



BT4301 Final Report

Half-time Superbowl Advertisement Online Sentiment Analysis

Group 11

Member	Matriculation Number
Lee Jie Long, Bryan	A0233537E
Foo Yan Rong	A0233628A
Jonathan Lim Yu Shun	A0234165J
Ryu Kairin Anaqi Suzuki	A0233269B
Abigail Lim Xiao Jun	A0240396A

Table of Contents

1 Executive Summary	4
2 Problem Statement	4
2.1 Problem Definition	4
2.2 Business Context	4
2.3 Business Value	5
3 Analytics Solution Statement	5
4 Agile Analytics Project Management With SCRUM	6
4.1 Overall Project Planning	6
4.2 SCRUM Sprint 1 and 2 Overview	6
4.3 Kanban Board	7
5 DataOps and MLOps Toolchain	9
5.1 Implementation Steps	11
6 DataOps	11
6.1 Dataset Construction	11
6.2 Data Collection and Ingestion	13
6.3 Data Storage	13
6.3.1 Database for Raw Data	14
6.3.2 Data Warehouse	14
6.4 Data Cleaning and Transformation	14
6.4.1 Data Cleaning	14
6.4.2 Transformation	15
6.5 Data Processing and Analysis	15
6.5.1 Standardisation of Variables	15
6.5.2 Data Labelling	16
6.6 Exploratory Data Analysis	16
6.7 Data Lineage & Pipeline	18
7 MLOps for Model Management and Deployment	18
7.1 Model development - Model Selection and Hyperparameter Tuning	18
7.1.1 Logistic Regression	19
7.1.1.1 Regularisation	19
7.1.2 Artificial Neural Network	20
7.1.2.1 Number of Units in Hidden Layer	20
7.1.3 Baseline Model	21
7.2 Performance of Models	21
7.3 Preparing for Production	22
7.4 Deploying to Production	23
7.4.1 Model Deployment	23
7.4.2 Batch Scoring	24
7.4.3 Canary Release	24

7.5 Monitoring and Feedback Loop	25
7.5.1 Ground Truth Evaluation	25
7.5.2 Model Evaluation	25
7.5.3 Model Retraining	26
7.6 Model Governance	26
7.7 Model Management	27
7.8 Version Control and Collaboration	28
8 Workflow Orchestration	28
9 Conclusion	29
10 References	30

1 Executive Summary

This project proposes leveraging the unstructured text data available on Reddit's online engagement, to measure advertising effectiveness by building a NLP model. Reddit's diverse and active community offers an ideal medium for assessing the volume of engagement, the sentiment expressed, and the longevity of interest in **Super Bowl Advertisements**. These are features we will then use to refine a machine learning sentiment model that is tailored for online discourse specifically for superbowl advertisements.

This approach seeks to provide a comprehensive understanding of advertisement impact, moving beyond traditional viewership metrics to capture the nuanced and dynamic viewer responses that conventional analytics miss. The model will be useful for advertisers to gauge the sentiment of their advertisements with this NLP model, so they have a better idea of how they can improve their advertising strategies.

2 Problem Statement

2.1 Problem Definition

Despite substantial investments in Super Bowl advertisements, accurately measuring their effectiveness remains a challenge for companies (Waldow, 2024). Traditional metrics fall short of capturing the multifaceted impact of these ads, leading to difficulties in determining their return on investment (ROI) and overall value to brands (Jarboe, 2024). Online textual commentary is difficult to mine insights without a NLP model specifically trained for this task. This gap underscores the need for a more nuanced approach to capture ad sentiments.

2.2 Business Context

We are representatives of Columbia Broadcasting System (CBS), Superbowl's Main TV Broadcasting Station.

As representatives of CBS, our goal is to predict the sentiments and effectiveness of Super Bowl ads to better inform future advertising decisions, ensuring that only the most impactful ads are selected for broadcast in the years to come. To achieve this, we will be studying the Super Bowl ads comments and identify ads with positive sentiments.

2.3 Business Value

Analysing Reddit comments on Super Bowl ads based on sentiment offers CBS valuable insights into viewer preferences and perceptions. By understanding which ads resonate positively with audiences and which fall short, CBS can tailor its ad selection to maximise viewer engagement and advertising revenue (Steinberg, 2024). This data-driven approach allows CBS to identify trends, refine its advertising strategy, and ultimately enhance the overall Super Bowl viewing experience for audiences, advertisers, and stakeholders. This targeted approach ensures that CBS remains competitive in the ever-evolving landscape of televised sports and advertising.

3 Analytics Solution Statement

To address the challenge of measuring Super Bowl ad effectiveness, we propose the development of a comprehensive analytics solution. This solution implements a data pipeline that streams Reddit Superbowl advertisement comment data in real-time, analyses the sentiments of these comments, and tracks engagement over time. The solution will leverage a robust toolchain to support DataOps for data management and MLOps for sentiment analysis

model deployment and monitoring. Companies can monitor ad performance in real-time, gaining actionable insights to optimise future ad campaigns and content strategies. This analytics approach will enable a more nuanced understanding of ad impact, supporting strategic decision-making and maximising the ROI of Super Bowl ad investments.

4 Agile Analytics Project Management With SCRUM

4.1 Overall Project Planning

In our project, we made use of SCRUM methodology to be the cornerstone of our collaborative efforts, enabling us to effectively manage the intricacies of both DataOps and MLOps. We began each sprint with a thorough planning session, where we meticulously outlined the objectives and tasks ahead. Stand-up meetings once every two days kept everyone in sync, allowing us to address any roadblocks swiftly and ensure progress remained on track. Throughout the sprint, regular check-ins and collaboration sessions fostered a dynamic environment where ideas flowed freely and solutions emerged organically. Sprint reviews also provided opportunities for feedback and iteration, ensuring that our approach remained aligned with the needs of the product. By harnessing the power of SCRUM, we not only optimised our workflow but also cultivated a culture of transparency, accountability, and continuous improvement within our team.

4.2 SCRUM Sprint 1 and 2 Overview

Sprint 1 Planning

We want to focus on setting up the necessary data pipeline to ingest, transform and load the data as we require for the project. The focus will mainly be on the DataOps side of things.

For Sprint 1, we will be doing DataOps, MLOps Setup, and Agile Analytics Project Management with Scrum. Each area plays a crucial role in achieving our overarching objective of establishing a robust data pipeline for real-time ingestion of Reddit data related to Super Bowl ads, developing orchestration and version control tools, and ensuring effective project management through Agile methodologies.

Sprint 2 Planning

The main goal of Sprint 2 is to advance our MLOps capabilities within our DataOps framework, enhancing orchestration and version control tools while adhering to Agile methodologies, all aimed at strengthening our data pipeline for real-time ingestion of Super Bowl ad-related Reddit data. For Sprint 2, we will be wrapping up the DataOps, and working on the MLOps, and the Agile Analytics Project Management with Scrum.

Sprint Backlog

The Sprint Backlog of Sprint 1 will consist of User Stories 1-12, as indicated in the SCRUM Document. This contains the DataOps related tasks, and the completion of portions of the deliverables.

The Sprint Backlog of Sprint 2 will consist of User Stories 2-3, 5 and 11-18, as indicated in the SCRUM Document. This contains the MLOps related tasks, and the completion of portions of the deliverables.

