Module Title: Programming in Python for Data Science\ Module Code: CS2PP22\ Lecturer responsible: Dr Todd Jones\ Individual / Group Assignment: Individual\ Weighting of the Assignment: 100%\ Hours spent for this assignment: 20 hours\ Student Number: 29000557

Table of Contents

CS2PP22 Programming in Python for Data Science

Assessment Task 2: Twitter Data Analysis

Scenario:

You have been asked by a client to analyse information from the social media platform, Twitter. On Twitter, users can compose "tweets" (messages) of up to 280-characters in length that will be shared in real-time to friends and followers on the platform. Millions of tweets are shared by millions of users every day. The client has an interest in developing a **regression-based predictive model** based on tweet characteristics to determine how many **likes** (a.k.a "favourites") a specific future tweet will ultimately receive.

Basic Requirements:

- **Extract** a dataset of tweets.
- Perform a simple **exploratory data analysis**.
- **Manipulate** the data into a form suitable for regression analysis.
  - This might involve some data cleaning.
- **Save** this data to the `./data/Task2/` space.
  - It is sensible to save one file per user.
- Design and implement a **regression analysis model** based on one of the `sklearn` methods described in the module.
- **Evaluate** your model's performance.
- **Test** your model on a tweet that your model has not seen.
  - You can do this with a train/test split of the collected data, collect new data for testing, or develop an alternative protocol.
- Provide a **written report** that follows your steps along the way, including:
  - justification of your selected analyses;
  - analysis of the findings;
  - descriptions of and reasoning behind your workflow design, implementation, decisions, and assumptions.

Considerations:

- You will need to extract **at least 300** tweets (perhaps, the 300 most recent tweets) from **at least 3** Twitter accounts. That’s **at least 900** tweets, in total. Typically, though, a larger dataset will yield more robust results.

- **Critical:**
  - Tweet extraction will rely on your choice of extraction package.
  - To extract Tweets, you will need to `pip` -install helper packages, such as:
    - `tweepy` : https://docs.tweepy.org/en/stable/
    - `twint` : https://pypi.org/project/twint/
    - `json` : https://docs.python.org/3/library/json.html
  - To extract Tweets with `tweepy` , you **MUST** have a Twitter developer account to access the API.
    - Further details about these requirements are available in this notebook, below, and in the assessment description: `CS2PP22_Assessment.pdf`

- You are free to determine the predictive features of your model. Be sure to select the appropriate **kinds** of data for this type of analysis. You might consider:
  - the number of mentions in a tweet
  - the number of followers or friends of anyone mentioned
  - the number of hashtags
  - the number of emojis
  - the number of words/characters

  - the time of day, week, or year
  - the sentiment of the words
    - this would require the use of an additional package, like `TextBlob` : https://textblob.readthedocs.io/en/dev/
      - `sentiment.subjectivity`
      - `sentiment.polarity`

- How does the distribution of observed likes appear?
  - Might you need to **transform** or **scale** the likes to conform to the assumptions of the model?

- How should you choose users for analysis?
  - You might want to think about how frequently a user is active on the platform. Less active accounts will likely provide the same number of tweets over a much longer time period than a more active account, and comparing these would mean comparing very different periods in time.

- How should tweets be selected for analysis?
  - It typically takes some amount of time for a tweets number of likes to plateau. You might wish to ignore more recent tweets and focus on those that have "matured."

- How should the data be visualised in the various parts of this Task?

- **Report format:**
  - Use **markdown headings** to denote report sections and subsections.
  - Use the notebook **markdown cells** to describe work done in the code cells and their outputs (e.g., properly labeled figures). Markdown entries should describe features of (at most, a few) immediately preceding cells and any imported routines.
  - Use concise, descriptive language to fully explain your methods and results.
  - Refer to the **Assessment** section on Blackboard to review resources for support and guidance on clear communication.
  - All code should be accompanied by concise, descriptive **comments**.

- **Exceptional reports** will include deeper analysis. For instance, you might consider comparing models trained for individual users versus a multi-user training set and testing such models on other users. You might also investigate other `sklearn` regression techniques and analyse differences in their behaviours and performance. Exceptional reports will also likely include novel model feature selection.

- Completion will require that you write Python code to perform each of the sub-tasks of the Task2 work.
  - Try to follow the PEP 8 – Style Guide for Python Code: https://peps.python.org/pep-0008
  - Function and variable annotations are not required.

- Some parts of this assignment may require further self-study of Python documentations or other resources.

- You may also refer to other documentation/self-study resources, such as those suggested in the Lecture Notes or a multitude of other resources that you have independently discovered.

Items to be submitted:

1. A modified version of this Jupyter notebook file (`.ipynb`)
   - This should be fully executed in a serial fashion, from top to bottom.
   - Try **Kernel** --> **Restart & Run All** to verify that this works as intended.
   - Add a cell at the top of the notebook to note the installation method and version of any additional Python packages not included in the module's Anaconda distribution or that was not instructed to be installed during the module.

1. Files containing the **Twitter data** used in your analysis. These should be:
   - stored under `./data/Task2/` in your archived submission;
   - and separated by your selected Twitter users.

1. A copy of this notebook (**with the `CS2PP22_Assessment_Task2` label** as in Item 1.) but in `.pdf` format, which displays all content independently. This can be included in the overall assessment archive, but an unarchived copy should be submitted to Blackboard alongside the archive.

Marking scheme:

Marks	Item
10	**Organisation:** Preparation and submission of all required files
10	**2.1:** Extraction of tweet datasets
20	**2.2:** Exploratory data analysis
10	**2.3:** Data processing
20	**2.4:** Regression analysis
10	**2.5:** Model evaluation and testing
20	**Overall:** Report structure and reasoning (format, clarity, logic, quality of written communication)

Tweepy Example

Installed as:

```
pip install tweepy
```

Remember to check your version and use the corresponding documentation!

E.g.: https://docs.tweepy.org/en/v4.10.1/api.html shows how the package methods correspond to those of the Twitter API.

In [1]:

```
# API configuration and test


import tweepy
tweepy.__version__
```

Out[1]:

```
'4.12.1'
```

In [2]:

```
# Set Twitter keys and tokens
#  --  Enter your values here  --
API_key = 'Pf5aWLXbe2awYJNzolLS7PcxI'
API_secret_key = 'ytzkjVLDVmR1RYutqOnn9L5NdAGrKpuYXedca1bFLQMPRPQEk4'
bearer_token =  'AAAAAAAAAAAAAAAAAAAAOCYlQEAAAAATDVE4e0E2pjrLIlxkg8dtEhKEoE%3DWwOkO4Co6RvLBj3xDCdQKLX8mvU3nnp5hy
access_token = '1289223601678364675-pGQDJlbvCHd4DJmZ3x3Zies9VroZ5b'
access_token_secret = 'GNjhPIxKJswD4DGAyMWlbUxcriHfPs6L1lpr8iJymOeCo'

# Configure authentication and initialise API instance
auth = tweepy.OAuthHandler(API_key, API_secret_key)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# Test the API
user = api.get_user(screen_name='twitter')
print(user.screen_name)
print(user.followers_count)

# Expect output similar to:
#     Twitter
#     64959605
```

```
Twitter
65689545
```

