Adobe Behaviour Simulation Challenge

Green Kedia Mukil M Abhinav Jha Geeth Chand Chinni

Abstract

In the evolving world of digital marketing, understanding and boosting user engagement is key to successful brand outreach. This work addresses the challenge of simulating user behavior on social media by predicting how tweets will perform in terms of likes, based on factors such as content, company, and media. Additionally, it explores the creation of content tailored to achieve specific engagement goals. By equipping marketers with insights into user interactions, brands can develop smarter strategies to connect with audiences, optimize content, and effectively drive their online marketing efforts.

1 Solution Approach for Task 1

1.1 Data Exploration

During our exploration, we discovered that the dataset was highly skewed in terms of tweet engagement. Specifically, only about 1 percent of the data samples had more than 10,000 likes. This pronounced skew made it challenging to train predictive models, as the majority of tweets fell into lower engagement ranges. Upcoming sections deal with how we handle such a dataset.

1.2 Baseline BERT Model Using CLS Embeddings

To establish a baseline, we leveraged a pre-trained BERT model and extracted the CLS token embedding from each tweet. The CLS token serves as a summary representation of the entire tweet. These embeddings were then passed through a neural network to predict the number of likes. This model resulted in a root mean squared error (RMSE) of approximately 3.7k. While this provided a starting point, there was room for improvement, especially considering the simplicity of using only the CLS token as input.

1.3 Enhancing Representations with Mean Token Embeddings

In order to capture a richer representation of the tweet, we shifted from using only the CLS token to averaging the embeddings of all tokens within the tweet. This approach is more effective because BERT's hidden layers distribute relevant information across all token embeddings, not just the CLS. By averaging these token embeddings, we can capture a more comprehensive representation of the tweet's content, thereby giving the model access to a broader range of contextual information, which can lead to better prediction accuracy.

1.4 Log-Transformation of Likes to Address Skewness

One significant challenge in predicting likes was the highly skewed distribution of the dataset. Approximately 98% of tweets in the dataset had fewer than 5,000 likes, with a small number of tweets receiving much higher counts. To mitigate the effects of this skewness, we applied a log-transformation to the number of likes, effectively compressing the range of values and allowing the model to better capture patterns in the lower and middle ranges of the data. This approach also prevents the model from being disproportionately influenced by a few outlier tweets with an excessively high number of likes, which could otherwise skew the predictions.

1.5 Handling Data Leakage with Group Shuffle Split

Due to the data leakage issue (discussed in Section 4), we opted to avoid using the standard train-test split. Instead, we implemented a group shuffle split, which further improved the performance and robustness of our model.

1.6 Novelty of Our Model

The key novelty in our approach lies in combining these multiple techniques:— mean token embeddings, log-transformed targets, and group-based data splitting. Each of these adjustments individually improved the model's performance, but together they formed a coherent strategy that allowed us to effectively handle the challenges posed by the dataset. By addressing both data representation and data distribution, our model stands out in its ability to generalize across different tweets and user behaviors. This integrated approach helped us significantly improve prediction accuracy beyond the initial baseline.

2 Alternative Approaches for Task 1

The below subsections deal with the atlernative approaches have attempted.

2.1 Limitations and Exploration of Image Data

It may seem surprising that our final model does not incorporate image data, but this was due to two key reasons. First, we lacked the time and compute resources to fine-tune an image-based model. The other reason was that the images in the dataset were too diverse and lacked consistent features for the model to learn effectively. While a larger, more complex model might have been able to capture these intricate visual features, this was beyond the scope of our project due to limited computational capacity.

We did experiment with EfficientNetB0, a model known for its efficiency in low-resource settings. Although we fine-tuned it on the available image data, the results were constrained by the limited diversity and detail in the dataset. The model offered a useful baseline to assess the potential impact of visual features on predicting tweet likes, but the performance suggested that significantly larger datasets and more powerful resources would be needed to gain meaningful improvements beyond the text-only approach.

