**Tweets Engagement Predictions Report**

*Team members: Thi Xuan Nguyen, Quy Dam Son, Tao Zhang*

1. **Project overview**

The goal of this project is to predict which tweets are likely to receive higher engagement using a combination of textual and metadata features. Engagement refers to user interactions such as likes, retweets, and reach. We aim to build predictive models using machine learning and NLP techniques grounded in both statistical analysis and feature engineering. This project was completed using a combination of tools and learning resources. We referred to course material and code examples from the NLP module as a foundation for our preprocessing, modeling, and evaluation pipeline. Additionally, we used generative AI tools to assist in refining code structure, debugging, and improving documentation clarity.

1. **Dataset summary**

The original dataset Twitterdatainsheets.csv has 206,294 rows and 15 columns, covering a wide range of metadata and engagement signals. It includes unique identifiers for each tweet (TweetID), user (UserID), and optionally location (LocationID). Temporal dimensions are captured through Weekday, Hour, and Day, while tweet content is stored in the text field. Language (Lang), sentiment (Sentiment), and reshare status (IsReshare) provide contextual and behavioral metadata. The core engagement indicators are Reach, RetweetCount, and Likes, which serve as the primary basis for building engagement prediction models. Additional user reputation or influence is reflected in the Klout score. Some columns contain inconsistent formatting or mixed types, which necessitates preprocessing before analysis. Overall, the dataset offers a rich foundation for modeling tweet engagement, blending content-based features with metadata and user activity.

1. **Data preprocessing & Feature engineering**

To prepare the dataset for modeling, we conducted multiple rounds of data cleaning, feature transformation, and feature engineering, aimed at enriching the input space for predicting tweet engagement.

**Preprocessing:**

* Language Filtering: Only tweets in English (Lang == 'en') were retained.
* Text Cleaning: Missing tweets (text column) were dropped to ensure consistent textual analysis.
* Standardization: Columns were stripped of whitespace and unified in formatting to avoid errors during processing.
* Generate dummy variables for both categorical and discrete variables.

**Feature Engineering:**

We created several new variables from the raw tweet content and metadata:

* Text-based Features:
  + num\_hashtags: Count of hashtags using regex #\w+
  + num\_mentions: Count of mentions using @\w+
  + num\_urls: Number of URLs (using patterns like "http" or "www")
  + num\_questions and num\_exclamations: Counts of ? and ! respectively
* NLP Sentiment Features:
  + polarity: Tweet-level sentiment polarity from TextBlob (ranges from -1 to +1)
  + subjectivity: Degree to which the tweet is opinionated vs objective
  + sentiment: Segment the polarity into 3 types: Positive, negative and neutral
* Time-based Features:
  + is\_weekend: Flag for tweets posted on Saturday or Sunday
  + is\_workhour: Flag for tweets posted between 9AM and 5PM
  + hour\_bin: Categorical binning of tweet hour into morning, afternoon, evening, night
* Call-to-Action:
  + has\_call\_to\_action: Binary indicator for presence of CTA phrases like "apply", "click", "read", "check", etc.

**Topic Detection :**

We choose BERTopic here instead of simple word embeddings for several key advantages:

1. **Contextual understanding**: BERTopic leverages BERT's contextual embeddings which capture semantic meaning based on the surrounding context, unlike traditional word embeddings that assign fixed vectors to words regardless of context.
2. **Automated topic discovery**: BERTopic can automatically identify coherent topics without needing to specify the number of topics in advance, making it more flexible for exploratory analysis of social media data.
3. **Document-level analysis**: Rather than just embedding individual words, BERTopic works at the document level, capturing the overall themes across entire tweets.
4. **Hierarchical clustering**: BERTopic uses clustering techniques on the embeddings to group similar documents, which allows for discovering natural topic structures in the data.
5. **Topic interpretation**: BERTopic provides methods to extract representative terms for each topic, making the results more interpretable than raw embeddings.

This approach is particularly valuable for social media analysis where content may be short, use informal language, and cover a diverse range of topics that might be difficult to identify with traditional topic modeling approaches.