In [3]:

```
# Example of tweet extraction and inspection of result

# If using this method, double check the resulting number of entries, as they
#     may differ from what is requested for a variety of reasons.
# For one, this method is limited to 200 status retrievals.
# Also, because suspended or deleted content is removed after the count
#     has been applied.

# tweet_mode='extended': yields `full_text`, rather than a truncated version.

# The resulting set of statuses can be further parsed and transformed into
# other formats.

# Because of the 200-status limit, you would need to iterate over this
# extraction method and keep track of the `max_id` parameter.  This refers to
# a numerical ID that is sequential in time.

tweets = api.user_timeline(screen_name='twitter',
                           count=20, tweet_mode='extended')
print(len(tweets))
tweets[0]._json  # present tweet in JSON format
```

```
17
```

Out[3]:

```
{'created_at': 'Wed Feb 08 20:00:46 +0000 2023',
 'id': 1623411536243965954,
 'id_str': '1623411536243965954',
 'full_text': 'more words more words more words more words more words more words more words more words more words
more words more words more words more words more words more words more words more words more words more words mor
e words more words more words more words more words more words more… https://t.co/0mcFJ1wwZK',
 'truncated': False,
 'display_text_range': [0, 304],
 'entities': {'hashtags': [],
  'symbols': [],
  'user_mentions': [],
 'urls': [{'url': 'https://t.co/0mcFJ1wwZK',
   'expanded_url': 'https://twitter.com/i/web/status/1623411536243965954',
   'display_url': 'twitter.com/i/web/status/1…',
   'indices': [281, 304]}]},
'source': '<a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a>',
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'in_reply_to_screen_name': None,
'user': {'id': 783214,
 'id_str': '783214',
 'name': 'Twitter',
 'screen_name': 'Twitter',
 'location': 'everywhere',
 'description': "What's happening?!",
 'url': 'https://t.co/DAtOo6uuHk',
 'entities': {'url': {'urls': [{'url': 'https://t.co/DAtOo6uuHk',
     'expanded_url': 'https://about.twitter.com/',
     'display_url': 'about.twitter.com',
     'indices': [0, 23]}]},
  'description': {'urls': []}},
 'protected': False,
 'followers_count': 65689545,
 'friends_count': 5,
 'listed_count': 88082,
 'created_at': 'Tue Feb 20 14:35:54 +0000 2007',
 'favourites_count': 6149,
 'utc_offset': None,
 'time_zone': None,
 'geo_enabled': True,
 'verified': True,
 'statuses_count': 15046,
 'lang': None,
 'contributors_enabled': False,
 'is_translator': False,
 'is_translation_enabled': False,
 'profile_background_color': 'ACDED6',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme18/bg.gif',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme18/bg.gif',
 'profile_background_tile': True,
 'profile_image_url': 'http://pbs.twimg.com/profile_images/1488548719062654976/u6qfBBkF_normal.jpg',
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1488548719062654976/u6qfBBkF_normal.jpg',
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/783214/1646075315',
 'profile_link_color': '1B95E0',
 'profile_sidebar_border_color': 'FFFFFF',
 'profile_sidebar_fill_color': 'F6F6F6',
 'profile_text_color': '333333',
 'profile_use_background_image': True,
 'has_extended_profile': True,
 'default_profile': False,
 'default_profile_image': False,
 'following': False,
 'follow_request_sent': False,
 'notifications': False,
 'translator_type': 'regular',
 'withheld_in_countries': []},
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 10521,
'favorite_count': 106562,
'favorited': False,
'retweeted': False,
'possibly_sensitive': False,
'lang': 'en'}
```

In [4]:

```
# Example of tweet extraction and manipulation to form a DataFrame

# A DataFrame can easily be written to a file.

import pandas as pd
import tweepy

API_key = 'Pf5aWLXbe2awYJNzolLS7PcxI'
API_secret_key = 'ytzkjVLDVmR1RYutqOnn9L5NdAGrKpuYXedca1bFLQMPRPQEk4'
bearer_token =  'AAAAAAAAAAAAAAAAAAAAAOCYlQEAAAAATDVE4e0E2pjrLIlxkg8dtEhKEoE%3DWwOkO4Co6RvLBj3xDCdQKLX8mvU3nnp5hy
```

```
access_token = '1289223601678364675-pGQDJlbvCHd4DJmZ3x3Zies9VroZ5b'
access_token_secret = 'GNjhPIxKJswD4DGAyMWlbUxcriHfPs6L1lpr8iJymOeCo'


# Configure authentication and initialise API instance
auth = tweepy.OAuthHandler(API_key, API_secret_key)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# Test the API
user = api.get_user(screen_name='twitter')
print(user.screen_name)
print(user.followers_count)


def get_user_timeline(account_name, number_of_tweets, include_retweets):
    created_at = []
    tweet = []
    favorite_count = []
    retweet_count = []
    source = []
    is_quote_status = []
    favorited = []
    mentions = []
    hashtags = []

    for status in tweepy.Cursor(api.user_timeline, screen_name=account_name,
                                include_rts=include_retweets, tweet_mode='extended').items(number_of_tweets):
        created_at.append(status.created_at)
        tweet.append(status.full_text)
        favorite_count.append(status.favorite_count)
        retweet_count.append(status.retweet_count)
        is_quote_status.append(status.is_quote_status)
        favorited.append(status.favorited)

    timeline_df = pd.DataFrame({'created_at': created_at,
                                'tweet': tweet,
                                'likes': favorite_count,
                                'retweet_count': retweet_count,
                                'is_quote_status': is_quote_status,
                                'favorited': favorited, })
    return timeline_df


get_user_timeline('twitter', 10, True)
```

```
Twitter
65689546
```

Out[4]:

	created_at	tweet	likes	retweet_count	is_quote_status	favorited
0	2023-02-08 20:00:46+00:00	more words more words more words more words mo...	106535	10521	False	False
1	2022-12-10 21:38:10+00:00	thanks for your patience as we've worked to ma...	5991	624	False	False
2	2022-12-10 21:38:10+00:00	subscribers will be able to change their handl...	6221	784	False	False
3	2022-12-10 21:38:09+00:00	we'll begin replacing that “official” label wi...	4375	622	False	False
4	2022-12-10 21:38:09+00:00	when you subscribe you'll get Edit Tweet, 1080...	12829	1258	False	False
5	2022-12-10 21:38:08+00:00	we're relaunching @TwitterBlue on Monday – sub...	28987	5262	False	False
6	2022-10-13 21:41:45+00:00	@ElenbaasHier	1832	22	False	False
7	2022-10-13 21:41:17+00:00	@kufesteezz does this help	682	14	False	False
8	2022-10-13 21:41:04+00:00	@MasonCollects only have one	133	4	False	False
9	2022-10-13 21:40:47+00:00	@abdulsabooh789 not happening	278	0	False	False

Twint Example

`twint` allows for scraping Tweets from Twitter profiles without using Twitter's API.

https://github.com/twintproject/twint/wiki

The `twint` documentation suggests several paths to installation. It was recently shown on a University machine that the following worked best:

```
pip3 install --user --upgrade -e
git+https://github.com/twintproject/twint.git@origin/master#egg=twint
```

NOTE: `nest_asyncio` is needed to execute `twint` in a Jupyter notebook. Remember to import this and `apply` it in your code, as shown below.

