A

Project Report on

## DATA BEHAVIOUR ANALYSIS USING INTELLIGENT BIG

## DATA ANALYTICS

*Submitted for partial fulfilment of the requirements for the award of the degree of*

### BACHELOR OF TECHNOLOGY

**in**

### DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

#### by

|  |  |
| --- | --- |
| **M. KOUSHIL** | **20K81A0597** |
| **L. SHIVANI** | **20K81A0590** |
| **P. MANASA** | **20K81A05A3** |
| **P. YASHWANTHI** | **20K81A05A2** |

Under the Guidance of

**Dr. B. RAJALINGAM ASSOCIATE PROFESSOR DEPARTMENT OF CSE**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

|  |
| --- |
| **St. MARTIN'S ENGINEERING COLLEGE** |
| UGC Autonomous  Affiliated to JNTUH, Approved by AICTE, Accredited by NBA & NAAC A+, ISO 9001:2008 Certified  Dhulapally, Secunderabad - 500 100  **MARCH - 2024** |

|  |  |  |
| --- | --- | --- |
| A logo with text on it  Description automatically generated | **St. MARTIN'S ENGINEERING COLLEGE** | A yellow and red logo  Description automatically generated with medium confidence |
| UGC Autonomous  NBA & NAAC A+ Accredited Dhulapally, Secunderabad - 500 100  [www.smec.ac.in](http://www.smec.ac.in/) |

**Certificate**

This is to certify that the project entitled **“Data Behaviour Analysis Using Intelligent Big Data Analytics”** is being submitted By **M.KOUSHIL (20K81A0597), L.SHIVANI (20K81A0590), P.MANASA (20K81A05A3), P.YASHWANTHI (20K81A05A2)** in fulfilment of the requirement for the award of degree of **BACHELOR OF TECHNOLOGY in DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** is recorded of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

|  |  |
| --- | --- |
| **Guide** | **Head of the Department** |
| **Dr. B. RAJALINGAM**  **Associate Professor Department of CSE** | **Dr. R. SANTHOSHKUMAR**  **Professor & Head Department of CSE** |

### Internal Examiner External Examiner

**Date:**

### Place:

|  |  |  |
| --- | --- | --- |
| A logo with text on it  Description automatically generated | **St. MARTIN'S ENGINEERING COLLEGE** | A yellow and red logo  Description automatically generated with medium confidence |
| UGC Autonomous  NBA & NAAC A+ Accredited Dhulapally, Secunderabad - 500 100  [www.smec.ac.in](http://www.smec.ac.in/) |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### DECLARATION

We, the students of ‘**Bachelor of Technology in Department of Computer Science and Engineering’**, session: 2020 - 2024**, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad,** hereby declare that the work presented in this Project Work entitled “**Data Behaviour Analysis Using Intelligent Big Data Analytics”** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. The result embodied in this project report has not been submitted to any university for award of any degree.

|  |  |
| --- | --- |
| **M. KOUSHIL** | **20K81A0597** |
| **L. SHIVANI** | **20K81A0590** |
| **P. MANASA** | **20K81A05A3** |
| **P. YASHWANTHI** | **20K81A05A2** |

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded our efforts with success.

We extend our deep sense of gratitude to Group Director, **Dr. P. SANTOSH KUMAR PATRA**, St. Martin’s Engineering College Dhulapally, for permitting us to undertake this project.

We wish to record our profound gratitude to **Dr. M. SREENIVAS RAO**, Principal, St. Martin’s Engineering College, for his motivation and encouragement.

We are also thankful to **Dr. R. SANTHOSH KUMAR**, Head of the Department, Department of Computer Science and Engineering, St. Martin’s Engineering College, Dhulapally, Secunderabad. for his support and guidance throughout our project as well as our Project Coordinators **Dr. G. JAWAHERLALNEHRU**, Associate Professor and **Dr. B. MADHAVA RAO**, Associate Professor, Department of Computer Science and Engineering for their valuable support.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Dr. B. RAJALINGAM**, Associate Professor, Department of Computer Science and Engineering, St. Martins Engineering College, Dhulapally, for his support and guidance throughout our project.

Finally, we express thanks to all those who have helped us successfully complete this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement. We express thanks to all those who have helped us in successfully completing the project.

|  |  |
| --- | --- |
| **M. KOUSHIL** | **20K81A0590** |
| **L. SHIVANI** | **20K81A0597** |
| **P. MANASA** | **20K81A05A3** |
| **P. YASHWANTHI** | **20K81A05A2** |

|  |  |
| --- | --- |
| **CONTENTS** | **PAGE NO** |
| **CERTIFICATE** | II |
| **DECLARATION** | III |
| **ACKNOWLEDGEMENT** | IV |
| **ABSTRACT** | VII |
| **LIST OF FIGURES** | VIII |
| **LIST OF ACRONYMS AND DEFINITIONS** | XI |
| **CHAPTER 1 INTRODUCTION** | 1 |
| 1.1 Introduction | 1 |
| **CHAPTER 2 LITERATURE OVERVIEW** | 4 |
| **CHAPTER 3 EXISTING SYSTEM** | 7 |
| 3.1 Apache Hadoop | 7 |
| 3.2 Drawbacks of Hadoop Apache | 9 |
| **CHAPTER 4 PROPOSED SYSTEM** | 10 |
| 4.1 Overview | 10 |
| 4.2 pyspark | 11 |
| 4.3 Apache Spark  4.4 Advantages of Apache Spark | 12 |
| **CHAPTER 5 UML DIAGRAMS** | 14 |
| 5.1 Class Diagram | 15 |
| 5.2 Use Case Diagram | 15 |
| 5.3 Sequence Diagram | 16 |
| 5.4 Activity Diagram | 17 |
| 5.5 Deployment Diagram | 18 |
| 5.6 Component diagram | 19 |
| 5.7 Dataflow Diagram | 19 |
| **CHAPTER 6 SOFTWARE ENVIRONMENT** | 21 |
| 6.1 What is Python? | 21 |
| 6.2 Advantages of Python | 21 |
| 6.3 Advantages of Python Over Other Languages | 23 |
| 6.4 Disadvantages of Python | 24 |
| 6.5 Modules Used in Project | 25 |
| **CHAPTER 7 SYSTEM REQUIREMENTS** | 34 |
| 7.1 Software requirements | 34 |
| 7.2 Hardware requirement | 34 |
| **CHAPTER 8 FUNCTIONAL REQUIREMENTS** | 35 |
| 8.1 Output Design | 35 |
| 8.2 Output Definition | 35 |
| 8.3 Input Design | 35 |
| 8.4 Input Stages | 36 |
| 8.5 Input Types | 36 |
| 8.6 Input Media | 36 |
| 8.7 User Interface Design | 37 |
| 8.8 Computer-Initiated Interfaces | 38 |
| 8.9 Performance Requirements | 38 |
| **CHAPTER 9 SOURCE CODE** | 40 |
| **CHAPTER 10 RESULT AND DISCUSSION** | 49 |
| 10.1 Modules | 49 |
| 10.2 Implementation Description | 49 |
| 10.3 Results and Discussion | 50 |
| **CHAPTER 11 CONCLUSION AND FUTURE SCOPE** | 56 |
| **CHAPTER 12 REFERENCES** | 57 |

# ABSTRACT

Intelligent big data analysis is an evolving pattern in the age of big data science and artificial intelligence (AI). Analysis of organized data has been very successful but analysing human behaviour using social media data becomes challenging. The social media data comprises a vast and unstructured format of data sources that can include likes, comments, tweets, shares, and views. Data analytics of social media data became a challenging task for companies, such as Dailymotion, that have billions of daily users and vast numbers of comments, likes, and views. Social media data is created in a significant amount and at a tremendous pace. There is a very high volume to store, sort, process, and carefully study the data for making possible decisions. This project proposes an architecture using a big data analytics mechanism to process the huge social media datasets efficiently and logically. In addition, this work employing parallel processing techniques called spark, which will create multiple threads and then distribute work between those thread to perform task parallelly and then send result back to spark. All existing algorithms works on single thread but spark will distribute works in multiple threads so its paralleling processing will be faster and suitable for big data applications.

This proposed work uses hive, spark, and Hadoop where first two will be used to store the data and spark will be used to read and process that data. Here, the dataset of reviews is gathered from Dailymotion website as .csv file and then extracting useful information such as most talk countries with many likes and then extracting likes, view, and comments from so many categories called fashion, entertainment, news etc. Finally, this project compares the execution time of processing with and without spark algorithm.

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Title** | **Page No.** |
| 4.1 | Block Diagram Of Proposed System | 10 |
| 5.1 | Class Diagram | 15 |
| 5.2 | Use Case Diagram | 16 |
| 5.3 | Sequence Diagram | 17 |
| 5.4 | Activity Diagram | 18 |
| 5.5 | Deployment Diagram | 19 |
| 5.6 | Component Diagram | 19 |
| 5.7 | Data Flow Diagram | 20 |

# LIST OF ACRONYMS AND DEFINITIONS

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ACRONYM** | **DEFINITION** |
| 01. | AI | Artificial intelligence |
| 02. | JSON | JavaScript Object Notation |
| 03. | SNA | Social Network Analysis |
| 04. | HDFS | Hadoop Distributed File System |
| 05. | DAG | Directed Acyclic Graph |
| 06. | TCP | Transmission Control Protocol |
| 07. | IP | Internet Protocol |
| 08. | SQL | Structured query language |
| 09. | GUI | Graphical User Interface |
| 10. | RDD | Resilient Distributed Dataset |
| 11. | JVM | Java Virtual Machine |
| 12.  13. | ML  DAG | Machine Learning  Directed Acyclic Graph |
| 14. | WORA | Write Once Run Anywhere |

**CH APTER - 1**

* 1. **Introduction**

**INTRODUCTION**

Intelligent big data analysis is an evolving pattern in the age of data science, big data, and artificial intelligence (AI).

