Natural Language Processing Final Project Report Spring 2024

Video Link: NLP Final Video Spring 2024

Sheyda Nazarian, Yingyin Yu, Jedrick Regala Zablan

DATA 200S

ABSTRACT

In the present day, artificial intelligence and machine learning models are reshaping countless fields with natural language processing (NLP) at the forefront of improving human interaction with these systems. This project delves into the field of chatbot performance by evaluating different Chatbot Arena duels and predicting outcomes using machine learning models. Our primary objectives were to develop machine learning models to determine the winner of chatbot duels and assess the hardness of conversation questions posed in Chatbot Arena. Utilizing a dataset of 33,000 conversations, extensive data exploration, and preparation were undertaken to ensure our deep understanding of the dataset. This foundational understanding would be applied to develop our machine learning models for training. Our project's results demonstrate our models' effectiveness while exemplifying the challenges throughout the development process. Our project reveals the complexities of modeling chatbot interactions while opening new avenues for further analysis which exhibits possible advancements for chatbot and machine learning models.

1. INTRODUCTION

The evolution of artificial intelligence and machine learning models has revolutionized the technological landscape, expanding the possibilities within data analysis. These models are now crucial in driving decisions and shaping the future of various industries such as scientific research, healthcare, and finance. Chatbots have emerged as a critical interface that enhances user interactions and addresses user needs through natural language processing.

A study within the field of NLP and chatbots is highlighted in “Chatbot Using Natural Language Processing” by Pushkar Shinde and colleagues. The research evaluates the different practical applications of chatbots within existing language learning and college inquiry systems, delving into how natural language processing enhances user interaction quality in these chatbot systems. Specifically, the chatbots within this paper provided language learning assistance in a real-time and interactive platform, eliminating the need for direct human intervention in the learning process.

Another notable publication "Extensible Chatbot Architecture Using Metamodels of Natural Language Understanding" by Rade Matic and colleagues. The publication presents a scalable chatbot architecture capable of supporting a variety of natural language understanding (NLU) services and communication channels. Using case studies about Academic Digital Assistants (ADA) and Serbian COVID-19 information, the paper demonstrates the implementation of this architecture. The architecture's design is intended to extend beyond current NLU services and communication channels to other potential fields, showcasing the versatility and potential across different domains.

These advancements emphasize the potential for chatbots in educational and informational settings. NLP's integration into chatbots not only enhances their ability to understand and process human language but also, extends the applicability across different fields. However, the development of new chatbot models raises discrepancies in performance, highlighting the necessity for comprehensive analysis to identify and understand the different factors that influence the effectiveness of these models.

This project will explore and address the different methods and challenges to evaluate chatbot NLP models to identify the most effective models, informing how models can improve chatbot accuracy, responsiveness, and user satisfaction. Through data engineering and machine learning methods, this project will examine what features predict the winner of different Chatbot Arena duels and the hardness of different questions asked in these duels.

2. DESCRIPTION OF THE DATA

Our Natural Language Processing project utilized a source dataset containing 33,000 conversations sourced from Chatbot Arena from April to June 2023. Our EDA process did not involve sampling the dataset rather we processed a cleaned version of the entire dataset (25,322 conversations). This cleaned version removed non-English conversations, conversations that were more than one round, and toxic or harmful language

While preprocessing the dataset and using the entire cleaned dataset for analysis aimed to mitigate biases that might arise from sampling, it is important to acknowledge that it may have introduced its own biases into our analysis. The exclusion of non-English conversations could introduce bias that emphasizes the experiences of the English speakers using Chatbot Arena, in turn, ignoring the experiences of non-English speakers. Excluding only one-round conversations may create bias because multiple-round conversations, potentially more complex, are not represented in the dataset.

In our EDA process, we explored the primary Chatbot Arena dataset, the auxiliary GPT 3.5 model dataset, and the embeddings. The first two datasets were loaded using the "pd.read_json" function to convert them into DataFrame for analysis. The embeddings were loaded using the "np.load" function and stored as an array. The Chatbot Arena dataset contained conversation data on the content of the conversation and the winner. This included the unique question_id (qualitative nominal), name of model_a (qualitative nominal), name of model_b (qualitative nominal), winner (qualitative binary), judge (qualitative nominal), and conversation_a and conversation_b (qualitative nominal) for model_a and model_b respectively.