Given the limitations of the image data, we also explored image captioning as a way to enhance the model's understanding of visual content. We tested methods such as BLIP Captioning, BLIP2's Conditional Image Captioning, and ViT-GPT, which generate captions based on the image context. Theoretically this would have boosted our model's performance by a lot, however, due to our resource constraints, we were unable to fully integrate these models into our pipeline, further highlighting the practical challenges of handling computationally intensive models in low-resource environments.

2.2 Using BERTweet for Classification

Given the high level of skewness in our data, we developed a two-step approach to address this challenge: classification followed by regression. Theoretically, it is difficult for a single model to capture the nuances across the entire range of likes, especially when the distribution is heavily imbalanced. By first training a classifier, we were able to segment the data into more manageable subsets, each representing a different range of likes. This made it easier to train specialized regressors for each subset, allowing the model to focus more effectively on the distinct patterns within each range, ultimately improving prediction accuracy.

- Classification: We binned the number of likes into categories and trained Bertweet as a classifier to predict the appropriate bin for each tweet. By classifying tweets into smaller, more manageable bins, we aimed to simplify the prediction task.
- Bin-Specific Regression: In theory, after the classification step, we could train separate regression models tailored to each bin. These specialized regressors would focus on predicting the exact number of likes within their specific range, potentially improving our overall prediction accuracy.

The classification step with Bertweet performed well, but due to time and resource limitations, we were unable to proceed with training the individual regressors or fully fine-tuning Bertweet for classification. Nevertheless, this experiment laid the groundwork for a potential future approach that combines classification and regression for improved performance.

2.3 Feature Engineering and Contextual Enhancements

To enhance the model's predictive capability, we extracted several key features from the dataset.

First, Sentiment Analysis was performed on each tweet, providing insights into the tone of the content. Since sentiment-driven content, whether

positive or negative, often garners more engagement, this feature offered a valuable signal for predicting likes. We also incorporated Emoji Handling, recognizing that emojis play a significant role in conveying emotions succinctly on social media. By identifying and processing emojis within the tweets, we enabled the model to capture their potential influence on user engagement.

Next, we considered Mentions and Hyperlinks, counting the number of user mentions and external links in each tweet. These elements often affect the visibility and interaction levels of tweets, offering additional data for the model to learn from. Furthermore, we derived Date and Time Features, such as the day of the month, month, and day of the week, allowing the model to capture temporal patterns in engagement, which can fluctuate based on external factors like weekends, holidays, or seasonal trends.

We also applied Topic Modeling with Latent Dirichlet Allocation (LDA) to uncover underlying themes in the tweet content. LDA assigns each tweet a probability distribution over a set of topics, enabling the model to learn from the contextual themes that drive engagement. Lastly, we Tokenized Usernames and Companies, treating them as categorical variables. By doing so, we aimed to capture any specific engagement patterns associated with particular users or brands, further enriching the model's feature set.

3 Comprehensive Overview of Models and Techniques Tested

3.1 CLIP and Random Forest for Image-Text Classification

In this approach, CLIP was used to generate embeddings that capture semantic and visual features from both images and text, which were then fed into a Random Forest classifier. Despite the potential of combining CLIP's powerful feature extraction with a classic classifier, this approach yielded suboptimal results. The likely causes were the limitations of Random Forest in handling high-dimensional data and the complexity of the embeddings. This suggests that deeper models or more sophisticated classifiers may be more effective for such tasks.

3.2 BERTweet Classification Model

To handle tweet classification, a BERTweet model was fine-tuned on a labeled dataset. BERTweet, being pre-trained on Twitter data, is well-suited for dealing with informal language, abbreviations, and social media syntax. The model's performance was evaluated using accuracy, providing a comparison point against the primary approach. The fine-tuning process allowed BERTweet to generalize effectively to classify social media-specific text.

3.3 BLIP Embeddings for Multimodal Tasks

BLIP (Bootstrapping Language-Image Pre-training) was employed to handle multimodal tasks, extracting embeddings that represented both visual and textual data in a shared space. This allowed the model to capture rich, multimodal information. The BLIP-generated embeddings were used for classification and clustering, providing additional insights that text-only models couldn't offer.