* Data has been the backbone of any enterprise and will do so moving forward. Storing, extracting, and utilizing data has been key to any operations of a company. When there were no interconnected systems, data would stay and be consumed in one place. With the onset of Internet technology, the ability and requirement to share and transform data have been exploited. With the spread of social media, the nature of data has changed. Social media can consist of billions of users who continuously provide their digital traces with incredible velocity.
* As the data comes from many sources and in an unstructured format, it is not easy to handle in traditional relational databases. The need for handling unstructured data gives birth to another type of data called big data, which is unstructured, semi-structured, and unpredictable. This data is created real-time, and the amount of data is increasing daily.
* The data generated from these social media sites can take the form of text, images, videos, and documents. Only structured data can be processed and stored using an RDBMS. Big data is used to process data with a huge volume that is not possible to process using old database techniques and traditional relational databases, within an acceptable processing time.
* Big data is characterized by a large volume of data with a large variety and higher velocity Data generated moves through cables, either TV or internet, and data on local TV cables broadcast with large volume, variety, and velocity.
* The amount of data generated every day in the world is increasing exponentially. The rate of data growth is surprising, and this data comes at a speed, with variety (not necessarily structured), and contains a wealth of information that can be key for gaining an edge in competing businesses. The ability to analyse this massive amount of data brings a new era of innovation, productivity growth, and consumer surplus. “Big data is the term for a collection of data sets so large and complex that it becomes difficult to process it using traditional database management tools or data processing applications”.
* The challenges include capturing, curating, storing, searching, sharing, transferring, analysing, and visualizing this data. This section discusses the related literature.
* Big data is described with 5V’s instead of 3V (volume, velocity, and variety) and included veracity and value. The widely known big data examples are social networking sites, such as Facebook, YouTube, Dailymotion, Google, and Twitter. These sites receive a tremendous amount of data regularly with different variety, velocity, and veracity. The data include value as well.
* As the number of users increases, the amount of data also increases day by day. Users and data both keep growing on these sites, and this amount of data is a big challenge for owners and companies. This data contains all useful information that needs to be processed in a concise period. To generate more revenue and increase sales, the companies need the processed and analyzed data. The analysis of this data is not possible through relational or traditional database systems within a given time frame as the resources of this traditional system are not sufficient to accomplish processing and storing this huge amount of data; hence, Hadoop comes into the existence for fulfilling this need. In recent years, a large amount of unstructured data is generated from social media sites, such as Facebook, Twitter, Google, and some Dailymotion forums in the form of images, text, videos, and documents, to access and analyse this type of data, this work is best for practicing in the entire field.
* Twitter and Facebook are some of the most famous social media platforms, and the companies find that it is very crucial for obtaining customer feedback and maintaining goodwill.
* Dailymotion is one of the best video-sharing social media websites. It is a viral platform that publishes community feedback through its videos and comments, likes, dislikes, published videos, and subscriber information for a particular channel. The analysis of this type of data is important for acquiring knowledge about users, categories, and interests of users. Most of the production companies have their channels to share daily their movie trailers for getting user feedback before releasing them to the general public. Furthermore, individual users upload their videos to get more subscribers and views. These data points are critical for owners to analyse data to understand the views and feelings of customers about their video and service. Dailymotion has billions of users, who watch hours of videos on their site and generate a massive amount of views. It is estimated that more than a hundred hours of videos are watched per minute, and this amount is increasing day by day. To analyse such a huge amount of data, relational databases are not applicable. Users can use this data to understand how much their marketing program is effective. They can check their view counts and subscribers based on the date range that will show them the peak and downtime of views in a particular time. This will also help to check social trends and behaviour of people over time. For example, users can check how many views their videos have received and how much people have liked their video or product. They can also analyse likes and dislikes from the diverse nature of people around the world.
* In this research, we utilized Apache Spark to process datasets of social media. Apache Spark is a parallel and distributed platform that overcomes the challenges faced by the traditional processing mechanisms. The main objective of the project is to demonstrate the use of Apache Spark parallel and distributed framework technologies with other storage and processing mechanisms. The social media data generated from Dailymotion is taken under consideration in this article.

# CHAPTER - 2

## LITERATURE OVERVIEW

**A. P., Chiplunkar N. N, et al, (2018)**

authors analysed tweets streamed in real time. They have used Apache Flume to capture real-time tweets. As an analysis, they have proposed a method for finding recent trends in tweets and performed sentiment analysis on real-time tweets. The analysis is done using Hadoop ecosystem tools such as Apache Hive and Apache Pig. Performance in terms of execution time is compared for analysis of real-time tweets using Pig and Hive. From the experimental results, conclusion can be drawn that Pig is more efficient than Hive as Pig takes less time for execution than Hive.

**Rodrigues A. P., Rao A., Chiplunkar N. N, et al, (2017)**

] In this work Authors have considered a real-time streaming data on political issue which are loaded in the JSON (JavaScript Object Notation) format. In JSON format, every data is represented in key/value pairs and separated by comma. This paper is organized as follows. describes various methods used for sentiment analysis. The proposed methodology is discussed. explains and analyses the results obtained from their proposed method. Finally, the conclusion and future work are drawn.

**Blomberg J. et al, (2012)**

In this paper, they will outline the concept and execution of two social media analytics applications that use SAS to address law enforcement issues. The applications incorporate social media in very different ways. The first is as an investigative tool to find social media related to specific people. Using an adaptation of our Social Network Analysis (SNA), they present Facebook and Twitter searches of multiple suspects in an easily digestible form for the analyst. The second application focuses on monitoring social media across a much broader spectrum, looking for the proverbial “needle in a haystack”. In this example, they show how to collect and analyse historical Twitter data to try to understand precursors to dangerous activity at events, such as riots at concerts or flash mobs.

**Mahalakshmi R., Suseela S.et al (2015)**

In this research, they utilized Apache Spark to process datasets of social media. Apache Spark is a parallel and distributed platform that overcomes the challenges faced by the traditional processing mechanisms. The main objective of the project is to demonstrate the use layers. In the data collecting and storage layer, data sources in big data services are needed to be collected by corresponding equipment, and then the data in “pre-processed” state will be stored and processed in a distributed file system or database system. In the data processing layer, different processing frameworks are adopted according to different forms of data. The in-depth analysis of big data is currently mainly based on large-scale machine learning technologies, which can deeply mine the potential value of data. Finally, visualization tools are used to present results to data service consumers.

**Cui Y., Kara S., Chan K. C. et al (2020)**

Manufacturing big data ecosystem: a systematic literature review. Robotics In this paper presents a systematic literature review 21 of the state-of-the-art of big data in manufacturing. Six key drivers of big data 22 applications in manufacturing have been identified. The key drivers are system 23 integration, data, prediction, sustainability, resource sharing and hardware. 24 Based on the requirements of manufacturing, nine essential components of big 25 data ecosystem are captured. They are data ingestion, storage, computing, 26 analytics, visualization, management, workflow, infrastructure and security. 27 Several research domains are identified that are driven by available capabilities 28 of big data ecosystem.

**Grover V., Lindberg A., Benbasat I., Lyytinen K.et al**

The perils and promises of big data research in information systems. J. Assoc. Inf. Syst. 21:9. This paper addresses the key factors that cause social marketing programs (typically consisting of discrete programs or interventions, but also including broader-scale initiatives) to fail. It argues that understanding these failures offers greater insight to researchers and practitioners than publications solely focused on successes. Focus: this paper discusses the causes of the failure of social marketing programs, an area that has largely been ignored in extant research. Research Question: What causes social marketing programs to fail? Importance: As the majority of practitioner-oriented social marketing research focuses on how to develop a successful program, they identified a tendency to ignore failed programs. they suggested that both researchers and practitioners can arguably learn more useful lessons from failures rather than successes. Thus, this paper contributes to social marketing literature by exploring the key causes of social marketing failures. Methods: they conducted ten semi-structured interviews with social marketing practitioners recruited using purposive sampling

# CHAPTER - 3

## EXISTING SYSTEM

**3.1 Apache Hadoop**

**What is the Apache Hadoop?**

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

**Hadoop distributed file system:**

The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file system written in Java for the Hadoop framework. Some consider it to instead be a data store due to its lack of POSIX compliance, but it does provide shell commands and Java application programming interface (API methods that are similar to other file systems. A Hadoop instance is divided into HDFS and MapReduce. HDFS is used for storing the data and MapReduce is used for processing data. HDFS has five services as follows:

* + 1. Name Node
    2. Secondary Name Node
    3. Job tracker
    4. Data Node
    5. Task Tracker

Top three are Master Services/Daemons/Nodes and bottom two are Slave Services. Master Services can communicate with each other and in the same way Slave services can communicate with each other. Name Node is a master node and Data node is its corresponding Slave node and can talk with each other.

**Name Node:**

HDFS consists of only one Name Node that is called the Master Node. The master node can track files, manage the file system and has the meta data of all of the stored data within it. In particular, the name node contains the details of the number of blocks, locations of the data node that the data is stored in, where the replications are stored, and other details. The name node has direct contact with the client.

**Data Node:**

A Data Node stores data in it as blocks. This is also known as the slave node and it stores the actual data into HDFS which is responsible for the client to read and write. These are slave daemons. Every Data node sends a Heartbeat message to the Name node every 3 seconds and conveys that it is alive. In this way when Name Node does not receive a heartbeat from a data node for 2 minutes, it will take that data node as dead and starts the process of block replications on some other Data node.

Secondary Name Node:

This is only to take care of the checkpoints of the file system metadata which is in the Name Node. This is also known as the checkpoint Node. It is the helper Node for the Name Node. The secondary name node instructs the name node to create & send fsimage & editlog file, upon which the compacted fsimage file is created by the secondary name node.

**Job Tracker:**

Job Tracker receives the requests for Map Reduce execution from the client. Job tracker talks to the Name Node to know about the location of the data that will be used in processing. The Name Node responds with the metadata of the required processing data.

**Task Tracker:**

It is the Slave Node for the Job Tracker and it will take the task from the Job Tracker. It also receives code from the Job Tracker. Task Tracker will take the code and apply on the file. The process of applying that code on the file is known as Mapper.

Hadoop cluster has nominally a single namenode plus a cluster of datanodes, although redundancy options are available for the namenode due to its criticality. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses TCP/IP sockets for communication. Clients user emote procedure calls (RPC) to communicate with each other.

**3.2 Drawbacks of Hadoop Apache:**

While Apache Hadoop is a powerful and widely used big data processing system, it does have some drawbacks and limitations. Here are some of the main drawbacks of Hadoop:

**Complexity:**

Hadoop can be complex to set up and configure, especially for smaller organizations that don't have a dedicated IT team. It requires significant hardware resources, and there can be a steep learning curve for developers who are new to the platform.

**Performance:**

Hadoop is slower than some other big data processing systems, especially for iterative processing and real-time streaming. This is because Hadoop writes intermediate results to disk, which can slow down processing times. This can be mitigated to some extent with additional hardware and tuning, but it can still be a significant limitation for some applications.

**Scalability:**

While Hadoop is designed to be scalable, it can be difficult to add or remove nodes from a cluster, and this process can sometimes result in data loss. In addition, Hadoop clusters can be difficult to manage, especially for organizations with limited IT resources.

**Limited SQL support:**

While Hadoop includes support for SQL queries through Hive and other tools, it is not as robust as some other big data processing systems that were specifically designed for SQL queries, such as Apache Spark or Apache Flink.

**Data Security:**

Hadoop's security model is often criticized for being complex and difficult to manage. It can be challenging to secure Hadoop clusters, especially in multi-tenant environments where different users or teams need different levels of access to data.

Overall, while Hadoop is a powerful and widely used big data processing system, it does have some limitations and drawbacks that organizations need to consider when choosing a big data platform.

# CHAPTER - 4

## PROPOSED SYSTEM

**4.1 OVERVIEW**

The project is a Tkinter application that performs data behaviour analysis using both Pandas and PySpark.

* **GUI Setup:**

The code sets up a Tkinter window with various buttons and a text widget to display

output.

* **Functionality:**

initSpark(): Initializes a Spark session and loads a dataset from a CSV file using

PySpark.

* **uploadDataset**:

Allows the user to upload a dataset (assumed to be related to Daily Motion reviews) in

CSV format using a file dialog.

* **Run without Spark**:

Analyzes the dataset without using Spark. It performs computations using Pandas.

* **Run with Spark**:

Analyzes the dataset using Spark. It performs computations using PySpark.

* **Graph**

Plots a bar graph comparing the execution time of the analysis with and without Spark.

* **close:**

Closes the Tkinter application.

* **Analysis:**

Both run withoutSpark() and runwithSpark() analyze the dataset based on likes, dislikes,

view counts, comment counts, and descriptions. They also categorize data based on

countries and categories (like fashion, lifestyle, etc.). Results are displayed in the text

widget in the Tkinter window. A comparison graph of execution times is displayed using

Matplotlib.

* **Dependencies:**

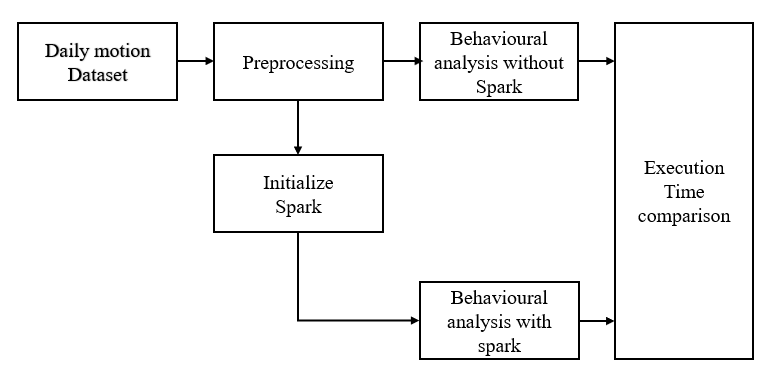
The code depends on several Python libraries including Tkinter, Matplotlib, Pandas,

PySpark, and NumPy.

