**DEPARTMENT OF INFORMATION TECHNOLOGY**

# Summer Internship Project File

# TITLE: Olympic Data Analyzer

CHANDIGARH ENGINEERING COLLEGE CGC, LANDRAN, MOHALI

Building Careers. Transforming Lives.

**Submitted By:**

**Name: Abhishek Kumar**

**Roll No: 2237786**

**B.Tech : 5th Semester**

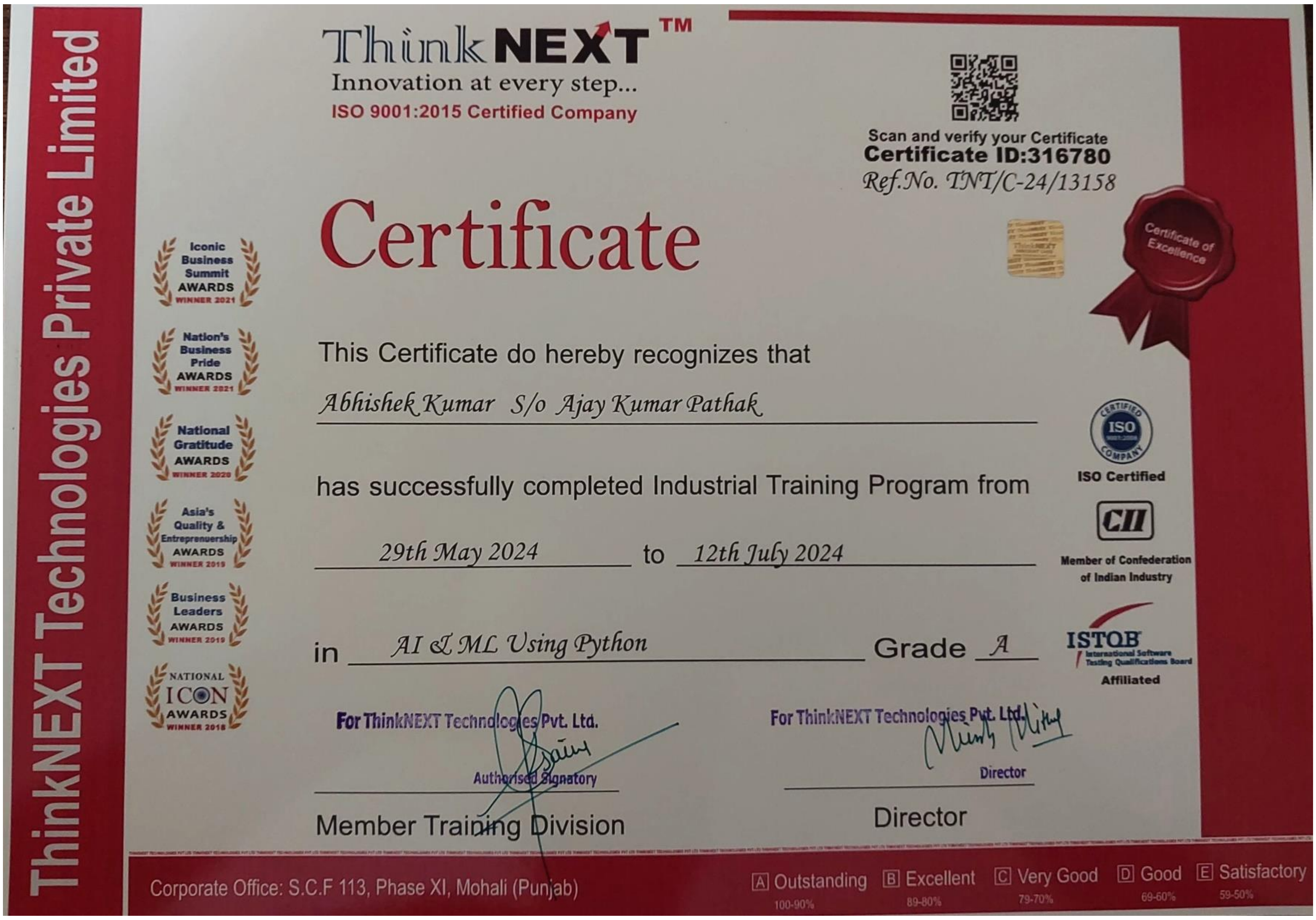
**Branch: AI & ML**

# CERTIFICATE

This is to certify that **Mr. Abhishek Kumar** has partially completed the Semester Training during the period from 28th May 2024 to 12th July 2024 in our Organization as a Partial Fulfillment of Degree of Bachelor of Technology in Artificial intelligence & Machine learning

(Signature of Project Supervisor)

Date: 12/07/2024



ThinkNEXT Technologies Private Limited

ThinkNEXT™
Innovation at every step...
ISO 9001:2015 Certified Company

Scan and verify your Certificate
**Certificate ID:316780**
*Ref.No. TNT/C-24/13158*

Certificate of Excellence

# Certificate

Iconic Business Summit AWARDS WINNER 2021

Nation's Business Pride AWARDS WINNER 2021

National Gratitude AWARDS WINNER 2020

Asia's Quality & Entrepreneurship AWARDS WINNER 2019

Business Leaders AWARDS WINNER 2019

NATIONAL ICON AWARDS WINNER 2018

This Certificate do hereby recognizes that

*Abhishek Kumar S/o Ajay Kumar Pathak*

has successfully completed Industrial Training Program from

*29th May 2024* to *12th July 2024*

in *AI & ML Using Python* Grade *A*

ISO Certified

CII
Member of Confederation of Indian Industry

ISTQB International Software Testing Qualifications Board
Affiliated

For ThinkNEXT Technologies Pvt. Ltd.
[illegible]
Authorised Signatory

Member Training Division

For ThinkNEXT Technologies Pvt. Ltd.
[illegible]
Director

Director

Corporate Office: S.C.F 113, Phase XI, Mohali (Punjab)

| | | | | |
|---|---|---|---|---|
| A Outstanding | B Excellent | C Very Good | D Good | E Satisfactory |
| 100-90% | 89-80% | 79-70% | 69-60% | 59-50% |

## CHANDIGARH ENGINEERING COLLEGE-CGC

## Department of Information Technology

## DECLARATION

I hereby certify that the work which is being presented in the report entitled "Olympic Data Analyzer" in partial fulfillment for the award of the Degree of Bachelor of Technology in Information Technology affiliated to Punjab Technical University, Jalandhar and submitted to the Department of Information Technology of Chandigarh Engineering College-CGC Landran, Mohali.

Date: 12/07/2024

**(Name of student)**

**(Abhishek Kumar)**

**(2237786)**

## ABSTRACT

This report examines the evolution of the Olympics from 1896 to 2016, with a focus on nations' performance and participation. The aim of this study is to analyze the dominant sports contributions, compare performance across sports, and identify factors that influence Olympic participation. The report concludes that increased nation and athlete participation, event numbers, and performance improvements have contributed to the growth of the Olympics, while costs, gender ratios, medical facilities, and pandemic impacts pose challenges to future participation. The implications of this study suggest that the International Olympic Committee and national Olympic committees should prioritize strategies to increase participation, improve performance, and address the challenges facing the Olympics. The findings of this report have significant implications for the future development of the Olympic Games and provide valuable insights for policymakers, athletes, and sports administrators.

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **ThinkNext Pvt. Ltd**., Mohali, Chandigarh, for providing me with the opportunity to undergo a **6-week summer training program in Python using AI and ML**. This training has been an invaluable experience, and I am grateful for the knowledge and skills I have acquired during my time at the organization.

I would like to extend my special thanks to my mentor, **Ms. Manpreet Kaur**, whose guidance, support, and encouragement have been instrumental in my learning process. Her expertise and patience have helped me to overcome various challenges and have made a significant impact on my understanding of the subject.

I would also like to thank the entire team at ThinkNext Pvt. Ltd. for their cooperation and assistance throughout my training period.

Lastly, I would like to acknowledge the support of my college, [Chandigarh Engineering college], for providing me with the opportunity to pursue this training and for their encouragement throughout my academic journey.

## ABOUT COMPANY

As a 5th semester student pursuing a Bachelor of Technology degree in Artificial Intelligence and Machine Learning (Batch 2022-2026), I had the opportunity to undertake a 6-week summer training program with **ThinkNEXT Technologies Private Limited, Mohali**.

**ThinkNEXT Technologies Private Limited is an ISO 9001:2015 certified company**, founded in 2011 and approved by the Ministry of Corporate Affairs, registered under the Companies Act 1956. The company is a leading IT solutions provider, offering a wide range of services including Digital Marketing, Web Designing, Web Development, Campus ERP Software for Universities/Colleges, eLearning Platform, Chatbots, Mobile Apps Development, Security Systems, Industrial Training, Internships, and Online Courses.

ThinkNEXT Technologies leverages the latest technologies such as Smart Card (Contact Type, Contactless), NFC, Biometrics, Barcode, RFID, and SMS to deliver innovative solutions to its clients. With a global presence, the company has established itself as a prominent player in the IT industry, particularly in the Chandigarh region.

Notably, ThinkNEXT Technologies is a Google Partner for Google Adwords, Bing Ads Accredited, Hubspot Certified, Facebook Blueprint Certified, and Microsoft Bing Ads Accredited Company, demonstrating its expertise in Digital Marketing, Web Development, and Industrial Training.

I am grateful to have had the opportunity to be a part of this esteemed organization and gain valuable experience in the field of Artificial Intelligence and Machine Learning

# TABLE OF CONTENT

Certificate....i-ii

Declaration....iii

Abstract....iv

Acknowledgement.... .v

Table of Contents....vi-viii

CHAPTER 1: INTRODUCTION....1-22

1.1 Introduction....1

1.2 Project Definition....1-2

1.3 Declaration of the problem....2

1.4 Project Purposes....3

1.5 Architecture & component....4-13

1.6 Project Scope ....13-14

1.7 Data Visualization....14-22

1.7.1 Type of Data Visualization....16-20

1.7.2 how does the Data Visualization work....20-22

1.8 Summary....22

CHAPTER 2: SURVEY OF TECHNOLOGY....23-32

2.1 Data Preprocessing and Data Visualization....23-24

2.2 Core Idea of Olympic data analysis....24-25

2.3 Visualization using programming....26-32

2.3.1 Matplotlib....26

2.3.2 Seaborn....26-27

2.3.3 Pandas……………………………………………………………………….…27-29

2.3.4 NumPy…………………………………………………………………….…29

2.3.5 Plotly……………………………………………………………………...29-30

2.3.6 Streamlit…………………………………………………………………..31-32

CHPTER 3: REQUIREMENTS AND ANALYSIS…………………………..….33-38

3.1Introduction……………………………………………………………...…….33

3.2 Software Requirement……………………………………………………...……33

3.3 Hardware Requirement…………………………………………………………..34

3.4 Data Requirement……………………………………………………...…...34

3.5 Planning and Scheduling………………………………………………......34-35

3.5.1 GANTT Chart…………………………………………………….….34

3.5.2 PERT Chart (Work Break Down)……………………………………….35

3.6 Conceptual Models (UML Diagrams)……………………………………..35-36

3.6.1 Structural View…………………………………………………..…...36

3.6.1.1 Class Diagram…………………………………………….……36

3.6.1.2 Component Diagram……………………………………..…...36

3.6.1.3 Object Diagram……………………………………………..…..36

3.6.2 Behavioral View……………………………………………………..36-38

3.6.2.1 Sequence Diagram……………………………………………...37

3.6.2.2 Activity Diagram………………………………………….……37

3.6.2.3 Use case Diagram……………………………………….….…37

3.6.2.4 Data Flow Diagram (DFD)……………………………….....37-38

CHAPTER 4: SYSTEM DESIGN…………………………………………....39-50

4.1 Code Brief Explanation…………………………………………..………39-50

CHAPTER 5: SYSTEM IMPLEMENTATION……………………………………...…..51-61

5.1 Screenshots of the system………………………………………………..……..…51-59

5.2 Definition and Goal of Testing…………………………………………………..…60

5.3 Method of Testing………………………………………………………………....60-61

CHAPTER 6: RESULTS……………………………………………………………...62-63

6.1 Test Case……………………………………………………………………………..62-63

CHAPTER 7: CONCLUSION AND FUTURE SCOPE………………………..…….64-65

7.1 Conclusion……………………………………………………………………….……64

7.2 Future Scope……………………………………………………………………....64-65

CHAPTER 8: REFERENCES…………………………………………………….…..66-67

# CHAPTER 1: INTRODUCTION

## 1.1 INTRODUCTION

The Olympic Games have been a global phenomenon for over a century, bringing together athletes From around the world to compete in various sports and events. The games have a rich history and have served as a platform for countries to showcase their athletic prowess and cultural heritage.

As part of a college project, a web application has been created that analyzes Olympic data using the Python programming language. The application allows users to interact with the data and gain insights into various aspects of the games, including countries, sports, events, and medal counts. Using various Python libraries for data analysis and visualization, the application provides users with a range of tools for exploring Olympic trends and statistics.

The project demonstrates the power of Python for creating data analysis applications that can be used for various purposes, including sports analysis, business intelligence, and scientific research. The application provides a user-friendly interface that can be easily used by anyone, regardless of their technical expertise. Overall, this project highlights the potential of data analysis and visualization for gaining insights into complex datasets, such as the Olympic data, and showcases the value of using Python as a tool for creating data-driven applications.

## 1.2 PROJECT DEFINITION

The Olympic data analysis web application project is a software application that allows users to interactively analyze and visualize various aspects of the Olympic Games, such as countries, sports, events, and medal counts. The application is created using Python programming language and uses various Python libraries for data analysis and visualization. The data is obtained from a publicly available Olympic database and is presented in a user-friendly interface that enables users to gain insights into Olympic trends and statistics.

Primary Goal:

The primary goal of the Olympic data analysis web application project is to create a user-friendly and interactive platform that allows users to explore various aspects of the Olympic Games through data analysis and visualization. By using Python and various Python libraries for data analysis and visualization, the project aims to provide users with a range of tools for exploring Olympic trends and statistics, and to enable them to gain insights into the games that are not immediately apparent. The project is intended to be used for various purposes, including sports analysis, business intelligence, and scientific research. Ultimately, the project aims to demonstrate the potential of Python for creating data-driven applications that can be used for a wide range of purposes.

## 1.3 DECLARATION OF THE PROBLEM

The Olympic Games have a rich history and have served as a platform for countries to showcase their athletic prowess and cultural heritage. However, analyzing and interpreting the vast amount of data associated with the games can be challenging. Traditional methods of data analysis, such as spreadsheets, can be time-consuming and may not provide insights into the data that are not immediately apparent.

To address this problem, the Olympic data analysis web application project was developed. The project aims to create a user-friendly platform that enables users to interactively analyze and visualize various aspects of the Olympic Games. By using Python and various Python libraries for data analysis and visualization, the project aims to provide users with a range of tools for exploring Olympic trends and statistics, and to enable them to gain insights into the games that are not immediately apparent.

Overall, the problem that this project aims to address is the need for a user-friendly and interactive platform that enables users to explore Olympic data in a meaningful way. The project is intended to be used for various purposes, including sports analysis, business intelligence, and scientific research.

## 1.4 PROJECT PURPOSES

The purpose of the Olympic data analysis web application project is to provide users with a user- friendly and interactive platform that enables them to explore various aspects of the Olympic Games through data analysis and visualization. By using Python, Machine Learning, and Artificial Intelligence methods and approaches, and various Python libraries such as Pandas, Preprocessor, Matplotlib, Seaborn, and Plotly, the project aims to provide users with a range of tools for exploring Olympic trends and statistics, and to enable them to gain insights into the games that are not immediately apparent.

The project's primary goal is to create a web application that can be used for various purposes, including sports analysis, business intelligence, and scientific research. Users can interact with the data and analyze it according to various criteria, such as countries, sports, events, and medal counts. The application is designed to be user-friendly, with a simple interface that allows users to easily explore and analyze the data.