Review the documentation to access the data you are looking for (e.g., parameters such as: `Retweets`, `Profile_full` and `twint.run.Profile`). Example: https://analyticsindiamag.com/complete-tutorial-on-twint-twitter-scraping-without-twitters-api/

In [5]:

```
# Prerequisite packages and setup
#import twint
#import nest_asyncio
#nest_asyncio.apply()


# Configure twint instance
#c = twint.Config()

#c.Username = "noneprivacy"
#c.Limit = 20                  # Increments of 20
#c.Pandas = True               # Enable Pandas processing

# Run the search
#twint.run.Search(c)

# Store results in DataFrame
#twint_df = twint.storage.panda.Tweets_df
```

In [6]:

```
# Inspect the resulting DataFrame
#twint_df
```

Introduction

In this report we will aim to extract tweets from multiple accounts using the tweepy package and apply various techniques of data processing and exploration to ultimately build and evaluate a regression model.

In [7]:

```
!pip install tweepy
!pip install emoji --upgrade
```

```
Requirement already satisfied: tweepy in c:\programdata\anaconda3\lib\site-packages (4.12.1)
Requirement already satisfied: requests-oauthlib<2,>=1.2.0 in c:\programdata\anaconda3\lib\site-packages (from tw
eepy) (1.3.1)
Requirement already satisfied: requests<3,>=2.27.0 in c:\programdata\anaconda3\lib\site-packages (from tweepy) (2
.28.2)
Requirement already satisfied: oauthlib<4,>=3.2.0 in c:\programdata\anaconda3\lib\site-packages (from tweepy) (3.
2.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from reque
sts<3,>=2.27.0->tweepy) (3.0.1)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.27
.0->tweepy) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests
<3,>=2.27.0->tweepy) (1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests<3,
>=2.27.0->tweepy) (2020.12.5)
Requirement already satisfied: emoji in c:\programdata\anaconda3\lib\site-packages (2.2.0)
```

In [8]:

```
import pandas as pd
import tweepy

API_key = 'Pf5aWLXbe2awYJNzolLS7PcxI'
API_secret_key = 'ytzkjVLDVmR1RYutqOnn9L5NdAGrKpuYXedca1bFLQMPRPQEk4'
bearer_token =  'AAAAAAAAAAAAAAAAAAAAAOCYlQEAAAAATDVE4e0E2pjrLIlxkg8dtEhKEoE%3DWwOkO4Co6RvLBj3xDCdQKLX8mvU3nnp5hy
access_token = '1289223601678364675-pGQDJlbvCHd4DJmZ3x3Zies9VroZ5b'
access_token_secret = 'GNjhPIxKJswD4DGAyMWlbUxcriHfPs6L1lpr8iJymOeCo'

# Configure authentication and initialise API instance
auth = tweepy.OAuthHandler(API_key, API_secret_key)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# Test the API
user = api.get_user(screen_name='twitter')
print(user.screen_name)
print(user.followers_count)


def get_user_timeline(account_name, number_of_tweets, include_retweets):
    created_at = []
    tweet = []
    favorite_count = []
    retweet_count = []
    source = []
    is_quote_status = []
    favorited = []
    mentions = []
    hashtags = []

    for status in tweepy.Cursor(api.user_timeline, screen_name=account_name,
                                include_rts=include_retweets, tweet_mode='extended').items(number_of_tweets):
        created_at.append(status.created_at)
        tweet.append(status.full_text)
        favorite_count.append(status.favorite_count)
        retweet_count.append(status.retweet_count)
        is_quote_status.append(status.is_quote_status)
        favorited.append(status.favorited)

    timeline_df = pd.DataFrame({'created_at': created_at,
                                'tweet': tweet,
                                'favorite_count': favorite_count,
                                'retweet_count': retweet_count,
                                'is_quote_status': is_quote_status,
                                'favorited': favorited, })
    return timeline_df


get_user_timeline('twitter', 10, True)
```

```
Twitter
65689550
```

Out[8]:

	created_at	tweet	favorite_count	retweet_count	is_quote_status	favorited
0	2023-02-08 20:00:46+00:00	more words more words more words more words mo...	106535	10521	False	False
1	2022-12-10 21:38:10+00:00	thanks for your patience as we've worked to ma...	5991	624	False	False
2	2022-12-10 21:38:10+00:00	subscribers will be able to change their handl...	6221	784	False	False
3	2022-12-10 21:38:09+00:00	we'll begin replacing that “official” label wi...	4375	622	False	False
4	2022-12-10 21:38:09+00:00	when you subscribe you'll get Edit Tweet, 1080...	12829	1258	False	False
5	2022-12-10 21:38:08+00:00	we're relaunching @TwitterBlue on Monday – sub...	28987	5262	False	False
6	2022-10-13 21:41:45+00:00	@ElenbaasHier	1832	22	False	False
7	2022-10-13 21:41:17+00:00	@kufesteezz does this help	682	14	False	False
8	2022-10-13 21:41:04+00:00	@MasonCollects only have one	133	4	False	False
9	2022-10-13 21:40:47+00:00	@abdulsabooh789 not happening	278	0	False	False

In [9]:

```
import pandas as pd

elonmusk_df = get_user_timeline('elonmusk',500, True)
```

In the cell above I have extracted 500 tweets from Elon Musk's twitter account and saved them to a dataframe named elonmusk_df. I will repeat the same process for other twitter accounts in the following steps. You can see the resulting dataframes in the output of the cells below.

In [10]:

```
import pandas as pd

print(elonmusk_df)
elonmusk_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\elonmusk.csv",index =False)
```

	created_at	tweet	favorite_count	retweet_count	is_quote_status	favorited
0	2023-03-14 06:35:42+00:00	@MuskUniversity I hope so	6775	397	False	False
1	2023-03-14 05:17:16+00:00	Fight for truth, whole truth &amp; nothin but!...	40851	5222	True	False
2	2023-03-14 05:10:55+00:00	@TrungTPhan	2668	134	False	False
3	2023-03-14 04:56:16+00:00	@SawyerMerritt That media report is false. Rel...	9931	806	False	False
4	2023-03-14 04:31:13+00:00	@_LOVELYSPAIN_ Reminds me of an Overwatch map	17275	611	False	False
...	...	...	...	...	...	...
495	2023-02-24 03:08:33+00:00	@DavidSacks A Russia-China alliance is inevita...	41894	4465	False	False
496	2023-02-23 22:05:41+00:00	@PeterDiamandis Yeah	13319	477	False	False
497	2023-02-23 21:54:38+00:00	@lopatonok @UnderSecStateP Interesting thread	5387	659	False	False
498	2023-02-23 20:44:25+00:00	@alx While there is relative good &amp; bad, t...	45388	3719	False	False

499	2023-02-23 20:40:52+00:00	@growing_daniel	4741	194	False	False

500 rows × 6 columns

In [11]:

```python
import pandas as pd

lebron_df = get_user_timeline('KingJames',500, True)
print(lebron_df)
lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames.csv")
```

	created_at	tweet	favorite_count	retweet_count	is_quote_status	favorited
0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-09 22:26:18+00:00	RT @fox8news: The LeBron James Family Foundati...	0	368	False	False
4	2023-03-09 16:07:03+00:00	RT @SLAMKicks: What started out in the digital...	0	130	False	False
...	...	...	...	...	...	...
495	2022-05-09 01:06:19+00:00	OMFG!!!!!!!! https://t.co/7cAyX6KuXs	37518	2737	True	False
496	2022-05-08 16:19:51+00:00	Where y'all finding all this content lately. M...	34928	2450	True	False
497	2022-05-08 01:54:31+00:00	Yessir!!!! Went yard then hit the "Silencer"!!...	51669	4687	True	False
498	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
499	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

