Estd. 1990

University of Management & Technology

Real-Time Twitter Sentiment Analysis

A Big Data Project using Apache Kafka, PySpark MLlib, MongoDB & Django

Instructor	Dr. Umar Rashid
Subject	Big Data Analytics
Course Code	CS6168 - Section D1

Group Members

Name	Student ID
Kaleemullah Younas	F2022332074
M Dawood Jaffar	F2022332064
Asadullah Masood	F2022332082

1. Introduction

Sentiment analysis is a critical field in Natural Language Processing (NLP) and machine learning that focuses on identifying and extracting subjective information from text data. With the exponential growth of social media platforms like Twitter, there is an unprecedented volume of user-generated content expressing opinions, emotions, and reactions on various topics.

This project implements a **real-time sentiment analysis pipeline** that processes streaming Twitter data, classifies tweets into sentiment categories, and presents the results through an interactive web dashboard. The system leverages Big Data technologies to handle high-velocity data streams efficiently.

1.1 Project Objectives

- Implement real-time data ingestion using Apache Kafka message broker
- Process streaming data with Apache Spark (PySpark) for distributed computing
- Train and deploy a Logistic Regression classifier for sentiment prediction
- Store classified results in MongoDB NoSQL database
- Visualize real-time analytics through Django web dashboard

2. System Architecture

The system follows a microservices architecture pattern where each component operates independently but communicates through well-defined interfaces. This design ensures scalability, fault tolerance, and maintainability of the overall pipeline.

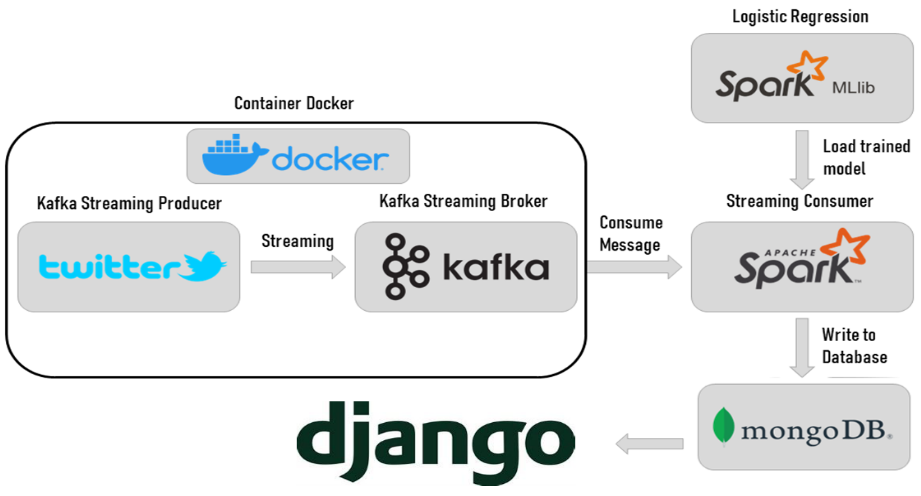


Figure 1: System Architecture - Data Flow from Twitter to Dashboard

2.1 Component Description

Component	Technology	Function
Data Source	Twitter Dataset	Provides tweet data for streaming simulation
Message Broker	Apache Kafka	Handles real-time message queuing and streaming
Stream Processor	PySpark Streaming	Consumes, processes, and classifies tweets
ML Engine	PySpark MLlib	Logistic Regression model for classification
Database	MongoDB	Stores classified tweets and predictions
Web Server	Django	Serves dashboard and API endpoints

3. Dataset Description

The project utilizes the **Twitter Entity Sentiment Analysis** dataset from Kaggle. This dataset contains tweets labeled with sentiment categories, making it ideal for training and evaluating our classification model.

Property	Training Set	Validation Set
File Name	twitter_training.csv	twitter_validation.csv
Number of Tweets	74,682	998
Features	Tweet ID, Entity, Sentiment, Tweet Content	Same as Training
Sentiment Classes	Positive, Negative, Neutral, Irrelevant	Same 4 Classes

4. Machine Learning Model

4.1 Data Preprocessing

Before training the model, extensive preprocessing was performed on the raw tweet data to improve classification accuracy:

- **Text Cleaning:** Removal of special characters, URLs, and HTML entities
- **Lowercasing:** Converting all text to lowercase for consistency
- **Tokenization:** Splitting text into individual words/tokens
- **Stop-word Removal:** Eliminating common words with low semantic value
- **TF-IDF Vectorization:** Converting text to numerical feature vectors

4.2 Model Training & Evaluation

We experimented with multiple classification algorithms including Naive Bayes, Random Forest, and Logistic Regression. After comprehensive evaluation using cross-validation and accuracy metrics, **Logistic Regression** was selected as the final model due to its superior performance and computational efficiency.