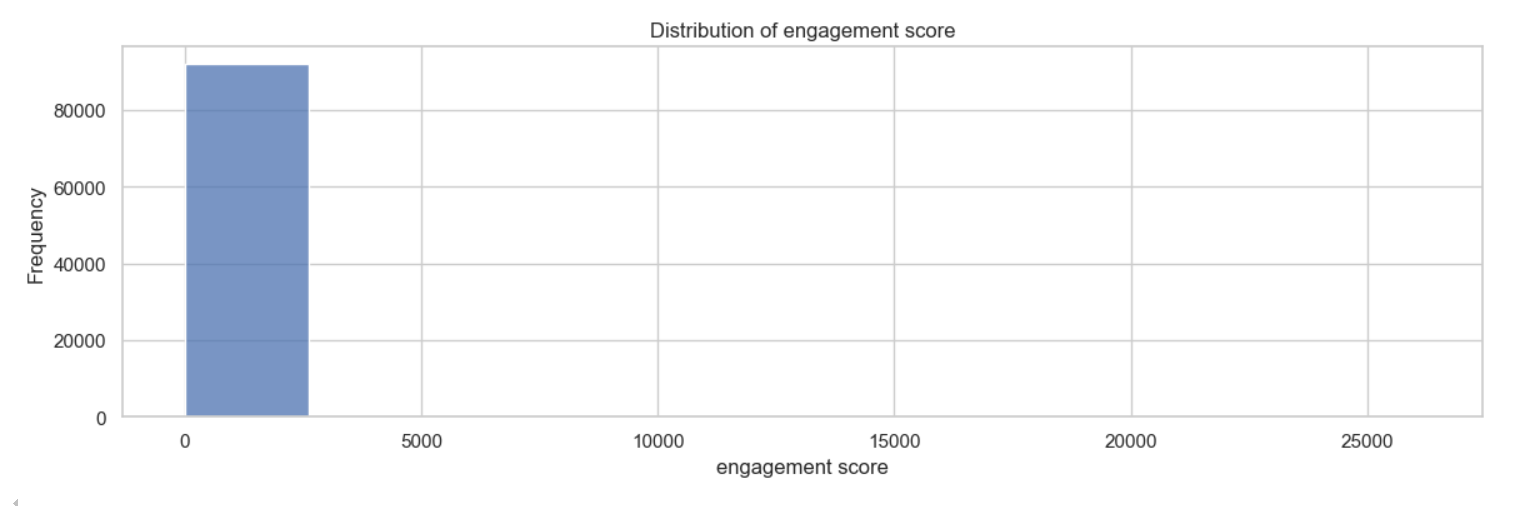
**Engagement Target Engineering**

We defined target variable as :

* engagement\_score = Likes + RetweetCount

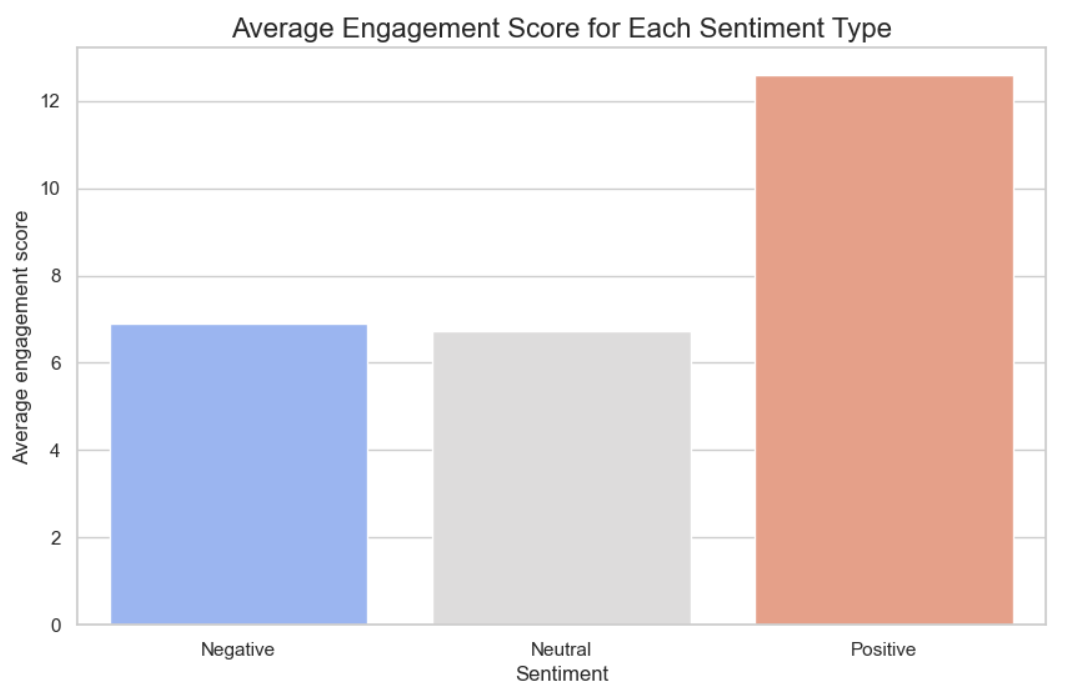
1. **Data analysis**

We began the exploratory analysis by visualizing the distribution of key engagement metrics. Histograms of engagement score showed a highly right-skewed distribution, with the majority of tweets receiving low engagement and only a small fraction going viral.