500 rows × 6 columns

In [12]:

```python
import pandas as pd

RishiSunak_df = get_user_timeline('RishiSunak',500, True)
print(RishiSunak_df)
RishiSunak_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\rishi_sunak.csv")
```

	created_at	tweet	favorite_count	retweet_count	is_quote_status	favorited
0	2023-03-14 10:02:07+00:00	I've always asked people to judge me by my act...	779	138	False	False
1	2023-03-14 09:33:47+00:00	RT @GregHands: Our 5 @Conservatives priorities...	0	53	False	False
2	2023-03-13 23:30:21+00:00	Protecting our people. \n\nDefending our value...	5489	557	False	False
3	2023-03-13 21:04:57+00:00	NEW: The first generation of #AUKUS nuclear-po...	4516	728	False	False
4	2023-03-13 17:24:20+00:00	RT @jensstoltenberg: I spoke w/ PM @RishiSunak...	0	167	False	False
...	...	...	...	...	...	...
495	2022-07-29 07:49:24+00:00	2/ We will create a new rule of ‘three strikes...	122	19	False	False
496	2022-07-29 07:49:22+00:00	1/ I will double the number of foreign nationa...	535	89	False	False
497	2022-07-29 07:10:43+00:00	RT @Ready4Rishi: Many criminals convicted of s...	0	61	False	False
498	2022-07-28 19:35:36+00:00	When my grandparents emigrated here they emigr...	461	96	False	False
499	2022-07-28 19:29:36+00:00	We must have a system of control for our borde...	350	77	False	False

500 rows × 6 columns

In [13]:

```python
import pandas as pd

eminem_df = get_user_timeline('Eminem',500, True)
print(eminem_df)
eminem_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\eminem.csv")
```

	created_at	tweet	favorite_count	retweet_count	is_quote_status	favorited
0	2023-02-23 22:26:09+00:00	RT @ShadyRecords: “Fuck droppin' a jewel, I'm ...	0	1886	False	False
1	2023-02-19 00:29:45+00:00	You better show some respect whenever the doc'...	62468	5238	False	False
2	2023-02-18 00:28:44+00:00	Thank u to all those that can, please suppor...	11263	1754	False	False
3	2023-02-09	"Y’all know what time it is, soon as 50 signs ...	28378	2829	True	False

	03:18:38+00:00					
4	2023-01-15 22:24:26+00:00	Sending out best wishes for a quick recovery f...	18251	1590	False	False
...	...	...	...	...	...	...
495	2018-04-20 04:14:00+00:00	Motown in the building. #selfie https://t.co/...	30371	3064	False	False
496	2018-04-17 16:50:08+00:00	#COACHƎLLA HIT THE SITE FOR WEEKEND 1 GALLERY ...	20478	2965	False	False
497	2018-04-16 02:57:11+00:00	Me, Al &amp; Denaun gettin ready for Coachella...	32779	3131	False	False
498	2018-04-15 02:32:19+00:00	.@COACHELLA STANS - CATCH US ACROSS FROM THE D...	10664	1301	False	False
499	2018-04-15 00:31:59+00:00	Selfie https://t.co/L3U2KV6aVZ	56178	7004	False	False

500 rows × 6 columns

Firstly, we used the API service of Twitter which includes the tokens to extract tweets of various famous people. To do that we imported the pandas and tweepy packages. Then, we used the cursor function built in tweepy to search through twitter to return our data. We used the get_user_timeline function to save the data as a panda dataframe and assigned a name to that dataframe. Each of the elements in the dataframe include certain attributes we have determined such as; the date of the tweet, the number of retweets, the number of likes, if it has been favourited etc. We then saved the dataframes produced in a specific location in the computer. To get the best possible analysis, various people from different backgrounds such as art, politics, business and sport was selected. Furthermore, to increase the accuracy of our analysis and to increase our sample frame, we have extracted 500 tweets per person.

Some of the things I learnt are, how to extract tweets and, how to utilize various developer tools the API comes with.

Explatory Data Analysis

In [17]:

```
import pandas as pd

eminem_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\eminem.csv")
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames.csv")
RishiSunak_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\rishi_sunak.csv")
elonmusk_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\elonmusk.csv")


dataframes = {'eminem' : eminem_df, 'lebron' : lebron_df, 'Rishi Sunak' : RishiSunak_df,'elon': elonmusk_df}
# loop through the data frames and print the total likes and retweets
for name, df in dataframes.items():
    total_likes = df['favorite_count'].sum()
    total_retweets = df['retweet_count'].sum()
    print(f"From the tweets we collected, {name} has {total_likes} likes and {total_retweets} retweets.")
```

```
From the tweets we collected, eminem has 16812825 likes and 2761418 retweets.
From the tweets we collected, lebron has 16354041 likes and 1608645 retweets.
From the tweets we collected, Rishi Sunak has 4000621 likes and 402751 retweets.
From the tweets we collected, elon has 20832806 likes and 2127024 retweets.
```

To find out which one of the accounts is the most popular, I have written the code above that prints the total number of retweets and likes for each influencer from the tweets we extracted. The output tells us Elon is the most liked, but Eminem has the most retweets.

In [3]:

```
import pandas as pd

#This is code to see how many retweets and likes these people have got to assess their popularity.

eminem_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\eminem.csv")
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames.csv")
RishiSunak_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\rishi_sunak.csv")
elonmusk_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\elonmusk.csv")


dataframes = {'eminem' : eminem_df, 'lebron' : lebron_df, 'Rishi Sunak' : RishiSunak_df,'elon': elonmusk_df}

# Create an empty list to store the total tweet and like counts
totals_list = []

# Looping through all of the dataframes found in the dictionary.
for key, df in dataframes.items():
    # the column created_at is converted to datetime
    df['created_at'] = pd.to_datetime(df['created_at'])
    # only the tweets in 2022
    tweets_2022 = df[df['created_at'].dt.year == 2022]
    # Calculate total for tweets and total likes in 2022
    total_tweets = len(tweets_2022)
    total_likes = tweets_2022['favorite_count'].sum()
    # Add them to a list and update the list as you iterate.
    totals_list.append({'dataframe': key, '2022_tweet_sum': total_tweets, '2022_like_sum': total_likes})

# Create a new dataframe with the totals
totals_df = pd.DataFrame(totals_list)

print(totals_df)
```

	dataframe	2022_tweet_sum	2022_like_sum
0	eminem	134	2729812
1	lebron	423	12038576
2	Rishi Sunak	324	3138755
3	elon	0	0

To understand which influencer is the most active I have written code in the cell above that prints us the total number of tweets and the total number of likes for 2022 per each influencer. The results were suprising as I was expecting Elon to have at least some likes from 2022 in our dataset. In order to analyze further, I seperated the tweets per influencer for the years 2022 and 2023.

