A Mini Project Report

On

Twitter Hashtag Tracker

Held at

COE, MIT Moradabad

Submitted in partial fulfillment of award of

BACHELOR OF TECHNOLOGY

Degree

in

Computer Science & Engineering

By

Kratika - 1900820100062

Under The Guidance Of

Dr. Neelaksh Sheel

Associate Professor

Ms. Prachi Agarwal

Assistant Professor

Department of Computer Science & Engineering
Moradabad Institute of Technology Moradabad(U.P.)
SESSION – 2022-23

CERTIFICATE

Certified that the Mini Project Report entitled "**Twitter Hashtag Tracker**" submitted by Kratika (1900820100062) is her own work and has been done under our supervision. It is recommended that the candidate may now be evaluated for their project work by the University.

Dr. Neelaksh Sheel
Associate Professor
Dept. CS&E
MIT, Moradabad

Ms. Prachi Agarwal
Assistant Professor
Dept. CS&E
MIT, Moradabad

CERTIFICATE

ACKNOWLEDGEMENT

First of all, we would like to express our gratitude towards Moradabad Institute of Technology, Moradabad for providing us a platform to present our project at such an esteemed institute.

We would like to thank Dr. Manish Gupta HOD ,Computer Science & Engineering Department, Moradabad Institute of Technology, Moradabad for his constant support.

We are also thankful to Dr. Neelaksh Sheel, Associate Professor, Ms. Prachi Agarwal, Assistant Professor, Computer Science & Engineering Department ,Moradabad Institute of Technology, Moradabad for their guidance.

We are also thankful to all the staff members of the department for their full cooperation and help.

KRATIKA - 1900820100062

ABSTRACT

This is a Project Report on “Twitter Hashtag Tracker”. During the making or Development of the project we explored new ideas and libraries in python for implementing Web based Application, Data Science and Web Scrapping.

The Project is the output of our planning, schedule, programming skills and the hard work, and this report reflect our steps taken at various levels of programming skills, planning and schedule. We have learnt a lot during this project in our coding skills and deep concept related to these kind of projects.

Our project is “Twitter Hashtag Tracker ”. This is a python based Web application. It is a dashboard that converts raw data into useful, accurate and insightful metrics and predictive Twitter Hashtag analytics.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Millions of people are using social network sites to express their emotions, opinion and disclose about their daily lives. However, people write anything such as social activities or any comment on products. Through the online communities provide an interactive forum where consumers inform and influence others. Moreover, social media provides an opportunity for business that giving a platform to connect with their customers such as social media to advertise or speak directly to customers for connecting with customer's perspective of products and services. In contrast, consumers have all the power when it comes to what consumers want to see and how consumers respond. With this, the company's success & failure is publicly shared and end up with word of mouth. However, the social network can change the behavior and decision making of consumers, for example, it mentions that 87% of internet users are influenced in their purchase and decision by customer's review. So that, if organization can catch up faster on what their customer's think, it would be more beneficial to organize to react on time and come up with a good strategy to compete their competitors.

Twitter is a popular real time microblogging service that allows users to share short information known as tweets which are limited to 140 characters .Users write tweets to express their opinion about various topics relating to their daily lives. Twitter is an ideal platform for the extraction of general public opinion on specific issues .A collection of tweets is used as the primary corpus for sentiment analysis, which refers to the use of opinion mining or natural language processing. Twitter, with 500 million users and million messages per day, has quickly became a valuable asset for organizations to invigilate their reputation and brands by extracting and analyzing the sentiment of the tweets by the public about their products, services market and even about

competitors highlighted that, from the social media generated opinions with the mammoth growth of the world wide web, super volumes of opinion texts in the form of tweets, reviews, blogs or any discussion groups and forums are available for analysis, thus making the world wide web the fastest, most comprising and easily accessible medium for sentiment analysis.

It is defined a social media as a group of Internet-based applications that create on the ideological and technological foundations of Web2.0 which is allowed to build and exchange of user generated contents. In a discussion of Internet World Start, identified that a trend of internet users is increasing and continuing to spend more time with social media by the total time spent on mobile devices and social media in the U.S.across PC increased by 37percent to121billion minutes in 2012, compared to 88 billion minutes in 2011. On the other hand, businesses use social networking sites to find and communicate with clients, business can be demonstrated damage to productivity caused by social networking . As social media can be posted so easily to the public, it can harm private information to spread out in the social world . On the contrary, discussed that the benefits of participating in social media have gone beyond simply social sharing to build organization's reputation and bring in career opportunities and monetary income. In addition, mentioned that the social media is also being used for advertisement by companies for promotions, professionals for searching, recruiting, social learning online and electronic commerce. Electronic commerce or E-commerce refers to the purchase and sale of goods or services online which can via social media, such has Twitter which is convenient due to its 24-hours availability, ease of customer service and global reach. Among the reasons of why business tends to use more social media is for getting insight into consumer behavioral tendencies, market intelligence and present an opportunity to learn about customer review and perceptions.

Opinion mining refers to the broad area of natural language processing, text mining, computational linguistics, which involves the computational study of sentiments, opinions and emotions expressed in text. Although, view or attitude based on emotion instead of reason is often colloquially referred to as a sentiment. Hence, lending to an equivalent for opinion mining or sentiment analysis. stated that opinion mining has many application domains including

accounting, law, research, entertainment, education, technology, politics, and marketing. In earlier days many social media have given web users avenue for opening up to express and share their thoughts and opinions.

Tweet Binder is the hashtag tracking tool for agencies and marketers. Our Twitter Hashtag Analytics are used by clients all around the world. Thousands of hashtag trackers are created everyday with our social listening platform. We try to listen to all the requests our clients and users make in order to improve our tweet counter. We are all ears when trying to give the most of Twitter's potential. Tracking hashtags with Tweet Binder is very simple and it will let you know the impact of your campaign. Our Twitter plans have been conceived to fulfill all your needs and to help you to amaze your clients.

CHAPTER 2

TECHNOLOGY SPECIFICATION

2.1 Data Science

Data science combines math and statistics, specialized programming, advanced analytics, artificial intelligence (AI), and machine learning with specific subject matter expertise to uncover actionable insights hidden in an organization's data. These insights can be used to guide decision making and strategic planning.

The accelerating volume of data sources, and subsequently data, has made data science is one of the fastest growing field across every industry. As a result, it is no surprise that the role of the data scientist was dubbed the "best job of the 21st century" by Harvard Business Review (link resides outside of IBM). Organizations are increasingly reliant on them to interpret data and provide actionable recommendations to improve business outcomes.

