Random forests predict winning LLM in Chatbot Arena as well as prompt hardness

By Emma Brugman, Caroline Martin and Mariëtte Peutz

Abstract

Chatbots by Large Language Models (LLMs) show immense potential across industries. While becoming increasingly powerful, their performance on prompts and which factors impact their performance remain difficult to quantify. In this research, we will build a classification model to predict which model a human user would appoint as winner in the Chatbot Arena (Chiang *et al.*, 2024), and an inference model to predict the hardness score ChatGPT3.5 would give to each prompt. We build these models on aspects like prompt topic and textual properties like sentiment and subjectivity. To predict the winning chatbot model for a given prompt, we implemented a Random Forest classification model and evaluated its performance using accuracy, precision, recall, F1 score, and AUC from an ROC curve. To predict the hardness score of the prompt, we implemented a Random Forest model as well, and utilized mean squared error (MSE), root mean squared error (RMSE) and adjusted R-squared to evaluate the performance of the model. The OLS model had an RMSE that implied that the model was predicting well, however, the adjusted R-squared score implied that the model was not fully capturing the complexity of the data. The Random Forest model performed well in this, after implementing oversampling for the class of prompts with hardness scores under 3. The included features do provide relevant information and play a role in human preference of a model's response as well as the perceived hardness. However, this subjective data remains complex and further research is needed to build more accurate models.

Presentation Link: https://youtu.be/A11uEx3qapI

Introduction

Chatbots supported by Large Language Models (LLM) have revolutionized the world of IT. Ever since the first chatbot was created by MIT researcher Joseph Weizenbaum in 1966 (Weizenbaum, 1966), simulating human conversation has become the holy grail for computer scientists and AI researchers. The launch and widespread use of LLMs GPT in 2018 and ChatGPT in 2022 revolutionized the way knowledge is organized and accessed globally (OpenAI, 2023). With the amount of LLMs in circulation steadily increasing, it remains relevant to test performance of LLMs on similar or the same prompts. In 2023, Chiang *et al.* (2024) launched the Chatbot Arena, where two LLMs reply to the same prompt and human users indicate their preference between the models' responses. Human preferences are broadly used as a source of reinforcement learning for Machine Learning in general and chatbots specifically (Christiano *et al.*, 2023; Kadavath *et al.*, 2022). Besides human preferences, LLMs are being trained as judges as well on Chatbot Arena conversations (Zheng *et al.*, 2023; Jung, Brahman & Choi, 2024). It is still unknown what factors exactly cause these human preferences between chatbot responses, and how we can predict the human preference. In this research, we present a classification model to predict which chatbot response a human judge would appoint as winner. We will use ELO-ratings to assess the relative performance of each LLM, and use the difference of these ELO-ratings as a feature in our classification model (Elo, 1967; Boubdir *et al.*, 2023).

The ability of LLMs to give accurate responses to prompts varies per prompt. Several studies have researched which qualities of prompts influence LLM performance. For example, previous research has shown that more complicated prompt syntax is negatively correlated with LLM performance (Linzbach *et al.*, 2023). Several other linguistic qualities such as verb tense and sentence mood affect LLM performance as well (Leidinger, van Rooij & Shutova, 2023). In this research, we present a regression model that predicts the hardness level of a prompt, based on data generated by LLM ChatGPT 3.5.

Description of the Data

We train these models on data collected from the Chatbot Arena (Chiang *et al.,* 2024). The original dataset is a sample of 33000 sets of conversations between a human user and two LLM chatbots. The data is collected from roughly 13000 unique IP addresses. Cleaned to contain only non-toxic, English and singular-round conversations, the dataset consists of 25282 conversations, of which none are duplicates or null values. The data is granular at the level of sets of 2 conversations. For every conversation, the user appointed a 'winner', with four possibilities: model A wins, model B wins, there is a tie, or there is a tie annotated 'bothbad'. Our analysis is supported by four auxiliary datasets: one dataset where ChatGPT3.5 modeled topics and appointed a hardness score for each topic, and three embeddings datasets for the prompts and two responses. Embeddings are representations of words or phrases w as a d-dimensional vector $\vec{w} \in \mathbb{R}^d$, with $\|\vec{w}\| = 1$ for each $w \in W$ (Bolukbasi *et al.*, 2016). In the dataset containing topics & hardness scores, 59 prompts included null values. These prompts all contained similar characteristics, the chatbots were asked to 'act as' a terminal or AI assistant, create a JSON file, or the prompt contained multiple prompts at once. Given their low proportion of the main dataset and the fact that they are a different category of prompts which means imputing with the mean, mode or median would not be accurate, we chose to exclude these prompts. We furthered our analysis with the mean hardness score per prompt.

The length of the prompts and responses a and b were right-skewed with outliers, with medians of 72, 591 and 582 characters respectively, these were used as features. These outliers were not removed from the dataset, as longer prompts and responses may provide valuable insights. Out of all 25282 prompts, 20727 were unique, the prompt with the highest frequency occurring 33 times. The topics modeled were also mainly unique, with 11174 unique topics. The most frequently occurring topics were creative writing (596 prompts), factual accuracy (487 prompts), problem-solving & creativity (407 prompts), and factual knowledge (314 prompts). The 20 LLMs in the dataset performed differently on prompts which were modeled into the topic categories of mathematics, fact-based prompts, creativity and problem-solving. They also performed differently on prompts with different hardness scores, which is why we merged the main dataset with the topic & hardness score dataset. We also found that ChatGPT3.5 on average found prompts containing the auxiliary verbs 'where', 'when', 'why' and 'which' were given a higher hardness score than the average prompt, and prompts containing 'who' and 'what' were given a lower hardness score than the average prompt. Based on this, we included the presence of auxiliary verbs in our models. We also looked for the presence of negation words in the responses, spelling mistakes, and assessed the subjectivity using textblob and readability using textstat. All detailed information on features used is elaborated on in the methodology section. Lastly, we computed the cosine similarity between the embeddings, computed according to the following formula:

$$\text{Cosine Similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}} .$$

We did this as we do not normalize our embeddings, and they are generated by ChatGPT3.5, instead of by the models responding. This means our embeddings mainly encode semantics, which are represented by the cosine similarity, and any information encoded in magnitude such as confidence in the response is not relevant (Schwaber-Cohen, 2023; Zhou *et al.*, 2022).

We then analyzed the impact these features had on whether a response was chosen as the winner, and on the predicted hardness score. Correlation plots are shown below. The correlation coefficients are all very low, below 0.015. However, removing features with low correlations lowered the performance of our models. Thus, we can conclude that the majority of these low-correlating features still hold valuable information.

