Project Scoping: Sentiment Analysis-Transforming Reviews into BI

Valli Meenaa Vellaiyan , Niresh Subramanian, Venkata Subbarao Shirish Addaganti, Harshit Sampgaon, Prabhat Chanda, Praneeth Korukonda

1. Introduction

In today's competitive e-commerce landscape, understanding customer feedback is critical for improving product offerings and enhancing the overall customer experience. This project focuses on leveraging sentiment analysis of Amazon reviews to gain deeper insights into customer sentiment. By categorizing reviews into positive, neutral, or negative, businesses like Amazon can extract actionable insights to inform product decisions, optimize customer service, and drive strategic initiatives aimed at improving customer satisfaction.

The goal of this project is to automate the end-to-end process of analyzing review data, from ingestion and pre-processing to model training, deployment, and monitoring. This automated solution not only increases operational efficiency but also helps Amazon to better understand how customers feel about their products and services. By analyzing sentiment trends over time, the model provides valuable insights into key pain points, emerging trends, and areas for improvement across different product categories.

Using advanced sentiment analysis, this project enables Amazon to identify critical feedback faster, enhance the customer experience, and make data-driven decisions that align with business goals such as increasing customer retention, boosting sales, and reducing return rates. This ultimately contributes to a more responsive and customer-focused business strategy, directly impacting business growth.

2. Dataset Information

2.1 Dataset Introduction

The UCSD Amazon Reviews 2023 dataset is a comprehensive collection of customer reviews gathered from various product categories on Amazon. It includes millions of reviews that cover a broad spectrum of customer experiences and sentiments. Each review contains valuable information such as the review text, rating, product category, and metadata like timestamps and customer IDs, which are essential this project. Its size and diversity make it suitable for building scalable machine learning models that can generalize across different product categories.

It will drive the development of an end-to-end Machine Learning Operations pipeline for sentiment analysis. The pipeline will automate data ingestion, preprocessing, and model training to classify reviews as positive, negative, or neutral. The output will be visualized on a dashboard, offering executives and businesses actionable insights on customer sentiment, satisfaction trends, and areas for improvement across product categories, as identified from the reviews.

2.2 Data Card

Dataset Name	UCSD Amazon Reviews 2023
Size	338 million reviews
Format	CSV/JSON

Data Types	String, Numeric, List, Boolean, Dictionary, Timestamps
Storage & Transformation	Data is stored in CSV/JSON formats and will be transformed into embeddings for sentiment analysis tasks, with preprocessing pipelines for cleaning, tokenization, and vectorization.

Features:

- Review Text: The main content of customer feedback
- Star Rating: Ratings from 1 to 5 stars
- Product Category: Categorizes the product reviewed
- Review Timestamp: Date and time of the review
- Product Metadata: Includes product information and details
- Verified Purchase: Marks if the purchase and review genuine or not
- Review Helpfulness: Number of upvotes or downvotes a review received (if available)

Data Quality: The dataset has highly diverse reviews across multiple categories. It contains some noise and potential duplicates. It may also include biases due to review platform mechanics (e.g., more extreme ratings). The metadata, including product categories, timestamps, and star ratings, provides a structured context for the review text. However, certain aspects of the metadata (e.g., missing or inaccurate timestamps, miscategorized products) may require cleaning and validation during the preprocessing stage.

2.3 Data Sources

UCSD Amazon Reviews 2023, a publicly available dataset that aggregates customer reviews and associated metadata such as ratings and timestamps, for Amazon-listed products. The dataset contains reviews sourced from various Amazon product listings, offering a diverse array of consumer feedback on numerous product categories. The dataset is composed not only of customer feedback in textual form but also includes valuable metadata such as product categories, star ratings, and review timestamps. This metadata enables the model to contextualize reviews more effectively, leading to a more accurate classification of sentiment and offering deeper insights into customer behaviour.

By utilizing the UCSD Amazon Reviews 2023 dataset, we adhere to all the licensing and data usage policies provided by the dataset's maintainers. We ensure that our project conforms to ethical and legal standards for data use, including proper handling of anonymized customer data and compliance with any applicable privacy regulations.