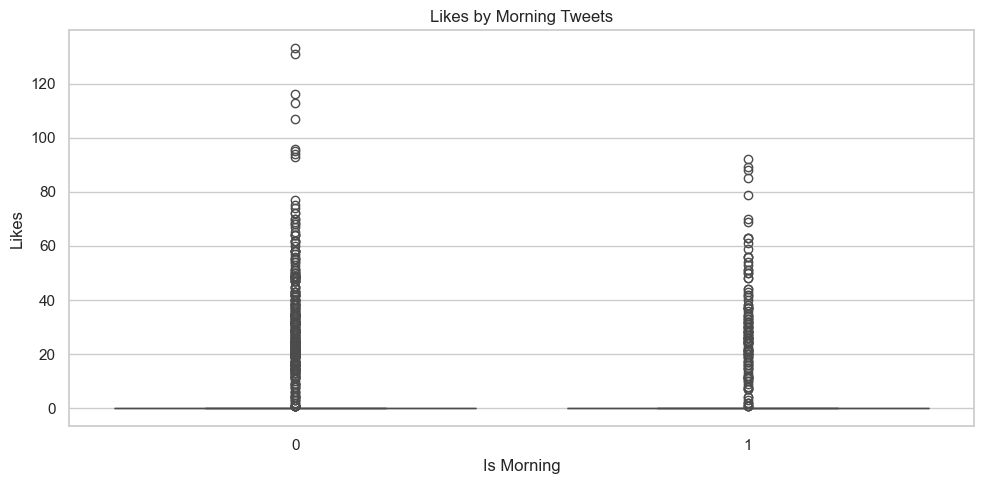
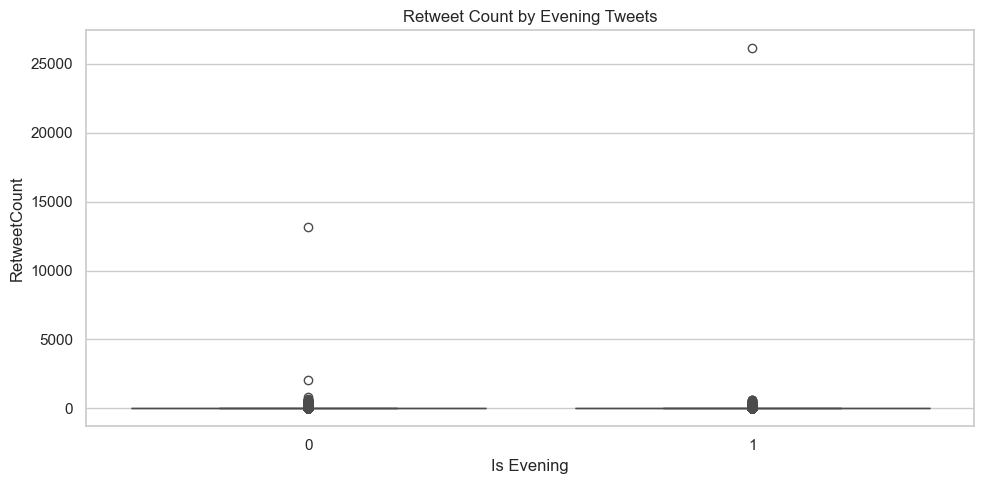


This visualization demonstrates that positive content generates nearly twice the engagement compared to negative or neutral content. The minimal difference between negative and neutral engagement is notable, while positive sentiment shows a dramatic increase in audience interaction.