The data science lifecycle involves various roles, tools, and processes, which enables analysts to glean actionable insights. Typically, a data science project undergoes the following stages. Data ingestion: The lifecycle begins with the data collection--both raw structured and unstructured data from all relevant sources using a variety of methods. These methods can include manual entry, web scraping, and real-time streaming data from systems and devices. Data sources can include structured data, such as customer data, along with unstructured data like log files, video, audio, pictures, the Internet of Things (IoT), social media, and more.

Data storage and data processing: Since data can have different formats and structures, companies need to consider different storage systems based on the type of data that needs to be captured. Data management teams help to set standards around data storage and structure, which

facilitate workflows around analytics, machine learning and deep learning models. This stage includes cleaning data, deduplicating, transforming and combining the data using ETL (extract, transform, load) jobs or other data integration technologies. This data preparation is essential for promoting data quality before loading into a data warehouse, data lake, or other repository.

Data analysis: Here, data scientists conduct an exploratory data analysis to examine biases, patterns, ranges, and distributions of values within the data. This data analytics exploration drives hypothesis generation for a/b testing. It also allows analysts to determine the data's relevance for use within modeling efforts for predictive analytics, machine learning, and/or deep learning. Depending on a model's accuracy, organizations can become reliant on these insights for business decision making, allowing them to drive more scalability.

Communicate: Finally, insights are presented as reports and other data visualizations that make the insights—and their impact on business—easier for business analysts and other decision-makers to understand. A data science programming language such as R or Python includes components for generating visualizations; alternately, data scientists can use dedicated visualization tools.

2.2 Web Scrapping

Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites.

These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API's that allow you to access their data in a structured format. This is the best option, but there are other sites that don't allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it's best to use Web Scraping to scrape the website for data.

Web scraping requires two parts, namely the crawler and the scraper. The crawler is an artificial intelligence algorithm that browses the web to search for the particular data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract data from the website. The design of the scraper can vary greatly according to the complexity and scope of the project so that it can quickly and accurately extract the data.

Web Scrapers can extract all the data on particular sites or the specific data that a user wants. Ideally, it's best if you specify the data you want so that the web scraper only extracts that data quickly. For example, you might want to scrape an Amazon page for the types of juicers available, but you might only want the data about the models of different juicers and not the customer reviews.

So, when a web scraper needs to scrape a site, first the URLs are provided. Then it loads all the HTML code for those sites and a more advanced scraper might even extract all the CSS and Javascript elements as well. Then the scraper obtains the required data from this HTML code and outputs this data in the format specified by the user. Mostly, this is in the form of an Excel spreadsheet or a CSV file, but the data can also be saved in other formats, such as a JSON file.

Python seems to be in fashion these days! It is the most popular language for web scraping as it can handle most of the processes easily. It also has a variety of libraries that were created specifically for Web Scraping. Scrapy is a very popular open-source web crawling framework that is written in Python. It is ideal for web scraping as well as extracting data using APIs. Beautiful soup is another Python library that is highly suitable for Web Scraping. It creates a parse tree that can be used to extract data from HTML on a website. Beautiful soup also has multiple features for navigation, searching, and modifying these parse trees.

2.3 Snscrape

Snscrape is a scraping tool for social networking services (SNS). It scrapes information like user profiles, hashtags, searches, and threads and returns the discovered items, e.g. the relevant posts. It was released on July 8, 2020, and it is capable of scraping data from a variety of platforms, including the following

- Twitter
- Instagram
- Reddit
- Facebook
- Weibo
- Telegram
- Mastodon

You can use snscrape by typing its command-line interface (CLI) commands into the command prompt/terminal. If you don't feel comfortable using a terminal, you can use snscrape as a Python library, but this is not yet documented. On Twitter, it can scrape users, user profiles, hashtags, searches, tweets (single or surrounding thread), list posts, and trends.

2.4 Data Visualization

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

Data visualization can be utilized for a variety of purposes, and it's important to note that is not only reserved for use by data teams. Management also leverages it to convey organizational structure and hierarchy while data analysts and data scientists use it to discover and explain patterns and trends. Harvard Business Review (link resides outside IBM) categorizes data visualization into four key purposes: idea generation, idea illustration, visual discovery, and everyday dataviz.

Data visualization is commonly used to spur idea generation across teams. They are frequently leveraged during brainstorming or Design Thinking sessions at the start of a project by supporting the collection of different perspectives and highlighting the common concerns of the collective. While these visualizations are usually unpolished and unrefined, they help set the foundation within the project to ensure that the team is aligned on the problem that they're looking to address for key stakeholders.

Data visualization for idea illustration assists in conveying an idea, such as a tactic or process. It is commonly used in learning settings, such as tutorials, certification courses, centers of excellence, but it can also be used to represent organization structures or processes, facilitating communication between the right individuals for specific tasks. Project managers frequently use Gantt charts and waterfall charts to illustrate workflows. Data modeling also uses abstraction to represent and better understand data flow within an enterprise's information system, making it easier for developers, business analysts, data architects, and others to understand the relationships in a database or data warehouse.

Data visualization is a critical step in the data science process, helping teams and individuals convey data more effectively to colleagues and decision makers. Teams that manage reporting systems typically leverage defined template views to monitor performance. However, data visualization isn't limited to performance dashboards. For example, while text mining an analyst may use a word cloud to capture key concepts, trends, and hidden relationships within this unstructured data. Alternatively, they may utilize a graph structure to illustrate relationships between entities in a knowledge graph. There are a number of ways to represent different types of data, and it's important to remember that it is a skillset that should extend beyond your core analytics team.

2.5 Dash

Downloaded 800,000 times per month, Dash is the original low-code framework for rapidly building data apps in Python, R, Julia, and F# (experimental). Written on top of Plotly.js and React.js, Dash is ideal for building and deploying data apps with customized user interfaces. It's particularly suited for anyone who works with data.

Through a couple of simple patterns, Dash abstracts away all of the technologies and protocols that are required to build a full-stack web app with interactive data visualization. Dash is simple enough that you can bind a user interface to your code in less than 10 minutes. Dash apps are rendered in the web browser. You can deploy your apps to VMs or Kubernetes clusters and then share them through URLs. Since Dash apps are viewed in the web browser, Dash is inherently cross-platform and mobile ready.

There is a lot behind the framework. To learn more about how it is built and what motivated Dash, read our announcement letter or our post Dash is React for Python. Dash is an open source library released under the permissive MIT license. Plotly develops Dash and also offers a platform for writing and deploying Dash apps in an enterprise environment.

