Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1052-1066

Real-Time Social Media Sentiment Analysis Using Apache Spark and Streamlit for Public Opinion Monitoring

[1]Koyana Dekate, [2]Mr. Pawan Kumar Jaiswal
[1]Student, [2]Assistant Professor
[1,2]Amity University Chhattisgarh
[1]koynadekate07@gmail.com, [2]pkumar@rpr.amity.edu

Abstract
Social media platforms have become one of the most powerful sources for understanding public opinion, especially in areas such as politics, business, public policy, and social issues. Millions of users express their views daily through tweets, comments, posts, and reviews, creating a large amount of unstructured textual data. Analyzing this data manually is difficult, time-consuming, and inefficient. This research paper presents a Real-Time Social Media Sentiment Analysis System using Apache Spark and Streamlit for monitoring and analyzing public opinion effectively. The proposed system collects social media data from platforms such as Twitter, Facebook, Instagram, and YouTube, and processes the text using Natural Language Processing (NLP) techniques. Text preprocessing is performed to remove unwanted symbols, URLs, hashtags, punctuation, and stop words. Sentiment analysis is carried out using TextBlob to calculate polarity scores and classify the text into Positive, Negative, and Neutral categories. Apache Spark is integrated to improve performance through distributed computing and in-memory data processing, making the system scalable for large datasets. The frontend dashboard is developed using Streamlit, which provides an interactive interface for live tweet simulation, sentiment distribution graphs, platform-wise analysis, and real-time reporting. The system helps users understand public reactions quickly and supports better decision-making in political campaigns and social trend analysis. The results show that the system is fast, cost-effective, and suitable for both academic and practical applications. Future improvements may include live API integration, Apache Kafka streaming, multilingual analysis, and deep learning models such as BERT and LSTM for higher accuracy.

Keywords: Social Media Sentiment Analysis, Apache Spark, Streamlit, Natural Language Processing (NLP), TextBlob, Real-Time Data Processing, Public Opinion Monitoring, Big Data Analytics, Political Campaign Analysis, Sentiment Classification

1. Introduction

In the digital era, social media platforms have become one of the most important sources of communication and public expression. Platforms such as Twitter, Facebook, Instagram, and YouTube allow users to share their opinions, emotions, and reactions instantly on political events, government policies, products, services, and social issues. These platforms generate a massive amount of unstructured textual data every second, which can be used to understand public sentiment and behavior.

Traditional methods of collecting public opinion, such as surveys and interviews, are often slow, expensive, and limited in scope. In contrast, social media provides real-time access to public reactions from a large and diverse audience. However, manually analyzing thousands of posts, tweets, and comments is not practical. This creates the need for an automated system that can process large volumes of data quickly and accurately.

Sentiment Analysis is a Natural Language Processing (NLP) technique used to determine whether a piece of text expresses a positive, negative, or neutral opinion. It helps organizations, researchers, and political campaign teams understand how people feel about a particular topic. When combined with Big Data technologies such as Apache Spark, sentiment analysis becomes faster and more scalable for handling large datasets.

This research paper proposes a Real-Time Social Media Sentiment Analysis System using Apache Spark and Streamlit**.** The system uses TextBlob for sentiment classification, Apache Spark for large-scale data processing, and Streamlit for creating an interactive dashboard for live visualization and reporting. The objective is to provide a simple, efficient, and scalable platform for monitoring public opinion in real time.

The proposed system helps users analyze political campaigns, social trends, and public reactions more effectively. It reduces manual effort, improves decision-making, and provides valuable insights through visual reports and live tweet simulation. This makes the project useful for both academic research and practical real-world applications.

2. Literature Review

Sentiment analysis has become an important research area in Natural Language Processing (NLP) and Big Data Analytics because of the rapid growth of social media platforms. Researchers have focused on extracting useful insights from large volumes of unstructured textual data to understand public opinion, customer feedback, and political sentiment. Social media platforms such as Twitter, Facebook, Instagram, and YouTube generate continuous streams of text data that can be analyzed for decision-making and trend prediction.