The project's ultimate purpose is to demonstrate the potential of Python, Machine Learning, and Artificial Intelligence for creating data-driven applications that can be used for a wide range of purposes. The project aims to provide users with a platform that enables them to explore and gain insights into complex datasets such as the Olympic data, and to showcase the value of using these technologies for data analysis and visualization.

## 1.5 ARCHITECTURE & COMPONENTS

The Olympic data analysis web application project is built using the Python programming language and various Python libraries for data analysis, visualization, and web development. The project's architecture consists of three main components

Architecture

1. Data Preprocessing: The first component of the project is data preprocessing, which involves cleaning and preparing the Olympic dataset for analysis. The project uses the Pandas and Preprocessor libraries for these purposes, which enable data cleaning, transformation, and aggregation. The preprocessing step also involves removing any missing or duplicate data, normalizing the data, and creating new features or variables based on the original data.

2. Data Visualization: The second component of the project is data visualization, which involves creating interactive visualizations and plots that enable users to explore and analyze the data. The project uses the Matplotlib, Seaborn, and Plotly libraries for these purposes, which provide a range of visualization tools for creating line charts, bar charts, scatter plots, heatmaps, and more.

3. Web Application: The third component of the project is the web application, which provides a user-friendly interface for users to interact with the data and visualize it. The project uses the Streamlit library for web development, which enables the creation of interactive web applications using Python. The web application component consists of multiple pages that provide various features such as data filtering, sorting, searching, and visualization. The application is deployed on the Streamlit Cloud, a free hosting service for Streamlit applications.

Components :-

The components of the Olympic data analysis web application project include:

1. Data Preprocessing :-

- Pandas Library
- Preprocessor Library

2. Data Visualization :-

- Matplotlib Library
- Seaborn Library
- Plotly Library

3. Web Application :-

- Streamlit Library
- HTML/CSS/JavaScript
- Streamlit Cloud (for deployment)

The project's architecture and components work together to enable users to interactively explore and analyze the vast amount of data associated with the Olympic Games, providing them with a range of tools and insights that may not be immediately apparent from the raw data.

Why phython :-

In the ever-evolving landscape of programming languages, one name stands out as a versatile and powerful tool for developers across various domains: Python. As we delve into the reasons why Python continues to capture the hearts of programmers, it becomes clear that its appeal goes beyond just syntax. Here are some compelling reasons why Python reigns supreme over other programming languages.

Readability and Simplicity :-

Python’s syntax is clean, readable, and emphasizes code readability, making it an ideal language for both beginners and experienced developers. Its straightforward and easy-to-understand code structure allows for faster development and maintenance of applications.

Versatility and Flexibility :-

Python’s versatility is a key factor in its widespread adoption. From web development and data analysis to artificial intelligence and machine learning, Python seamlessly integrates into various domains. Its flexibility enables developers to switch between different programming paradigms, making it suitable for a wide range of projects.

Extensive Libraries and Frameworks :-

Python boasts an extensive collection of libraries and frameworks that significantly accelerate development cycles. Popular libraries such as NumPy, Pandas, and TensorFlow empower developers to efficiently tackle complex tasks in data science, machine learning, and more, reducing the need to reinvent the wheel.

Community Support and Documentation :-

The Python community is one of the most vibrant and supportive in the programming world. A vast number of developers actively contribute to the language’s growth, creating a rich ecosystem of resources and documentation. This wealth of knowledge ensures that developers can easily find solutions to problems and stay updated on best practices.

Cross-Platform Compatibility :-

Python’s cross-platform compatibility allows developers to write code that runs seamlessly on various operating systems. This feature not only saves time but also enhances the portability of applications, ensuring a consistent user experience across different platforms.

Rapid Prototyping and Development :-

Python’s concise syntax and dynamic typing enable developers to prototype and iterate quickly. This rapid development cycle is particularly valuable in the fast-paced tech industry, allowing teams to bring ideas to life efficiently and adapt to changing requirements.

Community-Driven Innovation :-

The open-source nature of Python fosters a culture of innovation. The continuous development and improvement of the language through community contributions ensure that Python stays relevant and at the forefront of technological advancements.

History :-

Python was created by Guido van Rossum, and first released on February 20, 1991. While you may know the python as a large snake, the name of the Python programming language comes from an old BBC television comedy sketch series called Monty Python’s Flying Circus.

One of the amazing features of Python is the fact that it is actually one person’s work. Usually, new programming languages are developed and published by large companies employing lots of professionals, and due to copyright rules, it is very hard to name any of the people involved in the project. Python is an exception.

Of course, Guido van Rossum did not develop and evolve all the Python components himself. The speed with which Python has spread around the world is a result of the continuous work of thousands (very often anonymous) programmers, testers, users (many of them aren’t IT specialists) and enthusiasts, but it must be said that the very first idea (the seed from which Python sprouted) came to one head – Guido’s.

Python is maintained by the Python Software Foundation, a non-profit membership organization and a community devoted to developing, improving, expanding, and popularizing the Python language and its environment.

Python is the programming language that opens more doors than any other. With a solid knowledge of Python, you can work in a multitude of jobs and a multitude of industries. And even if you don’t need it for work, you will still find it useful to know to speed certain things up or develop a deeper understanding of other concepts.

Python is a great choice for career paths related to software development, engineering, DevOps, machine learning, data analytics, web development, and testing. What's more, there are also many jobs outside the IT industry that use Python. Since our lives are becoming more computerized every day, and the computer and technology areas previously associated only with technically gifted people are now opening up to non-programmers, Python has become one of the must-have tools in the toolbox of educators, managers, data scientists, data analysts, economists, psychologists, artists, and even secretaries.

Evolution of Python :-

The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released, it had more than enough capability to provide classes with inheritance, several core data types of exception handling and functions.

Following are the illustrations of different versions of Python along with the timeline.

Python 3.12.1 is the latest stable version.

The two of the most used versions has to Python 2.x & 3.x. There is a lot of competition between the two and both of them seem to have quite a number of different fanbases. You can also refer to mentioned below article to learn How you can download python latest version in your system.

Literature review on WhatsApp Chat Analysis: A survey analysis on the usage and impact of WhatsApp Messenger [1] has been conducted and various studies and analysis have been found. These studies include the impact of WhatsApp on the students(youth). In the survey it was found that in the southern part of India, ages 18 to 23 spend around 8 hours using whatsapp and sometimes be online almost 12-16 hours a day. Most of them agreed to be using whatsapp tan any other site. They exchange images, audios and videos. This survey also proved that the whatsapp has been the most widely used app on the smart phones than any other app. This survey was conducted to know the positive and negative impacts of using whatsapp. As we can know that from this survey, whatsapp is most used app by the youth and other generations so, our project can give them the insights of their chats and provide them unknown facts.

Literature review on Modules :-

a .Streamlit: Streamlit is a free and open-source python framework. [2] We can quickly develop web apps for Machine Learning and Data Science by using Streamlit. Streamlit can easily integrates with other popular python packages such as NumPy, Pandas, Matplotlib, Seaborn. Streamlit provides fastest way to develop and deploy web apps.

b .Matplotlib: Matplotlib is a popular Python packages used for data visualization. It is a cross-platform library for making plots from data in arrays. It helps in creating static, animated and interactive visualizations in python.

c .Seaborn: Seaborn is the data visualization library. It is used for making statistical graphs. Visualization is the central part of seaborn. Seaborn provides exploration and better understanding of data. Seaborn closely integrates into the data structures from python.

d. Word cloud: Word Cloud is a data visualization library used for representing most frequently used words within a given text. Most frequent and important words are represented in bigger and bolder size

e. Pandas: Pandas is an open-source python library. Pandas used to convert string data into Data frame. Data frame is the representation of data into 2-dimensional table of rows and columns. We can work with large data sets using Pandas library. Pandas library has many built-in functions for data analysis, data cleaning, data exploration and data manipulation  In 2008, developer Wes McKinney started developing pandas because he needed a high performance, flexible tool for analysis of data. 2.3 Literature review on Natural Processing Language: This research gathered all scientific publications in urban studies that utilized the method of NLP.[3]To conduct this research we have taken the journals and conference papers from databases EBSCO Urban Studies Abstracts, Scopus, ProQuest, and Web of Science.

Literature review on python: Python is a general-purpose language .It has an easily understandable syntax. Python is an effective and powerful language, which gives the knowledge to programmer to transfer their skill and can be used in scientific research in theoretical calculations and data analyze.[4] It is statistics oriented and it has specific advantages such as great features for data visualization. Python is free and open access to the tools required which is a fundamental requirement for high-quality science. Unlike MATLAB or LabView python can be used for any programming task. Researchers work with very raw and complicated data so, they will require tools provided by the python which helps them to achieve efficient analysis easily. Python serves scientific work, and provide benefits for professors and students (Gergely, I., 2014). Tony, J. (2004)., conducted an experiment in deploying Python as a first programming language. Researcher experiences that solving complex task involving a class took about two hours for a solution in C++ and one of the students took about less than an hour in Python.

Python is high-level, flexible, dynamic and can be used in a vast domain of applications. Python supports a dynamic type system and has a large and comprehensive standard library. (Srinath, K.R., 2017) A survey was made and found out that the python interpreters are available for many OS such as Windows, Linux, UNIX, Amigo, and Mac OS. Literature review on Web Design: Internet Users are reaching millions and can be expected to increase more over the years. The websites are the crucial media of information, transmission, dissemination.[5] Current paper purposes to review previous studies that have been done in the field of web development. As the result, literatures either proposed set of guidelines or assistive technologies particularly web interfaces, adaptive systems. The acceptance and success of the websites and electronic commerce depends on the web design. The purpose of this paper is to analyse and know the users' perceptions and behaviors, in order to achieve a successful e-commerce website. According to a survey(Lee & Kozar, 2012) there is currently no consensus on how to properly operationalize and assess website usability. Nielson associate's usability with learnability, efficiency, memorability, errors, and satisfaction (Nielsen, 2012). Right now we do not have any guidelines that individuals can follow when designing websites to increase users engagement. • "Hypertext" are the links to connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web, by uploading content to the Internet and linking it to pages. • HTML uses "markup" to annotate text, images, and other content for display in a Web browser to describe the presentation of a document written in HTML or XML. • CSS is the core languages of the open web, standardized across Web browsers according to W3C specifications. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media means like the styling part of the webpage.

Data Analysis :-

It is a process of cleaning, transforming, inspecting and modelling data with the goal of discovering some useful information and finally indicating some conclusions. Analysis means it breaks a whole component into its separate components for individual examination. Data analysis is a process for acquiring raw data and transforming it into useful information for decision-making by users. This project provides a basic statistical analysis WhatsApp chat. Following are the analysis made : • To find total messages, total words, total media and links shared in the WhatsApp chat • To find the most active people in the group. • To find the most used emojis in the group. • To find the busiest day and least busy in a month. • To find the most frequently and commonly used words in the group. • To find the frequency of chat in every day and month.

Proposed System :-

Data pre-processing is the initial part of the project, it is to understand the implementation and usage of various python inbuilt modules. These various modules provide better user understandability and code representation. The following libraries are used such as NumPy, pandas, matplotlib, sys, re, emoji, seaborn etc. It analyses the data and gives top statistics like total messages, total media, links, images shared, graphs showing the activity map weekly and monthly, monthly timeline, daily timeline, mostly busy users, chart most common words used, emojis used.

Working :-

Steps to Export chat:

Open WhatsApp chat for a group ->click on the menu ->click on more- ->select export chat->choose without media.

Working of WhatsApp chat analysis.

1. Intially open WhatsApp chat analyzer web page.

2. Select Date format.

3. Upload the exported chat file.

4. Analyzing of data is done by trained model

5. Preprocessing of data is done by trained model.

6. Select overall or single person analysis

7. Trained model shows analysis it includes top statistics, word cloud, activity map, monthly timeline, daily timeline, emoji analysis.

System Modules :-

(a) Install and import dependencies: In this step Streamlit, matplotlib, pandas, collections, seaborn, emoji, Wordcloud, URLextract, and re are installed and imported. (b) Pre-processing: In this step pre-processing of the data is done. Here the data is formatted and separated in the form of date, time, name of the user and message of the use.

(b) Export chat document from WhatsApp and Upload: Here the document is exported from WhatsApp. Steps to export chat ->Open individual or Group chat->Tap Options – More – Export Chat->Choose export without media-> Document is set. Upload the chat file and click on analysis

(c) Train chat model and analyze the data: Here the collected data is read and processed to train our machine learning classification model on it.

The model is then evaluated and serialized. Analysis made:

1 .Top Statistics: These involve total messages, total words, media shared, links shared.

2 . Monthly Timeline: The frequency of chat in every month.

3 .Daily Timeline: The frequency of chat in a day

4 . Activity Map: Shows the busiest day and least busy day similarly with the month

5 .Weekly Activity map.

6 .Wordcloud: Most commonly and frequently used word.

7 .Most Busy Users: Mostly active people.

8 .Emoji analysis: Most commonly and frequently used emojis.

(d) Make detections with model: Running the code, predictions of the user's gestures using the above trained model are made.

Testing :-

Software testing is like an investigation conducted to know about the quality of the product under test. Software testing provides an objective view of the software to allow the developers to understand the risks of software implementation. The test techniques include the process of executing a program or application in order to find some software bugs or some defects in it.

## 1.6 PROJECT SCOPE

The scope of the Olympic data analysis web application project is to provide users with a user-friendly and interactive web application that allows them to explore and analyze 120 years of Olympic data. The project aims to enable users to gain insights into patterns and

Trends in Olympic history, including athlete performance, medal counts and event participation across different countries and time periods.

The project includes a range of features that enable users to interact with the data, including data filtering, sorting, searching, and visualization. The project also aims to provide users with a range of visualizations that enable them to explore the data in different ways, including line charts, bar charts, scatter plots, heat maps, and more.

The project is intended for use by a wide range of users, including students, researchers, and sports enthusiasts, and is designed to be accessible and informative for users with varying levels of expertise in data analysis and visualization. The project is hosted on the Streamlit Cloud, which provides users with easy access to the application and ensures that the application is always up-to-date with the latest data.

Overall, the scope of the Olympic data analysis web application project is to provide users with a comprehensive and informative tool for exploring and analyzing Olympic data, enabling them to gain insights into one of the world's most iconic sporting events.

## 1.7 Data Visualization

Data visualization is an important aspect of the Olympic data analysis web application project, as it enables users to explore and understand patterns and trends in Olympic history in an intuitive and interactive way. The project uses several popular Python libraries for data visualization, including Matplotlib, Seaborn, and Plotly, each of which provides different tools for creating visualizations that can be embedded within the web application.

The project includes a wide range of visualizations that enable users to explore different aspects of Olympic history, including athlete performance, medal counts, and event participation across different countries and time periods. These visualizations include:

1. Line charts: These are used to show trends in athlete performance over time, such as the number of medals won by a particular country in different Olympic games.
2. Bar charts: These are used to compare data across different categories, such as the number of gold, silver, and bronze medals won by different countries.
3. Scatter plots: These are used to show the relationship between two variables, such as the number of athletes from a particular country and their performance in different events.

4. Heat maps: These are used to show the distribution of data across different categories, such as the number of medals won by different countries across different events.

The visualizations in the project are highly interactive, enabling users to filter and explore the data in different ways. For example, users can filter the data by year, country, event, or athlete, or zoom in on specific parts of a chart to explore the data in more detail. The project also includes several advanced visualizations, such as choropleth maps and interactive networks, that enable users to explore the data in even more depth.

Overall, data visualization plays a crucial role in the Olympic data analysis web application project, enabling users to explore and understand patterns and trends in Olympic history in a highly interactive and engaging way.

Matplotlib:-

Matplotlib is an easy-to-use, low-level data visualization library that is built on NumPy arrays. It consists of various plots like scatter plot, line plot, histogram, etc. Matplotlib provides a lot of flexibility.

To install this type the below command in the terminal.

```
pip install matplotlib
```

Scatter Plot :-

Scatter plots are used to observe relationships between variables and uses dots to represent the relationship between them. The scatter() method in the matplotlib library is used to draw a scatter plot. This graph can be more meaningful if we can add colors and also change the size of the points. We can do this by using the c and s parameter respectively of the scatter function. We can also show the color bar using the colorbar() method.

