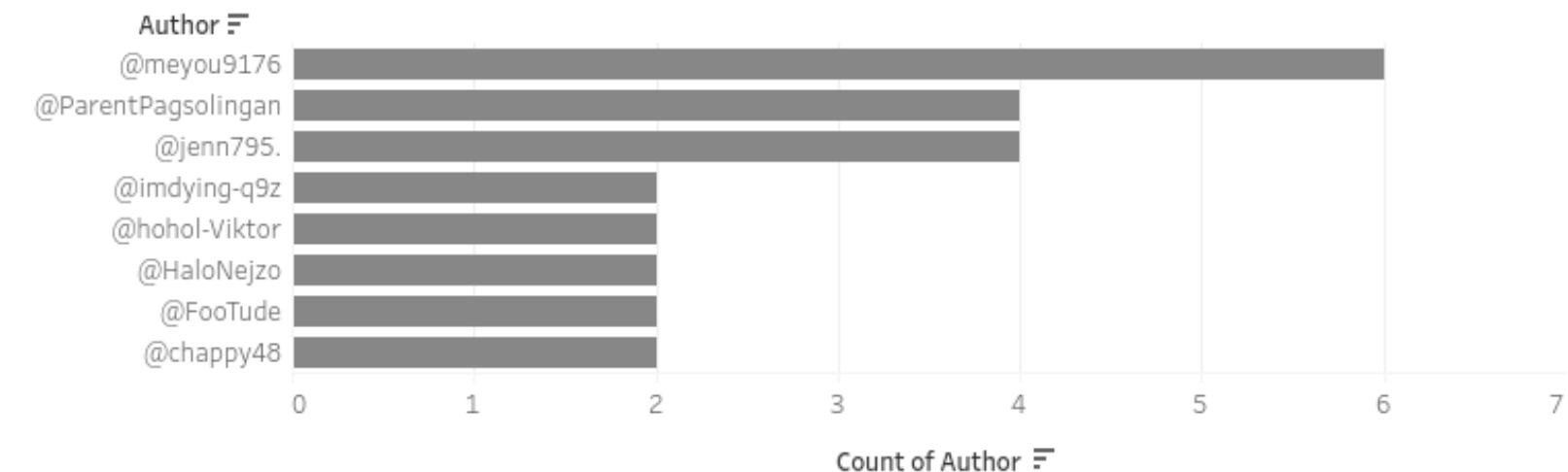
This bar chart displays the users who commented the most across all Reddit posts in your dataset.

- Why it's useful:

Helps identify who is contributing the most in Porsche discussions.

You can detect influencers, fans, or repeat commenters.

Bar Chart - Most Active Users (Top Commenters)



2. Bar Chart – Comments per Video

- What it shows:

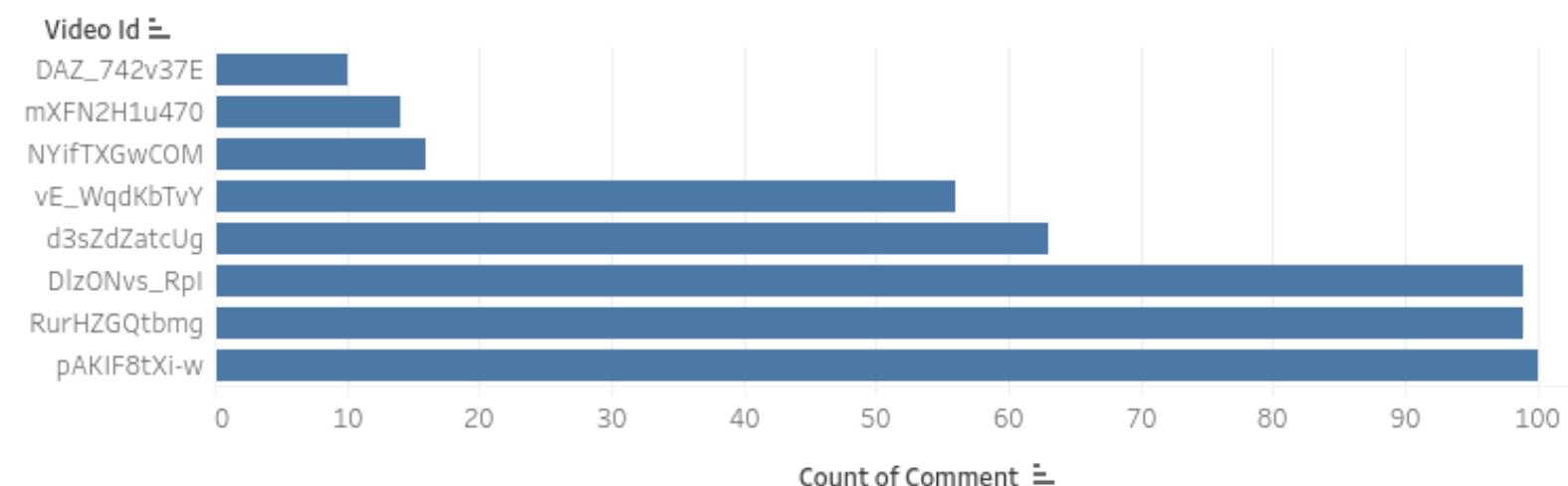
This chart summarizes how many comments each video (Reddit post) received.