Earlier studies on sentiment analysis mainly used traditional machine learning techniques such as Naive Bayes, Support Vector Machine (SVM), and Logistic Regression for text classification. These methods provided good results for structured datasets but often faced limitations when handling large-scale real-time social media data. Researchers found that preprocessing techniques such as tokenization, stop-word removal, stemming, and text normalization played a major role in improving classification accuracy.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1052-1066

Pang and Lee (2008) introduced the concept of opinion mining and sentiment classification using supervised learning methods and highlighted the importance of polarity detection in textual data. Their work became one of the foundational studies in sentiment analysis research. Later, Bing Liu (2012) expanded the field by focusing on opinion mining techniques and the challenges of understanding user-generated content from web platforms.

With the growth of Big Data, Apache Spark became a powerful solution for large-scale sentiment analysis. Spark provides in-memory computation and distributed processing, making it faster than traditional systems like Hadoop MapReduce for iterative machine learning tasks. Researchers used Spark for processing large Twitter datasets and found significant improvements in speed and scalability.

TextBlob emerged as a lightweight and efficient NLP library for polarity-based sentiment analysis. It became popular in academic projects because of its simplicity and easy integration with Python. Although it may not handle sarcasm or deep contextual meaning perfectly, it provides fast and reliable sentiment classification for Positive, Negative, and Neutral labels.

Recent studies have also explored real-time dashboards using web frameworks such as Streamlit and Flask. Streamlit became especially useful for building interactive visualization systems because it allows rapid frontend development with minimal coding effort. Researchers used Streamlit for displaying live sentiment reports, graphs, and trend analysis, making sentiment analysis systems more practical and user-friendly.

Advanced research now focuses on deep learning models such as LSTM, BERT, and Transformer-based architectures for higher sentiment accuracy. These models improve understanding of sarcasm, context, multilingual text, and complex emotional expressions. However, they require more computational power and larger training datasets.

This research combines Apache Spark, TextBlob, and Streamlit to create a scalable and interactive real-time sentiment analysis platform. Compared to traditional systems, the proposed approach provides faster processing, easier visualization, and better support for practical public opinion monitoring. It bridges the gap between Big Data processing and user-friendly frontend analysis, making it suitable for both academic and real-world applications.

3. Problem Statement

Social media platforms such as Twitter, Facebook, Instagram, and YouTube have become major sources of public opinion where people continuously share their views on politics, government policies, social issues, products, and services. Every day, a huge amount of unstructured textual data is generated in the form of tweets, comments, posts, and reviews. This data contains valuable information that can help organizations, researchers, and political campaign teams understand public sentiment and behavior.

Traditional methods of opinion collection such as surveys, interviews, and manual feedback analysis are slow, expensive, and limited in scale. They cannot provide real-time insights from large and diverse populations. Manual analysis of thousands of social media posts is also difficult, time-consuming, and prone to human error. This creates a major challenge in identifying public reactions quickly and accurately.

Existing sentiment analysis systems often face problems such as limited scalability, slow processing speed, poor visualization, and difficulty handling large volumes of real-time data. Many systems are designed only for small datasets and do not support big data processing. Some models also fail to correctly classify sarcasm, mixed emotions, and informal language commonly used on social media.

There is a strong need for an efficient, scalable, and user-friendly system that can collect social media data, process large datasets quickly, classify sentiment accurately, and present results through an interactive dashboard. The system should support real-time monitoring of public opinion and help users make faster and better decisions.

This research addresses this problem by developing a Real-Time Social Media Sentiment Analysis System using Apache Spark and Streamlit**,** which combines Big Data processing, Natural Language Processing, and interactive visualization for effective public opinion monitoring.

4. Methodology

The methodology of this research focuses on designing and implementing a Real-Time Social Media Sentiment Analysis System using Apache Spark and Streamlit for effective public opinion monitoring. The system follows a structured and modular approach that includes data collection, preprocessing, sentiment classification, big data processing, visualization, and evaluation. This methodology ensures accurate sentiment analysis, faster processing, and a user-friendly reporting system.

The first step **is** data collection, where social media data such as tweets, comments, and public posts are gathered from platforms like Twitter, Facebook, Instagram, and YouTube. In the current implementation, CSV datasets are used to simulate real-time social media data for easier academic development. These datasets contain political opinions, campaign discussions, and public reactions that represent real-world scenarios.