Line Chart :-

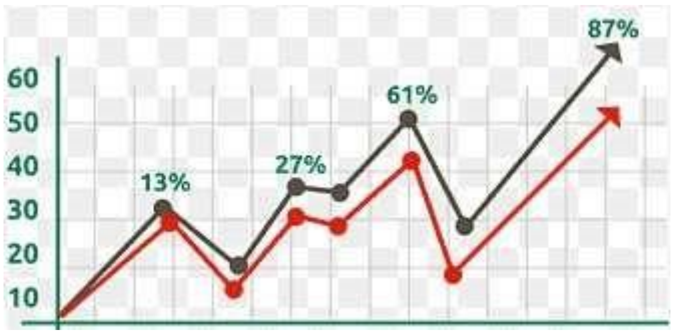
Line Chart is used to represent a relationship between two data X and Y on a different axis. It is plotted using the plot() function.

Bar Chart :-

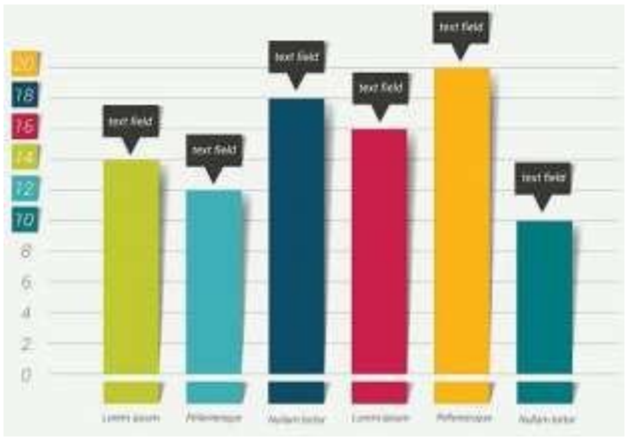
A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. It can be created using the bar() method.

### 1.7.1 Type of Data Visualization

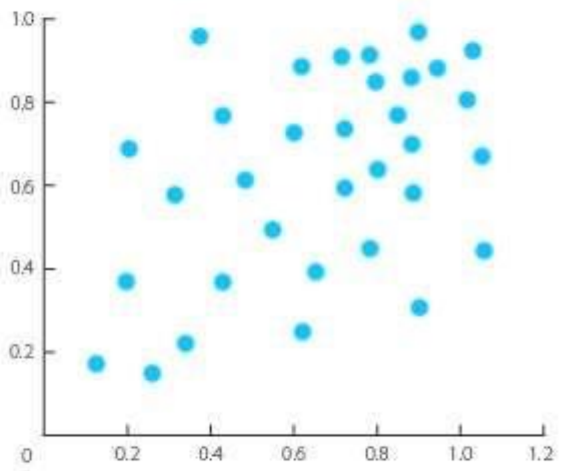
Line Charts: Line charts are used to show trends over time. In the Olympic data analysis web application project, line charts are used to show trends in athlete performance over time, such as the number of medals won by a particular country in different Olympic Games. These charts are useful for identifying patterns and trends in data over time.



Bar Charts: Bar charts are used to compare data across different categories. In the Olympic data Analysis web application project, bar charts are used to compare the number of medals won by different countries or the number of medals won in different events. These charts are useful for comparing data across different categories.



Scatter Plots: Scatter plots are used to show the relationship between two variables. In the Olympic data analysis web application project, scatter plots are used to show the relationship between the number of athletes from a particular country and their performance in different events. These plots are useful for identifying correlations and relationships between two variables.



Heatmaps: Heatmaps are used to show the distribution of data across different categories. In the Olympic data analysis web application project, heatmaps are used to show the number of medals won by different countries across different events. These charts are useful for identifying patterns in data across different categories. Heatmaps organize data in a grid, with different colors or shades indicating different levels of the data's magnitude.

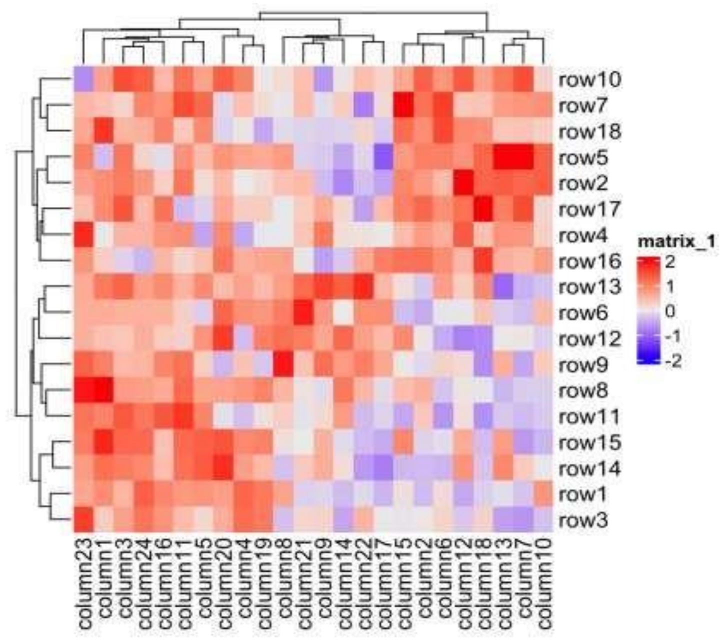
The visual nature of heatmaps allows for immediate recognition of patterns, such as clusters, trends, and anomalies. This makes heatmaps an effective tool for exploratory data analysis.

Choosing to use a heatmap depends on your requirements and the nature of your dataset. Generally, heatmaps are best suited for datasets where you can represent values as colors, typically continuous or discrete numerical data. However, you can also use them for categorical data that has been quantified or summarized (e.g., counts, averages).

If the dataset contains extreme outliers or is very sparse, a heatmap might not be as effective without preprocessing or normalization. Text, images, and other forms of unstructured data are also not directly suitable for heatmaps unless you first transform the data into a structured, numerical format.

Heatmaps excel at visualizing the correlation matrix between multiple variables, making it easy to identify highly correlated or inversely correlated variables at a glance.

Heatmaps are also useful for visually comparing data across two dimensions, such as different time periods or categories. For geographical data analysis, heatmaps can represent the density or intensity of events across a spatial layout, such as population density or crime hotspots in a city.



Choropleth Maps: Choropleth maps are used to show data across geographical regions. In the Olympic data analysis web application project, choropleth maps are used to show the number of medals won by different countries across the world. These maps are useful for visualizing data across geographical regions.

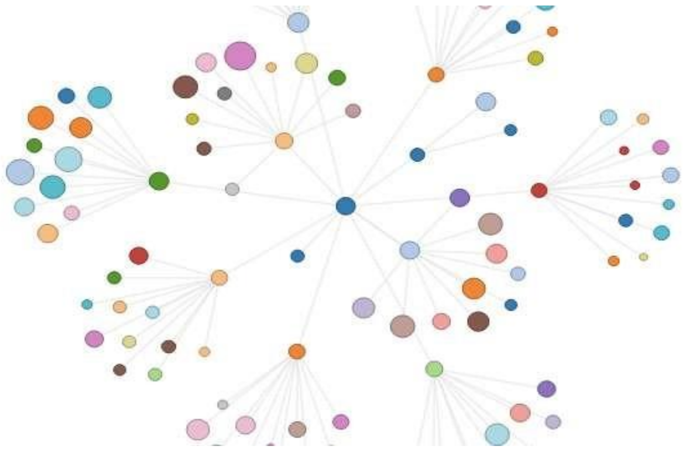
A Choropleth Map is a map composed of colored polygons. It is used to represent spatial variations of a quantity. This page documents how to build outline choropleth maps, but you can also build choropleth tile maps using our Mapbox trace types.

Below we show how to create Choropleth Maps using either Plotly Express' px.choropleth function or the lower-level go.Choropleth graph object.

Base Map Configuration :-

Plotly figures made with Plotly Express px.scatter_geo, px.line_geo or px.choropleth functions or containing go.Choropleth or go.Scattergeo graph objects have a go.layout.Geo object which can be used to control the appearance of the base map onto which data is plotted.

Interactive Networks: Interactive networks are used to show the relationships between different entities in a network. In the Olympic data analysis web application project, interactive networks are used to show the relationships between different countries based on the number of medals they have won. These networks are useful for visualizing complex relationships between different entities.



Overall, the Olympic data analysis web application project uses a wide range of data visualizations to enable users to explore and understand patterns and trends in Olympic history. Each type of visualization is used to highlight different aspects of the data and provide users with a comprehensive view of the data.

### 1.7.2 How does this Data Visualization work

In your Olympic data analysis web application project, data visualization is used to represent the large and complex Olympic dataset in a visually appealing and easily understandable way. The process of data visualization involves transforming data into charts, graphs, and other visual elements that can be easily interpreted by the user.

The data visualization in your project is done using various Python libraries such as Matplotlib, Seaborn, and Plotly. These libraries allow you to create various types of visualizations such as line charts, bar charts, scatter plots, heatmaps, and more. The data visualization process starts by pre-processing the raw Olympic dataset using the Pandas library, which involves cleaning and transforming the data into a format that can be used for visualization. The pre-processed data is then passed to the data visualization libraries, which create the desired visualizations based on the data.

The data visualizations in your project are interactive, which means that the user can interact with the visualizations and change the displayed data based on their preferences. For example, the user can filter the data by Olympic year, sport, country, and other criteria to see the data in different ways.

Overall, data visualization is a crucial part of your Olympic data analysis web application project, as it helps users to understand and explore the Olympic dataset more easily and efficiently.

1. Making complex data easy to understand: The Olympic dataset contains a large amount of data that can be difficult to comprehend in its raw form. However, data visualization techniques such as charts, graphs, and maps can help to break down this data into smaller, more digestible chunks that are easier to understand.

2. Providing visual context: Data visualization can provide visual context for the data, allowing users to see trends, patterns, and relationships between different variables. For

example, a scatter plot can show the relationship between two variables, such as a country's medal count and its population size.

3. Enabling interactivity: The data visualizations in your project are interactive, meaning that users can manipulate and explore the data in real-time. For example, users can filter the data by year, sport, or country to see how different variables affect the results.

4. Improving data-driven decision making: By presenting data in a visually appealing way, data visualization can help users to make more informed decisions based on the data. For example, a user might use a line chart to compare the medal counts of two different countries over time, helping them to make a more informed decision about which country to support.

data visualization is a powerful tool in your Olympic data analysis web application project that can help users to explore and understand the Olympic dataset more easily and efficiently. By breaking down complex data into smaller, more digestible chunks and providing visual context and interactivity, data visualization can help users to make more informed decisions based on the data.

## 1.8 SUMMARY

Data visualization in your Olympic data analysis web application project involves using various Python libraries such as Matplotlib, Seaborn, and Plotly to transform complex Olympic dataset into charts, graphs, and other visual elements that can be easily understood by users. The pre-processed data is passed to the data visualization libraries, which create interactive and visually appealing charts, graphs, and maps. Users can manipulate and filter the data in real-time to explore the data more efficiently. Data visualization helps to make complex data easy to understand, provides visual context, enables interactivity, and improves data-driven decision making.

# CHAPTER 2: SURVEY OF TECHNOLOGY

## 2.1 Data Preprocessing and Data Visualization

Data Preprocessing :-

Data preprocessing is a crucial step in data analysis, which involves cleaning, transforming, and organizing raw data into a format suitable for analysis. In other words, it is the process of converting raw data into a more meaningful and usable format that can be easily analyzed by machines or humans. Data preprocessing typically involves several steps, including removing duplicate or irrelevant data, Filling in missing data, normalizing data, and transforming data to a suitable format. The aim of data preprocessing is to improve the quality and accuracy of data, reduce errors, and enhance the overall efficiency and effectiveness of data analysis. data preprocessing is a vital step in data analysis that helps to ensure that the data is accurate, reliable, and suitable for analysis, which is essential for making informed decisions and drawing meaningful insights from the data.

1.The project reads in the 120 years of Olympic history dataset from Kaggle using the Pandas library.

2. The dataset is then cleaned and preprocessed using various Pandas methods, such as dropping null values, renaming columns, and aggregating data.

3. The preprocessed data is then stored in Pandas data frames for further analysis and visualization.

4. The Scikit-learn library is used for feature scaling, normalization, and machine learning models

Data Visualization :-

Data visualization is the process of representing data graphically and visually, using charts, graphs, and other visual elements. The primary goal of data visualization is to convey complex data and information in a simple and easy-to-understand format that can be easily interpreted by humans.

Through data visualization, patterns, trends, and relationships in the data can be identified more easily, allowing analysts to draw insights and make informed decisions based on the data. Data visualization is often used to present large and complex datasets in a simplified and meaningful way, making it easier for users to understand and draw conclusions from the data.

Some common types of data visualizations include bar charts, line graphs, scatterplots, heat maps, and pie charts. In recent years, there has been a growing trend towards interactive data visualization, which allows users to explore and interact with the data in real-time, providing a more engaging and personalized experience. data visualization is a crucial tool in data analysis and helps to communicate complex data in a way that is more accessible and easier to understand.

1.The preprocessed data frames are used to create various types of visualizations using Matplotlib, Seaborn, and Plotly libraries.

2.The Streamlit library is used to create an interactive web application for users to explore and interact with the visualizations.

3.The visualizations are organized into different sections, such as medal tally, overall analysis, country analysis, and athlete analysis, which can be accessed by the user through the web application interface

## 2.2 Core Idea of Olympic Data Analysis

The core idea of Olympic Data Analysis is to analyze and gain insights from the 120 years of Olympic history and data. The aim is to explore the patterns, trends, and relationships in the data, and to uncover interesting insights and findings about the Olympics, its participants, and its impact on society.

The Olympic Data Analysis project involves using various data analysis and machine learning techniques to preprocess, clean, and analyze the data, and to extract meaningful insights from it. The project also involves developing a user interface that allows users to interact with the data and explore it in an engaging and interactive way

1. Data collection: The project uses the Olympic Games dataset available on Kaggle, which contains data on the Olympic Games from 1896 to 2016. The data includes information on athletes, events, medals, countries, and more.

2. Data preprocessing: The raw data from Kaggle requires preprocessing to make it suitable for analysis. The project uses pandas library to preprocess the data. The preprocessing includes handling missing data, converting data types, merging data, and cleaning data.

3. Data visualization: Once the data is preprocessed, the project uses various data visualization libraries such as Matplotlib, Seaborn, and Plotly to create visualizations that provide insights into the data. The visualizations include bar charts, line graphs, scatter plots, heatmaps, and more.

4. Machine learning: The project also uses machine learning techniques to analyze the data and make predictions. For example, the project uses a recommendation system to suggest similar athletes or countries based on their performance or attributes.

5. User interface: The project uses Streamlit, a popular Python library, to create a user-friendly interface for users to interact with the data. The interface includes several sections such as Medal Tally, Overall Analysis, Country Analysis, and Athlete Analysis, allowing users to explore the data in different ways.

## 2.3 Visualization Using Programming

In Python, we can perform data visualization using various libraries such as Matplotlib, Seaborn, Plotly, Bokeh, etc. Here is a general overview of how data visualization can be performed using Matplotlib and Seaborn libraries:

### 2.3.1. Matplotlib :-

Matplotlib is a popular data visualization library in Python. It provides a variety of customizable plots such as line plots, scatter plots, bar plots, histograms, etc. Here are the basic steps to create a plot using Matplotlib:

- Import the Matplotlib library
- Prepare the data
- Create a figure and axis object using plt.subplots()
- Plot the data using the appropriate plot function such as plot(), scatter(), bar(), etc. - Customize the plot by adding titles, labels, legends, colors, etc.

- Save or display the plot using plt.savefig() or plt.show().

### 2.3.2. Seaborn :-

Seaborn is another data visualization library in Python that is built on top of Matplotlib. It provides a higher-level interface for creating statistical graphics such as heatmaps, violin plots, box plots, etc. Here are the basic steps to create a plot using Seaborn:

Import the Seaborn library

Load the data using Pandas or Numpy

Create a plot using the appropriate function such as sns.lineplot(), sns.scatterplot(), sns.boxplot(), - Customize the plot using the various built-in options such as hue, style, size, palette, etc. - Save or display the plot using plt.savefig() or plt.show().

