DATA7201 REPORT

Student name: Yuanhao Jiang

Student ID: 46548649

Abstract

With the popularization of mobile devices, we are generating massive amounts of data every day. Traditional storage technologies as well as analytics software can no longer meet the performance demands. Spark is a cutting-edge technology that addresses the three challenges of big data: velocity, volume and variety. The motivation of this analysis is mainly using spark to process and analyze the data from the 2020 US presidential election ad dataset on Facebook, and visualize it with tools such as Excel and Tableau. The contribution of this analysis is that I found some interesting phenomena, such as ads with low impressions are generally more cost-effective. But if the expensive ads are impressive, the more often they are placed the better the cost-effective. Campaign teams prefer to place ads on their campaign websites rather than mainstream media sites in order to control costs. In addition to strengthening the camp of your own party, gaining support from swing states is also quite important for the campaign.

Table of contents

Introduction

The general area of big data analytics.

The technology of big data analytics is widely used in many fields, such as healthcare, social media and finance. For example, Healthcare center can provide the personalized medical services for different group of patients via big data analytics. Doctors and nurses can collect real-time information from medical devices, then using big data analysis to provide more accurate treatment plan. Some social media platform could analyze user behaviors to give more useful recommendations.

Hadoop is the most popular technology in big data analytics which provides a distributed file system and a framework for the analysis and transtormation of massive data sets using MapReduce paradigm.

Motivation of the need for distributed system solutions.

With the development of technology, the massive amount of data needs to be processed and stored in time. How to process and store data is key to the problem. The traditional method is taking advantage of the centralized system to tackle the transaction. The centralized system has the advantage that it is easy to maintain and operate. But the drawback is not fault-tolerance and the limitation of performance. To solve this problem, the distributed system provides a solution. The distributed system could take a scale-out strategy to organize many standard computers and distribute massive data and computation over them. It also breaks free of the capacity and performance constraints that provide access to new and improved storage, RAM, CPU and other hardware.

For example, the data of advertising API on Facebook about US 2020 presidential election is about hundreds of MB in size in a month. Traditional data storage and analysis tools, such as Excel, which is unable to handle such scalable data. Some computational engines are designed to solve the problem. Such as Spark, which is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters. Spark can be 100 times faster than Hadoop for large scale data processing by exploiting memory computing.

Dataset analysis

The dataset is a collection of sponsored political posts on Facebook targeted at US users during 23 months from 03/2020 to 01/2022. This includes the period preceding the latest US Presidential election in November 2020. The dataset contains 16 features and the format is JSON. Because the size of the dataset is too large, which is stored in HDFS. In order to analyze more efficiently, we need to do pre-processing for the data first.

Step 1: Pre-process

- Only selected data for 2020 because the U.S. presidential election ends on December 4th, 2020.
- Converts the range of spending to an exact number (the average of the range). It is to simplify the calculation for later analysis.
- Converts the range of impression to an exact number (the average of the range). It is to simplify the calculation for later analysis.

Step 2: General Analysis

Let us start with a general analysis of the entire data set, followed by a specific analysis of the campaigns of the two popular presidential candidates, Biden and Trump.

It is interesting to see which countries funded the U.S. election. We can view which countries explicitly influenced the U.S. elections by the currency in which they paid for the ads. Select the currency, count the currency and rank the count by descending. From the figure [1] below is shown the top 20 countries that are influential in the process of election.

The EU is most concerned about the US election, followed by India, Australia, Vietnam and the UK. The campaign can keep an eye on whether the funding entities behind these countries have interfered in the U.S. election. U.S. elections are an internal U.S. affair, and if a campaign team joins with these foreign forces to interfere in the election, other teams can leverage this against the campaign’s impure motives.

