REALTIME SOCKET STREAMING ON YELP DATASET

Group 2:
Aishwarya Gulab
Sheetal Patnaik
Lincy Rebello
Dhruv Shetty

CONTENT

01 INTRODUCTION

DATASET

SYSTEM ARCHITEC

ALGORITHMS

DEMO

INTRODUCTION

Overview

This project presents a real-time data streaming pipeline for sentiment analysis and data indexing, processing 7 million Yelp customer reviews. Leveraging TCP/IP sockets, Apache Spark, Apache Kafka and AWS services, our system demonstrates large-scale data handling capabilities.

Objective

To develop a real-time data pipeline integrating cutting-edge technologies for efficient processing, analysis, and storage of large-scale data. The project aims to demonstrate the practical application of AI in streaming workflows, addressing the full data lifecycle and creating a flexible architecture adaptable to various industries requiring real-time insights.

Scope

The system demonstrates scalability, real-time processing, and adaptability across various industries.

02 DATASET

SYSTEM ARCHITECTU

ALGORITHMS

DEMO

CHALLENGES

DATASET

WHY YELP DATASET?

01

Approximately 7 million Yelp customer reviews

02

Rich source of diverse customer feedback

03

Structured review ratings and detailed comments

04

Ideal for sentiment analysis across various industries

03

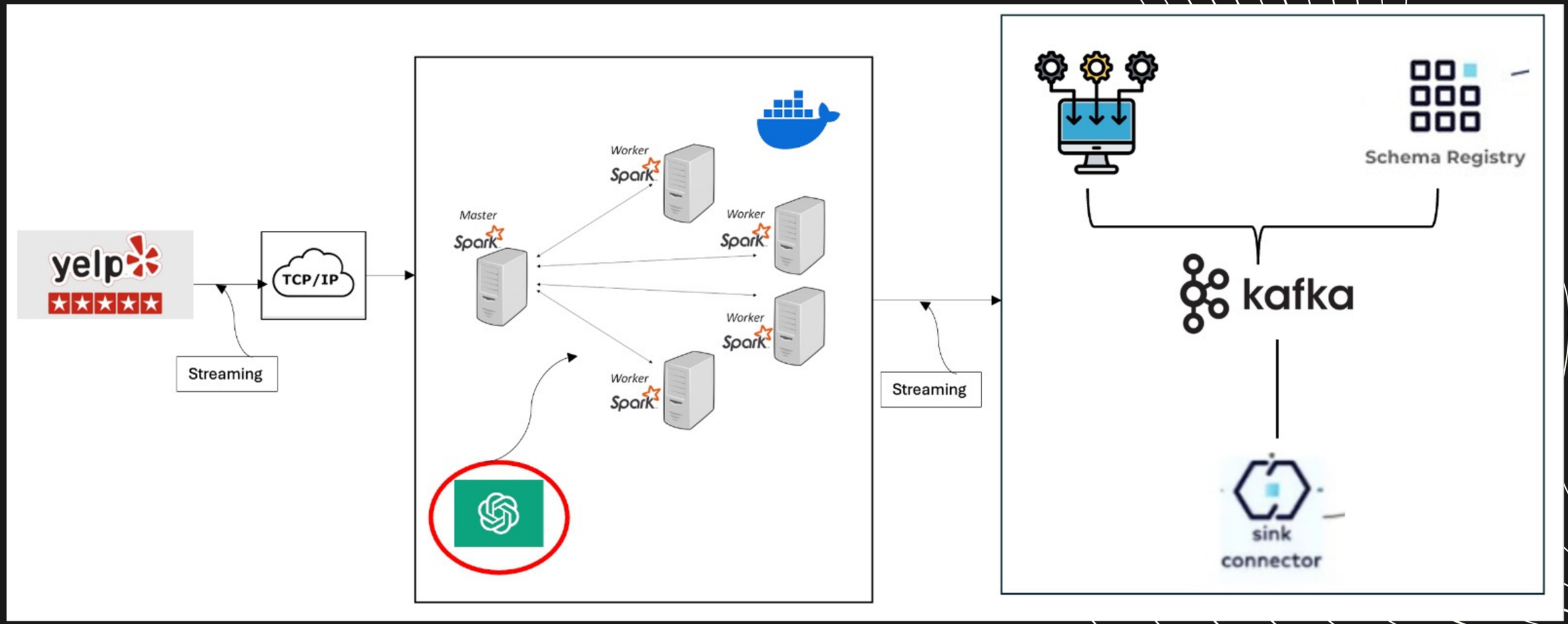
SYSTEM ARCHITECTUR

ALGORITHMS

DEMO

CHALLENGES FACED

SYSTEM ARCHITECTURE



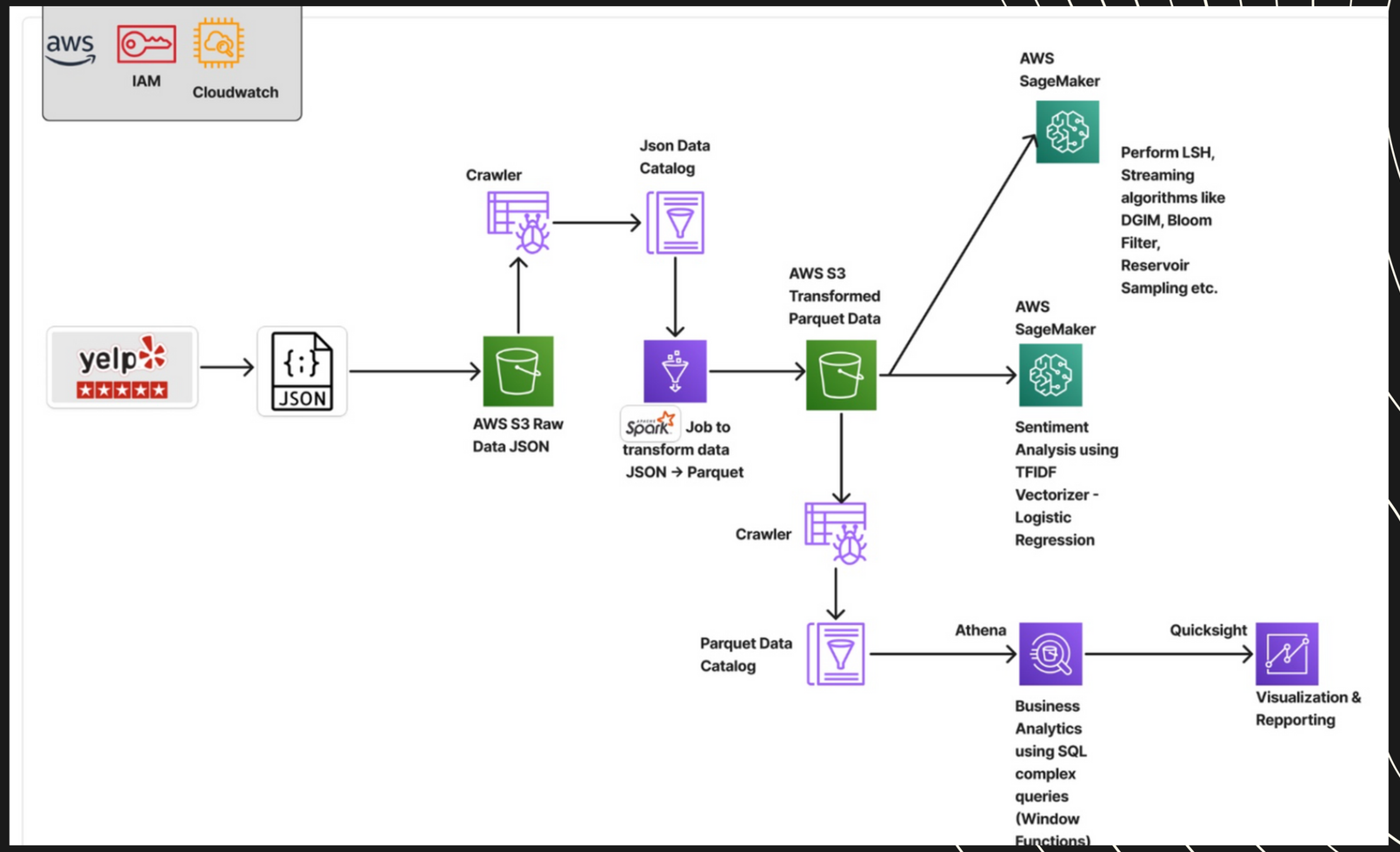
05 DEMO

CHALLENGES FACED

LEARNINGS

DEMO

NEW SYSTEM ARCHITECTURE



04

ALGORITHMS

DEMO

CHALLENGES FACED

ALGORITHMS

01

Reservoir Sampling for Yelp Data Analysis: Randomly analyzing the Yelp reviews is performed by Reservoir Sampling, keeping the equal probability of every review that is in the dataset.