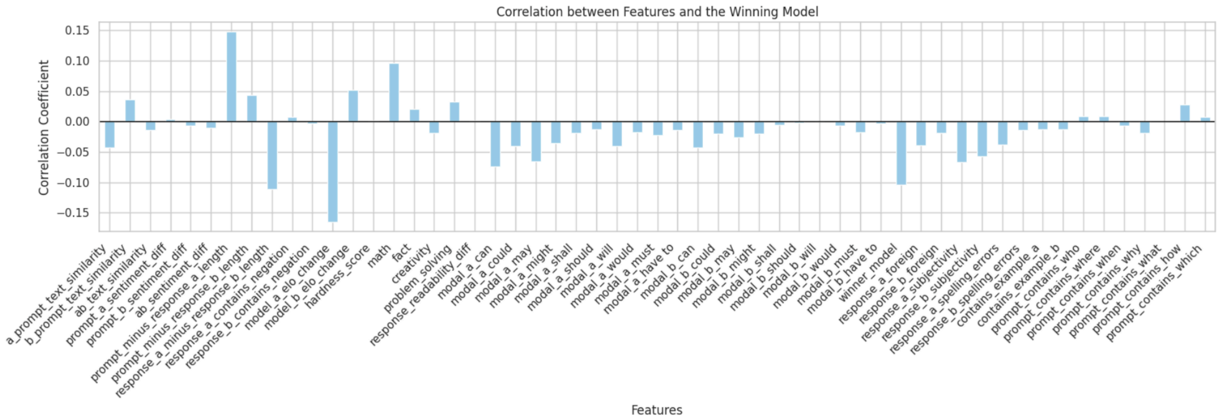


Fig. 1: Correlations between features and a correct prediction of the winning model.

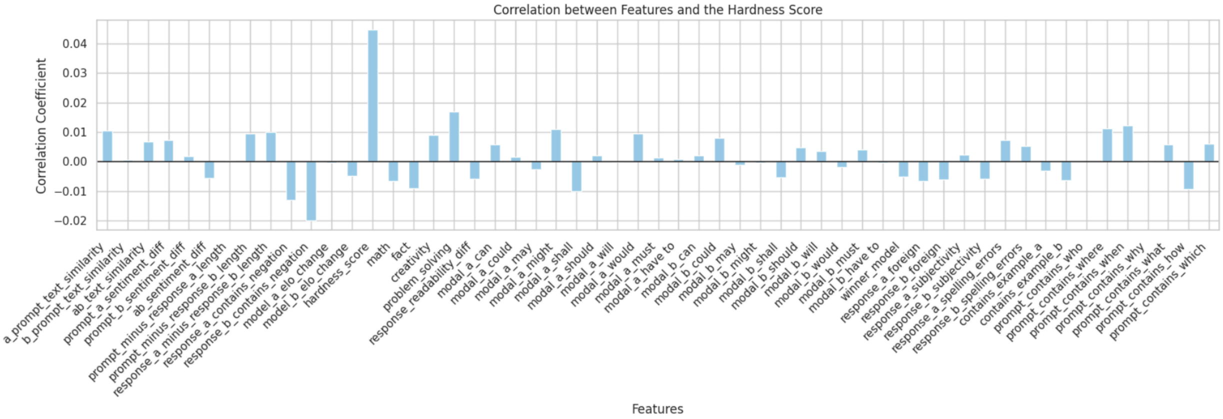


Fig. 2: Correlations between features and the hardness score.

Methodology

Task A

We used the following features to predict the winning chatbot model in task A: one hot encoded names of chatbot models for models a and b, difference in chatbot model elo rating between models a and b, difference in response readability, presence of auxiliary verbs in each of the responses, presence of question words in the prompt, presence of an example in the responses, presence of grammatical errors in the responses, subjectivity score of responses, prompt topic category, presence of negation words in the responses, difference in sentiment score between the prompt and each response, difference in sentiment score between the 2 responses, presence of non-English characters in the responses, cosine similarity of prompt embeddings and each of the response embeddings, similarity of both response embeddings, difference in length between the prompt and each response, and difference of the 2 response lengths.

A correlation analysis was done to see which features had a significant correlation with the winning model. Features with a correlation coefficient value of 0.05 or greater were considered to have a significant impact on determining the winning model. The difference in prompt and response B length, ELO change for both responses, prompts topics categorized as math, presence of "may" or "can" in model A responses, and names of chatbot models for models A and B.
and were the only features that showed significant correlation with the winning chatbot model. We created 2 feature sets, one containing the full set of features and one with only significant features and tested to see which combination produced the most efficient results.

To predict the winning model - or the possibilities of a tie and a tie ‘bothbad’ - for a given prompt, we built 2 classification models: a Logistic Regression model and a Random Forest model. The Logistic Regression model calculated the following probability based on our dataset x and feature-vector θ:

$$P(Y = 1 \mid x) = \frac{1}{1+e^{-x^\top \theta}}$$

In the further equations, this probability is represented by $\sigma\left(\mathbf{X}_i^\top \boldsymbol{\theta}\right)$. We used cross-entropy loss to evaluate the model, and regularized with L2- or Ridge regularization. The optimal parameters were then calculated as follows:

$$\arg\min_{\boldsymbol{\theta}} -\frac{1}{n} \sum_{i=1}^{n} \left(y_i \log\left(\sigma\left(\mathbf{X}_i^\top \boldsymbol{\theta}\right)\right) + (1-y_i)\log\left(1-\sigma\left(\mathbf{X}_i^\top \boldsymbol{\theta}\right)\right)\right) + \lambda \sum_{j=1}^{d} \theta_j^2$$

Our Logistic Regression model takes feature inputs and utilises an Iterative Reweighted Least Squares (IRLS) optimization algorithm to assign weights to each of the features. IRLS utilizes Newton’s method to minimize the log-likelihood loss function based on both the gradient and the hessian, resulting in a more stable and robust optimization. Secondly, we implemented a Random Forest model. Random Forest models were introduced in 2001 by L. Breiman at UC Berkeley. The Random Forest contains 200 decision trees, each of which is trained on random subsets of the data and features. This improves robustness and reduces overfitting. Each tree splits the feature space into regions, and aggregates final predictions through majority voting over these trees, making it an ensemble approach. Logistic regression models can be beneficial due to their simplicity and interpretability. These models also have lower rates of computation, which can significantly impact execution time when you have large datasets. However, the implementation of feature thresholds and decision trees in the random forest model allows it to capture non-linear relationships between variables, which may increase the accuracy of predictions if there are non-linear patterns in our data. The logistic regression also assumes independence of features, which could cause issues with multicollinearity and could reduce accuracy if it fails to capture feature interactions. The results of these models were compared to find an optimum balance between accuracy and efficiency. Randomized predictions of the winning model were generated using np.random_choice and were compared to the results of the Logistic Regression and Random Forest model to analyze the efficacy of their classifications.

Each model was tested on the significant features set as well as the original features set to determine if features with lower correlations had a significant impact on the overall accuracy.

The dataset was split into 80 percent training data and 20 percent testing data. Categorical features were one-hot encoded, and features were scaled between 0 and 1 to ensure consistency in the models.