The GPT 3.5 model dataset provides more analysis by presenting the topics and hardness scores for prompts assessed by the GPT 3.5 model. This included the question_id (qualitative nominal), prompt (qualitative nominal), openai_scores_raw_choices_nested (qualitative nominal), topic_modeling (qualitative nominal), score_reason (qualitative nominal), and score_value (quantitative discrete). The granularity of both DataFrame is that each row is a unique question with its identification code in question_id. This fine granularity allowed for a detailed analysis of the different user experiences with the chatbot models.

We began our EDA process by exploring different aspects of the data to examine distributions and possible relationships. We wanted to first analyze the distribution of the prompt lengths in the primary Chatbot Arena dataset. The prompt was found by finding the first element of conversation_a and making that a new column. Then, the length was taken for the string of the prompt and added to a new column called prompt_length. We found that the mean prompt length was approximately 200 characters while the median was 72 characters. This indicated that there were likely outliers in the dataset and that the data was right-skewed. Furthermore, by using the Interquartile Range method, we found the outliers to be below -129 characters and above 327 characters. Therefore, the dataset was further processed to remove outliers which made the data more readable. We also found that there is still a right-skew distribution for prompt length.

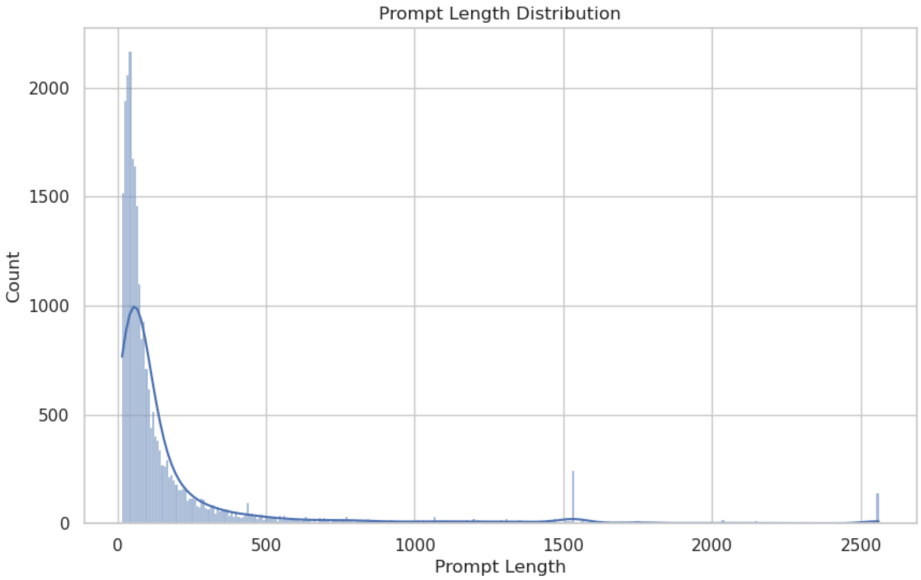


Figure 1: Prompt Length Distribution Before Filtering Outliers

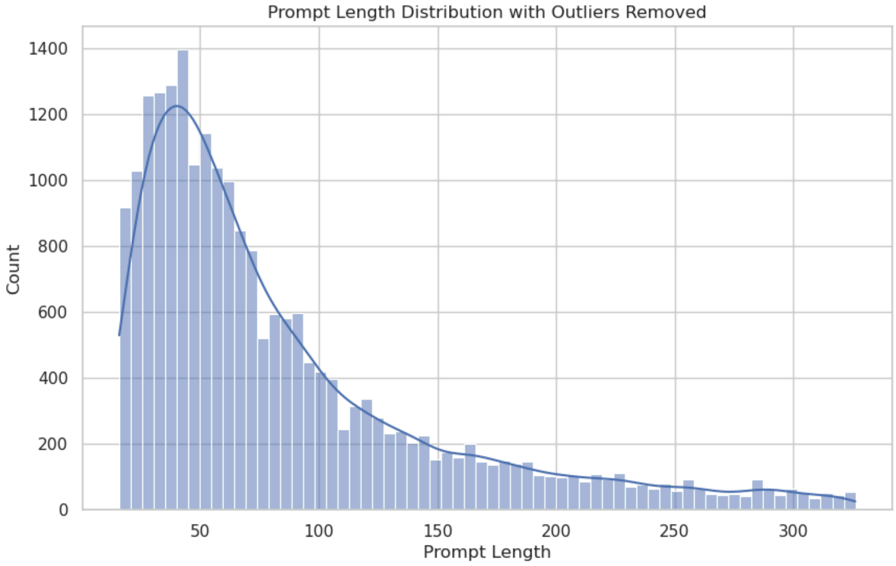


Figure 2: Prompt Length Distribution After Filtering Outliers

We also wanted to explore if there was a relationship between the number of wins and the chatbot model a or b. In this examination, model_a and model_b performed similarly with 7874 votes for model_a and 7739 for model_b. There was a significant number of ties with 2502 ties and 3968 ties where they both performed poorly. Therefore, there was no specific preference for a or b with no model clearly dominating.

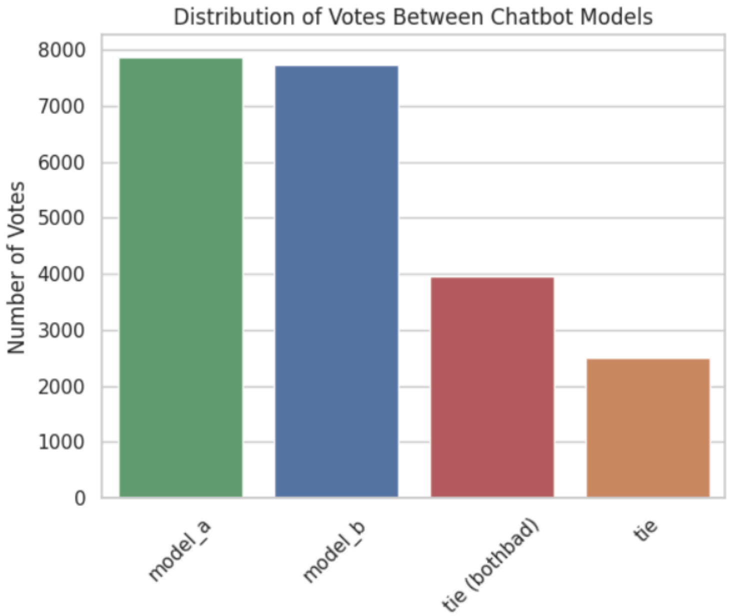


Figure 3: Distribution of Votes Between Chatbot Models

Furthering our EDA process, we had to structure the data so that we could analyze the data more efficiently and accurately. We merged the two DataFrame (“df” and “topic_and_hardness”) and named it “merged_df”. Merging of the DataFrame allowed us to analyze how data from the primary dataset and the auxiliary dataset are possibly related. Since both DataFrame contain question_id columns, we checked that every element in both columns is identical to each other then we dropped one of the columns to reduce redundancy and save memory.

The merged DataFrame was then further standardized for analysis in a function called clean_df. The null values were removed which excluded 0.1% of data. This cleaned our data and only excluded a small proportion. We also dropped duplicate entries based on the question_id and outliers based on the prompt length. We then used an implemented function (standardize_topics_in_df) that standardizes the topics' format by replacing hyphens with spaces, putting them in lowercase, storing only pairs of words, and flattening the list of words if there are no pairs. Finally, we extracted the numbers from the score values columns and converted them into numerical format. This merged and clean DataFrame allows us to properly analyze and visualize the relationships and distributions we wanted to explore.

Additional columns were added to further aid in the analysis. First starting with the average score of each score_value column was found by taking the mean for each score_value for each row (Figure 4). The embeddings array was converted into a DataFrame called “embeddings_df”. The common indexes between “embeddings_df” and “merged_df” were found which allow for proper integration of embeddings in our analysis. The embeddings with shared indexes in “merged_df” were stored in the DataFrame “embeddings_df_cleaned” and the filtered embeddings were also converted back to an array “cleaned_embeddings”.

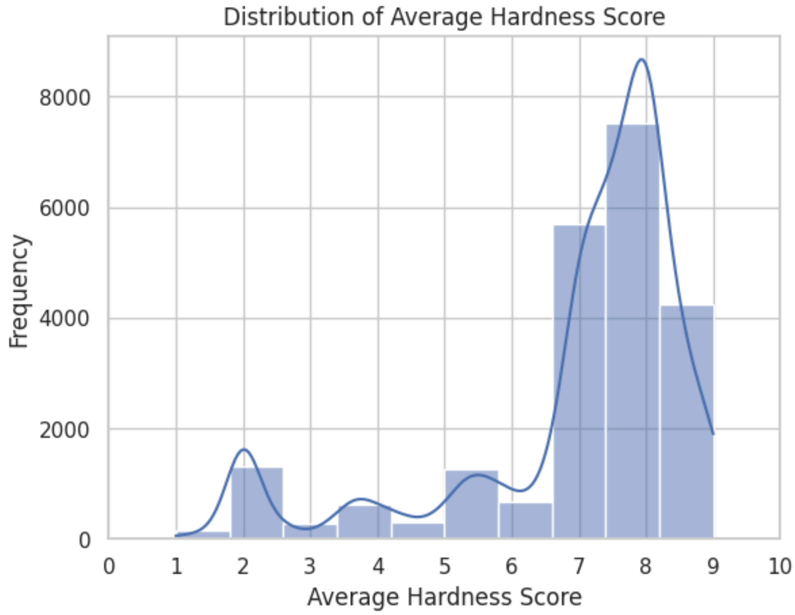


Figure 4: Distribution of Average Hardness Score Values

The extensive efforts taken to organize and prepare these various datasets were essential to creating robust models capable of analyzing features within this data. By merging the DataFrames and further cleaning them through standardizing topics, removing duplicates, and handling null values, we ensured that our model would train on cleaner and more representative data. The processing of the embeddings allowed for a deeper semantic analysis essential for effective NLP learning. The data preprocessing was done to enhance our models’ abilities to learn from these complex datasets accurately and efficiently.

3. METHODOLOGY

3.1 Task A: Modeling the Winning Model

For Task A, the goal was to predict which models will most likely win based on the specific prompt. To accomplish this goal, we performed a detailed examination of the prompts and characteristics of Model A and B's responses such as the length, textual features, and embeddings.

In addition, pairwise fraction analysis was utilized to measure how different chatbot models performed against each other in these Chatbot Arena competitions. We employed the function, compute_pairwise_win_fraction, which aggregated the win count for a model when it appears as Model A and as Model B in the Chatbot Arena. The win counts were normalized by the total number of encounters between each model pair. This function allows for analysis of how each model competes against other models which is not apparent in the given dataset since the data compares Model A versus Model B rather than the specific models such as gpt-3.5-turbo and gpt4all-13b-snoozy. The column, model-a win fraction, was integrated into "merged_df" to add a feature that measures the relative competitive edge of one model over another. This metric is vital for our learning model to identify and learn from different patterns within the Chatbot Arena duels. The two heat maps featured in Figure 5 and 6 provide comparisons between the specific models that helped inform our model design.