* **User Interaction:**

Users can initialize Spark, upload datasets, perform analyses with or without Spark,

view results, and exit the application using buttons provided in the GUI.

****

**Fig. 4.1: Block diagram of proposed system.**

**4.2 pyspark**

PySpark is the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing. If you’re already familiar with Python and libraries such as Pandas, then PySpark is a good language to learn to create more scalable analyses and pipelines.

Apache Spark is basically a computational engine that works with huge sets of data by processing them in parallel and batch systems. Spark is written in Scala, and PySpark was released to support the collaboration of Spark and Python. In addition to providing an API for Spark, PySpark helps you interface with Resilient Distributed Datasets (RDDs) by leveraging the Py4j library.

The key data type used in PySpark is the Spark data frame. This object can be thought of as a table distributed across a cluster, and has functionality that is similar to data frames in R and Pandas. If you want to do distributed computation using PySpark, then you’ll need to perform operations on Spark data frames and no other Python data types.

Py4J is a popular library which is integrated within PySpark and allows Python to dynamically interface with JVM (Java Virtual Machine) objects. PySpark features quite a few libraries for writing efficient programs. Furthermore, there are various external libraries that are also compatible, including:

**PySparkSQL** - A PySpark library to apply SQL-like analysis on a huge amount of structured or semi-structured data. You can also use SQL queries with PySparkSQL.

**MLlib**- A wrapper over PySpark and Spark’s machine learning (ML) library. MLlib supports many machine learning algorithms for classification, regression, clustering, collaborative filtering, dimensionality reduction, and underlying optimization primitives.

**GraphFrames -** A graph processing library that provides a set of APIs for performing graph analysis efficiently, using the PySpark core and PySparkSQL. It is optimized for fast distributed computing.

**4.3 Apache Spark**

Apache Spark is a data processing framework that can quickly perform processing tasks on very large data sets, and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools. These two qualities are key to the worlds of big data and machine learning, which require the marshalling of massive computing power to crunch through large data stores. Spark also takes some of the programming burdens of these tasks off the shoulders of developers with an easy-to-use API that abstracts away much of the grunt work of distributed computing and big data processing.

From its humble beginnings in the AMP Lab at U.C. Berkeley in 2009, Apache Spark has become one of the key big data distributed processing frameworks in the world. Spark can be deployed in a variety of ways, provides native bindings for the Java, Scala, Python, and R programming languages, and supports SQL, streaming data, machine learning, and graph processing. You’ll find it used by banks, telecommunications companies, games companies, governments, and all of the major tech giants such as Apple, IBM, Meta, and Microsoft.is a data processing framework that can quickly perform processing tasks on very large data sets, and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools. These two qualities are key to the worlds of big data and machine learning, which require the marshalling of massive computing power to crunch through large data stores. Spark also takes some of the programming burdens of these tasks off the shoulders of developers with an easy-to-use API that abstracts away much of the grunt work of distributed computing and big data processing.

From its humble beginnings in the AMP Lab at U.C. Berkeley in 2009, Apache Spark has become one of the key big data distributed processing frameworks in the world. Spark can be deployed in a variety of ways, provides native bindings for the Java, Scala, Python, and R programming languages, and supports SQL, streaming data, machine learning, and graph processing. You’ll find it used by banks, telecommunications companies, games companies, governments, and all of the major tech giants such as Apple, IBM, Meta, and Microsoft.

**Assumptions of Apache Spark:**

Spark makes certain assumptions about the underlying system and data it operates on. Some of

the key assumptions of Spark are:

Data Parallelism: Spark assumes that the data is partitioned across multiple nodes in a cluster, and computations can be performed on each partition in parallel.

Memory-Based Computation: Spark assumes that the data is stored in memory or in a distributed file system, and computations can be performed in-memory for faster processing.

Resilience: Spark assumes that the underlying system may have faults or failures, and provides mechanisms to handle such failures, such as storing data redundantly across nodes and recomputing lost data.

DAG-Based Computation: Spark assumes that the computation can be represented as a Directed Acyclic Graph (DAG) of stages, where each stage contains multiple tasks that can be executed in parallel.

Immutable Data: Spark assumes that the data is immutable and cannot be modified, and instead, new transformations create new RDDs.

Lazy Evaluation: Spark assumes that transformations are evaluated lazily, i.e., they are not executed immediately, but rather when an action is triggered.

Functional Programming: Spark assumes a functional programming paradigm, where operations on RDDs are expressed as transformations and actions on the data.

These assumptions help Spark optimize its execution plan and provide high performance and fault tolerance for large-scale data processing.

**4.4 Advantages of Apache Spark:**

Speed: Spark is designed to be faster than Hadoop's MapReduce by leveraging in-memory data processing and efficient computation.

Ease of Use: Spark provides a high-level API for programming that supports multiple languages, including Scala, Java, Python, and R. This makes it easier for developers to write and maintain their code.

Flexibility: Spark provides a wide range of libraries for various data processing tasks, including SQL, machine learning, graph processing, and streaming.

Real-time Processing: Spark Streaming enables real-time processing of data streams, making it possible to process data as it arrives.

Fault Tolerance: Spark is designed to handle system failures gracefully and can recover from failures automatically.

Scalability: Spark can scale from a single server to thousands of nodes in a cluster, making it suitable for large-scale data processing.

Integration: Spark integrates with several popular big data tools and frameworks, including Hadoop, Cassandra, and Kafka.

Overall, Spark's speed, ease of use, flexibility, fault tolerance, and scalability make it a popular choice for big data processing and analytics.

# CHAPTER - 5

## UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was cre ated by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

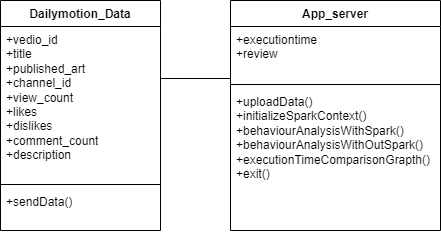
The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:** The Primary goals in the design of the UML are as follows:

* Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
* Provide extendibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development process.
* Provide a formal basis for understanding the modeling language.
* Encourage the growth of OO tools market.
* Support higher level development concepts such as collaborations, frameworks, patterns and components.
* Integrate best practices.

**5.1 Class diagram**

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



**Fig 5.1 class diagram**

**5.2 Use case diagram:** The purpose of use case diagram is to capture the dynamic aspect of a system.

Diagram

Description automatically generated

**Fig 5.2 usecase diagram**

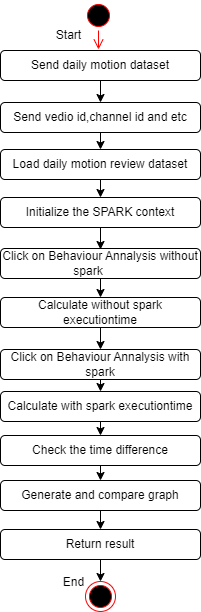
**5.3 Sequence Diagram**: Sequence diagram is an interaction diagram that details how operations are carried out.

Diagram

Description automatically generated

**Fig 5.3 sequence diagram**

**5.4 Activity diagram**: Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.



**Fig 5.4 activity diagram**

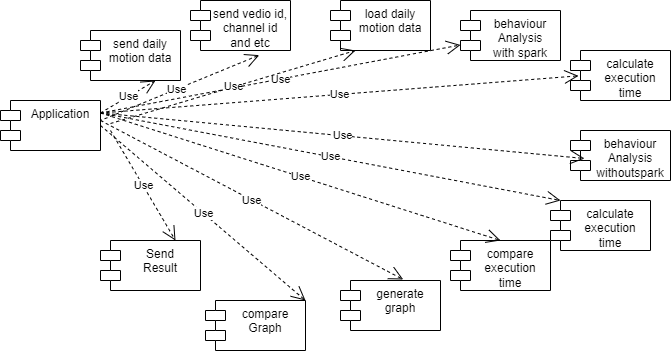
**5.5 Deployment diagram:** The deployment diagram visualizes the physical hardware on which the software will be deployed.

Diagram, schematic

Description automatically generated

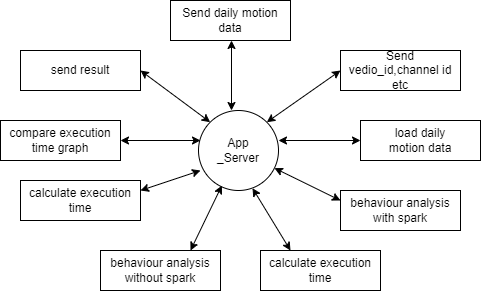
**Fig 5.5 deployment diagram**

**5.6 Component diagram:** Component diagram describes the organization and wiring of the physical components in a system.



**Fig 5.6 component diagram**

**5.7 Dataflow diagram:** A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.



**Fig 5.7 dataflow diagram**

# CHAPTER - 6

## SOFTWARE ENVIRONMENT

**6.1 What is Python?**

Below are some facts about Python.

* Python is currently the most widely used multi-purpose, high-level programming language.
* Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
* Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
* Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* Machine Learning
* GUI Applications (like Kivy, Tkinter, PyQt etc. )
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like Opencv, Pillow)
* Web scraping (like Scrapy, BeautifulSoup, Selenium)
* Test frameworks
* Multimedia

**6.2 Advantages of Python**

Let’s see how Python dominates over other languages.

* **Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

* **Extensible**

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

* **Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

* **Improved Productivity**

The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

* **IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

* **Simple and Easy**

When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

* **Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

* **Object-Oriented**

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

* **Free and Open-Source**

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

* **Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

* **Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

**6.3 Advantages of Python Over Other Languages**

* **Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

* **Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

* **Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

**6.4 Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

* **Speed Limitations**

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

* **Weak in Mobile Computing and Browsers**

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

* **Design Restrictions**

As you know, Python is dynamically typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

* **Underdeveloped Database Access Layers**

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

* **Simple**

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**6.5 Modules Used in Project**

**NumPy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and Ipython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with Ipython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

**Install Python Step-by-Step in Windows and Mac**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

**How to Install Python on Windows and Mac**

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

**Download the Correct version into the system**

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www>.python.org

A screenshot of a computer

Description automatically generated with medium confidence

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Graphical user interface, application

Description automatically generated

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Graphical user interface, application

Description automatically generated

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Graphical user interface, text

Description automatically generated

* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

**Installation of Python**

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

Graphical user interface, text, application

Description automatically generated

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

Graphical user interface, text, application, chat or text message

Description automatically generated

Step 3: Click on Install NOW After the installation is successful. Click on Close.

Graphical user interface, text, application, chat or text message

Description automatically generated

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

**Verify the Python Installation**

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.

Graphical user interface, application

Description automatically generated

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.

A screenshot of a computer

Description automatically generated with medium confidence

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

**Check how the Python IDLE works**

Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.

Application

Description automatically generated with low confidence

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save

Graphical user interface, text, application, email

Description automatically generated

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print (“Hey World”) and Press Enter.

Graphical user interface, text, application, email

Description automatically generated

You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won’t work.

# CHAPTER - 7

# SYSTEM REQUIREMENTS

**7.1 Software Requirements**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

* Python IDLE 3.7 version (or)
* Anaconda 3.7 (or)
* Jupiter (or)
* Google colab

**7.2 Hardware Requirements**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

* Operating system : Windows, Linux
* Processor : minimum intel i3
* Ram : minimum 4 GB
* Hard disk : minimum 250GB

# CHAPTER - 8

## FUNCTIONAL REQUIREMENTS

**8.1 Output Design**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely within the computer department.
* Interface outputs, which involve the user in communicating directly.