2.4 Data Rights and Privacy

- **Data Rights**: The dataset is publicly accessible under non-commercial use agreements, ensuring that it can be leveraged for academic and research projects without infringing on intellectual property rights. However, the data is subject to Amazon's policies regarding public APIs and data usage.

- **Privacy Considerations and Compliance with Data Protection Regulations**: The dataset includes user-generated content, which may contain identifiers such as reviewer IDs and timestamps. These identifiers will be excluded to prevent any privacy breaches and safeguards will be implemented to avoid any misuse of personally identifiable information. The project will adhere to ethical standards, avoiding any linkage between reviews and specific individuals. All analyses will be conducted in compliance with data minimization principles, ensuring that only the necessary information is processed.

3. Data Planning and Splits

1. **Time-Based Data Splitting**:
    - We will use review timestamps to split the dataset into training, testing, and validation sets. This ensures that the model is always tested and validated on future reviews.
    - By splitting data based on time, we can observe how well the model generalizes to future sentiments and identify if model performance deteriorates as new trends or product versions emerge.

2. **Training, Testing, and Validation Ratios**:
    - The dataset will be split based on time periods to reflect a realistic scenario:
        - Training Data: This includes reviews from the earliest available period until a specified date (a date sometime in past than the latest date in the dataset) (70% of the dataset). This portion is used to teach the model the initial patterns in sentiments.
        - Testing Data: This includes reviews from a subsequent period (15%). The model is tested on data that follows the training period to understand its performance as sentiments evolve.
        - Validation Data: This includes reviews from the most recent period (15%). The validation dataset is used to observe how the model performs with the latest customer feedback and helps detect signs of data drift.
3. **Simulating Sentiment Changes During Inference**:
    - During inference, the model will be evaluated on the most recent reviews i.e. data from when the validation data ends, which are always newer than those used for training. This allows the system to simulate real-world conditions where consumer feedback changes as time progresses.

    - Sentiments for specific products may shift due to product updates, seasonal trends, or external influences. By continuously feeding recent reviews into the model, we can monitor how these changes impact overall product sentiment over time.

4. GitHub Repository

Link: https://github.com/MLOps-Group-3/Amazon-Reviews-Sentiment-Analysis

Folder Structure:

- **README.md** - The main readme file providing an overview of the project, installation instructions, and usage guidelines.
- **LICENSE -** The license file outlining the terms and conditions for using the repository
- **.github/** - GitHub-related files (e.g., workflows, templates, GitHub actions)
- **data/** - Raw and unprocessed data files
- **models/** - Saved machine learning models and related metadata
- **notebooks/** - Jupyter/Google Colab notebooks for exploration and prototyping/modelling
- **scripts/** - Deployment and monitoring scripts
- **src/** - Source code for data processing, modelling, and evaluation, DAGs for Airflow
- **tests/** - For unit tests of scripts and methods
- **milestones/scoping/** - Files logging milestones

5. Project Scope

5.1 Problems

This project addresses several challenges businesses face (Amazon in this use case) when analyzing customer sentiment from user reviews:

- **Lack of Automation**: Sentiment analysis and insights generation from customer reviews often involve manual, time-consuming, and error-prone processes.
- **Data Scalability**: Handling, processing, and analyzing a growing volume of customer reviews at scale is difficult with traditional methods.
- **Sentiment Classification**: Current methods may not accurately categorize reviews into positive, negative, or neutral sentiments, impacting the quality of business decisions.
- **Monitoring and Model Retraining**: Machine learning models can degrade over time, and without automated monitoring and retraining, model performance can decline unnoticed.
- **Executive Insights**: Executives require detailed, real-time insights into customer satisfaction across various product categories, but existing dashboards lack the interactivity and detail needed for effective sentiment analysis.
- **Handling Edge Cases and Anomalies**: Reviews can be ambiguous, sarcastic, or irrelevant, making classification challenging, especially when dealing with complex or domain-specific language.

