

HPDP Project 2.pdf

by Navachander Navasantar

Submission date: 07-Jul-2025 06:32PM (UTC-0700)

Submission ID: 2711696555

File name: HPDP_Project_2.pdf (1.66M)

Word count: 5024

Character count: 28206



Department of Computer Science
Faculty of Computing

Real-Time Sentiment Analysis using Apache Spark and Kafka

Programme	: Bachelor of Computer Science (<i>Data Engineering</i>)
Subject Code	: SECP3133
Subject Name	: High Performance Data Processing
Session-Sem	: 2024/2025-2

BIL	Prepared By	MATRIC NUMBER
1	LOW JIE SHENG	A22EC0075
2	MUHAMMAD DANIAL BIN AHMAD SYAHIR	A22EC0206
3	NADHRAH NURSABRINA BINTI ZULAINI	A22EC0224
4	NAVACHANDER NAVASANTAR	A22EC0226

9
Table of Contents

1. Introduction	3
2. Data Acquisition & Preprocessing	5
3. Sentiment Model Development	7
4. Apache System Architecture	10
5. Analysis & Results	18
6. Optimization & Comparison	23
7. Conclusion & Future Work	27
8.0 References	30
9.0 Appendix	31

1. Introduction

Product, services, and events have their perception deeply influenced by online reviews and the corresponding social media postings owing to the role played by user-generated content in the current digital world. A method of analyzing such opinions that is gaining more and more popularity is the real-time sentiment analysis which transforms these opinions into the value-added information that is useful both by the business companies and other organizations.

As big data continues to grow by leaps and bounds, the familiar tools of data processing simply cannot address real-time analysis demands. That is why real-time sentiment analysis platform powered by scalable and distributed technologies including Apache Kafka and Apache Spark has become essential. This project will combine these technologies together in an effort to create a high performance low-latency sentiment analysis system capable of processing streaming data.

The project looks to build a streaming sentiment-analysis system on Apache Spark and Kafka. The most important goals are:

- Design a sentimental analysis tool on a large scale based on Apache Spark to handle huge amounts of data and Apache Kafka to stream and ingest data.
- Obtain information of reviews at Steam Store to understand the opinion of the people concerning the popular games.
- Apply Natural Language Processing (NLP) technique to pre, and feature extraction of the text.
- Build and test machine and deep learning models to do sentiment analysis.
- Produce interesting visualizations that provide the stakeholders with insights into sentiment trend, polarity distribution and game specific feedback.

Workflow Overview

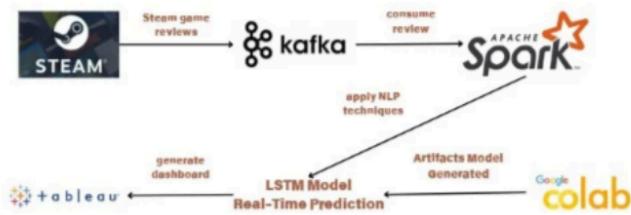


Figure 1: Overview Diagram of Workflow

The workflow of the project incorporates a diverse environment to construct a strong sentiment analysis system in real-time. The whole thing starts with the gathering of Steam game reviews as the major source of data that is filled with user-generated content full of sentiments. Such reviews are then streamed into Apache Kafka which is in effect a real time data reception and message queue which is effective in receiving never-ending load of information. Apache Spark Structured Streaming then takes over, consuming the reviews through Kafka, and using various Natural Language Processing (NLP) models on it to preprocess the text in real-time, such as text cleaning, tokenization, removing the stopwords and stemming it. Sentiment analysis is done using the LSTM model which is trained independently on Google Colab under TensorFlow/Keras and upon completion, then a model file is exported to be integrated into the pipeline. Spark in turn utilizes this trained LSTM architecture to a real-time sentiment analysis, by determining whether the relevant incoming review is positive or negative. Finally, the processed information is presented in the form of Tableau dashboards, which grant all stakeholders with an insight and the interactive capabilities regarding the sentiment trends, variations in time, and game-specific feedback. This end-to-end process makes it possible to serve low-latency performance, sophisticated deep learning analysis and real-time visualization, thus resulting in decision-making based on this timely analysis..

2. Data Acquisition & Preprocessing

2.1 Data Sources

The data used in this project is obtained through Steam Store API by utilizing several Python scripts that enables us to access reviews of popular games such as Counter-Strike 2, Dota 2, and GTA V reviews. Reviews are in form of texts, time stamps and the voting indications..

2.2 Tools and Technologies

Tool/Technology	Purpose
Python	For data acquisition, preprocessing, and streaming integration
Pandas	Data manipulation, cleaning, and transformation
TextBlob	For initial polarity scoring during EDA
Kafka	Streaming data ingestion to enable real-time processing
Jupyter Notebook	Exploratory data analysis, script development, and model training

2.3 Preprocessing Steps

It was significantly preprocessed before inserting it into models to secure quality and consistency

1. Kept in lower case- Converted all words to lowercase to normalise the input and reduce vocabulary.
2. Punctuation Removal & Symbol Removal - Used regular expressions to remove non-letter characters, such as punctuations, special symbols and emojis, to retain only meaningful text.

3. Tokenization – Break down sentences into separate words or tokens for examination.
4. Stopword Removal – Common English stopwords that contribute minimal semantic value to sentiment analysis were removed.
5. Stemming – Converted words to their base forms with PorterStemmer, successfully consolidating similar terms under one stem for improved model generalization.