In [46]:

```
#Code to see how often our users are on social media by checking how many times they tweeted in the past year.

# a dictionary to save dataframes so we can loop through them:
dataframes = {'eminem' : eminem_df, 'lebron' : lebron_df, 'Rishi Sunak' : RishiSunak_df, 'elon': elonmusk_df}

tweet_2022= {}

for name, df in dataframes.items():
    df['date'] = pd.DatetimeIndex(df['created_at']).year
    total_2022 = df[df['date'] == 2022]['tweet'].count()
    #print(f"{name} has tweeted: {total_2022} times last year \n")
    tweet_2022[name] = total_2022

tweet2022_df = pd.DataFrame(tweet_2022.items(), columns=['Name', '2022_tweets'])

print(tweet2022_df)
```

	Name	2022_tweets
0	eminem	134
1	lebron	423
2	Rishi Sunak	324
3	elon	0

At first glance it might seem like the users with the highest tweets are on social media the most, however, that is not necessarily correct. Since we begin collecting data from the most recent date, it is likely that all 500 of the tweets by Elon that we extracted were on 2023, which means we can assume he is the most active on social media. Our findings also suggest that Eminem is the second most popular since he has the second lowest tweets in the past year. However, it should be noted that the majority of Eminem's tweets might be from before 2022 as such, it is too early to come to a conclusion yet.

In [36]:

```
dataframes = {'eminem' : eminem_df, 'lebron' : lebron_df, 'Rishi Sunak' : RishiSunak_df, 'elon': elonmusk_df}

tweet2023 ={}

for name, df in dataframes.items():
    df['date'] = pd.DatetimeIndex(df['created_at']).year
    total_2023 = df[df['date'] == 2023]['tweet'].count()
   # print(f"{name} has tweeted: {total_2023} times this year \n")
    tweet2023[name] = total_2023

tweet2023_df = pd.DataFrame(tweet2023.items(), columns=['Name', '2023_tweets'])

print(tweet2023_df)
```

	Name	2023_tweets
0	eminem	5
1	lebron	77
2		176

	Rishi Sunak	
3	elon	500

Our assumption about Elon being the most active is confirmed, but unlike our previous finding Eminem is not the second most active after Elon. In fact, this year he has been the least active.

In [33]:

```
lebron_df.isnull().sum() #To show if there are null values present.
```

Out[33]:

```
Unnamed: 0         0
created_at         0
tweet              0
favorite_count     0
retweet_count      0
is_quote_status    0
favorited          0
dtype: int64
```

In [35]:

```
RishiSunak_df.isnull().sum()
```

Out[35]:

```
Unnamed: 0         0
created_at         0
tweet              0
favorite_count     0
retweet_count      0
is_quote_status    0
favorited          0
dtype: int64
```

In [34]:

```
elonmusk_df.isnull().sum()
```

Out[34]:

```
created_at         0
tweet              0
favorite_count     0
retweet_count      0
is_quote_status    0
favorited          0
dtype: int64
```

In [32]:

```
eminem_df.isnull().sum()
```

Out[32]:

```
Unnamed: 0         0
created_at         0
tweet              0
favorite_count     0
retweet_count      0
is_quote_status    0
favorited          0
dtype: int64
```

No null values found in any of the dataframes as shown.

In [17]:

```
import emoji
```

In [37]:

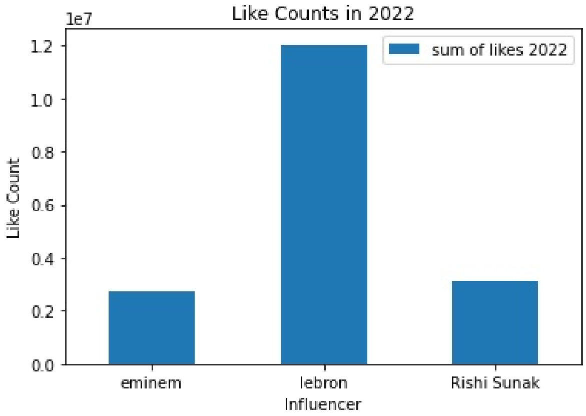
```
import pandas as pd
import emoji

#This code loops through a dictionary of dataframes and prints out how many emojies they have used in our data sa

dataframes = {'eminem' : eminem_df, 'lebron' : lebron_df, 'Rishi Sunak' : RishiSunak_df, 'elon': elonmusk_df}

total_emoji = {}   #initialize a dictionary

for name, df in dataframes.items(): #loop through dataframes
    def count_emojis(tweet):
        return len([char for char in tweet if char in emoji.EMOJI_DATA])
    emoji_sum = df['tweet'].apply(count_emojis).sum()
    total_emoji[name] = emoji_sum

emoji_df = pd.DataFrame(total_emoji.items(), columns=['Name', 'Emoji Count']) #save it into a new dataframe



print(emoji_df) #display the new dataframe
```

	Name	Emoji Count
0	eminem	266
1	lebron	1702
2	Rishi Sunak	223
3	elon	144

We can see that politicians and businessman are less likely to use emojis in their tweets, probably due to the serious nature of their work.

In [103…

```
import pandas as pd
import matplotlib.pyplot as plt

#Code to see how many retweets and likes these people have got to assess their popularity by printing it as a bar

eminem_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\eminem.csv")
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames.csv")
RishiSunak_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\rishi_sunak.csv")

dataframes = {'eminem' : eminem_df, 'lebron' : lebron_df, 'Rishi Sunak' : RishiSunak_df}


totals_list = [] #Initialize an empty list

# Loop thorugh the dictionary containing the dataframes
for key, df in dataframes.items():
    df['created_at'] = pd.to_datetime(df['created_at'])
    # Only the tweets from 2022
    tweets_2022 = df[df['created_at'].dt.year == 2022]
    # The total of tweets and likes calculated
    total_tweets = len(tweets_2022)
    total_likes = tweets_2022['favorite_count'].sum()
    # Add them to the list initialized earlier
    totals_list.append({'dataframe': key, 'sum of likes 2022': total_likes})


# Create a new dataframe with the totals
totals_df = pd.DataFrame(totals_list)

totals_df.set_index('dataframe', inplace=True)
# Bar chart plotted
ax = totals_df.plot(kind='bar', rot=0)

# Add axis labels and a title
ax.set_xlabel('Influencer')
ax.set_ylabel('Like Count')
ax.set_title('Like Counts in 2022')


#save the image
plt.savefig("C:\\Users\\ahmet\\Downloads\\CS2PP22_Assessment\\data\\Task2\\like_tweet.png")

# Show the plot
plt.show()
```



To visualise our previous findings in the form of a bar chart, I decided to not include Elon Musk. That is because all the tweets extracted from him are from this year and it takes some time for likes and retweets to plateau. I wanted to work with data that was already finalised to produce the most accurate representation that I could, which would be possible by using data from 2022.

The resulting graph suggests that despite tweeting 134 times (compared 324 for Rishi Sunak)in 2022, Eminem's overall likes are almost the same as that of Rishi Sunak. This likely due to the Eminem having many followers more than Rishi Sunak.

Furthermore, for my dataset, I wanted to research the relationship between the number of tweets and the number of emojis used. So I plotted another graph comparing the two.