2.6 Plotly

Plotly is a technical computing company headquartered in Montreal, Quebec, that develops online data analytics and visualization tools. Plotly provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST.

Plotly offers open-source and enterprise products. Dash is an open-source Python, R, and Julia framework for building web-based analytic applications. Many specialized open-source Dash libraries exist that are tailored for building domain-specific Dash components and applications. Some examples are Dash DAQ, for building data acquisition GUIs to use with scientific instruments, and Dash Bio, which enables users to build custom chart types, sequence analysis tools, and 3D rendering tools for bioinformatics applications.

Dash Enterprise is Plotly's paid product for building, testing, deploying, managing and scaling Dash applications organization-wide. Chart Studio Cloud is a free, online tool for creating interactive graphs. It has a point-and-click graphical user interface for importing and analyzing data into a grid and using stats tools. Graphs can be embedded or downloaded.

Chart Studio Enterprise is a paid product that allows teams to create, style, and share interactive graphs on a single platform. It offers expanded authentication and file export options, and does not limit sharing and viewing. Data visualization libraries Plotly.js is an open-source JavaScript library for creating graphs and powers Plotly.py for Python, as well as Plotly. R for R, MATLAB, Node.js, Julia, and Arduino and a REST API. Plotly can also be used to style interactive graphs with Jupyter notebook. Figure Converters which convert matplotlib, ggplot2, and IGOR Pro graphs into interactive, online graphs.

CHAPTER 3

MODULE WISE DESCRIPTION

3.1 Pre requisites

It’s possible to work with data in Python, but there are quite a few open-source libraries that make Python data tasks much, much easier.

NumPy : NumPy (Numerical Python) is a perfect tool for scientific computing and performing basic and advanced array operations.

Pandas: Pandas is a library created to help developers work with "labeled" and "relational" data intuitively. It's based on two main data structures: "Series" (one-dimensional, like a list of items) and "Data Frames".

Matplotlib: This is a standard data science library that helps to generate data visualizations such as two-dimensional diagrams and graphs (histograms, scatterplots, non-Cartesian coordinates graphs). Matplotlib is one of those plotting libraries that are really useful in data science projects — it provides an object-oriented API for embedding plots into applications.

OS: The OS module in python provides functions for interacting with the operating system. OS, comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality.

Snscrape: Snscrape is a scraping tool for social networking services (SNS). It scrapes information like user profiles, hashtags, searches, and threads and returns the discovered items, e.g. the relevant posts. It was released on July 8, 2020, and it is capable of scraping data from a variety of platforms.

Plotly: Plotly is a technical computing company headquartered in Montreal, Quebec, that develops online data analytics and visualization tools. Plotly provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST.

Dash: Dash abstracts away all of the technologies and protocols that are required to build a full-stack web app with interactive data visualization. Dash is simple enough that you can bind a user interface to your code in less than 10 minutes. Dash apps are rendered in the web browser. You can deploy your apps to VMs or Kubernetes clusters and then share them through URLs. Since Dash apps are viewed in the web browser, Dash is inherently cross-platform and mobile ready.

3.2 Data Pre-Processing

It contains the following steps The convolution is performed on the input data with the use of a filter or kernel to then produce a feature map. We execute a convolution by sliding the filter over the input. At every location, a matrix multiplication is performed and sums the result onto the feature map.

Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer.

Fully Connected Layer is simply, feed forward neural networks. Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.

3.3 Feature Extraction

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data. Feature extraction can be accomplished manually or automatically.

Manual feature extraction requires identifying and describing the features that are relevant for a given problem and implementing a way to extract those features. In many situations, having a good understanding of the background or domain can help make informed decisions as to which features could be useful. Over decades of research, engineers and scientists have developed feature extraction methods for images, signals, and text. An example of a simple feature is the mean of a window in a signal.

Automated feature extraction uses specialized algorithms or deep networks to extract features automatically from signals or images without the need for human intervention. This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms. Wavelet scattering is an example of automated feature extraction.
With the ascent of deep learning, feature extraction has been largely replaced by the first layers of deep networks – but mostly for image data. For signal and time-series applications, feature extraction remains the first challenge that requires significant expertise before one can build effective predictive models.

2.4 Data Visualization

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

Data visualization can be utilized for a variety of purposes, and it's important to note that is not only reserved for use by data teams. Management also leverages it to convey organizational structure and hierarchy while data analysts and data scientists use it to discover and explain patterns and trends. Harvard Business Review (link resides outside IBM) categorizes data visualization into four key purposes: idea generation, idea illustration, visual discovery, and everyday dataviz.

Data visualization is commonly used to spur idea generation across teams. They are frequently leveraged during brainstorming or Design Thinking sessions at the start of a project by supporting the collection of different perspectives and highlighting the common concerns of the collective. While these visualizations are usually unpolished and unrefined, they help set the foundation within the project to ensure that the team is aligned on the problem that they're looking to address for key stakeholders.

Data visualization for idea illustration assists in conveying an idea, such as a tactic or process. It is commonly used in learning settings, such as tutorials, certification courses, centers of excellence, but it can also be used to represent organization structures or processes, facilitating communication between the right individuals for specific tasks. Project managers frequently use Gantt charts and waterfall charts to illustrate workflows. Data modeling also uses abstraction to represent and better understand data flow within an enterprise's information system, making it easier for developers, business analysts, data architects, and others to understand the relationships in a database or data warehouse.

Data visualization is a critical step in the data science process, helping teams and individuals convey data more effectively to colleagues and decision makers. Teams that manage reporting systems typically leverage defined template views to monitor performance. However, data visualization isn't limited to performance dashboards. For example, while text mining an

analyst may use a word cloud to capture key concepts, trends, and hidden relationships within this unstructured data. Alternatively, they may utilize a graph structure to illustrate relationships between entities in a knowledge graph. There are a number of ways to represent different types of data, and it's important to remember that it is a skillset that should extend beyond your core analytics team.

Plotly is a technical computing company headquartered in Montreal, Quebec, that develops online data analytics and visualization tools. Plotly provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST.

Plotly offers open-source and enterprise products. Dash is an open-source Python, R, and Julia framework for building web-based analytic applications. Many specialized open-source Dash libraries exist that are tailored for building domain-specific Dash components and applications. Some examples are Dash DAQ, for building data acquisition GUIs to use with scientific instruments, and Dash Bio, which enables users to build custom chart types, sequence analysis tools, and 3D rendering tools for bioinformatics applications.

