Name : Siddhant Gaikwad
Student id : 47719848

THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

ANALYSIS OF FACEBOOK LIBRARY API DATA USING BIG DATA ANALYTICS

by Siddhant Gaikwad, 47719848

School of Information Technology and Electrical

Engineering, The University of Queensland.

Submitted for Master of Data Science, DATA7201
Supervisor Dr. Gianluca Demartini

20th May 2024

Name : Siddhant Gaikwad
Student id : 47719848

STRUCTURED ABSTRACT

1. **Title :** Big Data Analysis of Facebook Ad Library API Dataset using PySpark

2. **Introduction :** This report explores a dataset of Facebook Ads targeted at Australian users from March 2020 to February 2024, encompassing the recent federal election and Voice referendum. The report also tells us the importance and the motivation driving the need for the distributed system solutions in the realm of big data analytics. It investigates the challenges that are associated with managing and processing large-scale and intricate datasets, and underscore the practical examples that demonstrate the critical role of distributing computing in addressing these issues.

3. **Methods :** Report examines the attributes of the big data and the constraints of conventional computer methods. It explores the concept of distributed systems and their ability to meet the demands of scalability, processing speed, and fault tolerance in the field of big data analytics. Employing PySpark on the DATA7201 cluster, the study investigates ad volume trends over time for specific topics, demographic targeting by political pages, and spending patterns throughout the election campaign.

4. **Results :** The results section analysis thus demonstrates how the big data analytics when applied to social media platforms can illuminate the motivations behind the distributed system solutions. By examining the advertising trends, insights can be gleaned regarding targeted campaign strategies.

5. **Discussions & Conclusions :** In this section we intensify the advantages of distributed systems like PySpark, Map-Reduce etc, for overcoming the big data hurdles. PySpark's strengths lie in its capacity to handle massive datasets, leverage parallel processing power, ensure fault tolerance, and seamlessly integrate with existing big data infrastructure.

Name : Siddhant Gaikwad
Student id : 47719848

TABLE OF CONTENTS

Name : Siddhant Gaikwad
Student id : 47719848

INTRODUCTION

This report dives into political advertising on Facebook in Australia (March 2020 - Feb 2024) using big data tools. We analyse Facebook Ad Library data to understand online political discourse around the 2022 election and 2023 referendum.

Big data analytics, analysing massive and complex datasets like social media data, requires tools like PySpark for efficient processing. It extracts valuable insights from structured, semi-structured, and unstructured data to inform decision-making in areas like business intelligence, customer churn analysis, Fraud Detection and Recommendation systems etc.

Remember the Four V's of Big data, where big data presents unique challenges that traditional data analysis methods struggle to overcome. The challenges include Velocity (It is the speed at which the data is generated and needs to be processed is a hurdle. Where traditional methods often struggle with **real-time** or **near real-time** data streams), Variety (The diversity of data types encountered in big data is immense). Big data analytics tools must be adept at handling this **heterogeneity** to extract meaningful insights. Then comes Volume (sheer **immensity** of data generated in today's world is staggering. Social media platforms, sensor networks, and financial transactions all contribute to a data deluge), and finally we have Veracity (Ensuring the **accuracy** and **completeness** of data is crucial for reliable analysis). It also includes the application of cutting-edge technologies like Cloud Computing, and distributed computing frameworks like (Apache Hadoop/Hive, Pig Latin, and PySpark/Spark). The Data Pre-Processing (cleansing, transforming), storage and management, analysis part, and story telling of data for our stakeholders/audience.

The need for HDFS arises from the essential requirements of fault tolerance, high availability, and scalability. These attributes are crucial: a dataset of 39.58 million records generated or streamed over a period of four years.

Name : Siddhant Gaikwad
Student id : 47719848

MAIN FOCUS OF THE ANALYSIS

The objective of the analysis was to first look at the overall image of the dataset before delving deeper into it by selecting subsets of the datasets based on different concerns and accounts. So the analysis include :

A. Look at ad volume over time for a certain topic.
B. Focus on certain accounts (e.g., Facebook pages supporting a certain party and see which demographic segments they target most)
C. Look at URLs included in ads to understand which internet domains are most popular during the campaign.
D. Look at the specific event or hashtag and who is talking about it.
E. Look at spend per demographic group during an election campaign.
F. Look at the duration of ad campaigns over topics and political alignment.

DATASET DESCRIPTION

The Facebook Ad Library API data set consists of sponsored political posts on Facebook that are focused on Australian users between March 2020 and February 2024, which also includes the period preceding the latest Australian Federal election in May 2022 and the Voice referendum in October 2023. On the next page is an image of the schema of the dataset that consists of the attributes like ad_creation_time, ad_creative_body , ad_creative_link_description and impressions based on the demographics such as age, gender and regions. The demographic_distribution and region_distribution, which are array-struct type features, and impressions were struct type features. The data given is in the format of JSON with 39.53 M records.