5.2 Current Solutions

Several existing solutions attempt to address these issues but have limitations:

- **Manual Review Analysis**: Manually analyzing reviews doesn't scale for large datasets and can introduce biases.
- **Basic Sentiment Analysis Models**: Off-the-shelf models are available but often lack the nuance needed for specific domains like Amazon, leading to lower accuracy.
- **Traditional Dashboards**: Standard BI dashboards provide aggregated metrics but often miss specific insights related to customer sentiment by category.
- **Ad-hoc Monitoring**: Without automation, pipelines rely on manual monitoring, leading to delays in retraining models when performance drops.
- **Rigid Data Pipelines**: Many pipelines lack flexibility, making it difficult to adapt or improve without causing disruptions.

5.3 Proposed Solutions

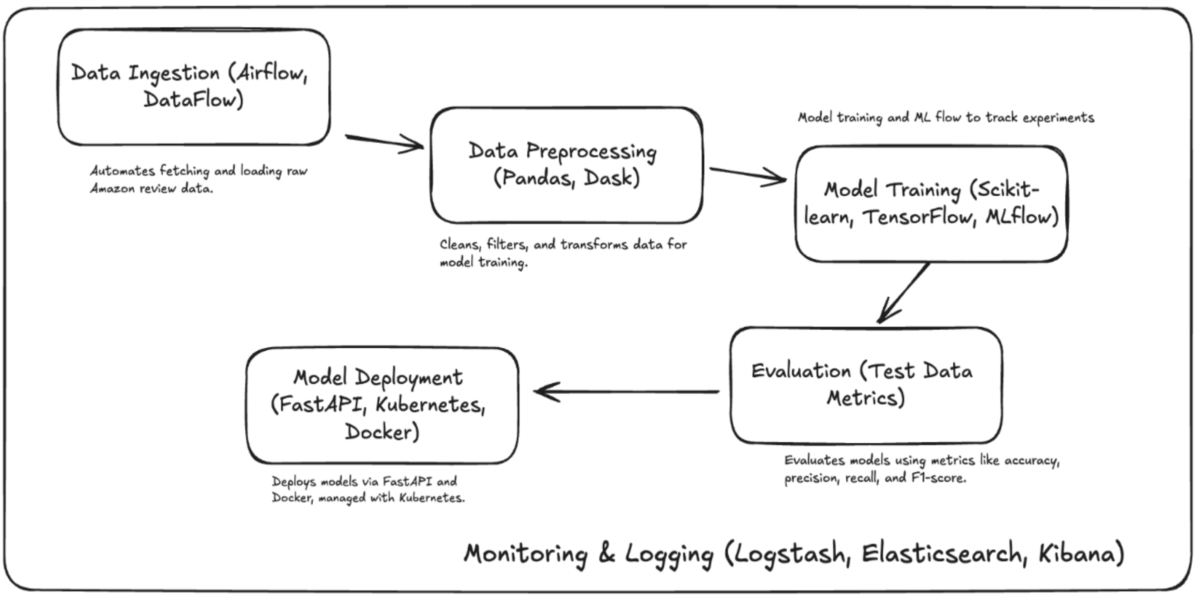
The project will implement a robust MLOps pipeline that automates, scales, and customizes sentiment analysis with the following innovations:

- **Fully Automated MLOps Pipeline**: Using Airflow, Docker, Kubernetes, and GCP, the project will automate data ingestion, pre-processing, model training, evaluation, and deployment, ensuring scalability and reliability.
- **Scalable Cloud Infrastructure**: Google Cloud and Kubernetes will provide scalability to handle the growing volume of Amazon reviews, ensuring seamless performance.
- **Custom Sentiment Classification**: Custom sentiment classification models will be built using TensorFlow, designed to classify reviews as positive, negative, or neutral with greater accuracy by addressing domain-specific language (terms and phrases unique to Amazon's product reviews) and handling edge cases (ambiguous reviews, sarcasm, and spam content).