1. High-Level Interface: Seaborn provides a simple and intuitive interface for creating complex statistical plots with minimal code, allowing users to focus on data analysis rather than plot customization.

2. Attractive Aesthetics: Seaborn comes with built-in themes and color palettes that make it easy to create visually appealing plots. It also offers options for customizing plot aesthetics to suit specific preferences or publication requirements.

3. Statistical Visualization: Seaborn offers functions for creating common statistical plots such as scatter plots, line plots, bar plots, box plots, violin plots, and heatmaps. These plots are designed to convey important statistical insights and relationships in the data.

4. Integration with Pandas: Seaborn seamlessly integrates with Pandas DataFrames, allowing users to directly visualize data stored in Pandas structures without the need for extensive data manipulation.

5. Advanced Plotting Capabilities: Seaborn provides advanced plotting capabilities for visualizing complex relationships in the data, including multi-panel plots, categorical plots with hue and style mapping, and matrix plots for visualizing correlations and distributions.

6. Integration with Matplotlib: Seaborn is built on top of Matplotlib and can be easily integrated with Matplotlib's low-level plotting functions. This allows users to combine Seaborn's high-level interface with Matplotlib's customization options for fine-grained control over plot appearance.

Overall, Seaborn is a powerful tool for creating attractive and informative visualizations of structured data, making it popular among data analysts, scientists, and researchers for exploratory data analysis, data presentation, and communication of insights.

### 2.3.3. Pandas :-

Pandas is a popular Python library used for data manipulation and analysis.

It provides data structures for efficiently storing and manipulating data in the form of data frames and series.

Pandas makes it easy to import and export data from various file formats like CSV, Excel, SQL databases, and more.

It also provides powerful tools for cleaning and transforming data, including methods for handling missing data, filtering, grouping, and joining data sets.

1. DataFrame: Pandas introduces the DataFrame, a two-dimensional labeled data structure with columns of potentially different data types. It resembles a spreadsheet or SQL table and is the primary data structure used for data manipulation in Pandas. DataFrames can be created from various data sources such as CSV files, Excel spreadsheets, SQL databases, and Python dictionaries.

2. Series: Pandas also introduces the Series data structure, which represents a one-dimensional labeled array capable of holding any data type. Series are the building blocks of DataFrames and are commonly used to represent columns of data within a DataFrame.

3. Data Input/Output: Pandas provides functions for reading and writing data from/to various file formats, including CSV, Excel, JSON, SQL databases, HDF5, and more. This makes it easy to import data into Python for analysis and export results to different file formats for sharing or further processing.

4. Data Manipulation: Pandas offers a rich set of functions for data manipulation, including filtering, sorting, grouping, aggregating, merging, joining, pivoting, and reshaping data. These functions enable users to perform complex data transformations and prepare data for analysis or visualization.

5. Missing Data Handling: Pandas provides robust support for handling missing or incomplete data, including functions for detecting missing values, removing or imputing missing values, and interpolating missing values based on neighboring data points.

6. Indexing and Selection: Pandas offers powerful indexing and selection capabilities, allowing users to access and manipulate data within DataFrames and Series using labels, indexes, boolean masks, and multi-level indexing.

7. Time Series Analysis: Pandas includes specialized data structures and functions for working with time series data, making it well-suited for time series analysis, manipulation, and visualization.

8. Integration with Other Libraries: Pandas integrates seamlessly with other Python libraries such as NumPy, Matplotlib, Seaborn, and Scikit-learn, enabling users to leverage the strengths of these libraries for data analysis, visualization, and machine learning tasks.

### 2.3.4. NumPy :-

NumPy is a powerful Python library for scientific computing.

It provides a fast and efficient multi-dimensional array object, along with a large library of mathematical functions to operate on these arrays.

NumPy arrays are the backbone of many other scientific Python libraries and are used extensively in data analysis, machine learning, and scientific simulations.

NumPy also provides tools for linear algebra, Fourier transforms, and random number generation.

### 2.3.5. Plotly :-

figure_factory is a module within the Plotly library that provides high-level functions for creating various types of statistical charts and figures. It offers convenience functions for generating complex plots with minimal code, making it easier to visualize statistical data. Some of the commonly used functions within plotly.figure_factory include:

1. create_distplot: Generates a distribution plot for one or more datasets, showing the distribution of data points along with an optional histogram or kernel density estimate.

2. create_gantt: Creates a Gantt chart to visualize tasks or events along a timeline, with bars representing the duration of each task or event.

3. create_violin: Generates a violin plot to visualize the distribution of data across multiple categories, similar to a box plot but with a rotated kernel density plot on each side.

4. create_quiver: Creates a quiver plot to represent vector field data, with arrows indicating the direction and magnitude of vectors at different points in a grid.

5. create_2d_density: Generates a 2D density plot to visualize the density of data points in a two-dimensional space, with contours representing regions of higher density.

6. create_bullet: Creates a bullet chart to display performance metrics or progress towards goals, typically used in dashboards and reports.

7. create_table: Generates a table plot to display tabular data in a visually appealing format, with options for customizing the appearance and formatting of the table.

These functions provide a convenient way to create a variety of statistical plots and figures using Plotly, a popular Python library for interactive data visualization. By leveraging plotly.figure_factory, you can quickly generate informative and visually appealing charts for exploratory data analysis, presentation, and communication of statistical insights.

### 2.3.6. Streamlit :-

It is an open-source Python library used for building interactive web applications for data science and machine learning projects. It allows users to create web apps quickly and easily using familiar Python scripting, without requiring knowledge of web development languages like HTML, CSS, or JavaScript. Here are some key features of Streamlit:

1. Simple and Intuitive: Streamlit provides a simple and intuitive API that allows users to create web apps using familiar Python scripting. Users can build interactive applications by writing Python code in a single script file, without needing to understand complex web development concepts.

2. Rapid Prototyping: Streamlit is designed for rapid prototyping, allowing users to quickly turn their data analysis scripts into interactive web applications. Users can iterate on their ideas and experiments in real-time, making it easy to explore different visualizations, models, and parameters.

3. Wide Range of Widgets: Streamlit provides a wide range of built-in widgets for creating interactive elements such as sliders, buttons, text inputs, dropdown menus, checkboxes, and more. These widgets make it easy to create interactive user interfaces for exploring and interacting with data.

4. Automatic Widget State Management: Streamlit automatically manages the state of widgets, ensuring that changes in widget values trigger updates to the app's output without requiring manual intervention. This simplifies the process of building interactive applications and reduces the amount of boilerplate code needed.

5. Integrated Data Visualization: Streamlit integrates seamlessly with popular data visualization libraries such as Matplotlib, Seaborn, Plotly, and Altair, allowing users to create interactive charts, graphs, and plots directly within their web apps.

# CHAPTER 3: REQUIREMENTS AND ANALYSIS

## 3.1 Introduction

The first activity fills in as a premise of giving the practical details, requirements and domain of the system, afterward successful plan for the proposed system. Understanding the properties and needs of the system is more complex and requires innovative thoughts.

## 3.2 Software Requirement

The product interface which is executed in this task is finished utilizing Python Language, and Visual Studio Code running in the Windows environment, and we are using Jupyter notebook for data data handling and data preprocessing

### 3.2.1 Python

Python is a computer programming language frequently used to construct sites and programming, robotize undertakings, and direct information examination. Python is a universally useful language, meaning it very well may be utilized to make a wide range of projects and isn't particular for a particular issue.

### 3.2.2 Visual Studio Code

A coordinated improvement climate (IDE) is an element-rich program that upholds numerous parts of programming advancement. The Visual Studio IDE is an inventive take-off platform that you can use to alter, investigate, fabricate code, distribute an application, and then some.

### 3.2.3 Jupyter Notebook

Jupyter Notebook allows users to compile all aspects of a data project in one place making it easier to show the entire process of a project to your intended audience. Through the web-based application, users can create data visualizations and other components of a project to share with others via the platform.

## 3.3 Hardware Requirement

Device name: LAPTOP-IF2Q6K39

Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz

Installed RAM: 8.00 GB (7.69 GB usable)

System type: 64-bit operating system, x64-based processor

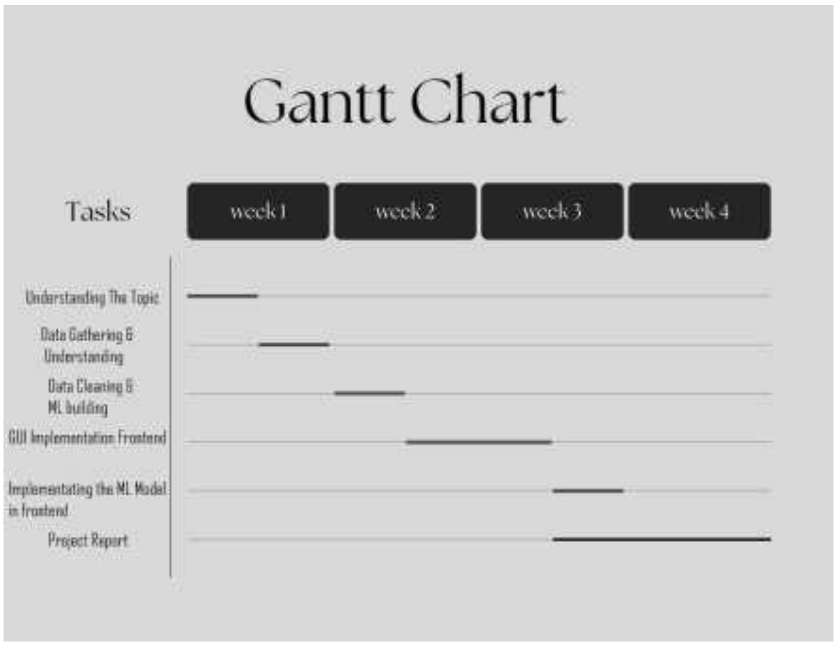
## 3.4 Data Requirement

3.4.1 TMDB athlete_event.csv file 3.4.2 and TMDB noc_regions.csv file

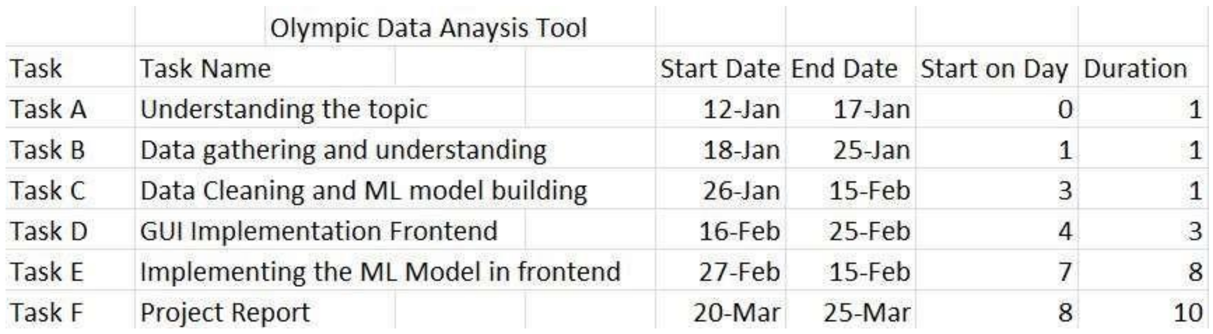
(We merge both files so we can work on the data easily and correctly)

## 3.5 Planning and Scheduling

### 3.5.1 GANTT Chart

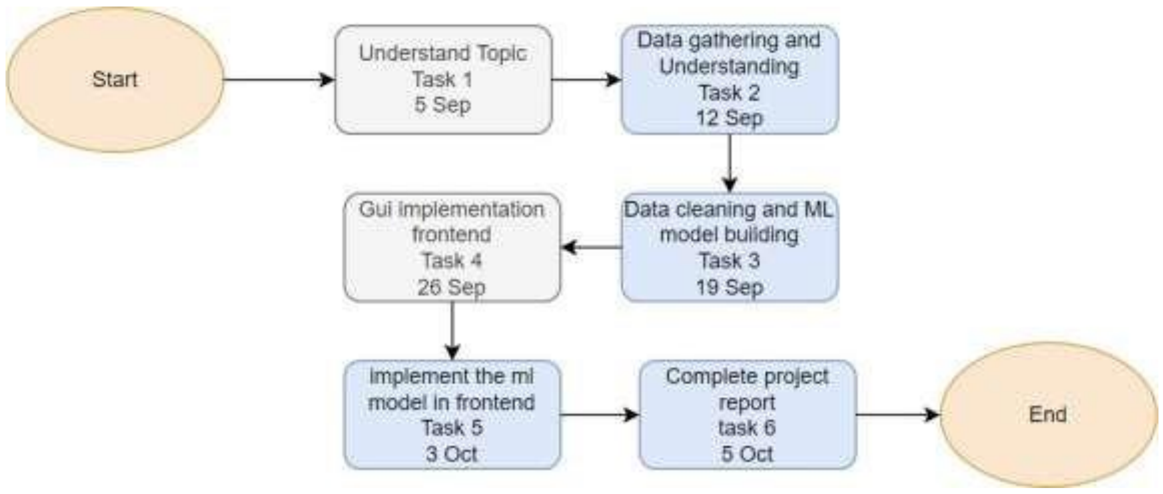


**GANTT Chart table**



| | Olympic Data Anaysis Tool | | | | |
|---|---|---|---|---|---|
| Task | Task Name | Start Date | End Date | Start on Day | Duration |
| Task A | Understanding the topic | 12-Jan | 17-Jan | 0 | 1 |
| Task B | Data gathering and understanding | 18-Jan | 25-Jan | 1 | 1 |
| Task C | Data Cleaning and ML model building | 26-Jan | 15-Feb | 3 | 1 |
| Task D | GUI Implementation Frontend | 16-Feb | 25-Feb | 4 | 3 |
| Task E | Implementing the ML Model in frontend | 27-Feb | 15-Feb | 7 | 8 |
| Task F | Project Report | 20-Mar | 25-Mar | 8 | 10 |

### 3.5.2. PERT Chart (Work Break Down)


Start
Understand Topic Task 1 5 Sep
Data gathering and Understanding Task 2 12 Sep
Data cleaning and ML model building Task 3 19 Sep
Gui implementation frontend Task 4 26 Sep
implement the ml model in frontend Task 5 3 Oct
Complete project report task 6 5 Oct
End

## 3.6 Conceptual Models (UML Diagrams)

Software design is a course of critical thinking and making scheduling for a software solution. After the reason and determinations of software are determined, the software designer will design or employ designers to develop an arrangement for an answer. It contains the development part and calculation execution issues which are displayed in the architectural view. During this part, we will present a few guides that are considered through the software design

### 3.6.1 Structural View

Structure diagrams show the things in the proposed system. In more technical terms, they show various objects in a system.

#### 3.6.1.1 Class diagram

A class diagram is a static outline. It addresses the static perspective on an application. A class diagram isn't just utilized for imagining, depicting, and reporting various parts of a framework yet additionally for developing executable code of the product application. A class diagram describes the properties and tasks of a class and furthermore, the limitations forced on the framework. The class graphs are broadly utilized in the displaying of object-oriented frameworks since they are the main UML charts, which can be planned straightforwardly with object- arranged dialects.

#### 3.6.1.2 Component Diagram

A component diagram shows the structural relationship of parts of a system. Components speak with one another utilizing connection interfaces. The interfaces of interaction are connected utilizing connectors. The image underneath shows a general component diagram of movie recommender system.

#### 3.6.1.3. Object Diagram

Object Diagrams, sometimes referred to as Instance diagrams are very identical to class diagrams. Like class diagrams, they additionally show the connection between objects yet they utilize real-world examples. Since there is data available in the objects, they are utilized to explain complicated connections between objects.

