Sentiment Analysis Project Report

1. Introduction

a. Project Overview:

This project focused on sentiment analysis of X (formerly Twitter) customer support interactions. The goal was to classify tweets into sentiment categories (positive, negative, neutral), aiding businesses in understanding customer sentiments. The analysis is crucial for enhancing customer support strategies and improving overall customer experience.

b. Business Context:

In the digital era, customer sentiments, expressed through social media, significantly impact business strategies. Analyzing sentiments from customer interactions on platforms like X provides valuable insights into customer satisfaction and brand perception.

c. Problem Statement:

The primary business challenge is managing customer satisfaction efficiently at scale for popular brands. By applying sentiment analysis to X data, we can predict the sentiment of customer interactions and identify areas for improvement in customer support which would directly impact their customer retention metrics.

2. Data Acquisition and Wrangling

One of the challenges of our sentiment analysis project was the extensive dataset encompassing over half a gigabyte of customer interaction records from X (formerly Twitter). To efficiently manage these data, we leveraged PostgreSQL, a robust and scalable relational database system. This strategic decision was pivotal in enabling us to efficiently handle, sample, and reduce the dataset's load, thereby streamlining the initial stages of the project.

ETL Process and Data Handling:

We meticulously designed and implemented a comprehensive ETL (Extract, Transform, Load) process, targeting effective resource management and data integrity. The process involved several crucial steps:

- Database Creation and Schema Development: Initiated by creating a dedicated PostgreSQL database, named “customer_support”. We utilized the psycopg2 library to develop a detailed database schema, ensuring structured and organized data storage.

- Data Sampling and Load Management: To work around computational bottlenecks, we employed strategic data sampling techniques. This involved loading a carefully selected subset of the data, optimized for initial exploratory and model development purposes.

- Advanced SQL Queries for Data Processing: Leveraging SQL's powerful querying capabilities, we efficiently filtered and preprocessed the records. This step was instrumental in refining the dataset for subsequent analysis.

- Data Cleaning and Transformation: Our data wrangling efforts were focused on maintaining the integrity and consistency of the dataset. We filtered the data to isolate initial customer queries, crucial for accurate sentiment analysis. The text data underwent a thorough standardization process, involving the removal of duplications and standardizing date formats. Utilizing regular expressions, we meticulously cleaned the text data, removing extraneous elements like special characters, mentions, and emojis. These measures were critical in elevating data quality, paving the way for nuanced analysis.

Integration with Jupyter Notebooks:

In conjunction with PostgreSQL, we utilized Jupyter notebooks for data exploration and preliminary analysis. These notebooks served as an interactive platform for data visualization, allowing us to gain early insights into the dataset's characteristics and potential challenges. The initial findings from these explorations informed our subsequent data processing and modeling strategies.