Code Snippet:

```
def clean_text(text):  
    # Lowercase, remove non-letters  
    text = re.sub(r"[^a-zA-Z\s]", "", str(text).lower())  
    tokens = text.split()  
    # Apply stemming  
    stemmed = [stemmer.stem(word) for word in tokens if len(word) > 2]  
    return " ".join(stemmed)
```

3. Sentiment Model Development

3.1 Model Choices

We picked two models to compare their performance:

3.1.1 Naive Bayes Classifier (scikit-learn)

- **Input:** The text is transformed into a vector format with TF-IDF.
- **Advantages:** It's simple, computationally efficient, and requires less time to train.
- **Limitations:** It assumes that words are independent, which means it misses out on understanding context or the order of words.

3.1.2 LSTM Neural Network (TensorFlow/Keras)

- **Input:** It uses embeddings with tokenized sequences.
- **Advantages:** This model captures sequential relationships and understands the context, making it great for NLP tasks.
- **Limitations:** It requires large, balanced datasets, has a high computational cost, and takes longer to train.

3.2 Training Process:

Model	Preprocessing / Input	Model Details	Output	Accuracy
Naive Bayes	- Texts vectorized using TfidfVectorizer - Labels encoded for sentiment analysis	Trained using 80% train / 20% test split	Sentiment label (binary or multi-class)	>80%

LSTM	<ul style="list-style-type: none"> - Tokenized with vocabulary size = 10,000 - Input sequences padded to length = 200 	Final layer uses sigmoid activation for binary classification	Binary sentiment (positive/negative)	Not explicitly stated, implied good
-------------	---	--	--------------------------------------	-------------------------------------

Code Snippet:

```
model = Sequential([
    Embedding(input_dim=10000, output_dim=64, input_length=200),
    LSTM(64, return_sequences=False),
    Dropout(0.5),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.summary()
```

Model Evaluation Comparison:

Metric	Naive Bayes (%)	LSTM Model (%)
Accuracy	82.0%	74.0%
Precision (Negative)	96.0%	0.0%
Recall (Negative)	35.0%	0.0%

F1-score (Negative)	51.0%	0.0%
Precision (Positive)	81.0%	74.0%
Recall (Positive)	100.0%	100.0%
F1-score (Positive)	89.0%	85.0%
Macro Avg F1-score	70.0%	42.0%
Weighted Avg F1-score	79.0%	62.0%

4. Apache System Architecture

4.1 Apache Kafka

Apache Kafka is the backbone of our project when it comes to real-time data ingestion. Think of it as a distributed publish-subscribe messaging platform that neatly separates data producers, like our review scrapers, from consumers, which include Spark streaming and sentiment analysis processors. Kafka ensures that streaming data flows reliably, resiliently, and at scale, making it perfect for our analysis needs.

Key Components and Functions:

- **Producer:**

The kafka_producer.py file is your go-to Kafka producer. It pulls review data from a CSV file (steam_game_reviews.csv) that was collected earlier and sends each review as a JSON message to the Kafka topic named game-reviews. It uses the game title as the message identifier to keep everything organized in partitions.

- **Consumer:**

The kafka_consumer.py script functions as a Kafka consumer that interacts with game reviews. It performs additional sentiment analysis using TextBlob on the incoming reviews and saves the processed results in CSV format for archiving and future analysis.

Advantages:

- **Increased Throughput & Scalability:** Kafka efficiently handles large volumes of messages per second, making it perfect for real-time data streams.
- **Fault Tolerance:** It duplicates messages across brokers, ensuring that no data is lost if a broker fails.
- **Decoupling:** This allows producers and consumers to scale independently without needing to be tightly integrated.
- **Persistence:** Stores messages on disk, enabling consumers to reprocess data if needed.

4.1.1 Setting Up and Running Apache Kafka

Requirements:

- **Docker Desktop:** Used to deploy Kafka and Zookeeper containers seamlessly.
- **docker-compose.yml:** Configures services for Zookeeper and Kafka.

Setup Steps:

1. **Start Kafka and Zookeeper:** Navigate to the project directory and run:

```
docker-compose up -d
```

This spins up Zookeeper and Kafka services in the background.

2. **Verify Kafka Topics:** To list all topics:

```
docker exec kafka kafka-topics --list --bootstrap-server localhost:9092
```

4.1.2 Source Code

Kafka Producer (kafka_producer.py)

Purpose: Streams game reviews from a CSV file to the game-reviews Kafka topic as JSON messages.

Kafka Setup:

```

from kafka import KafkaProducer
import json

class GameReviewProducer:
    def __init__(self, bootstrap_servers='localhost:9092'):
        self.producer = KafkaProducer(
            bootstrap_servers=bootstrap_servers,
            value_serializer=lambda v: json.dumps(v).encode('utf-8'),
            key_serializer=str.encode,
            acks='all',
            retries=3,
            linger_ms=10,
            batch_size=16384
        )

```

KafkaProducer: Instantiates a KafkaProducer connected to localhost:9092 with JSON serialization and batching configurations for performance.

Streaming Reviews:

```

def send_reviews_from_csv(self, csv_file, topic='game-reviews', batch_size=100):
    df = pd.read_csv(csv_file)
    for idx, row in df.iterrows():
        review_data = {
            "id": int(row['id']),
            "game": row['game'],
            "review": row['review'],
            "voted_up": bool(row['voted_up']),
            "timestamp": row['timestamp'],
            "platform": "steam",
            "sentiment": row['sentiment']
        }
        self.send_review(topic, review_data)
        if (idx + 1) % batch_size == 0:
            print(f"sent {idx + 1} reviews so far...")
            time.sleep(1)

```

Reads reviews from CSV, sends them in batches, and logs progress every 100 reviews to monitor ingestion status

Kafka Consumer (kafka_consumer.py)

Purpose: Consumes reviews from game-reviews topic, performs TextBlob sentiment analysis, and saves results to CSV.