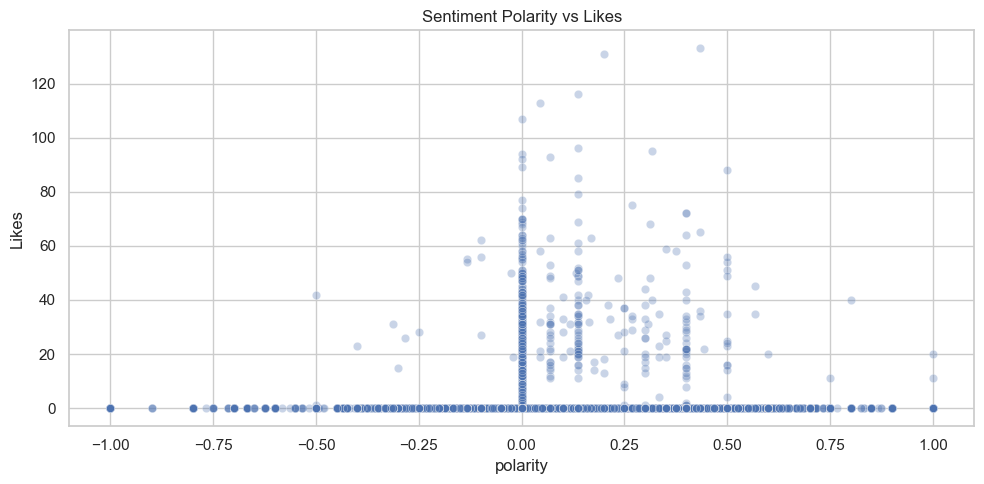
The chart suggests a strong correlation between positive sentiment and higher engagement levels in this dataset.



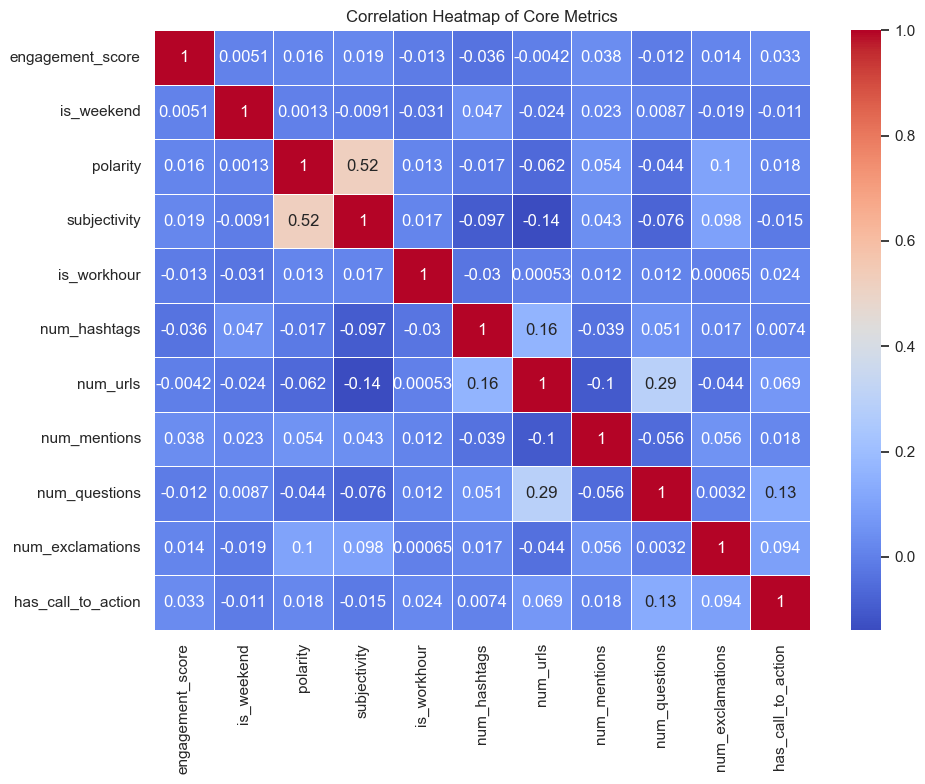
To better understand contextual effects, we visualized engagement across different time categories. Boxplots showed that tweets posted in the evening tended to receive more retweets, while tweets in the morning did not show a clear advantage in terms of likes.



Sentiment analysis also revealed meaningful patterns: scatterplots of polarity and subjectivity against Likes and Retweets indicated a mild positive correlation, particularly for tweets with moderately positive sentiment.



To quantify these relationships, we generated a correlation heatmap for core features including Reach, Likes, RetweetCount, Klout, Sentiment, polarity, and subjectivity. The results confirmed that while some variables were moderately correlated (e.g., polarity and sentiment with engagement), there was no extreme multicollinearity, supporting the inclusion of diverse features in our models. Finally, a word cloud of tweet content revealed frequent themes and keywords, offering insight into the topics that drive user interaction.



The correlation heatmap revealed that most features had very weak linear relationships with the target variable engagement\_score, suggesting that engagement is influenced by more complex, nonlinear interactions. The strongest correlation observed was between polarity and subjectivity (0.52), which is expected as more opinionated tweets tend to carry clearer sentiment. Features like num\_mentions (0.038) and has\_call\_to\_action (0.033) showed slightly higher correlations with engagement, but still remained weak. These results support the use of tree-based models like Random Forest, which are better suited to capturing subtle patterns and interactions that linear models might miss.