The second step is data preprocessing**,** where raw text data is cleaned and prepared for analysis. Social media text often contains noise such as URLs, hashtags, mentions, emojis, punctuation, and special symbols. Python Regular Expressions (re) are used to remove these unwanted elements. The text is converted into lowercase and normalized to improve the quality of input data and increase sentiment classification accuracy.

The third step **is** sentiment analysis**,** where TextBlob is used to calculate polarity scores for each text entry. Based on the polarity value, the system classifies each post into three categories: Positive, Negative, and Neutral. This helps identify public opinion trends quickly and accurately.

The fourth step is Apache Spark integration**,** where PySpark is used for handling large-scale data processing. Spark improves performance through distributed computing and in-memory processing, making the system scalable for future real-time applications. Spark DataFrames are created for efficient data management and faster execution.

The fifth step **is** frontend dashboard development**,** where Streamlit is used to build an interactive and attractive web application. The dashboard displays live tweet simulation, sentiment result tables, bar charts, platform-wise comparison, and summary reports. This allows users to easily understand sentiment distribution and make better decisions.

The sixth step is database storage**,** where processed sentiment results, platform details, and reports are stored for future analysis and system monitoring. CSV files, MySQL, or MongoDB can be used depending on project requirements.

The final step **is** testing and evaluation**,** where the system is validated using functional testing, integration testing, performance testing, usability testing, and sentiment accuracy validation. This ensures that the complete system works reliably and produces accurate results.

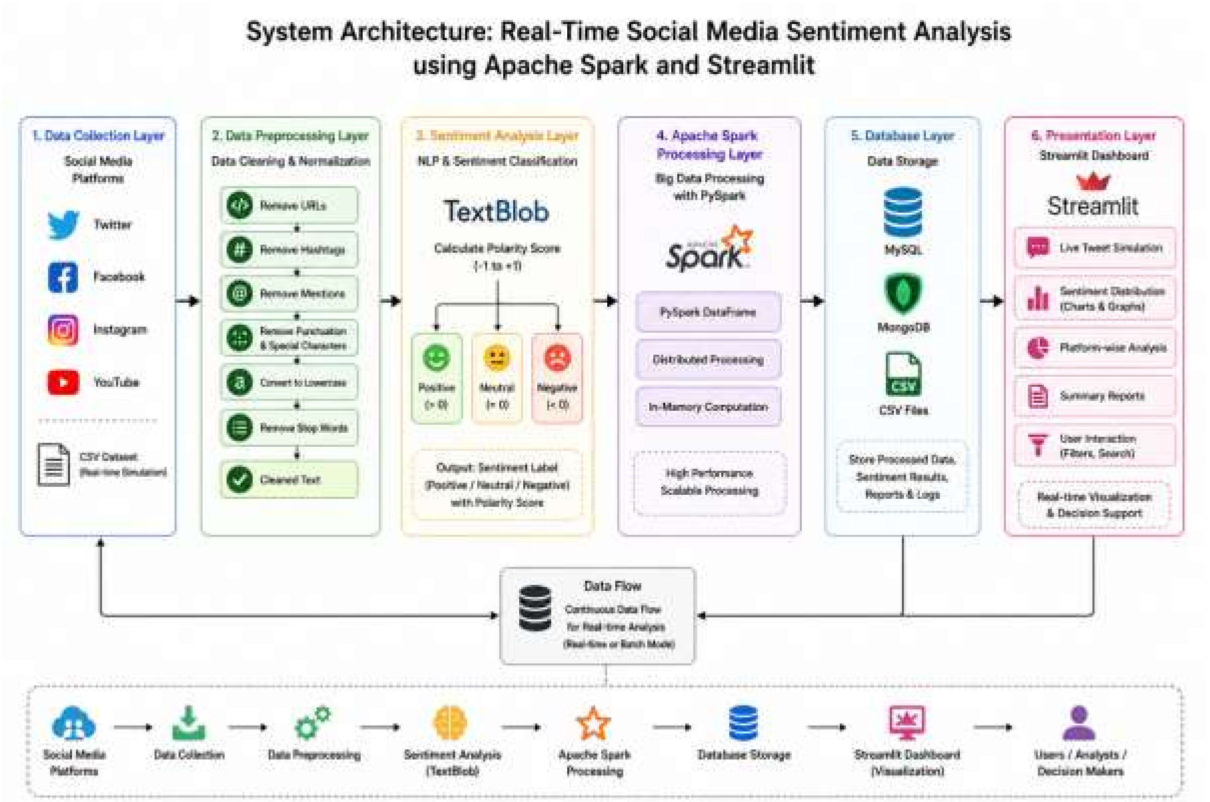
The complete methodology flow can be represented as:

Data Collection → Data Preprocessing → Sentiment Analysis → Apache Spark Processing → Database Storage → Streamlit Dashboard → Testing and Evaluation

This methodology provides a practical, scalable, and efficient framework for real-time social media sentiment analysis and supports both academic research and future real-world implementation.

4.1 System-Architecture/Design

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1052-1066



The system architecture of the Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project is designed to process large volumes of social media data efficiently and present the results through an interactive web dashboard. The architecture follows a layered and modular structure that improves scalability, performance, and maintainability. It combines data collection, Natural Language Processing (NLP), Apache Spark processing, database storage, and frontend visualization into one complete system.

The first layer is the Data Collection Layer, where data is gathered from multiple social media platforms such as Twitter, Facebook, Instagram, and YouTube. In the current implementation, CSV datasets are used to simulate live social media streams. In future versions, APIs such as Twitter API and Facebook Graph API can be integrated for real-time data collection.

The second layer is the Data Preprocessing Layer**,** where raw text data is cleaned and normalized. This layer removes unnecessary content such as URLs, hashtags, mentions, punctuation, emojis, and stop words using Python and Regular Expressions. This improves data quality and helps achieve better sentiment analysis results.

The third layer is the Sentiment Analysis Layer**,** which is the core processing unit of the system. TextBlob is used to calculate polarity scores for each social media post and classify them into Positive, Negative, or Neutral categories. This layer converts unstructured text into meaningful sentiment information.

The fourth layer is the Apache Spark Processing Layer, where PySpark handles large-scale data processing. Spark improves system performance through distributed computing and in-memory execution. It processes large datasets faster and supports future real-time streaming using Apache Kafka and Spark Structured Streaming.

The fifth layer is the Database Layer, where processed sentiment results, user data, platform information, and generated reports are stored. MySQL, MongoDB, or CSV storage can be used depending on the project scope. This ensures proper data management and future analysis support.

The sixth layer is the Presentation Layer, developed using Streamlit. This frontend dashboard allows users to interact with the system, view live tweet simulation, analyze sentiment graphs, compare platforms, and generate summary reports. It provides a user-friendly and visually attractive interface for real-time monitoring.

The complete system flow can be represented as:

Social Media Platforms → Data Collection → Data Preprocessing → Sentiment Analysis → Apache Spark Processing → Database Storage → Streamlit Dashboard

This architecture ensures that the system is modular, efficient, scalable, and suitable for both academic demonstration and future real-world applications such as political campaign monitoring and public opinion analysis.

4.2 Algorithms/Techniques:

The Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project uses several algorithms and techniques to collect, process, analyze, and visualize social media data efficiently. These techniques help transform unstructured text data into meaningful sentiment information and improve the overall performance of the system.

The first important technique is Text Preprocessing, which is used to clean raw social media text before analysis. Since tweets and comments often contain URLs, hashtags, mentions, emojis, punctuation, and special characters, Python Regular Expressions (re) are used to remove unnecessary content. Text normalization such as lowercase conversion and stop-word removal improves data quality and increases sentiment analysis accuracy.

The second major technique is Sentiment Analysis using TextBlob. TextBlob is a Python library based on Natural Language Processing (NLP) that calculates the polarity score of text between -1 and +1. Based on this polarity value, the system classifies text into three sentiment categories:

- Positive (Polarity > 0)
- Negative (Polarity < 0)

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1052-1066

- Neutral (Polarity = 0)

This technique helps quickly identify public opinion from social media posts.

