Automated Reddit Data Pipeline for Airline Sentiment Analysis Using Prefect and AWS S3

Panchami Dinesh*, Maanya Anil†, Meghana G‡
Department of Computer Science and Engineering, RV University, Bengaluru, India
*panchamid.btech23@rvu.edu.in, †maanyaa.btech23@rvu.edu.in, ‡meghanag.btech23@rvu.edu.in

***Abstract*—Social media platforms such as Reddit provide valuable insights into public opinions and customer experiences related to airline services. This paper presents an automated data pipeline designed to extract, process, and store Reddit posts mentioning major airlines including Emirates, Lufthansa, Cathay Pacific, and others. The system leverages the Prefect orchestration framework to automate the Extract–Transform–Load (ETL) workflow, ensuring seamless data collection via the Reddit API (PRAW) and storage on Amazon S3 for long-term accessibility and analysis. The pipeline is deployed and scheduled using Prefect Cloud, enabling periodic data retrieval and near real-time sentiment monitoring. This framework supports scalable, continuous collection of airline-related data, forming a foundation for sentiment analysis and trend prediction in customer satisfaction.**

***Index Terms*—Reddit Data Extraction, Airline Sentiment Analysis, Prefect, AWS S3, ETL Pipeline, Automation, Social Media Analytics**

I. Introduction

In the digital age, social media platforms have become vital channels for users to share their experiences, opinions, and feedback on various services. Reddit, one of the largest discussion-based platforms, hosts a vast amount of user-generated content that reflects real-world sentiments and public perception across industries. In the aviation sector, discussions on airline performance, customer service, and travel experiences provide a valuable resource for analyzing customer satisfaction and identifying service improvement areas.

Airlines and researchers can leverage such online discourse to monitor brand reputation, assess customer grievances, and predict emerging trends in passenger expectations. However, manually gathering and analyzing this data from Reddit poses significant challenges due to the platform's large volume, unstructured nature, and continuous data generation. To address these challenges, automation and cloud-based data pipelines offer efficient, scalable, and real-time solutions for data acquisition and management.

This paper presents an automated Reddit data pipeline designed to collect, process, and store airline-related discussions for sentiment analysis. The proposed system integrates the Reddit API with the Prefect orchestration framework to automate the Extract–Transform–Load (ETL) process. Collected data is systematically stored on Amazon Simple Storage Service (AWS S3), serving as a centralized data lake for further analysis and visualization. The pipeline is scheduled via Prefect Cloud, enabling periodic execution without manual intervention. This approach not only simplifies data engineering workflows but also establishes a robust foundation for sentiment analytics, enabling continuous monitoring of public sentiment toward major airlines such as Emirates, Lufthansa, and Cathay and others..

II. Literature Review

[1] Identifies Apache Airflow as one of the most prominent open-source workflow orchestration platforms in modern data engineering. Their study highlights Airflow's Directed Acyclic Graph (DAG) architecture, which enables efficient scheduling, dependency management, and monitoring of complex ETL tasks. The framework's adaptability is reinforced through integration with Kafka for real-time data streaming and support for cloud-based deployments, ensuring scalability and reliability in distributed environments. Airflow's flexibility allows organizations to automate end-to-end data pipelines, simplifying data ingestion, transformation, and analysis workflows. This makes it an essential tool for large-scale data processing where performance and fault tolerance are critical. [2] Demonstrates a real-world implementation of Airflow on Google Cloud Platform (GCP), showcasing its built-in operators, task dependencies, and parallel execution capabilities. The study emphasizes how Airflow's orchestration features improve workflow performance and scalability, streamlining ETL operations and facilitating data preparation for analytics and machine learning pipelines. Overall, these works position Apache Airflow as a foundational orchestration framework for managing complex, automated, and cloud-integrated data engineering pipelines.

[3] Explore Amazon Simple Storage Service (S3) as a core component of modern distributed storage systems within cloud infrastructures. Their study highlights S3's durability and fault tolerance achieved through automatic replication across multiple regions, ensuring continuous data availability and integrity. The paper emphasizes S3's seamless integration with analytics and data processing platforms, allowing for consistent, reliable, and large-scale data storage solutions suitable for data-intensive workloads. By leveraging its distributed and scalable architecture, S3 serves as a backbone for cloud-based storage, supporting diverse enterprise analytics and data-driven applications.

[4]Analyzes Amazon Redshift, a data warehouse system that relies heavily on S3 for storage durability and backup. Their work details Redshift's columnar storage design and

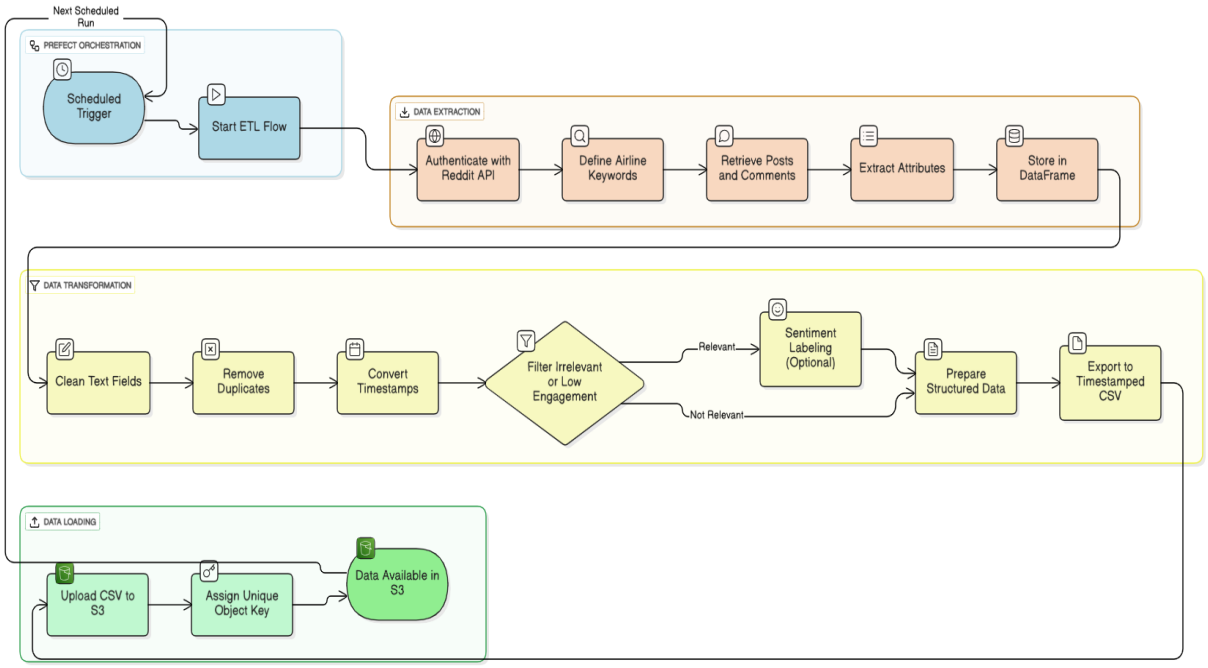

Fig. 1. Architecture of the Reddit-to-S3 ETL Pipeline with Prefect Orchestration

massively parallel processing (MPP) capabilities, which enable high-performance querying across petabyte-scale datasets. The authors note that S3's reliability and elasticity form the foundation for Redshift's analytical efficiency, enabling fast, cost-effective, and scalable analytics in enterprise data ecosystems. Together, these studies position Amazon S3 as a key enabler of distributed data storage and analytics, providing the underlying infrastructure for scalable, resilient, and high-throughput data systems in cloud environments.

[5]Recent advancements in data engineering have underscored the critical role of efficient data orchestration, governance, and analytics in managing large-scale and heterogeneous datasets. Introduced an integrated framework incorporating tools such as Apache Spark, Apache Airflow, and Kafka to facilitate scalable data ingestion, transformation, and analytical processing within distributed environments. This framework highlights how orchestration layers can effectively streamline both batch and streaming data workflows while ensuring data quality, consistency, and compliance. Subsequent research has extended these ideas through the adoption of cloud-native architectures that leverage data lakes—such as Amazon S3—and automated workflow orchestration systems for real-time data analytics. Nevertheless, existing enterprise-level orchestration tools like Airflow often present steep learning curves and demand significant computational resources, posing challenges for academic and small-scale implementations. To mitigate these limitations, lightweight workflow orchestrators such as **Prefect** have emerged as viable alternatives, offering enhanced usability, simplified scheduling, and intuitive dependency management. Prefect's seamless integration with cloud platforms and APIs, including AWS S3 and Reddit, enables efficient automation and monitoring of data pipelines. This evolution signifies a broader shift toward accessible, resource-efficient, and cloud-oriented orchestration frameworks that promote scalability, transparency, and reproducibility in data-driven research environments.

III. Methodology

The project follows an end-to-end Extract–Transform–Load (ETL) pipeline (Fig. 1) for collecting, processing, and storing Reddit data related to airline discussions. The workflow is implemented in Python using open-source libraries such as PRAW for data extraction, pandas for transformation, and AWS S3 for cloud-based storage.

The entire process is orchestrated using Prefect, which automates task execution, scheduling, and monitoring. Prefect was chosen over Apache Airflow because it offers an easier setup, Python-native syntax, and a lightweight structure suitable for students and research projects. It can be run directly from scripts without requiring additional infrastructure like web servers or schedulers.

All tools and platforms used—Reddit's public API, Prefect's free community version, and the AWS Free Tier—make the project completely cost-effective and accessible for academic experimentation.

A. Data Extraction

The data extraction phase retrieves Reddit posts and comments mentioning major airline names such as *Emirates*, *Lufthansa*, *Qatar Airways*, and *Cathay Pacific*. The Python Reddit API Wrapper (PRAW) is used to authenticate and

access Reddit's public API. Queries are defined using airline-related keywords, and relevant attributes such as post titles, timestamps, upvotes, comment counts, and subreddit names are extracted. The extracted data is stored temporarily in Pandas DataFrames for further transformation.

B. Data Transformation

Once extracted, the data undergoes transformation to ensure consistency and usability. This involves cleaning text fields, removing duplicates, converting timestamps to standard formats, and filtering irrelevant or low-engagement posts. Sentiment labels can be assigned in later stages using Natural Language Processing (NLP) models. The structured data is then prepared for export to a CSV format. Each execution of the pipeline generates a timestamped file (e.g., `reddit_airline_posts_20251015.csv`) for version control and reproducibility.

C. Data Loading

The processed CSV file is uploaded to an Amazon Web Services (AWS) S3 bucket (fig-2) using the Boto3 library. S3 provides a reliable and scalable cloud storage solution for preserving large volumes of collected Reddit data. Each dataset version is stored under a unique object key, allowing future retrieval, comparison, and downstream analysis. This cloud-based design ensures that all team members and systems can securely access data without relying on local storage.

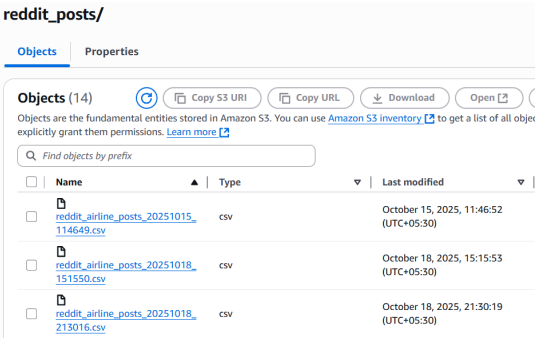

Fig. 2. aws-s3

D. Orchestration with Prefect

The workflow automation and scheduling are handled by the Prefect orchestration framework. The main flow is defined in the `reddit.py` file:

```
from prefect import flow
from reddit_to_s3 import reddit_to_s3_flow

if __name__ == "__main__":
    reddit_to_s3_flow.serve(
        name="reddit-flow",
        cron="0 */10 * * *
"   # runs every hour
    )
```

This code deploys the `reddit_to_s3_flow` as a Prefect flow named *reddit-flow*, configured to execute automatically every 10 hour's based on the provided CRON expression. Prefect Cloud provides real-time visibility of task states, allowing users to monitor runs, reschedule flows, or handle retries seamlessly.

IV. Visualization and Analysis

For visualization and analysis, individual CSV datasets were processed using a simple Python script to extract and visualize key insights. The analysis focused on three main aspects: sentiment distribution, airline mentions, and the frequency of positive and negative words. Each dataset was loaded individually from CSV files and processed using a Python-based machine learning script to compute counts, perform sentiment classification, and visualize patterns in the data.

A bar chart fig-3 was generated to represent the distribution of sentiments, showing the overall balance between positive, negative, and neutral feedback. Another graph illustrated the number of mentions for each airline, revealing that Cathay Pacific had the highest number of mentions (4), while airlines such as Emirates, Lufthansa, Delta Airlines, and others had fewer or no mentions.

In addition, a separate visualization captured the most common positive and negative words fig-4 found in the dataset. The results showed that the word "good" appeared six times and "bad" appeared four times, indicating a generally positive tone across user feedback. These visualizations collectively provided a clear, data-driven overview of both sentiment trends and airline popularity, helping to interpret textual data and draw meaningful insights from user opinions.

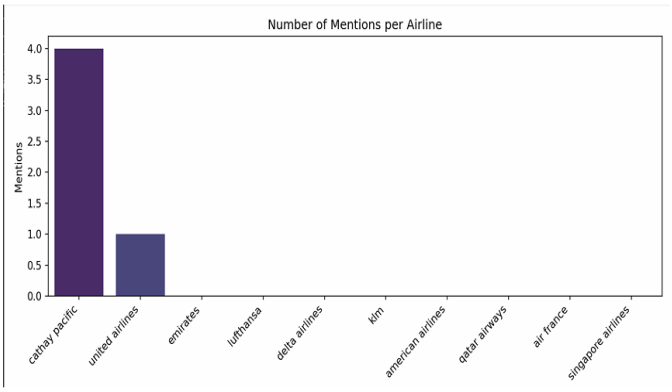

Fig. 3. RAG framework

A. Why Prefect Over Apache Airflow

Although Apache Airflow is an industry-standard orchestration tool, Prefect was chosen for this project due to its lightweight setup, Pythonic design, and user-friendly interface. Prefect eliminates the overhead of configuring DAG files and managing external schedulers, making it ideal for academic and research-oriented projects. Its seamless integration with

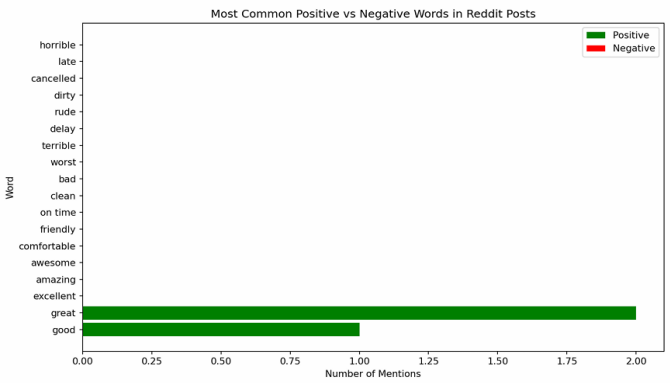

Fig. 4. RAG framework

Python functions allows developers to define and test flows locally while maintaining full cloud-based visibility. This made Prefect more suitable and accessible for a student research pipeline like this one.

V. Future Work

In the future, this project can be enhanced by adopting more scalable and efficient technologies for both data storage and analysis. One major improvement would be to migrate to a paid version of Amazon S3, which offers higher availability, faster retrieval speeds, and enhanced security for managing large-scale datasets. Additionally, integrating workflow orchestration tools such as Prefect can further automate and optimize data pipeline execution, ensuring better reliability and monitoring compared to manual or script-based scheduling. On the analytical side, the project can be extended by implementing a more robust machine learning model capable of handling larger and more complex datasets. Advanced techniques such as transformer-based sentiment models or topic modeling could provide deeper insights into user feedback. Together, these improvements would make the system more scalable, accurate, and suitable for enterprise-level data analysis and decision-making.

References

[1] Anthony Mbata, Yaji Sripada, and Mingjun Zhong. A survey of pipeline tools for data engineering. *arXiv preprint arXiv:2406.08335*, 2024.

[2] Sameer Shukla. Developing pragmatic data pipelines using apache airflow on google cloud platform. *Int J Comput Sci Eng*, 10(8):1–8, 2022.

[3] Om Goel. Enhancing data integrity and availability in distributed storage systems: The role of amazon s3 in modern data architectures.

[4] Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, and Vidhya Srinivasan. Amazon redshift and the case for simpler data warehouses. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1917–1923, 2015.

[5] Yaron David Lipman. An integrated framework for data engineering: Orchestration, governance, and analytics in modern data architectures. *International Journal of Emerging Trends in Computer Science and Information Technology*, 2(3):9–19, 2021.