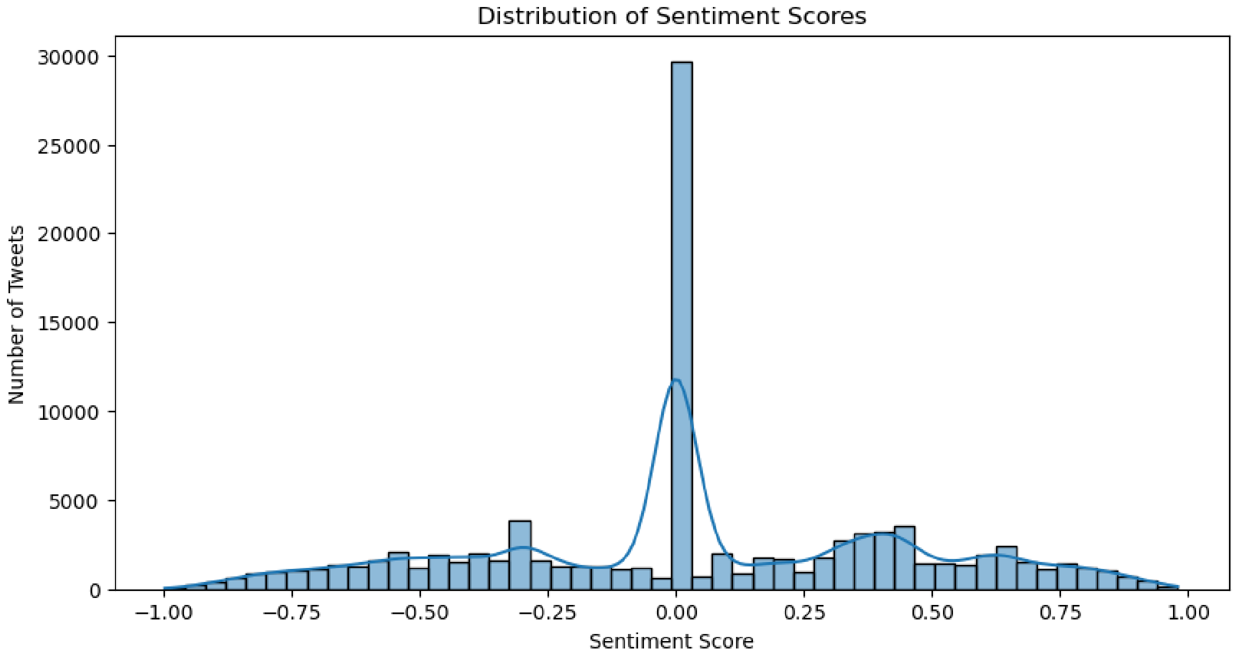
This enhanced approach to data acquisition and wrangling laid a solid foundation for our sentiment analysis project, ensuring that the data was not only robust and comprehensive but also primed for extracting meaningful insights through advanced analytical techniques.

3. Exploratory Data Analysis (EDA)

Starting the next stage, exploratory data analysis, we deployed a suite of visual and statistical methods to unravel the multifaceted narrative of customer sentiment as captured in social media interactions. Our EDA was anchored by the Sentiment Intensity Analyzer from the Natural Language Toolkit (NLTK), which meticulously scored each tweet, setting the stage for sentiment categorization.

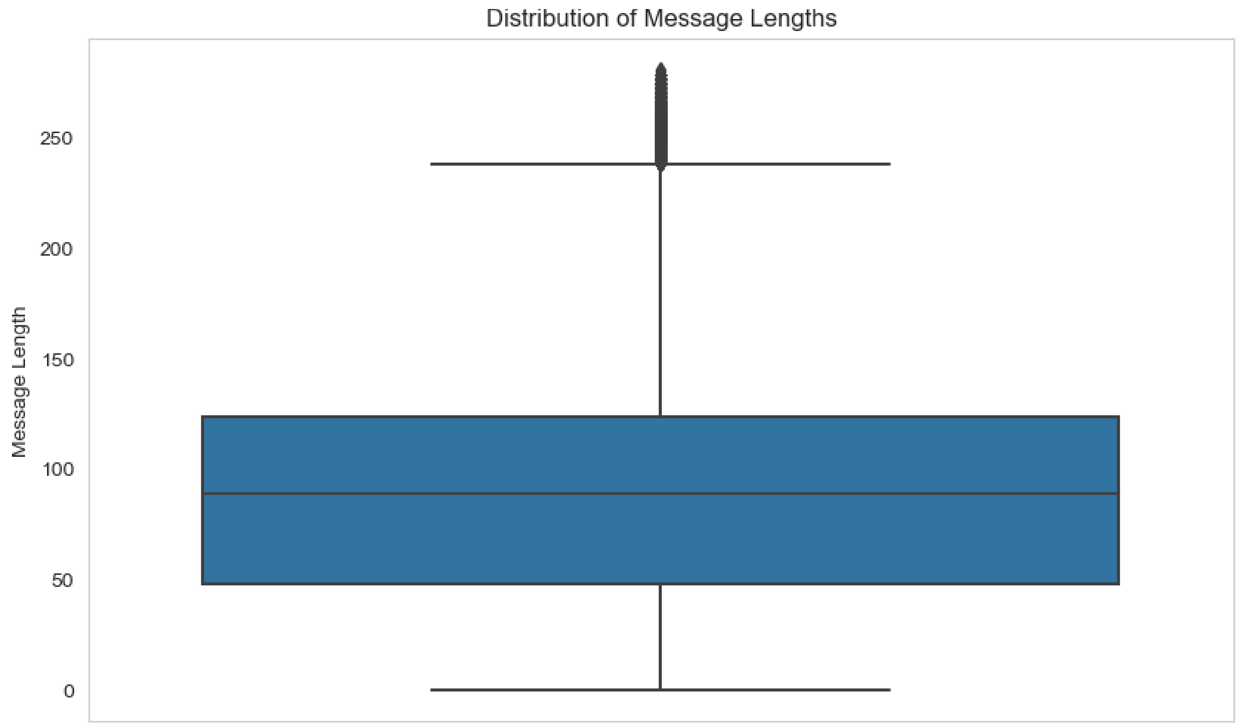
Sentiment Score Distribution Analysis:

The sentiment score distribution, as illustrated by the histogram, depicted a pronounced central tendency, with a significant aggregation of neutral sentiments. This neutral dominance highlighted the balanced nature of customer discourse on the platform.



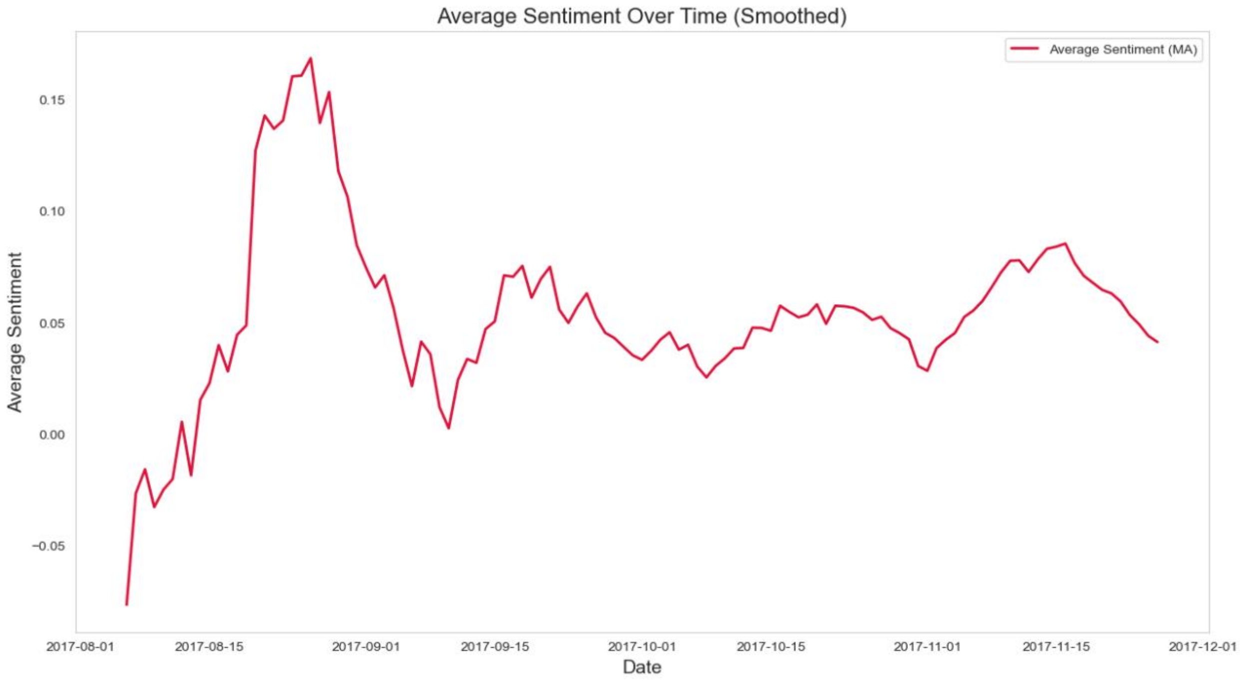
Message Length Dissection:

The box plot analysis of message lengths offered an intriguing glimpse into communication patterns, revealing a prevalent brevity in customer tweets. The compact message structure was reflective of the concise communication style favored in social media exchanges.