Name : Siddhant Gaikwad
Student id : 47719848

```
[27]: data.printSchema()
```

```
root
 |-- ad_creation_time: string (nullable = true)
 |-- ad_creative_bodies: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- ad_creative_body: string (nullable = true)
 |-- ad_creative_link_caption: string (nullable = true)
 |-- ad_creative_link_captions: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- ad_creative_link_description: string (nullable = true)
 |-- ad_creative_link_descriptions: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- ad_creative_link_title: string (nullable = true)
 |-- ad_creative_link_titles: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- ad_delivery_start_time: string (nullable = true)
 |-- ad_delivery_stop_time: string (nullable = true)
 |-- ad_snapshot_url: string (nullable = true)
 |-- bylines: string (nullable = true)
 |-- currency: string (nullable = true)
 |-- delivery_by_region: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- percentage: string (nullable = true)
 |    |    |-- region: string (nullable = true)
 |-- demographic_distribution: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- age: string (nullable = true)
 |    |    |-- gender: string (nullable = true)
 |    |    |-- percentage: string (nullable = true)
 |-- estimated_audience_size: struct (nullable = true)
 |    |-- lower_bound: string (nullable = true)
 |    |-- upper_bound: string (nullable = true)
 |-- funding_entity: string (nullable = true)
 |-- id: string (nullable = true)
 |-- impressions: struct (nullable = true)
 |    |-- lower_bound: string (nullable = true)
 |    |-- upper_bound: string (nullable = true)
 |-- languages: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- page_id: string (nullable = true)
 |-- page_name: string (nullable = true)
 |-- publisher_platforms: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- region_distribution: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- percentage: string (nullable = true)
 |    |    |-- region: string (nullable = true)
 |-- spend: struct (nullable = true)
 |    |-- lower_bound: string (nullable = true)
 |    |-- upper_bound: string (nullable = true)
```

Next I will provide some detailed description of the data using images.

1) Displaying the lowest and highest values for each column.
2) Display the count of records in a feature, so indirectly indicating the amount of rows with null values.

Name : Siddhant Gaikwad
Student id : 47719848

[5]:
```
data.describe().toPandas()
```

```
24/05/19 00:44:08 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be ad
sted by setting 'spark.sql.debug.maxToStringFields'.
```

[5]:

	summary	ad_creation_time	ad_creative_body	ad_creative_link_caption	ad_creative_link_description	ad_creative_link_title	ad_delivery_start_time	a
0	count	39584139	9568757	8249081	6875053	8327851	39584139	
1	mean	None	None	None	None	2020.2068965517242	None	
2	stddev	None	None	None	None	0.4122508203949039	None	
3	min	2019-12-26		"Prospective Issues, 2021" with Ross Cameron	\n		2019-12-26	
4	max	2024-03-21	🥵 Have you had ENOUGH of the bs, lack of clim...	普通话性别平等问答 Mandarin Q & A on Gender Equality	🪶 Gentle on your hair & our planet 🌱	🪶 Use code RETURN15 to get 15% Off now 🌐	2024-03-21	

[26]:
```
from pyspark.sql.functions import col, sum as spark_sum

null_counts = data.agg(*(spark_sum(col(c).isNull().cast("int")).alias(c) for c in data.columns))
```

[27]:
```
null_counts.toPandas().T
```

[27]:

	0
ad_creation_time	0
ad_creative_bodies	11669121
ad_creative_body	30015382
ad_creative_link_caption	31335058
ad_creative_link_captions	11546859
ad_creative_link_description	32709086
ad_creative_link_descriptions	25558747
ad_creative_link_title	31256288
ad_creative_link_titles	15805496
ad_delivery_start_time	0
ad_delivery_stop_time	39075947
ad_snapshot_url	0
bylines	33538360
currency	23846574
delivery_by_region	34388094
demographic_distribution	25665988
estimated_audience_size	33517647
funding_entity	23973162
id	0

Name : Siddhant Gaikwad
Student id : 47719848

PRE-PROCESSING

DATA CLEANING

In the data cleaning part, what I have done is, wherever there were null values present in the dataset, replaced them with empty string.

```
7]: data_cleaned = data.na.fill({"ad_creative_body": "","ad_creative_link_caption": "",
                              "ad_creative_link_title": "","ad_creative_link_description": "",
                              "funding_entity": "","page_name": "", "ad_delivery_stop_time": "2028-01-31" })
```

```
[10]: data_cleaned_4 = data_cleaned_4.filter((col("ad_creative_body") != "Unknown") | (col("ad_creative_link_caption") != "Unknown")| (col("ad_creative_link_description") != "Un
```

```
[11]: data_cleaned_4=data_cleaned_4.withColumn('year', split(data['ad_creation_time'], '-').getItem(0)
                                    ).withColumn('month', split(data['ad_creation_time'], '-').getItem(1)
                                    ).withColumn('day', split(data['ad_creation_time'], '-').getItem(2)
                                    ).withColumn("spend_lower_bound", col("spend.lower_bound").cast("double")
                                    ).withColumn("spend_upper_bound", col("spend.upper_bound").cast("double")
                                    ).withColumn("impressions_lower_bound", col("impressions.lower_bound").cast("double")
                                    ).withColumn("impressions_upper_bound", col("impressions.upper_bound").cast("double")
                                    ).withColumn("ad_delivery_start_time", col("ad_delivery_start_time").cast("timestamp")
                                    ).withColumn("ad_delivery_stop_time", col("ad_delivery_stop_time").cast("timestamp")
                                    ).withColumn("demographic_distribution", col("demographic_distribution").cast("array<struct<age: string, gender: string, percentage
                                    ).withColumn("region_distribution", col("region_distribution").cast("array<struct<percentage: double, region: string>>")
                                    ).withColumn("year", col("year").cast("integer")
                                    ).withColumn("month", col("month").cast("integer")
                                    ).withColumn("day", col("day").cast("integer")
                                    ).withColumn("ad_creation_time", col("ad_creation_time").cast("timestamp"))
```