**8.2 Output Definition**

The outputs should be defined in terms of the following points:

* Type of the output
* Content of the output
* Format of the output
* Location of the output
* Frequency of the output
* Volume of the output
* Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

**8.3 Input Design**

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

**8.4 Input Stages**

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

**8.5 Input Types**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

**8.6 Input Media**

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**Error Avoidance**

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

**Error Detection**

Even though every effort is make to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

**Data Validation**

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

**8.7 User Interface Design**

It is essential to consult the system users and discuss their needs while designing the user interface:

**User Interface Systems Can Be Broadly Classified As:**

* User-initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
* Computer initiated interfaces

In the computer initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

**User Initiated Interfaces**

User initiated interfaces fall into two approximate classes:

* Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
* Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

**8.8 Computer-Initiated Interfaces**

The following computer–initiated interfaces were used:

* The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
* Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

**Error Message Design**

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

**8.9 Performance Requirements**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system
* The existing system is completely dependent on the user to perform all the duties.

.

# CHAPTER – 9

# SOURCE CODE

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

import matplotlib.pyplot as plt

import numpy as np

from tkinter import ttk

from tkinter import filedialog

import pandas as pd

import os

from pyspark.sql import SparkSession

from pyspark import SparkConf, SparkContext

import time

main = Tk()

main.title("Data Behavior Analysis Using Intelligent Big Data Analytics")

main.geometry("1300x1200")

global filename

global execution\_time

global df\_frame

def initSpark():

global df\_frame

spark = SparkSession.builder.master("local[1]").appName("HDFS").getOrCreate()

sparkcont = SparkContext.getOrCreate(SparkConf().setAppName("HDFS"))

logs = sparkcont.setLogLevel("ERROR")

dataset = spark.read.format("csv").option("header",True).load(r"D:\SAK INFORMATICS\KPRIT\Saturday\7. DataBehaviour\Dataset\dailymotion.csv")

frame = dataset[['likes','dislikes','view\_count','comment\_count','description']]

frame = frame.toPandas()

df\_frame = frame.iloc[:,0:5].values

text.delete('1.0', END)

text.insert(END,"SPARK initialization Done!")

def uploadDataset():

global filename

global dataset

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="Dataset")

text.insert(END,filename+" loaded\n\n")

dataset = pd.read\_csv(filename,encoding='iso-8859-1',usecols=['likes','dislikes','view\_count','comment\_count','description'])

dataset.fillna(0, inplace = True)

text.insert(END,str(dataset.head()))

def runwithoutSpark():

global dataset

global execution\_time

text.delete('1.0', END)

execution\_time = []

start = time.time()

countries = ['pakistan','united states','england','india','sri lanka', 'italy', 'china']

country\_comments = {}

country\_likes = {}

for i in range(len(countries)):

country\_comments[countries[i]] = 0

country\_likes[countries[i]] = 0

views\_dict = {}

comment\_dict = {}

likes\_dict = {}

category = ['fashion', 'lifestyle', 'music', 'news', 'comedy', 'entertainment', 'games', 'cartoons']

for i in range(len(category)):

views\_dict[category[i]] = 0

comment\_dict[category[i]] = 0

likes\_dict[category[i]] = 0

dataset = dataset.values

for i in range(len(dataset)):

like\_value = dataset[i,0]

dislike\_value = dataset[i,1]

comment = str(dataset[i,4])

comment = comment.strip().lower()

for j in range(len(countries)):

if countries[j] in comment:

if like\_value > dislike\_value:

country\_likes[countries[j]] = country\_likes.get(countries[j]) + 2

country\_comments[countries[j]] = country\_comments.get(countries[j]) + 1

data = []

columns = ["Type","Countries","Count"]

for i in range(len(countries)):

data.append(["Comments",countries[i],country\_comments.get(countries[i])])

data.append(["Likes",countries[i],country\_likes.get(countries[i])])

output = pd.DataFrame(data,columns=columns)

#text.insert(END,str(output)+"\n\n")

fig, axs = plt.subplots(1,2)

output.pivot("Type","Countries", "Count").plot(kind='bar',ax=axs[0])

for i in range(len(dataset)):

like\_value = dataset[i,0]

dislike\_value = dataset[i,1]

view\_count = dataset[i,2]

comment = str(dataset[i,4])

comment = comment.strip().lower()

for j in range(len(category)):

if category[j] in comment:

comment\_dict[category[j]] = comment\_dict.get(category[j]) + 1

if view\_count > 0:

views\_dict[category[j]] = views\_dict.get(category[j]) + 1

if like\_value > dislike\_value:

likes\_dict[category[j]] = likes\_dict.get(category[j]) + 1

data = []

columns = ["Type", "Category", "Count"]

for i in range(len(category)):

data.append([category[i],"Views",views\_dict.get(category[i])])

data.append([category[i],"Comments",comment\_dict.get(category[i])])

data.append([category[i],"Likes",likes\_dict.get(category[i])])

output = pd.DataFrame(data,columns=columns)

end = time.time()

execution\_time.append(end-start)

text.insert(END,"Without spark Execution Time : "+str(end-start)+"\n\n")

text.update\_idletasks()

output.pivot("Type","Category","Count").plot(kind='bar',ax=axs[1])

plt.show()

def runwithSpark():

text.delete('1.0', END)

global df\_frame

#print(df\_frame[0])

global execution\_time

text.delete('1.0', END)

start = time.time()

countries = ['pakistan','united states','england','india','sri lanka', 'italy', 'china']

country\_comments = {}

country\_likes = {}

for i in range(len(countries)):

country\_comments[countries[i]] = 0

country\_likes[countries[i]] = 0

views\_dict = {}

comment\_dict = {}

likes\_dict = {}

category = ['fashion', 'lifestyle', 'music', 'news', 'comedy', 'entertainment', 'games', 'cartoons']

for i in range(len(category)):

views\_dict[category[i]] = 0

comment\_dict[category[i]] = 0

likes\_dict[category[i]] = 0

for i in range(len(dataset)):

like\_value = dataset[i,0]

dislike\_value = dataset[i,1]

view\_count = dataset[i,2]

comment = str(dataset[i,4])

comment = comment.strip().lower()

for j in range(len(category)):

if category[j] in comment:

comment\_dict[category[j]] = comment\_dict.get(category[j]) + 1

if view\_count > 0:

views\_dict[category[j]] = views\_dict.get(category[j]) + 1

if like\_value > dislike\_value:

likes\_dict[category[j]] = likes\_dict.get(category[j]) + 1

for j in range(len(countries)):

if countries[j] in comment:

if like\_value > dislike\_value:

country\_likes[countries[j]] = country\_likes.get(countries[j]) + 2

country\_comments[countries[j]] = country\_comments.get(countries[j]) + 1

data = []

columns = ["Type","Countries","Count"]

for i in range(len(countries)):

data.append(["Comments",countries[i],country\_comments.get(countries[i])])

data.append(["Likes",countries[i],country\_likes.get(countries[i])])

output = pd.DataFrame(data,columns=columns)

#text.insert(END,str(output)+"\n\n")

fig, axs = plt.subplots(1,2)

output.pivot("Type","Countries", "Count").plot(kind='bar',ax=axs[0])

data = []

columns = ["Type", "Category", "Count"]

for i in range(len(category)):

data.append([category[i],"Views",views\_dict.get(category[i])])

data.append([category[i],"Comments",comment\_dict.get(category[i])])

data.append([category[i],"Likes",likes\_dict.get(category[i])])

output = pd.DataFrame(data,columns=columns)

#text.insert(END,str(output))

end = time.time()

execution\_time.append(end-start)

text.insert(END,"With spark Execution Time : "+str(end-start)+"\n\n")

text.update\_idletasks()

output.pivot("Type","Category","Count").plot(kind='bar',ax=axs[1])

plt.show()

def graph():

global execution\_time

height = execution\_time

bars = ('Without Spark Execution Time','With Spark Execution Time')

y\_pos = np.arange(len(bars))

plt.bar(y\_pos, height)

plt.xticks(y\_pos, bars)

plt.title("With & Without Spark Execution Time Comparison")

plt.show()

def close():

main.destroy()

font = ('times', 15, 'bold')

title = Label(main, text='Data Behavior Analysis Using Intelligent Big Data Analytics')

title.config(bg='darkviolet', fg='gold')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

ff = ('times', 12, 'bold')

initButton = Button(main, text="Initialize Spark Context", command=initSpark)

initButton.place(x=20,y=100)

initButton.config(font=ff)

uploadButton = Button(main, text="Upload Daily Motion Reviews Dataset", command=uploadDataset)

uploadButton.place(x=20,y=150)

uploadButton.config(font=ff)

withoutSparkButton = Button(main, text="Behaviour Analysis without SPARK", command=runwithoutSpark)

withoutSparkButton.place(x=20,y=200)

withoutSparkButton.config(font=ff)

sparkButton = Button(main, text="Behaviour Analysis with SPARK", command=runwithSpark)

sparkButton.place(x=20,y=250)

sparkButton.config(font=ff)

graphButton = Button(main, text="Execution Time Comparison Graph", command=graph)

graphButton.place(x=20,y=300)

graphButton.config(font=ff)

closeButton = Button(main, text="Exit", command=close)

closeButton.place(x=20,y=350)

closeButton.config(font=ff)

font1 = ('times', 12, 'bold')

text=Text(main,height=30,width=110)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=360,y=100)

text.config(font=font1)

main.config(bg='forestgreen')

main.mainloop()

# CHAPTER – 10

# RESULT AND DISCUSSION

**10.1 MODULES**

* Initialize Spark Context: using this module we will initialize SPARK CONTEXT for parallel processing.
* Upload Daily Motion Reviews Dataset: using this module we will upload dataset file path to application.
* Behaviour Analysis without SPARK: using this module we will analyse human behaviour such as their LIKES, DISLIKES from their reviews without using SPARK technology and then capture its execution time.
* Behaviour Analysis with SPARK: using this module we will perform same task of behaviour analysis by using SPARK technology and then capture its execution time.
* Execution Time Comparison Graph: using this module we will plot execution time comparison between without and with SPARK processing.

**10.2 IMPLEMENTATION DESCRIPTION**

This Python is a GUI application built using Tkinter for performing data behavior analysis on a dataset, both with and without the use of Apache Spark.

* **GUI Setup:**

The GUI window is created using Tkinter (main = Tk ()). Various widgets such as buttons, labels, and a text widget are placed within the window to facilitate user interaction.

* **Functionality:** initSpark(): Initializes Spark session and context. It reads a CSV file using Spark, selects specific columns, converts the Spark DataFrame to a Pandas DataFrame, and stores it in df\_frame.
* **Upload Dataset:** Allows the user to upload a dataset from a file. The selected file is read using Pandas and displayed in the text widget.
* **Run without Spark:** Conducts data analysis without Spark. It processes the dataset to analyze comments, likes, and dislikes based on predefined categories and countries. Results are displayed using Pandas DataFrames and matplotlib plots.
* **Run with Spark**: Conducts data analysis with Spark. It performs similar data processing as runwithoutSpark() but uses the Spark DataFrame (df\_frame) for computation.
* **Graph:** Plots a bar graph comparing the execution time of analysis with and without Spark.
* **Close:** Closes the application window.
* **Data Processing**: Data analysis involves parsing through the dataset to extract relevant information such as likes, dislikes, view counts, comment counts, and descriptions. Analysis is conducted based on predefined categories (e.g., fashion, lifestyle) and countries (e.g., Pakistan, United States). For both Spark and non-Spark analysis, dictionaries are used to store counts for various categories and countries. Matplotlib is utilized to visualize the results through bar plots.
* **Execution Time Tracking**: Execution times for both Spark and non-Spark analysis are tracked and displayed to the user. The execution time list stores the time taken for each analysis.
* **User Interaction:** Buttons are provided for various actions such as initializing Spark, uploading dataset, running analysis with and without Spark, and plotting execution time comparison. The text widget serves as an output console, displaying messages, dataset information, and analysis results.
* **Styling:** Fonts, colors, and widget placements are adjusted to enhance the visual appeal and usability of the application.

