Data 200s Final Project
Austin Jeffrey Ly , Joyce Yu , Sam Alexandre McCarthy-Potter
4/28/2025
Link to Presentation: https://youtu.be/I-eruWQShBg

Natural Language Processing Final Draft

1.0 Introduction

With the advancements in Natural Language Processing (NLP) different Large Language Models (LLM) have increasingly become critical to user engagement and the quality of those engagements. Evaluating these models can become a challenge. Recent research has shown that LLM outputs may not always align with human preferences. In the paper “Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences”, the research team introduces a mixed-initiative system that assists users in generation evaluation criteria and implementing assertions for LLM outputs (Shanker, et al 2024). This allows outputs to satisfy human requirements. Additionally, the study “LLM Evaluators Recognize and Favor their own generations” investigates the phenomenon of self-preference in LLM. Researchers were able to demonstrate that LLMs like GPT-4 and LLaMA 2 exhibit a bias towards their own outputs (Panickssey, et al, 2024). Our project aims to build upon these developments and provide additional insights. This project utilizes data from Chatbox Arena, where hyman users engage in pairwise comparisons of responses of different models. We will provide a dynamic assessment of model performance.

This project aims to evaluate these models by performing 2 task

a. Modeling the winning model
b. Hardness prediction

Modeling the winning model will be explored by utilizing logistic regression models (See section 3.1). Hardness prediction will be performed by utilizing techniques such as k means clustering and linear regression modeling (see section 3.2)

By integrating topic modeling, hardness score, and embedded-based features, we aim to explore the influences that contribute to a model’s perceived performance. This approach will provide an additional perspective to existing research.

2.0 Description of data

The Exploratory Data Analysis (EDA) is based on two main datasets from the Chatbot Arena initiative.

2.1 Main Dataset: Conversion Comparisons

The primary dataset contains 25,252 conversation entries. The granularity includes each row being a conversation entry between two models. The data includes:

- question_id : unique prompt identifier
- model_am model_b: model names
- conversation_a , conversation_b : model responses
- Winner: winning model selected by judge
- judge : judge ID

2.1.1 Dataframe characteristics and cleaning

- Exploring the prompt length shows:

Figure 1 - Prompt Length description

```
count    12500.000000
mean       199.510960
std        375.692894
min         16.000000
25%         42.000000
50%         72.000000
75%        158.000000
max       2560.000000
Name: prompt_length, dtype: float64
```

- Distribution of prompt length:

Shows that the mean of the prompt is about 200 characters, while the median is 72 characters. This suggests that the distribution is right-skewed.

- Conversation a and b distribution

Figure 2 - Distribution of conversation a

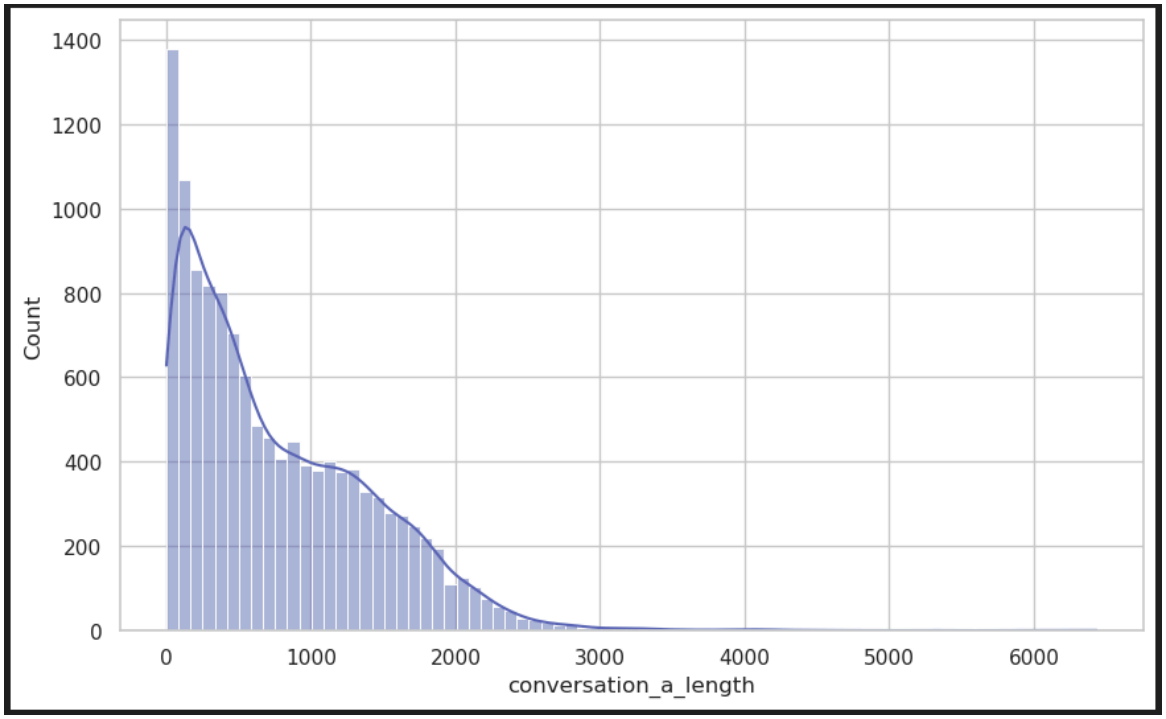
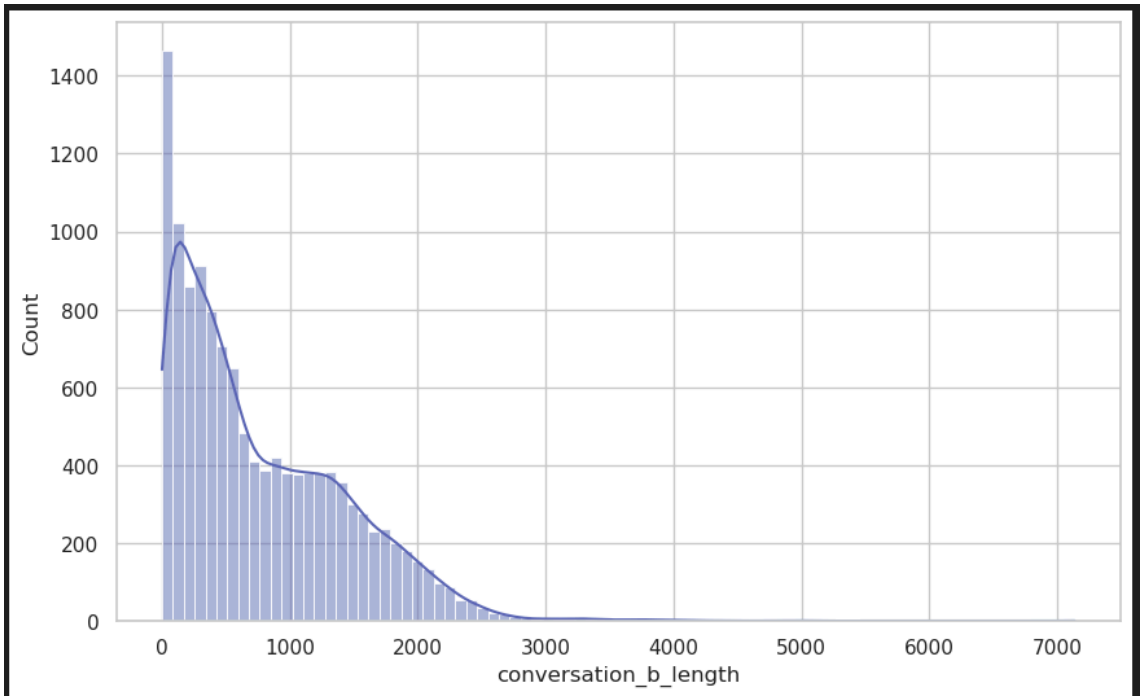


Figure 3 - Distribution of conversation b



2.2 Embedded Data

We also included pre-computed embedding vectors for prompts and responses. They are utilized to evaluate similarity between responses and user preferences.