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Flow Diagrams

Flow diagram is a collective term for a diagram representing a flow or set of relationships in a system. The term flow diagram is also used as a synonym for flowchart, and sometimes as a counterpart of the flowchart.

The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Any drawing program can be used to create flowchart diagrams, but these will have no underlying data model to share data with databases or other programs such as project management systems or spreadsheet. Many software packages exist that can create flowcharts automatically, either directly from a programming language source code, or from a flowchart description language.

A flowchart is described as "cross-functional" when the chart is divided into different vertical or horizontal parts, to describe the control of different organizational units. A symbol appearing in a particular part is within the control of that organizational unit.

A cross-functional flowchart allows the author to correctly locate the responsibility for performing an action or making a decision, and to show the responsibility of each organizational unit for different parts of a single process.

So we have made following flow diagram to explain the flow of our project.

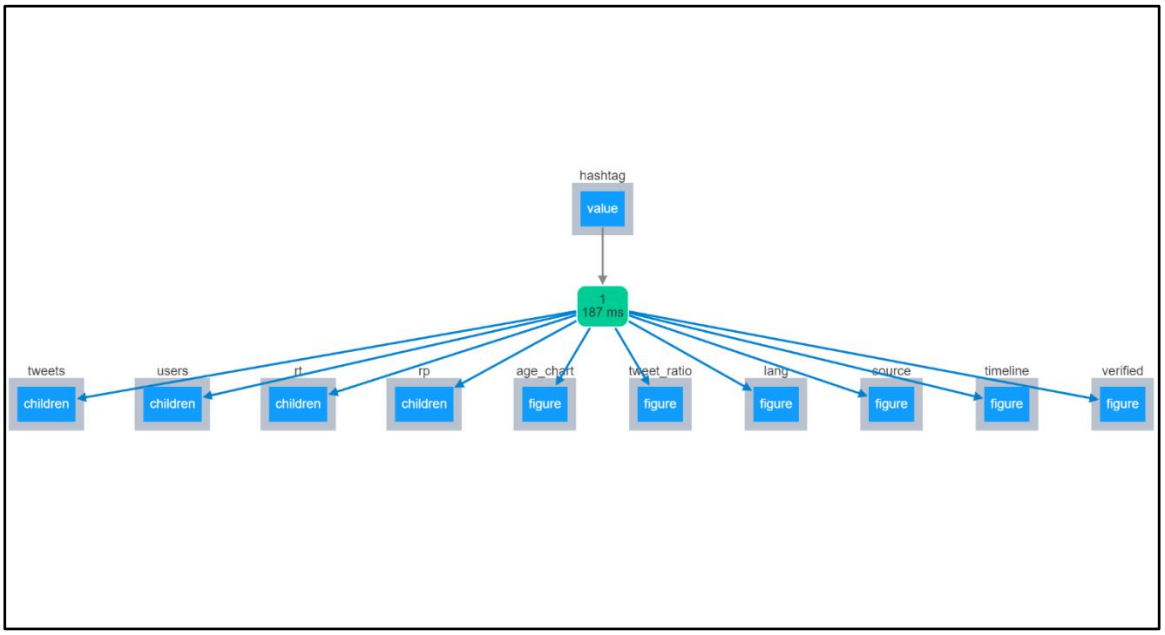

Figure 4.1 Flow Chart

4.2 Code

This Project includes various files such as for backend, frontend and external CSS. Here is the implementation of the project in python :

```
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input,Output
import plotly.graph_objs as go
from twitter_hashtag_tracker import twitter_analytics

app = dash.Dash(__name__)


app.title = 'Twitter Hashtag Tracker'
server = app.server
```

```
fig_age= dict(
        data=[go.Bar(x=[0,0,0], y=[0,0,0])],

        layout=go.Layout(title="Twitter User Age Distribution" ,
                xaxis=dict(title='Years',showgrid=False),
                yaxis=dict(title='Number of Users',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)',
                font= dict(color='rgb(255,255,255)')
                  ))
fig_ratio= dict(
        data=[go.Bar(x=[0,0,0], y=[0,0,0])],

        layout=go.Layout(title="" ,
                xaxis=dict(title='',showgrid=False),
                yaxis=dict(title='',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)',
                font= dict(color='rgb(255,255,255)')
                  ))
fig_lang =  dict(
        data=[go.Bar(x=[0,0,0], y=[0,0,0])],

        layout=go.Layout(title="Language Distribution" ,
                xaxis=dict(title='Languages',showgrid=False),
                yaxis=dict(title='Number of Tweets',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)',
                font= dict(color='rgb(255,255,255)')
                  ))
fig_src= dict(
        data=[go.Bar(x=[0,0,0], y=[0,0,0])],

        layout=go.Layout(title="Source Distribution" ,
                xaxis=dict(title='',showgrid=False),
                yaxis=dict(title='Number of Users',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)'
                  ))
```

```
fig_time= dict(
        data=[go.Scatter(x=[0,0,0], y=[0,0,0])],

        layout=go.Layout(title="Timeline" ,
                xaxis=dict(title='',showgrid=False),
                yaxis=dict(title='Number of Tweets',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)',
                font= dict(color='rgb(255,255,255)')
                  ))


fig_v= dict(
        data=[go.Pie(values=[0,0,0])],

        layout=go.Layout(title="Verified Users Distribution" ,
                  paper_bgcolor='rgba(0,0,0,0)',
                  plot_bgcolor='rgba(0,0,0,0)',
                  font= dict(color='rgb(255,255,255)')
                   ))




app.layout = html.Div([ html.Div([
                  html.Div([html.P([html.B('Twitter Hashtag Tracker',style={'color' :
'#538FF8'}),html.Br(),
                                    html.P('Our real-time hashtag tracker converts raw data into useful,
accurate and insightful metrics and predictive twitter Hashtag analytics.',
                                          style={'fontSize': '22px',
                                               'color' : '#b8b8b8'})]),
                  dcc.Input(id='hashtag',type='text',placeholder="enter a hashtag with
#",debounce=True,
                             )],

                          className = 'header',

  #header


                          ),
                  html.Div(id='sideimg')],style={'display': 'flex'}),
```

```
                html.Div([html.Div(html.P([html.P(html.B("0"),id='tweets',style={'fontSize' :
'56px'}),html.P("Tweets")],
                                    style={'textAlign' :'center',
                                        'width':'100%',
                                        'height' : '100%',
                                        'padding' : '30px 10px 10px 10px'}),
  #Row1 Column1
                        className = 'figures'),

                       html.Div(html.P([html.P(html.B("0"),id='users',style={'fontSize' :
'56px'}),html.P("Users")],
                                    style={'textAlign' :'center',
                                        'width':'100%',
                                        'height' : '100%',
                                        'padding' : '30px 10px 10px 10px'}),

                         className = 'figures'
   #Row1 Column2
                        ),

                       html.Div(html.P([html.P(html.B("0"),id='rt',style={'fontSize' :
'56px'}),html.P("Retweets")],
                                    style={'textAlign' :'center',
                                        'width':'100%',
                                        'height' : '100%',
                                        'padding' : '30px 10px 10px 10px'}),
   #Row1 Column3
                        className = 'figures'),

                       html.Div(html.P([html.P(html.B("0"),id='rp',style={'fontSize' :
'56px'}),html.P("Replies")],
                                    style={'textAlign' :'center',
                                        'width':'100%',
                                        'height' : '100%',
                                        'padding' : '30px 10px 10px 10px'}),
   #Row1 Column4
                        className = 'figures'),


                    ],
```

```
                    style=
                    {
                        'display': 'flex',
                        'align-items': 'center',
                        'paddingTop' : '50px',
                        'paddingLeft' : '30px',
                        'paddingRight' : '30px',
                        'textAlign' : 'center',
                        'marginRight' : '120px',
                        'marginLeft' : '120px'


                    }),
                html.Div([html.Div(

                    dcc.Graph(id='age_chart',
                            figure=fig_time,
                    style=
                    {
                        'width' : '100%',
                        'height' : '100%',


                    }),

                    className='graph',
#Row2 Column1
                        style=
                        {

                            'width' : '75%',
                            'height' : '400px',
                            'display': 'flex',
                            'align-items': 'center',
                            'padding' : '5px 5px 5px 5px',
                            'marginLeft' : '10px',
                            'marginRight':'10px',
```

```
                    }),
                 html.Div(

                dcc.Graph(id='tweet_ratio',
                     figure=fig_ratio,
                style=
                {
                   'width' : '100%',
                   'height' : '100%'

                }),

                 className='graph',
#Row2 Column2
                   style=
                   {

                      'width' : '25%',
                      'height' : '400px',
                      'display': 'flex',
                      'align-items': 'center',
                      'padding' : '5px 5px 5px 5px',
                      'marginLeft' : '10px',
                      'marginRight':'10px',


                   }),


             ],