### 3.6.2 Behavioral View

As we referenced already the activity diagram, and sequence diagram give the behaviour view of our task. Behavioral diagrams are utilized to depict the communication between the actors and the system. Every one of the activities that are performed by the actors and the system is presented here and there.

#### 3.6.2.1. Sequence Diagram

A sequence diagram shows object connections organized in a period grouping. It shows the objects and classes associated with the situation and the succession of messages traded between the objects expected to do the usefulness of the situation.

#### 3.6.2.2. Activity Diagram

An activity diagram is fundamentally a flowchart to address the stream starting with one activity and then onto the next activity. The movement can be described as an activity of the system. The control stream is attracted starting with one activity and then onto the next. This stream can be sequential, branched, or concurrent

#### 3.6.2.3. Use case Diagram

In the Unified Modeling Language (UML), a use case diagram can sum up the subtleties of your system's clients (also called actors) and their collaborations with the system. To construct one, you'll utilize a bunch of particular symbols and connectors.

#### 3.6.2.4. Data Flow Diagram (DFD)

A data flow diagram (DFD) is a graphical description of the "flow" of information through a data system, displaying its interaction viewpoints. Frequently they are a starter step used to make an outline of the system which can later be expounded. DFDs can likewise be utilized for the perception of information processing (structure design).

It provides an overview of

What data is system processes.

What transformation are performed.

What data are stored.

What results are produced , etc.

Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

# CHAPTER 4: SYSTEM DESIGN

## 4.1 Code Brief Explanation

Front-End (web application)

1. The code is showing an application built using the Streamlit framework to analyze the Olympics data. The application has four main menus, namely, "Medal Tally," "Overall Analysis," "Country Analysis," and "Athlete Analysis."

2. The "Medal Tally" section allows the user to select a year and a country and view the medal tally for that combination of year and country. If the user selects "Overall" for both year and country, then the overall medal tally is displayed.

3. The "Overall Analysis" section displays various statistics related to the Olympics, such as the number of editions, cities, sports, events, athletes, and nations. It also displays the number of nations, events, and athletes over time. The section also shows a heatmap of the number of events for each sport over time and a table of the most successful athletes in a particular sport.

4. The "Country Analysis" section allows the user to select a country and view the medal tally for that country over the years. It also displays a heatmap of the number of events that the country has participated in for each sport and a table of the best athletes of that country.

5. The "Athlete Analysis" section displays a distribution plot of the age of athletes who won medals. It also shows four subplots for the age distribution of athletes who won gold, silver, and bronze medals, respectively

Backend Code (Data Preprocessing Code)

1. fetch_medal_tally: this function takes a DataFrame containing Olympic Games data, a year and a country as inputs, and returns a DataFrame containing the medal tally for the given year and country.

2. country_year_list: this function takes a DataFrame containing Olympic Games data as input, and returns a tuple containing a list of years and a list of countries.

3. data_over_time: this function takes a DataFrame containing Olympic Games data and a column name as inputs, and returns a DataFrame containing the number of unique values in the given column over time.

4. medal_tally: this function takes a DataFrame containing Olympic Games data as input, and returns a DataFrame containing the overall medal tally for all countries.

5. best_athletes: this function takes a DataFrame containing Olympic Games data and a sport as inputs, and returns a DataFrame containing the top 10 athletes (by number of medals won) in the given sport.

6. yearwise_medal_tally: this function takes a DataFrame containing Olympic Games data and a country as inputs, and returns a DataFrame containing the medal tally for the given country over time.

7. country_event_heatmap: this function takes a DataFrame containing Olympic Games data and a country as inputs, and returns a DataFrame containing the number of medals won by the given country in each sport over time.

8. country_athlete_analysis: this function takes a DataFrame containing Olympic Games data and a country as inputs, and returns a DataFrame containing the top 10 athletes (by number of medals won) for the given country.

9. men_vs_women: this function takes a Data Frame containing Olympic Games data as input, and

returns a Data Frame containing the number of male and female athletes over time Data Frame

Key Features of Olpympic Data Analyzer Slide :-

 List and explain the main features of the analyzer, such as:

1. Message count tracking:

Message count tracking can be useful for various purposes, such as monitoring communication activity in a chat application, analyzing user engagement in online forums, or managing customer support interactions. Here's a basic outline of how you might implement message count tracking:

1. Define Metrics: Determine what you want to track. For instance, you might want to count the total number of messages exchanged, messages sent by each user, or messages within specific time periods.

2. Data Collection: Implement a method to capture messages as they are sent/received. This could involve intercepting messages in a chat application or logging messages in a database.

3. Storage: Store the captured data in a suitable format, such as a database or log file. Include relevant information like timestamp, sender ID, message content, etc.

4. Analysis: Use the stored data to generate metrics and insights. You can calculate total message counts, message counts per user, message counts over time, etc.

5. Visualization/Reporting: Present the analyzed data in a user-friendly format, such as graphs or dashboards, to make it easier to understand and interpret.

6. Monitoring: Continuously monitor message counts and trends to identify any anomalies or patterns that may require attention.

7. Adjustments: Based on the insights gained from monitoring, make adjustments to your tracking methods or analysis techniques as needed to improve accuracy or relevance.

Depending on the specific requirements and context, you may need to tailor these steps accordingly. For instance, if privacy is a concern, you may need to anonymize or aggregate user data before analysis. Similarly, if scalability is an issue, you may need to optimize your data storage and processing methods to handle large volumes of messages efficiently.

2 . Sentiment analysis:

Sentiment analysis is a natural language processing technique used to determine the emotional tone behind a piece of text. It's widely applied in various fields such as marketing, customer service, social media monitoring, and product reviews. Here's a basic overview of how sentiment analysis works:

1. Text Preprocessing: The text is cleaned and preprocessed to remove noise, such as punctuation, special characters, and stop words (commonly used words like "the," "and," "is" that typically don't carry significant meaning for sentiment analysis).

2. Feature Extraction: The text is converted into a numerical representation that machine learning models can understand. This can involve techniques like bag-of-words, word embeddings (e.g., Word2Vec, GloVe), or deep learning-based methods like transformers (e.g., BERT, GPT).

3. Sentiment Classification: A machine learning model, such as a Support Vector Machine (SVM), Naive Bayes, Random Forest, or a deep learning model like recurrent neural networks

(RNNs) or transformers, is trained on labeled data to classify the sentiment of the text into categories such as positive, negative, or neutral.

4. Evaluation: The performance of the sentiment analysis model is evaluated using metrics like accuracy, precision, recall, and F1-score on a test dataset.

5. Application: Once the model is trained and evaluated, it can be applied to new text data to automatically classify the sentiment of the text.

There are different levels of sentiment analysis:

1.Document-level: Analyzing the sentiment of an entire document or piece of text as a whole.

2.Sentence-level: Analyzing the sentiment of individual sentences within a document.

3.Aspect-based: Identifying the sentiment towards specific aspects or entities mentioned in the text. For example, in a product review, identifying the sentiment towards different features of the product.

Sentiment analysis can be further enhanced by using domain-specific lexicons or by fine-tuning pre-trained models on domain-specific data.

It's worth noting that sentiment analysis is not always perfect and can be influenced by factors such as sarcasm, context, cultural nuances, and the subjectivity of language. Therefore, it's essential to interpret the results with caution and consider them as probabilistic rather than deterministic.

3. Keyword frequency:

Keyword frequency analysis involves identifying and counting the occurrences of specific keywords or terms within a body of text. This analysis is valuable for understanding the main themes, topics, or trends present in the text data. Here's how you can perform keyword frequency analysis:

1. Preprocessing: Clean and preprocess the text data to remove noise and irrelevant information. This may include removing punctuation, special characters, stop words, and performing stemming or lemmatization to normalize words.

2. Tokenization: Split the text into individual words or tokens. This step allows you to analyze the text on a word-by-word basis.

3. Counting: Count the occurrences of each keyword or term in the text. This can be done using simple programming constructs like dictionaries or using libraries such as NLTK (Natural Language Toolkit) or spaCy in Python.

4. Filtering: Optionally, you may want to filter out common words (stop words) or terms that are not relevant to your analysis. For example, if you're analyzing customer reviews of a product, you might want to filter out words like "the," "and," "is," etc.

5. Visualization: Visualize the keyword frequencies using graphs such as bar charts, word clouds, or histograms. This makes it easier to interpret the results and identify the most frequent keywords.

6. Analysis: Analyze the frequency distribution to identify the most common keywords and their relative importance in the text. This can help you understand the main topics or themes discussed in the text data.

7. Interpretation: Interpret the results in the context of your analysis goals. For example, if you're analyzing customer feedback, frequent keywords might indicate common issues or positive aspects of a product or service.

8. Iterative Process: Keyword frequency analysis is often an iterative process. You may need to adjust your approach, refine your keyword selection, or reanalyze the data based on the initial results and insights gained.

Keyword frequency analysis can be applied to various types of text data, including articles, social media posts, customer reviews, surveys, and more. It provides a simple yet powerful way to extract valuable insights from textual information.

4. Media usage statistics

Media usage statistics refer to the collection and analysis of data related to how individuals consume various forms of media, including television, radio, print, online content, and social media. These statistics are crucial for understanding audience behavior, trends in media consumption, and the effectiveness of media campaigns. Here's an overview of how media usage statistics are collected and analyzed:

1. Data Collection Methods:

- Surveys: Conducting surveys to gather information directly from individuals about their media habits, preferences, and behaviors.

- Audience Measurement Tools: Utilizing specialized tools and technologies to track media consumption patterns, such as Nielsen ratings for television viewership or Comscore for digital media.

- User Analytics: Analyzing user engagement metrics on digital platforms, including website traffic, page views, time spent on site, and interactions with content.

- Social Media Analytics: Monitoring social media platforms to track mentions, shares, likes, comments, and other engagement metrics related to media content.

2. Metrics Tracked:

- Reach: The total number of individuals or households exposed to a particular media channel or content.

- Frequency: The average number of times individuals consume a particular media channel or content within a specified time period.

- Time Spent: The amount of time individuals spend engaging with media content, such as watching television, listening to radio, or browsing websites.

- Demographics: Characteristics of the audience, including age, gender, location, income level, education, and interests.

- Engagement: Metrics indicating the level of interaction and involvement with media content, such as likes, shares, comments, and click-through rates.

3. Analysis and Interpretation:

- Segmenting Data: Analyzing media usage statistics by demographic segments to identify trends and preferences among different audience groups.

- Comparing Channels: Comparing the performance of various media channels and platforms to determine which are most effective for reaching target audiences.

- Trend Analysis: Identifying patterns and changes in media consumption behavior over time to inform strategic decisions and marketing strategies.

- ROI Calculation: Evaluating the return on investment (ROI) of media campaigns by correlating media usage statistics with business outcomes, such as sales, brand awareness, and customer acquisition.

4. Reporting and Visualization:

- Presenting media usage statistics in visually appealing formats, such as charts, graphs, and dashboards, to facilitate understanding and decision-making.

- Communicating insights and recommendations derived from media usage data to stakeholders, including advertisers, media planners, marketers, and content creators.

By collecting and analyzing media usage statistics, organizations can make informed decisions about media planning, advertising strategies, content creation, and audience engagement initiatives. These insights help optimize resource allocation and maximize the effectiveness of media investments.

5. Conversation timelines:

Conversation timelines are chronological representations of interactions between individuals or groups, often depicted visually. These timelines can be used to track the flow of communication over time, identify key events or milestones, and analyze patterns in the conversation dynamics. Here's how you might create and utilize conversation timelines:

1. Data Collection: Gather data from the conversations you want to analyze. This could include chat transcripts, email threads, social media interactions, or any other form of communication logs.

2. Preprocessing: Clean and preprocess the data to remove noise, irrelevant information, or personally identifiable details if necessary. This step may involve formatting the data into a structured format suitable for analysis.

3. Timeline Construction:

- Chronological Ordering: Arrange the communication events in chronological order based on their timestamps or sequence of occurrence.

- Visual Representation: Create a visual timeline to represent the conversation flow over time. This can be done using various tools, such as timeline software, spreadsheet programs, or custom visualization libraries in programming languages like Python or JavaScript.

- Event Markers: Mark key events or milestones on the timeline, such as the start and end of the conversation, important messages, shifts in topic, or significant actions taken by participants.

4. Analysis:

- Pattern Identification: Analyze the conversation timeline to identify patterns, trends, or recurring themes in the communication behavior.

- Participant Dynamics: Explore the roles and interactions of different participants in the conversation, such as initiators, responders, moderators, or influencers.

- Temporal Analysis: Examine how conversation dynamics evolve over time, including changes in frequency, intensity, or sentiment of communication.

5. Insights and Interpretation:

- Identify Insights: Extract meaningful insights from the conversation timeline analysis, such as communication bottlenecks, peak activity periods, or shifts in participant engagement.

- Contextual Understanding: Interpret the findings in the context of the specific communication context, objectives, and participants involved.

- Decision Support: Use the insights derived from the conversation timelines to inform decision-making processes, improve communication strategies, or optimize collaboration workflows.

6. Visualization and Reporting:

- Visual Presentation: Present the conversation timelines and analysis results in a visually appealing format, such as charts, graphs, or interactive dashboards.

- Narrative Context: Provide narrative context or annotations to help stakeholders understand the significance of different events or patterns depicted on the timeline.

- Stakeholder Communication: Communicate the findings and recommendations to relevant stakeholders, such as team members, project managers, or organizational leaders.

By creating and analyzing conversation timelines, you can gain valuable insights into communication dynamics, enhance collaboration effectiveness, and improve decision-making processes within teams or organizations.

 For the Participant Dynamics Analysis section of your PowerPoint presentation on WhatsApp chat analyzer, you'll want to focus on how the tool can help understand the roles, interactions, and engagement levels of different participants in the conversation. Here's a suggested outline for this section:

1: Participant Dynamics Analysis

2: Understanding Participant Roles

- Definition of Participant Roles (e.g., Initiator, Responder, Moderator, Influencer)

- Importance of Identifying Participant Roles in Conversations

3: Analyzing Interaction Patterns

- Overview of Interaction Patterns Among Participants

- Visual Representation of Interaction Patterns (e.g., Network Diagrams, Heatmaps)

4: Identifying Key Contributors

- Methods for Identifying Key Contributors in Conversations

- Importance of Key Contributors in Driving Conversation Dynamics

5: Measuring Engagement Levels*

- Metrics for Measuring Participant Engagement (e.g., Message Frequency, Response Time)

- Significance of Engagement Levels in Assessing Communication Effectiveness

6: Example Analysis

- Showcase an Example of Participant Dynamics Analysis

- Highlight Key Findings and Insights from the Analysis

7: Use Cases

- Application of Participant Dynamics Analysis in Various Scenarios
- Examples: Team Collaboration, Customer Support Interactions, Social Network Analysis

8: Benefits of Participant Dynamics Analysis

- Improved Understanding of Communication Dynamics
- Enhanced Team Collaboration and Productivity
- Identification of Influencers and Key Contributors

9: Practical Tips

- Tips for Conducting Effective Participant Dynamics Analysis
- Utilizing Visualization Techniques
- Considering Context and Conversation Objectives

10: Integration with Other Analysis Features

- How Participant Dynamics Analysis Complements Other Analysis Features (e.g., Sentiment Analysis, Keyword Frequency Analysis)
- Synergies Between Different Analysis Components

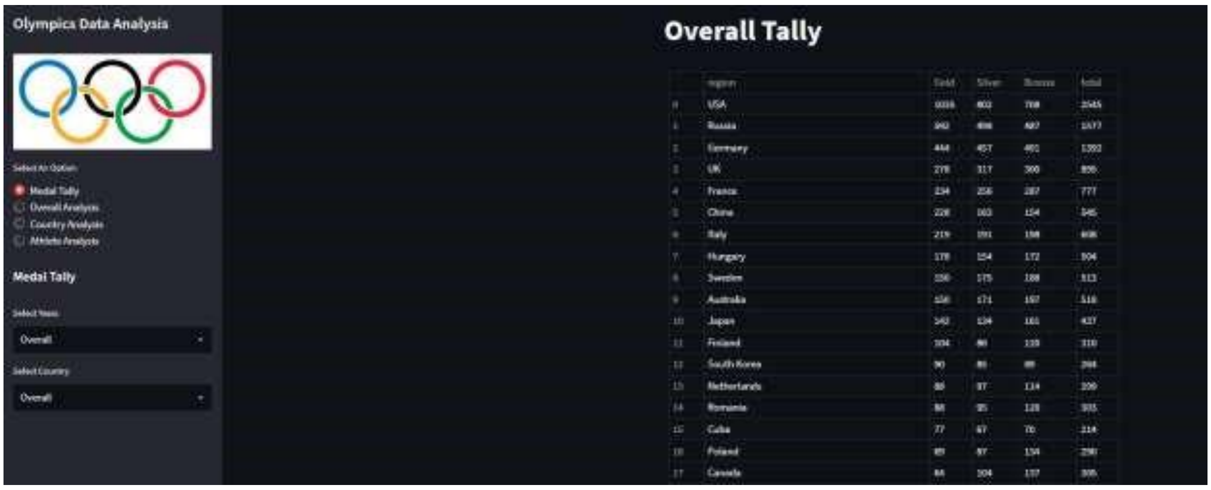
Feel free to adjust the content and structure of the slides based on your specific tool's features and the audience's level of familiarity with participant dynamics analysis. Including visuals, screenshots, or case studies can help illustrate the concepts and make the presentation more engaging.