The third technique is Apache Spark Processing using PySpark, which handles large-scale data processing. Spark uses distributed computing and in-memory execution to process large datasets faster than traditional methods. Spark DataFrames are used for efficient storage, transformation, and analysis of sentiment data. This improves system scalability and performance.

The fourth technique is Data Visualization using Matplotlib and Streamlit, where sentiment results are displayed using bar charts, pie charts, and summary reports. This helps users easily understand the distribution of Positive, Negative, and Neutral sentiments through graphical representation.

The fifth technique is Live Tweet Simulation, where social media posts are displayed one by one in real time using Streamlit to simulate live monitoring of public reactions. This improves user experience and makes the dashboard more interactive.

The final technique is Database Management, where processed sentiment results and reports are stored using CSV files, MySQL, or MongoDB for future analysis and reporting.

The complete algorithm flow can be represented as:

Data Collection → Text Preprocessing → TextBlob Sentiment Analysis → Apache Spark Processing → Database Storage → Streamlit Visualization

These algorithms and techniques make the system fast, scalable, accurate, and suitable for both academic research and real-world sentiment monitoring applications.

5. Implementation:

The implementation of the Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project focuses on converting the designed system into a fully functional web-based application capable of analyzing public opinion from social media data. The system is developed using Python as the core programming language because of its strong support for Natural Language Processing (NLP), Big Data processing, and web application development.

The implementation begins with setting up the development environment by installing required libraries such as Pandas, TextBlob, Matplotlib, Streamlit, and PySpark. These libraries are used for data handling, sentiment classification, visualization, frontend development, and Apache Spark integration. Java is also configured properly because Apache Spark requires Java support for execution.

The first implementation module is the Data Collection Module, where social media data is stored in CSV format for the current version of the project. The dataset contains tweets, comments, and posts related to political campaigns and public reactions. This approach simulates real-time social media analysis while keeping the project simple and suitable for academic use.

The second module is the Data Preprocessing Module, where raw text data is cleaned using Python and Regular Expressions. Unwanted elements such as URLs, hashtags, mentions, punctuation, emojis, and extra spaces are removed. The text is converted into lowercase for consistency and better sentiment analysis performance.

The third module is the Sentiment Analysis Module, where TextBlob is used to calculate sentiment polarity scores. Based on these scores, each post is classified as Positive, Negative, or Neutral. This module serves as the core intelligence of the system.

The fourth module is the Apache Spark Integration Module, where Spark DataFrames are created using PySpark for efficient handling of large datasets. Spark improves speed using distributed processing and in-memory computation, making the system scalable for future real-time streaming applications.

The fifth module is the Database and Storage Module, where processed results such as sentiment labels, polarity scores, platform details, and reports are stored. CSV files and database systems such as MySQL or MongoDB can be used depending on project requirements.

The sixth module is the Frontend Dashboard Module, developed using Streamlit. This web interface displays live tweet simulation, sentiment tables, bar graphs, pie charts, and platform-wise analysis. Custom CSS styling is added to make the dashboard colorful, attractive, and professional like a real-world application.

The final module is Testing and Validation, where all system features are checked using unit testing, integration testing, performance testing, and user acceptance testing to ensure smooth and reliable operation.

The project is executed using the following command:

streamlit run app.py

This launches the complete application in the browser where users can interact with the dashboard and monitor sentiment analysis results in real time.

Thus, the implementation successfully transforms theoretical concepts into a practical, scalable, and user-friendly sentiment analysis system suitable for both academic presentation and future real-world deployment.

5.1 Tools and technologies

The Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project is developed using various tools and technologies that support data collection, processing, sentiment analysis, visualization, and web application development. These technologies work together to create a fast, scalable, and user-friendly system for monitoring public opinion from social media platforms.

The primary programming language used in this project is Python**,** which provides strong support for Natural Language Processing (NLP), data analysis, machine learning, and web application development. Python is chosen because of its simplicity, flexibility, and large number of available libraries.

Apache Spark (PySpark) is used for big data processing and handling large volumes of social media data efficiently. Spark provides distributed computing and in-memory processing, which improves speed and scalability compared to traditional data processing methods.