                    style=
                   {
                      'display': 'flex',
                      'align-items': 'center',
                      'paddingTop' : '50px',
                      'paddingLeft' : '30px',
                      'paddingRight' : '30px',


                   }),
```

```
                html.Div([html.Div(

                    dcc.Graph(id='lang',
                        figure=fig_lang,
                    style=
                    {
                       'width' : '100%',
                       'height' : '100%'

                    }),

                     className='graph',
#Row3 Column1
                       style=
                       {

                           'width' : '50%',
                           'height' : '450px',
                           'display': 'flex',
                           'align-items': 'center',
                           'padding' : '5px 5px 5px 5px',
                           'marginLeft' : '10px',
                           'marginRight':'10px',


                       }),
                     html.Div(

                    dcc.Graph(id='source',
                        figure=fig_src,
                    style=
                    {
                       'width' : '100%',
                       'height' : '100%'

                    }),

                     className='graph',
#Row3 Column2
```

```
                style=
                {

                    'width' : '50%',
                    'height' : '450px',
                    'display': 'flex',
                    'align-items': 'center',
                    'padding' : '5px 5px 5px 5px',
                    'marginLeft' : '10px',
                    'marginRight':'10px',


                }),


            ],
                 style=
                {
                    'display': 'flex',
                    'align-items': 'center',
                    'paddingTop' : '50px',
                    'paddingLeft' : '30px',
                    'paddingRight' : '30px',


                }),
            html.Div([html.Div(

              dcc.Graph(id='timeline',
                   figure=fig_age,
              style=
              {
                 'width' : '100%',
                 'height' : '100%'

              }),
               className='graph',
#Row4 Column1
                style=
                {
```

```
                    'width' : '75%',
                    'height' : '400px',
                    'display': 'flex',
                    'align-items': 'center',
                    'padding' : '5px 5px 5px 5px',
                    'marginLeft' : '10px',
                    'marginRight':'10px',


                }),
                html.Div(

                dcc.Graph(id='verified',
                    figure=fig_v,
                style=
                {
                    'width' : '100%',
                    'height' : '100%'

                }),
                 className='graph',
#Row4 Column2
                    style=
                    {

                        'width' : '25%',
                        'height' : '400px',
                        'display': 'flex',
                        'align-items': 'center',
                        'padding' : '5px 5px 5px 5px',
                        'marginLeft' : '10px',
                        'marginRight':'10px',


                    }),


            ],
                    style=
                    {
                        'display': 'flex',
```

```
                    'align-items': 'center',
                    'paddingTop' : '50px',
                    'paddingLeft' : '30px',
                    'paddingRight' : '30px',


                }),
            html.Div([html.P([html.B('Twitter Hashtag Tracker'),
                              html.Br(),
                              html.P('Our real-time hashtag tracker converts raw data into useful,
accurate and insightful metrics and predictive twitter Hashtag analytics.',
                                     style={'fontSize': '16px',
                                          'color' :'#b8b8b8'}),

                              html.P('Connect Us:',
                                   style={'fontSize' : '16px',
                                        'color' :'#b8b8b8'}),
                              html.P(['LinkedIn : ',html.A('Kratika
Saxena',href='https://www.linkedin.com/in/kratika-saxena-996616229/',target='_blank',
                                                           style={'color':'#b8b8b8',
                                                                'fontSize':'18px'})],
                                     style={'fontSize' : '20px'}),

                                 ])
                    ],
                    style=
                    {
                      'padding': '120px 30px 10px 30px',
                      'width' : '101%',
                      'height' : '200px',
                      'backgroundColor' : 'rgba(241,241,241,0)',
                      'textAlign' : 'center',
                      'fontSize': '25px'   ,
                      'color' : '#538FF8',

                    })

            ],
               )

@app.callback(Output('tweets','children'),
```

```
            Output('users','children'),
            Output('rt','children'),
            Output('rp','children'),
            Output('age_chart', 'figure'),
            Output('tweet_ratio', 'figure'),
            Output('lang', 'figure'),
            Output('source', 'figure'),
            Output('timeline', 'figure'),
            Output('verified', 'figure'),
            Input('hashtag','value'))
def callback_in1(hashtag_val):
    if hashtag_val is not None:
        ans= twitter_analytics(hashtag_val)
    else:
        ans= (0,0,0,0,[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0])

    fig_age= dict(
        data=[go.Bar(x=ans[4], y=ans[5])],

       layout=go.Layout(title="Twitter User Age Distribution" ,
                    xaxis=dict(title='Years',showgrid=False),
                    yaxis=dict(title='Number of Users',showgrid=False),
                    paper_bgcolor='rgba(0,0,0,0)',
                    plot_bgcolor='rgba(0,0,0,0)',
                    font= dict(color='rgb(255,255,255)')
                       ))
    fig_ratio= dict(
        data=[go.Bar(x=ans[6], y=ans[7])],

        layout=go.Layout(title="" ,
                    xaxis=dict(title='',showgrid=False),
                    yaxis=dict(title='',showgrid=False),
                    paper_bgcolor='rgba(0,0,0,0)',
                    plot_bgcolor='rgba(0,0,0,0)',
                    font= dict(color='rgb(255,255,255)')
                       ))
    fig_lang =  dict(
        data=[go.Bar(x=ans[8], y=ans[9])],

       layout=go.Layout(title="Language Distribution" ,
                    xaxis=dict(title='Languages',showgrid=False),
```

```
                yaxis=dict(title='Number of Tweets',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)',
                font= dict(color='rgb(255,255,255)')
                  ))
    fig_src= dict(
      data=[go.Bar(x=ans[10], y=ans[11])],

      layout=go.Layout(title="Source Distribution" ,
                xaxis=dict(title='',showgrid=False),
                yaxis=dict(title='Number of Users',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)',
                font= dict(color='rgb(255,255,255)')
                  ))
    fig_time= dict(
      data=[go.Scatter(x=ans[12], y=ans[13])],

      layout=go.Layout(title="Timeline" ,
                xaxis=dict(title='',showgrid=False),
                yaxis=dict(title='Number of Tweets',showgrid=False),
                paper_bgcolor='rgba(0,0,0,0)',
                plot_bgcolor='rgba(0,0,0,0)',
                font= dict(color='rgb(255,255,255)')
                ))
    fig_v= dict(
      data=[go.Pie(labels=ans[15], values=ans[14])],

      layout=go.Layout(title="Verified Users Distribution" ,
                  paper_bgcolor='rgba(0,0,0,0)',
                  plot_bgcolor='rgba(0,0,0,0)',
                  font= dict(color='rgb(255,255,255)')
                  ))
    return ans[0],ans[1],ans[2],ans[3],fig_time,fig_ratio,fig_lang,fig_src,fig_age,fig_v




if __name__== '__main__':
    app.run_server(debug=True)
```