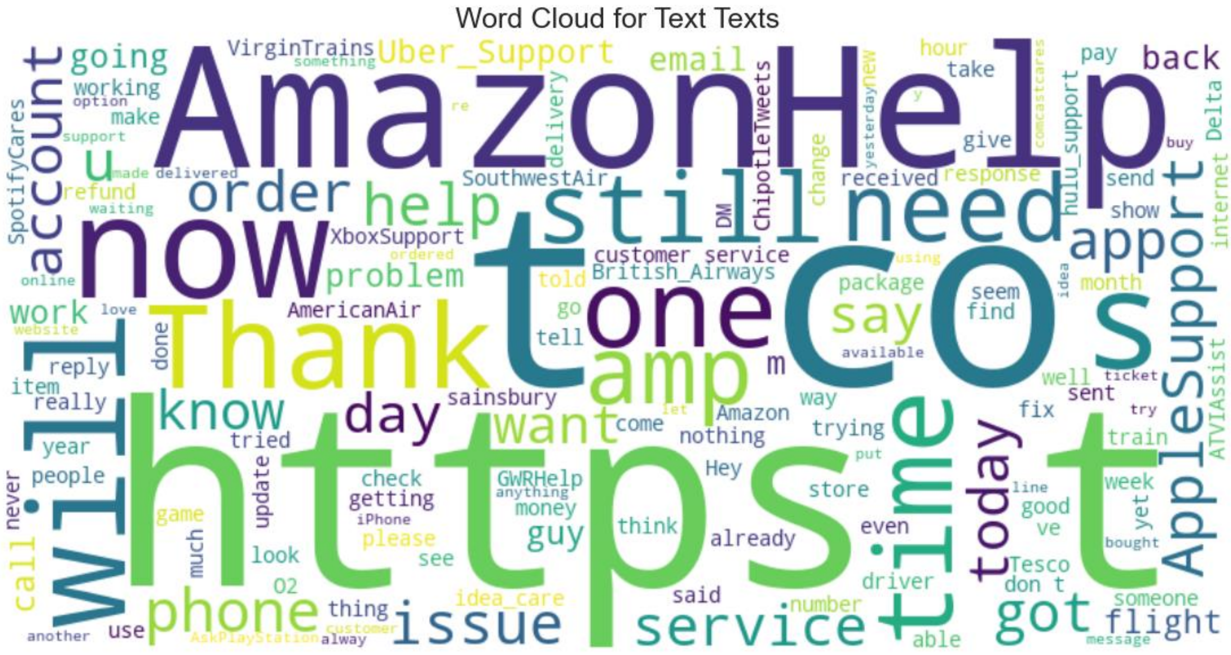
Temporal Sentiment Fluctuations:

A line graph mapping average sentiment over time surfaced underlying temporal trends, signifying how customer emotions flowed. We noted an upward trend in positive sentiments, potentially indicative of improving customer experiences or shifting social discourse.



Linguistic Landscape via Word Cloud:

A word cloud brought up the predominant themes within the corpus. Central to customer queries were concerns around service quality and product inquiries. The visualization provided a stark representation of the customer support lexicon, with terms like 'help', 'service', and 'issue' prominently featured.



Word Cloud for Text Texts

Key Insights from EDA:

- Service-related queries and product inquiries dominated the landscape, reflecting the primary concerns of customers.
- An overarching positive sentiment trend suggested gradual improvements in customer experiences or, perhaps, a shift in the public discourse.
- Peaks in tweet volume correlated with specific events or campaigns, pointing to the reactive nature of customer engagement.
- The complexity of customer inquiries, inferred from message length, suggested a preference for succinctness in service-related communication.