The cleaned data looks like

```
]: data_cleaned.show(10)

+---------------+------------------+--------------------+-------------------------+--------------------------+-----------------------------+------------------------------+---
------------------+--------------------+---------------------+-------------------+----------------+-------+--------+-----------------+------------------------+
----------------------+--------------------+-------------+---------+-----------+-----------------+--------------------+--------------------+------------------+-------
---+
|ad_creation_time|ad_creative_bodies|    ad_creative_body|ad_creative_link_caption|ad_creative_link_captions|ad_creative_link_description|ad_creative_link_descriptions|ad_
creative_link_title|ad_creative_link_titles|ad_delivery_start_time|ad_delivery_stop_time|     ad_snapshot_url|bylines|currency|delivery_by_region|demographic_distribution|
estimated_audience_size|       funding_entity|               id| impressions|languages|         page_id|          page_name|publisher_platforms| region_distribution|      sp
end|
+---------------+------------------+--------------------+-------------------------+--------------------------+-----------------------------+------------------------------+---
------------------+--------------------+---------------------+-------------------+----------------+-------+--------+-----------------+------------------------+
----------------------+--------------------+-------------+---------+-----------+-----------------+--------------------+--------------------+------------------+-------
---+
|      2021-11-21|              NULL|I'm ready to figh...|                        |                      NULL|                            |                          NULL|  C
ashless Pension ...|                  NULL|            2021-11-21|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|    [{55-64, unknown,...|
NULL|      Anika Wells MP| 966461410889234|{2000, 2999}|     NULL|2063329587276680|      Anika Wells MP|                NULL|   [{1, Queensland}]|{100, 199}|
|      2021-12-02|              NULL|Stay informed wit...|      nswliberal.org.au|                      NULL|       Sign up for our e...|                          NULL|
Stay Informed|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|   NSW Liberal Party| 214843134128684|    {0, 999}|     NULL|     44701328033|   NSW Liberal Party|                NULL|                NULL|   {0, 99}|
|      2021-12-02|              NULL|Stay informed by ...|      nswliberal.org.au|                      NULL|       Sign up for our e...|                          NULL|  U
pdate from our S...|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|   NSW Liberal Party| 445874000257357|    {0, 999}|     NULL|     44701328033|   NSW Liberal Party|                NULL|                NULL|   {0, 99}|
|      2021-12-02|              NULL|Stay informed wit...|      nswliberal.org.au|                      NULL|       Sign up for our e...|                          NULL|
Stay Informed|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|   NSW Liberal Party|2020401288140934|    {0, 999}|     NULL|     44701328033|   NSW Liberal Party|                NULL|                NULL|   {0, 99}|
|      2021-12-02|              NULL|Each week, receiv...|      nswliberal.org.au|                      NULL|       Sign up for our e...|                          NULL|
Stay Informed|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|   NSW Liberal Party| 278472954230096|    {0, 999}|     NULL|     44701328033|   NSW Liberal Party|                NULL|                NULL|   {0, 99}|
|      2021-12-02|              NULL|The Perrottet and...|      nswliberal.org.au|                      NULL|       Sign-up for our e...|                          NULL|  K
eeping you informed|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|   NSW Liberal Party| 597736024820393|    {0, 999}|     NULL|     44701328033|   NSW Liberal Party|                NULL|                NULL|   {0, 99}|
|      2021-12-02|              NULL|The Perrottet and...|      nswliberal.org.au|                      NULL|       Sign-up for our e...|                          NULL|  K
eeping you informed|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|   NSW Liberal Party| 854773098550102|    {0, 999}|     NULL|     44701328033|   NSW Liberal Party|                NULL|                NULL|   {0, 99}|
|      2021-12-02|              NULL|Each week, receiv...|      nswliberal.org.au|                      NULL|       {{product.descrip...|                          NULL|
Stay Informed|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|   NSW Liberal Party| 983391928923062|    {0, 999}|     NULL|     44701328033|   NSW Liberal Party|                NULL|                NULL|   {0, 99}|
|      2021-12-02|              NULL|Ward C team is ex...|    liberalhornsbyshi...|                      NULL|       Here is how to vo...|                          NULL|  H
ow to vote Liber...|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|    [{45-54, male, 0....|
NULL|Liberal Party - H...|4514506505298601|    {0, 999}|     NULL| 107271911712058|Liberal Party NSW...|                NULL|[{1, New South Wa...|   {0, 99}|
|      2021-12-02|              NULL|Your vote for Ali...|    eurobodallagreens...|                      NULL|       We will be connec...|                          NULL|  E
lections | eurob...|                  NULL|            2021-12-02|           2028-01-31|https://www.faceb...|   NULL|     AUD|              NULL|                    NULL|
NULL|     The Greens NSW|3084603238444857|    {0, 999}|     NULL| 107462368081783|Alison Worthingto...|                NULL|                NULL|   {0, 99}|
+---------------+------------------+--------------------+-------------------------+--------------------------+-----------------------------+------------------------------+---
------------------+--------------------+---------------------+-------------------+----------------+-------+--------+-----------------+------------------------+
----------------------+--------------------+-------------+---------+-----------+-----------------+--------------------+--------------------+------------------+-------
---+
only showing top 10 rows
```

Name : Siddhant Gaikwad
Student id : 47719848

DEDUPLICATION OF DATA

In this part, I had dropped all the duplicate values that were present in the dataset. Thus which helped in reducing the dimension of the data, which made the further analysis easier.

```
[9]: data_cleaned_4=data_cleaned.dropDuplicates()
```

DATA FILTERING

After performing exploratory data analysis (EDA), I emphasised on ads which correlated with specific subjects and political alignment. This was accomplished by applying appropriate filters depending on the text of the ad creative or other qualities.