- **MLflow for Experiment Tracking**: MLFlow will be used to track experiments, allowing for efficient comparison of different model configurations, hyperparameters, and performance metrics over time.
- **CI/CD with GitHub Actions**: Continuous integration and deployment will ensure that code changes, model updates, and configuration adjustments are automatically tested, validated, and deployed without manual intervention.
- **Comprehensive Monitoring with Kibana**: Real-time monitoring will be implemented using Kibana to track data quality, model performance, and detect anomalies, triggering retraining workflows as necessary.
- **Modular Pipeline Design**: Each component of the pipeline—data ingestion, preprocessing, modelling, deployment—will be designed to be modular and replaceable, enabling flexibility and easy integration of new components without breaking the pipeline.
- **Interactive Executive Dashboard**: The final product will include a dashboard that enables executives to drill down by product category, track sentiment over time, and interactively summarize reviews using a Retrieval-Augmented Generation (RAG) pipeline, providing actionable insights at both high-level and granular levels.

6. Current Approach Flowchart and Bottleneck Detection

6.1 Current Approach Flowchart



6.2 Bottlenecks and Improvements

- Slow ingestion due to large data volumes (solution: use batch processing and increase task concurrency in Airflow).
- Pandas struggles with large datasets (solution: use Dask for distributed data processing).
- Long training times on large datasets (solution: use distributed training on GCP (TPU/TF Distributed Strategy)).
- Limited test data affects generalization (solution: use cross-validation and regularly refresh test sets).
- High latency during deployment (solution: use Kubernetes rolling updates and autoscaling).

- Delayed detection of model drift or pipeline issues (solution: set up real-time alerts for drift and performance drops).

6.3 Pipeline Bottlenecks

- Model misclassifies ambiguous reviews (solution: log and address edge cases).
- Pipeline failure at any stage can stop processing (solution: use retry policies and circuit breakers to handle failures).

7. Metrics, Objectives, and Business Goals

7.1 Key Metrics

- **Accuracy**: Overall correctness in classifying reviews as positive, negative, or neutral.
- **Precision**: Correctly identifies positive/negative reviews without misclassifying neutral ones.
- **Recall**: Captures all relevant reviews in each sentiment category.
- **F1 Score**: Balances precision and recall, crucial for class imbalance.
- **True Negative Rate (Specificity)**: Accurately identifies negative reviews, critical for addressing customer dissatisfaction.

7.2 Project Objectives

- **Accurate Sentiment Classification**: Improve review classification for better organization and insights.
- **Product Insights**: Provide actionable insights for Amazon (and similar businesses) and sellers to enhance product offerings.
- **Customer Experience**: Help businesses address customer issues through accurate sentiment tracking.
- **Data-Driven Decisions**: Use sentiment analysis to guide product improvements and business strategies.
- **Increase Sales and Reduce Returns**: Identifying and addressing negative feedback will drive better customer retention and lower return rates.

7.3 Operational Efficiency Objectives

- **Process Automation**: Automate the entire MLOps pipeline (data ingestion, preprocessing, training, deployment) to reduce manual intervention and ensure scalability.
- **Pipeline Health Monitoring**: Ensure real-time monitoring using metrics like:
    - **Latency**: The time taken for the model to process and classify reviews, aiming for low response times under load.
    - **Throughput**: The number of reviews processed per second, ensuring the system handles large datasets efficiently.
    - **Error Rates**: Track errors or failed model predictions, ensuring consistent performance.
    - **Resource Utilization**: Monitor CPU, GPU, and memory usage to optimize cost and performance.

- **Retraining Triggers**: Set thresholds for automatic model retraining when performance drops below acceptable levels (e.g., F1 score or accuracy falls below a certain point).

7.4 Aligning Metrics with Business Goals

- **True Negative Rate**: Improves customer satisfaction by identifying and resolving negative reviews.
- **F1 Score**: Provides balanced sentiment classification for actionable insights.
- **Pipeline Efficiency**: High throughput and low latency ensure timely feedback processing, supporting real-time decision-making.
- **Dashboard Insights**: Offer real-time sentiment trends, enabling timely business interventions for improved performance. Additionally, it will feature interactive review summaries that allow users to extract key themes, identify customer pain points, and gain deeper insights into feedback, driving actionable business strategies.