Kafka Setup:

```
from kafka import KafkaConsumer
import json

class GameReviewConsumer:
    def __init__(self, bootstrap_servers='localhost:9092', topic='game-reviews'):
        self.consumer = KafkaConsumer(
            topic,
            bootstrap_servers=bootstrap_servers,
            value_deserializer=lambda m: json.loads(m.decode('utf-8'))
        )
```

Processing Reviews:

```
def consume_and_process(self, output_file='processed_reviews.csv'):
    with open(output_file, 'w', newline='', encoding='utf-8') as csvfile:
        writer = csv.DictWriter(csvfile, fieldnames=[...])
        writer.writeheader()

        for message in self.consumer:
            review = message.value
            review['processed_sentiment'] = self.analyze_sentiment(review['review'])
            writer.writerow(review)
            csvfile.flush()
```

analyze_sentiment: Uses TextBlob to classify review polarity as positive, negative, or neutral.

4.2 Apache Spark Streaming

Apache Spark Streaming is the heartbeat of this project, acting as the real-time processing engine. It pulls in streamed reviews from Kafka, processes them using natural language

processing and machine learning models, and then sends the analyzed data to various output sinks for visualization.

Key Components and Functions:

- **Spark Session:**

Started with setups to combine Spark SQL Kafka connector, adaptive query execution, and resource optimization for streaming tasks.

- **Streaming DataFrame:**

Extracts messages from the game-reviews topic, converts JSON into a structured DataFrame with a specified schema for subsequent processing.

Processing Steps within Spark:

Implemented in spark_streaming.py via SparkGameReviewProcessor class.

1. Spark Session Creation:

```
self.spark = SparkSession.builder \
    .appName("GameReviewSentimentAnalysis") \
    .config("spark.jars.packages", "org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.0") \
    .config("spark.sql.adaptive.enabled", "true") \
    .getOrCreate()
```

spark-sql-kafka-0-10 connector integrates Kafka streaming with Spark SQL processing.

2. Creating Kafka Stream:

```
def create_kafka_stream(self, topic="game-reviews"):
    return self.spark \
        .readStream \
        .format("kafka") \
        .option("kafka.bootstrap.servers", "localhost:9092") \
        .option("subscribe", topic) \
        .load()
    ....
```

Reads data continuously from game-reviews topic.

3. Parsing and Aggregation:

```
parsed_df = kafka_df.select(
    col("key").cast("string").alias("game_key"),
    from_json(col("value").cast("string"), self.review_schema).alias("data")
).select("game_key", "data.*")

sentiment_agg = parsed_df \
    .groupBy("game", "sentiment") \
    .count() \
    .withColumnRenamed("count", "sentiment_count")
```

Parses JSON messages into structured columns

Aggregates sentiment counts per game for dashboarding

Advantages:

- **Micro-Batch and Continuous Processing:** Provides both approaches to manage latency and throughput effectively.
- **Fault Tolerance:** Checkpointing guarantees that processing continues seamlessly after disruptions.
- **Unified Analytics Engine:** Provides SQL, MLlib, GraphX, and Streaming within one framework

4.2.1 Setting Up and Running Apache Spark

Requirements:

- 8
- Java Development Kit (JDK): Spark requires Java 8 or newer
 - Apache Spark: Download pre-built with Hadoop
 - Python 3.7+: For PySpark scripts

Setup Steps:

1. Set up JDK and Spark by following the guidelines provided at:
[OpenJDK](#)

[Apache Spark Downloads](#)

2. Set Environment Variables:
 - SPARK_HOME: Spark installation path
 - PATH: Include Spark's bin directory
 - JAVA_HOME: JDK installation path
3. Run Streaming Application: python spark_streaming.py

4.2.2 Source Code

Spark Streaming Processor (spark_streaming.py)

Processes streamed data using the SparkGameReviewProcessor class to perform:

- Batch Analysis: Loads CSV data, computes sentiment distribution, saves outputs.
- Streaming Processing: Reads from Kafka, parses JSON, aggregates sentiments, and writes results to console, files, and Kafka topics for further consumption.

4.3 Pipeline Orchestration

The run_pipeline.py script coordinates the entire pipeline by starting:

1. Spark Streaming Processor
2. Kafka Consumer
3. Kafka Producer

Each runs as a daemon thread to enable parallel, seamless execution.

Code Snippet Examples:

```
pipeline = GameReviewPipeline()  
pipeline.run_complete_pipeline()
```

Ensures Spark is ready first, then starts the consumer for processing, and finally the producer to inject data into the pipeline.

4.4 Overall Workflow Summary

1. Steam reviews CSV → Kafka Producer (kafka_producer.py) → topic game-reviews.
2. Spark Streaming (spark_streaming.py) → consolidates data → outputs to sentiment-results topic and files.
3. Kafka Consumer (kafka_consumer.py) → Classifies sentiment using TextBlob → stores processed_reviews.csv
4. Visualization: Processed insights are shown on external Tableau dashboards.

5. Analysis & Results

The following section gives the results of analysis and visualizations on the real-time sentiment analysis of Steam game reviews. The investigation remains in the area of deciphering sentiment standards in users based on dashboards made in Tableau. Data containing the names of games, text of reviews, time when reviews were written, tags to show the sentiment of the review (positive or negative), and upvote flags was visualized to show review behaviors, trends over time, and sentiments on a game-to-game basis. The objective was to convert hard sentiment-tagged data to consumable and practical results to know about the perceptions of the players.