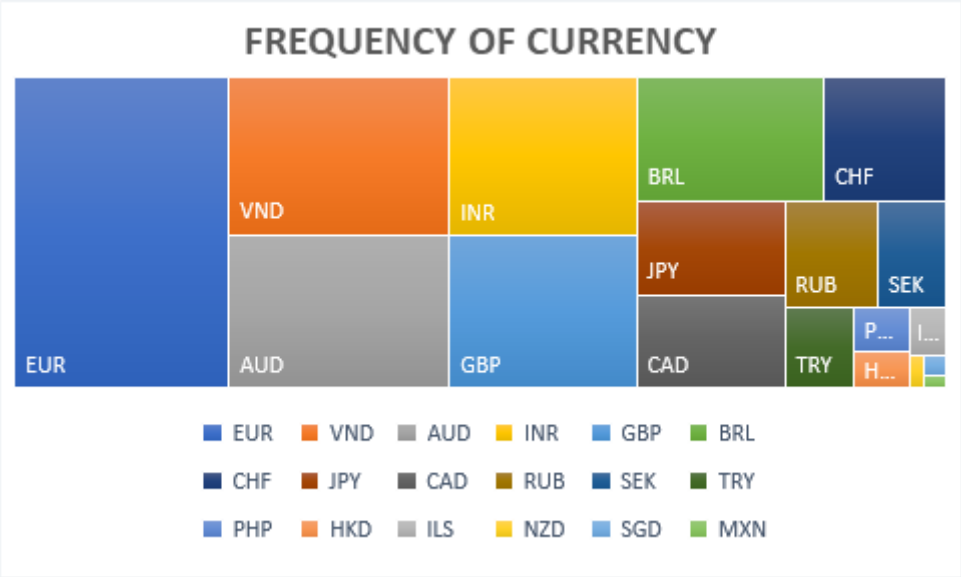

Figure 1 Frequency of appearance of different currencies in funding ads

Then, we might be curious about which funding entities sponsor advertising most frequently. Select funding entities, and count them, rank the count by descending. The top 10 funding entities are shown below [2]. These funding entities have the most ads on Facebook, and the campaign team can focus on them if the president wants to be re-elected or if other candidates want to compete.

Figure 2 The funding entities which sponsor Ad most frequently

Step 3: Detailed Analysis

Next, I would analyze the two most popular presidential candidates in detail, Biden and Trump. Because the names of some funding entities indicate who they support. To distinguish between the Trump and Biden campaigns by the names of the funding entities and extracted the two datasets. For Trump's team, these are the keywords I've chosen: " TRUMP MAKE AMERICA GREAT AGAIN COMMITTEE ", " TRUMP FOR PRESIDENT "and "America First Action". For Biden's team, these are the keywords I've chosen: "Democratic Governors Association", "Biden Victory Fund", "BIDEN FOR PRESIDENT" and "Independence USA PAC".

It is interesting to know how much did Biden and Trump's team spend on a single ad. Select mean spending and count them, then rank the count by descending for both datasets. Using Excel to aggregate the mean spend by under 1000 $ or above 10000 $. As you can see from the chart below [3], Trump's and Biden's teams' investment strategies for advertising are significantly different. The Trump team invested twice as much as the Biden team in ads costing less than $1,000, but the Biden team invested more than three times as much as the Trump team in ads costing more than $10,000. Trump's team prefers to invest in ads that cost less, while Biden's team prefers to invest in ads that cost more.