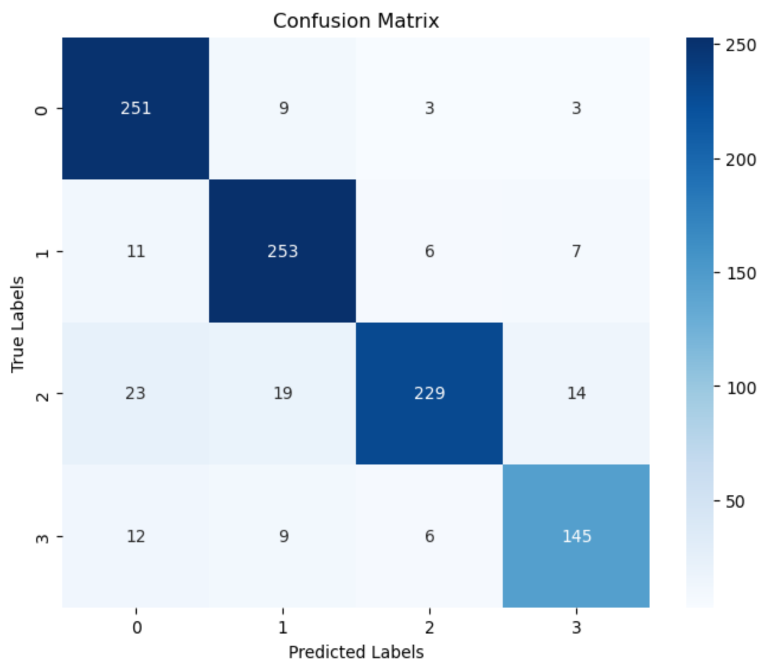


Figure 2: Confusion Matrix - Model Evaluation

The confusion matrix demonstrates strong diagonal values indicating high classification accuracy.

5. Implementation Details

5.1 Kafka Producer

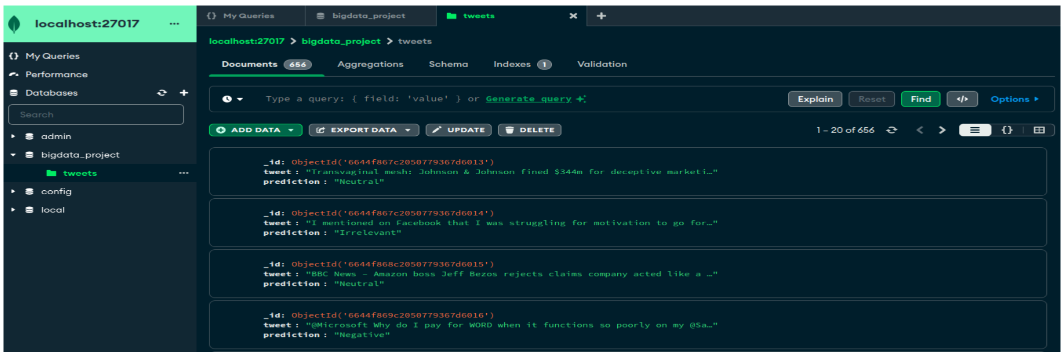
The Kafka producer simulates real-time tweet streaming by reading from the validation CSV file and sending messages to the 'numtest' Kafka topic. Each tweet is published every 3 seconds to simulate realistic streaming behavior.

5.2 PySpark Consumer

The consumer application connects to Kafka, receives streaming tweets, and processes them in real-time. Each tweet undergoes text preprocessing, feature extraction through the ML pipeline, and sentiment prediction. Results are immediately stored in MongoDB with timestamps for historical tracking.