5.1 Key Findings

- The general reviews are characterized by positive sentiment.

It can be seen in the sentiment distribution pie chart that most of the reviews on Steam in the given dataset are classified as positive. It indicates that the games which have been reviewed caused general satisfaction of most users.

- Trends in sentiments change over the time.

The line graph that depicts sentiment when measured against time shows that the number of positive, neutral, and negative reviews differs according to days and hours. These changes can indicate the occurrences in the real world (game patch, bug fixes, or offers).

- There are some games that have the largest number of reviews.

The bar chart of the Top Reviewed Games indicates that a small set of games are reviewed mainly. Most of these titles may be either widely discussed or popular within the steam community.

- The sentiments of games differ drastically.

The stacked bar chart with sentiment presented by game indicates that there are some games which get predominantly good reviews, whereas relatively balanced or even

negative distribution is observed in others. This means that the satisfaction and experience ranked differently between games.

- Bubble chart gives prominence to the game whose review is large and positive or negative.

The sentiment bubble view can be used to step down on the view of the games that are commonly reviewed as well as sentimentally charged. To give an example, a big red bubble where a popular game obtains predominantly negative reviews, a big green bubble shows a title about which positive reviews are published.

5.2 Visualizations

The visualisation part of the project was conducted in Tableau and became the main tool to turn structured sentiment data into an interactive dashboard. The below dashboard was aimed at providing straightforward, accessible knowledge regarding the player sentiment trends driven by Steam game reviews. Proper selection and combining of different charts to support multi-dimensional analysis of the data have been made.

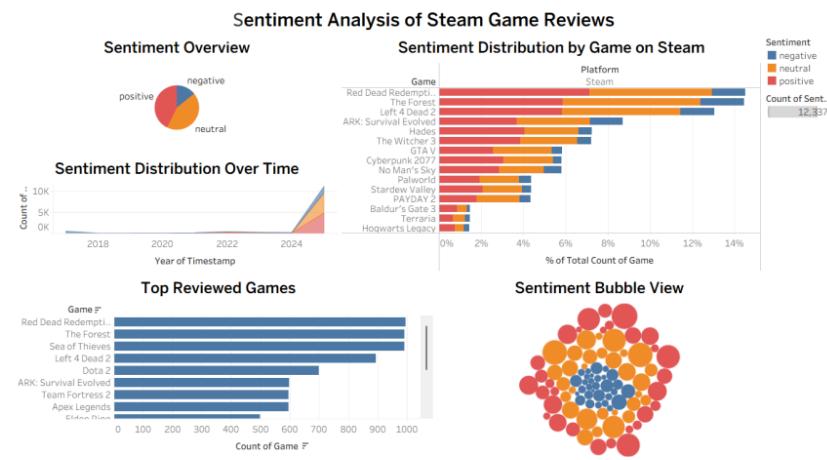


Figure 5.2.1 Dashboard

The pie chart as the first visual component gives a condensed picture of how the sentiment is distributed in the entire reviews. This figure classifies the data into three categories of sentiments positive, neutral, and negative thus summarizing the evaluator to easily view the general tone of the responses given by the players. The percentage of each type of sentiment acts as an anchor to the larger visualisations provided in the rest of the dashboard.

A stacked bar graph was built to show the breakdown of sentiments per game to provide game-specific information. Through this chart, users can compare the perception of the game individually, where each part of the bar indicates the amount of positive, neutral, or negative reviews that a particular game received in reviews. The given visualisation comes in handy in determining the games that have both negative and positive opinions, namely, when the opinion is mostly split into opposite sides. On the other hand, it can help determine the games that are overwhelmingly positively or negatively reviewed by the players.

Trends in time series sentiment were displayed by means of a line score, where the amounts of each sentiment category are recorded over a time scale. It is possible to modify the chart to show a closer view of the hourly, daily, or weekly patterns, in case of the fine detail of the analysis needed. This will allow identifying sentimentation spikes that can be related to the events external to the games, e.g. game updates, promotions, bug fixes, or controversies. Time-series perspective will provide a critical contextual background regarding how the perception of the people will change over time and respond to changes that are made in-game or by the community.

Another bar chart has been presented to indicate the most reviewed games. In this chart games are ranked relating to the total amount of reviews the game has obtained, which gives an impression of which games are most actively debated or touched within the community. This visualisation can be helpful to consider the most popular or relevant titles as the high amount of reviews can serve as a strong indicator of a certain popularity or relevance in a community.

Also, the bubble chart was applied to generate a visual map of games in terms of volume of review and prevailing mood. Every bubble in this chart is a game; the bigger the bubble the more

reviews the game has, and the colour that it has reflects the general mood. This dual-metric visualisation helps to distinguish games on the basis of popularity and, at the same time, on the basis of their provocation on both positive and negative emotional responses. It is especially useful in pinpointing outliers: games that have received a large amount of reviews, but are mostly negative, or small genre games that happen to have a very positive reception.

The whole dashboard is interactive in nature. Filters were introduced to enable one to search through the data based on the type of criteria like the name of the game, type of sentiment, and time duration. Such filters allow exploration at a dynamic time scale and offer the user a degree of freedom to concentrate on specific areas of interest without looking through the whole set of data at once. This is an interactive affair, which makes the dashboard even more convenient to work with, and gives one the means of a more individualized and informed process of analysis.

5.3 Insights

The created visualizations in the dashboard provide a set of very useful insights that can be used in informed decision-making by game developers, publishers, and community managers. These insights indicate the sentiment trends of a user and indicate where one can improve on their reviews and where one can engage.