Daily Standup, Sprint Review, Sprint Retrospective

The daily standup details, the sprint review and the sprint retrospective are documented in detail within the [SCRUM sheet](#). (Refer to link provided in the Point 11 Appendix) Points were collected and contributed from the team members.

4.3 Kanban Board

The kanban board was done up in conjunction with the SCRUM sheet in order to track the user stories / tasks done in real time. This allowed the team members to all be synchronised and aware of the progress of the project together. Our group used Trello, and here are big milestones at various checkpoints of our project. The link to Trello is: [Trello Link](#).

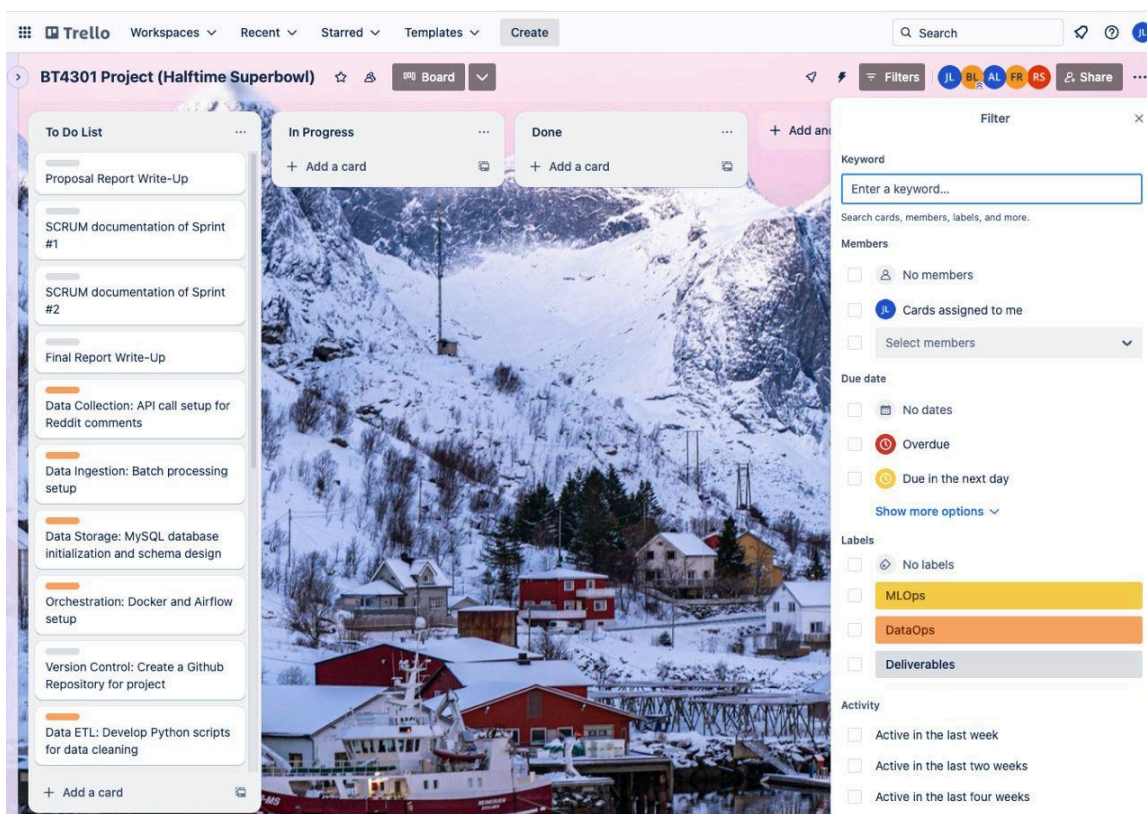


Fig 1: Start of Sprint 1

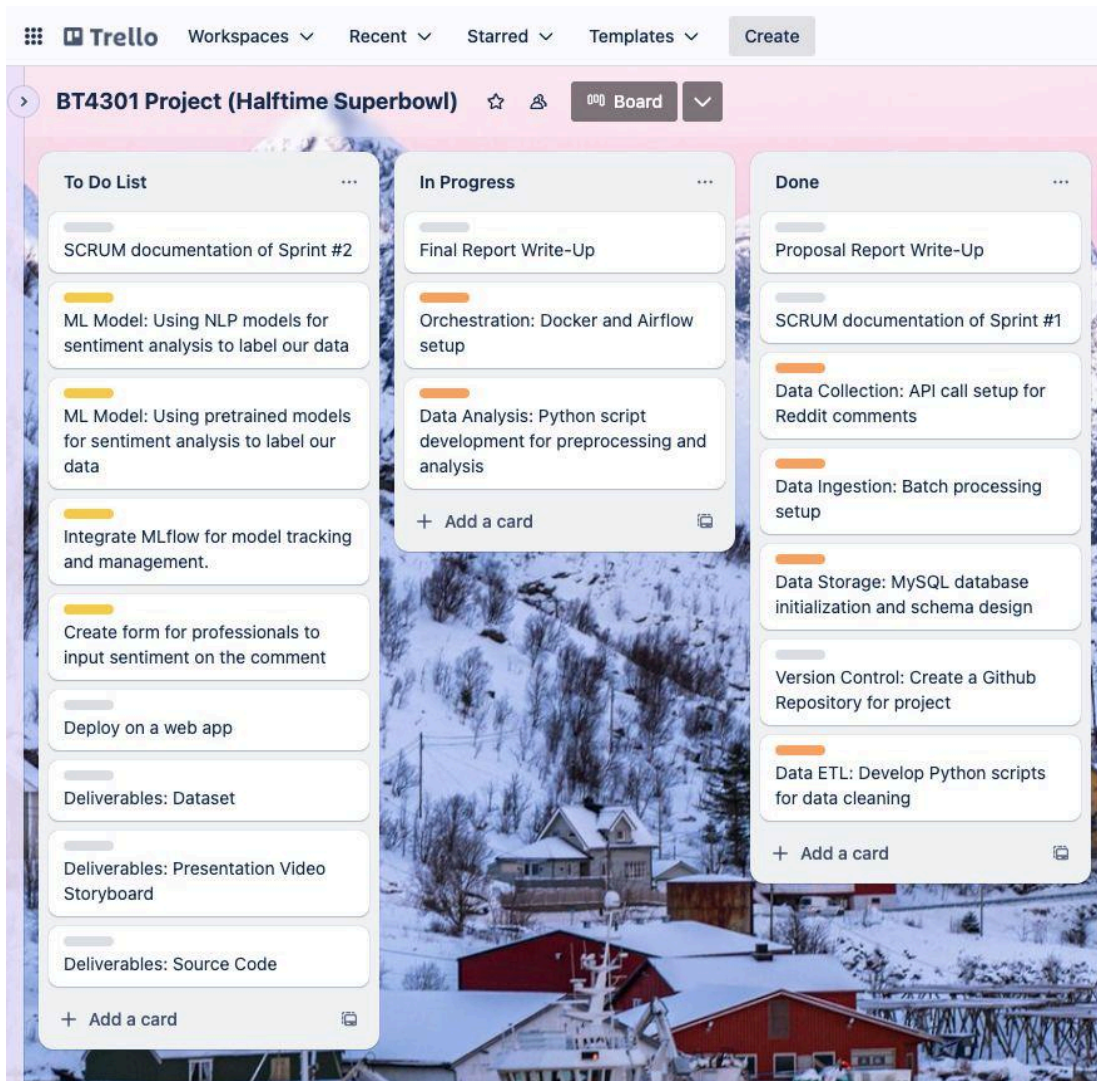


Fig 2: End of Sprint 1 / Start of Sprint 2

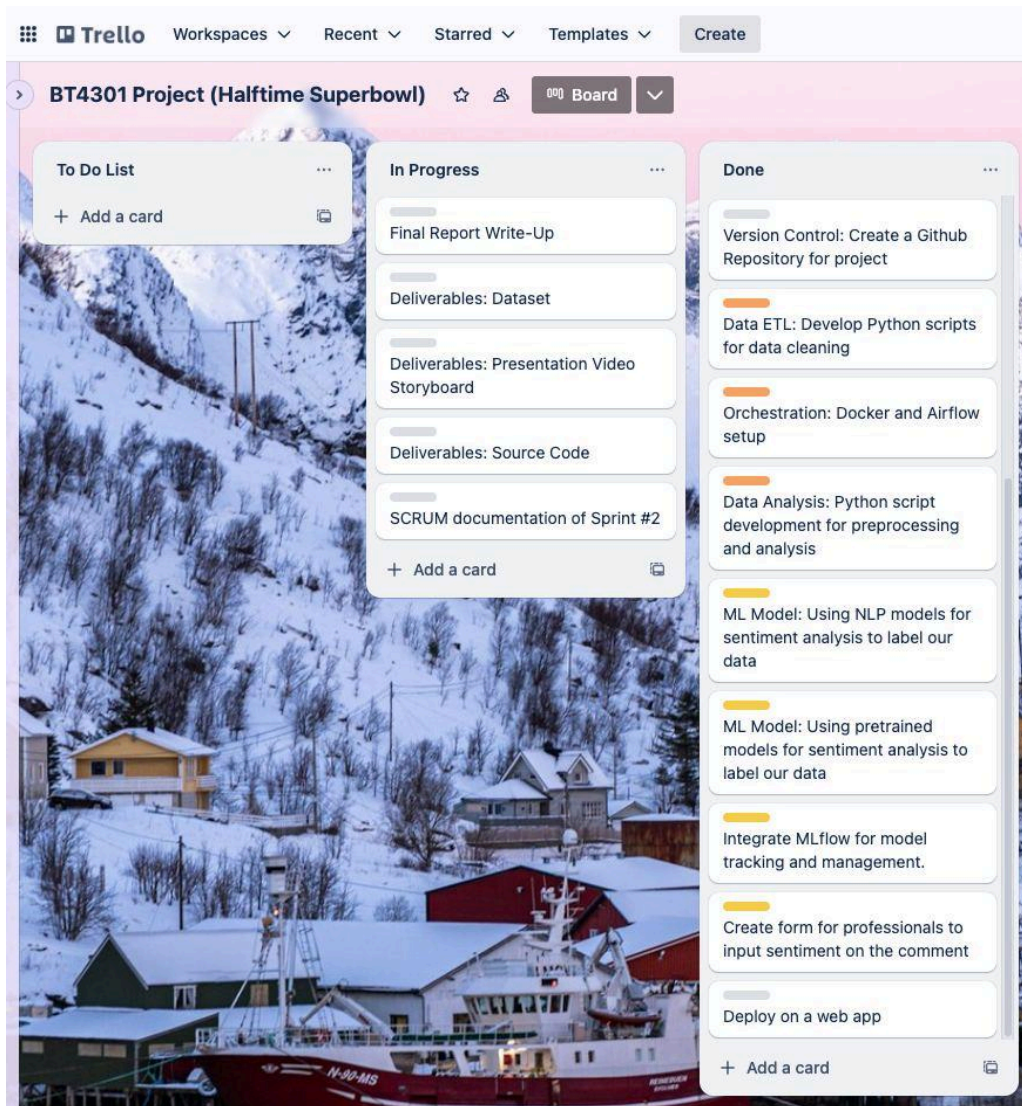


Fig 3: End of Sprint 2

5 DataOps and MLOps Toolchain

Considering the minimal requirements, the following toolchain offers a balance between functionality and simplicity:

Overview	Toolchain Purpose	Toolchain Used
1	Data Collection and Ingestion	PRAW

Overview	Toolchain Purpose	Toolchain Used
2	Data Storage	PostgreSQL
3	Data Processing and Analysis	Python (Pandas and NumPy)
		NLP Libraries: VADER, Scikit-learn
4	Workflow Orchestration	Docker
		Apache Airflow
5	MLOps for Model Management and Deployment	MLflow