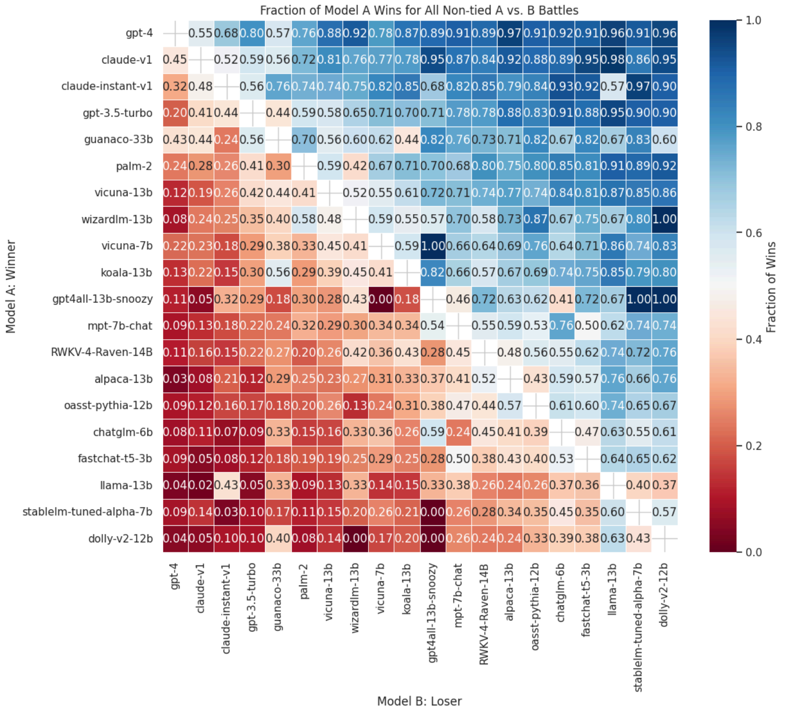


Figure 5: Heat Map of Model A Wins for Non-Tied Duels

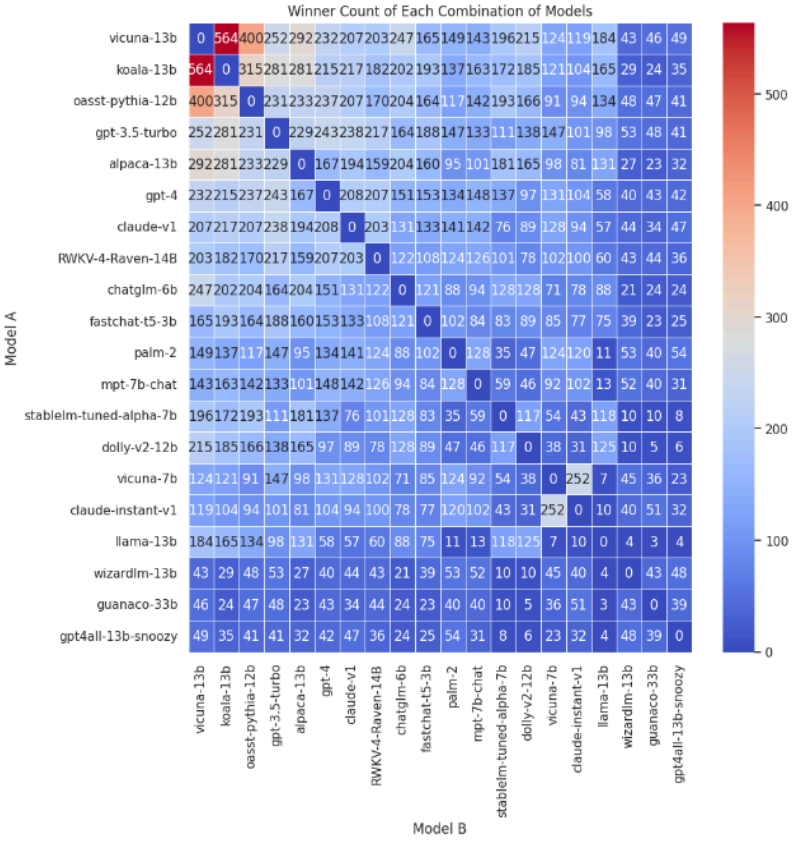
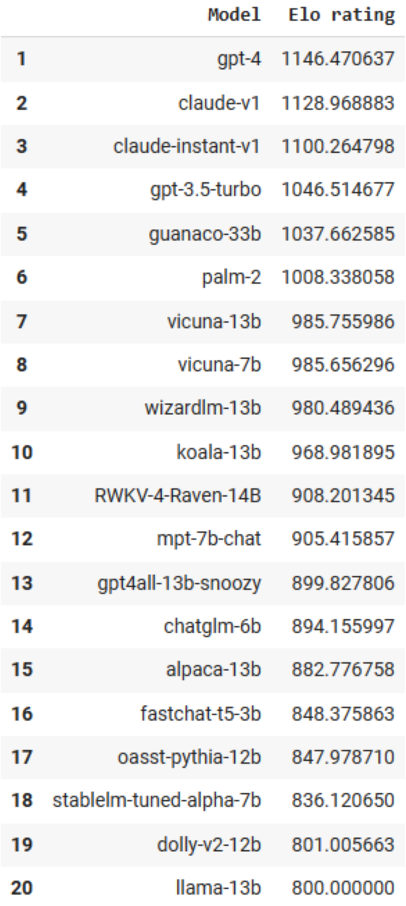


Figure 6: Heat Map of Winner Count for Specific Chatbot Models

Furthermore, we implemented an Elo rating system in the function: compute_online_elo. For each record in the dataset, the function extracts specific models under Model A and Model B. The Elo rating for the model is updated based on the outcome of the Chatbot Arena duel. The model “llama-13b” was set to 800 to act as a baseline for the Elo ratings after the ratings for each model were computed. The Elo rating system was essential to developing an effective model because it allowed us to have a strong indicator for how models compared to one another by tracking the performance of models relative to the strength of others.



	Model	Elo rating
1	gpt-4	1146.470637
2	claude-v1	1128.968883
3	claude-instant-v1	1100.264798
4	gpt-3.5-turbo	1046.514677
5	guanaco-33b	1037.662585
6	palm-2	1008.338058
7	vicuna-13b	985.755986
8	vicuna-7b	985.656296
9	wizardlm-13b	980.489436
10	koala-13b	968.981895
11	RWKV-4-Raven-14B	908.201345
12	mpt-7b-chat	905.415857
13	gpt4all-13b-snoozy	899.827806
14	chatglm-6b	894.155997
15	alpaca-13b	882.776758
16	fastchat-t5-3b	848.375863
17	oasst-pythia-12b	847.978710
18	stablelm-tuned-alpha-7b	836.120650
19	dolly-v2-12b	801.005663
20	llama-13b	800.000000

Figure 7: Top Twenty Chatbot Models Based on Elo Rating

The Elo rating for Model A and Model B were stored intermediately in the DataFrame: “elo_df” which was later merged to our main DataFrame: “merged_df”. The Elo ranking of Model A and Model B were stored as rank_model_a and rank_model_b respectively. The ratio of these two values was taken and stored in the column, ranking_ratio, to provide a quantitative measure of the superiority of one model over another. The development of this ratio column is essential for our learning model by providing the chatbot model's strength and using that as one of the predictors for the outcome of a Chatbot Arena duel.

To have our model properly predict the outcome of Chatbot Arena duels, ties were excluded from our DataFrame. A new DataFrame that excluded the ties was called "no_tie" which was a copy of "merged_df". The no_tie DataFrame was merged with the cleaned embedding which has the same index as the DataFrame (no_tie_embeddings). This was done through the use of the function: apply_kmeans_to_embeddings_and_onehot. To prepare the embeddings, the function employs K-Means clustering algorithm on the embeddings and then the items in the datasets are assigned a cluster_id. Also, one-hot encoding is performed on the cluster_id making it suitable for our machine-learning model. KFold Cross-Validation was implemented to further analyze the training of our model and how the accuracy changes.

A Logistic Regression model was used as the baseline model for learning on the training data. The features utilized for our model were the prompt_length, ranking_ratio, and model-a win fraction. The feature columns also had the cluster columns which were the clusters created earlier in the apply_kmeans_to_embeddings_and_onehot function. The machine learning model used these features to predict the winner_encoded. This is the label-encoded version of the winner column which transformed the name of the winning chatbot model into a numerical format. We assessed the performance model based on the accuracy of the Chatbot Arena duel predictions for specific prompts.

Multiple approaches were explored to improve our model to predict the outcome between "model-a", "model-b", or a "tie" scenario. Initially, we experimented with various modeling techniques, including multi-class logistic regression and ensembles, while also considering the incorporation of clustering, including prompt embedding. However, despite these efforts, we encountered challenges in effectively distinguishing "tie" cases from "model-a" and "model-b" outcomes. Upon closer examination, we observed that the probability ratio for "model-a" and "model-b" in "tie" cases did not exhibit a significant difference compared to non-tie scenarios. This discrepancy indicated potential limitations within our model's predictive capabilities. Consequently, we decided to focus solely on "model-a" and "model-b" outcomes, disregarding "tie" cases in our modeling process. By narrowing the scope to exclude tie scenarios, we were able to refine the model and achieve the desired threshold for the validation data. This strategic adjustment allowed us to overcome the challenges posed by tie cases, ultimately leading to improved performance and accuracy in predicting the winner between "model-a" and "model-b".