These visualizations proved instrumental in peeling back the layers of customer sentiment. They provided an essential foundation for our subsequent modeling efforts, where these insights would inform the development of predictive analytics.

4. Data Preprocessing and Feature Engineering

The preprocessing phase was a crucial step between the raw data and the analytical modeling. Leveraging the Python stack, we employed the Term Frequency-Inverse

Document Frequency (TF-IDF) vectorization to transform the textual data into a quantifiable format, thus capturing the relative significance of terms within the dataset.

Advanced Natural Language Processing (NLP) techniques such as lemmatization and stop-word removal, facilitated by the spaCy library, were diligently applied to refine our dataset. This process not only streamlined the textual content but also enriched the data quality, essential for robust sentiment analysis.

In the feature engineering realm, we meticulously crafted a set of predictive attributes designed to encapsulate the nuances of the data:

- Text Length and Complexity: These features were instrumental in gauging the depth and detail within each customer interaction.
- Entity Recognition: Using Named Entity Recognition (NER), we pinpointed product and brand mentions, distilling these into discrete features.
- Linguistic Attributes: The grammatical structure was illuminated through Part-of-Speech (POS) tagging, which informed us of the sentence compositions prevalent in customer communications.

A key development was the formation of a 'Derived Sentiment' column, intricately constructed from the EDA insights. This target variable was crafted with a fine-tuned classification threshold, harmonious with the intricate sentiment dimensions observed in the data.

Synthesizing the TF-IDF vectors with the engineered features into a unified sparse matrix, we took careful measures to split the data, effectively safeguarding against data leakage. This strategic partitioning prepared the dataset for the modeling stage, ensuring a clean, isolated testbed for our predictive algorithms.