Name : Siddhant Gaikwad
Student id : 47719848

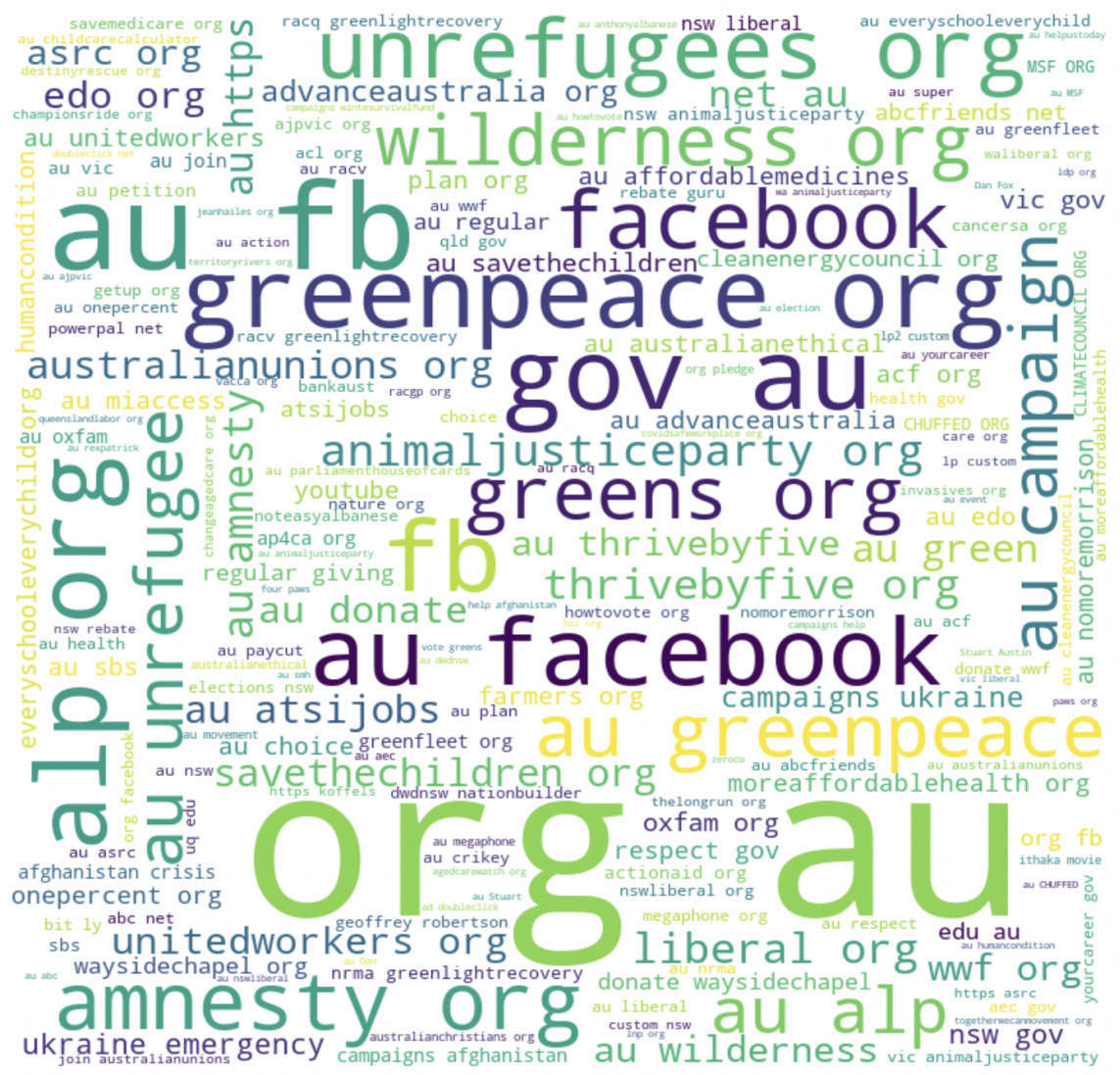
DATA ANALYSIS

- Here is the analysis of the word cloud of the name of the page from the data, highlighting the importance of the political parties, environmental and advocacy groups, in sharing the public debate.

The above word cloud mainly focuses on the Australian organisations. It is dominated by the Australian political parties, where the liberal, labor and One Nation are prominent parties. Environmental groups are also well represented, where Greenpeace, WWF Australia, and the Australian Conservation Foundation all appear

frequently. This reflects the growing importance of environmental issues in Australian politics, additionally advocacy groups are also prominent, such as Amnesty International etc.

- Now is the analysis of the word cloud for ad_creative_link_caption, it is mainly focused on encouraging users to take action on a range of social and political issues.



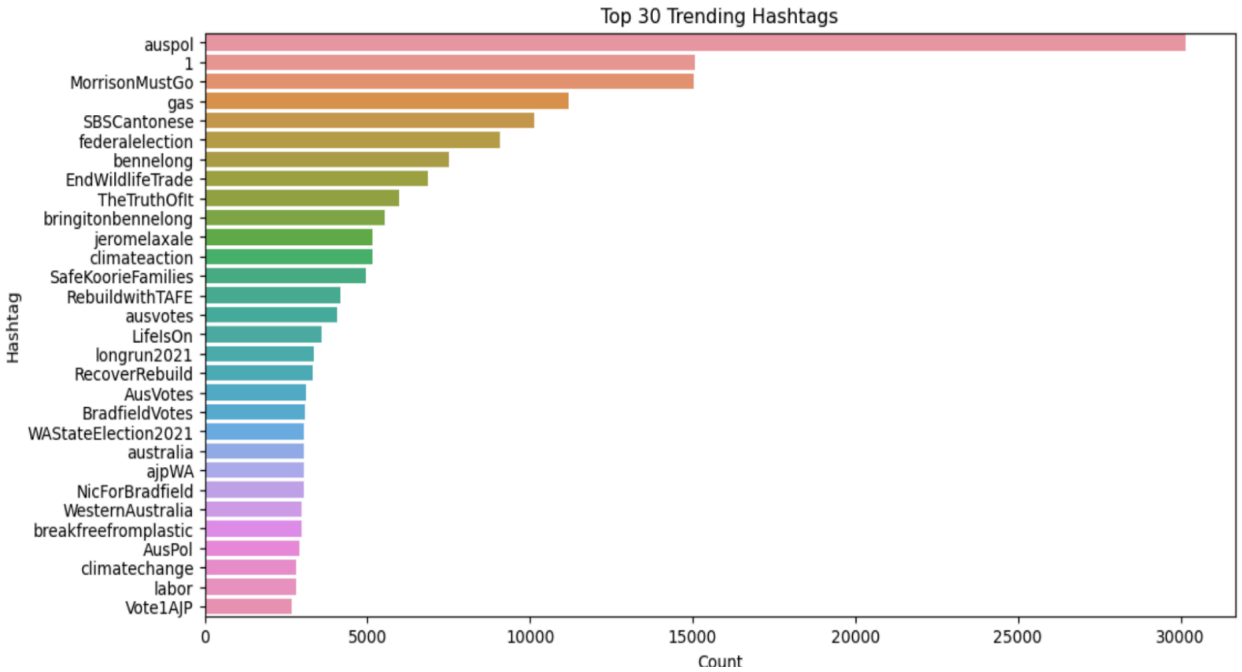
The ad cloud reveals a focus on activism. Calls to action like "donate" and "sign" urge users to engage with social issues like climate, health, and refugees. Words like "save," "protect," and "help" suggest emotional appeals to drive action. This reflects the range of advocacy efforts during this period.