6	Version Control and Collaboration	GitHub

Table 1: Key Components of the DataOps and MLOps Toolchain Utilised in the Project

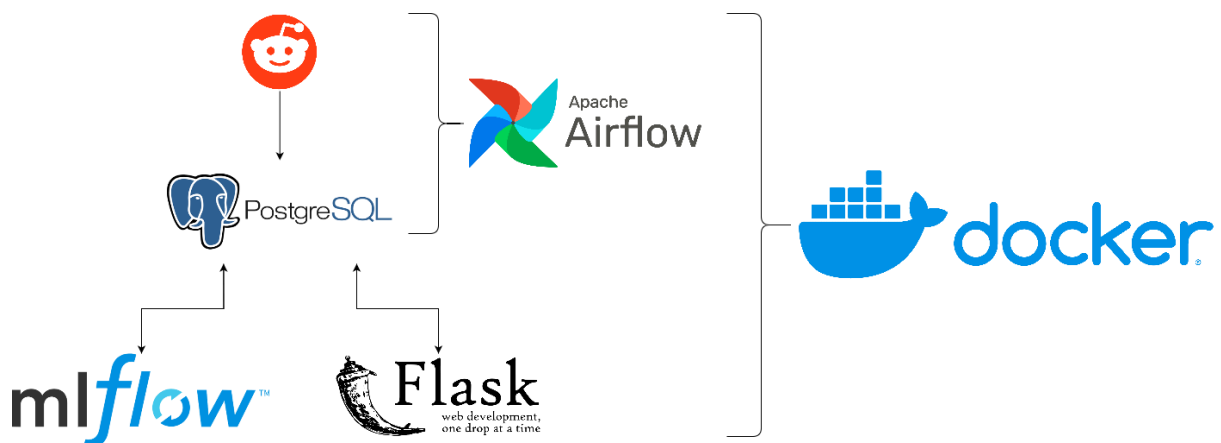


Fig 4: DataOps and MLOps toolchain diagram

5.1 Implementation Steps

Overview	Step
1	Data Collection and Ingestion with PRAW and Airflow
2	Data Storage with PostgreSQL
3	Data Processing and Analysis with Python
4	Workflow Orchestration with Docker and Apache Airflow
5	MLOps with MLflow for Model Management and Deployment
6	Continuous Monitoring and Improvement

Table 2: Overview of Implementation Steps for the DataOps and MLOps Pipeline

6 DataOps

6.1 Dataset Construction

The dataset was compiled through data extraction from Reddit utilising the PRAW Python library, focusing specifically on comments within Super Bowl advertisement megathreads and posts related to Super Bowl advertisements. Each comment entry in the dataset comprises the following extracted features:

- Comment ID
- Comment ID of the main parent comment

- Comment score (calculated as the number of upvotes minus downvotes)
- Date of the comment
- Creation date of the Redditor who authored the comment
- Karma points for the Redditor who authored the comment (derived from the cumulative comment scores across all comments authored by the Redditor)
- Comment text body
- Thread depth
- Subreddit name