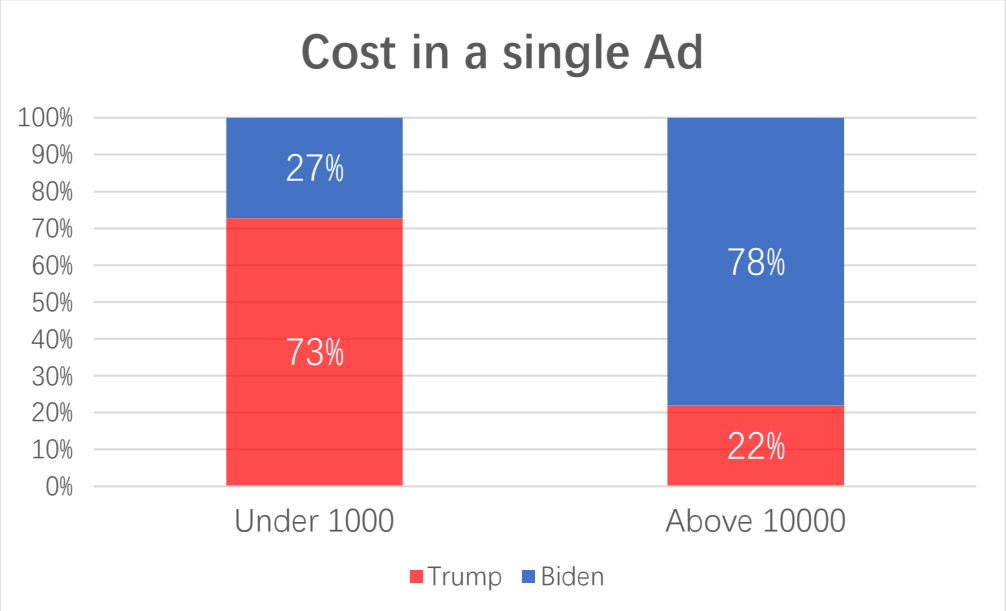

Figure 3 Cost in a single Ad

And then we might be curious about which Internet domain do Biden's and Trump's teams each like to place ads on and how much has Biden and Trump's team invested in these sites. Select mean spend and the URL of Internet domain, group by the Internet domain and sum the mean spend for both datasets. From the graph below [4], we could see Trump and Biden's team mainly spend money on ads for Internet domains with their own names, rather than on some mainstream media websites like Google and Facebook.

It is possible that the price of ads in mainstream media is costly. They choose the cost-effective way to focus on their own Internet domain.

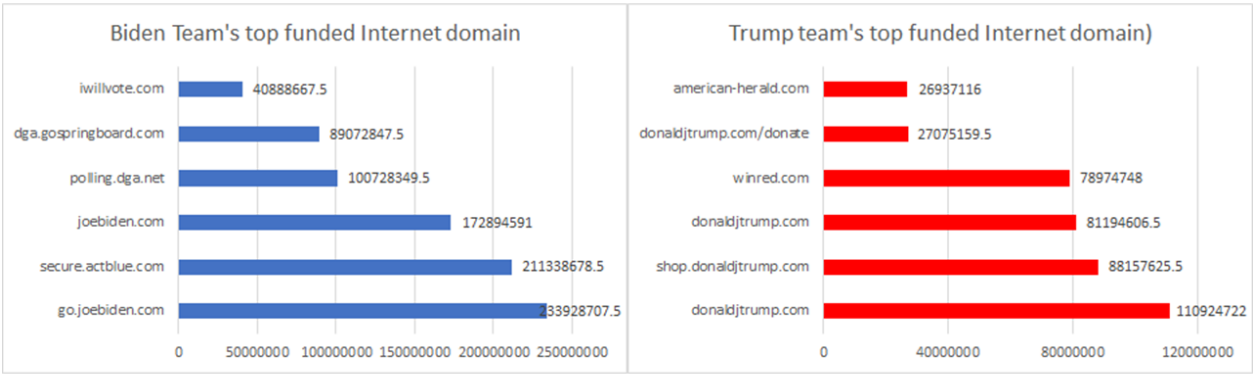

Figure 4 Trump's and Biden's team

In addition to the Internet domain, perhaps you would be interested in what is Trump and Biden's team spending on ads in each state? Split the feature of region distribution into two features percentage and region respectively. Then I convert the data format of the spend to integer and the data format of the percentage to floating point. The following visualizations are made by Tableau. The blue part indicates the Biden team and the red part represents the Trump team.

From the graph below [5][6], we can see that both Biden and Trump's teams are spending a lot of money to get votes in Florida and Pennsylvania. Because these two areas are "swing states". The Biden team mainly funded ads in Michigan, California and New York, and those states also supported Biden. The Trump team largely funded ads in Texas and North Carolina, and those states that endorsed Trump as well.