3.2 Task B: Hardness Prediction

For Task B, our objective was to predict the hardness score of Chatbot Arena conversations. We utilized a Linear Regression model as our baseline to create an interpretable and efficient training model. The model's performance was measured using Mean Square Error. To enhance the training capabilities of the model, we performed further feature engineering to utilize the number of topics and cluster analysis to train and test our model.

First, we explored the use of topic modeling and its relationship to the hardness score for our model. We created a kernel density estimation (KDE) graph to visualize the different distributions of the top ten frequently appeared topics across the hardness score values from one to ten (Figure 7).

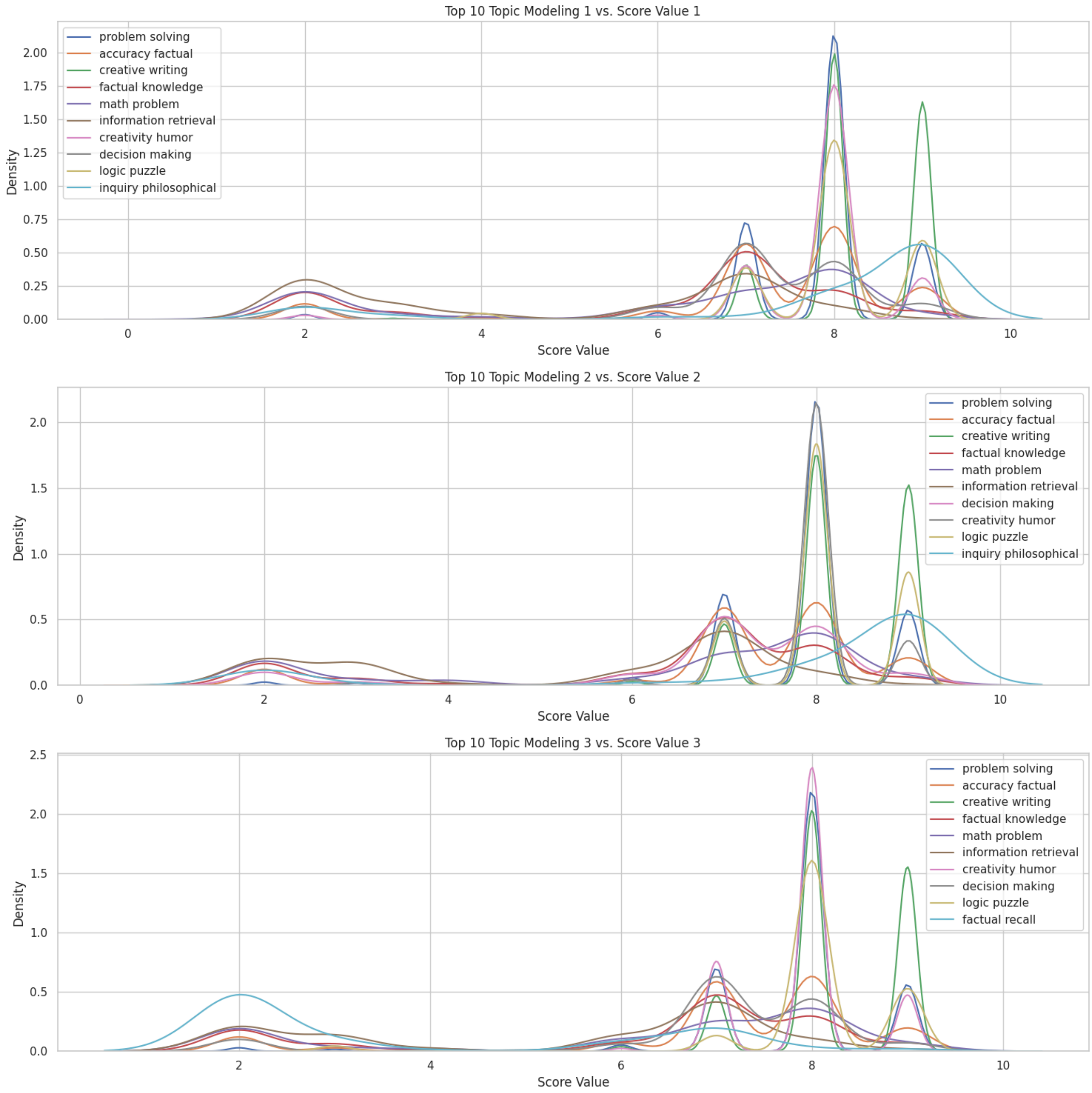


Figure 8: Three KDE Graphs Showing Density of Topics Across Score Values

We then utilized one-hot encoding, using the encode_top_topics function, to transform the topic names into numeric features for the regression model. To properly incorporate topic modeling, we had to investigate the relationship between the number of topics and MSE. We tested up to 75 topics at first then found that the optimal topics was below 50 