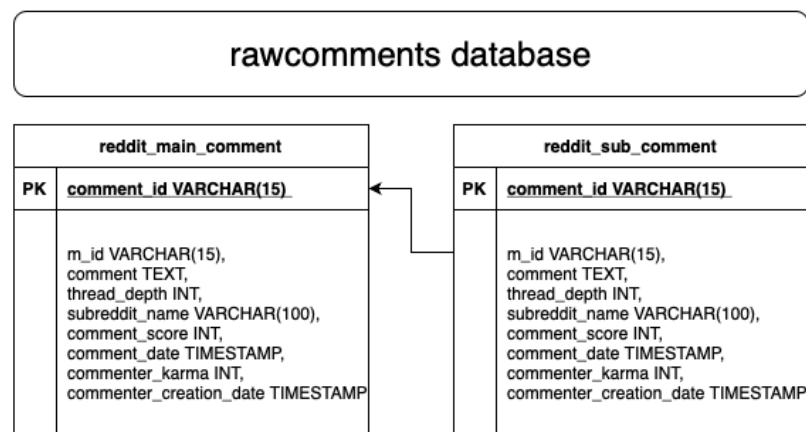


Fig 5: Shows ER diagram of rawcomments database

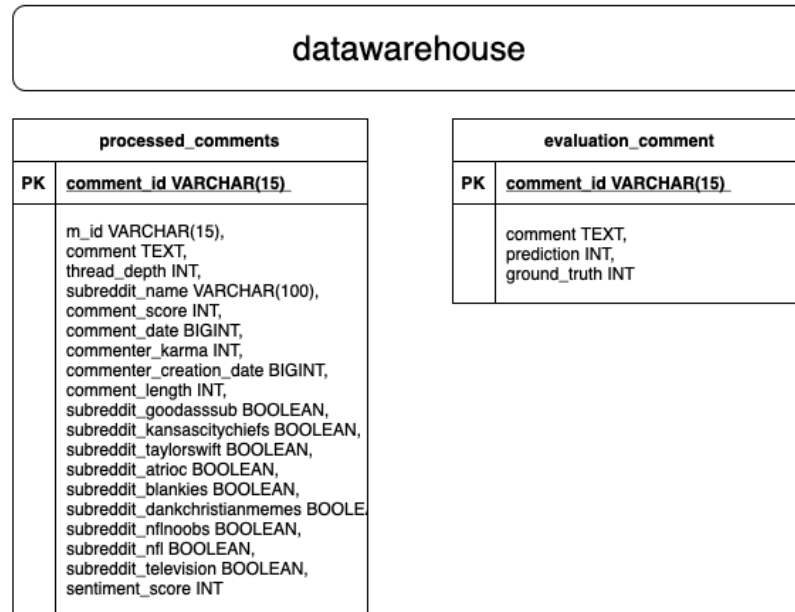


Fig 6: Shows ER diagram of datawarehouse database

6.2 Data Collection and Ingestion

For data collection, we used PRAW, a Python Reddit API wrapper that simplifies the process of interacting with Reddit's API in Python. It provides a convenient interface for accessing Reddit's vast array of content, including submissions, comments, user information, and more. PRAW handles authentication with Reddit's servers, allowing us to authenticate their requests and access Reddit's data securely. Through PRAW, we were able to fetch Reddit content as Python objects, manipulate and analyse this data.

We collected data from Reddit by extracting comments and their attributes from specific threads. It begins by initialising a Reddit instance using the PRAW library and defining functions to traverse comments recursively, extract comment attributes, and retrieve nested comment IDs. The main functionality revolves around iterating through a list of Reddit post URLs, fetching the submissions, and processing their comments. For each comment, we extract relevant attributes such as the comment ID, content, subreddit name, upvote score,

date, and commenter information. We also take into account nested comments, ensuring that all replies are captured. Finally, we consolidate the raw data into a dataframe which gets ingested into our PostgreSQL database. This will then be used later on for further analysis and processing.

6.3 Data Storage

We utilised PostgreSQL as a storage solution to persistently store the collected Reddit comment data as well as its essential features such as comment score and date, allowing for efficient retrieval and analysis through structured queries. Essentially, the raw data was ingested into the PostgreSQL database and the data was then extracted for data cleansing and transformation before being ingested into the data warehouse. Two databases were created. One for the loading of raw comments and the other for the loading of processed comments in a data warehouse. We chose PostgreSQL for its excellent integration with Docker and Airflow, this enhances our deployment strategy by simplifying containerization setup and facilitating efficient data pipeline orchestration for tasks like extraction, transformation, and loading.

6.3.1 Database for Raw Data

The 'rawcomments' database was created to store the raw data from the initial data extraction from Reddit. Two tables were created to store these comments: reddit_main_comments and reddit_sub_comments. The reddit_main_comments is used to store data from the main replies of a submission post. The reddit_sub_comments is used to store data from all the nested replies to the main replies of a submission post.

6.3.2 Data Warehouse

The data warehouse is used to store processed comments after the raw comments are extracted and transformed from the rawcomments database. Two tables were created in the data warehouse: processed_comments and evaluation_comments. The processed_comments is used to store all transformed data which can then be used for model training. The evaluation_comments is used to collect the ground truth for new data streaming into the database which will eventually be used to evaluate the performance of the trained model.

Data integrity is ensured by setting up appropriate constraints for attributes in each table.

Database	rawcomments		datawarehouse	
Table	reddit_main_comments	reddit_sub_comments	processed_comments	evaluation_comments

Table 3: Table of databases and tables created

6.4 Data Cleaning and Transformation

6.4.1 Data Cleaning

Null values were checked for all features. A few comments had null values specifically for features such as the date of creation and karma points for the Redditor who made the comment. After investigation, it was found that this issue arose due to the Redditor having terminated their account after making the comment. Hence, the PRAW library was unable to extract data from the deleted account. A Redditor's commenter karma points indicate how credible or well-received the comments that they make. Hence, the absence of such features

may significantly affect the performance of the model. Thus, we have decided to remove comments that do not belong to an existing Redditor account.

6.4.2 Transformation

Several transformations were done to provide more insightful and relevant features to our model. The first transformation includes extracting the number of words in the comment text body. This is a form of derivative feature engineering aimed to provide a more relevant feature to our model. We also decided to perform one-hot encoding for the name of the subreddit to convert the categorical data into a numerical format that can be fed into machine learning models. Given that there were 10 subreddits which data was extracted from, 9 dummy variables were created. A value of 1 is given for the subreddit variable that the comment was made on and a value of 0 for the rest of the subreddit variables.

6.5 Data Processing and Analysis

In our project, we utilised Python libraries such as Pandas and NumPy for data cleaning, manipulation, and preliminary analysis. Specifically, these tools were employed for data preprocessing tasks, including cleaning and transforming the dataset to prepare it for machine learning modelling. Following data preparation, we leveraged Scikit-learn to construct and train sentiment analysis models using the processed data. Additionally, cross-validation techniques were implemented to fine-tune model parameters and optimise performance.