CHAPTER 5

RESULTS

Figure 5.1 Home page part 1

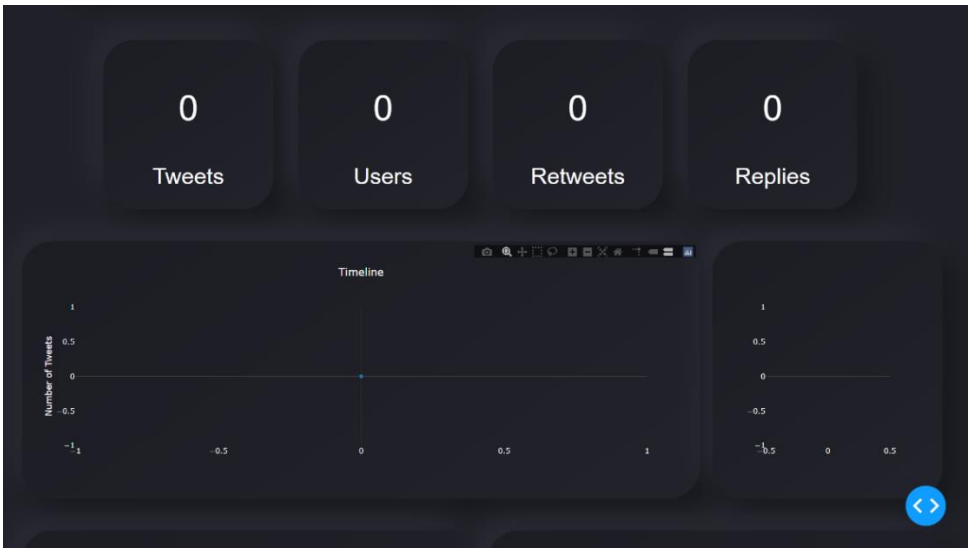

Figure 5.2 Home page part 2

Figure 5.3 Home page part 3

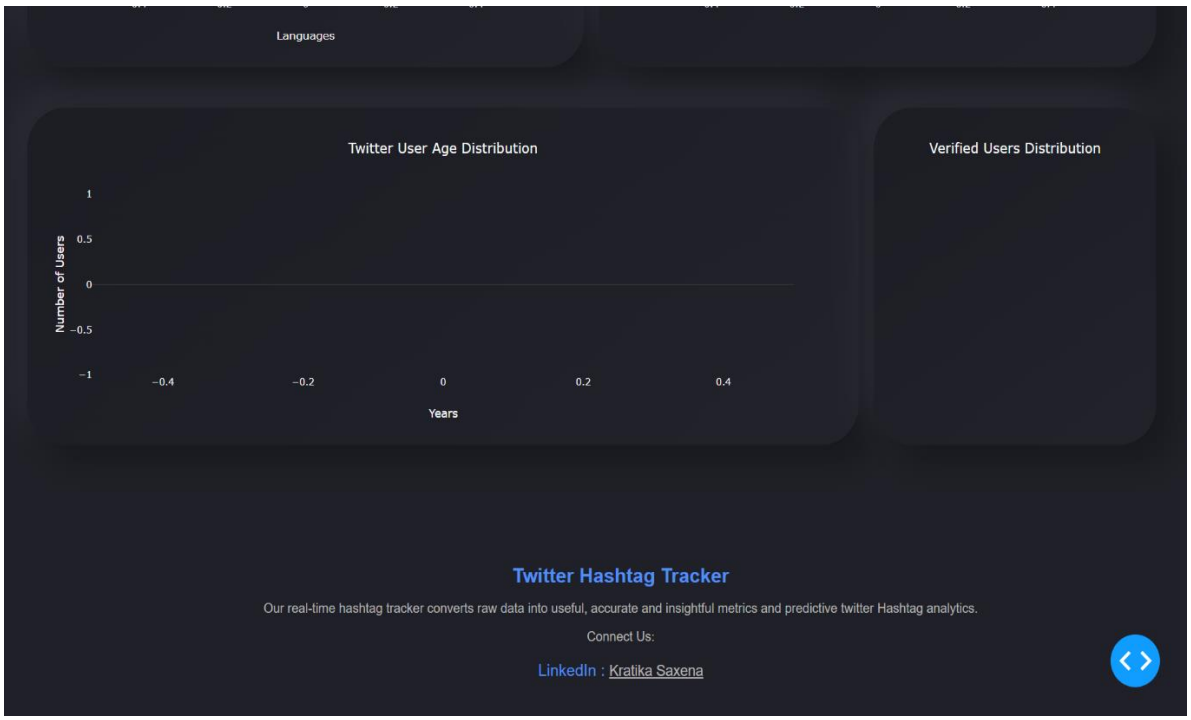

Figure 5.4 Home page part 4

Figure 5.5 Result part 1

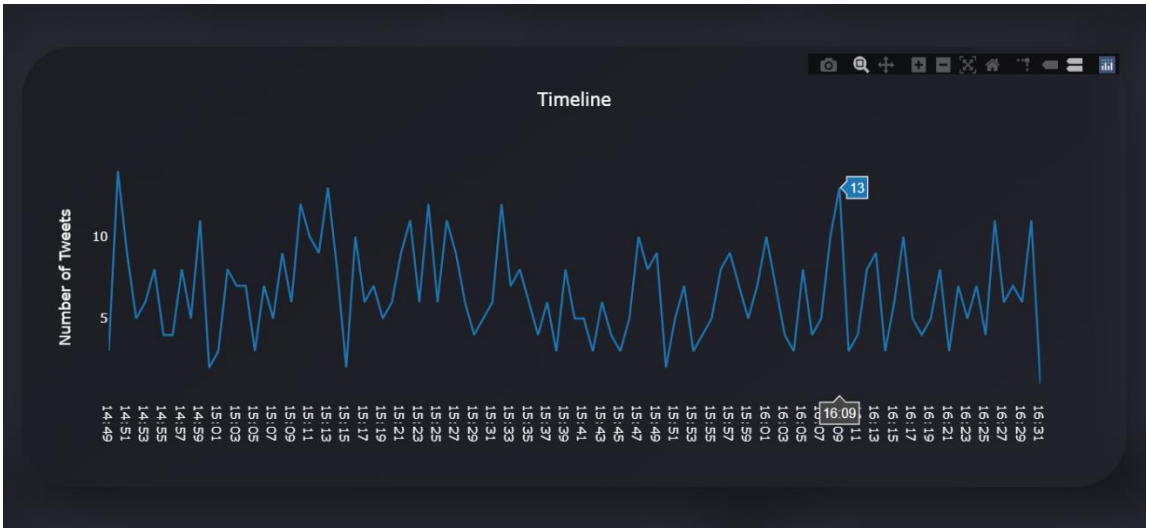

Figure 5.6 Timeline Graph

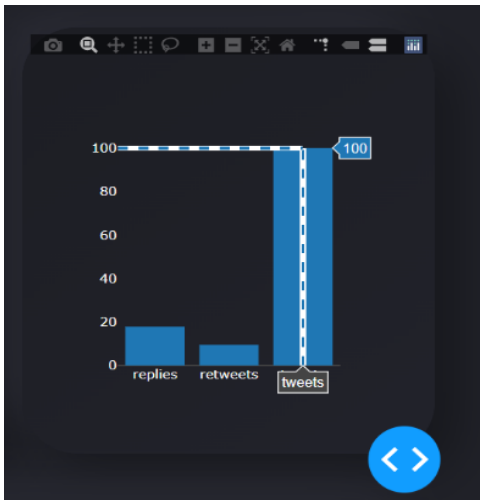

Figure 5.7 Tweet Distribution Graph

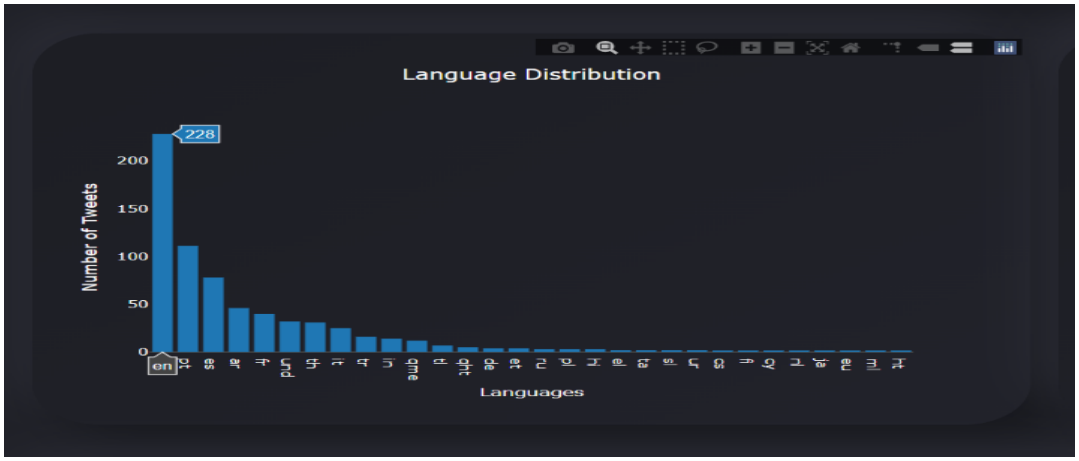

Figure 5.8 Language Distribution Plot

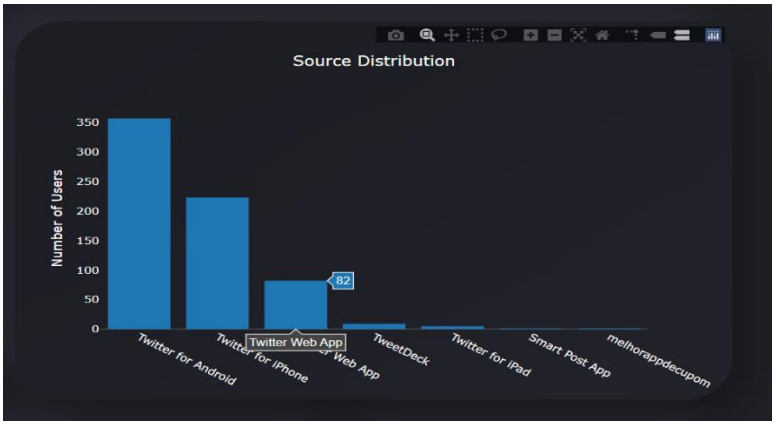

Figure 5.9 Source Distribution Plot

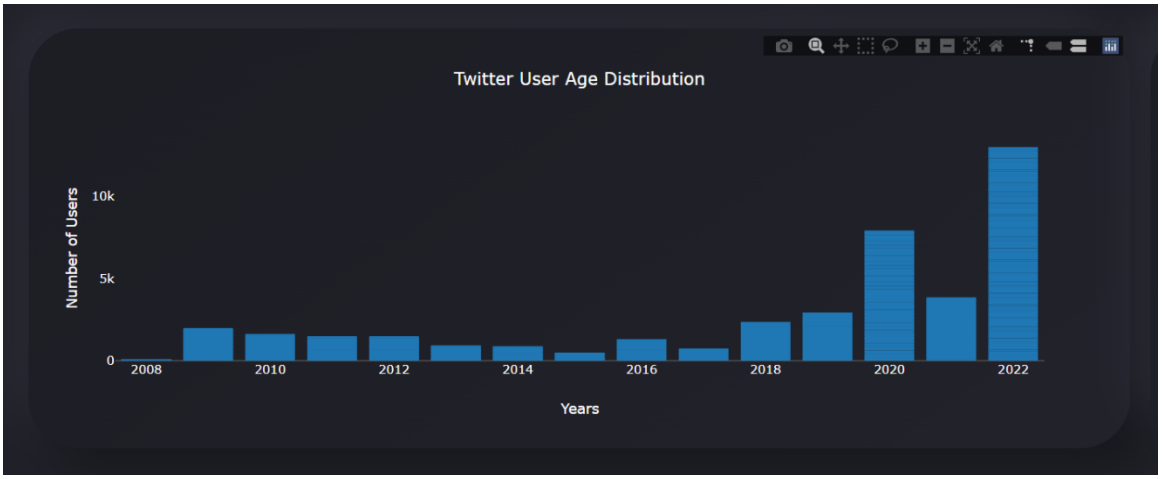

Figure 5.10 Twitter User Age Distribution Plot

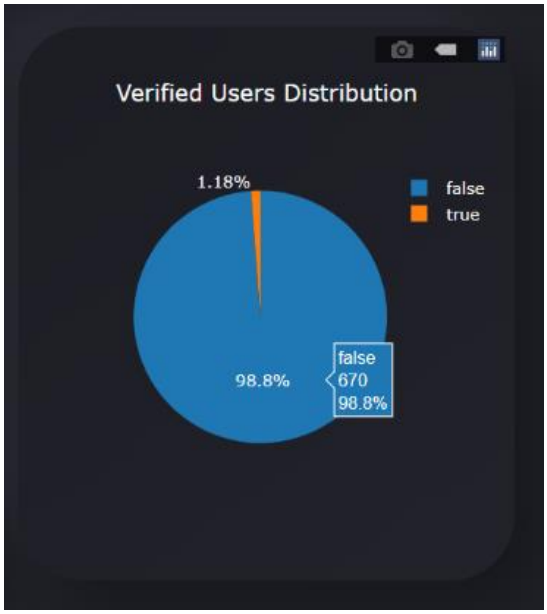

Figure 5.11 Verified User Distribution Chart

CHAPTER 6

CONCLUSION

We can draw conclusions on following parameters :

6.1 Features

This tracker as various features that are listed as follows :

- It allows Real-time Hashtag Tracking. It allows fetching the results in real time.
- Track any Twitter Hashtag, Keyword or @account mention in real-time and see the result on our online dashboard hence providing the multiple trackers facility.
- Find all the most influential Twitter users who are responsible to promote any Twitter Hashtag campaign
- Analyze all the activities of your competitors' to plan social media marketing strategy and track their main hashtags
- It provides Dashboard as User Interface which easier for the users to find all the charts in a presentable and easy to readable mode.