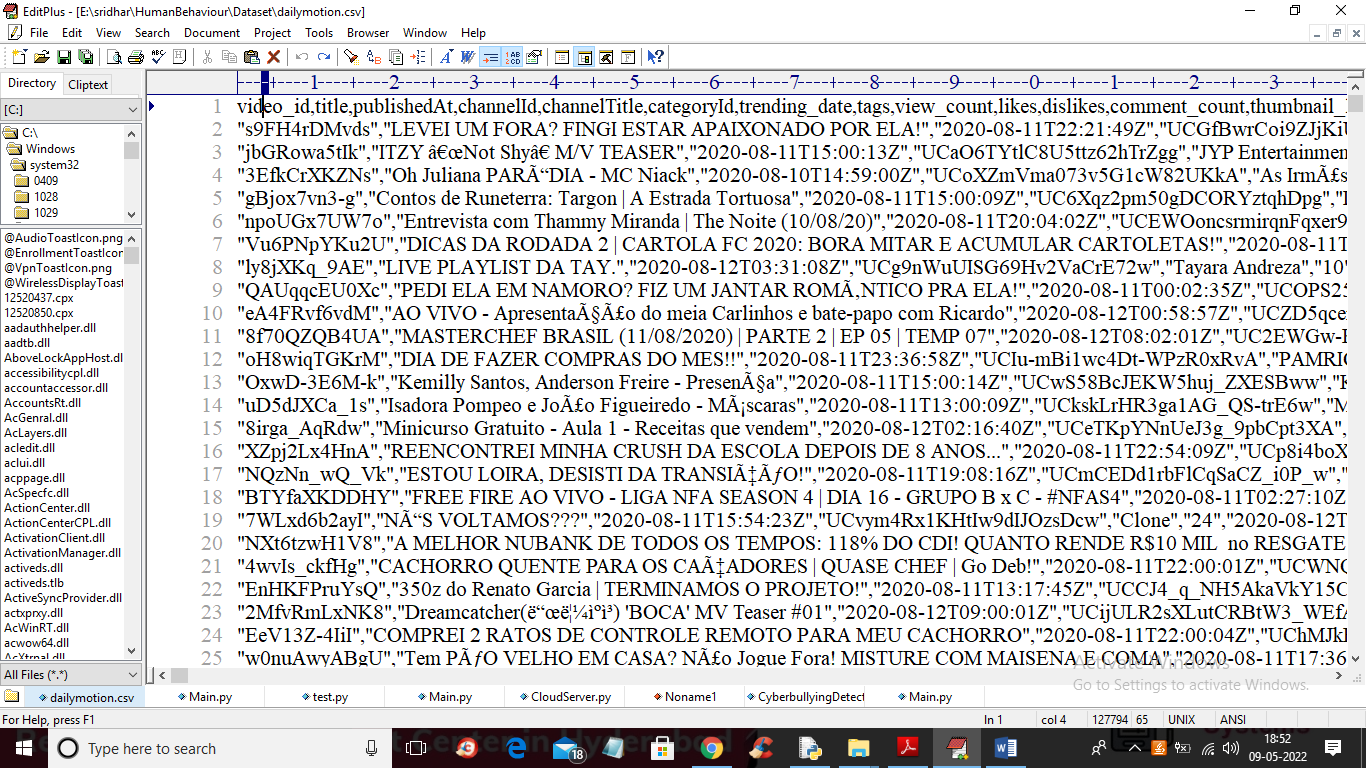
Overall, this script provides a user-friendly interface for conducting data behavior analysis on a dataset, allowing users to choose between Spark and non-Spark implementations based on their preferences and requirements.

**10.3 RESULTS AND DESCRIPTION**

Now-a-days almost all peoples are using social media to express their views and by analysing this view we can predict person behaviour as their view often describe their personality but this social media contains reviews as TWEETS, POSTS in unstructured format and everyday this unstructured data gather in terabytes and if we want to extract meaningful information such as famous brand, powerful leader, most trending entertainment then this terabytes data processing may take huge time with traditional algorithms so author of this paper employing parallel processing techniques called SPARK.

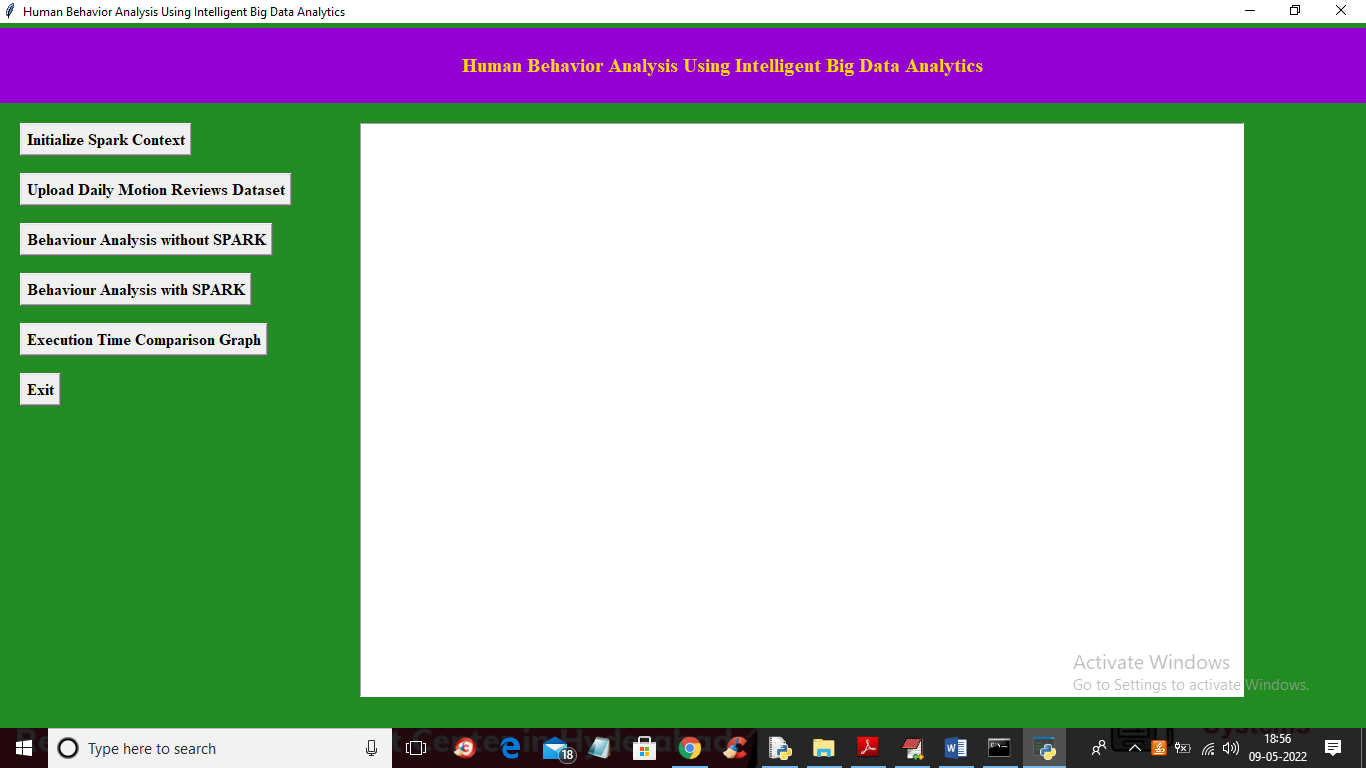
SPARK will create multiple threads and then distribute work between those thread to perform task parallelly and then send result back to SPARK. All existing algorithms works on single thread but SPARK will distribute works in multiple threads so its paralleling processing will be faster and suitable for BIG DATA applications.

So, this project used HIVE, SPARK and HADOOP where HIVE and HADOOP will store data and SPARK will read and process that data. In proposed work, we have gathered reviews from DAILYMOTION website as CSV file and then extracting useful information such as MOST TALK COUNTRIES with many LIKES and then extracting LIKES, VIEW and COMMENTS from so many categories called FASHION, ENTERTAINMENT, NEWS etc. We have compared the execution time of SPARK processing and without spark processing and this experiment proves that SPARK is faster than traditional single thread processing. We are using below CSV dataset of DAILYMOTION website to extract useful information

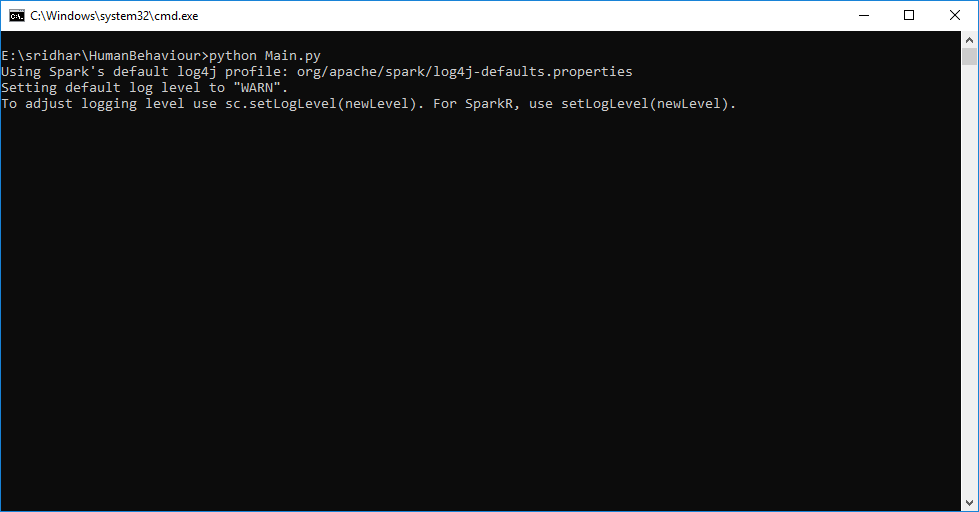


In above dataset screen first row contains dataset column names and remaining are the dataset values.