6.5.1 Standardisation of Variables

Depending on the model applied, feature scaling may be important to ensure that features with a large range and higher magnitude do not necessarily have a larger influence over the model due to their larger raw variance value. This is especially important for models such as

logistic regression. Hence, it is important for features to be on a similar scale. However, it is clear that the features are on different scales. For example, the "comment_score" variable has values ranging to magnitudes in the thousands while the "thread_depth" variable likely contains values in the single digits. Hence, the values are standardised with mean = 0 and standard deviation = 1 using the z-index standardisation formula:

$$Z = \frac{x - \mu}{\sigma}$$

where z is the Z-index, x is the data point, μ is the variable mean and σ is the variable standard deviation. Note that the target variable does not need and will not be standardised.

6.5.2 Data Labelling

We opted to use supervised machine learning as the approach to tackle this problem, hence data labelling was required. To achieve this, we employed the pre-trained sentiment analysis model VADER, which assigns sentiment scores to text inputs. While alternative pre-trained models like textblob and Flair were available, we deemed VADER most suitable due to its training on web-based media data. Given that our analysis focuses on sentiments expressed in Reddit comments, a platform known for social news and discussions, VADER aligns well with our context. The model assigns sentiment scores ranging from -1 to 1 to each comment text, which we then categorised as positive (1) or negative (0) based on a threshold of 0.01.

6.6 Exploratory Data Analysis

Based on the results of our exploratory data analysis, we observed that there is no significant class imbalance problem. To ensure a comprehensive understanding of the dataset's

characteristics, we plotted the distribution of positive and negative sentiments among the comments.

Additionally, we analysed the distribution of comments over time. This analysis was aimed at understanding patterns or trends in the data, such as periods of high activity which could influence model performance or require different handling during preprocessing. We also included a pair plot of the numerical columns in our dataset. The pair plot serves as a comprehensive visualisation that showcases the relationship between all pairs of numerical features. This type of plot is particularly useful for spotting structured relationships between variables, such as linear trends or clusters, which can inform feature engineering and model selection decisions.

To facilitate continuous monitoring of our findings, we incorporated a task within our airflow task orchestration to generate and compile these visualisations into a PDF report. This automation ensures that the latest data insights are always produced when the reddit comments are being ingested. By automating the creation of this report through our Airflow DAG, we maintain an up-to-date overview of the dataset characteristics, which is crucial for both current analysis and future audits or reviews of the project's methodologies.

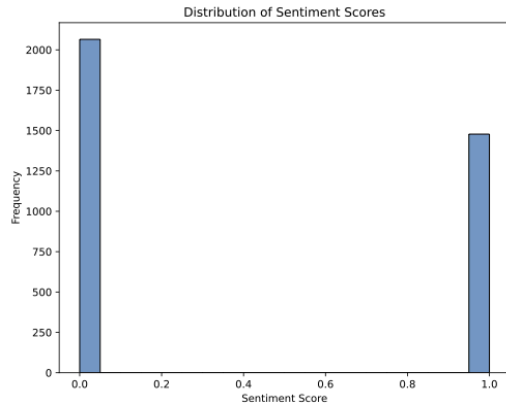


Fig 7: Distribution of Sentiment

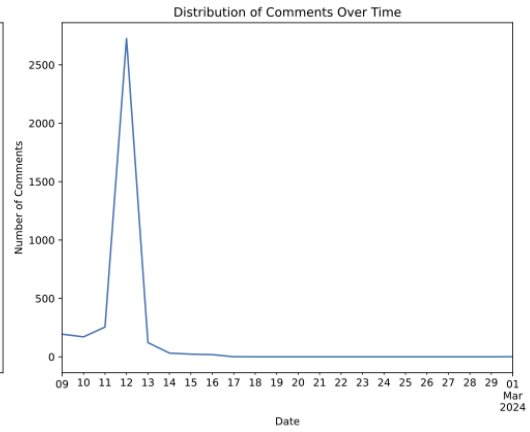


Fig 8: Distribution of comments over time

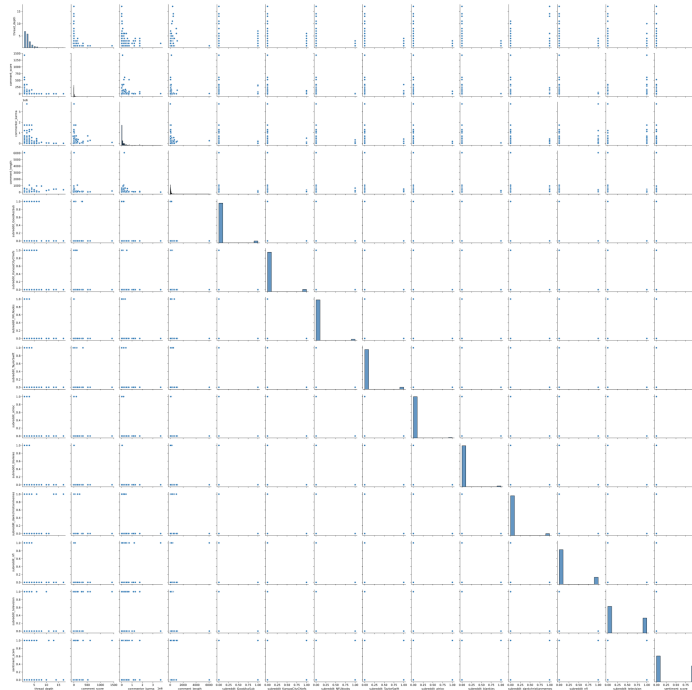


Fig 9: Pairplot diagram of numerical columns

6.7 Data Lineage & Pipeline

The setup of an Airflow DAG streamlines the systematic retrieval of Reddit comments pertaining to Super Bowl advertisements, ensuring a steady flow of data on a scheduled basis.

We define the schedule interval and orchestrate a sequence of tasks within the Directed

Acyclic Graph (DAG). Two DAGs were created to support the entire orchestration. The first DAG is a one-time data extraction that runs only once upon deployment. The main purpose of this DAG is to extract and transform data that would subsequently be used to train the model. The second DAG is a periodical data extraction that runs once a month. The main purpose of this DAG is to stream new data that would be fed into the model for prediction. Upon retrieving the ground truth for these predictions, model evaluation can then be done to evaluate the performance of the model. Further monitoring and logging is done to ensure reliability of the model. Should there be a drop in model performance, retraining of the model may be considered.

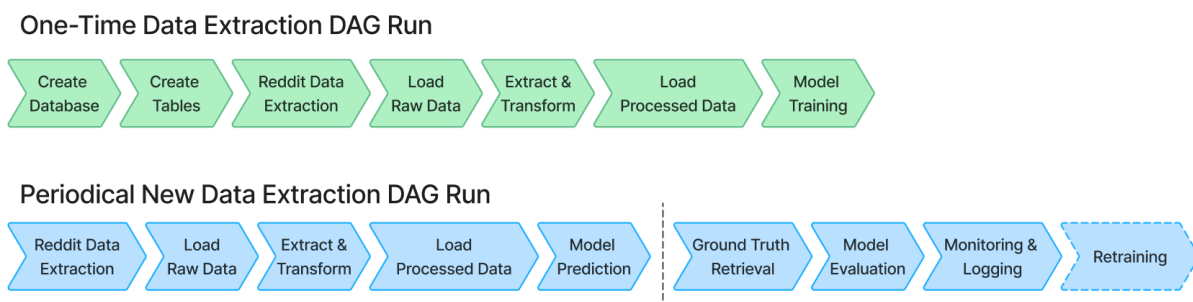


Fig 10: Workflows for different tasks

7 MLOps for Model Management and Deployment

7.1 Model development - Model Selection and Hyperparameter Tuning

In our exploration of machine learning algorithms, we experimented with two approaches: artificial neural networks and logistic regression. Artificial neural networks are highly flexible and can capture complex relationships between input features and output labels. Logistic regression provides interpretable results, making it easier to understand the contribution of each feature to the predicted sentiment score.