8. Failure Analysis

8.1 Data Ingestion & Pre-processing Risks

- Inaccurate or Unreliable Data
  - **Risk**: Noisy, incomplete, or biased data may degrade model accuracy.
  - **Mitigation**: Implement data validation checks, perform regular retraining, and apply data augmentation to improve quality.
- Data Drift
  - **Risk**: Changes in customer behaviour could cause performance declines due to shifts in data patterns.
  - **Mitigation**: Monitor for drift and set up scheduled model retraining when performance drops.
- Sentiment Imbalance
  - **Risk**: Imbalanced sentiment distribution can skew predictions, resulting in biased outputs.
  - **Mitigation**: Use techniques like SMOTE to balance the data and apply weighted loss functions in training to account for class imbalances.

8.2 Model Training Risks

- Poor Model Performance
  - **Risk**: Low accuracy, overfitting, or underfitting may result in a model that doesn't generalize well to new data.
  - **Mitigation**: Use cross-validation, hyperparameter tuning, and model explain ability.
- Incorrect Sentiment Classification
  - **Risk**: Misclassifications could mislead business decisions.
  - **Mitigation**: Implement confidence thresholds for uncertain predictions, triggering manual review for low-confidence cases.
- Model Bias
  - **Risk**: The model may reflect biases present in the training data, leading to skewed predictions.
  - **Mitigation**: Conduct regular bias audits and ensure the dataset is diverse and representative.

8.3 Infrastructure Risks

- Pipeline Failures

- **Risk**: Task failures in systems like Airflow or DataFlow can disrupt the pipeline.
- **Mitigation**: Implement retry logic, checkpointing, and set up monitoring to ensure smooth task execution.

- Latency
  - **Risk**: High latency during data processing or inference can slow down the entire pipeline.
  - **Mitigation**: Optimize resource allocation and monitor performance consistently.
- Deployment Failures
  - **Risk**: Issues with Docker or Kubernetes may prevent the model from deploying successfully.
  - **Mitigation**: Follow best practices for containerization, use auto-scaling, and implement load balancing.
- API Downtime
  - **Risk**: The FastAPI service could experience downtime during high traffic, affecting model predictions.
  - **Mitigation**: Leverage auto-scaling and blue-green deployment strategies to maintain availability.