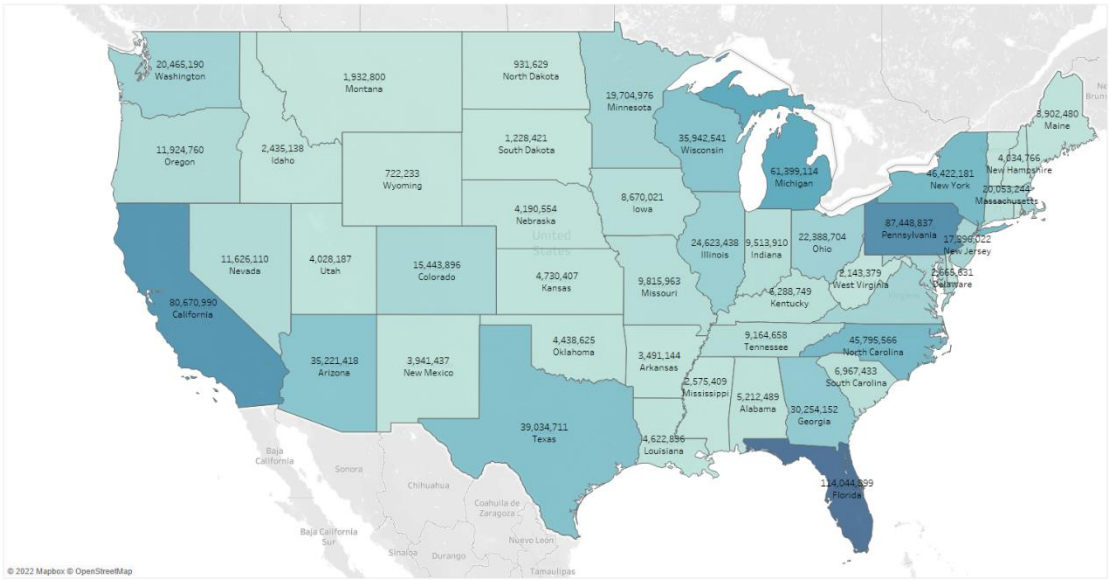

Figure 5 The distribution map for Biden team's ad funding

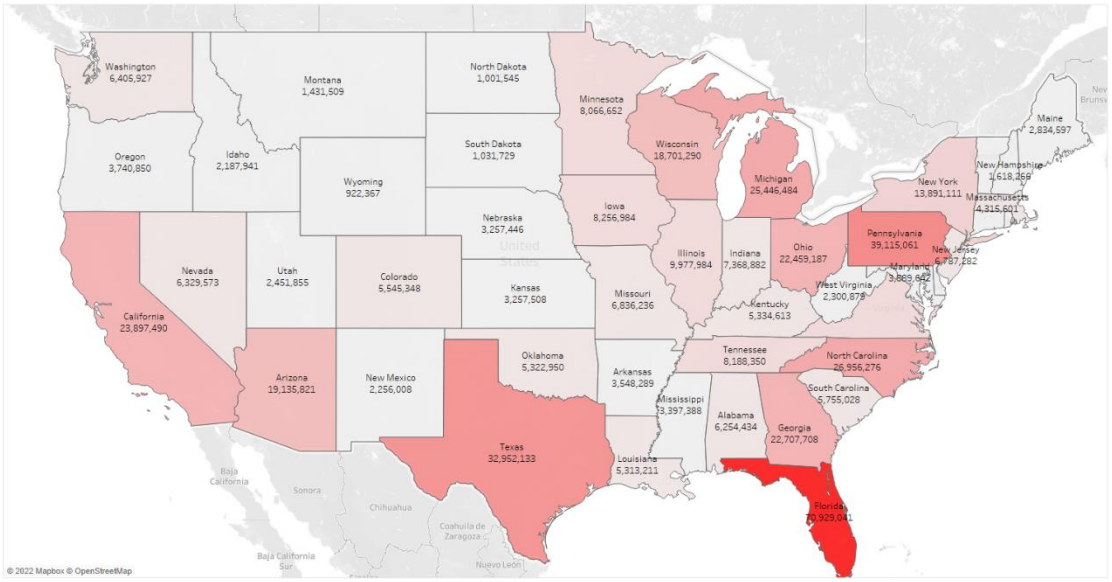

Figure 6 The distribution map for Trump team's ad funding

We already know that Biden and Trump's team are spending a lot on advertising. But you may be doubting that whether their investment in these ads match the impact the ads have created, and whether it is true that the more they spend on ads, the more impact they create. Group by the feature of Impression and sum value of mean spend, then ranking the Impression by ascending. I calculate the ratio of impression to spend to observe the cost effectiveness of their fund in advertising. From the chart below [7] we can see that low impact ads are generally more cost-effective, and the Trump team prefers to place such ads. Through the previous analysis, we know that Biden's team

put in more expensive ads than Trump's team, and these expensive ads tend to be very influential. This is why the Biden team's ads have been very cost effective. It also shows that as long as the expensive ads are impressive, the more frequent they are placed, the more cost effective they are.