7.1.1 Logistic Regression

The log-odds (logit) form of the logistic regression equation is

$$\ln\left(\frac{P(\textit{Sentiment} = 1)}{P(\textit{Sentiment} = 0)}\right) = w_0 + w_1 * \textit{thread_depth} + w_2 * \textit{comment_score} + w_3 * \textit{comment_date} + \\ w_4 * \textit{commenter_karma} + w_5 * \textit{commenter_creation_date} + w_6 * \textit{comment_length} + \\ w_7 * \textit{subreddit_GoodAssSub} + w_8 * \textit{subreddit_KansasCityChiefs} + w_9 * \textit{subreddit_NFLNoobs} + \\ w_{10} * \textit{subreddit_TaylorSwift} + w_{11} * \textit{subreddit_atrioc} + w_{12} * \textit{subreddit_blankies} + \\ w_{13} * \textit{subreddit_dankchristianmemes} + w_{14} * \textit{subreddit_nfl} + w_{15} * \textit{subreddit_television}$$

This equation outlines the weighted contributions of various features to the log-odds ratio, where $P(\textit{Sentiment} = 1)$ represents the probability of a positive sentiment and $P(\textit{Sentiment} = 0)$ represents the probability of a negative sentiment. The coefficients w_0, w_1, \dots, w_{15} capture the impact of each respective feature on the sentiment classification.

7.1.1.1 Regularisation

Introducing a regularisation parameter λ helps prevent overfitting and fine-tunes the logistic regression model to achieve a better balance between fitting the training data and generalising to unseen data. The loss function E with the regularisation parameter can be expressed as:

$$E_{\lambda} = ||\hat{y} - Xw||^2 + \lambda ||w||^2$$

The logistic regression model would be evaluated with regularisation parameter

$\lambda = \{0.01, 0.1, 0.5\}$ which were chosen based on a few test runs.

7.1.2 Artificial Neural Network

For the artificial neural network (ANN) model, a 3-layer model was created with 1 input, 1 hidden and 1 output layer. The input layer has 15 units since there are 15 input attributes. The hidden layer has h units with $h = \{1, 10, 20, 30, 50\}$ which was decided based on a few test-runs. The non-linear activation function used in the hidden layer is the rectified linear unit (ReLU). The output layer is one unit which returns the predicted output of the model.

Since the target variable is binary with classes = $\{0, 1\}$, the sigmoid function was used as the non-linear activation function for the output layer. The optimizer used for the model is the stochastic gradient descent with learning rate = 0.001. The cost function used is the binary cross-entropy since the output is a binary variable.

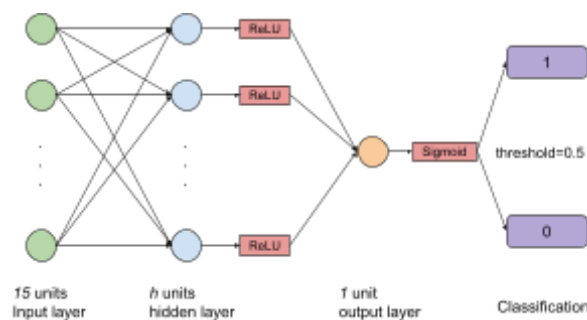


Fig 11: Layout of the Neural Network Feedforward Process

7.1.2.1 Number of Units in Hidden Layer

The number of units in the hidden layer, h , was selected as the hyperparameter to tune. While there are other hyperparameters such as the number of hidden layers and the activation

function for each hidden layer, we focused on h as it was the most straightforward hyperparameter that can be modified.

7.1.3 Baseline Model

A baseline model was created that returned the majority class label of the inner loop training set that gave the lowest validation error. This class label is then used to test against the outer loop testing set.

7.2 Performance of Models

The two-level cross validation is performed to evaluate the performance of each machine learning model with their associated optimised hyperparameter. K= 10 fold cross validation is performed twice. Each inner fold is used to train the models to evaluate which is the best hyperparameter for that model. The model, along with the optimised parameter, is then trained on the outer fold to have their model performance evaluated. The tools employed to assess model performance are sourced from the Python scikit-learn library

The error measure E used to determine the performance of the model is the misclassification rate, given as the following:

$$E = \frac{\{Number\ of\ misclassified\ observations\}}{N^{test}} = 1 - accuracy\ rate$$

Outer fold	ANN	Logistic Regression	Baseline
------------	-----	---------------------	----------

i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	50	0.499	0.5	0.425	0.442
2	50	0.442	0.5	0.363	0.389
3	50	0.459	0.5	0.425	0.431
4	50	0.421	0.5	0.333	0.373
5	50	0.528	0.1	0.367	0.381
6	50	0.472	0.5	0.376	0.415
7	50	0.486	0.5	0.379	0.418
8	30	0.480	0.01	0.398	0.438
9	50	0.525	0.1	0.387	0.427
10	50	0.446	0.5	0.404	0.458

Table 4: Results of two-level cross validation using ANN, logistic regression and baseline models

From the table of results, the logistic regression model seems to be the best performing model that consistently gives the lowest error value across all the outer folds. We also notice that $\lambda = 0.5$ seems to be the most optimised value for the regularisation parameter for the logistic regression model. Hence, we decided to proceed with logistic regression with $\lambda = 0.5$ as our final deployment model.

7.3 Preparing for Production

Before deploying our models into a production environment, several considerations need to be addressed to ensure smooth operation and optimal performance:

Performance Consideration: It is crucial to assess the computational resources required to run our models efficiently in a production environment. This includes ensuring that our model is able to process the incoming data and generate predictions at a fast enough speed for it to be usable for our end users.

Data Access Consideration: We ensured that the access to the required data sources were well-defined and secure to ensure the models have access to the necessary inputs for inference. Data pipelines and integration with databases or external APIs were also robustly implemented to fetch and preprocess data in real-time or at scheduled intervals.

Model Risk Evaluation: We established robust monitoring mechanisms through our Flask App, which will be discussed below, to detect any drift in model performance or unexpected behaviour post-deployment.

Model Testing: Tested the model by keying in an empty value as well as illegitimate values for the comment ID to see how the model reacts. No features will be extracted given these scenarios, hence checks are in place to prevent illegitimate data from being processed by the model.

7.4 Deploying to Production

7.4.1 Model Deployment

The sentiment analysis model is also deployed to production for individuals to test the model. The model takes in a Reddit comment ID and is then fed to the backend server for processing. The system retrieves the features of the Reddit comment using the ID and the associated features are then fed into the model in which a prediction will be made. The user will then be redirected to a page where details such as the comment ID, text body, prediction and the model inference time will be displayed. A simple health check can be carried out by periodically running some comment IDs into the model and checking model performance such as the prediction accuracy and model latency.