The word cloud analysis of tweets related to cloud computing highlights several key themes and trends within the tech industry, particularly in cloud services and job opportunities. Here are the significant observations:

**Focus on Major Cloud Platforms**:

The word cloud emphasises AWS, Google Cloud, and Microsoft Azure, indicating widespread discussion and usage of these major cloud platforms.

**Job Opportunities in Cloud Computing**:

There is a clear focus on job search, hiring, and specific roles like engineer, developer, and cloud architect, reflecting the growing demand for cloud-related jobs, particularly with an emphasis on AWS certifications.

**Seattle Tech Hub**:

The presence of Seattle and WA suggests that many tweets are centered around job opportunities and developments in Seattle, a key tech hub, especially for Amazon and cloud computing professionals.



1. **Data modeling**

In this project, we experimented with a wide range of regression models to predict tweet engagement, including Linear Regression, Ridge Regression, Lasso Regression, Random Forest, Gradient Boosting, and XGBoost. These were initially tested using different target formulations and feature sets to understand how well each approach captured engagement dynamics.

After iterative experimentation and evaluation, we streamlined our modeling process to focus on a single, interpretable target: **engagement\_score = Likes + RetweetCount**

This score served as a proxy for direct user interaction, combining the most immediate indicators of tweet popularity. We then applied a log transformation to this target **(log(engagement\_score + 1)) to reduce skewness and stabilize model performance**.

For the final evaluation, we focused on two high-performing tree-based models:

* Random Forest Regressor
* Gradient Boosting Regressor (GBR)

Both models were trained using a rich set of numeric features derived from the tweet text and metadata, including sentiment polarity, subjectivity, punctuation counts, time-of-day flags, and call-to-action indicators. The models were evaluated using 5-fold cross-validation, and performance was assessed using R², RMSE, and MAE.

1. **Evaluation**

| **Model** | **R2** | **RMSE** | **MAE** |
| --- | --- | --- | --- |
| Random Forest | 0.7305 | 0.6591 | 0.4180 |
| Gradient Boosting | 0.4654 | 0.9285 | 0.6547 |

In the final phase of the project, we evaluated two tree-based regression models — Random Forest and Gradient Boosting Regressor — using the log-transformed engagement\_score (defined as Likes + RetweetCount) as the target variable. Both models were trained on a set of engineered features derived from tweet content and metadata, including sentiment polarity, subjectivity, punctuation patterns, posting time, and call-to-action indicators. The results clearly indicate that Random Forest outperformed Gradient Boosting across all evaluation metrics.

With an R² of 0.7305, Random Forest explained over 73% of the variance in the target variable, significantly higher than Gradient Boosting's R² of 0.4654. It also yielded lower prediction errors, with an RMSE of 0.6591 and MAE of 0.4180, compared to 0.9285 and 0.6547, respectively, for Gradient Boosting. These results confirm that Random Forest is the most effective model for predicting tweet engagement in our pipeline. Its ability to handle nonlinear relationships, resist overfitting, and leverage diverse input features made it particularly well-suited for this task.

1. **Conclusion**

In this project, we aimed to predict tweet engagement using a combination of textual, temporal, and structural features extracted from tweet metadata. After exploring multiple modeling approaches and engineering a variety of relevant features, we evaluated the performance of several machine learning models. Our final modeling phase focused on Random Forest and Gradient Boosting, with the target variable defined as the sum of likes and retweets, log-transformed to reduce skewness.

The results demonstrate that Random Forest outperformed Gradient Boosting across all key evaluation metrics. It achieved an R² of 0.7305, RMSE of 0.6591, and MAE of 0.4180, indicating strong predictive performance and generalization. Gradient Boosting, while still outperforming basic linear models, fell short with an R² of 0.4654, highlighting its relative sensitivity to hyperparameters and overfitting risk.

Based on these outcomes, we recommend the following:

* Use Random Forest as the primary model for tweet engagement prediction due to its robustness, accuracy, and minimal need for hyperparameter tuning.
* Consider hyperparameter optimization (e.g., via GridSearchCV) for Gradient Boosting if computational resources allow, as it may narrow the performance gap.
* Incorporate more advanced NLP techniques in future work, such as BERT embeddings or topic modeling, to capture deeper semantic patterns in tweet text.
* Explore time-series behavior of users or engagement over different time windows to enhance temporal modeling.
* Deploy the model in a real-time or batch prediction setting to support applications like automated tweet scoring, content planning, or social media optimization.