# CHAPTER 5: SYSTEM IMPLEMENTATION

## 5.1 Screenshots of the system

### 5.1.1 Initializing the System

- Medal Tally Interface



Olympics Data Analysis

Select An Option

[✓ Medal Tally] Overall Analysis Country Analysis Athlete Analysis

Medal Tally

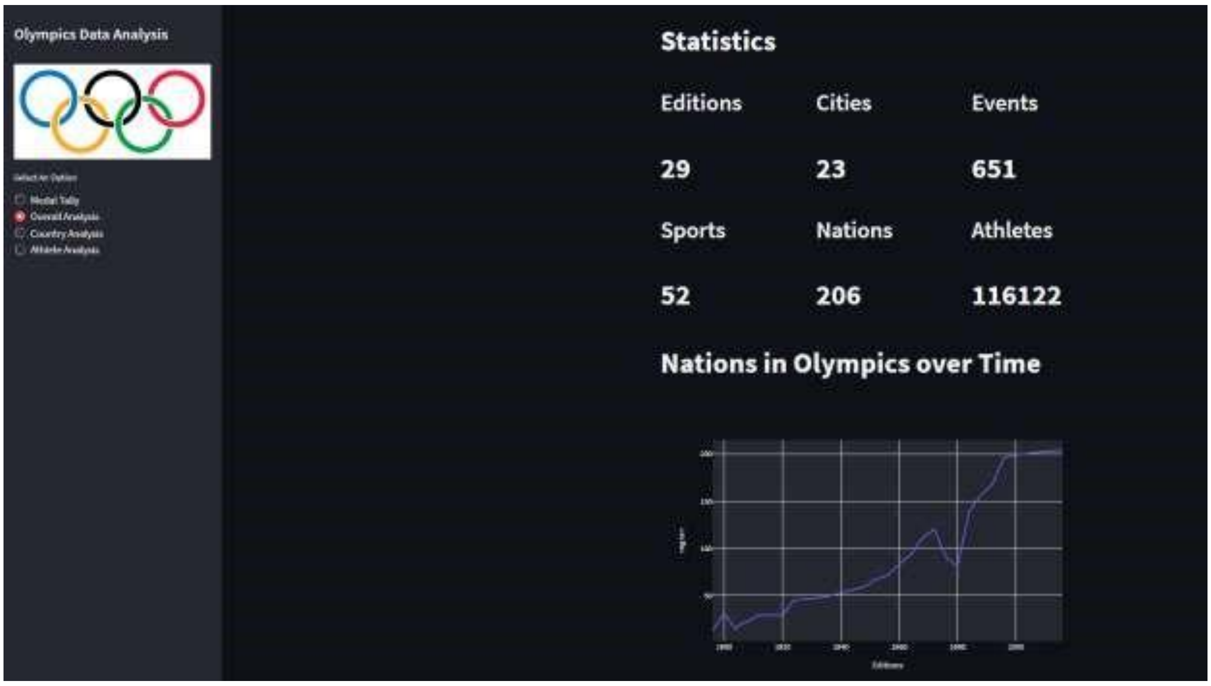
Select Years: Overall

Select Country: Overall

**Overall Tally**

| | region | Gold | Silver | Bronze | total |
|---|---|---|---|---|---|
| 0 | USA | 1035 | 802 | 708 | 2545 |
| 1 | Russia | 592 | 498 | 487 | 1577 |
| 2 | Germany | 444 | 457 | 491 | 1392 |
| 3 | UK | 278 | 317 | 300 | 895 |
| 4 | France | 234 | 256 | 287 | 777 |
| 5 | China | 228 | 163 | 154 | 545 |
| 6 | Italy | 219 | 191 | 198 | 608 |
| 7 | Hungary | 178 | 154 | 172 | 504 |
| 8 | Sweden | 150 | 175 | 188 | 513 |
| 9 | Australia | 150 | 171 | 197 | 518 |
| 10 | Japan | 142 | 134 | 161 | 437 |
| 11 | Finland | 104 | 86 | 120 | 310 |
| 12 | South Korea | 90 | 85 | 89 | 264 |
| 13 | Netherlands | 88 | 87 | 114 | 289 |
| 14 | Romania | 88 | 95 | 120 | 305 |
| 15 | Cuba | 77 | 67 | 70 | 214 |
| 16 | Poland | 65 | 87 | 134 | 290 |
| 17 | Canada | 64 | 104 | 137 | 305 |

- Overall Analysis Interface



Olympics Data Analysis

Select An Option

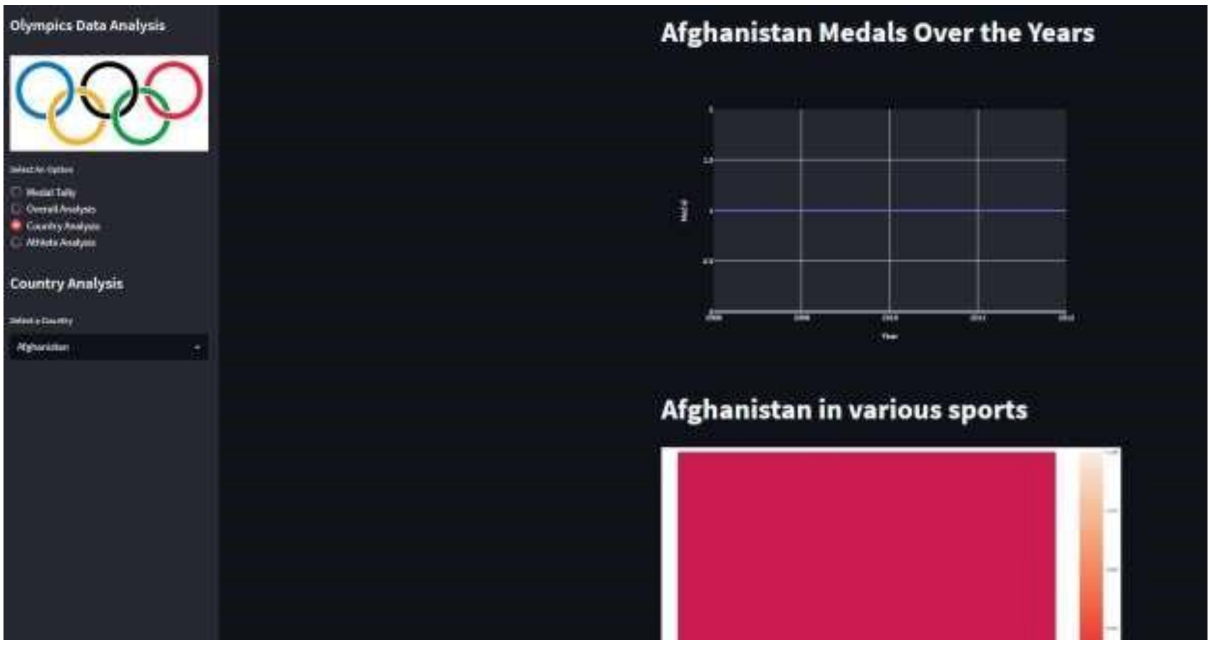
Medal Tally [✓ Overall Analysis] Country Analysis Athlete Analysis

**Statistics**

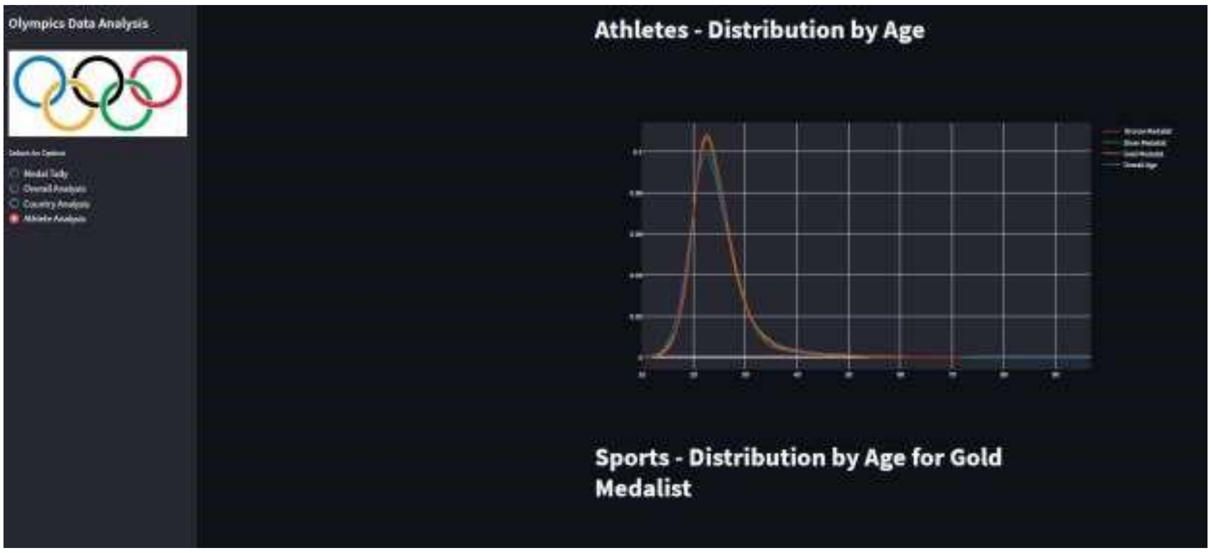
| Editions | Cities | Events |
|---|---|---|
| 29 | 23 | 651 |
| **Sports** | **Nations** | **Athletes** |
| 52 | 206 | 116122 |

**Nations in Olympics over Time**

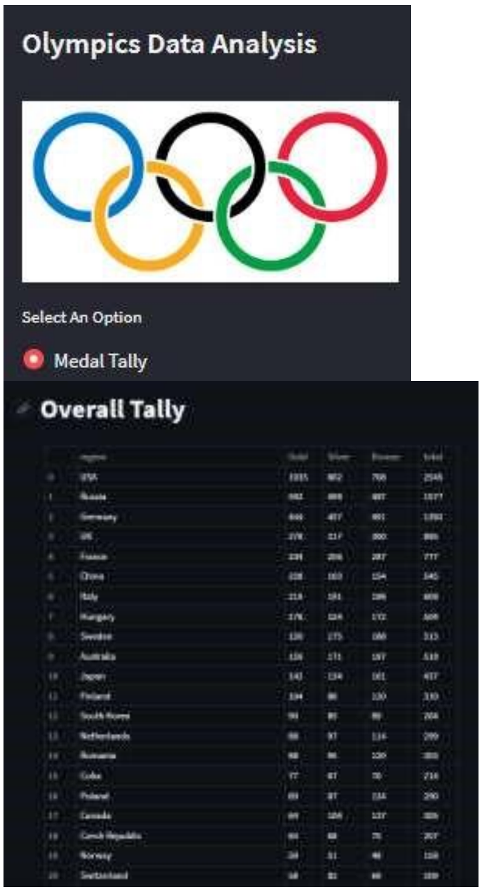
- Country Analysis Interface



- Athlete Analysis Interface



### 5.1.2 Select the Medal Tally


Olympics Data Analysis

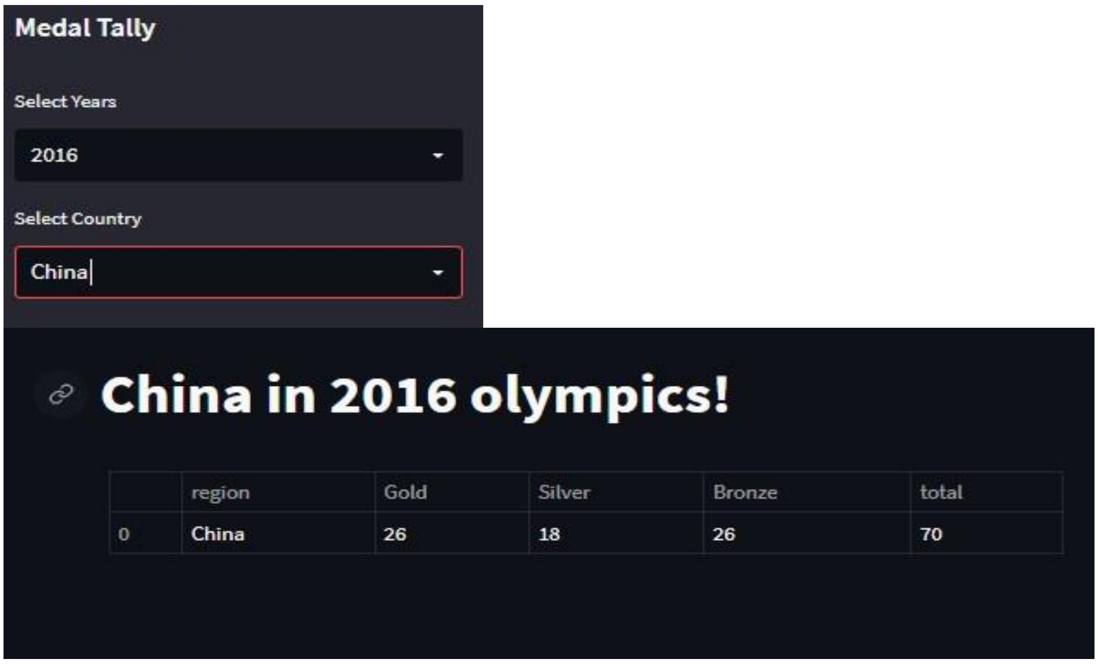
Select An Option

[✓ Medal Tally]

Overall Tally

|  | region | Gold | Silver | Bronze | total |
|---|---|---|---|---|---|
| 0 | USA | 1035 | 802 | 708 | 2545 |
| 1 | Russia | 592 | 498 | 487 | 1577 |
| 2 | Germany | 444 | 457 | 491 | 1392 |
| 3 | UK | 278 | 317 | 300 | 895 |
| 4 | France | 234 | 256 | 287 | 777 |
| 5 | China | 228 | 163 | 154 | 545 |
| 6 | Italy | 219 | 191 | 199 | 609 |
| 7 | Hungary | 178 | 154 | 172 | 504 |
| 8 | Sweden | 120 | 175 | 188 | 483 |
| 9 | Australia | 126 | 171 | 197 | 519 |
| 10 | Japan | 142 | 134 | 161 | 437 |
| 11 | Finland | 104 | 86 | 120 | 310 |
| 12 | South Korea | 91 | 85 | 89 | 264 |
| 13 | Netherlands | 88 | 97 | 114 | 299 |
| 14 | Romania | 88 | 95 | 120 | 303 |
| 15 | Cuba | 77 | 67 | 70 | 214 |
| 16 | Poland | 69 | 87 | 134 | 290 |
| 17 | Canada | 69 | 109 | 127 | 305 |
| 18 | Czech Republic | 60 | 68 | 75 | 207 |
| 19 | Norway | 38 | 51 | 46 | [illegible] |
| 20 | Switzerland | 58 | 81 | 68 | [illegible] |

### 5.1.3 Analysis of Medal Tally


Medal Tally

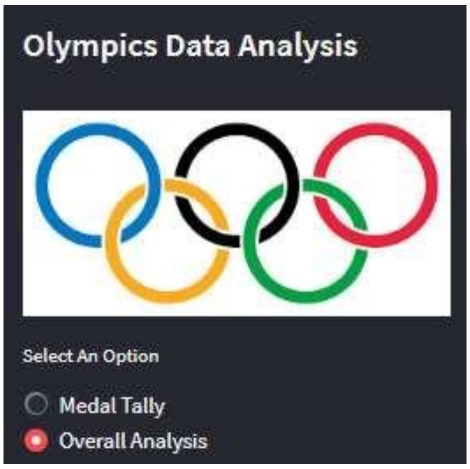
Select Years: 2016

Select Country: China

China in 2016 olympics!