3.4 Fusion of Text and Image Embeddings Using BERT

Another method involved combining BERT-based text embeddings with image embeddings for a multimodal approach. By integrating both textual and visual features, the model could capture richer insights than what could be achieved with text or image data alone, improving classification performance.

3.5 XGBoost on Non-Text, Non-Image Features

This approach focused on features that were not derived from text or images, utilizing XGBoost to model relationships in numeric, categorical, or structured data. XGBoost is a robust framework for tabular data, and this strategy provided complementary insights to the text and image-based models, further enhancing the solution's robustness.

3.6 CardiffNLP Twitter-RoBERTa Model with Tokenized Usernames and Date-Time Embeddings

A CardiffNLP Twitter-RoBERTa model was used to extract embeddings from tweets, along with tokenized usernames and rich date-time embeddings. This feature set, which included inferred company names, was used to train a fully connected neural network. The model achieved an RMSE of 1.6k and an R^2 score of 0.9, demonstrat-

ing strong performance in blending text, user information, and temporal data.

3.7 Exclusion of Tokenized Username Embedding

In a variation of the previous approach, the tokenized username embeddings were excluded, and the model was trained on the remaining features. The resulting RMSE was 1.7k, slightly higher than the model with the tokenized usernames, but still effective, showing that text, inferred company, and date-time features were sufficient for the task.

3.8 Exclusion of Inferred Company Information

Another variation excluded the inferred company information while retaining text, tokenized usernames, and date-time embeddings. This experiment helped assess the importance of company-related context in the feature set and how its absence impacted model performance.

4 Results

Our experiments yielded various performance metrics across different models and approaches. Here's a summary of the key results:

4.1 BERT Embeddings Model

We initially started with a baseline BERT model that relied solely on tweet content to predict the number of likes, achieving a root mean square error (RMSE) of around 3.7k. While this score was high, it provided a useful benchmark to understand the predictive power of text alone in forecasting engagement.

To improve performance, we implemented several key changes. First, we applied a log-normal distribution to better model the skewed distribution of likes, which significantly reduced error. Additionally, incorporating mean token embeddings captured richer semantic information from the tweets, bringing the RMSE down to about 1.6k.

Further optimization came with the introduction of the Group Shuffle Split technique, which addressed potential biases from recurring usernames in our dataset. By ensuring that tweets from the same user did not cross the train/test boundaries, this method helped the model generalize better, reducing the RMSE to 1.5k.

For the task where the test data involved different companies, we used Group Shuffle Split, achieving the 1.5k RMSE. For the time-based task, we explored more complex models incorporating rich text and date embeddings while excluding the year as a feature. This approach yielded an R^2 score of 0.8. Additionally, when we used BERT embeddings and set the validation set as the latest 20% of data based on date, the model achieved an RMSE of around 3.1k.

These progressive refinements highlight the importance of tailoring both data preprocessing and model architecture to better handle real-world social media data.

4.2 Results of Other Models Trained

Approach	RMSE	MSE	MAE	R2	MAPE	MdAPE	Accuracy
Omitted Company Info (FC Neural Network)	1.7k	2706654.38	411.86	0.3899	inf%	57.96%	N/A
Omitted Username Embedding (FC Neural Network)	1.7k	N/A	N/A	N/A	N/A	N/A	N/A
Combined Embeddings (FC Neural Network)	1.6k	2776206.53	417.56	0.3742	inf%	59.54%	N/A
XGBoost (Non-text, Non-image)	5065	N/A	N/A	N/A	N/A	N/A	N/A
CLIP + Random Forest	N/A	N/A	N/A	N/A	N/A	N/A	Train: 0.9129, Val: 0.4348
BLIP Embeddings	4.1k	N/A	N/A	N/A	N/A	N/A	N/A
BERT with Text & Image Embeddings	691.85	N/A	N/A	N/A	N/A	N/A	N/A
BERTweet Classification	N/A	N/A	N/A	N/A	N/A	N/A	95%

Table 1: Performance metrics for different model approaches