A grid search was applied to the logistic regression model to find optimal parameters. Due to the size of the data set and the increased number of parameters in the random forest model, a traditional grid search was too computationally expensive. Instead, we implemented manual grid sampling to find the optimal parameters for the Random Forest model. The results of each model were evaluated using accuracy, precision, recall, F1 score, and AUC value. These metrics were computed as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{\sum_{i=1}^{4} \text{True Positives}_i}{\text{Total number of prompts}}$$

$$\text{Precision}_i = \frac{\text{True Positives}_i}{\text{True Positives}_i + \text{False Positives}_i}$$

$$\text{Recall}_i = \frac{\text{True Positives}_i}{\text{True Positives}_i + \text{False Negatives}_i}$$

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

$$\text{AUC} = \int_0^1 \text{True Positive Rate}(\text{False Positive Rate})\, d(\text{False Positive Rate})$$

We observed significant class imbalances for the winning chatbot model. Model A won in 9002 conversations, model B won in 8862 conversations, the models tied in 2786 conversations, and they were labeled as a tie both bad in 4632 conversations. This translates to 35.6 percent, 35.1 percent, 11.0 percent, and 18.3 percent respectively. Figure 3 shows a visual representation of this imbalance. Variations of class weights and oversampling techniques were tested on the classification models to determine if the class imbalance impacted the results of each model.

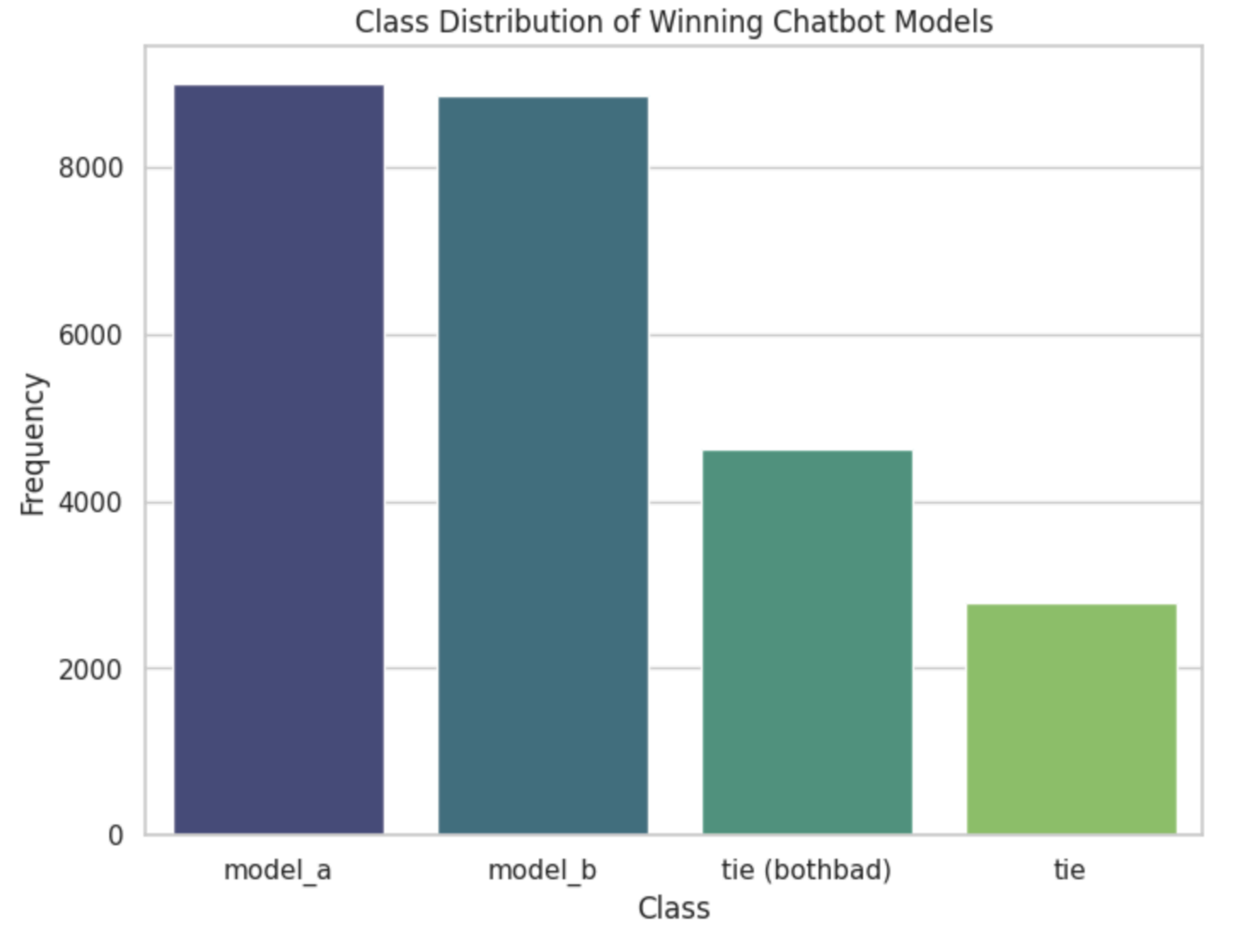


Fig. 3: The above figure depicts the spread of data across each winning model class.

Task B

The research question for this task was to predict the difficulty of a question, called hardness score, ranging from 1 to 10.

To start with our modeling process, we began with feature engineering. This is necessary for identifying the predictors that will help us predict hardness scores. We engineered several different features to include various situations that we believed could have an influence on our task. The final dataset we used included a large variety of features, so we included only binary and numeric features in our selection process. We created scatter plots for each feature to understand how they relate to hardness scores. We used these scatterplots, explained variance from the models, feature importance plots and correlation heatmaps to further look into the types of relationships the features had, not only with our target variable, but with each other as well. In order to avoid multicollinearity and overfitting, we removed features that contributed less than 1% to the variance and features that were highly correlated with each other. For some of the features that were highly correlated with each other but contribute meaningfully to the variance, we created feature interactions.

The final feature set includes: prompt_length, length_interaction, elo_modela_minus_modelb, response_a_length_polarity, response_b_length_polarity, polarity_difference, subjectivity_difference, model_a_response_noun_count, model_a_response_verb_count, model_b_response_noun_count, model_b_response_verb_count, prompt_noun_count, and prompt_verb_count. These features were selected for their contributions to the variance and observed correlations with hardness scores. For example, longer prompts (prompt_length) consistently led to higher hardness scores, while the interaction between the lengths of model a's and model b's responses (length_interaction) looks at whether the combined effort of the two models in how much they write, is linked to the difficulty of the question. Similarly, elo_modela_minus_modelb captured the performance gap between the models. We included features related to LLM performance and responses like these, as it is possible these have impacted the hardness score generation by ChatGPT3.5. Other features, such as creative and math topics, provide insights into whether different difficult topics will also influence the difficulty of the prompt. By including linguistic features, such as noun and verb counts, we allowed the model to also account for the composition of both the prompts and the responses.

To evaluate the performance of our model we used several different metrics. Mean-squared-error (MSE) was used to measure the average squared difference between the predicted and true values, which elucidates the overall prediction accuracy (James et al.). Root mean-squared error (RMSE) was used to evaluate the model's error in the same units as the target variable, making it more interpretable than MSE (James et al.). We also used calculated the RMSE over three bins in the hardness score - a low range of hardness scores 0-3, a mid range of hardness scores 4-6, and a high range of hardness scores 7-9, to evaluate patterns in our residuals. R^2 was used to assess how much variance in hardness scores was captured by the model (James et al.). A residual standard deviation was also calculated to look into the dispersion of errors to evaluate how consistent the model's predictions were (James et al.). Together, these metrics provided a thorough evaluation of the model's capabilities.