- Time Series Game Sentiment

The dashboard allows one to track the sentiment of the user on the numerous games within a continuous manner. Through the ratio of the positive, negative, and neutral reviews, the stakeholders can get a quick impression of a general reception and find out any changes in the level of player satisfaction that can lead to their startling of a problem or a change of improvement.

- Evaluation of the Effect of Game Events and Updates

Sentiment Trend Over Time graph enables the assessment of the impact of individual events e.g. updates, bug fixes, or promotions actions, on the player sentiment. Sudden

decrease in the positive sentiment, say, can suggest people are not satisfied with a new change, whereas growth can certify effectiveness of a new feature or event.

- Sentiment comparison Between Games

The sentiment by game breakdown gives a parallel answer in terms of the perception of various titles by game community. Such comparison will assist in determining which games are gaining the most positive feedback and which of them may need special attention as the percentage of negative or neutral feedback is higher.

- The process of Determining the improvement areas

Games that have a significant share of negative sentiment could reflect a problem deeper than performance problems, gameplay imbalances, or user interface issues. Being aware of these trends would allow development teams to prioritize addressing the areas of the greatest concern whenever they relate to the satisfaction of the players.

- Identifying the Strengths and the Preferences of the Player

The titles that have mainly positive reviews provide ideas of features and features of the design that can attract users. The factors which were successful can be used as a reference set when updating the game in the future or drawing up new titles so that the work of the developers accurately meets the expectations of the players.

- Data-Driven Decision making Enablement

The dashboard brings the sentiment labeled feedback to presentable and interpretive pieces of evidence. This enables different teams, e.g. product development, community management, and marketing teams to take strategic decisions that develop user experience, promote retention, and guide them towards long-term participation.

6. Optimization & Comparison

6.1 Model or Architecture Improvements

This section dives into the upgrades and tweaks made to the sentiment classification models, along with the core framework for processing data in real-time. The goal here is to boost the overall performance, accuracy, and effectiveness of the entire sentiment analysis process.

6.2 Sentiment Model Enhancement

In this project, we explored two different methods for sentiment analysis: the Naive Bayes Classifier and a Long Short-Term Memory (LSTM) neural network. Every model has its own preprocessing and training methods and a number of enhancements that may improve its results even further.

6.2.1 On Naive Bayes Model Incrementalities

TF-IDF vectorization (max_features=5000) was used to convert the reviews into numerical attributes into which the Naive Bayes model was run.

So, what are the standard TF-IDF feature engineering:

- N-grams:

It will also be possible to add n-grams (such as bigrams or trigrams), which will allow identifying sentiment-rich phrases like not fun or very immersive and will make models better detect subtle sentiment usage in Steam reviews.

- Lexicon based features:

Adding features, such as external sentiment lexica like VADER, in combination with TF-IDF can substantively increase the propensity of the model to detect semantic signals of polarity, thereby increasing accuracy of classification of casual gamer language, or reviews of any type.

Model Ensemble:

The predictions of the Naive Bayes can be combined with those of a Logistic Regression or a Decision Tree to form an ensemble model, which can use the strength of each of the two different classifiers to be more robust.

6.2.2 Enhancements of LSTM Model

The LSTM model used in this project had used a pre-trained tokenizer.pkl and labelencoder.pkl with the reviews getting tokenized and padded with maximum of 200 words..

Specific Opportunities for LSTM Optimization in Projects:

1. Hyperparameter Tuning:

- **Maxlen Adjustment:** Any maxlen values may be tried to find a balance between preserving the context of reviews (such as those rigorous playthroughs) and the efficiency in the calculating methods.
- **LSTM Units & Layers:** Consider increasing the LSTM units from 64 to 128 or adding more layers to enhance the model's ability to learn complex language patterns.
- **Dropout Rates:** Fine-tune dropout layers (currently at 0.5) to balance overfitting prevention with model generalizability.

2. Pre-trained Embeddings:

Swap out the embedding layer for Word2Vec, FastText, or GloVe embeddings that have been trained on gaming or Bahasa Melayu datasets. This can significantly boost semantic understanding beyond what the tokenizer learned during its training.

3. Transformer-Based Models:

Future improvements could involve using BERT or RoBERTa models fine-tuned on Malaysian gaming reviews for better contextual embeddings, which would greatly enhance sentiment accuracy, albeit with higher computational demands.

4. Model Quantization:

It's crucial to reduce memory usage and speed up inference by quantizing the LSTM model. This is essential for real-time integration with Spark while still keeping accuracy intact.

6.3 Architecture Improvement

The project brings together Apache Kafka for streaming, leverages Apache Spark Structured Streaming for processing, and outputs results to both CSV files and console displays. By making some improvements, we can enhance scalability, reduce latency, and increase reliability in operations.

6.3.1 Optimization in Apache Kafka

■ Relating Directly to Your Implementation:

1. Topic Partitioning:

At present, the subject of game reviews has restricted divisions. By increasing partitions, more Spark tasks will be able to read data concurrently, improving processing efficiency for extensive game review datasets.

2. Producer Configuration:

In your kafka_producer.py, tuning producer parameters:

- Raise batch_size over 16384 to improve network efficiency.
- Modify linger_ms to strike a balance between batching delay and latency needs.
- Maintain acks='all' for message persistence in production.