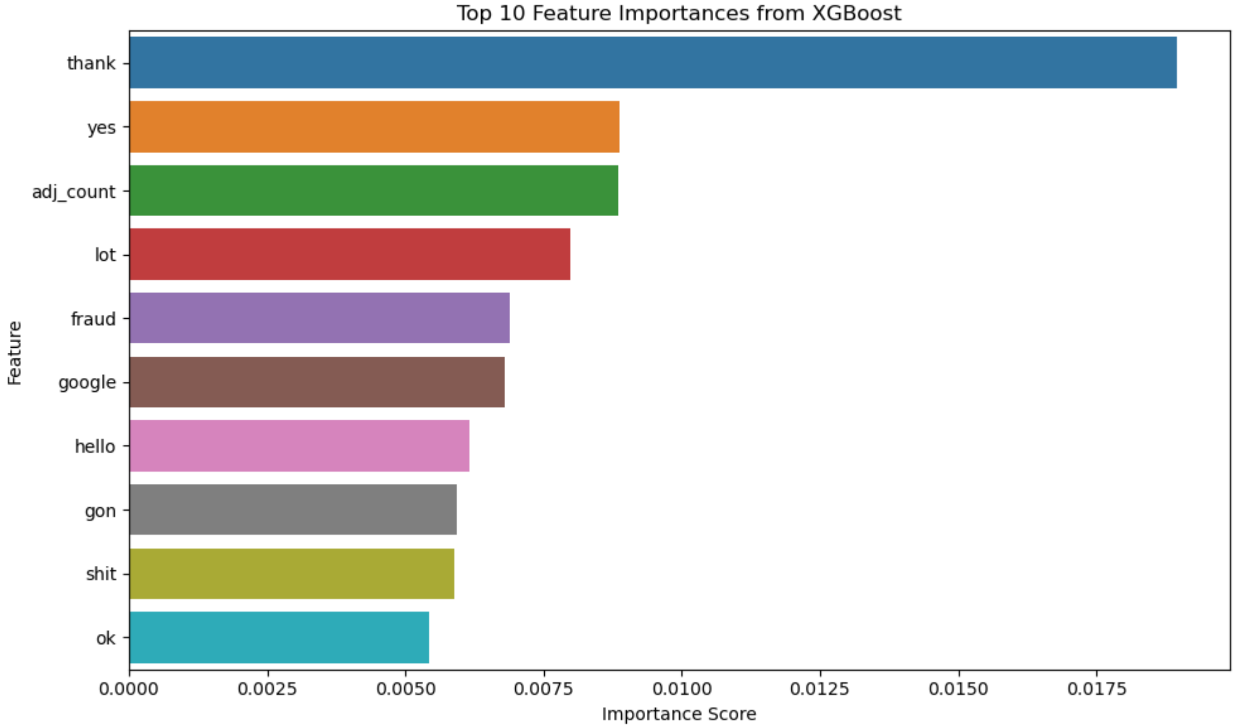
5. Model Building and Evaluation

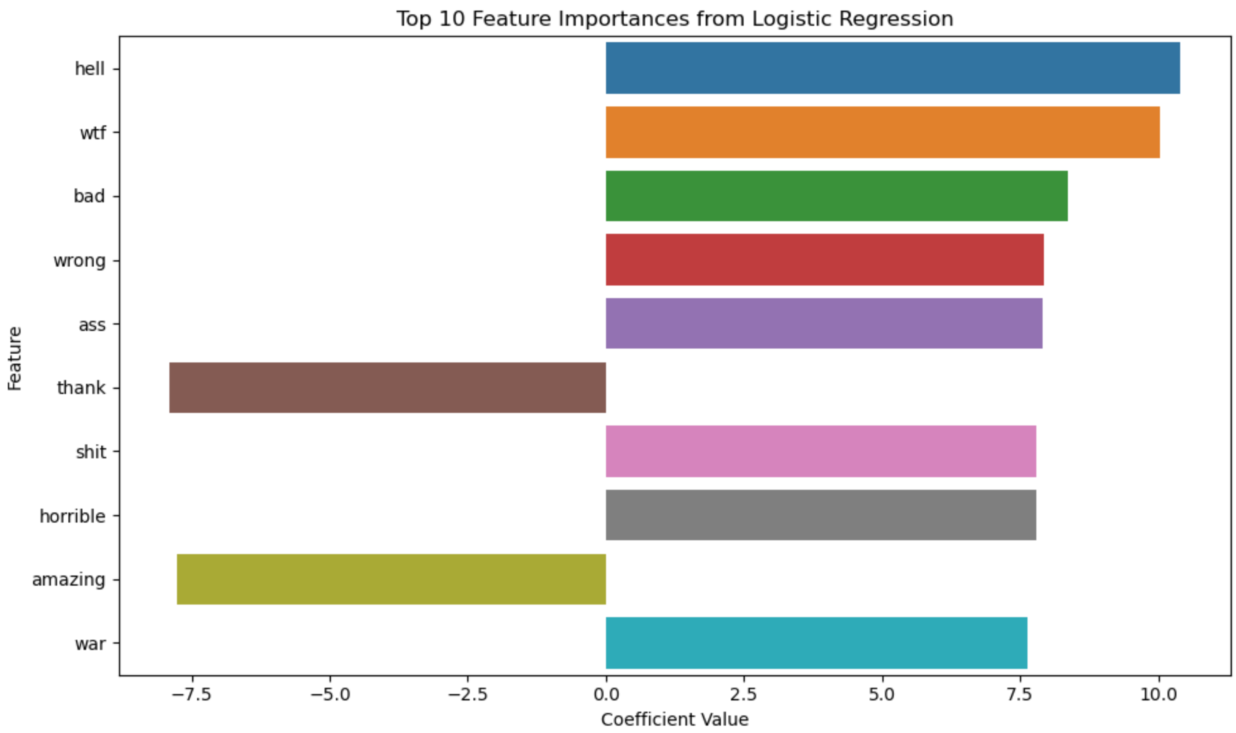
In this pivotal phase, we rigorously constructed and assessed various predictive models, utilizing a comprehensive suite of evaluation metrics including accuracy, precision, recall, and the F1 score to holistically appraise model performance. Through this multifaceted evaluation, the Logistic Regression model emerged as a strong contender, providing a well-balanced synthesis of predictive power and computational efficiency — vital for the real-time demands of sentiment analysis within customer support frameworks.