Name : Siddhant Gaikwad
Student id : 47719848

- Now is the analysis of the word cloud for ad_creative_link_description, where the ads seem to be emphasising the positive outcomes that can be achieved and using clear calls to action to get viewers to engage.

The word cloud highlights key persuasive terms like "better," "free," "stronger," and "safer" to emphasise benefits of taking action. Common terms like "today," "learn more," "important," and "information" simplify ads. Policy-specific words like "climate action," "aged care," and "child care" target vital topics in the ads.
Now is the bar chart titled “ Top 30 Trending Hashtags”, trending hashtags provide some insights into the political and social issues that were being discussed on Facebook by Australian users.

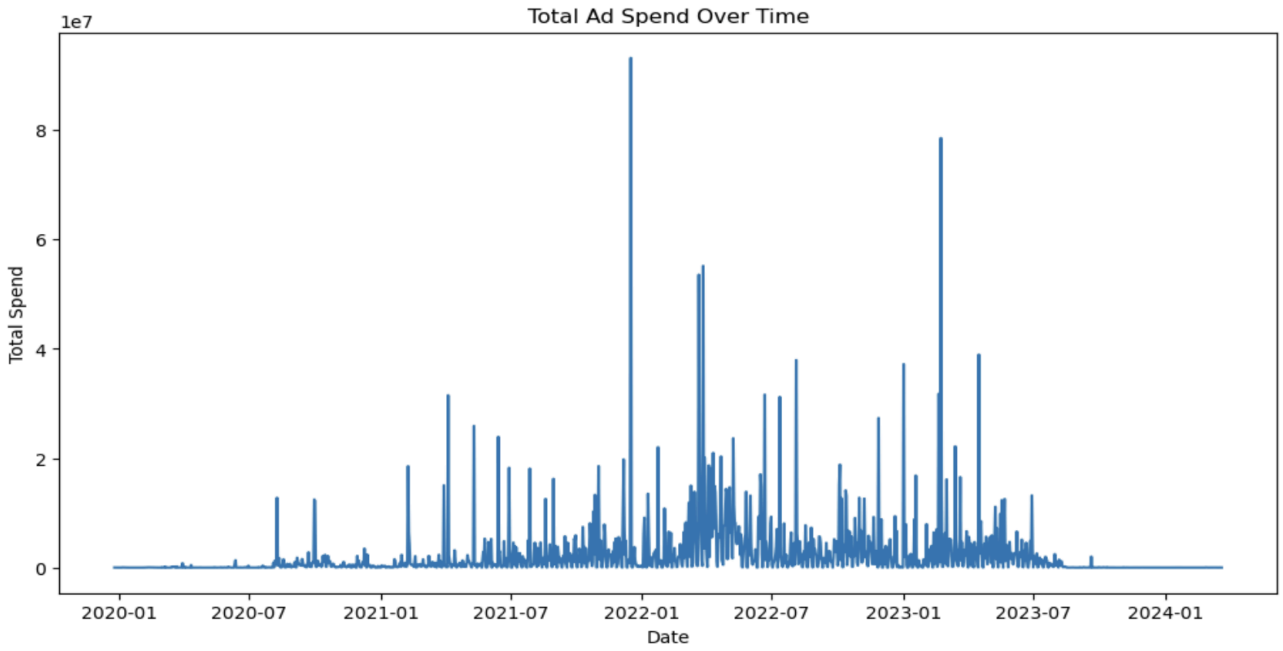
Name : Siddhant Gaikwad
Student id : 47719848