5.3 MongoDB Storage

All classified tweets are persisted in MongoDB, a NoSQL document database. Each document contains the original tweet text, the predicted sentiment label, and a timestamp. This enables historical analysis and trend visualization on the dashboard.



```
_id: ObjectId('6644f867c2050779367d6013')
tweet : "Transvaginal mesh: Johnson & Johnson fined $344m for deceptive marketi…"
prediction : "Neutral"

_id: ObjectId('6644f867c2050779367d6014')
tweet : "I mentioned on Facebook that I was struggling for motivation to go for…"
prediction : "Irrelevant"

_id: ObjectId('6644f868c2050779367d6015')
tweet : "BBC News - Amazon boss Jeff Bezos rejects claims company acted like a …"
prediction : "Neutral"

_id: ObjectId('6644f869c2050779367d6016')
tweet : "@Microsoft Why do I pay for WORD when it functions so poorly on my @Sa…"
prediction : "Negative"
```

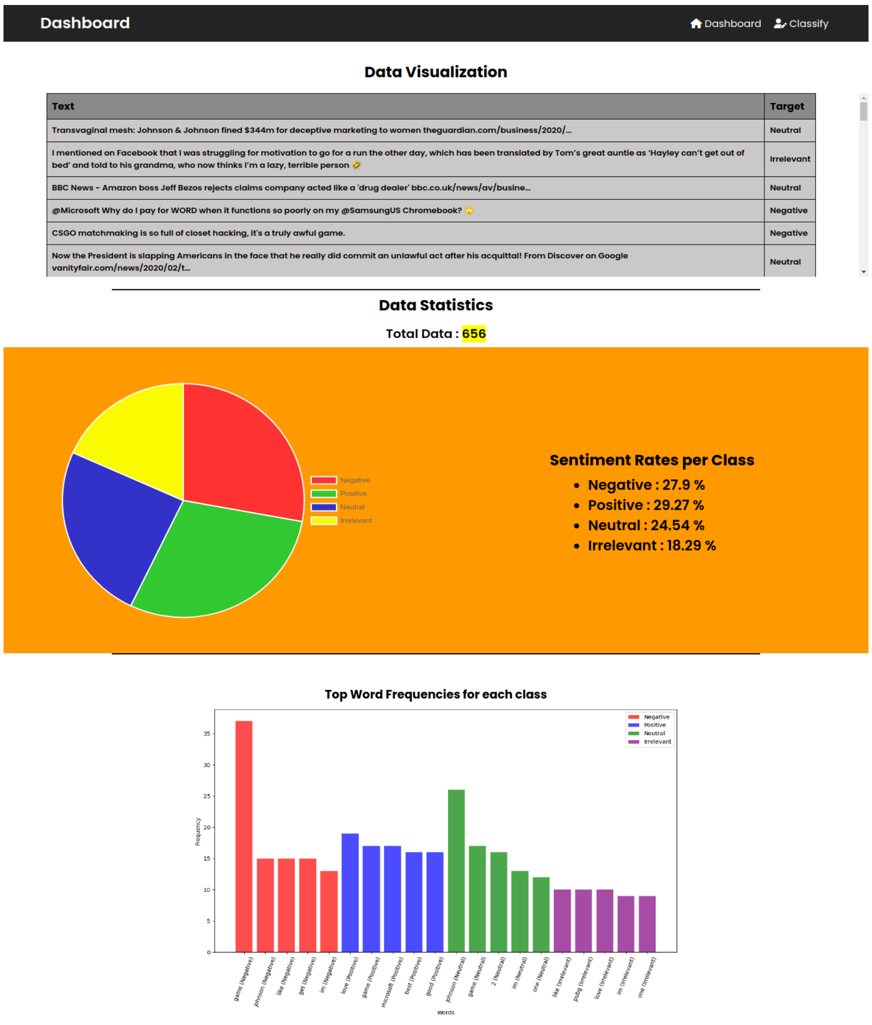
Figure 3: MongoDB Compass - Stored Tweet Documents

6. Web Dashboard & Visualization

The Django-based web dashboard provides comprehensive real-time analytics and visualization capabilities. Users can monitor sentiment trends, view individual tweet classifications, and analyze distribution patterns.