3. Replication Factor:

The existing Docker configuration employs one broker without any replication. To enhance production, raising the replication factor above 1 guarantees message persistence in the event of broker failure

6.3.2 Apache Spark Streaming Improvements

1. Resource Allocation:

Adjust Spark submit settings to configure:

- Set spark.executor.memory to a minimum of 2-4GB for loading ML models.
- spark.executor.cores to align with available CPUs for concurrent task processing
- spark.executor.instances determined by the size of the dataset and the capabilities of the cluster

2. Micro-Batch Interval Tuning:

Modify micro-batch timings to achieve a balance between:

- **Reduced latency:** Shorter batch intervals (e.g. 5 seconds) with increased overhead.
- **Increased throughput:** Bigger batch intervals (e.g. 30 seconds) with minimized context switching.

3. UDF Inference Optimization:

Predict_sentiment UDF is designed to handle a single row at a time. Transforming it into a Pandas UDF (vectorized UDF) can handle batches effectively, minimizing serialization costs.

4. Checkpointing:

Use Spark checkpointing to enable recovery of the pipeline without needing to start over from the beginning in the event of system failures.

5. Output Sink Improvements:

At present, results are saved in CSV files. For real-time dashboarding that can scale:

- Integrate with Elasticsearch for instant data querying and dashboard visualizations.

6

7. Conclusion & Future Work

7.1 Conclusion

This project effectively showcased the creation and implementation of a real-time sentiment analysis system for Steam game reviews utilizing Apache Kafka and Apache Spark Structured Streaming.

Major accomplishments consist of:

- Creating a Kafka producer to effectively stream reviews from CSV files to the game-reviews topic.
- Creating a Kafka consumer that receives these messages, executes TextBlob sentiment analysis, and saves the outcomes for future applications.
- Creating a Spark Streaming application that ingests reviews from Kafka, parses, processes, and compiles them for analytical insights.
- Evaluating two models:
 - Naive Bayes classifier demonstrated robust baseline results using TF-IDF features.
 - LSTM neural network, which understood sequential context more effectively despite greater computational demands.

The pipeline seamlessly combined data ingestion, real-time processing, and model inference, establishing a scalable basis for live sentiment analysis in gaming platforms.

7.2 Future Work

Though this project achieved its goals, numerous improvements can be made to enhance its readiness for production and analytical functions:

Enhancements to the Model:

- Refine the LSTM model using optimized hyperparameters and incorporate pre-trained embeddings (like Word2Vec or FastText) to enhance semantic comprehension.
- Utilize transformer models such as BERT for enhanced context-driven sentiment analysis, boosting classification precision on intricate review sentences.

Improvements in Architecture:

- Boost the number of Kafka topic partitions and Spark executors to enhance parallel processing and throughput for large-scale review data.
- Implement the pipeline on cloud-managed services like AWS EMR, MSK, or Databricks for scaling, oversight, and resource enhancement.
- Substitute the existing CSV storage with Elasticsearch or Apache Druid to facilitate real-time indexing and interactive dashboard visualizations with minimal latency.

Enhancements in Operations:

- Use model quantisation or TensorRT optimisation to run inference faster in Spark Streaming pipelines.
- Turn on Spark checkpointing and have a reliable data store that can prevent losing data and recover pipelines in case of failure.

Application Growth:

- Add more data sources to the pipeline such as tweets of the Twitter gaming populace or YouTube game reviews to generate broader multi-platform sentiments.
 - Develop user-friendly Streamlit or Tableau dashboard connected to the live data sources, so that the stakeholders could easily follow and study the trends in the evolution of the public sentiment.
- These future improvements will make the real-time sentiment analysis pipeline more accurate, scalable, and reliable and deliver analytical insights that will be used by game developers and marketers and platform managers in making informed decisions that would maximize user experience and satisfaction in the competitive gaming sector.

8.0 References

- Apache Kafka.* (n.d.). Apache Kafka. <https://kafka.apache.org/documentation/>
- Structured Streaming Programming Guide - Spark 4.0.0 Documentation.* (2025).
- Apache.org. <https://spark.apache.org/docs/latest/streaming/index.html>

- Bird, S., Ewan Klein, & Loper, E. (2009). *Natural Language Processing with Python*. ResearchGate; O'Reilly.
https://www.researchgate.net/publication/220691633_Natural_Language_Processing_with_Python
- TextBlob. (2018). *TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation*. Readthedocs.io. <https://textblob.readthedocs.io/en/dev/>
- Scikit-learn. (2024). *scikit-learn: Machine Learning in Python*. Scikit-Learn.org. <https://scikit-learn.org/stable/>
- Olah, C. (2015, August 27). *Understanding LSTM Networks*. Colah's Blog. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Scikit-learn. (2019). *1.9. Naive Bayes — scikit-learn 0.21.3 documentation*. Scikit-Learn.org. https://scikit-learn.org/stable/modules/naive_bayes.html
- Pascual, F. (2022, February 2). *Getting Started with Sentiment Analysis using Python*. Huggingface.co. <https://huggingface.co/blog/sentiment-analysis-python>

9.0 Appendix

- Code snippets for preprocessing and model training
- Kafka and Spark configuration logs

- Sample outputs from Elasticsearch queries

```
import requests
4 import pandas as pd
import time
from textblob import TextBlob
import matplotlib.pyplot as plt
games = {
    "Counter-Strike 2": "730",
    "Dota 2": "570",
    "PUBG: Battlegrounds": "578080",
    "Apex Legends": "1172470",
    "GTA V": "271590",
    "The Witcher 3": "292030",
    "Cyberpunk 2077": "1091500",
    "Rust": "252490",
    "Elden Ring": "1245620",
    "ARK: Survival Evolved": "346110",
    "Left 4 Dead 2": "550",
    "Team Fortress 2": "440",
    "DayZ": "221100",
    "Stardew Valley": "413150",
    "Among Us": "945360",
    "Red Dead Redemption 2": "1174180",
    "Terraria": "105600",
    "PAYDAY 2": "218620",
    "Don't Starve Together": "322330",
    "Rainbow Six Siege": "359550",
    "No Man's Sky": "275850",
    "Hogwarts Legacy": "990080",
    "Lethal Company": "1966720",
    "Palworld": "1623730",
    "Baldur's Gate 3": "1086940",
    "Hades": "1145360",
    "Sea of Thieves": "1172620",
    "Phasmophobia": "739630",
    "The Forest": "242760",
    "Project Zomboid": "108600"
}
```

CUSTOMER REVIEWS FOR COUNTER-STRIKE 2

Reviews written before 28 Sep, 2023 @ 6:30am were for CS:GO, the legacy version of Counter-Strike 2.