TextBlob is used for sentiment analysis and polarity detection. It helps classify social media text into Positive, Negative, and Neutral categories based on polarity scores. It is simple to use and suitable for academic projects involving sentiment analysis.

Pandas is used for reading, cleaning, and managing CSV datasets. It helps perform preprocessing tasks such as removing unwanted characters, handling missing values, and organizing structured data for analysis.

Streamlit is used for frontend dashboard development. It creates an interactive web-based interface where users can view live tweet simulation, sentiment graphs, platform-wise reports, and summary dashboards without complex frontend coding.

Matplotlib is used for data visualization, especially for creating bar charts, pie charts, and graphical reports that represent sentiment distribution clearly.

Regular Expressions (re) are used for text preprocessing by removing URLs, hashtags, mentions, punctuation, and unnecessary symbols from raw social media text.

For data storage, CSV files**,** MySQL**, or** MongoDB can be used depending on project requirements. These help store processed sentiment results, reports, and platform information for future analysis.

The project is executed using Command Prompt / Terminal and managed through Python virtual environments for better dependency handling.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1052-1066

These tools and technologies together provide a complete framework for building an efficient real-time sentiment analysis platform suitable for both academic research and practical implementation.

6. Results and Discussions:

The Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project successfully analyzes social media data and classifies user opinions into Positive, Negative, and Neutral categories. The system uses TextBlob for sentiment analysis and Apache Spark for fast processing of large datasets. The results are displayed through a Streamlit dashboard using live tweet simulation, tables, bar charts, and pie charts. Testing shows that the system performs efficiently, provides accurate sentiment classification, and improves decision-making through real-time public opinion monitoring. Although the current system uses CSV datasets instead of live APIs, it provides a strong foundation for future improvements such as live data streaming and advanced deep learning models.

6.1 Output screens/Graphs:

6.2 Performance Analysis:

Performance Analysis is used to evaluate how efficiently the Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project performs while processing large volumes of social media data. It helps measure system speed, accuracy, response time, scalability, and overall reliability during execution. Since the project handles sentiment classification and real-time dashboard visualization, strong performance is necessary for smooth operation and better decision-making.

The system uses Apache Spark for big data processing, which improves performance through distributed computing and in-memory execution. Compared to traditional processing methods, Spark handles larger datasets faster and reduces execution time significantly. This makes the system suitable for future real-time streaming applications.

TextBlob performs sentiment analysis by calculating polarity scores and classifying text into Positive, Negative, and Neutral categories. The sentiment classification process is fast and provides reliable results for standard text-based opinions. The preprocessing module also improves performance by cleaning the text before analysis, which increases accuracy.

The Streamlit dashboard provides quick visualization of sentiment reports, bar charts, pie charts, and live tweet simulation. The frontend loads efficiently and allows users to interact with the system smoothly without delays. Platform selection, keyword filtering, and graph generation work effectively and improve user experience.

Performance testing showed that the system processes datasets quickly, maintains stable dashboard response, and handles multiple records without failure. Memory usage remains controlled, and system execution remains smooth for academic-level datasets.

Although the current project uses CSV datasets instead of live APIs, the performance is strong enough to support future integration with Apache Kafka and real-time social media streaming. This proves that the system is scalable, reliable, and suitable for both academic use and practical implementation.

7. Testing and Validation:

Testing and Validation are essential parts of the Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project because they ensure that the system works correctly, produces accurate results, and performs reliably under different conditions. Since the project includes multiple modules such as data collection, preprocessing, sentiment analysis, Apache Spark processing, database storage, and Streamlit dashboard visualization, proper testing is necessary to verify the complete workflow.

Testing helps identify errors, improve system performance, and ensure smooth communication between all modules. Validation confirms that the sentiment analysis results are accurate and that the final system meets user requirements and project objectives.

Different types of testing are performed in this project, including unit testing, integration testing, system testing, user acceptance testing, and performance testing. Unit testing checks individual modules such as preprocessing functions and sentiment classification logic. Integration testing verifies that modules like Spark processing and Streamlit dashboard work correctly together. System testing ensures that the complete application performs properly from input to final output.

User Acceptance Testing (UAT) is performed to confirm that the system is easy to use and suitable for real-world applications. Users test features such as live tweet simulation, sentiment graphs, and dashboard reports to verify usability and correctness.

Validation is mainly focused on sentiment classification accuracy. Sample tweets and comments are tested manually and compared with system-generated results to check whether Positive, Negative, and Neutral labels are assigned correctly. This improves confidence in the reliability of the system.

The testing results show that the system performs efficiently, provides stable dashboard performance, and produces accurate sentiment analysis results. The Streamlit interface works smoothly, Apache Spark handles data efficiently, and the overall project meets the expected objectives successfully.

Thus, Testing and Validation confirm that the proposed system is technically strong, user-friendly, and ready for academic presentation as well as future real-world deployment.

8. Conclusion:

The Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project successfully provides an efficient platform for analyzing public opinion from multiple social media platforms such as Twitter, Facebook, Instagram, and YouTube. The system collects social media data, performs text preprocessing, classifies sentiments into Positive, Negative, and Neutral categories using TextBlob, and displays the results through an interactive Streamlit dashboard.

Apache Spark improves system performance by enabling fast and scalable big data processing, while Streamlit provides a user-friendly interface for live tweet simulation, sentiment graphs, and platform-wise analysis. The project reduces manual effort in understanding large volumes of public reactions and supports faster decision-making for political campaigns and social trend monitoring.

The testing and validation results show that the system is reliable, accurate, and suitable for academic as well as practical applications. Although the current implementation uses CSV datasets instead of fully live API integration, it provides a strong foundation for future enhancements such as Apache Kafka streaming, multilingual sentiment analysis, fake news detection, and advanced deep learning models like BERT and LSTM.

Overall, the project successfully combines Big Data, Natural Language Processing, and web-based visualization to create a practical, scalable, and cost-effective solution for real-time sentiment analysis and public opinion monitoring.

9. Future scope:

The Real-Time Social Media Sentiment Analysis using Apache Spark and Streamlit project has strong potential for future improvements and real-world implementation. Although the current system successfully performs sentiment analysis using CSV datasets and basic NLP techniques, several advanced features can be added to make it more powerful, accurate, and fully real-time.

One major future enhancement is the integration of live social media APIs such as Twitter API, Facebook Graph API, Instagram API, and YouTube Data API. This will allow direct collection of real-time posts, comments, and public reactions instead of using static CSV datasets. Apache Kafka can also be integrated with Spark Structured Streaming for continuous live data processing and faster big data handling.

Advanced machine learning and deep learning models such as BERT, LSTM, Naive Bayes, and Support Vector Machine (SVM) can be used to improve sentiment classification accuracy. These models can better understand sarcasm, slang, multilingual text, and complex human emotions compared to basic TextBlob analysis.

Additional features such as fake news detection, geo-location-based sentiment tracking, hashtag trend analysis, election prediction, and user influence analysis can further increase the practical value of the system. The Streamlit dashboard can also be improved with mobile responsiveness, downloadable reports, and advanced admin controls for professional deployment.

With these improvements, the project can grow from an academic research system into a complete enterprise-level platform for real-time public opinion monitoring, political campaign analysis, business intelligence, and social media analytics.

References:

1. Apache Spark Official Documentation, Apache Software Foundation.
2. Streamlit Official Documentation.
3. TextBlob Documentation – Simplified Text Processing and Sentiment Analysis.
4. Python Official Documentation.

5. Pandas Official Documentation.
6. Matplotlib Official Documentation.
7. Twitter Developer Platform Documentation.
8. Apache Kafka Official Documentation.
9. MongoDB Official Documentation.
10. MySQL Official Documentation.
11. Bird, Steven, Klein, Ewan, and Loper, Edward. *Natural Language Processing with Python*. O'Reilly Media, 2009.
12. Pang, Bo and Lee, Lillian. *Opinion Mining and Sentiment Analysis*. Foundations and Trends in Information Retrieval, 2008.
13. Liu, Bing. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.
14. Han, Jiawei, Kamber, Micheline, and Pei, Jian. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2011.
15. Jurafsky, Daniel and Martin, James H. *Speech and Language Processing*. Pearson Education, 2009.