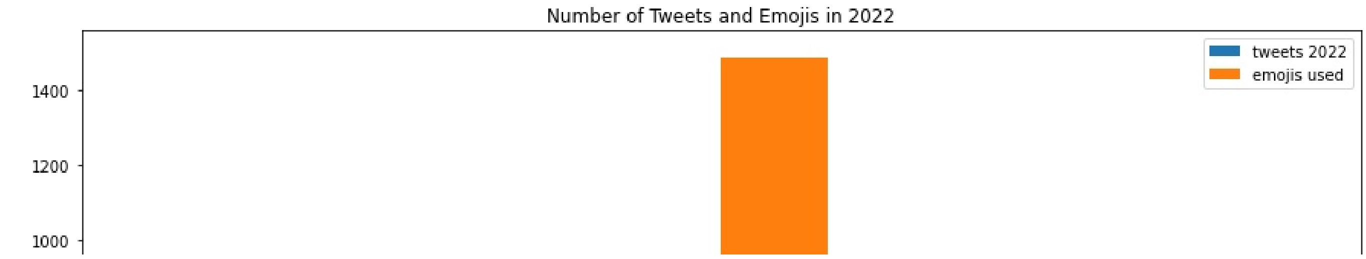
In [104…

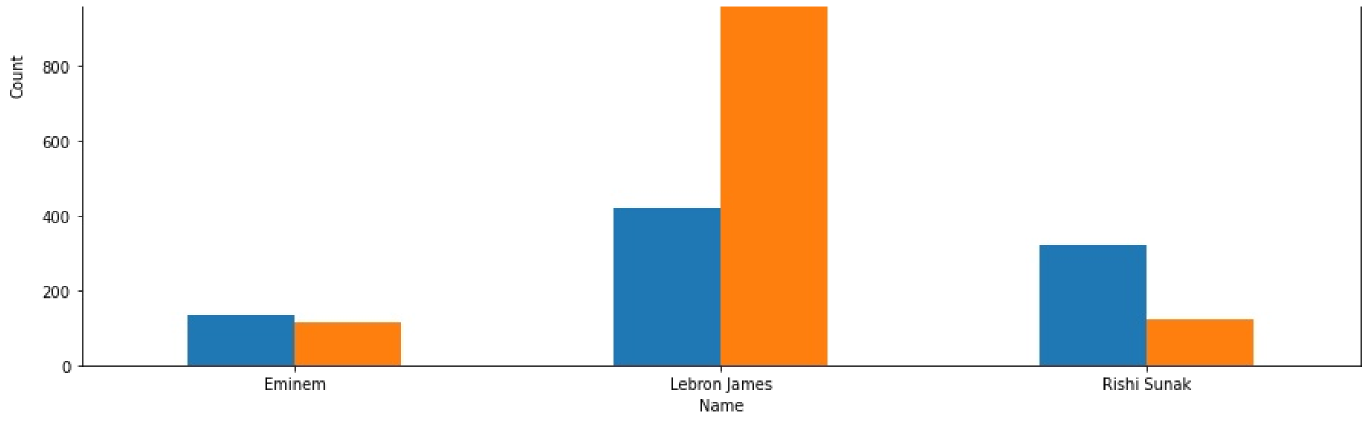
```
import pandas as pd
import emoji


eminem_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\eminem.csv")
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames.csv")
RishiSunak_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\rishi_sunak.csv")

#Dictionary containing the data
dataframes = {'Eminem' : eminem_df, 'Lebron James' : lebron_df, 'Rishi Sunak' : RishiSunak_df}


#new dataframe initialized.
tweet_emoji_df = pd.DataFrame(columns=['name', 'tweets 2022', 'emojis used'])

for name, df in dataframes.items(): #looping through the dictionary

    df['created_at'] = pd.to_datetime(df['created_at'])

#Only for 2022
    df_2022 = df[df['created_at'].dt.year == 2022]


    tweets_2022 = len(df_2022)

    def count_emojis(tweet):
        return len([char for char in tweet if char in emoji.EMOJI_DATA])
    emoji_sum = df_2022['tweet'].apply(count_emojis).sum()

#populate the new dataframe
    tweet_emoji_df = tweet_emoji_df.append({'name': name, 'tweets 2022': tweets_2022, 'emojis used': emoji_sum},

# print the aggregated data
print(tweet_emoji_df)
```

	name	tweets 2022	emojis used
0	Eminem	134	112
1	Lebron James	423	1484
2	Rishi Sunak	324	122

In [105…

```
import pandas as pd
import matplotlib.pyplot as plt

#use the recently populated dataframe
graph_df = tweet_emoji_df

# create the bar chart
ax = graph_df.plot(kind='bar', x='name', y=['tweets 2022', 'emojis used'], rot=0, figsize=(15,7))

# set the chart title and axis labels
ax.set_title('Number of Tweets and Emojis in 2022')
ax.set_xlabel('Name')
ax.set_ylabel('Count')

plt.savefig("C:\\Users\\ahmet\\Downloads\\CS2PP22_Assessment\\data\\Task2\\tweet_emoji.png")


plt.show()
```





The output shows that Lebron James has the highest ratio of emojis to tweets. So, our data so far suggests that there is a positive correlation between the ratio of emojis to tweets and total likes. Even though Rishi Sunak has used more emojis than Eminem, his emoji to tweet ratio is lower and so are his total likes. This conclusion is only based on our dataset with limited sample frame, and please note there may well be other external factors that we haven't taken into account such as the number of followers that can greatly influence the total number of likes. If a machine learning algortihm was trained with this data, it might possibly come to false conclusions such as the one I addressed.

Data Processing

This code changes the column "created_at" to "date" to make it more understandable.

In [114…

```
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames.csv")

#update the dataframe
lebron_df = lebron_df.rename(columns={'created_at': 'date'})


lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False

print(lebron_df)
```

	Unnamed: 0	date	tweet	favorite_count	retweet_count	is_quote_status	favorited
0	0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	3	2023-03-09 22:26:18+00:00	RT @fox8news: The LeBron James Family Foundati...	0	368	False	False
4	4	2023-03-09 16:07:03+00:00	RT @SLAMKicks: What started out in the digital...	0	130	False	False
...	...	...	...	...	...	...	...
495	495	2022-05-09 01:06:19+00:00	OMFG!!!!!!!! https://t.co/7cAyX6KuXs	37518	2737	True	False
496	496	2022-05-08 16:19:51+00:00	Where y'all finding all this content lately. M...	34928	2450	True	False
497	497	2022-05-08 01:54:31+00:00	Yessir!!!! Went yard then hit the "Silencer"!!...	51669	4687	True	False
498	498	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□◎□□	6180	126	False	False
499	499	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

500 rows × 7 columns

The "Unnamed: 0" column is dropped.

In [115…

```
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")

lebron_df = lebron_df.drop('Unnamed: 0', axis=1)
```

```
print(lebron_df)

lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False
```

	date	tweet	favorite_count	retweet_count	is_quote_status	favorited
0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-09 22:26:18+00:00	RT @fox8news: The LeBron James Family Foundati...	0	368	False	False
4	2023-03-09 16:07:03+00:00	RT @SLAMKicks: What started out in the digital...	0	130	False	False
...	...	...	...	...	...	...
495	2022-05-09 01:06:19+00:00	OMFG!!!!!!!! https://t.co/7cAyX6KuXs	37518	2737	True	False
496	2022-05-08 16:19:51+00:00	Where y'all finding all this content lately. M...	34928	2450	True	False
497	2022-05-08 01:54:31+00:00	Yessir!!!! Went yard then hit the "Silencer"!!...	51669	4687	True	False
498	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
499	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

500 rows × 6 columns

"favorite_count" column is changed to "likes" to make it more convenient and easy to understand.