6.1 Dashboard Features

- **Real-time Tweet Table:** Displays classified tweets with sentiment labels
- **Pie Chart:** Shows sentiment distribution percentages
- **Bar Graph:** Visualizes word frequencies per sentiment class
- **Statistics Panel:** Total tweet count and per-class percentages



Data Visualization

Text	Target
Transvaginal mesh: Johnson & Johnson fined $344m for deceptive marketing to women theguardian.com/business/2020/…	Neutral
I mentioned on Facebook that I was struggling for motivation to go for a run the other day, which has been translated by Tom's great auntie as 'Hayley can't get out of bed' and told to his grandma, who now thinks I'm a lazy, terrible person 🤣	Irrelevant
BBC News - Amazon boss Jeff Bezos rejects claims company acted like a 'drug dealer' bbc.co.uk/news/av/busine…	Neutral
@Microsoft Why do I pay for WORD when it functions so poorly on my @SamsungUS Chromebook? 🙄	Negative
CSGO matchmaking is so full of closet hacking, it's a truly awful game.	Negative
Now the President is slapping Americans in the face that he really did commit an unlawful act after his acquittal! From Discover on Google vanityfair.com/news/2020/02/t…	Neutral

Data Statistics

Total Data : 656

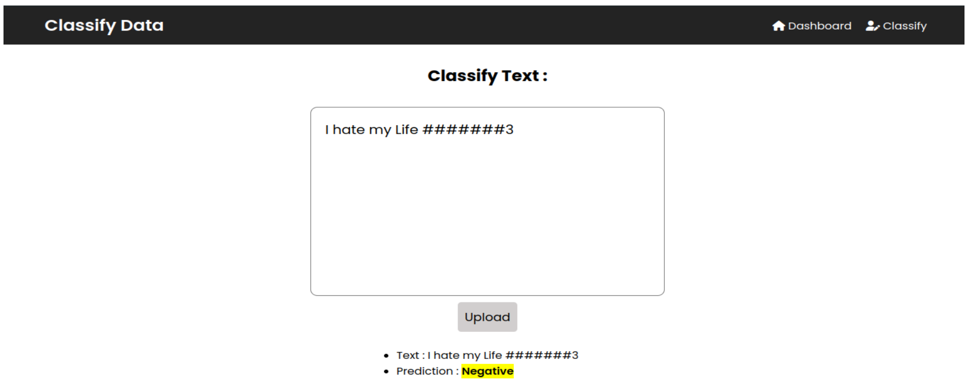
Sentiment Rates per Class

- Negative : 27.9 %
- Positive : 29.27 %
- Neutral : 24.54 %
- Irrelevant : 18.29 %

Figure 4: Django Dashboard - Sentiment Analytics

6.2 Custom Text Classification

In addition to streaming analysis, the dashboard provides a custom classification interface where users can input any text and receive an instant sentiment prediction using the trained model.



- Text : I hate my Life #######3
- Prediction : Negative

Figure 5: Custom Text Classification Interface

7. Results & Analysis

The implemented system successfully demonstrates real-time sentiment analysis capabilities with the following outcomes:

Metric	Result
Model Accuracy	Satisfactory classification performance
Processing Latency	Real-time (< 1 second per tweet)
Sentiment Classes	4 (Positive, Negative, Neutral, Irrelevant)
Dashboard Update	Real-time refresh with new data
Data Persistence	100% tweets stored in MongoDB

8. Conclusion

This project successfully demonstrates the integration of multiple Big Data technologies to build a complete real-time sentiment analysis pipeline. The combination of Apache Kafka for streaming, PySpark for distributed processing and ML, MongoDB for storage, and Django for visualization creates a robust and scalable solution.

8.1 Key Achievements

- Successfully implemented end-to-end streaming data pipeline
- Trained and deployed ML model for real-time classification
- Built interactive dashboard with real-time visualizations
- Achieved seamless integration between all system components

8.2 Future Enhancements

- Integration with live Twitter API for actual tweet streaming
- Implementation of deep learning models (BERT, Transformers) for improved accuracy
- Kubernetes deployment for horizontal scaling
- Real-time alerting system for sentiment anomalies

9. References

- **Twitter Entity Sentiment Analysis Dataset**
https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis
- **Apache Spark Documentation**
https://spark.apache.org/docs/latest/
- **Apache Kafka Documentation**
https://kafka.apache.org/documentation/
- **Django Framework Documentation**
https://docs.djangoproject.com/
- **MongoDB Python Driver (PyMongo)**
https://pymongo.readthedocs.io/