- Model Selection and Rationale: Our exploratory journey through the modeling landscape encompassed Logistic Regression, Random Forest, and XGBoost.

Each model was selected for its unique approach to classification — Logistic Regression for its simplicity and interpretability, Random Forest for its decision-driven logic, and XGBoost for its robust, ensemble-based methodology. This selection was grounded in a strategy to cover a spectrum of model complexities, ensuring a thorough probing of the dataset's characteristics.

- Hyperparameter Tuning and Optimization: To fine-tune the models, we undertook a diligent hyperparameter optimization process. Informed by the feature importance analyses from both XGBoost and Logistic Regression which highlighted key predictive features such as 'thank', ‘’bad’ and affective terms like 'fraud' and 'amazing', we steered our tuning strategies to amplify model responsiveness to these critical indicators.





The XGBoost Classifier particularly stood out, showcasing exemplary performance across all metrics, underpinned by its advanced tree-based approach. Notably, the iterative nature of model refinement was emphasized, with ongoing efforts to enhance performance by focusing on feature selection and leveraging insights from the feature importance plots to inform this process.

The insights drawn from the feature importance analysis were instrumental in refining our models, offering a data-driven avenue to bolster the interpretative capabilities of our predictive models. This meticulous process of iterative enhancement and validation exemplifies the dynamic nature of machine learning workflows — continually evolving and adapting to new findings and improved techniques.

6. Conclusions & Recommendations

The application of the Data Science pipeline has been an important part along with the structured and iterative nature of analytical modeling to the success of the project. The sentiment analysis conducted not only unveiled the emotional underpinnings of customer interactions with support agents but also illustrated the triggers for varied sentiments. The performance of the XGBoost model, while a robust baseline, invites further enhancement to refine its predictive prowess.

Recommendations:

- Utilizing insights from positive sentiment analysis to inform and tailor marketing strategies.
- Addressing prevalent negative sentiments to enhance customer service quality.
- Implementing ongoing monitoring of social media for real-time sentiment analysis to stay ahead of customer mood shifts.
- Investigating the potential for automated customer support via chatbots, which could revolutionize response times and personalization.

7. Next Steps – Deployment

In alignment with modern methodologies for ML model deployment, the forthcoming stage of our endeavor will facilitate the evolution of machine learning models from conceptual prototypes to functional ML applications. Labarta Bajo (2023) underscores the significance of implementing a methodical deployment strategy, which is fundamental to the scalability of ML applications. This involves the creation of distinct feature, model training, and batch-prediction pipelines.

Next steps:

- Refining the Feature Pipeline: Separating feature engineering and preprocessing into a dedicated pipeline, ensuring these processes are consistently applied to new data.
- Optimizing the Model Training Pipeline: Incorporating the latest features from the Feature Store and continuously updating the model registry with the refined models.
- Establishing a Batch-Prediction Pipeline: Systematizing the process of loading the model, applying it to new batches of features, and persisting predictions, thereby ensuring real-time relevance and responsiveness.
- Deployment Automation: Using serverless technologies and workflow automation tools to schedule and execute these pipelines, ensuring scalability and efficiency.

8. References

- Labarta Bajo, P. (2023). 3 Steps to transform an ML prototype into a real-world ML app. Real World Machine Learning. Retrieved from https://www.realworldml.xyz/blog/3-steps-to-transform-an-ml-prototype-into-a-real-world-ml-app.

- Thoughtvector. (n.d.). Customer Support on Twitter [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter.