

DTSA 5511 Project

May 31, 2024



LMSYS - Chatbot Arena Human Preference Predictions

The Kaggle logo, featuring the word "kaggle" in a light blue, lowercase, sans-serif font.The LMSYS Org logo, featuring the text "LMSYS Org" in a white, uppercase, sans-serif font on a dark background.

The Large Model Systems Organization develops large models and systems that are open, accessible, and scalable. It is currently run by students and faculty members from UC Berkeley Sky Lab.

Details for this dataset: <https://www.kaggle.com/competitions/lmsys-chatbot-arena/data>



University of Colorado **Boulder**



Overview

The goal for this competition is to create a model that could accurately classify responses as preferred by human users using data from a Chatbot Arena competition.

Objectives:

- Improve the accuracy of predicting user preferences.
- Enhance the interaction quality in chatbot systems.
- Leverage advanced machine learning techniques for better performance.



Problem Statement

The project addresses the challenge of predicting which chatbot response a user is likely to prefer. Accurate prediction can significantly improve user satisfaction and engagement with chatbot systems.

Predicting user preferences is crucial for refining chatbot interactions, making them more intuitive and user-friendly. It helps in tailoring responses that align closely with user expectations.



Data Overview

- Training Dataset: 57,477 rows
- Test Dataset: Approximately 25,000 rows
- This indicates a moderately large dataset, which should allow for robust model training and validation.



Data Structure

Each row in the dataset represents a user interaction. The columns include:

- **id**: A unique identifier for each interaction.
- **model[a/b]**: Identifiers for the two models involved in the interaction.
- **prompt**: The input prompt given to both models.
- **response[a/b]**: The responses generated by model_a and model_b respectively.
- **winnermodel[a/b/tie]**: Indicates which model's response was chosen as the winner by the judge. This column serves as the target variable for prediction.

Target Variable

- The target variable (**winnermodel[a/b/tie]**) indicating whether model_a, model_b, or a tie was selected as the winner. Predicting this target variable correctly is the objective.



Sample Chat

Model A - gpt-4-0314

Given 4 floats representing the elements of a quaternion, how can I identify which is the scalar component without knowing the convention beforehand?

It's impossible to definitively identify the scalar component of a quaternion without knowing the convention beforehand. There are two common conventions for representing quaternions: scalar-first (w, x, y, z) and scalar-last (x, y, z, w).

However, you can try to make an educated guess by analyzing the values of the elements. If one of the elements is significantly different from the others in magnitude, it might be the scalar component. However, this method is not foolproof and can lead to incorrect assumptions.

The best approach is to know the convention used for the quaternion beforehand or get this information from the source providing the quaternion values.

Model B - codellama-34b-instruct

Given 4 floats representing the elements of a quaternion, how can I identify which is the scalar component without knowing the convention beforehand?

There is no definitive way to identify the scalar component of a quaternion without knowing the convention used to represent it. The scalar component is often represented by the first element of the quaternion, but this is not always the case.

In general, the quaternion representation of a 3D rotation can be written as:

$$q = w + xi + yj + zk$$

where w is the scalar component, x , y , and z are the imaginary components, and i , j , and k are the imaginary units that satisfy the following relations:

$$i^2 = j^2 = k^2 = -1$$

The convention used to represent the quaternion can vary depending on the application and the person implementing it. Some common conventions include:

- w, x, y, z : This is the most common convention, where the scalar component is represented by the first element of the quaternion.
- x, y, z, w : In this convention, the scalar component is represented by the last element of the quaternion.
- $-w, x, y, z$: This convention is similar to the previous one, but the sign of the scalar component is flipped.
- z, y, x, w : This convention is used in some graphics applications, where the scalar component is represented by the first element of the quaternion and the order of the imaginary components is reversed.

Without knowing the convention used to represent the quaternion, it is not possible to determine which element is the scalar component. However, if you know the context in which the quaternion is being used, you may be able to infer the convention based on the expected behavior of the quaternion. For example, if the quaternion is being used to represent a 3D rotation, you would expect the scalar component to be positive and the imaginary components to be negative.

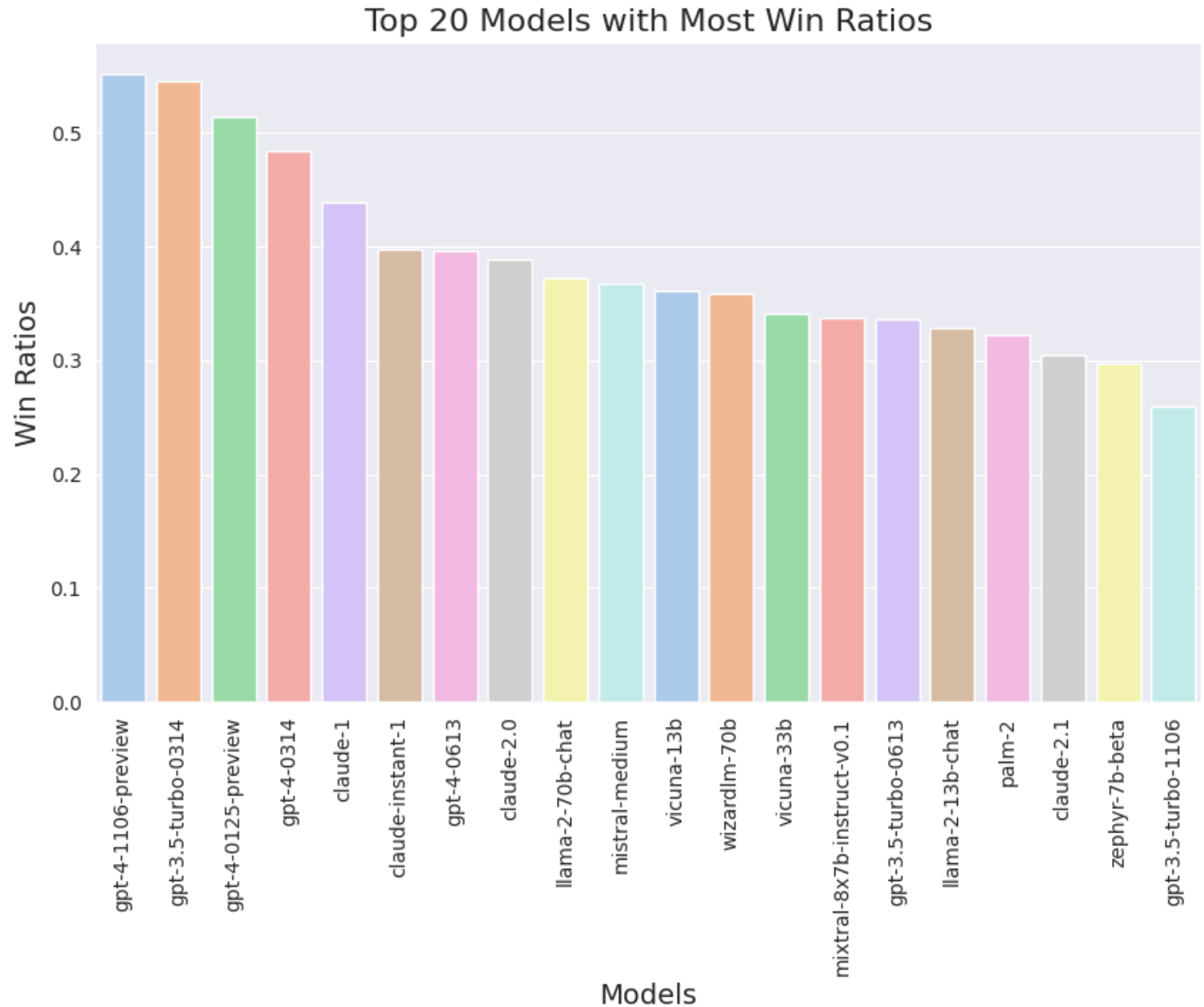
Tie!