In [116…

```
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")
lebron_df = lebron_df.rename(columns={'favorite_count': 'likes'})
print(lebron_df)

lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False
```

	date	tweet	likes	retweet_count	is_quote_status	favorited
0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-09 22:26:18+00:00	RT @fox8news: The LeBron James Family Foundati...	0	368	False	False
4	2023-03-09 16:07:03+00:00	RT @SLAMKicks: What started out in the digital...	0	130	False	False
...	...	...	...	...	...	...
495	2022-05-09 01:06:19+00:00	OMFG!!!!!!!! https://t.co/7cAyX6KuXs	37518	2737	True	False
496	2022-05-08 16:19:51+00:00	Where y'all finding all this content lately. M...	34928	2450	True	False
497	2022-05-08 01:54:31+00:00	Yessir!!!! Went yard then hit the "Silencer"!!...	51669	4687	True	False
498	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
499	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

500 rows × 6 columns

Duplicate rows that might exist are dropped

In [117…

```
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")
lebron_df = lebron_df.drop_duplicates()
print(lebron_df)
lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False
```

	date	tweet	likes	retweet_count	is_quote_status	favorited
0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-09 22:26:18+00:00	RT @fox8news: The LeBron James Family Foundati...	0	368	False	False
4	2023-03-09 16:07:03+00:00	RT @SLAMKicks: What started out in the digital...	0	130	False	False
...	...	...	...	...	...	...
495	2022-05-09 01:06:19+00:00	OMFG!!!!!!!! https://t.co/7cAyX6KuXs	37518	2737	True	False
496	2022-05-08 16:19:51+00:00	Where y'all finding all this content lately. M...	34928	2450	True	False
497	2022-05-08 01:54:31+00:00	Yessir!!!! Went yard then hit the "Silencer"!!...	51669	4687	True	False

498	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
499	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

500 rows × 6 columns

If the user retweets something, then the favorite_count (now likes) returns 0. So to avoid the tweets with 0 likes interfering with our regression model, we drop them.

In [119…

```
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")

lebron_df = lebron_df[lebron_df['likes'] != 0]


print(lebron_df)
lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False
```

	date	tweet	likes	retweet_count	is_quote_status	favorited
0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-08 05:27:59+00:00	YESSIR!!!!! #LakeShow @AntDavis23 you're a AN...	80884	9587	False	False
4	2023-03-07 04:43:15+00:00	Man Bronny definitely better than some of thes...	128851	8980	False	False
...	...	...	...	...	...	...
441	2022-05-09 01:06:19+00:00	OMFG!!!!!!!! https://t.co/7cAyX6KuXs	37518	2737	True	False
442	2022-05-08 16:19:51+00:00	Where y'all finding all this content lately. M...	34928	2450	True	False
443	2022-05-08 01:54:31+00:00	Yessir!!!! Went yard then hit the "Silencer"!!...	51669	4687	True	False
444	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
445	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

446 rows × 6 columns

If favorited for a tweet is true, if those tweets have been liked by me, we drop them.

In [122…

```
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")
lebron_df = lebron_df[lebron_df["favorited"] != True]
print(lebron_df)
lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False
```

	date	tweet	likes	retweet_count	is_quote_status	favorited
0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-08 05:27:59+00:00	YESSIR!!!!! #LakeShow @AntDavis23 you're a AN...	80884	9587	False	False
4	2023-03-07 04:43:15+00:00	Man Bronny definitely better than some of thes...	128851	8980	False	False
...	...	...	...	...	...	...
441	2022-05-09 01:06:19+00:00	OMFG!!!!!!!! https://t.co/7cAyX6KuXs	37518	2737	True	False
442	2022-05-08 16:19:51+00:00	Where y'all finding all this content lately. M...	34928	2450	True	False
443	2022-05-08 01:54:31+00:00	Yessir!!!! Went yard then hit the "Silencer"!!...	51669	4687	True	False
444	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
445	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

446 rows × 6 columns

If a particular tweet is a quote tweet, we drop them.

In [123…

```
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")

lebron_df = lebron_df[lebron_df["is_quote_status"] != True]
print(lebron_df)
```

```
lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False
```

	date	tweet	likes	retweet_count	is_quote_status	favorited
0	2023-03-12 04:05:59+00:00	□□□	60022	5213	False	False
1	2023-03-11 05:46:48+00:00	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11 00:53:12+00:00	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-08 05:27:59+00:00	YESSIR!!!!! #LakeShow @AntDavis23 you’re a AN...	80884	9587	False	False
4	2023-03-07 04:43:15+00:00	Man Bronny definitely better than some of thes...	128851	8980	False	False
...	...	...	...	...	...	...
438	2022-05-11 03:10:22+00:00	YES!!! CHUCK WON	49564	1725	False	False
439	2022-05-09 08:31:04+00:00	@bubbawatson Love you too brother!! □□✊□❤□	2339	27	False	False
440	2022-05-09 03:47:08+00:00	Hate on me, I blew but I'm the same OG\nPeople...	55325	3479	False	False
444	2022-05-07 21:42:19+00:00	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
445	2022-05-06 20:00:06+00:00	@KingJosiah54	12870	276	False	False

289 rows × 6 columns

Lastly, we edit the date column, so that all the tweets are shown in dd-mm-yy format, to make it more convenient.

In [125…

```
import pandas as pd

# Create example DataFrame
lebron_df = lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified

# Convert "date" column to datetime
lebron_df['date'] = pd.to_datetime(lebron_df['date'])

# Extract only date part and drop the time
lebron_df['date'] = lebron_df['date'].dt.date

#display the dataframe
print(lebron_df)

lebron_df.to_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv",index = False
```

	date	tweet	likes	retweet_count	is_quote_status	favorited
0	2023-03-12	□□□	60022	5213	False	False
1	2023-03-11	□□□□□ Man I love this team!!! #Lakeshow	106554	15329	False	False
2	2023-03-11	@LegionHoops HANDS DOWN!!!!!! Mike Brown got t...	34173	1289	False	False
3	2023-03-08	YESSIR!!!!! #LakeShow @AntDavis23 you’re a AN...	80884	9587	False	False
4	2023-03-07	Man Bronny definitely better than some of thes...	128851	8980	False	False
...	...	...	...	...	...	...
284	2022-05-11	YES!!! CHUCK WON	49564	1725	False	False
285	2022-05-09	@bubbawatson Love you too brother!! □□✊□❤□	2339	27	False	False
286	2022-05-09	Hate on me, I blew but I'm the same OG\nPeople...	55325	3479	False	False
287	2022-05-07	@patbev21 You already know bro!!! □□✊□□	6180	126	False	False
288	2022-05-06	@KingJosiah54	12870	276	False	False