The metrics were computed according to the following formulas:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where y_i denotes the actual hardness score, $\hat{y}_i$ denotes the predicted hardness score, $\bar{y}$ is the mean of the actual hardness scores, and n is the total number of prompts.

In our first attempt at completing the task, we utilized an Ordinary Least Squares (OLS) regression model, predicting hardness scores ($\hat{\mathbf{Y}}$) with the training data ($\boldsymbol{X}$) and optimal parameters (θ): $\hat{\mathbf{Y}} = \mathbf{X}\boldsymbol{\theta}$. Initially we chose to implement the OLS model because it has the capabilities to handle multiple input features (or predictors) and evaluate the effect of each feature on hardness scores, however, the performance of the model was limited by its inability to understand the non-linear relationships and interactions between the features (Sahu, 2023). This limitation was marked by a *R*^2 score of 0.14 and a Mean Squared Error (MSE) of 2.55. This tells us that the model is only able to capture ~14% of the variance and the model's predictions are significantly off from the actual hardness score (Sahu, 2023). In order to predict the hardness score of a conversation, we knew we had to find relationships between our linguistic features and our target variable, hardness score. The OLS model was not fully capturing these relationships, which is why we knew we needed to implement a new approach.

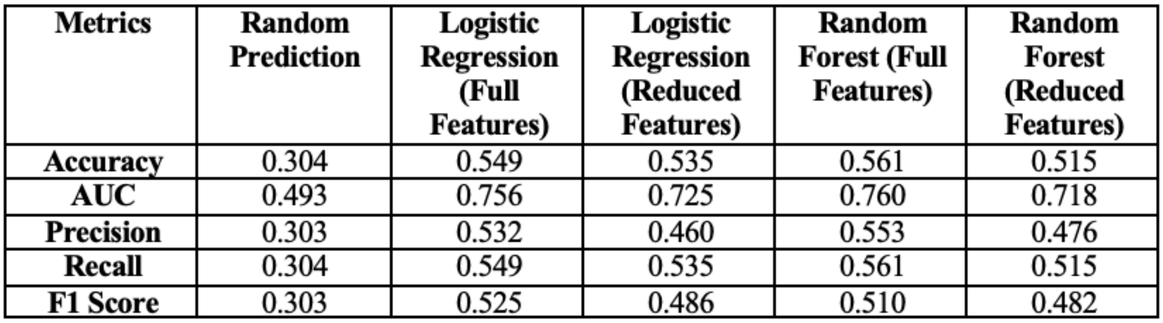
To address these problems, we decided to implement a Random Forest regressor because of this robust, ensemble-based model's ability to address our research question by capturing non-linear data and feature interactions (Lyashenko). The Random Forest model was evaluated with K-Fold cross-validation. K-Fold took an average of metrics across 5 splits to minimize overfitting and ensure robustness in the performance of the model. K-Fold also allows us to minimize bias and assess the generalizing capabilities of the model (Lyashenko). We noticed that there was an imbalance of data amongst the hardness scores, with a majority of scores being on the higher end. We noticed that when we split our data into 3 ranges, the RMSE of the lower range, <=3, was significantly higher than the other 2 ranges, 3-6 and >6, indicating that there may not be enough data for our model to learn how to accurately predict the lower range of hardness scores (Table 2). To address this, we introduced oversampling to level out the class imbalance. This ensured that the model learned effectively across all of the ranges. To further optimize our model's performance, we utilized a grid search using GridSearchCV to find our optimal hyperparameters (Lyashenko). Our optimal hyperparameters came out to be 200 decision trees (n_estimators), no maximum depth, a minimum of 2 samples at a split, and a minimum of 1 sample at a leaf.

One challenge we were unable to overcome was the problem of attempting to compare continuous data to non-continuous data. The output hardness score was required to be an integer scaling 1-10, however our hardness score column in our dataset was an average calculated from the original dataset. This led to our visualizations comparing the true values to the predicted values to be less interpretable due to the continuous nature of the true scores.

Results:

Task A

While the full features set contained variables that had minimal linear correlation with the winning model, the models trained on the full feature set resulted in significantly higher evaluation metrics for both classification models. Both feature sets had comparable execution times, indicating there was no negative impact on execution time when incorporating all features. Using the original feature set increased the accuracy of the Logistic Regression model and Random Forest model by approximately 1.5 percent and 4.5 percent respectively. Other evaluation metrics, such as F1 score and AUC score, also showed significant improvement when using the full feature set. These metric comparisons can be found in figure 4. The importance of each feature in the final model can also be viewed in figure 5.



Metrics	Random Prediction	Logistic Regression (Full Features)	Logistic Regression (Reduced Features)	Random Forest (Full Features)	Random Forest (Reduced Features)
Accuracy	0.304	0.549	0.535	0.561	0.515
AUC	0.493	0.756	0.725	0.760	0.718
Precision	0.303	0.532	0.460	0.553	0.476
Recall	0.304	0.549	0.535	0.561	0.515
F1 Score	0.303	0.525	0.486	0.510	0.482

Fig. 4, Table 1. The above figure contains a table of evaluation metrics for each of the task A models predicted on the validation folds.

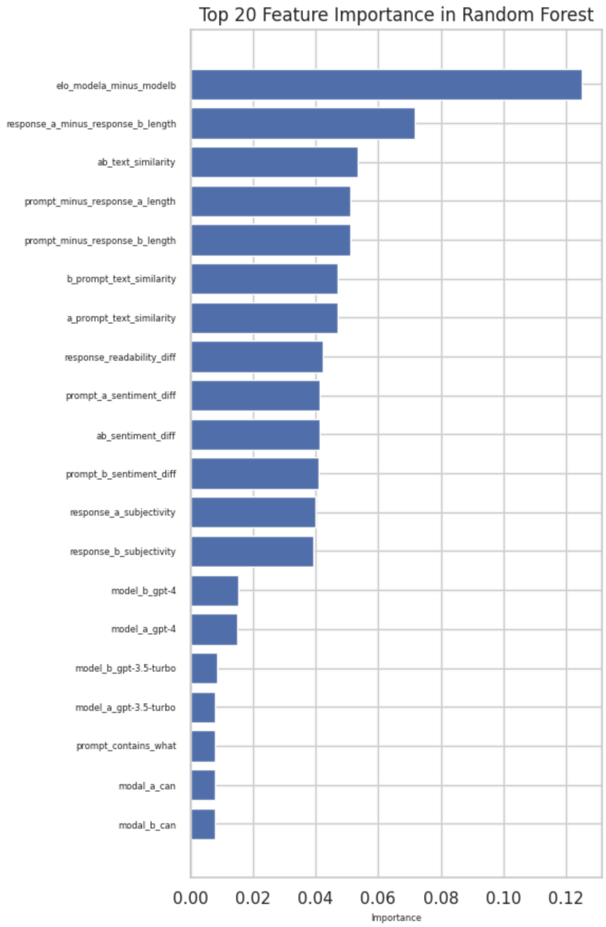


Figure 5. The above figure depicts the top 20 most important features in the Random Forest winning model predictions.