Top 20 Models

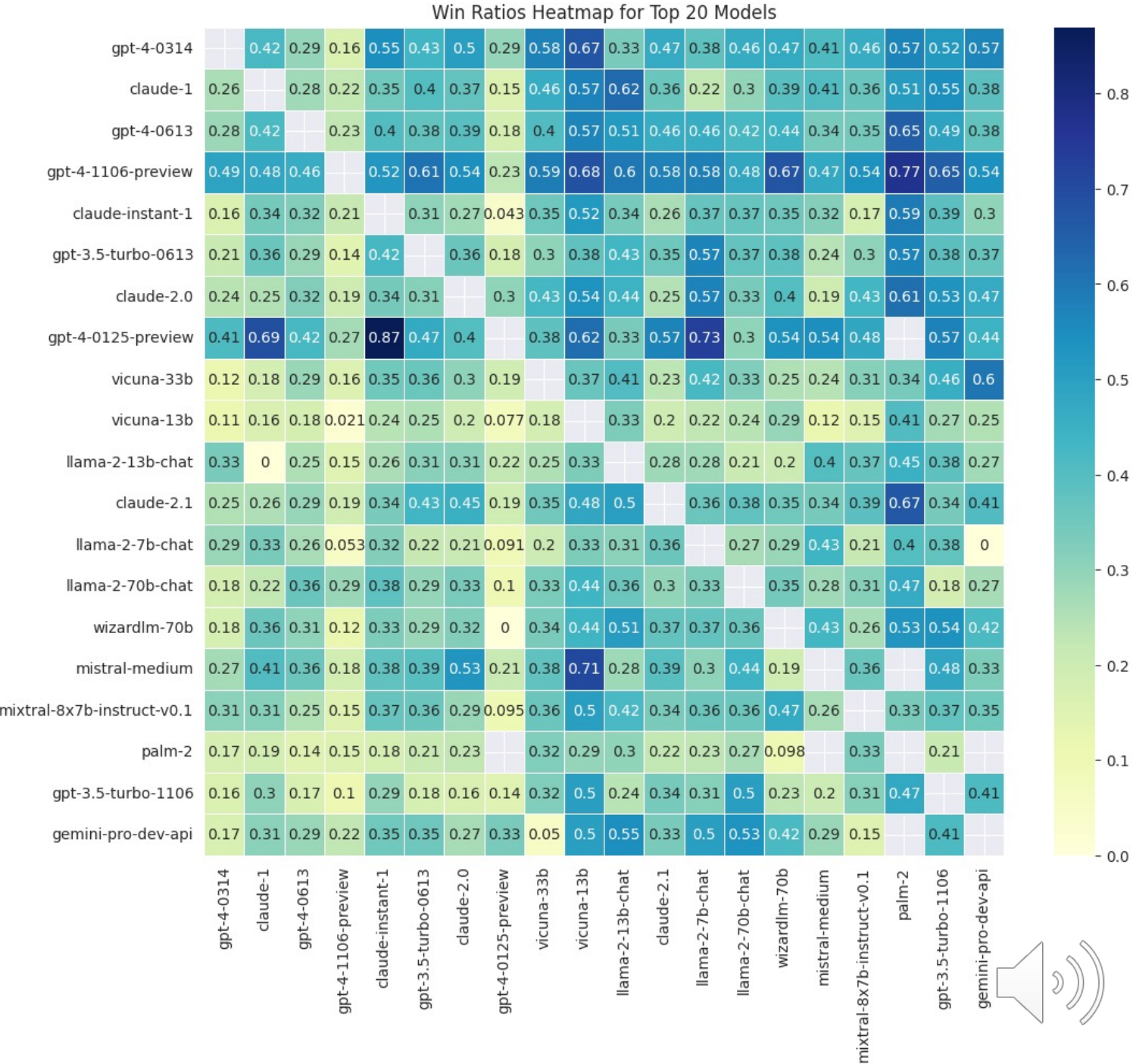
Three GPT models have win ratios exceeding 0.5.

However, this can be misleading because we don't know the strength of their opponents.



Heatmap

- GPT-4 Models' Dominance
- Consistent Performance
- Mixed Results
- Head-to-Head Analysis



