Adobe Behaviour Simulation Challenge

Abstract

Simulating User Engagement on Social Media with Feature Extractors, Boosting Models, and LLMs This work tackles the dual challenge of behavior simulation (predicting user engagement on existing content) by leveraging a powerful combina- tion of multi modal feature extractors, gradient boosting regressors, FAISS, image captioning models and large language models (LLMs).

Keywords: Social media marketing, user engagement, behavior simulation, content simulation, feature extraction, FAISS, boosting models, LLMs

1 Introduction

We leveraged a robust dataset from Twitter enterprise accounts, undertaking exploratory analysis and data cleaning. Processing text and images using pretrained models, we created a multi-modal dataset. An ensemble of regressors predicted user engagement.

2 Exploratory Data Analysis

Analyzing engagement in a large tweet corpus reveals varied patterns. While the average tweet receives 773 likes, a significant standard deviation of 4931 indicates a long-tail distribution. Noteworthy usernames like CNN and EuroLeague, and businesses such as IndependentNGR and AMCTheatres, dominate. Inferred companies span news outlets (CNN, CBC), tech giants (Cisco), and financial institutions. Daily trends hint at a potential weekly engagement pattern, with peak likes on Sundays (889) and a drop on Thursdays (675).

3 Data Preprocessing

We conducted a comprehensive text cleaning process, removing punctuation, stop words, URLs, and correcting spelling and grammatical errors to enhance text quality.

Temporal features, including day-of-week, hour-of-day, and time since the last company tweet, were extracted to consider temporal factors in user engagement. To facilitate a nuanced analysis, the inferred company name was appended to each cleaned tweet, enabling brand-specific investigation into engagement patterns.

Step	Description
Text Cleaning	Remove noise, standardize formatting
Date-Time Features	Extract temporal attributes for analysis.
Company Name Inferred	Append inferred company names to cleaned tweets

Table 1: Data Preparation Steps

4 Task-1: Behavior Simulation

Given the content of a tweet (text, company, username, media URLs, timestamp), the task is to predict its user engagement, measured by likes.

4.1 Feature Engineering

We employ advanced techniques for text and visual embedding in the analysis of social media content. DistilBERT, a purpose-built pre-trained transformer model, is utilized to generate contextual representations of tweet text, capturing the intricate nuances of meaning and sentiment. Additionally, the US Encoder enhances text processing capabilities. For the visual component, two prominent image embedding models, EfficientNet and CLIP, are employed. EfficientNet excels in extracting high-level semantic features from images, while CLIP leverages multi-modal learning to bridge the semantic gap between textual and visual representations. The combined application of these models facilitates a comprehensive understanding of the visual content associated with each tweet.

We combined these features with other features like date-time and inferred company names.

4.2 Model Selection and Training

XGBoost, a robust gradient boosting model renowned for its adeptness in handling diverse features, was employed in this study to discern intricate interactions within the dataset. The model's hyperparameter tuning process was guided by the validation loss, ensuring optimal performance.

Parameter	Value
max_depth	8
learning_rate	0.0001
subsample	0.8
colsample_bytree	0.6
tree_method	gpu_hist
predictor	gpu_predictor_T4
random_state	42

Table 2: XGBoost Parameters

In addition to XGBoost, we incorporated FAISS, specializing in similarity searches and nearest neighbor tasks. Unlike traditional gradient boosting regressors, FAISS excels in information retrieval. We customized our approach, configuring indexes and potentially training on representative data, expanding our toolkit for optimized performance in specific tasks.

To further enhance predictive capabilities, we adopted an ensemble strategy. Rather than relying on a single model, the ensemble approach involved weighting predictions from each regressor based on their validation losses. This nuanced averaging aimed at improving overall prediction accuracy and generalization.

4.3 Results

Model	Validation RMSE on log(1+likes)
XGBoost 1	5.3876
FAISS	3.6929
XGBoost 2	8.3262

Table 3: RMSE on Validation for Model 1, Model 2, and Model 3.

Model 1, a fusion of CLIP with MPNet, achieved an RMSE of 5.3876 on validation, showcasing strong performance in capturing intricate textual and visual relationships. In contrast, Model 2, yielded an RMSE of 3.6929, leveraging the efficient scaling of FAISS for images and distilled knowledge from BERT for text. Combining EfficientNet and DistilBERT, Model 3 yeiled an RMSE of 8.3262. This RMSE was measured on the logarithm of likes.

The ensemble of Model 1 and Model 2 demonstrated robust results, blending their complementary strengths to capture diverse aspects of the data, contributing to overall model performance. Inference took less than 30 seconds on the given test set.

References

a. Efficientnet pytorch model zoo. https://pytorch.org/hub/nvidia_deeplearningexamples_efficientnet/.

a. b. Efficientnet tensorflow hub model. https://blog.tensorflow.org/2019/02/effective-tensorflow-20-best-practices.html.

Google ai blog. https://blog.research.google/.

Hugging face model hub. https://huggingface.co/models?language=ai.

Papers with code. https://paperswithcode.com/.

Askaydevs. 2019. Distilbert github repository. https://github.com/askaydevs/distillbert-qa.

Authors. 2022a. Bit: Learning deep representations for image and text retrieval. *Google AI*.

Authors. 2022b. Bloom: Language models are multi-modal and many-modal. *BigScience*.

Authors. 2022c. Lora: Low-rank optimization for efficiently encoding large language models. *Google AI*.

Authors. 2022d. Point-efficient transformers. *Google AI*.

beta-eto code. 2022. Blip2 github repository. https://github.com/beta-eto-code/bx.model.

bigscience workshop. 2022. Bloom 7b github repository. https://github.com/bigscience-workshop/ model_card.

dmlc. Xgboost github repository. https://github.com/dmlc/xgboost.

Hugging Face. 2023a. Blip2 hugging face model card. https://huggingface.co/docs/transformers/main/model_doc/bit.

Hugging Face. 2023b. Distilbert hugging face model card. https://huggingface.co/docs/transformers/model_doc/distilbert.

microsoft. Lightgbm github repository. https://github.com/microsoft/LightGBM.

Microsoft. Lora github repository. https://github.com/microsoft/LoRA.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert for efficient inference. *Google AI*.

A References

- DistilBert:
  - Paper: (Sanh et al., 2019)
  - GitHub repository: (Askaydevs, 2019)
  - Hugging Face model card: (Face, 2023b)
- BLIP2:
  - Paper: (Authors, 2022a)
  - GitHub repository: (beta-eto code, 2022)
  - Hugging Face model card: (Face, 2023a)
- CLIP:
  - Paper: (**?**)
  - GitHub repository: (**?**)
  - Hugging Face model card: (**?**)

EfficientNet:

- Paper: (Tan and Le, 2020)
- TensorFlow Hub model: (eff, b)
- PyTorch model zoo: (eff, a)

A References

- DistilBert:
  - Paper: (Sanh et al., 2019)
  - GitHub repository: (Askaydevs, 2019)
  - Hugging Face model card: (Face, 2023b)
- BLIP2:
  - Paper: (Authors, 2022a)
  - GitHub repository: (beta-eto code, 2022)
  - Hugging Face model card: (Face, 2023a)
- CLIP:
  - Paper: (**?**)
  - GitHub repository: (**?**)
  - Hugging Face model card: (**?**)

EfficientNet:

- Paper: (Tan and Le, 2020)
- TensorFlow Hub model: (eff, b)