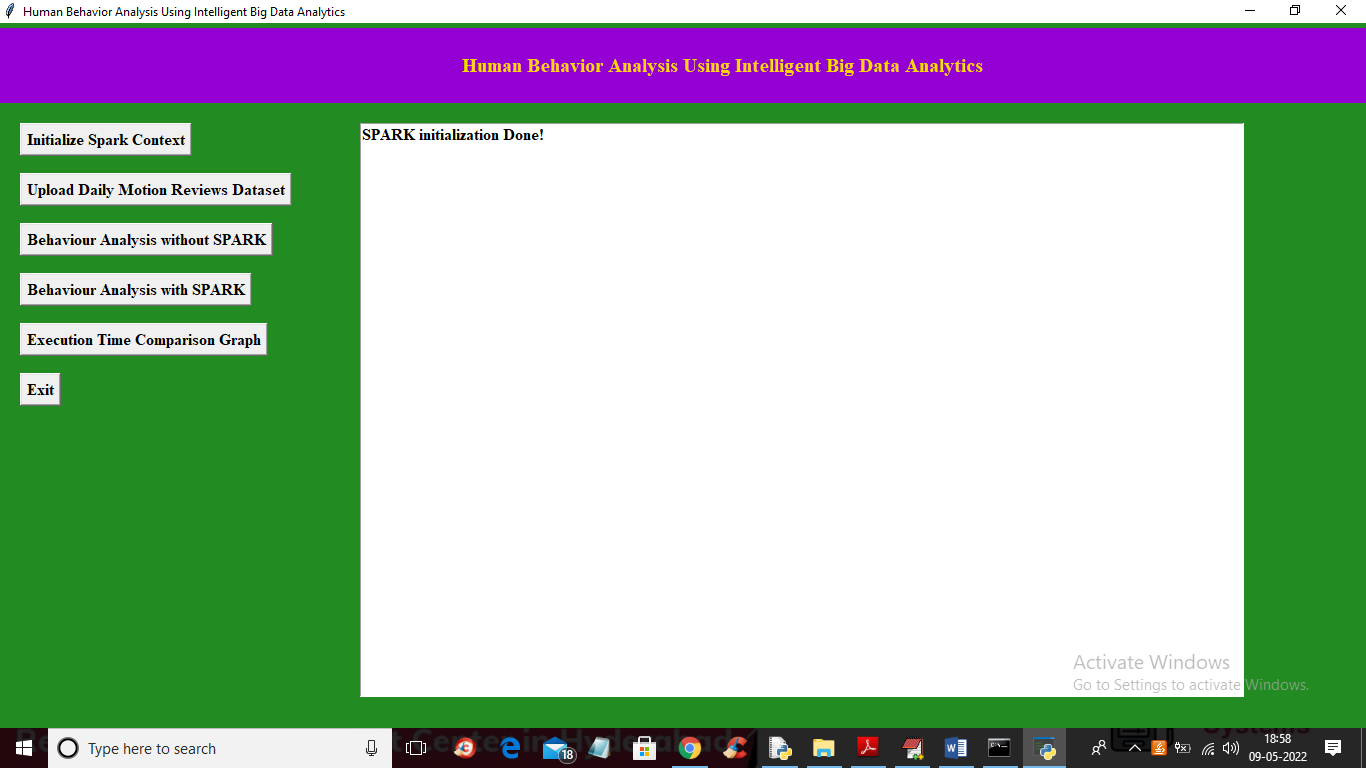
**SCREEN SHOTS**



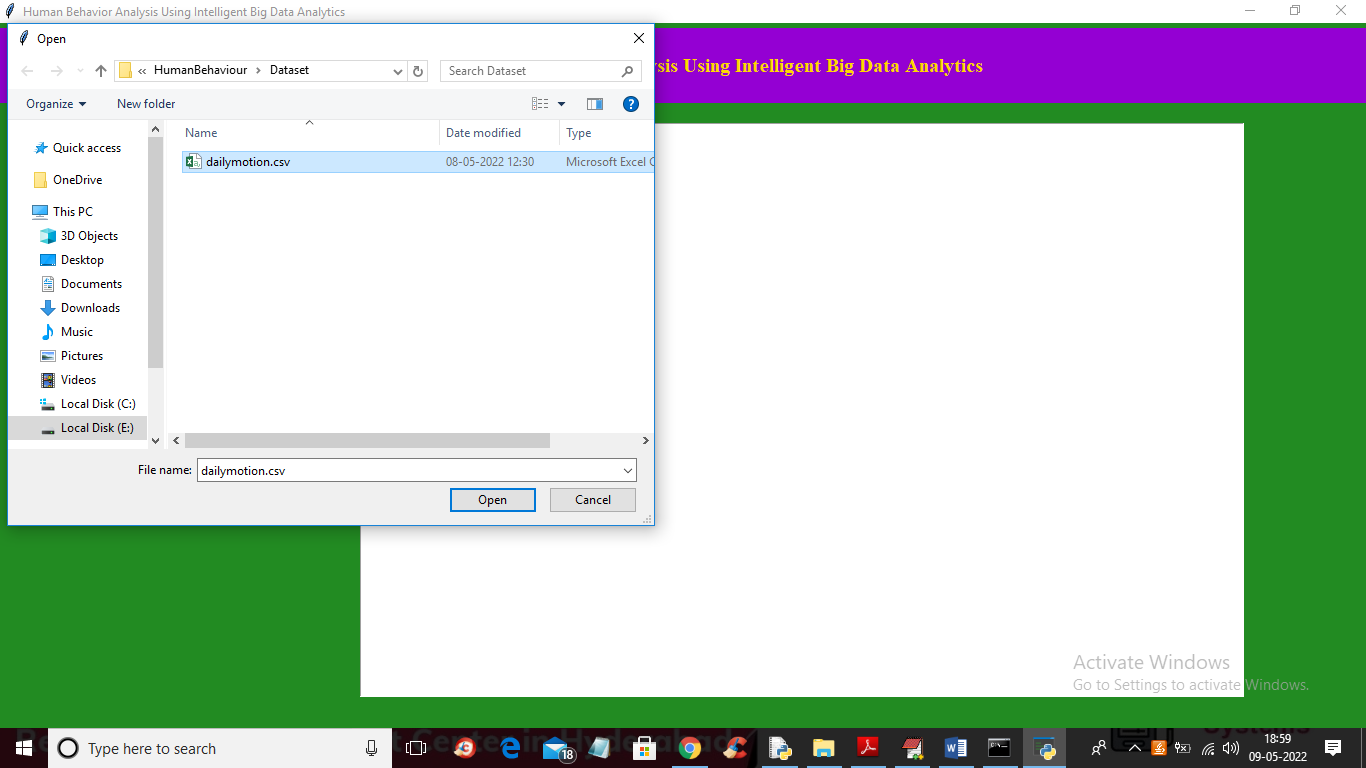
In above screen click on ‘Initialize Spark Context’ button to setup spark context and get below output after initialization



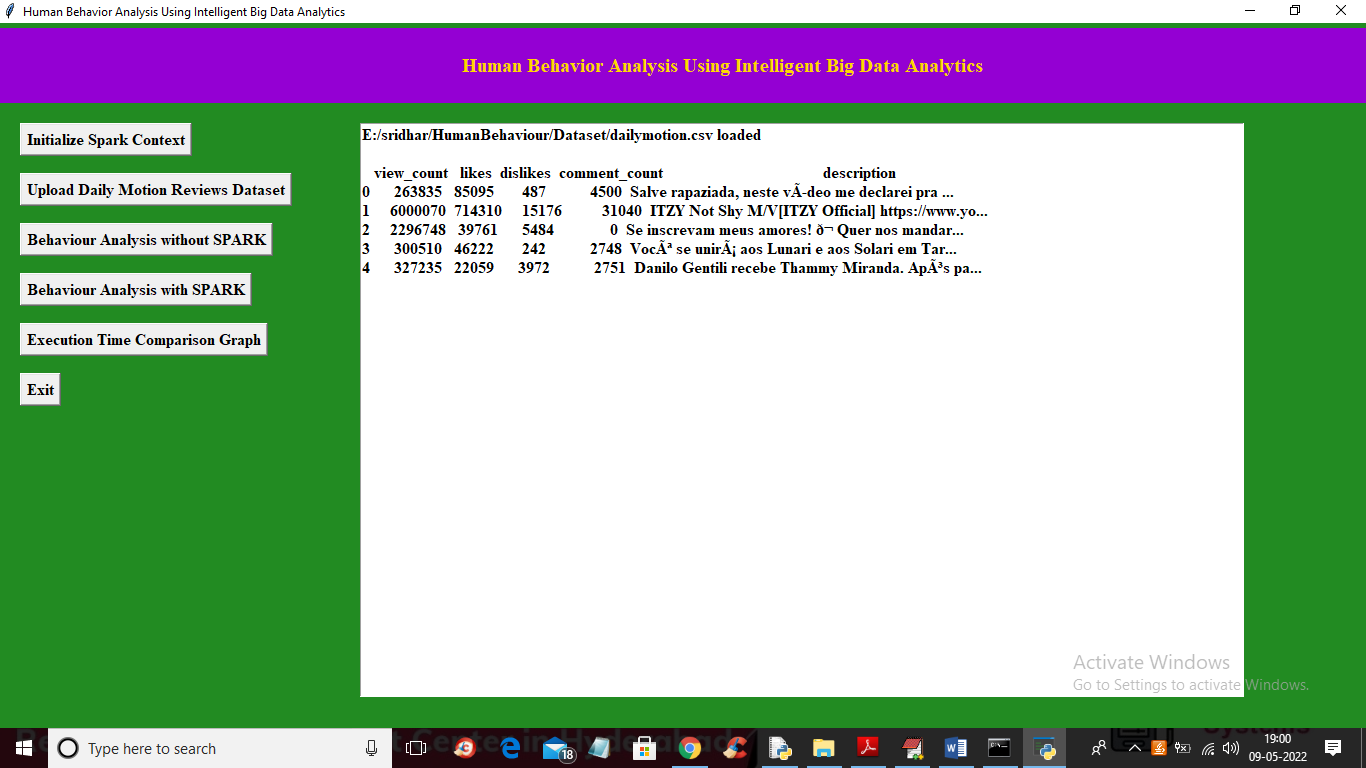
In above screen we can see SPARK object is getting initialized and after initialization will get below output



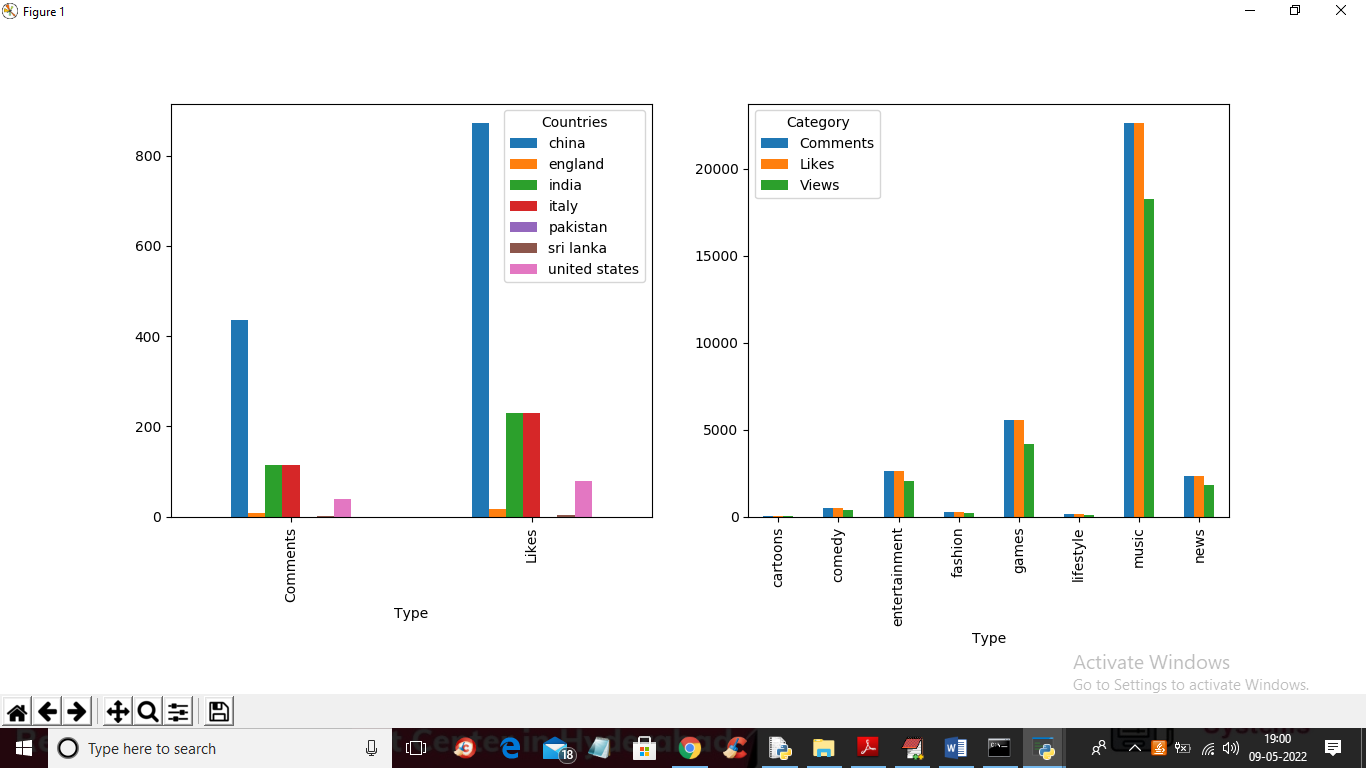
In above screen SPARK context is initialized and now click on ‘Upload Daily Motion Reviews Dataset’ button to upload dataset and get below output