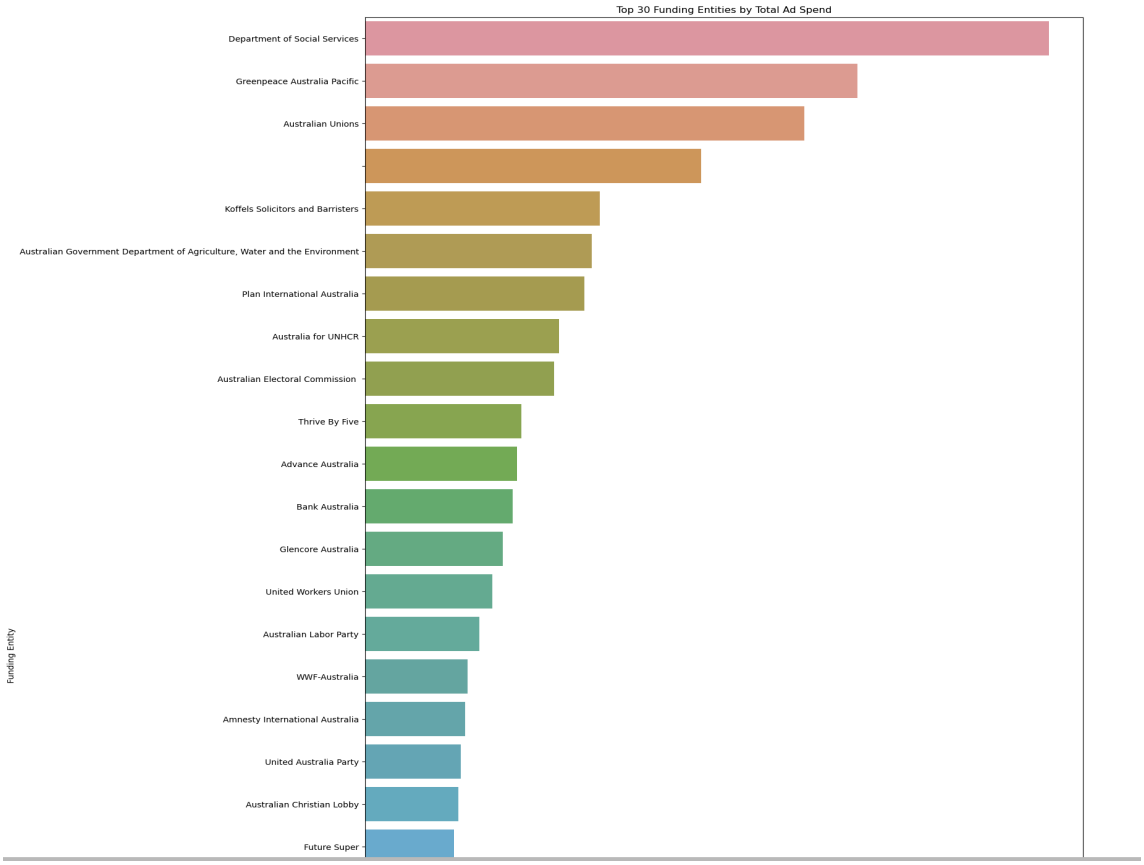
Many of the trending hashtags are political in nature, such as #auspol (#AustralianPolitics), #MorrisonMustGo (referencing the Prime Minister at the time, Scott Morrison), #federalelection (referencing the 2022 Australian federal election) and #nswelection.
Some of the trending hashtags relate to social issues, such as #climateaction and #achangeiscoming. Hashtags showing support for specific groups are also trending, such as #LNPAus (Liberal National Party of Australia), #VicLabor.

Now on the next page is the line plot for showing the “total ad spend over time”, the increase could have appeared due to a number of factors, such as the increasing importance of social media advertising in political campaigns. There are also some spikes in spending throughout the graph. These spikes likely correspond to major political events, such as the 2022 Australian federal election and the 2023 Voice referendum.

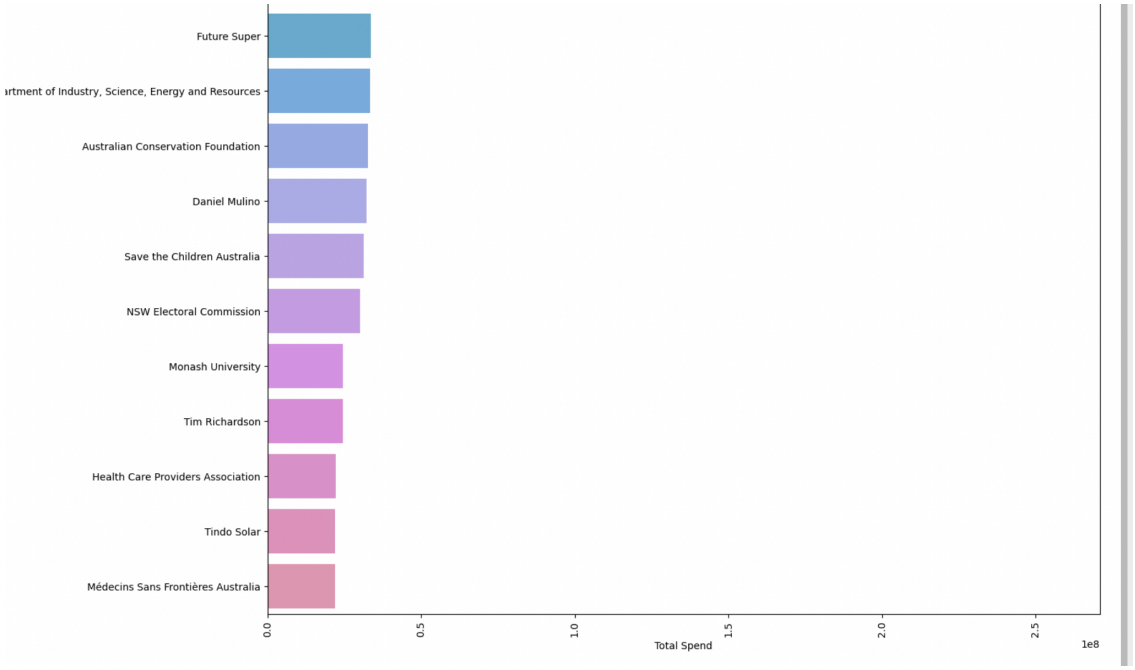
Name : Siddhant Gaikwad
Student id : 47719848



Now I will show the bar chart for funding_entity, where i will be concluding the " Top 30 Funding Entities by Total Ad Spend", or who the major spenders are in Facebook political advertising in Australia.