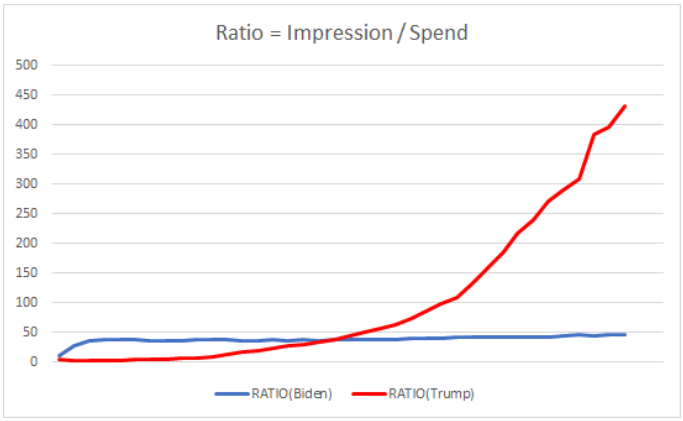

Figure 7 The ratio of impression and spend of advertisement

Apart from concerning Biden and Trump's funding in advertising, you may be interested in their campaign slogans, such as which campaign slogan does Biden and Trump's team favor respectively and which link description do Biden's and Trump's teams favor respectively.

Select the feature of ad body and link description, then count those sentences and calculate word frequency for these texts. Then I use python to generate the word cloud. From the graph below [8], we can see that the names of both candidates and the parties behind them appear most frequently in the slogans and link descriptions of both sides. Most of the words are appealing words, such as stop, will, now, need, and other words with strong emotion.

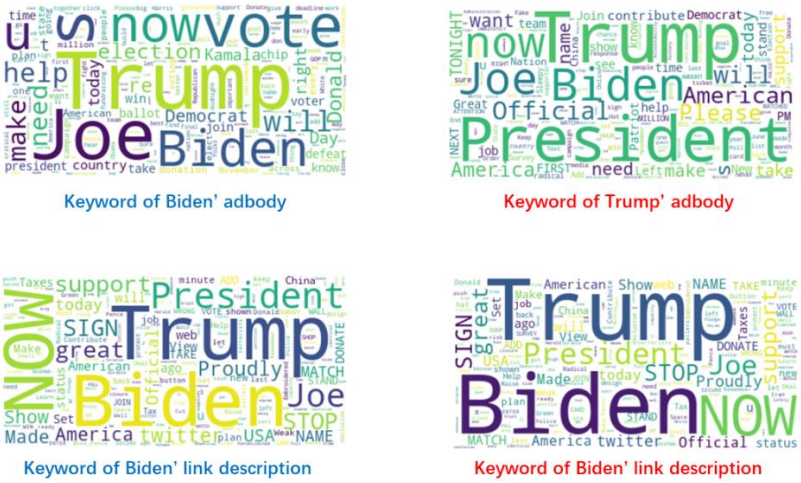

Figure 8 Word cloud for the link description and ad body of Trump's and Biden's team

Discussion and conclusion

Discussion

With the popularization of mobile devices, we are generating massive amounts of data in the social media every day. Traditional storage technologies as well as analytics software can no longer meet the performance demands. I use the advanced analytic software Spark to break the bottleneck of bid data. Spark is a cutting-edge technology that addresses the three challenges of big data: velocity, volume and variety.

The motivation of this analysis is mainly using spark to process and analyze the data from the 2020 US presidential election ad dataset on Facebook, and visualize it with tools such as Excel and Tableau.

Main finding

Overall, we could see the U.S. presidential election is being watched by forces outside the country, especially in Europe. Different campaigns have different strategies for placing ads. However, the two popular candidates have a similar style of advertising slogans, both with very straightforward language. The high-frequency words are mostly appealing words except for the candidates' names. Generally, ads with low impressions are more cost-effective. But if the expensive ads are impressive, the more often they are placed the better the cost-effective. Campaign teams prefer to place ads on their own campaign websites rather than mainstream media sites in order to control costs. In addition to strengthening the camp of your own party, gaining support from swing states is also quite important for the campaign.

Take-away message

Through these conclusions, stakeholders can observe whether there are foreign forces interfering in the election, and exposing this relationship can effectively challenge the motivation of other campaign teams which is not pure. Besides, swing states would be key of election. Stakeholders can focus on those states that funding more ads on those states. If stakeholders have sufficient budget, it is better to consider funding more in impressive ads, even if these ads are costly. Because the more you invest, the more cost-effective it is.