2.2.1 K Means

Utilizing the k means algorithm, we clustered the embedded data to look for similarities between topics within the dataset.

Looking at the responses utilizing 50 clusters we can compare model a and b responses.

Figure 4 - Model a response clusters

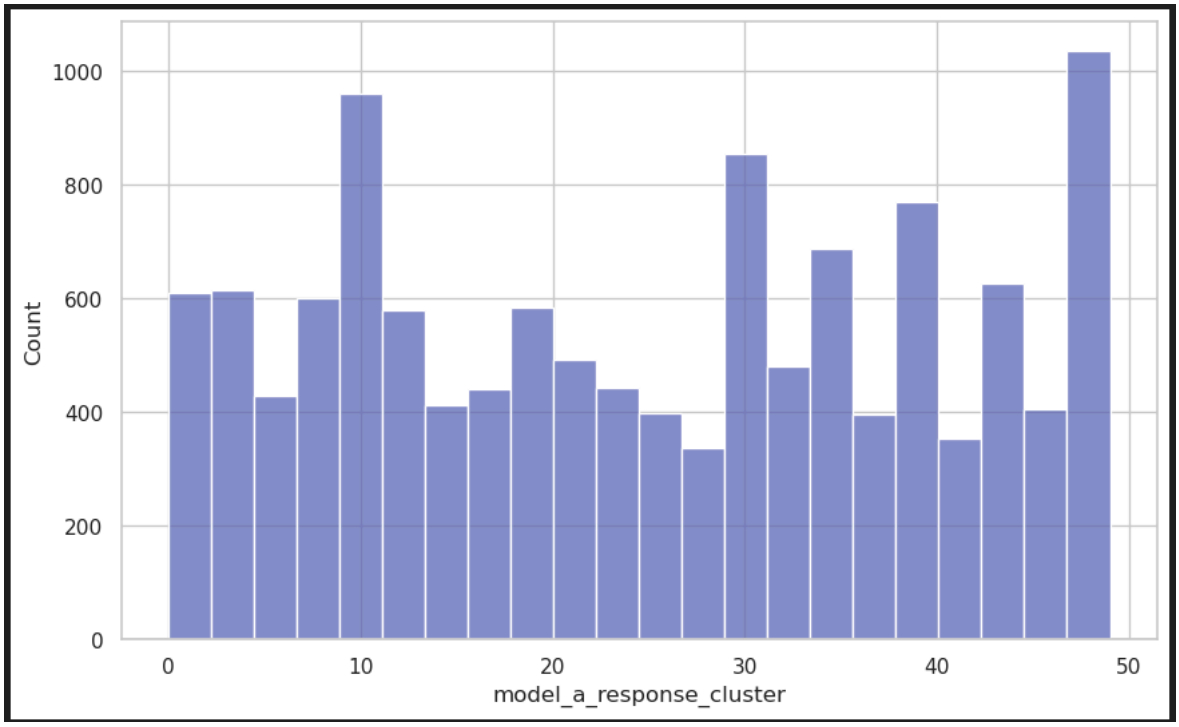
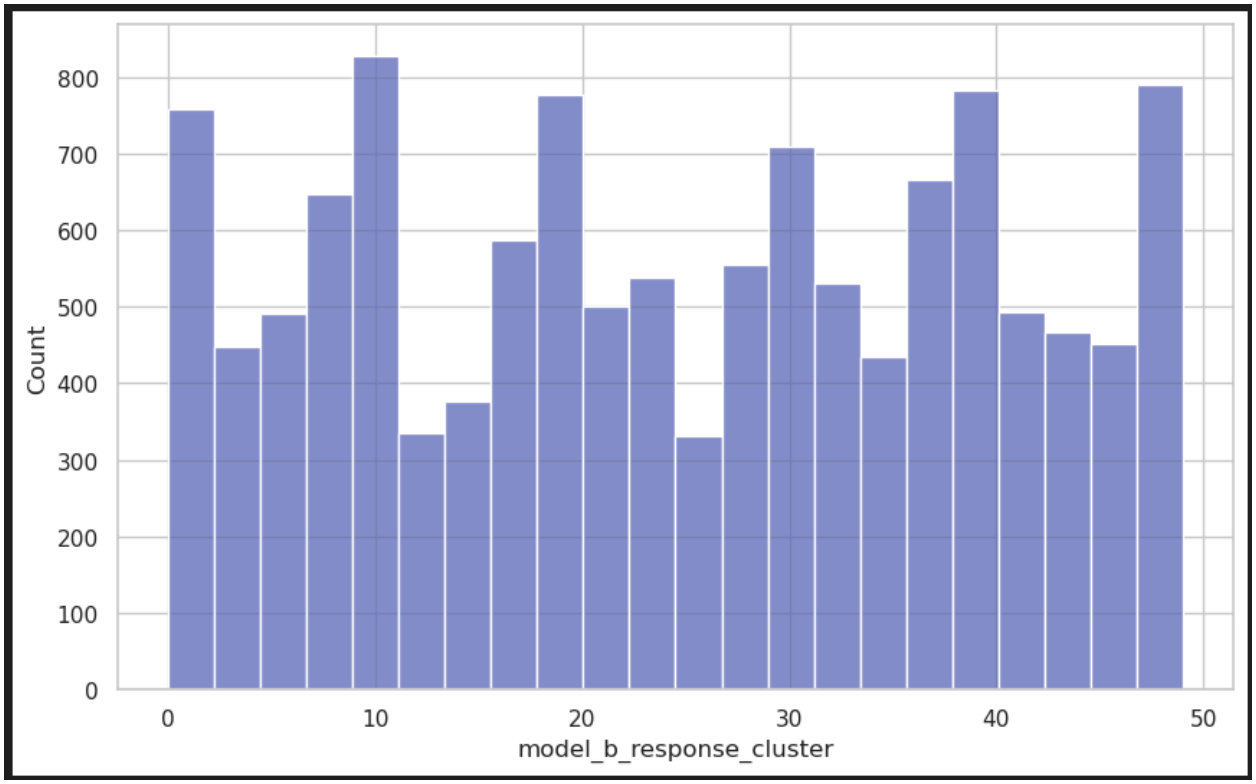


Figure 5 - Model b response clusters



Comparing the two responses. We can see that the responses differ between a model a and a model b indicating a deviation in output between the two through visual inspection.

2.3 Topic Modeling and Hardness Score Data

This data is utilized to determine the complexity of the prompts.

High values (e.g., 8–10) correspond to prompts that require deep reasoning, precise knowledge, or multi-step problem solving.

Low values (e.g., 1–3) correspond to more simple or factual queries that require minimal inference.

2.3.1 Data Cleaning

To clean the data:

1. We dropped rows with missing values entirely, as they represented a negligible portion of the dataset.

2. We wrote a custom function to extract scalar values from improperly formatted list entries.

3. Cleaned score columns (score_value_1, score_value_2, score_value_3) were then used to compute a row_average score representing overall prompt hardness.