Overall Reviews: Very Positive (8,876,811 reviews)	Recent Reviews: Very Positive (67,903 reviews)
REVIEW TYPE ▾ PURCHASE TYPE ▾ LANGUAGE ▾ DATE RANGE ▾ PLAYTIME ▾ DISPLAY ▾	Show graph

Filters Your Languages

Showing 2,373,717 reviews that match the filters above (**Very Positive**)

MOST HELPFUL REVIEWS IN THE PAST 30 DAYS

- Tududu du
4 products in account
1 review

Recommended
722.5 hrs on record (709.9 hrs at review time)

★

POSTED: 19 JUNE

The time I spent playing this game could've been spent...
 >studying
 >applying for jobs
 >making friends irl
 >getting a gf
 >spending time with my family
 >working out
 >becoming a better man in general

From the time I spent in this game, I got...
 >an extremely racist and homophobic vocabulary
 >free Russian lessons
 >anger management issues

[READ MORE](#)

Was this review helpful?
 Yes No Funny Award

228 people found this review helpful
 120 people found this review funny

2 8 2 8

8

zbyneekcigan
50.0 hrs
★

POSTED: 7 JULY

love hearing 7yo russian kids scream

Helpful?
 Yes No Funny Award

ariekaravaev
87.4 hrs
★

POSTED: 7 JULY

Helpful?

Yes No Funny Award

Petras
308.6 hrs
★

POSTED: 7 JULY

★★★★ game = ragequit do not play this ★★★★★ game

Helpful?
 Yes No Funny Award

CUSTOMER REVIEWS FOR DOTA 2

Overall Reviews:
Very Positive (2,515,771 reviews)

REVIEW TYPE ▾ PURCHASE TYPE ▾ LANGUAGE ▾ DATE RANGE ▾ PLAYTIME ▾ DISPLAY ▾

Show graph

Recent Reviews:
Mostly Positive (17,936 reviews)

Filters Your Languages

Showing 789,161 reviews that match the filters above (**Very Positive**)

MOST HELPFUL REVIEWS IN THE PAST 30 DAYS

OnnySix
1 review

Recommended
1,066.2 hrs on record (1,019.0 hrs at review time)

POSTED: 12 JUNE

Best game,
This game make me like at zoo with the entry ticket Is my soul,
if you never going to zoo this game will give you an experience
like at the zoo with so many Animal inside of it.

Was this review helpful?
thumb up Yes thumb down No smiley Funny award Award

42 people found this review helpful
27 people found this review funny

2

w3akdaze
3 reviews

Recommended
1,700.1 hrs on record (1,686.9 hrs at review time)

POSTED: 16 JUNE

Started playing this game because I was gambling on it without
having any idea what was going on. Fast forward I have 1600
hours and I am hardstuck 800 mmr because the only people
that play are smurfs. I would not trade my experience on this
game for anything in the world. Except for my grandma back.

3

RECENTLY POSTED

Faker
9.1 hrs

POSTED: 7 JULY

Helpful?

thumb up Yes thumb down No smiley Funny award Award

spf7001
6,131.8 hrs

POSTED: 7 JULY

A little addictive but good if you can pace
yourself.

Helpful?

thumb up Yes thumb down No smiley Funny award Award

Aphlatoon
2,006.3 hrs

POSTED: 7 JULY

good

Helpful?

thumb up Yes thumb down No smiley Funny award Award

CUSTOMER REVIEWS FOR PUBG: BATTLEGROUNDS

Overall Reviews:
Mixed (2,573,584 reviews)

REVIEW TYPE ▾ PURCHASE TYPE ▾ LANGUAGE ▾ DATE RANGE ▾ PLAYTIME ▾ DISPLAY ▾

Show graph

Recent Reviews:
Mixed (17,320 reviews)

Filters Your Languages

Showing 430,851 reviews that match the filters above (**Mixed**)

MOST HELPFUL REVIEWS IN THE PAST 30 DAYS

eDrinker
165 products in account
14 reviews

Not Recommended
1,749.6 hrs on record

POSTED: 11 JUNE

Update: After 20 days of my ban I received a message on Steam that stated.

"We've removed the incorrectly applied In-Game Ban on behalf of the PUBG: BATTLEGROUNDS team".

No explanation, no apology.

After 1,700 hours, banned without reason.
I've played PUBG since it launched in 2017, paid full price, and never once cheated in any online game. After years of on and off play, I suddenly received a game ban. I've checked my login history (only trusted devices), scanned my PC (no malware), and

[READ MORE](#)

Was this review helpful?

Yes No Funny Award

273 people found this review helpful
26 people found this review funny

4 8 3 7

RECENTLY POSTED

GuarjohN
2,767.7 hrs
POSTED: 7 JULY

There are many cheaters here in ASIA specially in ranked. Very frustrating to see these cheaters run rampant that it's not enjoyable to play anymore. They aren't even new accounts. Some of them are lv 500 and with progressive skins like I do. I feel like system is not banning high level accounts or high value accounts and just flag them as false ban. Everyday I play RANKED SQUADS and report cheaters. Everyday I get notifications on my reports that they got punished. Yes maybe it works? Play again and report again. You can clearly see the BAN reports and statistics that PUBG releases that

[READ MORE](#)

Helpful?