The grid search for our Logistic Regression model indicated that the best parameters were a C value ($\frac{1}{\lambda}$) of 10, 500 iterations, L2 regularization, and liblinear optimization. This produced a 54.9 percent accuracy, 53.2 percent precision, 54.9 percent recall, 0.525 F1 score, and a 75.6 percent AUC. The grid sampling for the Random Forest model indicated that the optimal parameters were 500 decision trees (n-estimators), a minimum sample split of 2 samples, and 1 sample minimum at the leaf a maximum depth of 20, and no class weights. This produced a 56.1 percent accuracy, a 55.3 percent precision, 56.1 percent recall, 0.510 percent F1 score, and a 76.0 percent AUC. The randomized predictions generated a 30.4 percent accuracy, 30.3 percent precision, 30.4 percent recall, 30.3 percent F1 score, and a 49.3 percent AUC value. Both models performed much better than the random model, indicating that these models are predicting well on the test data. However, the random forest model was slightly better for most metrics, indicating that this is the better model. The Random Forest gave an accuracy of 54.13% on the unseen test data.

There were significant differences in the execution times for each model. The algorithm to train the logistic regression model had an execution time of around 2 seconds, while the training algorithm for the random forest model had an execution time of around 19 seconds. This is nearly a 10 fold increase in execution time. However, both execution times were on the level of seconds, and did not significantly impact the success of our model.

The evaluation metrics improved by approximately 1 percent when adding a class weight of 1.33 to classes 2 and 3, representing both of the tie columns. All evaluation metrics decreased when the classes were fully balanced using weights that were inversely proportional to the proportion of data in each class. Class weights were run as a parameter in the grid search for the random forest model and optimal results were found without applying any weights. The models were also tested using oversampling of the tie classes. This decreased the accuracy by approximately 5 percent, thus oversampling was not incorporated in the final model.

Task B

To address the task of predicting hardness score on a scale from 1 to 10, we implemented and compared two different models: Ordinary Least Squares (OLS) regression and Random Forest regressors with and without oversampling. Each model was evaluated using a variety of different metrics, such as, RMSE, MSE, R^2, and a range-specific RMSE. These results were then interpreted using visualizations shown in the related figures.

OLS Regression

The OLS model performed poorly, achieving an R^2 score of 0.15, indicating that the model was only able to capture 15% of the variance in hardness scores. This coupled with a MSE value of 2.53 and an RMSE of 1.58, indicated that this was a poor fit for the task. The MAE suggests that the predictions were off by 0.92. A scatterplot comparing true values and their corresponding predicted values reveals that the model only predicted scores higher than 6 (Fig. 6A and 6C), which possibly explains the downward pattern in the residual plot (6B).

Additionally, the residuals had a high standard deviation, indicating high variability, indicating that the model was struggling to capture the relationships between the features and the target variable (*Figure 6A*). Overall, these results convey that the model was unable to handle the non-linear relationships and interactions.

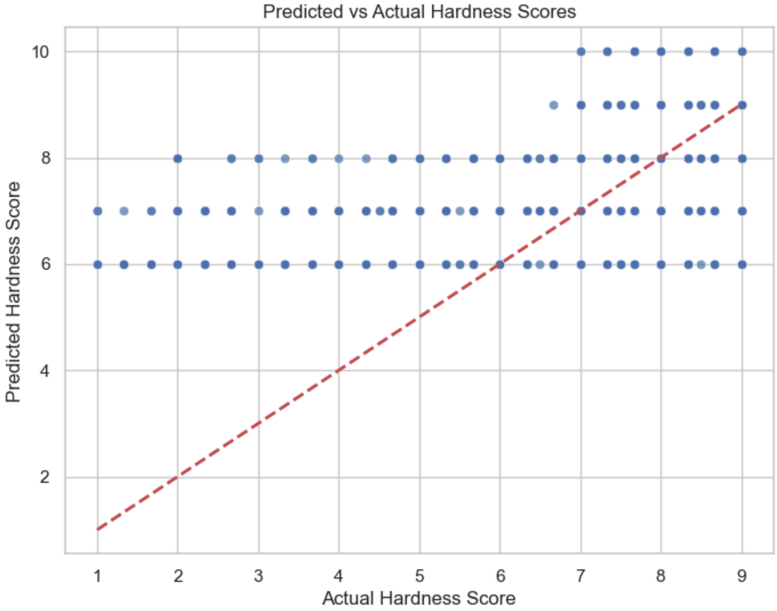


Figure 6A: This residual plot conveys the errors, represented by the residuals, against the predicted hardness scores. The red line at 0, represents zero error. This plot only shows hardness scores 6-10, conveying that the model is not fully capturing the variability in hardness scores.

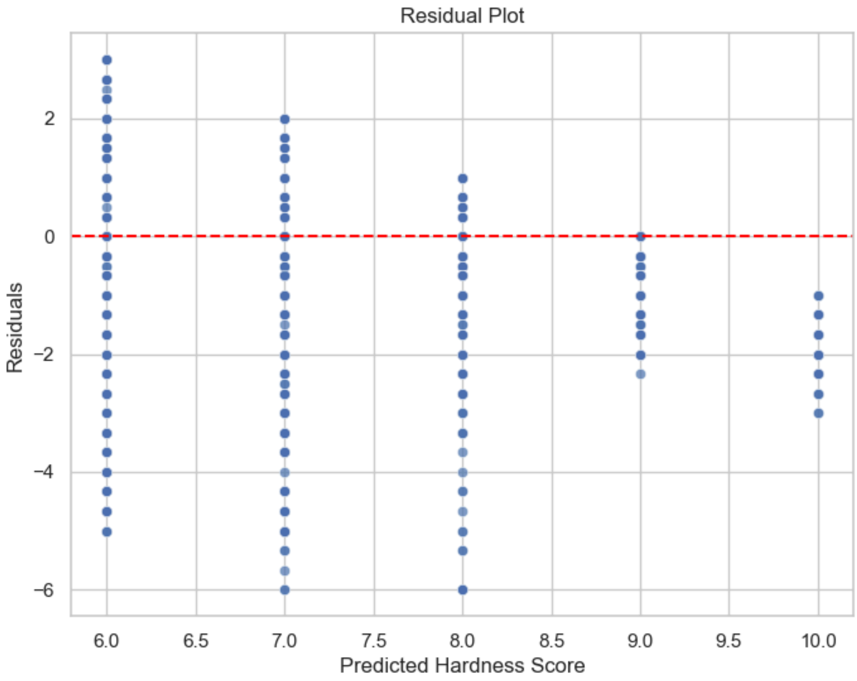


Figure 6B: This plot compares the predicted and actual hardness values. The red line represents where the predicted values equal actual values. The deviations from the red line indicate prediction errors.