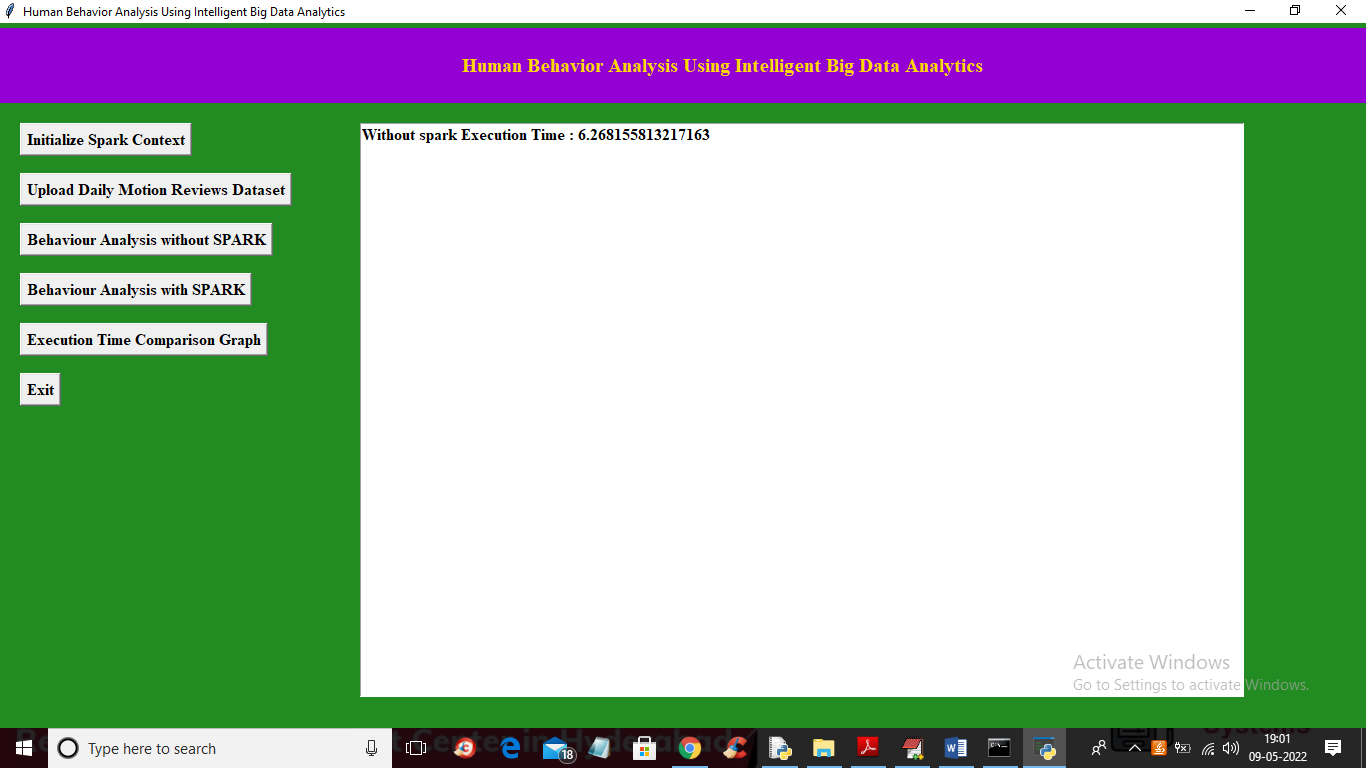
In above screen selecting and uploading dataset file and then click on ‘Open’ button to load dataset and get below output



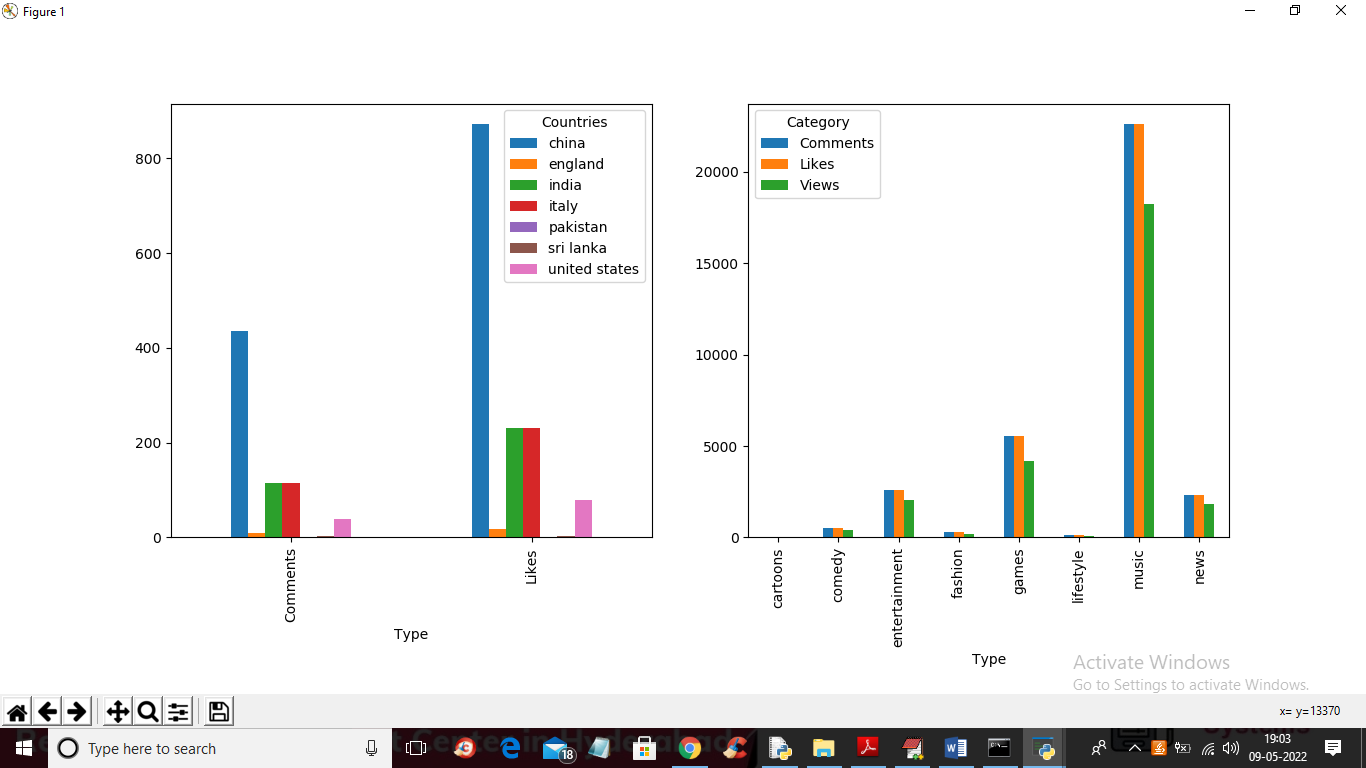
In above screen dataset loaded and now click on ‘Behaviour Analysis without Spark’ button to get below output



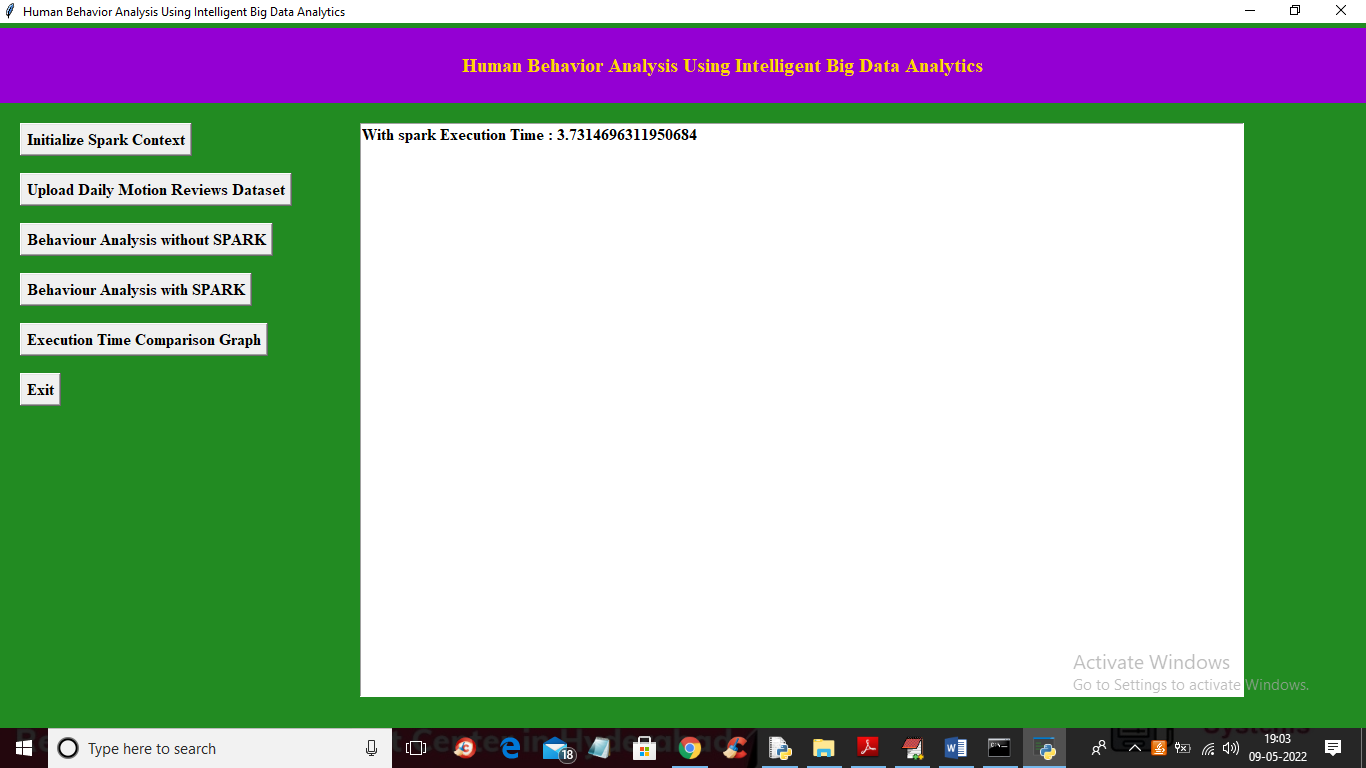
In above graph we have identify behaviour of persons like on which fashion or country they talk most with more LIKES and below screen showing execution time of WITHOUT spark processing