02

Bloom Filter for Duplicate Detection: Using the Bloom Filter algorithm, we efficiently detected 551 potential duplicate businesses from 150,346 entries.

03

DGIM Algorithm for Monitoring Stream Counts: we monitored review patterns over time, analyzing daily and weekly review counts along with average star ratings.

05

K-Anonymity for Privacy Preservation: (k=5), we anonymized 50,000 records by generalizing location, dates, and star ratings. Post-anonymization, unique combinations were reduced to 40,641, with 504 groups satisfying K-anonymity, ensuring privacy protection while retaining data utility for analysis.

04

Locality-Sensitive Hashing for Similar Reviews: Using Locality-Sensitive Hashing (LSH), we identified 1,000 highly similar review pairs from 10,000 reviews based on their TF-IDF vectors. The algorithm efficiently clustered reviews with similarity scores above 0.7, providing insights into overlapping opinions or duplicate content, and highlighted correlations in star ratings.

06

Explainable AI for Sentiment Analysis: AI with LIME, we analyzed Yelp review predictions to identify key words influencing sentiment classification. Words like 'exceptional' and 'bad' had the most significant impact on predictions, with detailed feature contributions provided for negative, neutral, and positive sentiments.

07

Machine Unlearning: Using machine unlearning, we modified our sentiment model to forget specific negative reviews without retraining the entire model. After unlearning, the model's confidence in the forgotten review decreased from 71.8% to 56.5%.

05 DEMO

CHALLENGES FACED

LEARNINGS

DEMO

06 CHALLENGES FACED

LEARNINGS

01

COST MANAGEMENT

Managing costs was challenging due to the high expenses of real-time and batch processing pipelines for services like SageMaker.Hence, the aim was to use some more compact format like Parquet in batch processing to achieve cost mitigation.

02

SCHEMA EVOLUTION

Frequent schema changes caused integration issues with tools like AWS Athena. Implementing consistent schema definitions and using AWS Glue Data Catalog ensured uniformity across the pipeline.

03

LEARNING CURVE

Integrating multiple tools such as Kafka, Spark, AWS Glue, and SageMaker involved a steep learning curve. Effective collaboration and iterative testing were essential to overcome these challenges and achieve smooth interoperability.

04

SCALABILITY

Real-time streaming stressed infrastructure with high data ingestion, causing latency issues, while batch processing required efficient resource management for large-scale transformations. Strategies like data partitioning and distributed computing were employed to address these challenges.

CHALLENGES FACED

07 LEARNINGS

PHASE: 1

Gained hands-on experience in setting up data pipelines for dynamic analytic

Worked with high-velocity data using Spark Streaming for efficient data processing.

Better Understood the role of message brokers like Kafka in maintaining seamless data flow.

applied models for sentiment classification, balancing latency and accuracy trade-offs.

PHASE: 2

Processed large-scale datasets using PySpark in AWS Glue for efficient transformations.

Optimized data storage and query performance by converting data into Parquet format.

Performed complex SQL analysis using AWS Athena, including advanced window functions.

Built a modular architecture by integrating AWS services for scalability and maintainability.

THANK YOU