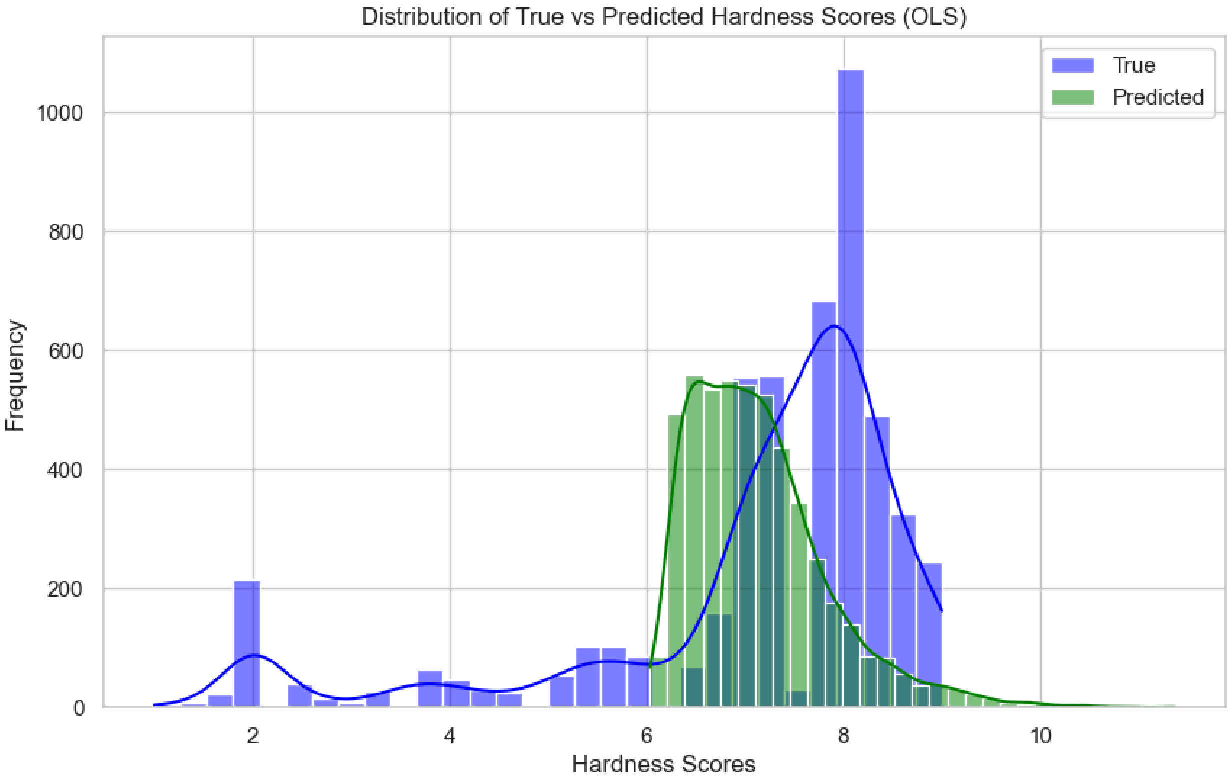


Figure 6C: KDE histogram of the distribution of true hardness scores (purple) and predicted hardness scores (green) in the training data by the OLS model.

Random Forest Regressor without Oversampling

By switching to a Random Forest regressor, we saw an immediate improved performance. Without oversampling, the Random Forest regressor achieved an R^2 score of 0.38, nearly doubling what we achieved with the OLS model. However, the training R^2 value was 0.92 and the validation R^2 was at 0.40, displaying a large and distinct difference. The difference indicates that the model was performing well on the training data but not as well on the validation data. The 0.12 gap can be explained by the presence of overfitting. With a test MSE score of 1.84, we were able to greatly improve the predictive power of the model. However, the data overly represents prompts with a hardness scores around 6-8, and has a lower representation of lower hardness scores (Fig. 7A). To address this we calculated an RMSE score for 3 ranges in the hardness scores: <=3, 3-6, and >6. After achieving our calculations, we found that the lower range was exhibiting an RMSE of 3.26, while the middle and high range showed an RMSE of 1.80 and 0.99, respectively (Table 2). This raised a problem that the low range of hardness scores are severely underpredicted and may be causing the aforementioned overfitting. We found that there was a disproportionate amount of data in each range, with the low range having the least amount of data.

Random Forest Regressor with Oversampling

To address the unbalanced data mentioned above, we incorporated oversampling on the lower range to be ⅓ of the data. By introducing oversampling to our model, we were able to greatly improve our model's performance. The model achieved an R^2-value of 0.84, indicating that the model was now capturing 84% of the variance in hardness scores (Table 2). As the R^2-value increased from 0.36 to 0.84, we can conclude that the oversampling helped reduce some of the overfitting as well. We also saw improvement with our validation MSE score of 1.12 and a validation RMSE score of 1.06, indicating smaller average errors (Table 2). Without oversampling, the RMSE is largest for the range of low hardness scores. The low range RMSE score (for hardness scores 0-3) successfully dropped from 3.26 to 0.40. The mid (4-6) and high (7-9) range scores stayed relatively the same at 1.75 and 1.15, respectively. This indicates that the model was able to learn effectively with oversampling. The model gave an MSE of 2.5978 on the unseen test data, which indicates overfitting.

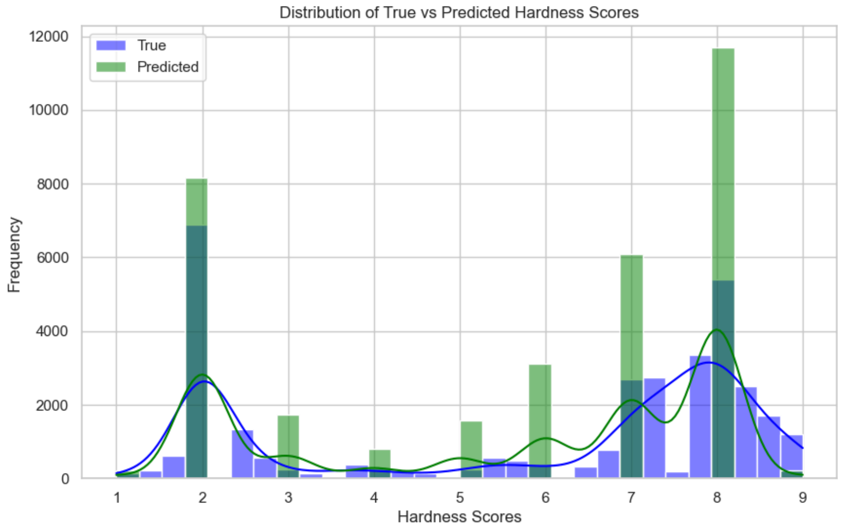


Fig. 7A: KDE histogram of the distribution of true hardness scores (purple) and predicted hardness scores (green) by the Random Forest model in the training data.

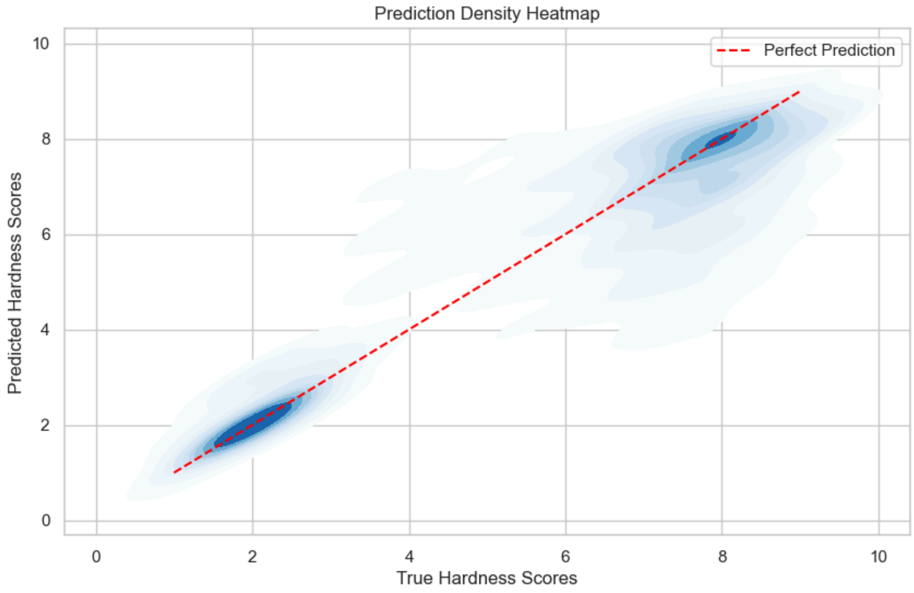


Fig. 7B: Predicted versus actual hardness scores in the training data by the Random Forest model, shown in blue. A darker blue corresponds to a higher density of datapoints, the slashed red line is plotted as an indication of a perfect prediction line.