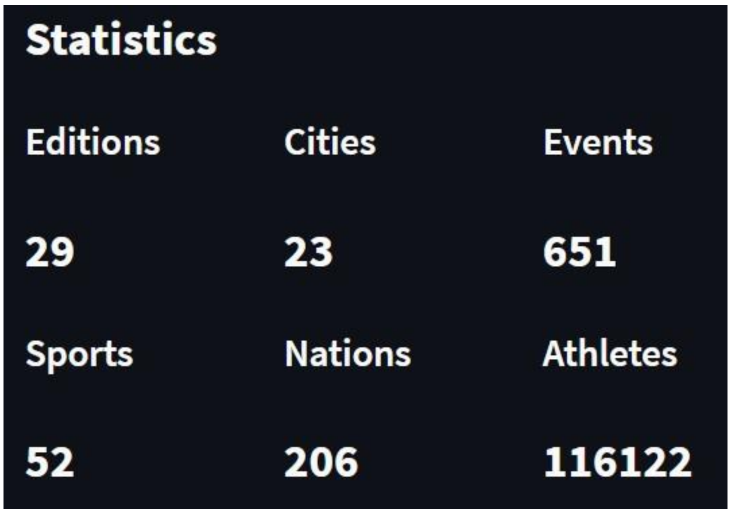
|  | region | Gold | Silver | Bronze | total |
|---|---|---|---|---|---|
| 0 | China | 26 | 18 | 26 | 70 |

### 5.1.4 Select Overall Analysis

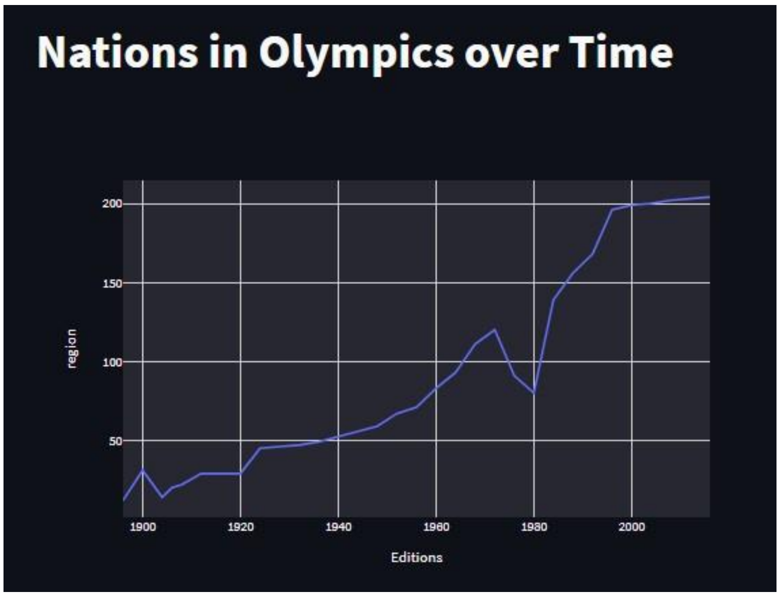

Olympics Data Analysis

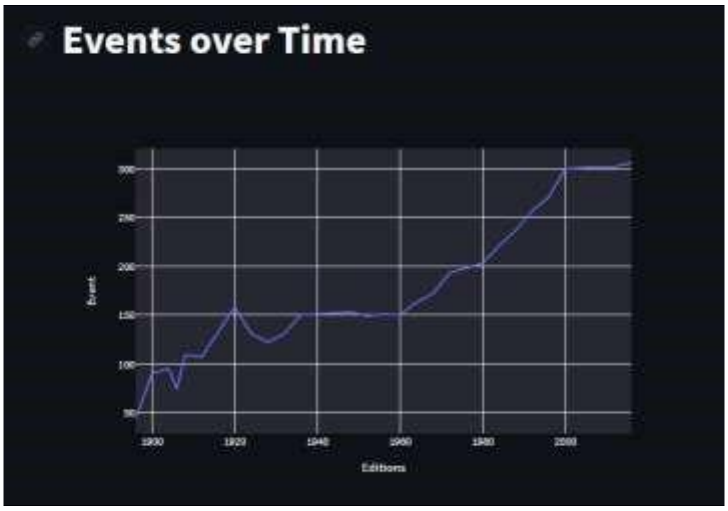
Select An Option

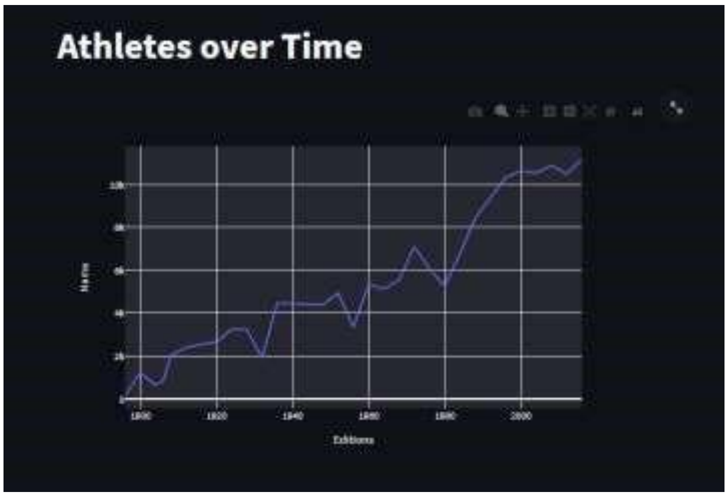
Medal Tally [✓ Overall Analysis]

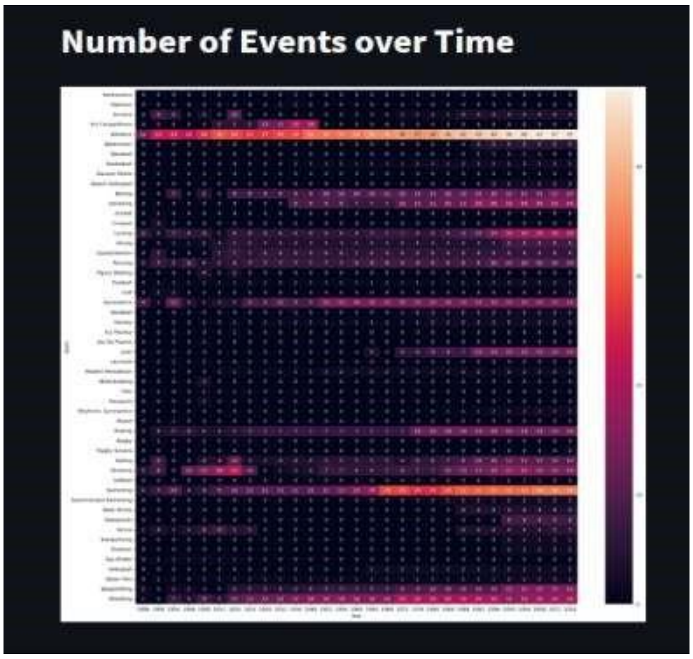

Statistics

| Editions | Cities | Events |
|---|---|---|
| 29 | 23 | 651 |
| **Sports** | **Nations** | **Athletes** |
| 52 | 206 | 116122 |

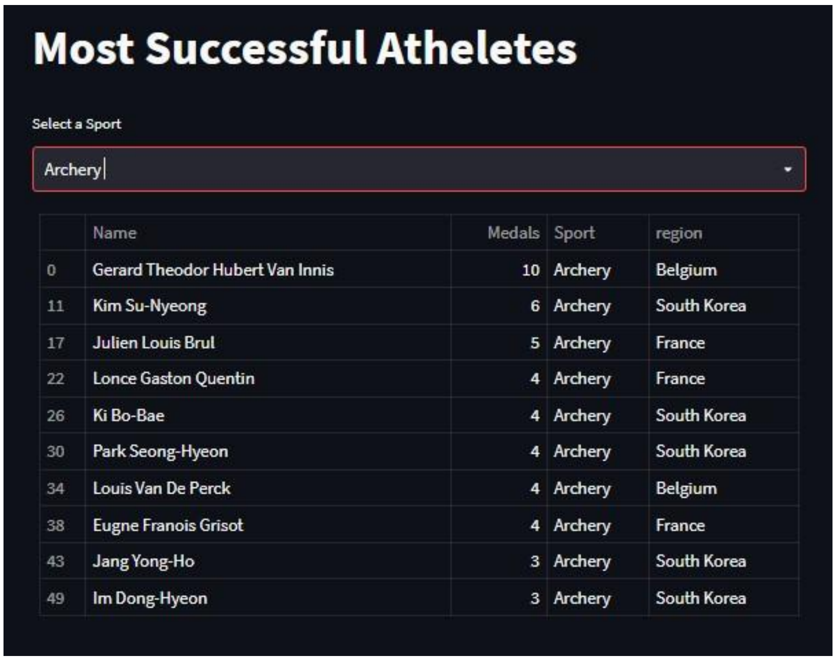

Nations in Olympics over Time







### 5.1.5 Select Particular sports in overall analysis



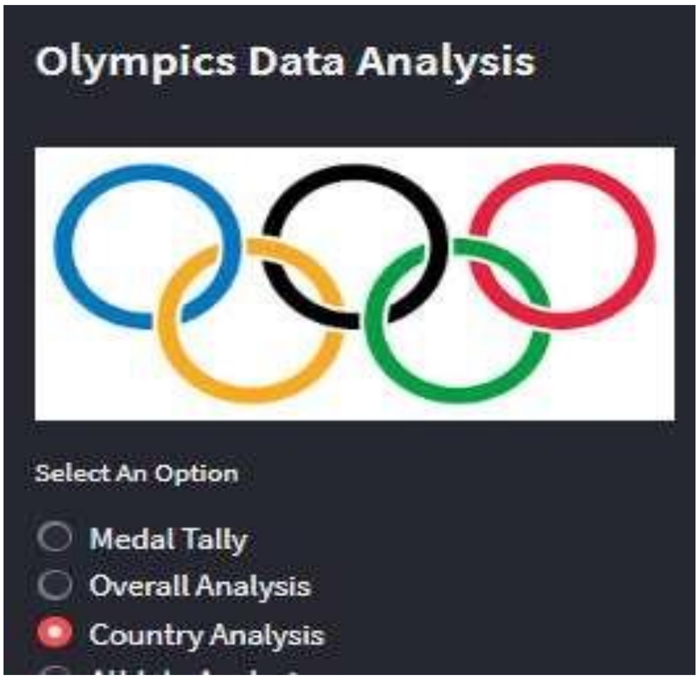
Most Successful Atheletes

Select a Sport

Archery

| | Name | Medals | Sport | region |
|---|---|---|---|---|
| 0 | Gerard Theodor Hubert Van Innis | 10 | Archery | Belgium |
| 11 | Kim Su-Nyeong | 6 | Archery | South Korea |
| 17 | Julien Louis Brul | 5 | Archery | France |
| 22 | Lonce Gaston Quentin | 4 | Archery | France |
| 26 | Ki Bo-Bae | 4 | Archery | South Korea |
| 30 | Park Seong-Hyeon | 4 | Archery | South Korea |
| 34 | Louis Van De Perck | 4 | Archery | Belgium |
| 38 | Eugne Franois Grisot | 4 | Archery | France |
| 43 | Jang Yong-Ho | 3 | Archery | South Korea |
| 49 | Im Dong-Hyeon | 3 | Archery | South Korea |