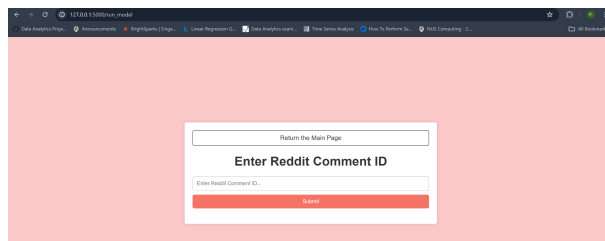


Fig 12: Comment ID input page

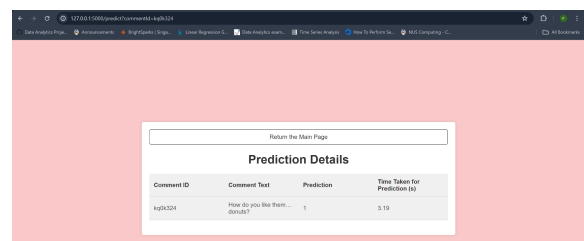


Fig 13: Model prediction results page

To prevent illegitimate comment IDs from being fed into the model, the system performs checks to ensure the comment ID is a legitimate one as well as ensuring no blank comment ID is submitted to the model for prediction. Should the features not be retrievable by the system, the Flask app will redirect users to the error page. This ensures the model is working only on real Reddit comments and not some fake inputs.

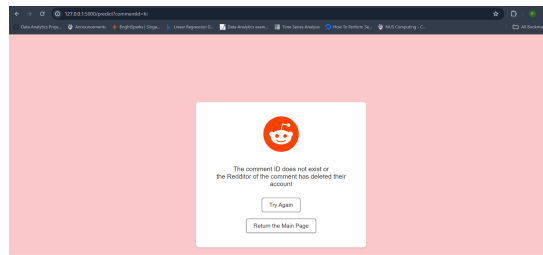


Fig 14: Error page for invalid comment IDs

7.4.2 Batch Scoring

The entire DAG operates on a batch scoring system, processing new comments collectively. Since the DAG executes monthly, comments accumulate within a single batch over the course of the month before being fed to the model for predictions on all new comments. This approach ensures efficiency in processing and prediction across the dataset, which allows for comprehensive analysis at regular intervals.

7.4.3 Canary Release

For future versions of the model, we plan to do a canary release strategy for deploying our sentiment analysis model. This will involve directing a small subset of incoming comments to the new model while the majority continue to use the existing model. Monitoring the new model's performance on this subset will allow us to collect feedback and assess its performance. As the new version becomes more stable, we can gradually increase the proportion of data directed to the new model. If issues arose, we could quickly revert to the previous version to minimise impact.

7.5 Monitoring and Feedback Loop

7.5.1 Ground Truth Evaluation

After training the model, we need to keep track of the model's performance to ensure it is working with high performance and accuracy as it continues to predict new data. The evaluations can only be done after obtaining the ground truth for the predictions of these comments. To collect the ground truths, a flask app was created to allow the comments to be rated by a highly specialised team of data annotators. By looking at the comment text body, the team will decide, using majority voting, if the comment has a positive or negative sentiment. After all new comments have been rated, the flask app will submit the ratings and have the ratings recorded in the PostgreSQL database.

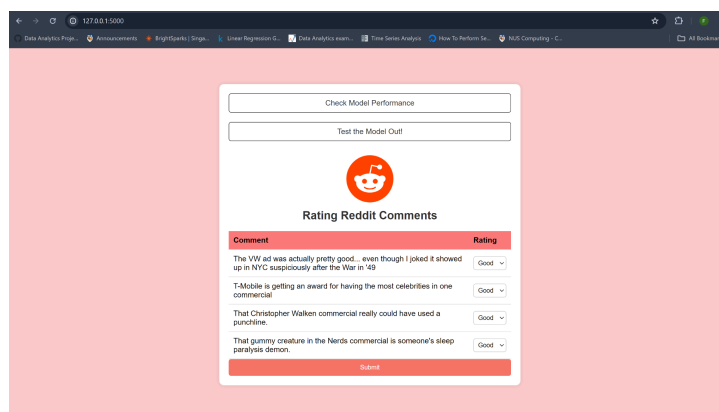
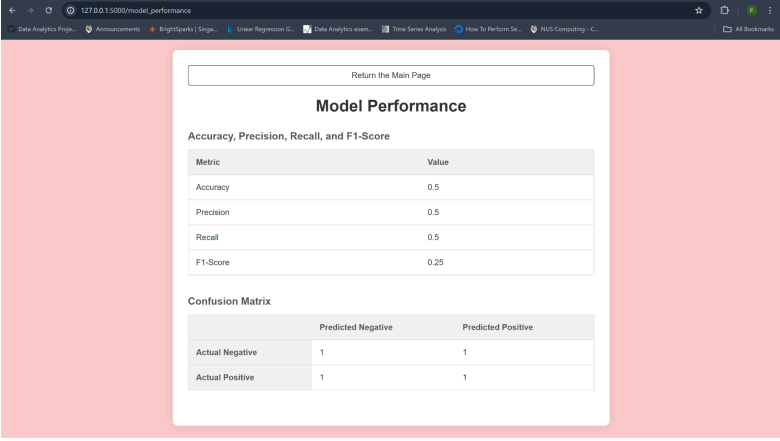


Fig 15: Comment rating page for Data Annotators

7.5.2 Model Evaluation

We monitored model performance and data pipelines using Airflow's and MLflow's integrated monitoring functionalities, enabling real-time identification of any issues or bottlenecks. Furthermore, the Flask app features a straightforward dashboard presenting various metrics for assessing model performance, including accuracy, recall, precision, and

f1-score. Additionally, the dashboard provides insights into the confusion matrix of the model.



The screenshot shows a web browser window with the URL '127.0.0.1:5000/model_performance'. The page has a light pink background. At the top, there is a button labeled 'Return the Main Page'. Below it, the title 'Model Performance' is centered. Under the title, the text 'Accuracy, Precision, Recall, and F1-Score' is displayed. This is followed by a table with two columns: 'Metric' and 'Value'. The table contains four rows: Accuracy (0.5), Precision (0.5), Recall (0.5), and F1-Score (0.25). Below this table, the text 'Confusion Matrix' is displayed. This is followed by a table with three columns: 'Actual Negative', 'Predicted Negative', and 'Predicted Positive'. The table contains two rows: Actual Negative (1, 1) and Actual Positive (1, 1).

Metric	Value
Accuracy	0.5
Precision	0.5
Recall	0.5
F1-Score	0.25

	Predicted Negative	Predicted Positive
Actual Negative	1	1
Actual Positive	1	1

Fig 16: Model performance page

7.5.3 Model Retraining

We implemented an iterative process for model refinement based on feedback and new data, employing a continuous improvement approach to uphold and augment model accuracy over time. As the DAG continues to monitor the model's accuracy, anytime the model accuracy falls below a predefined threshold, an email alert is triggered to prompt model retraining. Retraining involves extracting comments from the 'processed_comments' table in the data warehouse and replicating the initial model training steps.

Additionally, a separate model can be trained to differentiate between new and old comments to assess their relevance. If the classifier model successfully distinguishes between new and old comments, we may consider retiring old comments due to their lack of relevance or enhancing the model complexity by incorporating additional features.

7.6 Model Governance

Bias Testing for Model Development: As mentioned earlier, we used a two-level K-Fold cross validation to evaluate the performance of each machine learning model with their associated optimised hyperparameter. In doing so, we are able to monitor the consistency of the model performance across the different folds as well as the different hyperparameters. We are able to detect any potential biases or inconsistencies in the model's predictions. If the model performs consistently well across all folds, it suggests that it generalises well to different subsets of the data and is less likely to be biased towards specific data patterns present in one fold but not in others.