Metric	*OLS*	*Random Forest without Oversampling training data*	*Random Forest with Oversampling training data*
MSE on training folds	*2.51*	*0.25*	*0.21*
MSE on validation folds	*2.52*	*1.77*	*1.05*
R-squared	*0.147*	*0.36*	*0.84*
RMSE on hardness score range 0-3	-	*3.26*	*0.40*
RMSE on hardness score range 3-6	-	*1.80*	*1.74*
RMSE on hardness score range >6	-	*0.99*	*1.15*
MSE on unseen test data	-	-	*2.60*

Table 2. Performance metrics of Task B models.

Conclusion

Overall, the Random Forest regressor with oversampling was revealed as the best model to predict hardness scores. The model's ability to handle non-linear relationships coupled with the oversampling, addressed key problems we had with overfitting and unbalanced data. The improvement in the performance metrics demonstrates that, not only is this the best modeling technique, but also captured the complexity of predicting hardness scores.

Discussion

Task A

The Logistic Regression model was able to accurately predict the winning chatbot model 54.9 percent of the time. Compared to the randomized prediction accuracy of 30.4 percent, this model is able to give a pretty good prediction of user preferences and meets our accuracy goal. The random forest model had a prediction accuracy of 56.1 percent on the training data and 54.1% on the test data, which met our accuracy goal and performed slightly better than the logistic regression.. The random forest model also outperformed the logistic regression model in most other performance metrics, indicating that it was the better model.

Both models have a slightly higher recall than precision, which indicates that there are more false positive than false negative predictions. Since both responses are typically able to answer the prompt, the winning chatbot model boils down to subtle user preferences. Therefore, it doesn't make a large impact in the overall functionality of the model to prioritize either a higher precision or a higher recall. The AUC value was much higher than other performance metrics in both models. AUC values calculate the overall correctness, incorporating true positives and negatives as well as false positives and negatives. This is typically a better metric than accuracy when you have imbalanced data, since accuracy only incorporates true positives and negatives. Since our data is highly imbalanced, our AUC values are a much better indicator of the performance of our model. These values were above 70 percent for both models, but the random forest had a better prediction with an AUC value of 76.0 percent and shows that it is predicting the winning model much better than a random prediction would.

The AUC value indicates that the model is doing a good job at predicting the overall variance in our data. However, if we take a look at figure 7 we can see that our model is showing some bias, in which it overpredicts model A or model B and underpredicts the tie classes. Normally, this could be attributed to the class imbalance. However, since our model performed worse when either class weights or oversampling of minority classes was implemented, this is not likely to be the cause of the bias in our model. It is possible that this bias is caused by a class overlap, in which the tie classes incorporate features that are correlated with both model A and model B. This is supported by the fact that the tie class shows the greatest amount of bias. This class overlap is likely associated with all of the features in our model, so excluding certain features would not improve the bias.

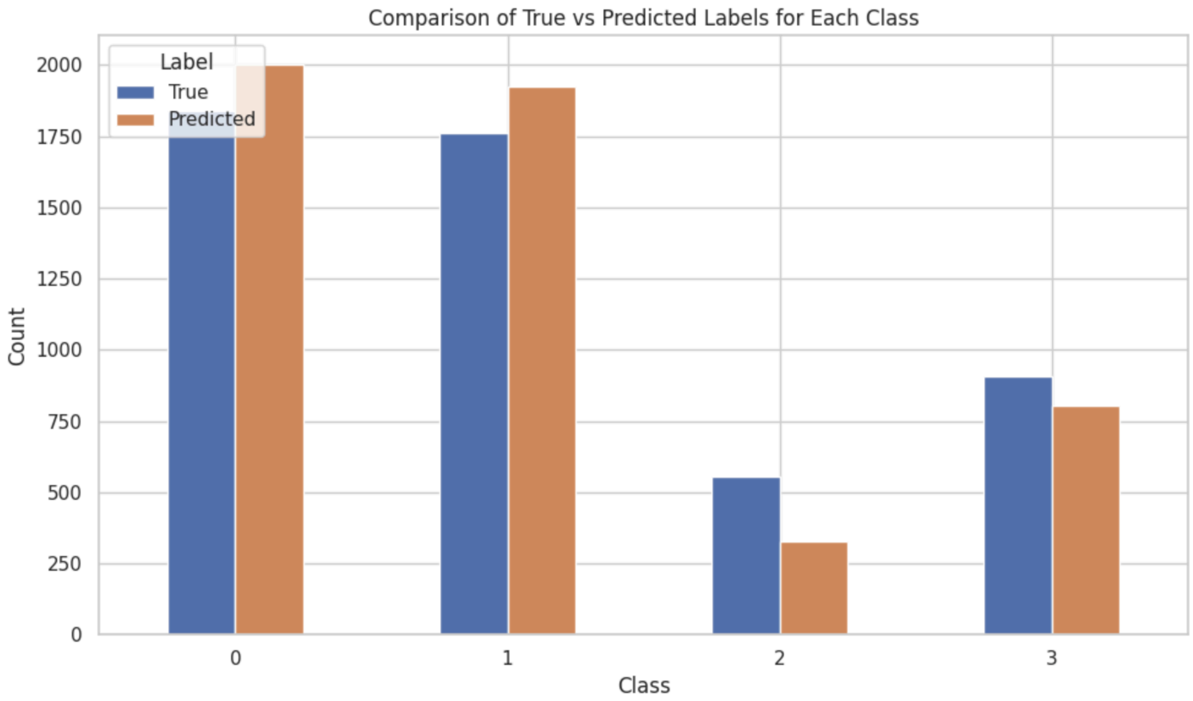


Figure 8. The figure above depicts the difference between true proportions and predicted proportions from the Random Forest model across the 4 winning model classifications. 0 represents model A, 1 represents model B, 2 represents tie, and 3 represents tie both bad.

The difference in prompt and response a length, ELO change for both responses, hardness score, prompts topics categorized as math, presence of “may” or “can” in model a responses, name of the winning model, and response subjectivity were the only features that showed significant correlation with the winning chatbot model. It was interesting that there were higher correlations between response a and the winning model for features like presence of auxiliary verbs. We expect that this is due to a class imbalance, where model A contained more conversations for the models to learn from during training. Reducing the number of features negatively impacted the prediction accuracy of our models without positively impacting the execution time, so we kept the full list of features in our final model.