289 rows × 6 columns

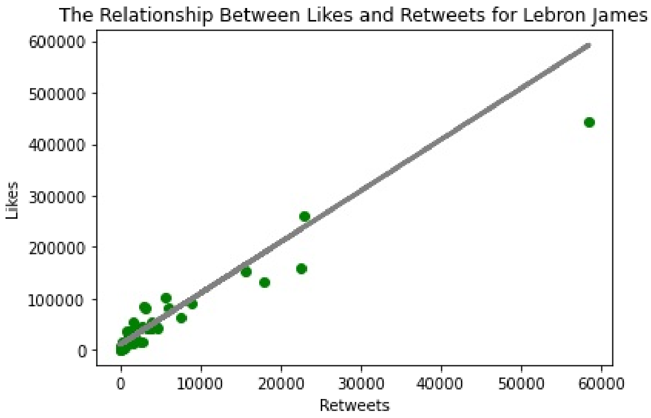
Regression Analysis

In [126…

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load a lebronjames_modified.csv file to a dataframe
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")

# the feature and target columns are extracted
x = lebron_df[['retweet_count']]
```

```
y = lebron_df['likes']

# Dataset split into training and testing sets.
x_train_set, x_test_set, y_train_set, y_test_set = train_test_split(X, y, test_size=0.2, random_state = 42)

# Linear regression model is fit to data
reg_model = LinearRegression()
reg_model.fit(x_train_set, y_train_set)

# Make predictions on the testing data
y_pred = reg_model.predict(x_test_set)

# Mean squared error and R squared are used to evaluate the data and printed
mse = mean_squared_error(y_test_set, y_pred)
r2 = r2_score(y_test_set, y_pred)
print("Mean squared error: {:.2f}".format(mse))
print("R squared is: {:.2f}".format(r2))

# plots the data and draws the regression line in the colours specified.
plt.scatter(x_test_set, y_test_set, color='green')
plt.plot(x_test_set, y_pred, color='grey', linewidth=3)

# Arrange the title and axis labels.
plt.title("The Relationship Between Likes and Retweets for Lebron James")
plt.xlabel("Retweets")
plt.ylabel("Likes")

# Display the plot
plt.show()
```

```
Mean squared error: 726698512.51
R squared is: 0.85
```



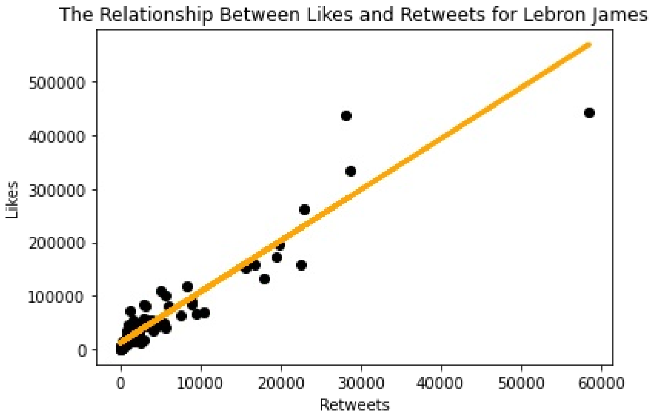
In the above cell, I have used the linear regression from sklearn to research and visualise the correlation between likes and retweets. Not suprisingly, these two elements have a strongly positive correleation with an R squared (for this model) value of 0.85. For this model, test_size was kept at 0.2 which means 80% of the data was used to train our model, whereas 20% is used to test it with unseen data.

In [127…

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load a lebronjames_modified.csv file to a dataframe
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")

# the feature and target columns are extracted
x = lebron_df[['retweet_count']]
y = lebron_df['likes']

# Dataset split into training and testing sets.
x_train_set, x_test_set, y_train_set, y_test_set = train_test_split(X, y, test_size=0.4, random_state = 42)

# Linear regression model is fit to data
reg_model = LinearRegression()
reg_model.fit(x_train_set, y_train_set)

# Make predictions on the testing data
y_pred = reg_model.predict(x_test_set)

# Mean squared error and R squared are used to evaluate the data and printed
mse = mean_squared_error(y_test_set, y_pred)
r2 = r2_score(y_test_set, y_pred)
print("Mean squared error: {:.2f}".format(mse))
print("R squared is: {:.2f}".format(r2))
```

```
# plots the data and draws the regression line in the colours specified.
plt.scatter(x_test_set, y_test_set, color='black')
plt.plot(x_test_set, y_pred, color='orange', linewidth=3)

# Arrange the title and axis labels.
plt.title("The Relationship Between Likes and Retweets for Lebron James")
plt.xlabel("Retweets")
plt.ylabel("Likes")

# Display the plot
plt.show()
```

```
Mean squared error: 660192365.93
R squared is: 0.88
```



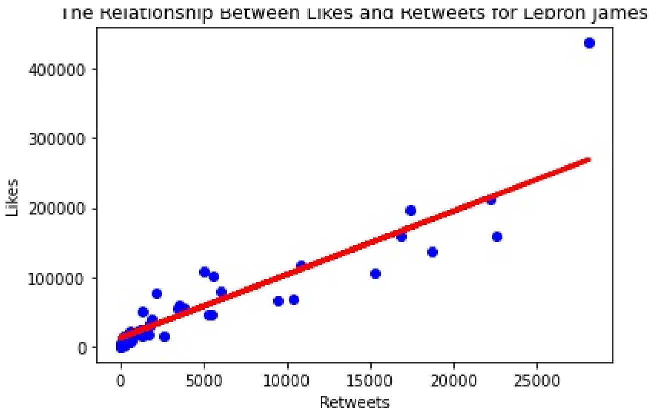
To evaluate how my model responds, for this instance, I have kept the random state the same at 42, however, I increased the test_size to 0.4. This means that the model now trains on 20% less data, so it is not as prepared for unseen data, however, it has access to more data that can be used for testing. Unlike the previous model, R squared is printed out to be 0.88 ,though still indicating a very positive correlation.

In [128…

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load a lebronjames_modified.csv file to a dataframe
lebron_df = pd.read_csv(r"C:\Users\ahmet\Downloads\CS2PP22_Assessment\data\Task2\lebronjames_modified.csv")

# the feature and target columns are extracted
x = lebron_df[['retweet_count']]
y = lebron_df['likes']

# Dataset split into training and testing sets.
x_train_set, x_test_set, y_train_set, y_test_set = train_test_split(X, y, test_size=0.25, random_state = 100)

# Linear regression model is fit to data
reg_model = LinearRegression()
reg_model.fit(x_train_set, y_train_set)

# Make predictions on the testing data
y_pred = reg_model.predict(x_test_set)

# Mean squared error and R squared are used to evaluate the data and printed
mse = mean_squared_error(y_test_set, y_pred)
r2 = r2_score(y_test_set, y_pred)
print("Mean squared error: {:.2f}".format(mse))
print("R squared is: {:.2f}".format(r2))

# plots the data and draws the regression line in the colours specified.
plt.scatter(x_test_set, y_test_set, color='blue')
plt.plot(x_test_set, y_pred, color='red', linewidth=3)

# Arrange the title and axis labels.
plt.title("The Relationship Between Likes and Retweets for Lebron James")
plt.xlabel("Retweets")
plt.ylabel("Likes")

# Display the plot
plt.show()
```

```
Mean squared error: 696149540.82
R squared is: 0.84
```

The Relationship Between Likes and Retweets for Lebron James



This time, I changed both the test_size and the random_state. The model now trains on 75% of the data and tests on 25% with random_state being equal to 100. We get an R squared value of 0.84, suggesting the least positive correlation compared to the two previous models. However, all things considered, all three models indicate a strong positive correlation with the R sqaured value ranging between 0.84-0.88.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js