- Why it's useful:

Lets you find which Porsche posts attracted the most attention or controversy.

It shows which topics/posts resonated with the community.

Comments per Video



3. Word Frequency (Word Count by Author)

- What it shows:

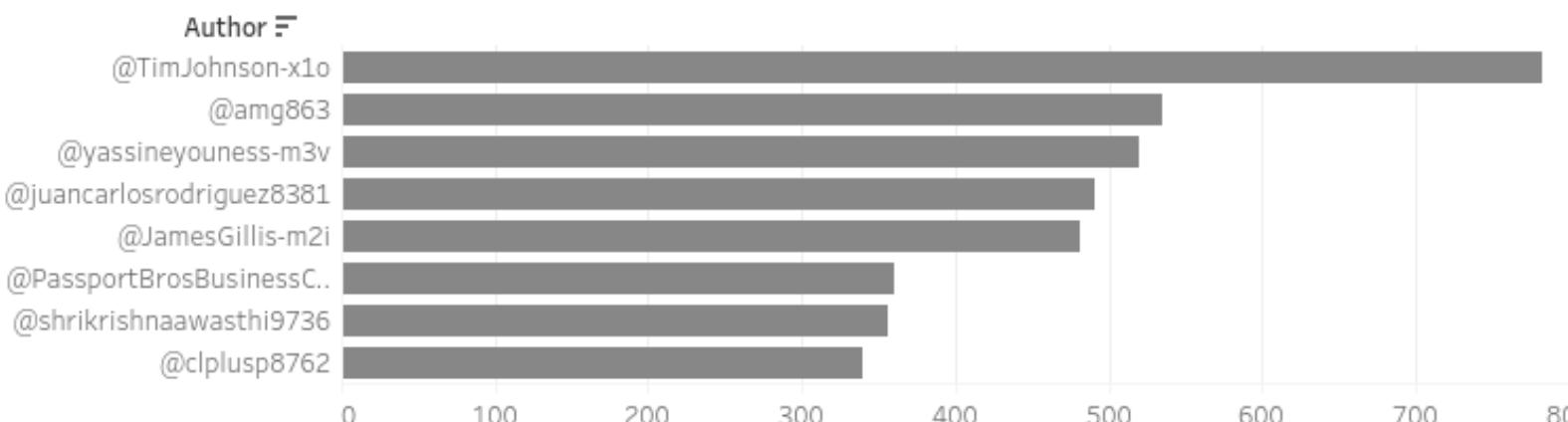
This bar chart shows the total number of words or characters used by each commenter (approximation via comment length).

- Why it's useful:

Shows which users are giving thoughtful, longer feedback.

May help find users writing detailed reviews or passionate responses.

Word frequency



INSIGHTS & RECOMMENDATIONS FOR PORSCHE



PORSCHE

- Porsche-related Reddit posts show mostly positive sentiment.
- A few active users dominate the conversation.
- Word cloud reveals focus on models like 911, Turbo, and EVs.
- Big Data tools helped convert raw Reddit chatter into brand insights.
- Future analysis could include longer time ranges and comparisons with competitors.



Thank You

We appreciate your attention
and welcome your questions.

In short, our analysis of Porsche discussions on Reddit from Jan to June 2025 found a predominantly positive sentiment. We used Python and Hadoop to analyze data, carried out sentiment classification, and visualized results with Tableau and word clouds. The process shows how Big Data technology is able to take online discourse and turn it into actionable insights for brands..