The training algorithm for the random forest model had a significantly higher execution time than the logistic regression model, but we had an incredibly large dataset and it was still able to train the model in under 20 seconds. Thus it seems that the increase in accuracy outweighs any decreases in efficiency when switching to the random forest model. If we had to apply this model to a larger dataset, this discrepancy in execution time might be a bigger issue, but for this task, the random forest model is clearly the best choice. This indicates that there are likely some non-linear patterns within our data that the random forest model is better equipped to analyze.

Task B

The goal of this task was to predict the difficulty, or hardness score, of the prompts on a scale of 1 to 10. While we did note that the Random Forest model with oversampling did display a strong performance in relation to the task, there are some limitations present. One limitation was that the target variable, hardness score, needed to be an integer. Our approach to calculating the hardness score from the given data, left the true values for the target variable as a continuous variable. This discrepancy introduced issues in interpreting our visualizations, particularly in our predicted vs true values plot (FIGURE). While rounding the true values was an option, we decided against it in order to maintain the integrity of the data and not introduce bias. K-Fold cross-validation ensured robustness in the training process. However, we still observed overfitting in the Random Forest model in differences between performance on the train and validation sets, even with oversampling. While the oversampling helped address some of that overfitting in addition to the data imbalance, we may have introduced possible bias to the model by inflating the lower range’s representation in the data.

Despite our limitations, we were able to successfully predict hardness scores with an MSE of 1.12 and an R^2 score of 0.84 on the validation data, and an MSE of 2.5978 on the unseen test data, which does indicate overfitting. The model displayed success in capturing the variance across all ranges of hardness scores and reducing errors in predictions. The inclusion of our features and feature interactions played a concrete role in the model's performance. There is still be room for improvement for the Random Forest regressor (with oversampling) with other features to further reduce overfitting.

One surprising finding was the influence that prompt length had on the prediction of hardness scores. Longer prompts were highly correlated with higher difficulty. This implies that the difficulty of the prompts for the model were reliant on the length of the prompt. Another surprising finding was that introducing oversampling helped reduce overfitting of the model in our training dataset. Our goal with introducing oversampling was to help balance out the data to improve our predictions of the lower range of hardness scores. It was a pleasant surprise to see that the overfitting also decreased when implementing oversampling.

There are a few extensions to our analysis that we would implement. One is to address the continuous nature of the true hardness scores versus the discrete nature of the predictions. Another would be to potentially try another, more complex model that may be able to better encapsulate the complexity of the features, like Neural Networks. Finally, testing the model on other datasets with different prompts to validate its usability in the real world.

Conclusion

Human subjective judgement of LLM performance, as well as the hardness assessment of a human prompt by an LLM remains difficult to model accurately, due to the complexity of human language and information. However, we have shown that several features do allow us to predict the winning model and hardness score better than in a random allocation, which shows these factors at least partly explain human preference as well as indicated hardness score. Our models have successfully predicted the winning model and prompt hardness score for Chatbot Arena conversations. We have found that model strength, encoded in the difference in ELO rating, was most influential towards the winning model. Additionally, response length, the cosine similarity of response and prompt, readability, sentiment and subjectivity were important. To predict hardness score, we found that the prompt length was most indicative. More research is needed to cover the full depth of these relationships. Assessing LLM performance and the hardness of prompts is important to build strong models, which is crucial if these LLMs were to be largely deployed for societal tasks such as legal classification and the allocation of medical resources. When known, these features could be purposefully manipulated. Additionally, if hardness scores were to impact the LLM's response, for example in tuning hyperparameters or even accessing different data, this could present different results to different people. This could potentially lead to discrimination in the presentation of information. Therefore, knowledge about which features impact LLM performance and prompt hardness must be handled with caution as technology continues to develop, to ensure the human interest remains at heart. Additionally, when using human-input prompts in machine learning, privacy remains an ethical concern. While there may be attempts to anonymize data, nothing is ever accurate 100% of the time and there may come a time that someone's private information is leaked due to an error. All in all, these findings contribute towards research on LLM performance, as the technology grows in power, revolutionizing the role of information in society.

References

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. Advances in Neural Information Processing Systems, 29, 4349–4357.

Boubdir, M., Kim, E., Ermis, B., Hooker, S., & Fadaee, M. (2023). Elo uncovered: Robustness and best practices in language model evaluation. arXiv. https://arxiv.org/abs/2311.17295

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M. I., Gonzalez, J. E., & Stoica, I. (2024). Chatbot Arena: An open platform for evaluating LLMs by human preference. arXiv. https://arxiv.org/abs/2403.04132

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. arXiv. https://arxiv.org/abs/1706.03741

Elo, A. E. (1967). The proposed USCF rating system, its development, theory, and applications. Chess Life, 22(8), 242–247.

James, Gareth, et al. An Introduction to Statistical Learning : With Applications in R. Springer, 2013,static1.squarespace.com/static/5ff2adbe3fe4fe33db902812/t/6009dd9fa7bc363aa822d2c7/1611259312432/ISLR+Seventh+Printing.pdf.

Jung, J., Brahman, F., & Choi, Y. (2024). Trust or escalate: LLM judges with provable guarantees for human agreement. arXiv. https://arxiv.org/abs/2407.18370

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., ... Kaplan, J. (2022). Language models (mostly) know what they know. arXiv. https://arxiv.org/abs/2207.05221

OpenAI. (2023). GPT-4 technical report. arXiv. https://arxiv.org/abs/2303.08774

Leidinger, A., van Rooij, R., & Shutova, E. (2023). The language of prompting: What linguistic properties make a prompt successful? arXiv. https://arxiv.org/abs/2311.01967

Linzbach, S., Tressel, T., Kallmeyer, L., Dietze, S., & Jabeen, H. (2023). Decoding prompt syntax: Analysing its impact on knowledge retrieval in large language models. In WWW '23 Companion: Companion Proceedings of the ACM Web Conference 2023 (pp. 1145–1149).

Sahu, Prashant. "A Comprehensive Guide to OLS Regression: Part-1." Analytics Vidhya, 27 Jan. 2023, www.analyticsvidhya.com/blog/2023/01/a-comprehensive-guide-to-ols-regression-part-1/.

Schwaber-Cohen, R. (2023). Vector similarity explained. Pinecone. Retrieved December 12, 2024, from https://www.pinecone.io/learn/vector-similarity/

Lyashenko, Vladimir. "Random Forest Regression - the Definitive Guide | Cnvrg.io." Cnvrg.io, cnvrg.io/random-forest-regression/

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), 36-45.

Zdaniuk, B. (2023). Ordinary Least-Squares (OLS) Model. In: Maggino, F. (eds) Encyclopedia of Quality of Life and Well-Being Research. Springer, Cham. https://doi.org/10.1007/978-3-031-17299-1_2008

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv. https://arxiv.org/abs/2306.05685

Zhou, K., Ethayarajh, K., Card, D., & Jurafsky, D. (2022). Problems with cosine as a measure of embedding similarity for high frequency words. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 401–423. https://doi.org/10.18653/v1/2022.acl-short.45