- Users can also download graphs For Free in the png format. This feature makes much easier for users to save their results.

6.2 Limitations

Although there are many features of this project, but it several limitations also. These limitations are mentioned as follows :

- **Only real time :** It shows the results of real time. It is unable to show historic analytics.
- **Limited choice of Graphs :** There are only limited number of predefined graphs available in the dashboard. It is not possible for a user to create a new graph or plot.

CHAPTER 7

FUTURE SCOPE

In this project we were able to fetch the results for hashtags, users, mentions, keywords, phrases. It is able to retrieve the data in real time. It can show the results in the form of numeric as well as graphic visualization form. Here real time refers to the 24 hour format day of that date.

But as we have observed it is unable to fetch the historic results. In future it can be expanded by including the feature to track the analytics for a period of time in past. It takes longer time for showing the results. This can also be handled by implementing more optimized methods of web scrapping.

There are only limited number of predefined graphs available in the dashboard. It is not possible for a user to create a new graph or plot. This can also be taken into consideration by giving freedom to users to chose layouts according to their choices. This project has a great scope of improvement and advancements.

REFERENCES

[1] M. Comesaña, A. P.Soares, M.Perea, A.P. Piñeiro, I. Fraga, and A. Pinheiro, " Author ' s personal copy Computers in Human Behavior ERP correlates of masked affective priming with emoticons," Computers in Human Behavior, 29, 588–595, 2013.

[2] A.H.Huang, D.C. Yen, & X. Zhang, "Exploring the effects of emoticons," Information & Management, 45(7), 466–473, 2008.

[3] D. Boyd, S. Golder, & G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on twitter," System Sciences (HICSS), 2010

[4] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5428313

[5] T. Carpenter, and T. Way, "Tracking Analysis through Twitter,". ACM computer survey. Villanova:VillanovaUniversity, 2010.

[6] D. Osimo, and F. Mureddu, "Research Challenge on Opinion Mining and Sentiment Analysis," Proceeding of the 12th conference of Fruct association, 2010, United Kingdom.

[7] A. Pak,and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," Special Issue of International Journal of Computer Application, France:Universitede Paris-Sud, 2010.

[8] S.Lohmann, M. Burch, H. Schmauder and D. Weiskopf, "Visual Analysis of Microblog Content Using Time-Varying Co-occurrence Highlighting in Tag Clouds," Annual conference of VISVISUS. Germany: University of Stuttgart, 2012.

[9] H. Saif, Y.He, and H. Alani, "SemanticSentimentAnalysisof Twitter," Proceeding of the Workshop on Information Extraction and Entity Analytics on Social Media Data. United Kingdom: Knowledge Media Institute, 2011.

[10] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R.Passonneau, "Sentiment Analysis of Twitter Data," Annual International Conferences. New York:Columbia University, 2012.