Furthermore, by optimising hyperparameters during the cross-validation process, we can assess the robustness of the model's performance to variations in hyperparameter settings. This helps ensure that the model's performance is not overly sensitive to specific hyperparameter choices and remains consistent across different configurations.

In summary, bias testing through a two-level K-Fold cross-validation enables us to evaluate the consistency and generalisation ability of the machine learning models, providing insights into their reliability and potential biases.

7.7 Model Management

MLflow: An open-source platform for managing the end-to-end machine learning lifecycle, facilitating experiment tracking, model versioning, and deployment

We integrated MLflow for tracking experiments, including model training metrics and parameters. This facilitates model comparison and selection by managing different iterations of the sentiment analysis model, ensuring reproducibility and rollback capabilities.

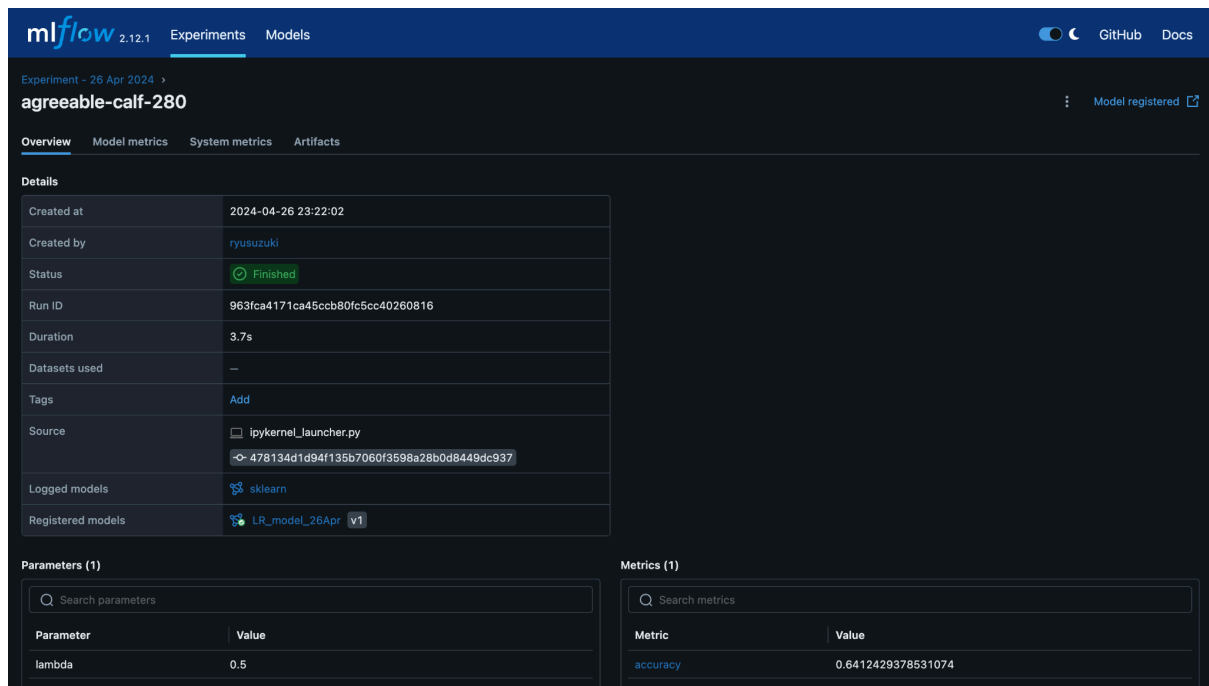


Fig 17: Screenshot of Best Model (26th April)

We then deployed the selected model to a production environment using MLflow's model serving capabilities, making it accessible for real-time sentiment analysis of incoming Reddit data.

7.8 Version Control and Collaboration

We utilised GitHub for version control, collaboration, and code management. This ensured that the development process was streamlined and that our team members could work together effectively on our project's codebase by documenting code changes and ensuring version control.

8 Workflow Orchestration

We utilised Docker to encapsulate our entire workflow for this project into containers. This allows us to create consistent and isolated environments for running the analytics workflows. By containerizing the Python environment and dependencies, we are able to ensure

consistency across our personal development environments as well as production environments. This allows us to eliminate the hassle of always having to resolve conflicts between our environments which would be very troublesome and inefficient.

Within our dockerized environment, we integrated our Apache Airflow service for orchestrating complex computational workflows and data processing pipelines. We set up Airflow to automate and schedule our DataOps pipeline, including steps for data ingestion, processing, model training, and validation. We also defined Airflow DAGs (Directed Acyclic Graphs) to manage task dependencies and execution order.

As mentioned earlier, we used MLflow for managing the end-to-end machine learning lifecycle, facilitating experiment tracking, model versioning, and deployment. To streamline this entire process, we also integrated our MLflow server along with its dependencies into our container. Similarly, this allows us to ensure consistency between environments.

9 Conclusion

In conclusion, we explored the potential of leveraging Reddit's online engagement to measure the effectiveness of Super Bowl ads through sentiment analysis. By developing a comprehensive analytics solution, we aimed to address the challenge of measuring Super Bowl ad effectiveness, providing CBS with valuable insights into viewer preferences and perceptions.

Through the integration of DataOps and MLOps methodologies, we have established a robust data pipeline and sentiment analysis model, enabling CBS to make informed decisions about ad selection and optimization.

Additionally, our agile approach, utilising SCRUM methodology, ensured efficient collaboration and adaptation throughout the project lifecycle, enhancing transparency and accountability within our team.

Leveraging Docker for containerization, MLflow for model management, and Airflow for workflow orchestration and scheduling further streamlined our process, facilitating seamless deployment and maintenance of our analytics solution.

Moving forward, our continuous monitoring and feedback loop, along with rigorous model governance and version control, ensure the sustainability and reliability of our analytics solution. This project showcases the importance of embracing data-driven approaches in advertising decision-making, empowering CBS to remain competitive in the dynamic landscape of televised sports and advertising.

10 References

OpenAI. (2023). ChatGPT (Mar 14 version) [GPT-3.5].

<https://chat.openai.com/chat>

Steinberg, B. (2024, February 6). Variety. Variety.

<https://variety.com/2024/tv/news/super-bowl-commercials-cbs-univision-nickelodeon-1235899592/#!>

Waldow, J. (2024, March 14). How brands are measuring Super Bowl ad success. Modern Retail.

<https://www.modernretail.co/marketing/how-brands-are-measuring-super-bowl-ad-success/>

Jarboe, G. (2024, February 20). Who won: Measuring the most effective Super Bowl 2024 Ads. Search Engine Land.

<https://searchengineland.com/measuring-most-effective-super-bowl-2024-ads-437699>

11 Appendix

GitHub Repo: <https://github.com/ryuniqlo/bt4301-group11>

Scrum Sheet:

<https://docs.google.com/spreadsheets/d/1FMA2N65mVHH4VswosteisAN1aYAHsIFXuCA175pe5qQ/edit?usp=sharing>

Trello Link:

<https://trello.com/invite/b/vLu5M634/ATTI42e2fe1caa8976745069278f2c3e8f2b06E3ACDB/bt4301-project-half-time-superbowl>