Appendix

```
import pyspark
from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession
from pyspark.sql import SQLContext, DataFrame
from pyspark.sql.functions import expr,upper,col,explode,udf,desc, mean,sum
from pyspark.sql.types import DoubleType
from pyspark.sql.types import IntegerType

spark = SparkSession \
  .builder \
  .appName("s4654864") \
  .config("spark.some.config.option", "some-value") \
  .getOrCreate()

# Load the data from HDFS
# Only select the period of 2020, because the president election end with 2020-12-4
df = spark.read.json("/data/ProjectDatasetFacebook/FBads-US-2020*")

# Observe which country fund the US election according to currency
# Select the currency == USD to align the spend range weight.
df.groupBy("currency").count().sort('count',ascending=False).toPandas().to_csv("currency.csv")
df=df.filter(df["currency"]=='USD')
df=df.drop("currency")

# Convert the spend range into a certain number(the average of spend.)
df = df.withColumn("lower_bound",col("spend.lower_bound").cast("Integer"))\
.withColumn("upper_bound",col("spend.upper_bound").cast("Integer"))\
.withColumn("mean_spend",col("upper_bound")+col("lower_bound"))\
.withColumn("mean_spend",col("mean_spend")*0.5)
df=df.drop("upper_bound","lower_bound","spend")

# Convert the impression range into a certain number(the average of impression.)
df = df.withColumn("lower_bound",col("impressions.lower_bound").cast("Integer"))\
.withColumn("upper_bound",col("impressions.upper_bound").cast("Integer"))\
.withColumn("mean_impressions",col("upper_bound")+col("lower_bound"))\
.withColumn("mean_impressions",col("mean_impressions")*0.5)
df=df.drop("upper_bound","lower_bound")
df=df.drop("impressions")

# Which funding entities sponsor advertising most frequently?
df.groupBy("funding_entity").count().sort('count',ascending=False).toPandas().to_csv('funding
```

```
_entity.csv')

# Filter out the funding_entity of the Trump team
T_team=df.filter(col("funding_entity").contains("TRUMP FOR PRESIDENT")|
                 col("funding_entity").contains("TRUMP MAKE AMERICA GREAT AGAIN COMMITTEE")|
                 col("funding_entity").contains("America First Action")).distinct()

# Filter out the funding_entity of the Biden team
B_team=df.filter(col("funding_entity").contains("BIDEN FOR PRESIDENT")|
                 col("funding_entity").contains("BIDEN VICTORY FUND")|
                 col("funding_entity").contains("Independence USA PAC")|
                 col("funding_entity").contains("Democratic Governors Association")).distinct()

# How much did Biden and Trump's team spend on a single ad?
T_spendrange_singlead=T_team.groupBy("mean_spend").count().sort('count',ascending=False)
T_spendrange_singlead.toPandas().to_csv('T_spendrange_singlead.csv')

B_spendrange_singlead=B_team.groupBy("mean_spend").count().sort('count',ascending=False)
B_spendrange_singlead.toPandas().to_csv('B_spendrange_singlead.csv')

# How much has Biden and Trump's team invested in these sites?
T_domain_spend=T_team.select("mean_spend","ad_creative_link_caption")
T_domain_spend=T_domain_spend.groupBy("ad_creative_link_caption").sum('mean_spend').sort("sum(mean_spend)",ascending=False).toPandas().to_csv('T_domain_spend.csv')

B_domain_spend=B_team.select("mean_spend","ad_creative_link_caption")
B_domain_spend=B_domain_spend.groupBy("ad_creative_link_caption").sum('mean_spend').sort("sum(mean_spend)",ascending=False).toPandas().to_csv('B_domain_spend.csv')

# Which Internet domain do Biden's and Trump's teams each like to place ads on?
T_team.groupBy("ad_creative_link_caption").count().sort('count',ascending=False).toPandas().to_csv('T_url.csv')

B_team.groupBy("ad_creative_link_caption").count().sort('count',ascending=False).toPandas().to_csv('B_url.csv')

# Which campaign slogan does Biden and Trump's team favor?
T_team.groupBy("ad_creative_body").count().sort('count',ascending=False).toPandas().to_csv('T_adbody.csv')
```

```
B_team.groupBy("ad_creative_body").count().sort('count',ascending=False).toPandas().to_csv
('B_adbody.csv')

# Which link description do Biden's and Trump's teams favor respectively?
T_link_description=T_team.groupBy("ad_creative_link_description").count().sort('count',ascen
ding=False)
T_link_description=T_link_description.filter(T_link_description["ad_creative_link_description"].i
sNotNull())
T_link_description=T_link_description.filter(T_link_description["ad_creative_link_description"] !