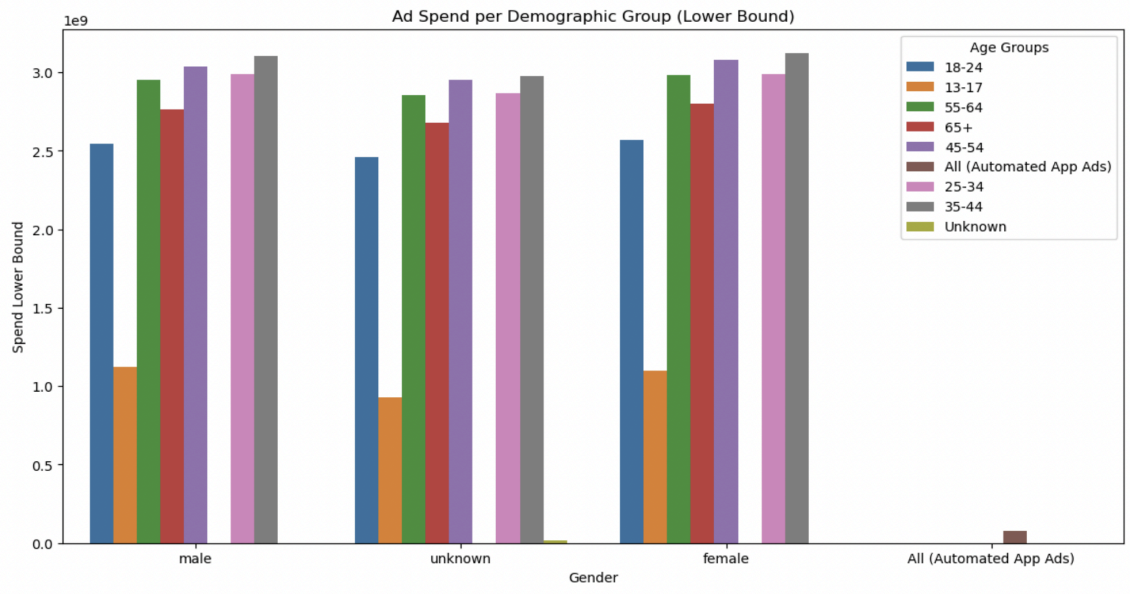


Name : Siddhant Gaikwad
Student id : 47719848



The top spenders include all of Australia's major political parties, including the Liberal Party, Australian Labour Party, Greens.
Interestingly, many Australian government departments are among the top spenders, including the Department of Social Services, the Department of Agriculture, Water and the Environment, and the Australian Electoral Commission. This shows that government agencies use Facebook to promote government programmes and services.

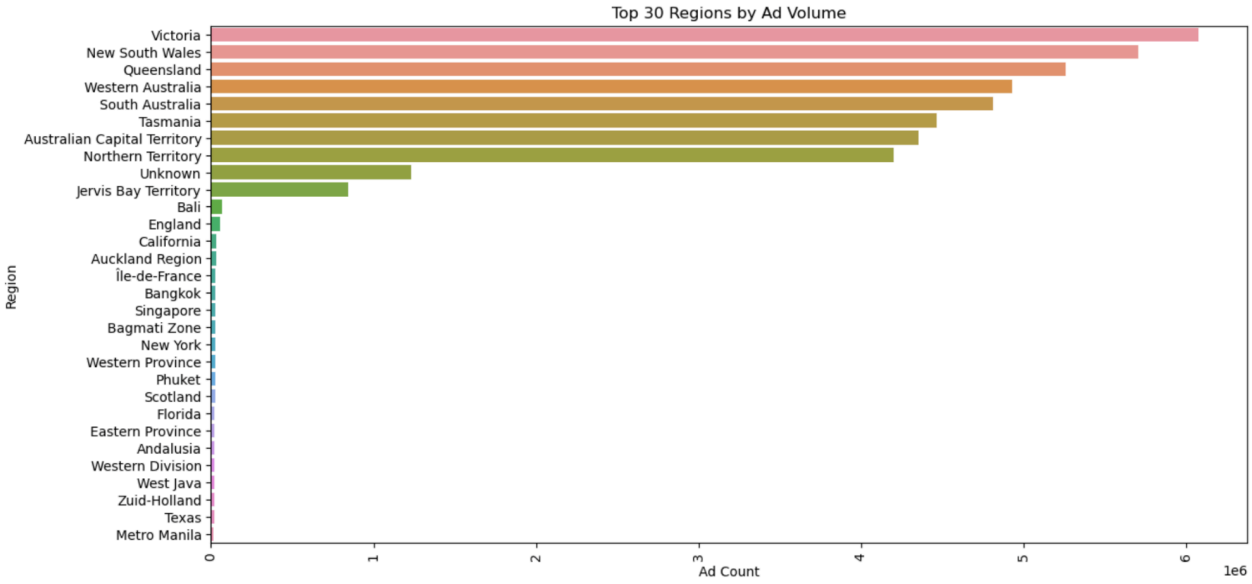
Now for concluding the Ad Spend per Demographic Group below :



Name : Siddhant Gaikwad
Student id : 47719848

The chart on the previous page suggests that more money is spent targeting males compared to females and unknowns. Advertisers might also believe that males are more likely to be interested in political advertising. Males may make up a larger proportion of the target audience for these political ads.

Now the bar plot for Top 30 Regions by Ad Volume below :



The chart above provides a good overview of the geographic distribution of ad spend in the dataset. Some of Australia's biggest towns, including Victoria, New South Wales, Queensland, and South Australia, are at the top of the list. This shows that advertisers are focusing their efforts on reaching people in these densely populated regions. Also there is some international reach, a list containing Bali, England, California, and New York, suggesting some worldwide influence. This could be due to a variety of things, including Australian political parties targeting overseas voters or advertisers that use Facebook's location settings incorrectly.

Name : Siddhant Gaikwad
Student id : 47719848

ANALYTICS METHODS

The analytical methods which were used for the process are :

1) Apache Spark : Utilised the potential of distributed computing with PySpark, which provides a high-level API for large data analytics.
2) Performing the DataFrame Operations : Used DataFrame transformations and actions to manipulation, group and filtering of the data.
3) Data Visualization : Leveraged the power of word cloud, seaborn, geopandas, matplotlib for good pictorial info.