Yes No Funny Award

coocchoo
6 reviews
POSTED: 15 JUNE

Not Recommended
318.0 hrs on record (311.8 hrs at review time)

21

tosum ajaa päältä autolla
mina heittää hanta pannu

Helpful?

Yes No Funny Award

CUSTOMER REVIEWS FOR APEX LEGENDS™

Overall Reviews:
Mixed (999,944 reviews)

REVIEW TYPE ▾ PURCHASE TYPE ▾ LANGUAGE ▾ DATE RANGE ▾ PLAYTIME ▾ DISPLAY ▾

Filters Your Languages

Showing 424,557 reviews that match the filters above (**Mostly Positive**)

Recent Reviews:
Mixed (6,567 reviews)

Show graph

MOST HELPFUL REVIEWS IN THE PAST 30 DAYS

Sunshine 2 reviews Not Recommended 1.1 hrs on record

POSTED: 18 JUNE

After spending 5 years on origin, I decided to play on steam. after linking steam to my ea account the game crashed and kept giving me network error after that. Now they've locked me out of my ea account making me unable to play on ea and steam both. I made case on customer support. Proceeds to ignore my case for 2 days now, thus preventing me from accessing the 7 heirloom, countless legendary skins id owned. Rip my hard-earned money and thousands of my hours i gave to apex which i am never getting back. its like they don't even care to help me recover my account. I lost everything

[READ MORE](#)

Was this review helpful? Yes No Funny Award

149 people found this review helpful
24 people found this review funny

4 2 15

ASTRO 1 review Not Recommended 2,049.1 hrs on record (2,029.5 hrs at review time)

POSTED: 22 JUNE

To the discerning public,

Apex Legends is a most chaotic and frustrating diversion.

RECENTLY POSTED

Bi-Yan 101.1 hrs

POSTED: 7 JULY

i shoot, firing sounds good, hitting sounds good, i feel good and i like

Helpful? Yes No Funny Award

mixbouz15 12.4 hrs

POSTED: 7 JULY

good

Helpful? Yes No Funny Award

ui beam 1,130.1 hrs

POSTED: 7 JULY

Disgusting game.

Helpful? Yes No Funny Award

Break 458.8 hrs

POSTED: 7 JULY

CUSTOMER REVIEWS FOR GRAND THEFT AUTO V ENHANCED

Overall Reviews:
Mixed (64,181 reviews)

REVIEW TYPE ▾ PURCHASE TYPE ▾ LANGUAGE ▾ DATE RANGE ▾ PLAYTIME ▾ DISPLAY ▾

Show graph

Filters Your Languages

Showing 26,480 reviews that match the filters above (Mixed)

Recent Reviews:
Mixed (11,680 reviews)

MOST HELPFUL REVIEWS IN THE PAST 30 DAYS

bruh 82 products in account 20 reviews

Not Recommended 30.8 hrs on record (29.6 hrs at review time)

POSTED: 13 JUNE

Preventing the game's anti-cheat work on Linux (as well as Steam Deck) PURPOSELY, while it is capable of running originally and technically without extra effort, just to blame the Linux platform as the source of in-game hackers, whereas 99.9999% of the hacks available is for Windows, is such a HUGE F***ING ***** MOVE ROCKSTAR. The funny part is that these cheaters can easily bypass the anti-cheat. In the end, nothing really changes except the iniquitous gamer experience for the Linux gaming community which these firms continuously f*cks again and again. There is no explanation other than a secret behind-the-shadow contract in between R* and Microsoft to bork the Linux support. I'm literally freaking out.

[READ MORE](#)

Was this review helpful?

Yes No Funny Award

761 people found this review helpful
21 people found this review funny

13 6 11 16

RECENTLY POSTED

.leepx0D 6.8 hrs

POSTED: 7 JULY

I can absolutely de olish obese and pregnant women

Helpful?

Yes No Funny Award

Sestra ti 8.0 hrs

POSTED: 7 JULY

i like it its good you can do everything you've ever wanted

Helpful?

Yes No Funny Award

ekambrar0082 38.7 hrs

POSTED: 7 JULY

GOOD

Helpful?

Yes No Funny Award

G0LD3N 31.5 hrs

POSTED: 7 JULY

HPDP Project 2.pdf

ORIGINALITY REPORT



PRIMARY SOURCES

- | | | |
|---|---|------|
| 1 | Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023
Publication | <1 % |
| 2 | Submitted to Universiti Teknologi Malaysia
Student Paper | <1 % |
| 3 | Submitted to BPP College of Professional Studies Limited
Student Paper | <1 % |
| 4 | Yasha Hasija, Rajkumar Chakraborty. "Hands-On Data Science for Biologists Using Python", CRC Press, 2021
Publication | <1 % |
| 5 | linguist.page
Internet Source | <1 % |
| 6 | digital.library.adelaide.edu.au
Internet Source | <1 % |
| 7 | scholar.archive.org
Internet Source | <1 % |
| 8 | www.coursehero.com
Internet Source | <1 % |

<1 %

9

www.floridabuilding.org

Internet Source

<1 %

10

pergamos.lib.uoa.gr

Internet Source

<1 %

11

Bassi, Tara. "Past, Present and Future Sentiments Towards Cad [Computer Aided Design]", Harvard University, 2024

Publication

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On