= "{{product.description}}")
T_link_description.toPandas().to_csv("T_link_description.csv")

B_link_description=B_team.groupBy("ad_creative_link_description").count().sort('count',ascen
ding=False)
B_link_description=B_link_description.filter(B_link_description["ad_creative_link_description"].i
sNotNull())
B_link_description=B_link_description.filter(B_link_description["ad_creative_link_description"]
!= "{{product.description}}")
B_link_description.toPandas().to_csv("B_link_description.csv")

# Biden and Trump's team are spending a lot on advertising.
# Does the investment in these ads match the impact the ads have created?
# Is it true that the more they spend on ads, the more impact they create?
T_impression=T_team.select("mean_impressions","mean_spend")
T_impression=T_impression.groupBy("mean_impressions").mean('mean_spend')
T_impression=T_impression.sort("mean_impressions",ascending=False).toPandas().to_csv("T_
impression.csv")

B_impression=B_team.select("mean_impressions","mean_spend")
B_impression=B_impression.groupBy("mean_impressions").mean('mean_spend')
B_impression=B_impression.sort("mean_impressions",ascending=False).toPandas().to_csv("B
_impression.csv")

# Split region_distribution, automatically generate percentage and region
B_region_spend = B_team.select("mean_spend",expr("inline(region_distribution)"))
B_region_spend=B_region_spend.select("percentage","region","mean_spend")

#Convert spend into int format
B_region_spend=                    B_region_spend.withColumn("spend_int",
B_region_spend["mean_spend"].cast(IntegerType()))
#Convert percentage to float format
B_region_spend=B_region_spend.withColumn("per_float",B_region_spend["percentage"].cast
("float"))
B_region_spend=B_region_spend.withColumn("single_cost",B_region_spend["per_float"]*B_r
```

```
egion_spend["spend_int"])

#Group by region, calculate the advertising investment in each region
B_region_spend=B_region_spend.groupby("region").agg({"single_cost":"sum"}).withColumnR
enamed("sum(single_cost)","total_cost")

B_region_spend.toPandas().to_csv("B_region_spend.csv")

# Split region_distribution into percentage and region
T_region_spend = T_team.select("mean_spend",expr("inline(region_distribution)"))
T_region_spend=T_region_spend.select("percentage","region","mean_spend")

# Convert mean spend into int type
T_region_spend=                                    T_region_spend.withColumn("spend_int",
T_region_spend["mean_spend"].cast(IntegerType()))
#Convert percentage to float type
T_region_spend=T_region_spend.withColumn("per_float",T_region_spend["percentage"].cast
("float"))
T_region_spend=T_region_spend.withColumn("single_cost",T_region_spend["per_float"]*T_re
gion_spend["spend_int"])

#Group by region, calculate the advertising investment in each region
T_region_spend=T_region_spend.groupby("region").agg({"single_cost":"sum"}).withColumnR
enamed("sum(single_cost)","total_cost")

T_region_spend.toPandas().to_csv("T_region_spend.csv")




! pip install wordcloud
import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

from wordcloud import WordCloud

#Importing Dataset

df = df1

#Checking the Data
```

```
df.head()

#Creating the text variable

text2 = " ".join(title for title in df.iloc[:,-1])

# Creating word_cloud with text as argument in .generate() method

word_cloud2 = WordCloud(collocations = False, background_color = 'white').generate(text2)

# Display the generated Word Cloud

plt.imshow(word_cloud2, interpolation='bilinear')

plt.axis("off")

plt.savefig("B_adbody.png", dpi=300)
plt.show()


df = pd.read_csv("T_link_description.csv")
df.head()

df1 = pd.read_csv("B_link_description.csv")
df1[['count', 'ad_creative_link_description']] = df1[['ad_creative_link_description', 'count']]
df1=df1.iloc[ : ,1:3]
df1=df1.dropna()
df1


import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

from wordcloud import WordCloud

#Importing Dataset

df = df1

#Checking the Data
```

```
df.head()

#Creating the text variable

text2 = " ".join(title for title in df.iloc[:,-1])

# Creating word_cloud with text as argument in .generate() method

word_cloud2 = WordCloud(collocations = False, background_color = 'white').generate(text2)

# Display the generated Word Cloud

plt.imshow(word_cloud2, interpolation='bilinear')

plt.axis("off")
plt.savefig("B_link description.png", dpi=300)
plt.show()
```