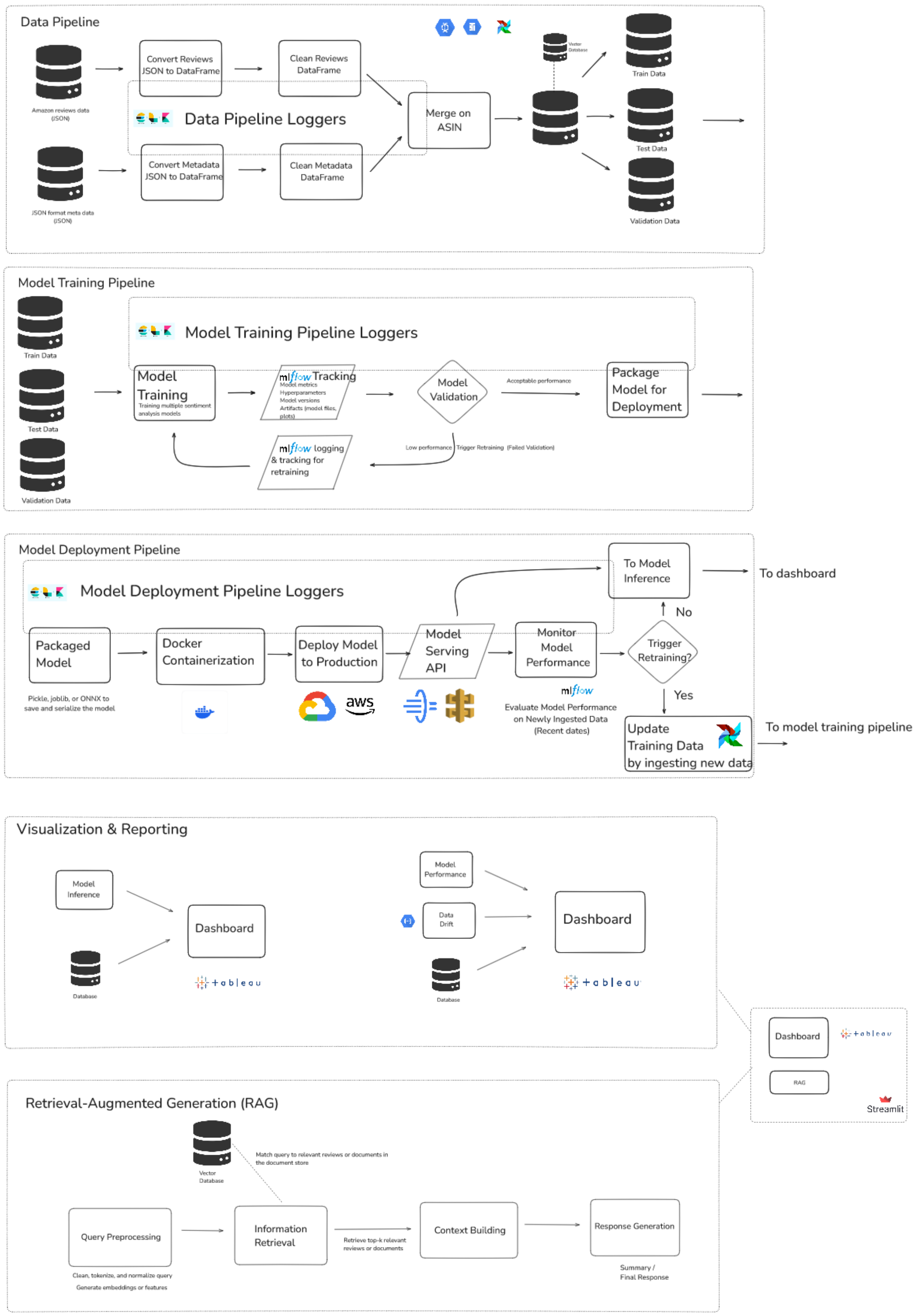
8.4 Post-Deployment & Monitoring Risks

- Model Degradation
  - **Risk**: Over time, the model's performance may degrade due to data drift or changes in customer sentiment.
  - **Mitigation**: Implement continuous monitoring and set up an automated retraining pipeline triggered by performance declines.
- Anomaly Detection Failures
  - **Risk**: Unusual sentiment spikes may go unnoticed, leading to faulty business insights.
  - **Mitigation**: Build anomaly detection algorithms to flag outliers and trigger automated or manual review.

8.5 CI/CD Pipeline Failures

- **Risk**: Breakdowns in the CI/CD pipeline due to dependency issues, merge conflicts, or configuration errors may halt deployment.
- **Mitigation**: Employ automated testing with GitHub Actions, ensure rollback mechanisms, and enforce code reviews to prevent errors before production.

9. Deployment Infrastructure



1. **Data Pipeline**:
   - **Storage**: Use **Google Cloud Storage (GCS)** for storing data.
   - **ETL Automation**: **Google Cloud Functions** for cleaning and merging data.
   - **Workflow Management**: **Apache Airflow** on **Google Compute Engine (GCE)** for ETL orchestration using Docker.
2. **Model Training Pipeline**:
   - **Compute Instances**: Use **GCE** for training.
   - **Tracking**: Deploy **MLflow** on GCE using Docker for experiment tracking.
   - **Data Preparation**: Use **Apache Spark** on GCE for distributed data processing.
3. **Model Deployment Pipeline**:
   - **Containerization**: Use **Docker** and store images in **Google Artifact Registry**.
   - **Deployment**: Deploy model endpoints as REST APIs using **FastAPI** or **Flask** on GCE with **Google Cloud Load Balancer**.
   - **Monitoring**: Use **MLflow** on GCE to monitor model and infrastructure metrics.
4. **Retraining and Improvement**:
   - **Retraining**: Automate retraining via **Apache Airflow** when performance declines.
   - **Data Ingestion**: Continuously ingest new data into the training pipeline for model updates.
5. **Visualization & Reporting**:
   - **Model Inference Metrics**:
     - Data from model inference is stored in a **database**.
     - Utilize **Tableau** for creating dashboards to visualize inference results, including performance metrics and predictions.
   - **Model Performance & Data Drift**:
     - **Google Cloud Monitoring** (**BigQuery** for storing drift metrics).
     - Visualize model performance and data drift metrics in **Tableau/Power BI** to monitor changes in input data characteristics, model accuracy, and overall health.
6. **RAG:**
   - **Information Retrieval**:
     - Utilize a **vector database** such as **Google Vertex AI Matching Engine, Pinecone, Weaviate** to store indexed document of product reviews as embeddings for efficient similarity search.
   - **Response Generation**:
     - Generate a response using a language model such as **Llama or similar open-source models** that provides answers, summaries, or insights based on the retrieved context.
   - **Deployment**:
     - Use **Streamlit** to develop an interactive user interface for querying and interacting with the RAG system.