Overall Discussions & Review:

After completing the analysis of the key finding are :

1. Political parties, environmental groups, advocacy groups are the main users of Facebook advertising in Australia.

2. The ads use a variety of calls to action, such as donate, sign, join, petition, and also cover a range of social topics like climate, health, and education.

3. The top trending hashtags are political in nature, such as #auspol, #MorrisonMustGo, #federalelection, and #nswelection.

4. More money is spent targeting males than females. This suggests that Facebook advertising is more effective for reaching males.

5. The top 30 regions by ad volume are all in Australia, however there is some foreign coverage as well. Facebook advertising targets both domestic and foreign audiences.

Conclusions :

The primary finding of the data analysis is that Facebook advertising serves as a crucial instrument for political parties, environmental groups, advocacy groups to effectively reach their audiences. The data indicates a significant increase in the utilisation of Facebook advertising, with the highest expenditures coming from prominent political parties and government departments. Finally, The results indicate that Facebook advertising is a highly efficient method for reaching an important demographic and shaping their opinions on matters of importance.

Name : Siddhant Gaikwad
Student id : 47719848

APPENDIX

```
pip install --upgrade pyspark
pip install wordcloud
pip install nltk
pip install spacy

import pyspark
print(pyspark.__version__)

import os
import warnings as w
w.filterwarnings('ignore')
from pyspark.sql import SparkSession, SQLContext
from pyspark.sql.types import *
from pyspark.sql.functions import *
import pyspark.sql.functions as F
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, explode, expr

# Create a SparkSession
spark = SparkSession.builder \
    .appName("FacebookAddataAnalysis") \
    .getOrCreate()

# Load the dataset from HDFS
file_path = "/data/ProjectDatasetFacebookAU/*"
data = spark.read.format("json").load(file_path)

data.printSchema()

data_cleaned = data.na.fill({"ad_creative_body": "","ad_creative_link_caption": "",
                    "ad_creative_link_title": "","ad_creative_link_description": "",
                    "funding_entity": "","page_name": "", "ad_delivery_stop_time":
"2028-01-31" })

from pyspark.sql import SparkSession, SQLContext
from pyspark.sql.types import *
from pyspark.sql.functions import avg, array_contains, to_date, mean, min, max, lit,
col, first, last, sum, count, countDistinct, desc, explode, split
pyspark.sql.DataFrameNaFunctions

data_cleaned_4=data_cleaned.dropDuplicates()
```

```
data_cleaned_4 = data_cleaned_4.filter((col("ad_creative_body") != "Unknown") |
(col("ad_creative_link_caption") != "Unknown")|
(col("ad_creative_link_description") != "Unknown") | (col("ad_creative_link_title")
!= "Unknown"))

data_cleaned_4=data_cleaned_4.withColumn('year', split(data['ad_creation_time'],
'-').getItem(0)
                                ).withColumn('month', split(data['ad_creation_time'],
'-').getItem(1)
                                ).withColumn('day', split(data['ad_creation_time'],
'-').getItem(2)
                                ).withColumn("spend_lower_bound",
col("spend.lower_bound").cast("double")
                                ).withColumn("spend_upper_bound",
col("spend.upper_bound").cast("double")
                                ).withColumn("impressions_lower_bound",
col("impressions.lower_bound").cast("double")
                                ).withColumn("impressions_upper_bound",
col("impressions.upper_bound").cast("double")
                                ).withColumn("ad_delivery_start_time",
col("ad_delivery_start_time").cast("timestamp")
                                ).withColumn("ad_delivery_stop_time",
col("ad_delivery_stop_time").cast("timestamp")
                                ).withColumn("demographic_distribution",
col("demographic_distribution").cast("array<struct<age: string, gender: string,
percentage: double>>")
                                ).withColumn("region_distribution",
col("region_distribution").cast("array<struct<percentage: double, region: string>>")
                                ).withColumn("year", col("year").cast("integer")
                                ).withColumn("month", col("month").cast("integer")
                                ).withColumn("day", col("day").cast("integer")
                                ).withColumn("ad_creation_time",
col("ad_creation_time").cast("timestamp"))

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('punkt')
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))
```

```
def preprocess_text(text):
    words = word_tokenize(text.lower())
    filtered_words = [word for word in words if word.isalnum() and word not in
stop_words and len(word) > 2]
    preprocessed_text = " ".join(filtered_words)

    return preprocessed_text

from pyspark.sql.functions import col
from wordcloud import WordCloud
import matplotlib.pyplot as plt

text_column = "page_name"

# Extract the text data from the DataFrame
text_data = data_cleaned_4.select(text_column).rdd.map(lambda row:
row[0]).filter(lambda x: x is not None)
text_combined = " ".join(text_data.collect())

wordcloud = WordCloud(width = 800, height = 800, background_color
='white').generate(text_combined)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()

text_column = "ad_creative_link_caption"

text_data = data_cleaned_4.select(text_column).rdd.map(lambda row:
row[0]).filter(lambda x: x is not None)

text_combined = " ".join(text_data.collect())

wordcloud = WordCloud(width = 800, height = 800, background_color
='white').generate(text_combined)

# Display the word cloud
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
```

```
plt.show()



region_distribution_pd = data_cleaned_4.withColumn("region",
explode("region_distribution.region")) \
                    .groupBy("region").count().orderBy("count", ascending=False) \
                    .toPandas().head(30)

plt.figure(figsize=(14, 7))
sns.barplot(data=region_distribution_pd, y='region', x='count')
plt.title('Top 30 Regions by Ad Volume')
plt.ylabel('Region')
plt.xlabel('Ad Count')
plt.xticks(rotation=90)
plt.show()
```