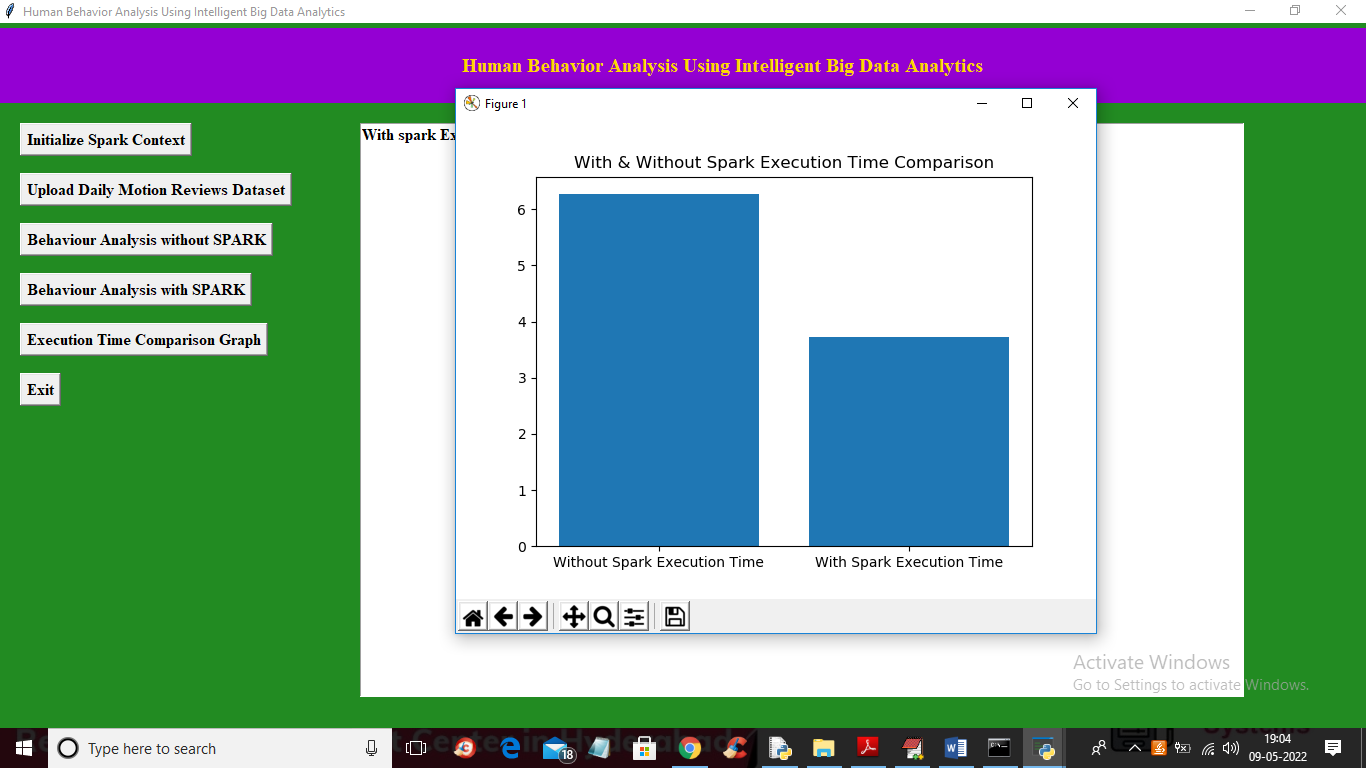
In above screen without spark processing it took 6.26 seconds and now click on ‘Behaviour Analysis with SPARK’ button to process same data using SPARK and get same output



In above screen with SPARK also we got same output but the difference is execution time and in below screen I am showing SPARK execution to process same data



In above screen SPARK took 3.73 seconds which is lesser than existing single thread processing and now click on ‘Execution Time Comparison Graph’ button to get below output



In above graph x-axis represents technique names and y-axis represents execution time and we can see SPARK processing took less execution time so it’s faster than traditional processing so BIG DATA processing with SPARK can be efficient

# CHAPTER – 11

# CONCLUSION AND FUTURE SCOPE

In conclusion, the proposed architecture leveraging intelligent big data analytics, particularly through the integration of Hive, Spark, and Hadoop, presents a promising solution for efficiently processing vast amounts of social media data. By addressing the challenges associated with analyzing unstructured social media data, such as likes, comments, tweets, and shares, this approach enables companies like Dailymotion to derive valuable insights from their massive user-generated content.

The utilization of Spark for parallel processing offers significant advantages in terms of scalability and performance compared to traditional single-threaded algorithms. By distributing tasks across multiple threads, Spark enables faster processing of large datasets, thereby enhancing the speed and efficiency of data analytics tasks.

The specific application of this architecture in analyzing Dailymotion's dataset, including identifying popular content categories and extracting relevant metrics like likes, views, and comments, demonstrates its effectiveness in deriving actionable insights from social media data.

However, there are several avenues for future research and improvement. Firstly, continued optimization of the architecture to further enhance performance and scalability is essential, particularly as the volume of social media data continues to grow exponentially. Additionally, exploring advanced machine learning and AI techniques for sentiment analysis, user behavior prediction, and content recommendation could further enrich the insights derived from social media data.

Furthermore, enhancing the real-time processing capabilities of the architecture to enable instantaneous analysis and response to emerging trends and user interactions represents another exciting direction for future development.

In conclusion, while the proposed architecture represents a significant step forward in intelligent big data analytics for social media data, ongoing research and innovation will be crucial for unlocking its full potential in driving informed decision-making and enhancing user experiences in the dynamic landscape of social media platforms.

# CHAPTER – 12

**REFERENCES**

[1] P., Chiplunkar N. N. (2018). Real-time twitter data analysis using Hadoop ecosystem. Cogent Eng. 5:1534519. 10.1080/23311916.2018.1534519.

[2] Rodrigues A. P., Rao A., Chiplunkar N. N. (2017). “Sentiment analysis of real time Twitter data using big data approach,” in Proceedings of the 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution

[3] Blomberg J. (2012). Twitter and Facebook Analysis: It’s Not Just for Marketing Anymore, Vol. 309. Denver, CO: SAS Global Forum.

[4] Mahalakshmi R., Suseela S. (2015). Big-SoSA: social sentiment analysis and data visualization on big data. Int. J. Adv. Res. Comp. common. Eng

[5] Xia Q., Yin X., He J., Chen F. (2018). Real-time recognition of human daily motion with smartphone sensor. Int. J. Performability Eng. 14 593–602.

[6] Barros, C. P., and Couto, E. (2013) This paper considers productivity changes in European airlines between 2000 and 2011

[7] Lee N. R., Kotler P. (2011). Social Marketing: Influencing Behaviours for Good. Thousand Oaks, CA: Sage Publications.

[8] Cui Y., Kara S., Chan K. C. (2020). Manufacturing big data ecosystem: a systematic literature review. Robotics Compute. Integr. Manuf. 62:101861. 10.1016/j.rcim.2019.101861

[9] Grover V., Lindberg A., Benbasat I., Lyytinen K.et al (2020). The perils and promises of big data research in information systems. J. Assoc. Inf. Syst. 21:9.

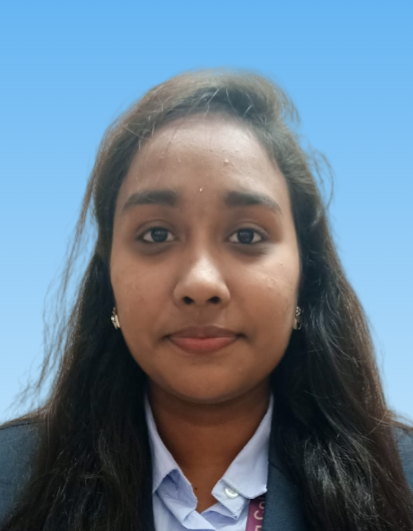
[10] Wang J., Yang Y., Wang T., Sherratt R. S., Zhang J.et al [ (2020). Big data service architecture: a survey. J. Internet Technol. 21 393–405

**STUDENTS PROFILE**



I am Muthyala Koushil, a dedicated student currently pursuing a Bachelor's degree in Computer Science and Engineering at St. Martins Engineering College. Throughout my academic journey, I have consistently demonstrated a fervent dedication to expanding my knowledge and refining my skills, evident in my current CGPA of 7.6.My educational voyage commenced at St. Joseph's High School, where I completed my 10th-grade education with an impressive aggregate of 92%. Building upon this strong foundation, I furthered my studies at Sri Chaitanya Junior College, culminating with an outstanding aggregate of 94% in my intermediate studies.The realm of programming languages has always held a profound fascination for me, particularly HTML, CSS, JavaScript, and Java. Exploring the intricacies of these languages and their practical applications energizes me, and I am continuously augmenting my proficiency and expertise in these domains. I pride myself on being a swift learner, eagerly embracing new challenges and opportunities to deepen my understanding of these programming languages.Beyond my academic pursuits, I am deeply passionate about sports, with cricket holding a special place in my heart. Engaging in sports not only ensures my physical well-being but also instills in me invaluable lessons in discipline, teamwork, and perseverance. I firmly believe in the transformative power of sports to nurture personal growth and overall well-being.As I advance in my studies, my objective is to consistently strive for excellence and remain abreast of the latest advancements in the ever-evolving field of computer science and engineering. Motivated by an insatiable thirst for knowledge and a desire to effect positive change through technological innovation, I am committed to making meaningful contributions to the world.

I am L. Shivani a motivated student pursuing a bachelor’s degree in computer science and engineering from St. Martins Engineering College. My academic journey thus far has been characterized by an unwavering commitment to knowledge acquisition and skill refinement, as evidenced by my current CGPA of 7.6. My academic journey began at army public School, where I completed my 10th grade education with an aggregate of 92%. Building upon that foundation, I enrolled in Sri Chaitanya Junior College to pursue my intermediate studies and completed with an aggregate 94. My educational background has provided me with a comprehensive understanding of various aspects of computer science and engineering. However, my true passion lies in programming, particularly in languages like C, Java, Python, HTML, and SQL. What excites me most is the opportunity to apply my skills to innovate and develop new solutions. I thrive on challenges and constantly seek to push the boundaries of what's achievable. Outside of my academic interests, I find solace and inspiration in reading, which not only provides balance to my life but also enhances my creativity and problem-solving abilities. My overarching goal is to make a meaningful impact in the field of computer science and engineering. In conclusion, I am enthusiastic about embarking on a fulfilling career journey, eager to expand my knowledge, explore new opportunities, and contribute positively to the world through my endeavors.

I am P.Manasa, a driven student currently pursuing a bachelor’s degree in computer science and engineering at St. Martins Engineering College. Throughout my academic journey, I have demonstrated a steadfast dedication to learning and honing my skills, reflected in my current CGPA of 7.6. My educational path began at Army Public School, where I achieved a remarkable aggregate of 92% in my 10th grade education. Continuing on this trajectory, I enrolled at Sri Chaitanya Junior College for my intermediate studies, culminating with an impressive aggregate of 94%.My educational background has equipped me with a comprehensive understanding of various facets of computer science and engineering. However, my passion truly lies in programming, with a particular focus on languages such as C, Java, Python, HTML, and SQL. The prospect of utilizing my skills to innovate and create novel solutions is what excites me the most. I thrive in challenging environments and continually strive to push the boundaries of what is achievable.Beyond academia, I find solace and inspiration in reading, which not only brings balance to my life but also enhances my creativity and problem-solving abilities. My overarching objective is to make a meaningful impact in the realm of computer science and engineering.In conclusion, I am enthusiastic about embarking on a fulfilling career journey. I am eager to broaden my knowledge, explore new avenues, and contribute positively to the world through my endeavors.



I am P. Yashwanthi, an ambitious student currently pursuing a Bachelor's degree in Computer Science and Engineering at St. Martins Engineering College. From the outset of my academic journey, I've displayed an unwavering commitment to learning and refining my skills, evident in my current CGPA of 8.6.My educational voyage commenced at Narayana etechno School, where I attained a remarkable aggregate of 97% in my 10th-grade education. Building on this foundation, I enrolled at Sri Chaitanya Junior College for my intermediate studies, achieving an impressive aggregate of 96%. I've delved deep into languages such as C, Java, Python, HTML, and SQL, finding immense joy and satisfaction in crafting elegant solutions to complex problems. The allure of harnessing these skills to innovate and devise novel solutions fuels my enthusiasm.I thrive in environments that challenge me to push the boundaries of what's achievable. Whether it's tackling intricate algorithms or architecting robust software systems, I relish the opportunity to stretch my intellectual faculties and expand my capabilities.

My overarching objective is to carve a meaningful niche in the realm of computer science and engineering. I am deeply committed to leveraging my skills and knowledge to address pressing societal challenges and drive positive change. In conclusion, I am enthusiastic about the prospects that lie ahead. I am poised to broaden my horizons, explore new avenues, and steadfastly pursue excellence in the dynamic field of computer science and engineering.