10. Monitoring Plan

Statistics and Metadata of the reviews received for Sentiment Analysis:

- **Purpose:** To ensure the quality, integrity, and appropriateness of each customer review used for sentiment analysis. This involves collecting and analyzing metadata such as review length, submission date, and content quality (e.g., detecting spam, irrelevant or ambiguous reviews).
- **Benefits:** Monitoring these aspects helps detect any anomalies or deviations from expected patterns, such as reviews with too few words or overly complex language, which could skew sentiment analysis results or impact model accuracy.

Metrics Representing Health of the Model:

- **Purpose:** To continuously assess the model's performance and operational status, ensuring its ability to classify sentiments (positive, neutral, negative) accurately. Key performance indicators will include precision, recall, F1-score, and accuracy.
- **Benefits:** Regularly evaluating these metrics ensures that the sentiment classification model maintains its accuracy over time. Significant shifts in performance metrics will trigger retraining, ensuring the model adapts to changing review content.

Monitoring for Dataset Shift and Dataset Skew:

Dataset Shift: Occurs when the statistical properties of incoming reviews differ significantly from the training data. This can degrade the performance of the sentiment classification model.
Dataset Skew: Happens when reviews used by the model do not accurately represent the diversity or structure of the larger customer base, potentially leading to biased sentiment predictions.

- **Purpose:** To detect and correct dataset shifts or skews, ensuring the model adapts to evolving customer feedback patterns.
- **Benefits:** Early detection of these issues allows for prompt retraining or adjustment of the model and data pipeline, ensuring that the model continues to generalize well to incoming data.

Real-Time Monitoring of Anomalies:

- **Purpose:** To identify and handle edge cases such as sarcastic, ambiguous, or spam reviews in real-time. Kibana will be used to visualize and monitor data quality and model performance, including detecting sudden spikes in anomalous reviews or unexpected patterns in sentiment classification.
- **Benefits:** Real-time anomaly detection helps prevent the model from producing inaccurate predictions due to edge cases and ensures timely retraining or adjustment workflows.

11. Success and Acceptance Criteria

Automated and Scalable Pipeline:

- The pipeline must automatically handle data ingestion, pre-processing, model training, evaluation, and deployment, ensuring that the system scales efficiently as the volume of data (e.g., customer reviews) increases.
- **Success Threshold:** Smooth operation under increasing data loads with minimal manual intervention.

Handling of Edge Cases and Ambiguous Reviews:

- The sentiment classification model must accurately handle complex and ambiguous reviews, including sarcastic or irrelevant content, and provide appropriate classifications or flags for further review.
- **Success Threshold:** A high classification accuracy rate for edge cases, ensuring that misclassifications are minimized.

Comprehensive Dashboard for Executive Insights:

- The dashboard should allow business users to explore sentiment data in detail, including the ability to drill down by product category, track sentiment trends over time, and derive actionable insights for decision-making.
- **Success Threshold:** High engagement and positive feedback from business users, indicating the dashboard's effectiveness in providing actionable insights.

End User Trust and Adoption:

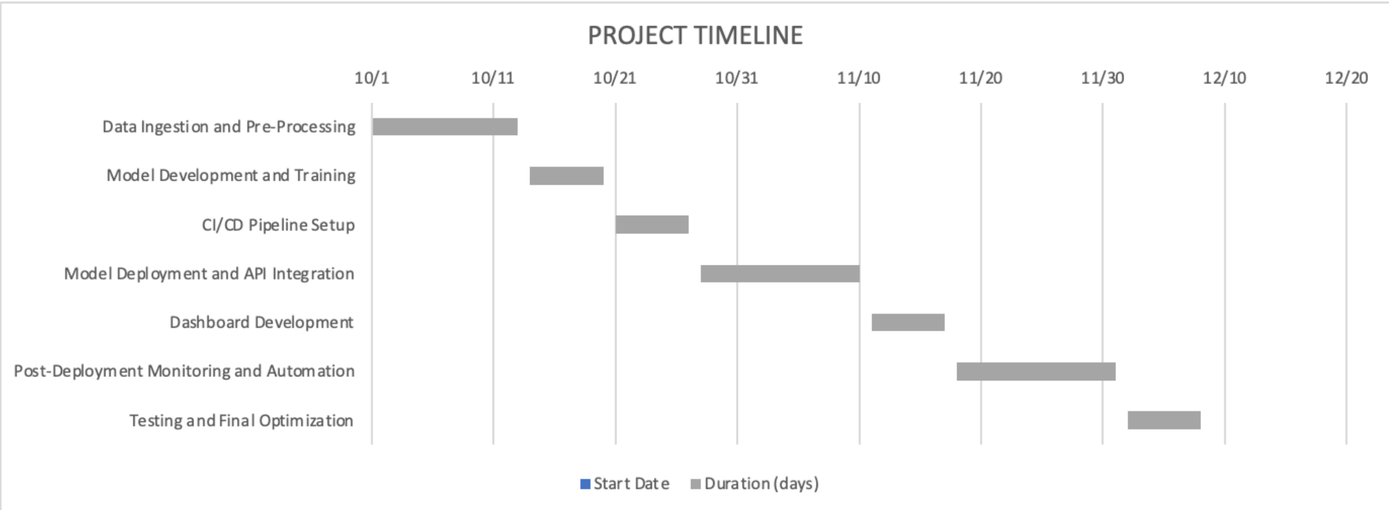
- **Definition:** Evaluate the level of trust users (executives, analysts) have in the insights generated by the sentiment analysis model. This will also include their willingness to rely on it for strategic decisions such as product improvements, marketing campaigns, and customer service optimizations.
- **Measurement:** Assess trust through both direct feedback and indirect metrics, such as the frequency of model usage in decision-making processes or reliance on the interactive dashboard for business insights.

Executive Insights and Decision-Making Improvement:

- **Definition:** Ensure that the interactive executive dashboard provides valuable, actionable insights into customer sentiment across product categories. This will involve measuring whether the system improves decision-making speed and accuracy.
- **Measurement:** Track user interactions with the dashboard, including drill-down frequency, sentiment trend analysis, and usage of the Retrieval-Augmented Generation (RAG) feature for summarizing reviews.

12. Timeline Planning

Data Ingestion and Pre-Processing	10/01 - 10/13 (2 Weeks)
Model Development and Training	10/14 - 10/20 (1 Week)
CI/CD Pipeline Setup	10/21 - 10/27 (1 Week)
Model Deployment and API Integration	10/28 - 11/10 (2 Weeks)
Dashboard Development	11/11 - 11/17 (1 Week)
Post-Deployment Monitoring and Automation	11/18 - 12/01 (2 Weeks)
Testing and Final Optimization	12/02 - 12/08 (1 Week)



13. Additional Information

- **Task Management and Sprint Planning:** Jira will be utilized to effectively manage tasks and facilitate sprint planning, ensuring teamwork and effective coordination among team members to meet project milestones efficiently.

- **Team Collaboration:** Collaborative efforts will be conducted on GitHub, emphasizing a feature-based development approach to streamline contributions and maintain code quality.
- **Documentation:** Comprehensive documentation for both the pipeline and the model will be maintained alongside the codebase, ensuring that all aspects of the project are well-documented for future reference and ease of understanding.
- **Citation:**
  Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption
  Jérémie Rappaz, Julian McAuley and Karl Aberer RecSys, 2021