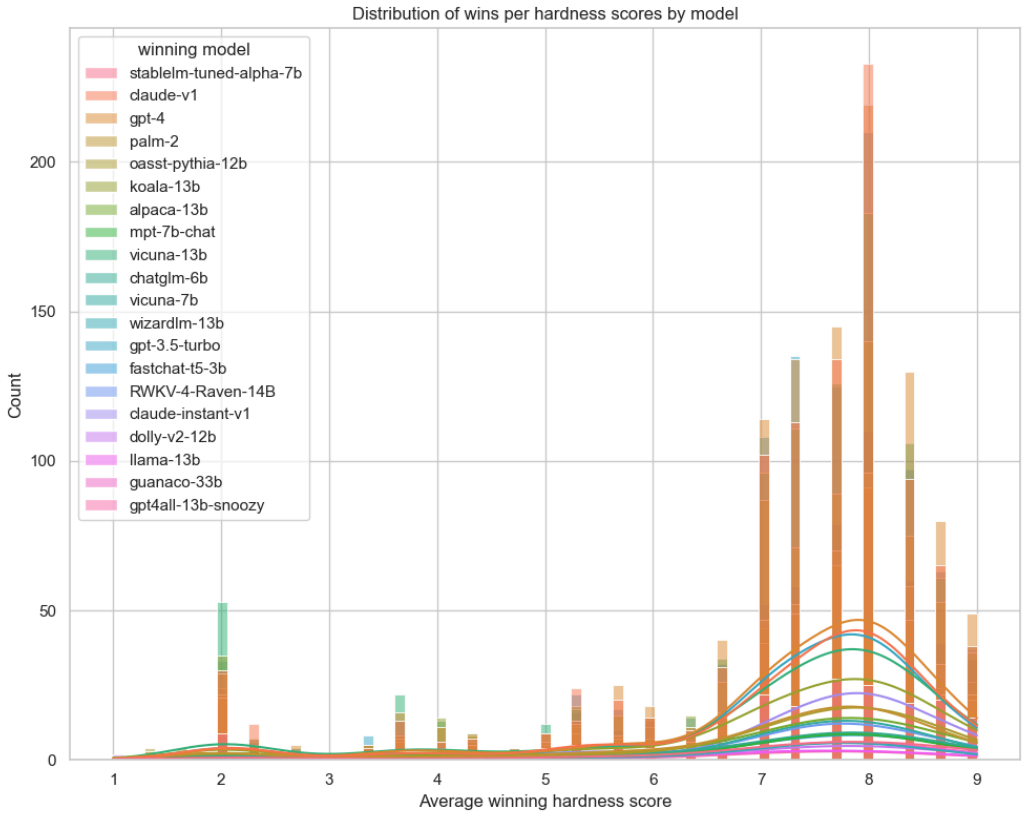
This ensured that all scores were numeric and consistently formatted for downstream analysis.

2.3.2 Analysis of Hardness scores

Once cleaned, we investigated whether prompt hardness scores had any influence on user preferences.

1. We calculated the average winning hardness scores for responses grouped by the winner label and chatbot name. See figure 6.

Figure 6 - Score Values by winners



2. The average hardness score was fairly consistent across winners.

This suggests that factors other than prompt complexity (e.g., style, verbosity, topic familiarity) may drive user preferences more than raw prompt difficulty.

2.4 Feature Engineering

To support downstream analysis and modeling tasks, several new features were engineered from the original Chatbot Arena dataset and the auxiliary hardness score data. These features capture model identity, prompt difficulty, and model performance patterns in structured formats.

2.4.1 Winning Model Assignment

Each prompt in Chatbot Arena includes a winner label indicating whether model_a, model_b, or neither was preferred. To simplify modeling, a new column called winning model was created. Each entry of the winning model column included 1 of the 19 chatbox names or “tie”.

2.4.2 One-Hot Encoding of Winning Models

To incorporate model identity into any supervised models later, the winning model column was one-hot encoded. This transformation allowed each model to be represented as a separate binary column. This is helpful for training classifiers or regressors that rely on categories.

2.4.3 Model visualization and summary

Several graphs were generated to show the relationships between models, their win counts, and the types of prompts they excelled at:

- Boxplots and violin plots showed the distribution of average winning hardness scores per model. While most models had similar central tendencies, some models showed tighter distributions, suggesting consistency in performance across prompt difficulty levels.

- Bar plots highlighted model performance in terms of:

    - Win frequency: how often a model was selected as a winner.
    - Participation frequency: how often a model appeared in any matchup, regardless of outcome.
    - Win ratio: How often a chatbot won vs. the number of times they participated in conversation.
- Histograms of average winning hardness score per chatbot. This showed how well each chatbot performed for each hardness score of a question, and on average, which hardness score it performed best at represented by a vertical black or red line where the mean lied on the histogram. This line was colored black if its overall mean was >6.9, and red otherwise. Two models appear to have a red line, indicating that its distribution’s overall mean is less than 6.9. These two models, Llama13-b and dolly-v2-12b, have an overall winning hardness score that is less than the other models on average. This suggests that when the two models win, they are winning in conversations where the hardness score of a question is not high.

These plots revealed that models like GPT-4, Claude-v1, and GPT-3.5-Turbo had the highest number of wins and participated frequently in comparisons. However, participation alone did not guarantee high win rates, as seen in models like Vicuna-13b, which had high participation but moderate win frequency.

Figure 7 - Distribution of average winning hardness score per model and outliers, boxplot

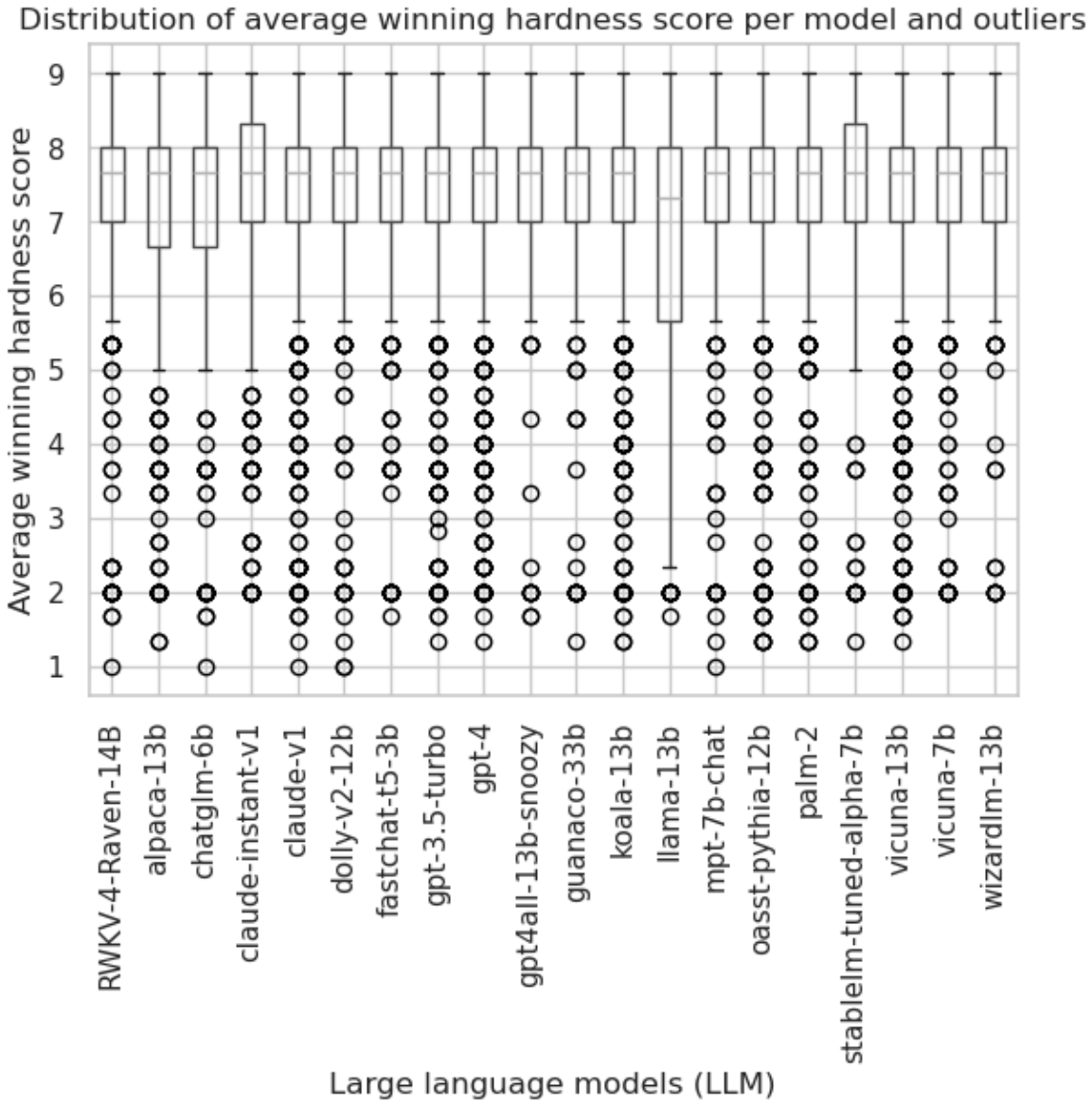


Figure 8 - Win ratio (frequency of wins / frequency of participation) per LLM

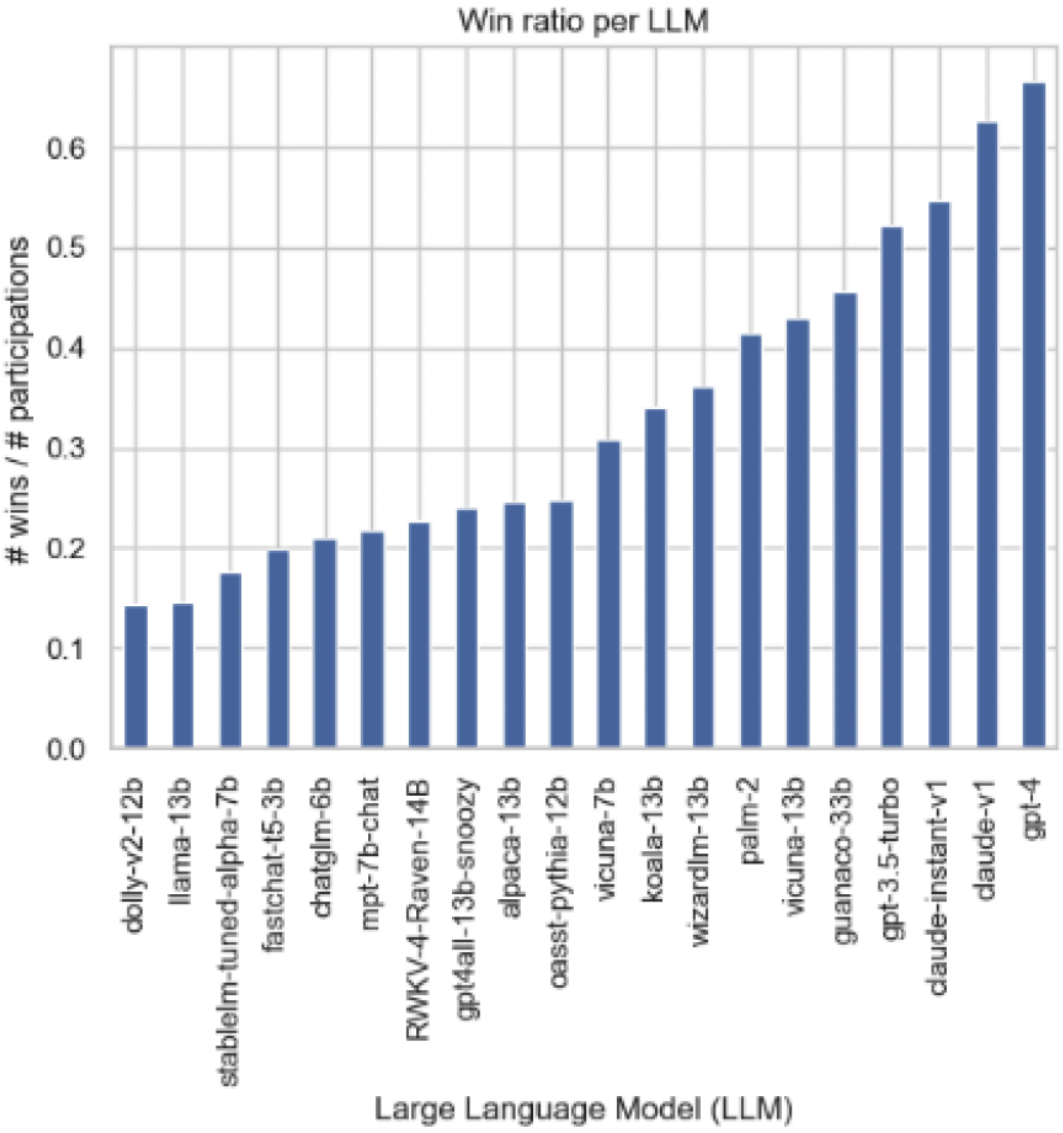
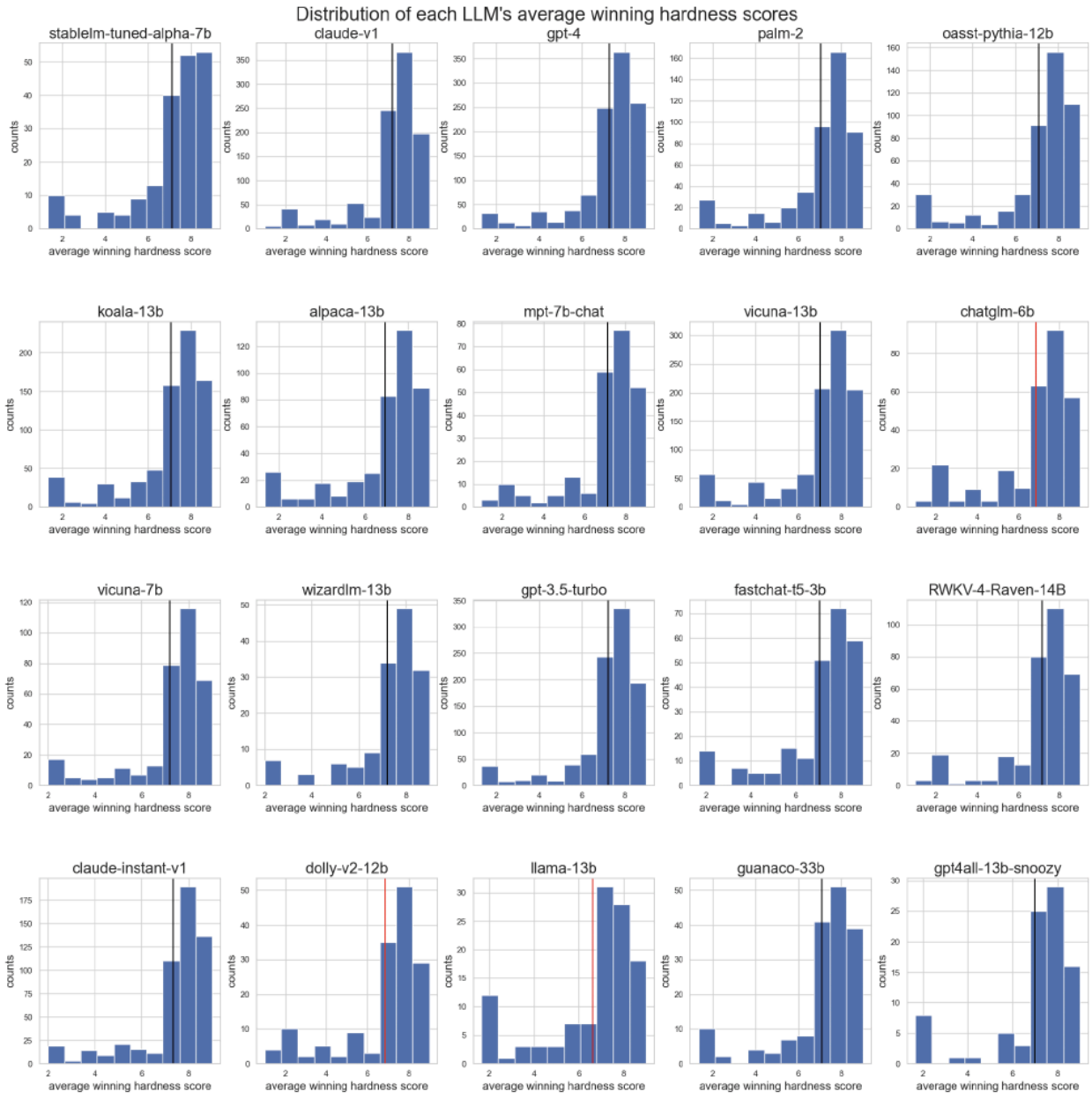


Figure 9 - Histogram showing frequency of wins per hardness score per chatbot.



Through this feature engineering process, the dataset is transformed into a structure that can be used for our models. This will help support and model by quantifying values. The feature engineering will be used to create interpretable modeling pipelines for bothe classification and regression tasks.

3.0 Methodology

3.1 Task A

For Task A, first we trained a logistic regression model using both the embedded prompts and embedded responses as input features, in addition to the one hot encoded data for the models occupying model a and model b. We applied PCA to reduce the dimensionality to 500 components. When then trained the model to classify between only two outcomes – model A or model B – the model achieved a cross-validated accuracy of 65%

Unfortunately this model only had a cross-validated accuracy of 44% when trying to predict the two additional outcomes —-Ties Ties(both bad). To Try to correct for this drop of accuracy a lazy classifier was constructed to train a random forest regressor for each of 20 K-means clusters found in the prompt embedding. Each model would undergo its own PCA based on the size of the available data cluster data set. Once cross validated this only had an accuracy of 46%. The f1 one score for tie is .05 and the f1 score for tie(both bad) is 0.25. These f1 scores led me to the conclusion that the model was predominantly picking model a or model b instead of randomly misclassifying the ties. We think this due the Ties being less prominent in the data set the model was over fitted to choose model A or model B.

Additional features were added – the model identification for model a and model b. The number of clusters were changed. The model was tested without the ties as outliers and the model was tested to just determine the difference between ties and outliers. Due to the failure of K-means clustering we are attempting to first cluster the result of the model by if there is likely a victor or if there is likely a tie then training additional models to determine if the sub-set of Tie or victor was most correct.

However, when we expanded the possible responses to include model A, model B, Tie, and Tie (both bad), the cross-validated accuracy dropped to 45%. To improve performance, we attempted encoding the model_a and model_b labels, but this had no significant impact on accuracy.

To further enhance model performance, we attended to cluster the data using K-means on the prompt embeddings and then trained individual models for each cluster. This failed to increase the accuracy. The final solution we decided on was to calculate the ELO scores for each model in the training set calculating how likely a model was to win an argument. We then added the ELO scores for the two models fighting into the logistic model features and this increased the accuracy into the acceptable range.

Additionally, we incorporated the estimated hardness from Task B as an additional feature to further fine-tune the model.

3.2 Task B

For Task B, we trained 3 different linear regression models to help determine the hardness score of a prompt since the scores are a quantitative variable that ranged from 1 to 10 on a continuous scale. To simplify training purposes, we averaged the hardness scores of each question and rounded them to the nearest integer. We also engineered features from the prompt embeddings, response embeddings, topic models, and hardness scores of the existing datasets using either PCA or K-means clustering for training.

The first linear regression model was trained on the top 500 most frequent topics found in the training set. These top 500 topics were one-hot encoded. To reduce the number of dimensions in our design matrix, we used PCA on the design matrix and used the number of components that captured 95% variability in our test sets.

*Figure 10 - Explained variance by PCA for the **first linear regression model**.*

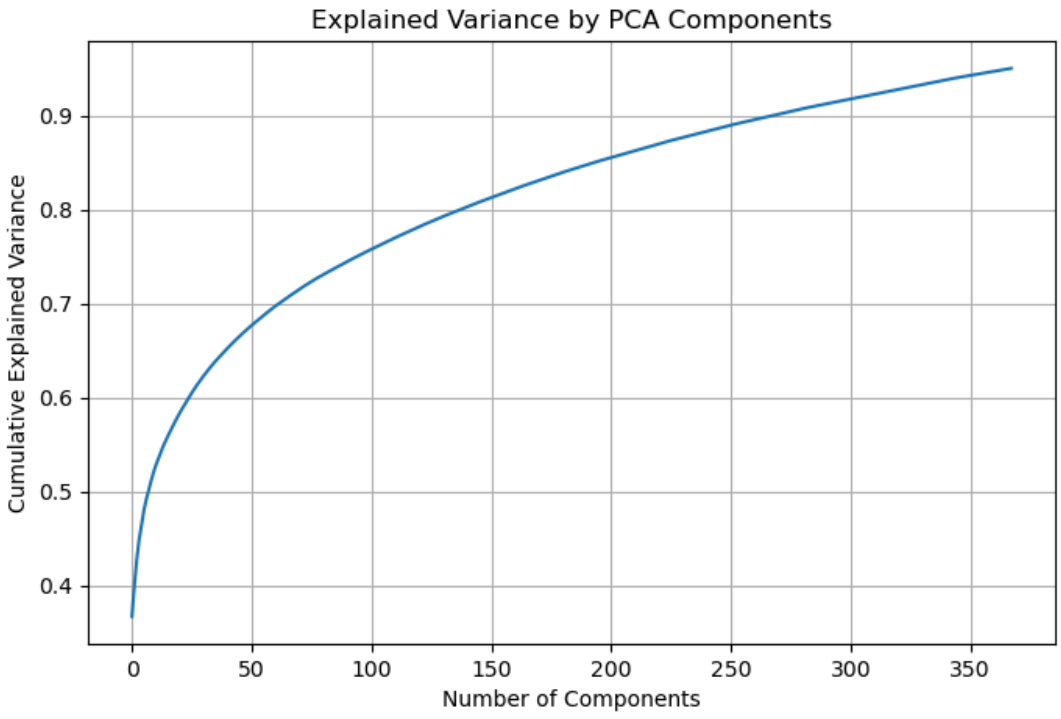


*Table 1 - Explained variance by PCA for the **first linear regression model**. Please note that the zeros under the "95% Explained Variability" are not actually zero, but a significantly small number rounded for visual purposes.*

Component	95% Explained Variability
0	0.37
1	0.03
2	0.03
3	0.02
4	0.02
...	...
363	0.00
364	0.00
365	0.00
366	0.00
367	0.00

The second linear regression model was trained on the top 500 most frequent topics found in the training set and prompt embeddings. These top 500 topics were one-hot encoded. To reduce the number of dimensions in our design matrix, we used PCA on the design matrix and used the number of components that captured 95% variability in our test sets.

*Figure 11 - Explained variance per component by PCA for the **second linear regression model**.*

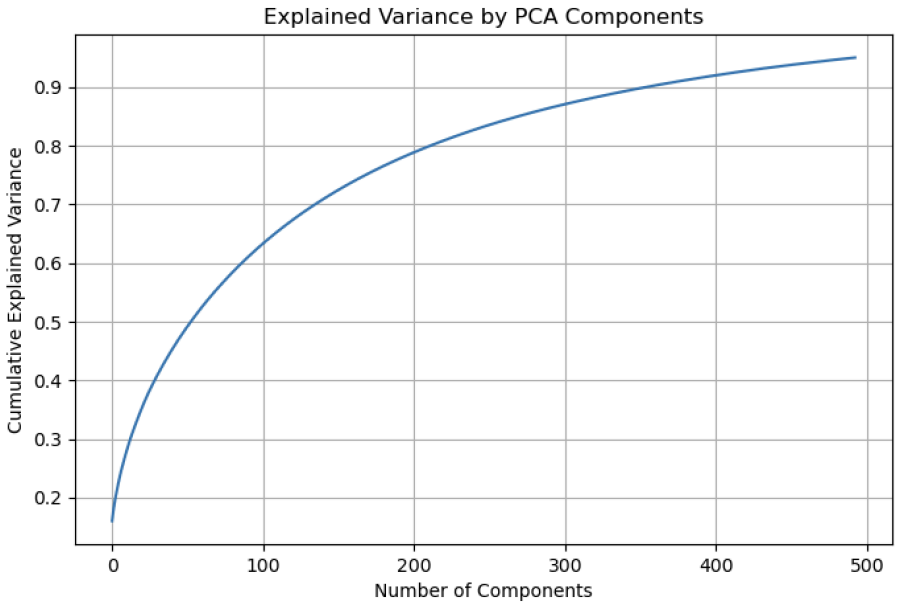


*Table 2 - Explained variance per component by PCA for the **second linear regression model**. Please note that the zeros under the "95% Explained Variability" column are not actually zero, but a significantly small number rounded for visual purposes.*

Component	95% Explained Variability
0	0.16
1	0.02
2	0.02
3	0.01
4	0.01
...	...
488	0.00
489	0.00
490	0.00
491	0.00
492	0.00

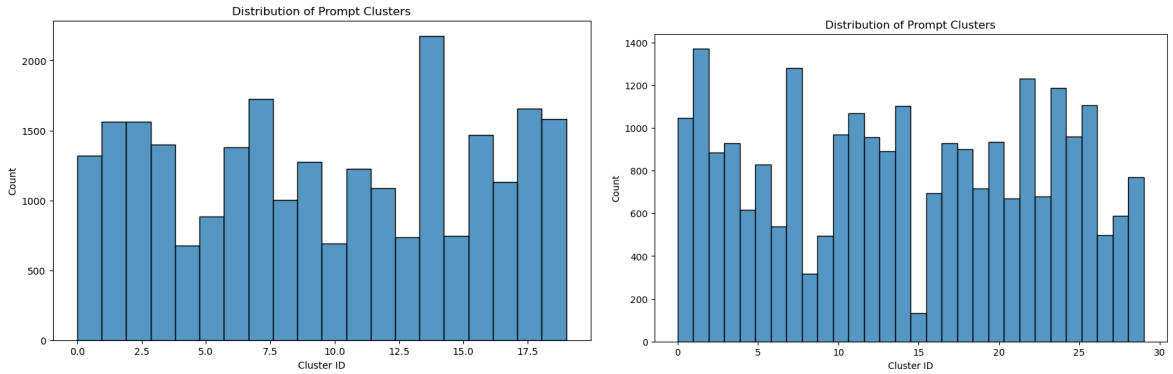
The third (and best) linear regression model was trained on the top 10 (instead of 500) most frequent topics found in the training set, prompt embeddings, and clusters obtained from K-means clustering performed with k = 50 cluster on each of the embedded data sets (prompts, response_a, and response_b). K-means clustering and the reduction on the number of one-hot encoded topics helped to generalize our model while still maintaining similar performance to our second linear regression model.

To evaluate our models, we calculated the absolute difference between each avg score and the avg predicted score. Along with seeing the absolute differences in each avg score, we evaluated our linear models with mean squared error (mse) and r^2 to explain how well our models explain the variability in the datasets and unseen data. To ensure a comprehensive evaluation, we measured the mse 10 times by splitting our data 80/20 10 different ways using k-fold cross validation to ensure our initial mse evaluation metric was not due to randomness of initial 80/20 splits in our data.

4.0 Summary of results

For task A, 20 K-means clusters were chosen for a relatively equal distribution of test data to prevent overfitting. The K-mean clustering 30 clusters below shows how the addition of extra clusters can lead to imbalances in cluster size. At the extreme end some small clusters appear with almost no data points.

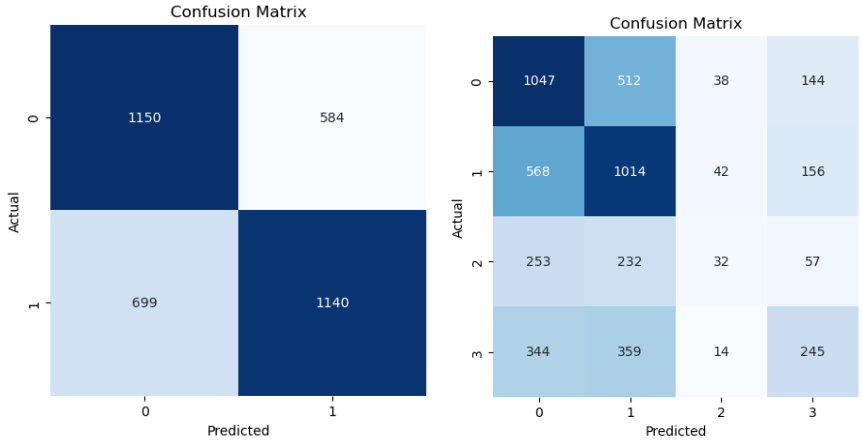
Figure 12 - K-means clustering of embedded prompts



The Confusion matrix for differentiating between model A or model B being the winner showed great promise, well surpassing the max point threshold for task A. Unfortunately the model was unable to predict both ties and model victory while maintaining accuracy. It appears that while the model can predict the winner with high accuracy when the possibilities of ties are introduced the model has difficulty finding clear differentiation. This problem with the ties is exacerbated by the ties being significantly less common but a large enough population that it cannot be ignored.

Figure 13 - Confusion matrix two classes vs four classes

(a) *(b)*



(a) The two classes are:

- **0 – Model A**
- **1 – Model B**

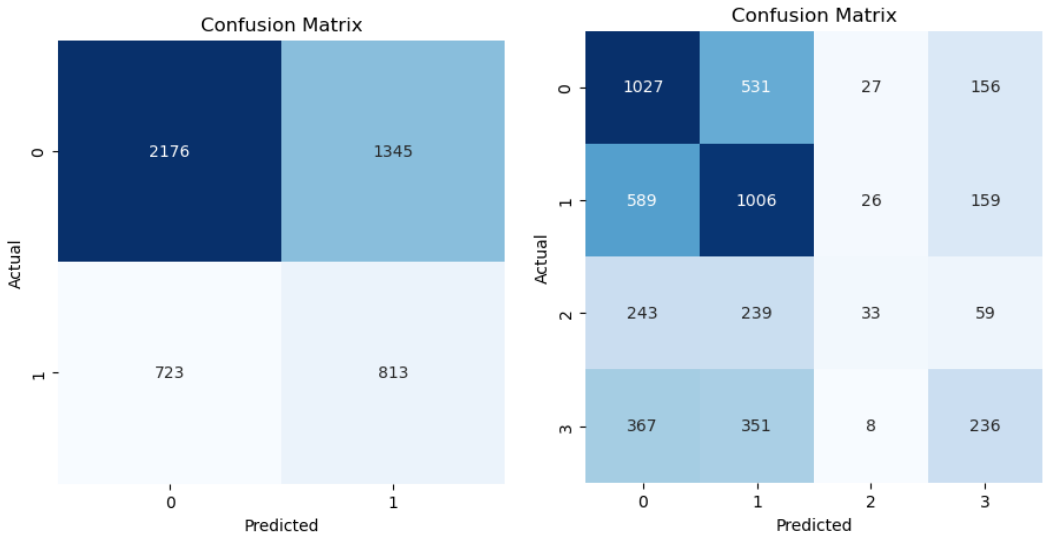
(b) The four classes are:

- **0 – Model A**
- **1 – Model B**
- **3 – Tie**
- **4 – Tie (both bad)**

As seen above in the confusion matrix, the logistic regression becomes less accurate with the addition of ties and ties(both bad). Due there being extensive data for model A winning or model B winning all models that are trained on the entire data set are over fitted to the detriment of ties. In an attempt to remedy this a new logistic model was trained showing if there was a victor or a tie.

Figure 14 - Confusion matrix Victor vs Tie

(a) *(b)*



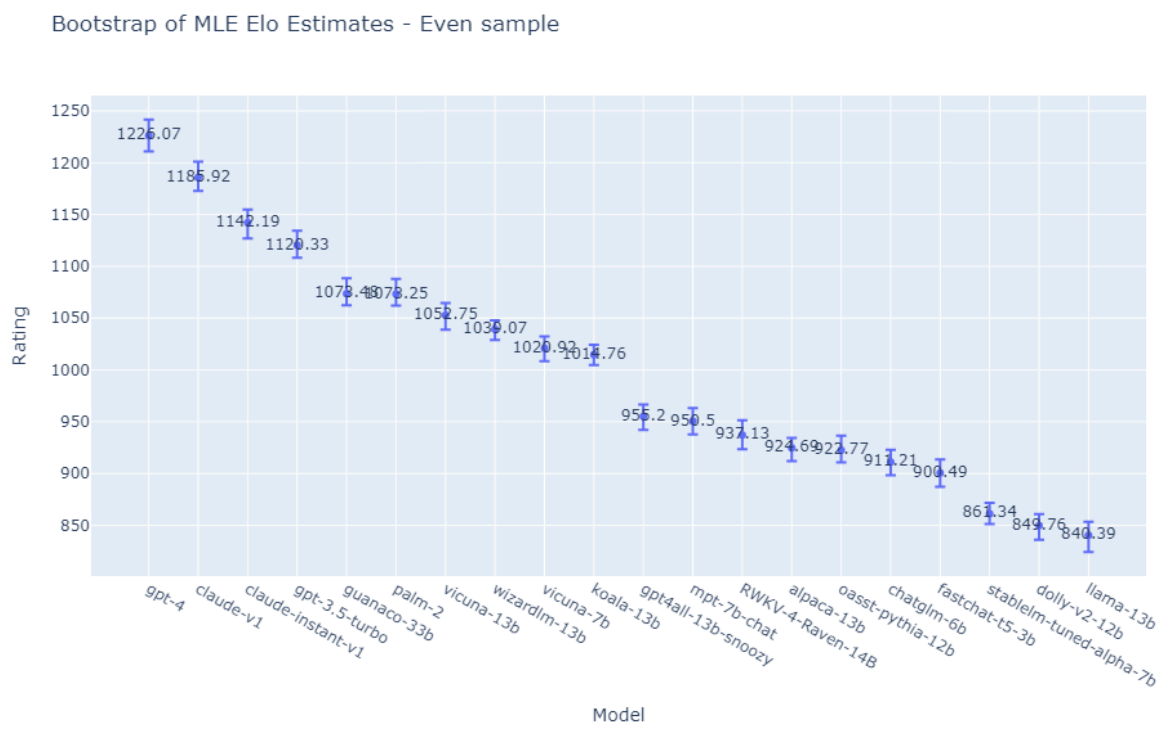
(a) The two classes are:

- **0 – Victor**
- **1 – Tie**

(b) The four classes are:

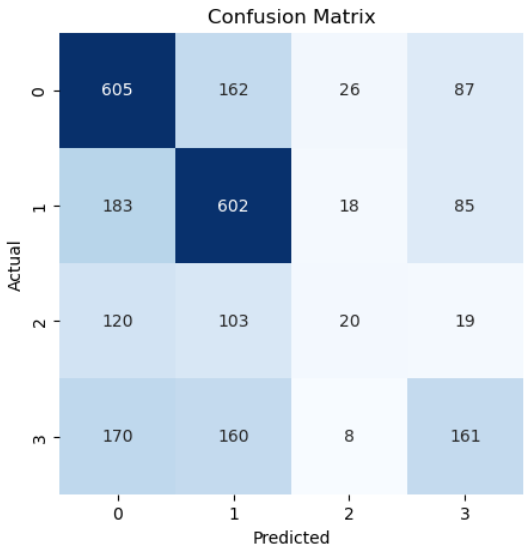
- **0 – Model A**
- **1 – Model B**
- **3 – Tie**
- **4 – Tie (both bad)**

Figure 15 - ELO rating



The graph above is the Elo rating calculated for each model that was then used as an additional feature in training the logistic regression model for Task A.

Figure 16 - Final Task A confusion matrix



For Task B, our first linear regression model had a mean squared error (mse) of 2.439 on the training set, 2.495 on the validation set, and an average mse (or loss) of about 2.453 using k-fold cross validation. The absolute difference between the average hardness score and the average predicted hardness score was also relatively high compared to the other 2 linear regression models. The model does not determine the hardness score of easier prompts (i.e. lower hardness scores) very well when compared to harder prompts. Although it may not seem obvious from looking at the absolute difference in average score minus avg predicted score, the model also has difficulty determining the true hardness score of difficult prompts (i.e. high hardness scores) when visualizing the true vs. predicted scores via box plots. Please note that box plots were used to visualize the true vs. predicted results to avoid overplotting with scatter plots. This also helped us visualize the range of values our model predicts for each true hardness score that was rounded to their nearest integer.

*Figure 17 - Visual inspection of **first linear regression model performance**. Box plots of predicted vs actual average hardness scores rounded to the nearest integer.*

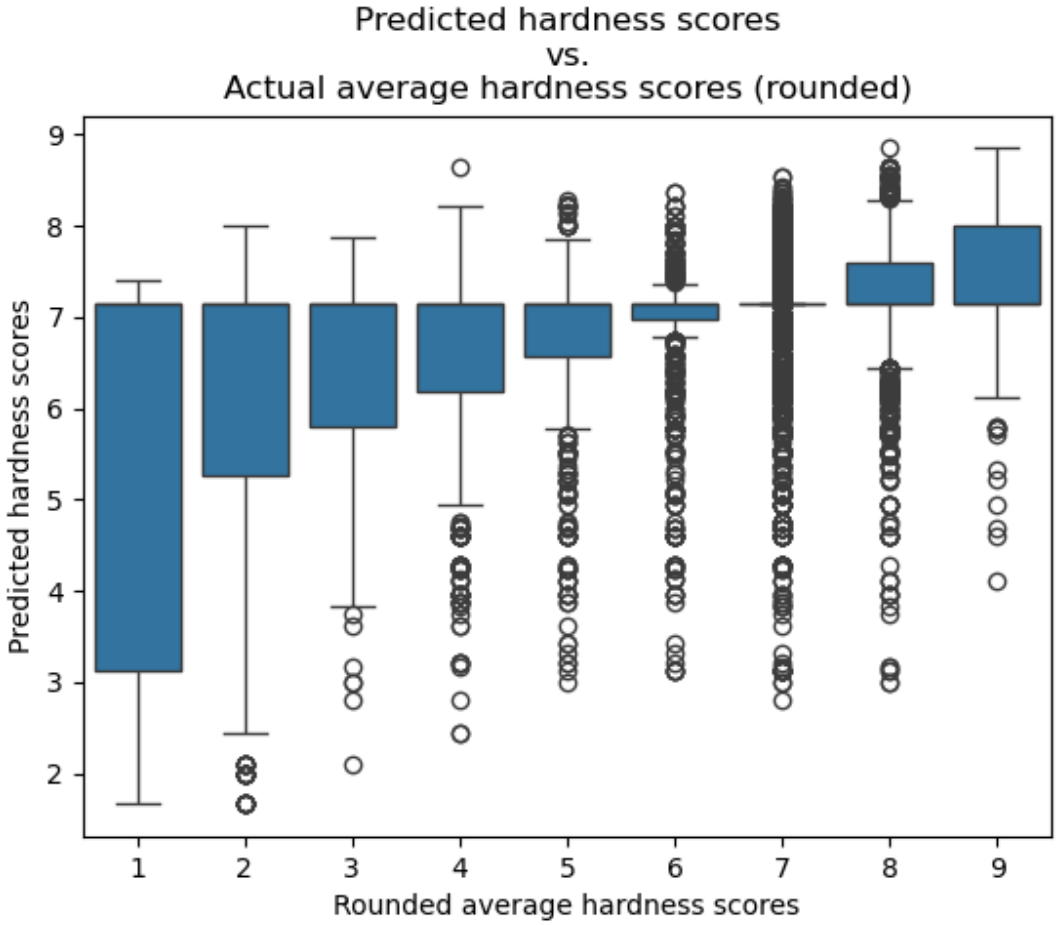


*Table 3 - Quantitative inspection of **first linear regression model performance.** Table of true average hardness scores in the dataset (avg y), predicted average hardness score (avg y^), and their absolute differences*

avg y	avg y^	abs(avg y - avg y^)
1	5.62	4.62
2	6.12	4.12
3	6.40	3.40
4	6.60	2.60
5	6.75	1.75
6	6.86	0.86
7	7.07	0.07
8	7.31	0.69
9	7.46	1.54

Our second linear regression model had a mean squared error (mse) of 1.759 on the training set, 1.862 on the validation set, and an average mse (or loss) of about 1.862 using k-fold cross validation. This is a significant improvement compared to the first model. The absolute difference between the average hardness score and the average predicted hardness score was also lowered compared to the first model. However, when applying this model to our test set, the mse using our test set was greater than 2.7, suggesting that our second linear regression model is overfitting to the training data.

*Figure 18 - Visual inspection of **second linear regression model performance**. Box plots of predicted vs actual average hardness scores rounded to the nearest integer.*

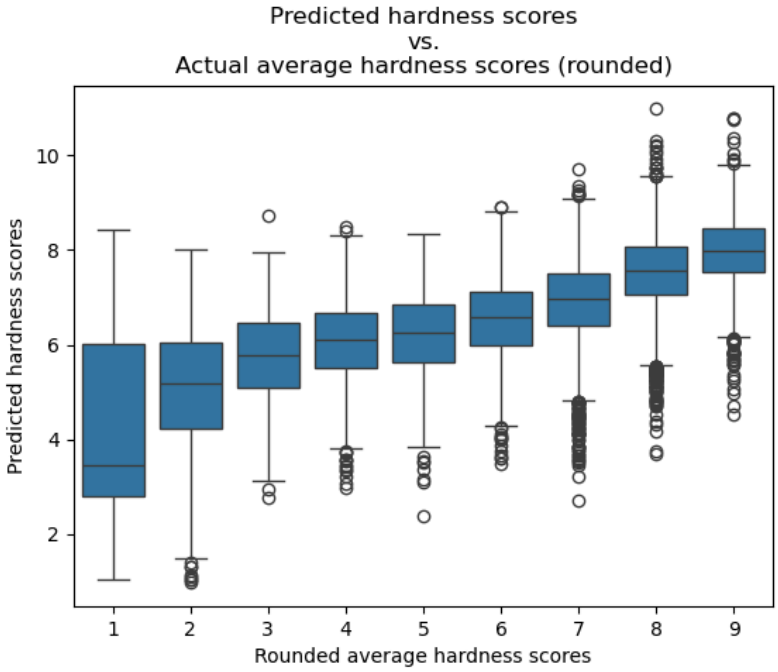


*Table 4 - Quantitative inspection of **second linear regression model performance.** Table of true average hardness scores in the dataset (avg y), predicted average hardness score (avg y^), and their absolute differences.*

avg y	avg y^	abs(avg y - avg y^)
1	4.00	3.00
2	5.04	3.04
3	5.73	2.73
4	6.05	2.05
5	6.19	1.19
6	6.54	0.54
7	6.92	0.08
8	7.54	0.46
9	7.96	1.04

Our third linear regression model helped generalize our model to unseen data. It had a mean squared error (mse) of 1.922 on the training set, 1.962 on the validation set, and an average mse (or loss) of about 1.979 using k-fold cross validation. The absolute difference between the average hardness score and the average predicted hardness score is also lower compared to the first model, but slightly higher than the second model. Using our test set, we had also gotten our mse to be less than 2.7.

*Figure 19 - Visual inspection of **third linear regression model performance**. (Right) Table of true average hardness scores in the dataset (avg y), predicted average hardness score (avg y^), and their absolute differences. (Left) Box plots of predicted vs actual average hardness scores rounded to the nearest integer. Please note that the box plots were used to visualize the true vs. predicted results to avoid overplotting with scatter plots and visualize the range of predicted values for each true value.*

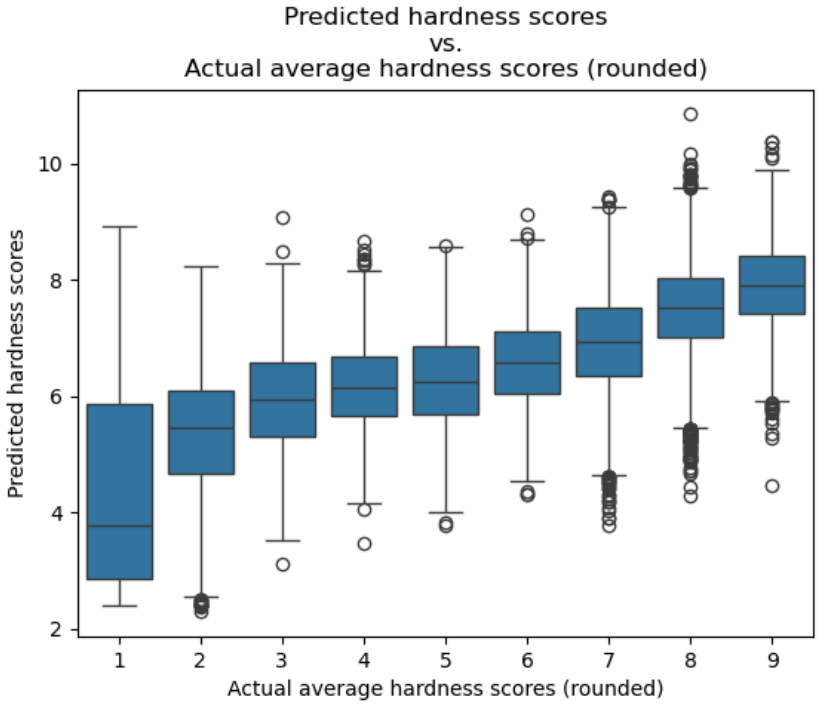


*Table 5 - Quantitative inspection of **third linear regression model performance.** Table of true average hardness scores in the dataset (avg y), predicted average hardness score (avg y^), and their absolute differences.*

avg y	avg y^	abs(avg y - avg y^)
1	4.30	3.30
2	5.36	3.36
3	5.97	2.97
4	6.19	2.19
5	6.27	1.27
6	6.58	0.58
7	6.92	0.08
8	7.49	0.51
9	7.89	1.11

5.0 Discussion

For Task A there are issues utilizing random forest models which seem to default mostly to the most prominent class. Logistic regression when the class weight is set to balanced produces better results. We have run out of features to add to differentiate the ties from a victor though and we are considering a more complicated clustering system using K-means and a preporcess step for determining if it is a tie or victory.

For task B, our third linear regression model reduced the loss between actual vs. predicted hardness scores and generalized well to unseen data. Our third model was also our best model because it was trained on data processed with only 10 frequently occurring topics and 50 clusters from K-means clustering of prompt embeddings, thus using the least amount of features / dimensions for training while still retaining similar loss metrics compared to our overfitted model that used more topic models and PCA. Across all three different linear regression models, it is interesting to see that the absolute difference between average actual hardness scores vs. average predicted hardness scores followed a similar trend: Larger differences were seen for easier questions (i.e. truly and relatively low hardness scores), while smaller differences were seen for harder questions (i.e. truly and relatively higher hardness scores). This would make sense when inspecting the data during our EDA process since a majority of the prompts were skewed towards hardness scores around 7~8, suggesting that our model may be limited by the data it was trained on, and had made better predictions with data that it saw more of. Furthermore, our best linear regression model captures about ~36% variability of the data, suggesting that there may be high bias when using our model to determine the hardness score of future prompts for discussion. This raises ethical concerns such as fasifly interpreting a prompt as hard, or choosing chatbot models for human use in the future. For example, if this model wrongfully interprets a prompt as difficult for an underperforming chatbot (such as llama-13b) that won in conversation, it may appear to a live human audience that llama-13b is a great chatbot to use for NLP related tasks. More data needs to be collected on truly easier prompts to train our model and make more accurate determinations of easier prompts. Acknowledging these limitations and concerns is important for the future improvement of hardness determination models and choices made in NLP related tasks.

6.0 Citations

1. Shankar, S., Gil, Y., Lee, S., Chien, S., & Heer, J. (2024). *Who validates the validators? Aligning LLM-assisted evaluation of LLM outputs with human preferences*. arXiv preprint arXiv:2404.12272. https://arxiv.org/abs/2404.12272

2. Panickssery, A., Bowman, S. R., & Feng, S. (2024). *LLM evaluators recognize and favor their own generations*. arXiv preprint arXiv:2404.13076. https://arxiv.org/abs/2404.13076