Models

DeBERTa-v3

Details: <https://huggingface.co/microsoft/deberta-v3-xsmall>

- Disentangled Attention Mechanism
- Enhanced Mask Decoder
- Training Efficiency



Training Configuration

A100 GPU 40GB GPU Memory

Model Name: microsoft/deberta-v3-xsmall

Number of Labels: 3

Learning Rate: $1e-5$

Weight Decay: 0.01

Warmup Steps: 10% of the total training steps

Training Epochs: 4

Train Batch Size: 8

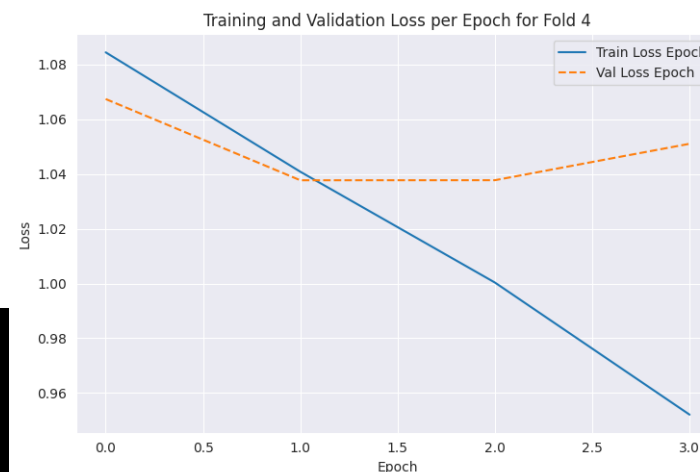
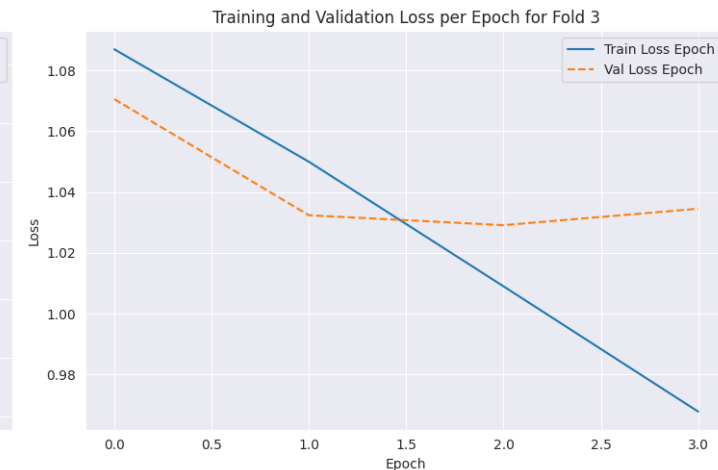
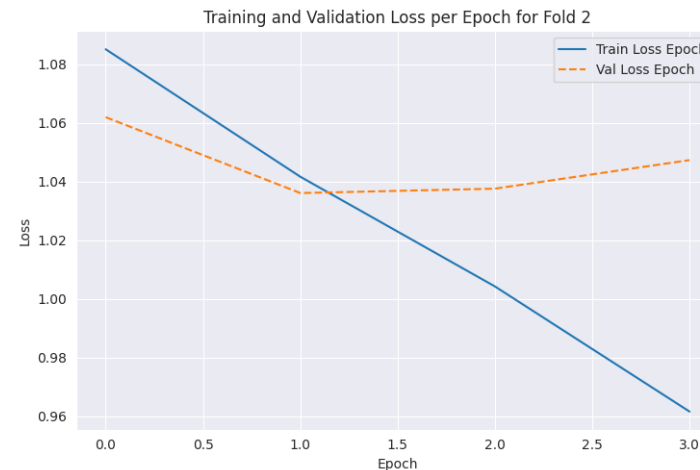
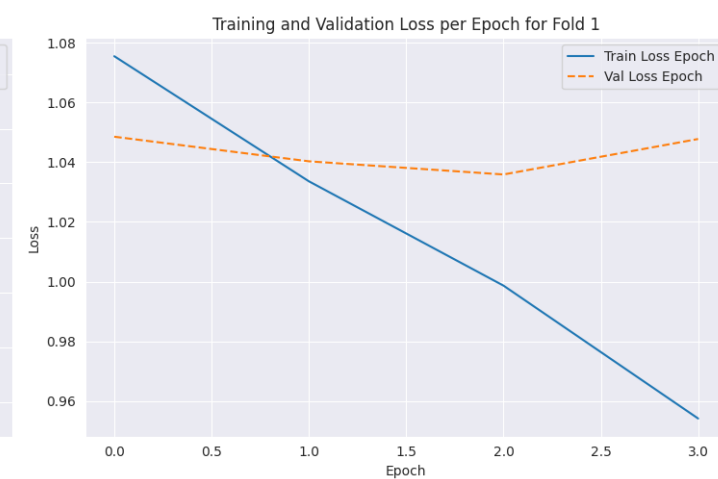
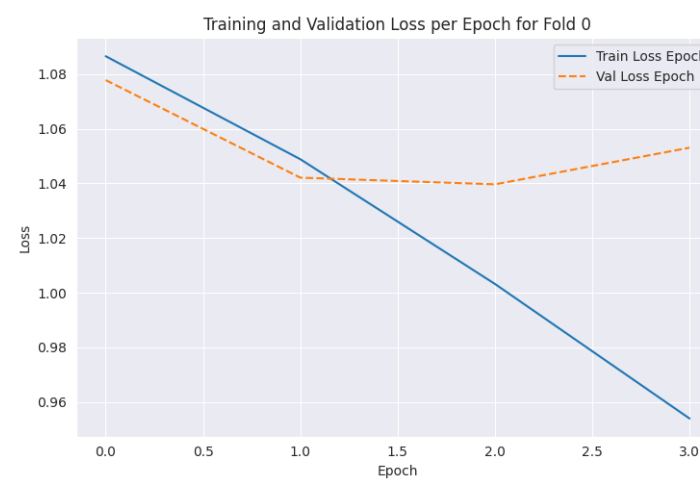
Evaluation Batch Size: 4

Fold: 5



Model Performance

- Time for training ~ 10 hours
- Effective learning during training
- Validation loss did not decrease as significantly
- Room for improvement



Conclusion & Final thoughts

- Successfully implemented a transformer-based model to predict user preferences between chatbot responses.
- Promising results, particularly in training performance.
- Validation results highlighted areas for improvement.
- Addressing these through:
 1. Hyperparameter tuning
 2. Regularization techniques
 3. Data augmentation strategies
 4. Exploring more advanced architectures
- Enhancing the model's performance will lead to more accurate predictions in real-world applications, ultimately improving the user experience with chatbot systems.



Thank You!