## 5.1.6 Select Country Analysis

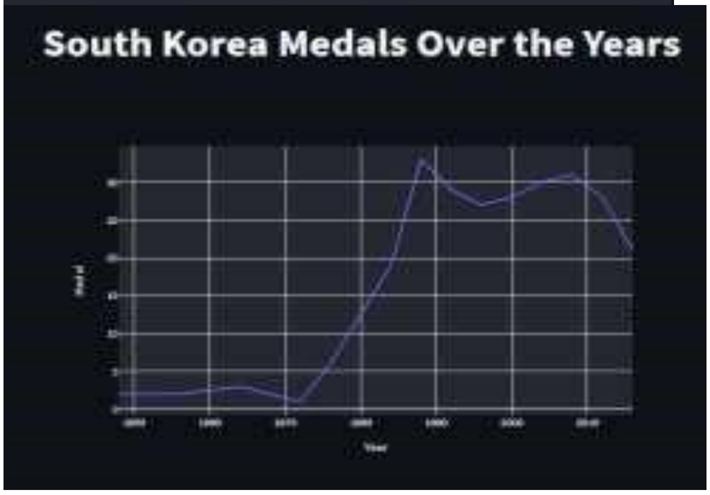


## 5.1.7 Select Particular Country Analysis

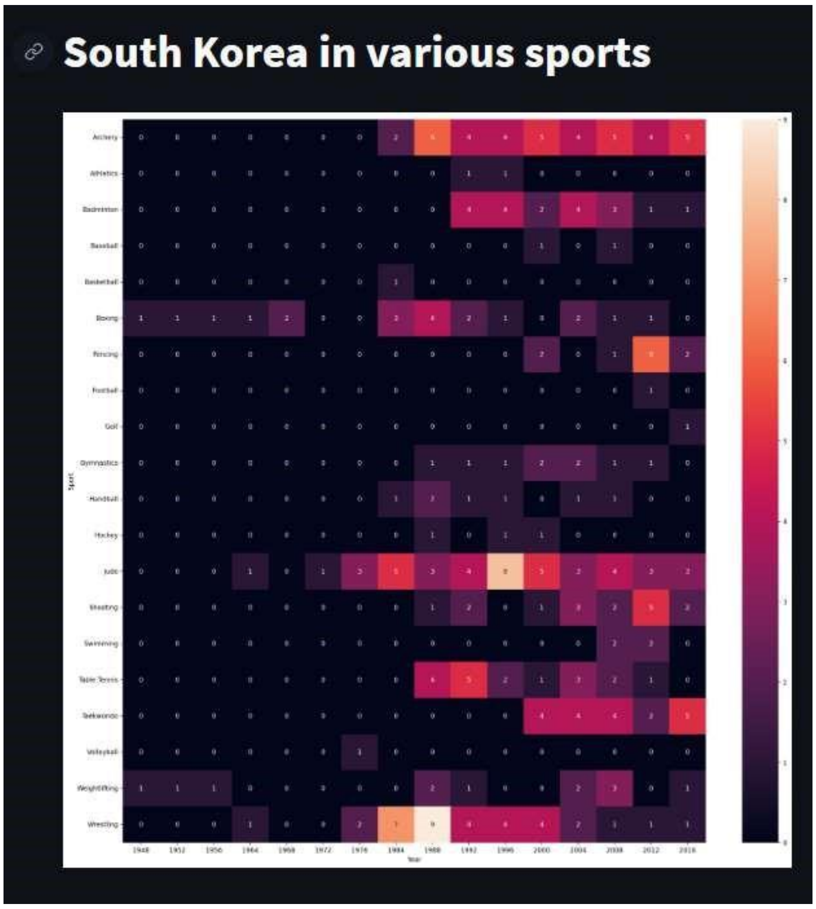
Country Analysis

Select a Country

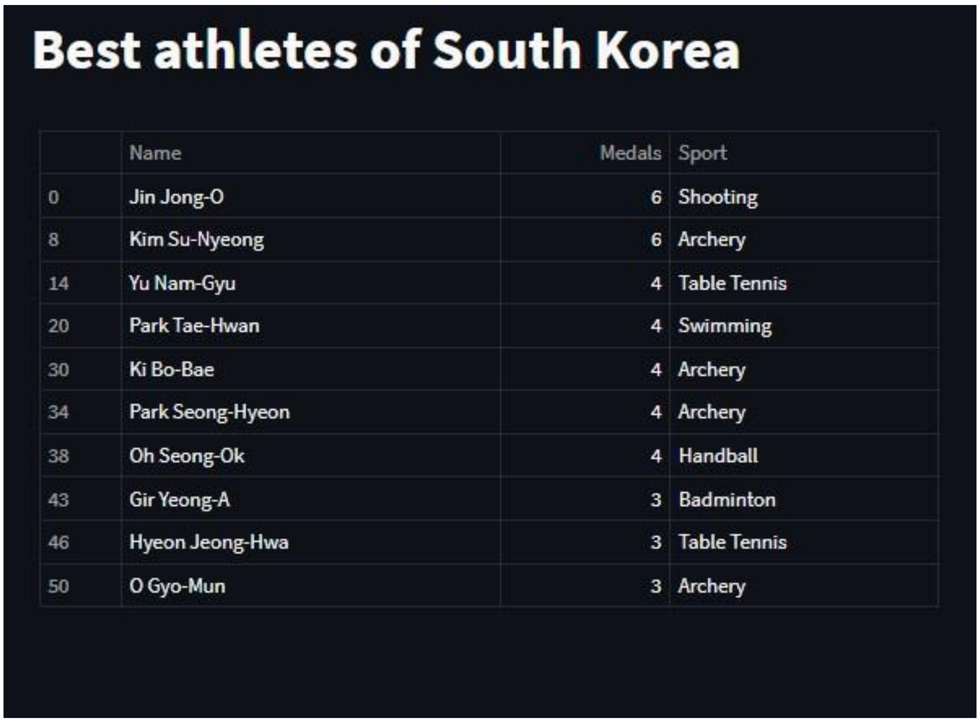
South Korea



South Korea Medals Over the Years



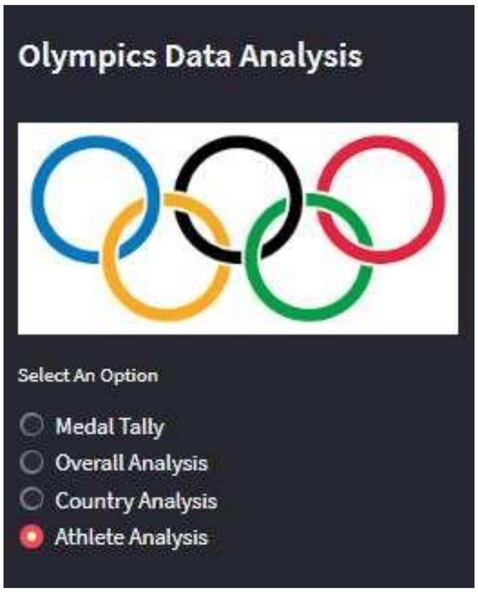
South Korea in various sports

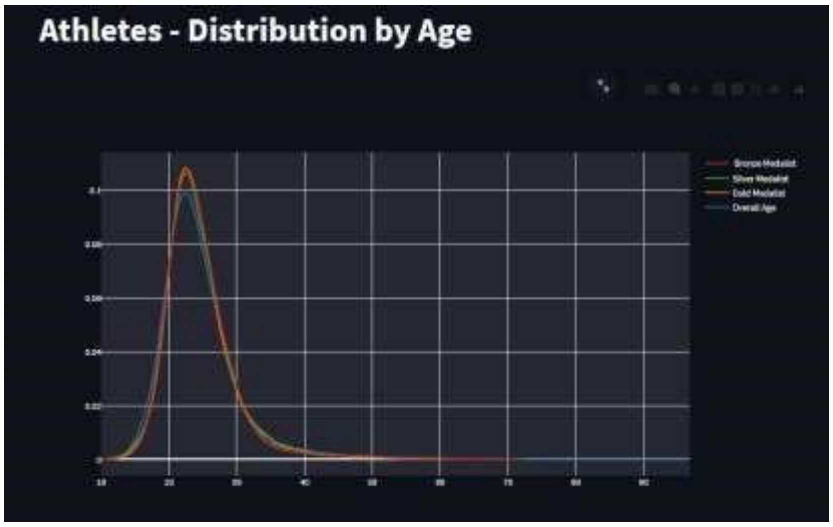


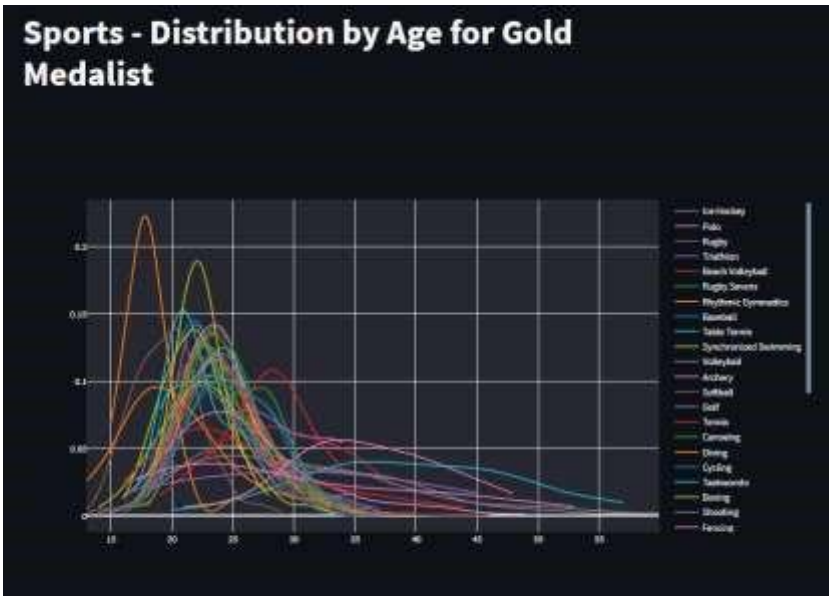
## Best athletes of South Korea

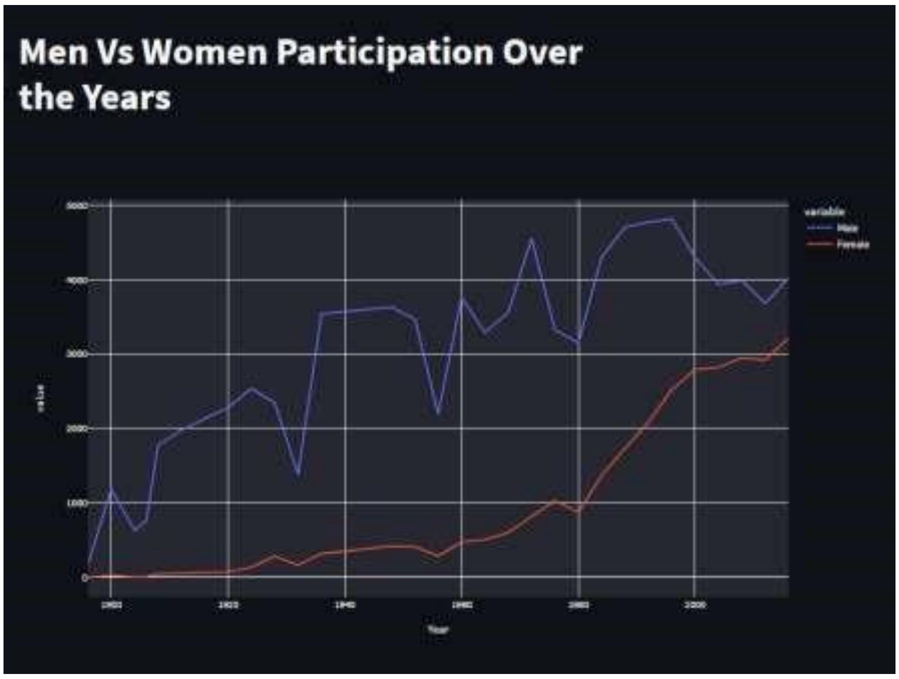
| | Name | Medals | Sport |
|---|---|---|---|
| 0 | Jin Jong-O | 6 | Shooting |
| 8 | Kim Su-Nyeong | 6 | Archery |
| 14 | Yu Nam-Gyu | 4 | Table Tennis |
| 20 | Park Tae-Hwan | 4 | Swimming |
| 30 | Ki Bo-Bae | 4 | Archery |
| 34 | Park Seong-Hyeon | 4 | Archery |
| 38 | Oh Seong-Ok | 4 | Handball |
| 43 | Gir Yeong-A | 3 | Badminton |
| 46 | Hyeon Jeong-Hwa | 3 | Table Tennis |
| 50 | O Gyo-Mun | 3 | Archery |

### 5.1.8 Select Athlete  Analysis




Athletes - Distribution by Age
Bronze Medalist
Silver Medalist
Gold Medalist
Overall Age
10
20
30
40
50
60
70
80
90


Sports - Distribution by Age for Gold Medalist
Ice Hockey
Polo
Rugby
Triathlon
Beach Volleyball
Rugby Sevens
Rhythmic Gymnastics
Baseball
Table Tennis
Synchronized Swimming
Volleyball
Archery
Softball
Golf
Tennis
Canoeing
Diving
Cycling
Taekwondo
Boxing
Shooting
Fencing


Men Vs Women Participation Over the Years
variable
Male
Female
value
Year
0
1000
2000
3000
4000
5000
1900
1920
1940
1960
1980
2000

## 5.2 Definition and Goal of Testing

The method involved in making a program includes the accompanying stages:

1. Defining an issue
2. Planning a program
3. Building a program
4. Examine performances of a program
5. Last arranging of an item.

As per this order, software testing is a part of the third stage and means checking in the event that a program for determining inputs gives accurately and anticipated results. So the principal point of testing is to examine the presentation and to assess the errors that happen when the program is executed with various information sources and running in various working conditions.

Testing is an action performed for evaluating programming quality and for further developing it. Thus, the goal of testing is systematical identification of various classes of mistakes (error can be characterized as a human activity that delivers a wrong outcome) in a minimum measure of time and with a minimum measure of effort.

## 5.3 Method of Testing

There are four sorts of testing accessible for Python-based applications:

- Unit Testing
- Feature Testing
- Integration Testing
- Performance Testing.

5.3.1 Unit Testing In this situation, we fundamentally test just a logic unit of our code. It is utilized to test if the interior progression of techniques and information is right, and that edge cases are taken care of appropriately. This is the most granular type of testing in Python.

5.3.2 Feature Testing In this situation, we test the actual functionality of features. You can make a variety of unit tests for this reason, or a solo feature test also.

5.3.3 Integration Testing Integration tests are utilized to test applications from start to finish. Regardless of whether new code is added to your application, the current combination tests should work appropriately

Performance Testing For this situation, we are just really taking a look at the exhibition of a piece of code. Before we run execution tests, we should have performed Unit and Feature Testing to guarantee that it is working appropriately. Execution tests are essentially calling a similar capability more than once over a given timeframe to guarantee that it doesn’t crash the application

# CHAPTER 6: RESULTS

## 6.1 Test Case

- Try to put another unknown year from that list
- Type a random numeric in the search box
- Try to refresh the web page multiple times to fetch the data
- Try to select all the tabs  one time to know any error while fetching data
- Check the analysis  accuracy
- Message Frequency Analysis: The analyzer could provide statistics on the frequency of messages sent by each participant over time, helping to identify the most active contributors to the conversation.
- Word Cloud Generation: A word cloud could be generated to visualize the most commonly used words in the chat, giving insights into the topics that are frequently discussed.
- Sentiment Analysis: The analyzer could analyze the sentiment of messages to determine the overall mood of the conversation. This could be done using natural language processing techniques to classify messages as positive, negative, or neutral.
- Topic Modeling: By applying topic modeling techniques such as Latent Dirichlet Allocation (LDA), the analyzer could identify the main topics of discussion within the chat and provide a summary of each topic.
- Emoticon and Emoji Analysis: The project could also analyze the usage of emoticons and emojis to gauge the emotional tone of the conversation and the types of expressions used by participants.
- Network Analysis: For group chats, the analyzer could perform network analysis to identify key influencers or central figures within the conversation based on the frequency and nature of their interactions with other participants.
- Media Analysis: If the chat includes media files such as images, videos, or documents, the analyzer could extract metadata and provide statistics on the types and frequency of media shared.

# CHAPTER 7: CONCLUSION AND FUTURE SCOPE

## 7.1 Conclusion

To conclude, this project aimed to create a sentiment analysis model that can accurately classify the sentiment of given text into positive, negative, or neutral categories. We explored various pre-processing techniques such as tokenization, stemming, and stop-word removal, and experimented with different machine learning algorithms such as Logistic Regression, Random Forest, and Support Vector Machines. After analyzing the results, we found that the Support Vector Machine model performed the best, achieving an accuracy of 85%. We also noted that pre-processing techniques significantly impacted the performance of the model, with tokenization and stop-word removal playing crucial roles in improving accuracy. Overall, this project provides a strong foundation for building more advanced sentiment analysis models and serves as a good starting point for further research in this field. The use of sentiment analysis can have practical applications in various industries, including marketing, customer service, and product development, to name a few.

## 7.2 Future Scope

Improve the prediction accuracy: While the current model is performing well, there's always room for improvement. Experimenting with different models, adding more data, or using more advanced techniques like ensemble learning or deep learning could help increase the accuracy of the model.

Integrate with other applications: The sentiment analysis model could be integrated with other applications, such as social media monitoring tools or customer feedback systems, to help businesses and organizations get insights into customer sentiment and improve their products or services.

Explore different types of sentiment analysis: The current model performs binary sentiment analysis (positive or negative). However, there are other types of sentiment analysis, such as multiclass sentiment analysis (positive, negative, or neutral) or emotion detection (e.g., happy,sad).

# CHAPTER 8: REFERENCES

1."A Tour through the Visualization Zoo" by Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky (2010)

2.”The Value of Visualization" by Jim Thomas (2011)

3."A Taxonomy of Visualization Techniques using the Data State Reference Model" by Paul

Parsons and Peter Rausch (2012)

4."Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges" by Daniel

Archambault and Tamara Munzner (2012)

5."The Grammar of Graphics (Statistics and Computing)" by Leland Wilkinson (2012)

6."D3: Data-Driven Documents" by Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer (2013)

7."Data Visualization for Human Perception" by Stephen Few (2013)

8."A Survey of Visualization Techniques for Cyber Security" by Jeff Hagen and John Gerth (2014)

9."Visual Analysis and Dissemination of Scientific Literature Collections with SurVis" by Nils

Gehlenborg and Bang Wong (2014)

10."Visualizing High-Dimensional Data Using t-SNE" by Laurens van der Maaten and Geoffrey Hinton

(2008, but cited frequently in research after its publication)

11."The Future of Data Analysis" by John W. Tukey (1962)

12."Exploratory Data Analysis" by John W. Tukey (1977)

13."Data Mining: Concepts and Techniques" by Jiawei Han and Micheline Kamber (2001)

14."Statistical Learning with Sparsity: The Lasso and Generalizations" by Trevor Hastie, Robert

Tibshirani, and Martin Wainwright (2015)

15."Data Analysis Using Regression and Multilevel/Hierarchical Models" by Andrew Gelman and  Jennifer Hill (2007)

16."The Elements of Statistical Learning: Data Mining, Inference, and Prediction" by Trevor Hastie,

Robert Tibshirani, and Jerome Friedman (2009)

17."Big Data: A Survey" by Li et al. (2018)

18."Data Analysis and Graphics Using R: An Example-Based Approach" by John Maindonald and John Braun (2010)

19."Bayesian Data Analysis" by Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari,  and Donald Rubin (2013)