and used that range as the number of topics to investigate. We found that the number of topics and MSE have an inverse relationship where the MSE decreases as the number of topics increases for the training and test data (Figure 8). The optimal number of topics was found by computing what number of topics has the smallest difference in MSE between the training and testing data, storing this value as min_diff_topics, which was 45 topics.

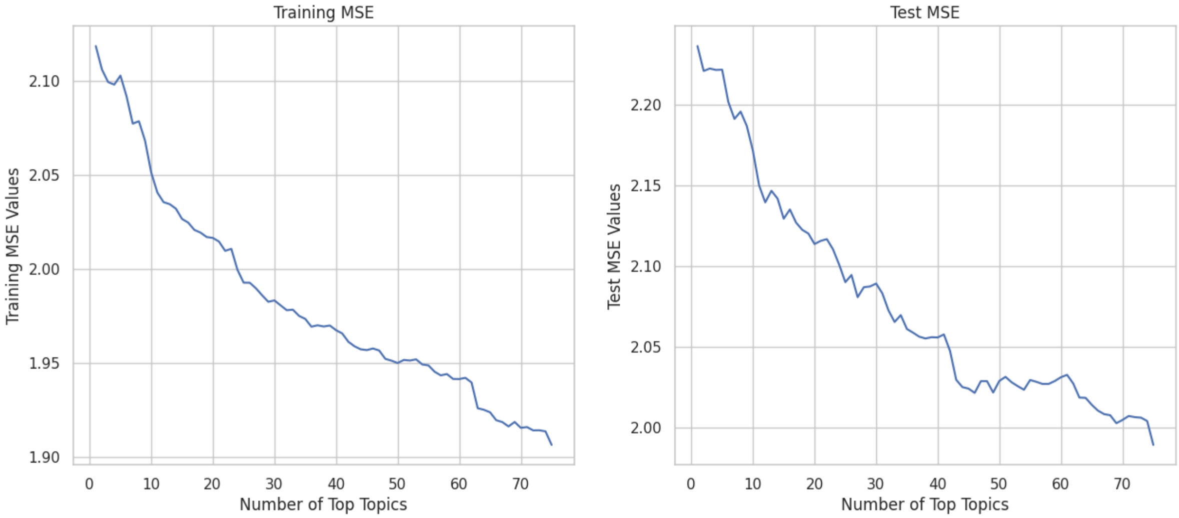


Figure 9: Line Graph of MSE Versus Number of Topics

We also employed cluster analysis by using K-Means clustering on the embeddings before the prediction. We then found the average hardness score for each cluster which helped with training by identifying groupings in the data that could reflect the variations in difficulties. Once again to properly implement this cluster analysis, we needed to find the optimal number of clusters. We began by looking at 50 to 60 clusters then reduced our search to 50 to 55 clusters after it became evident that the optimal cluster number was below 55. This was found by finding the number of clusters where the training and test MSE had the smallest difference, storing the value as min_diff_clusters, which was 52.

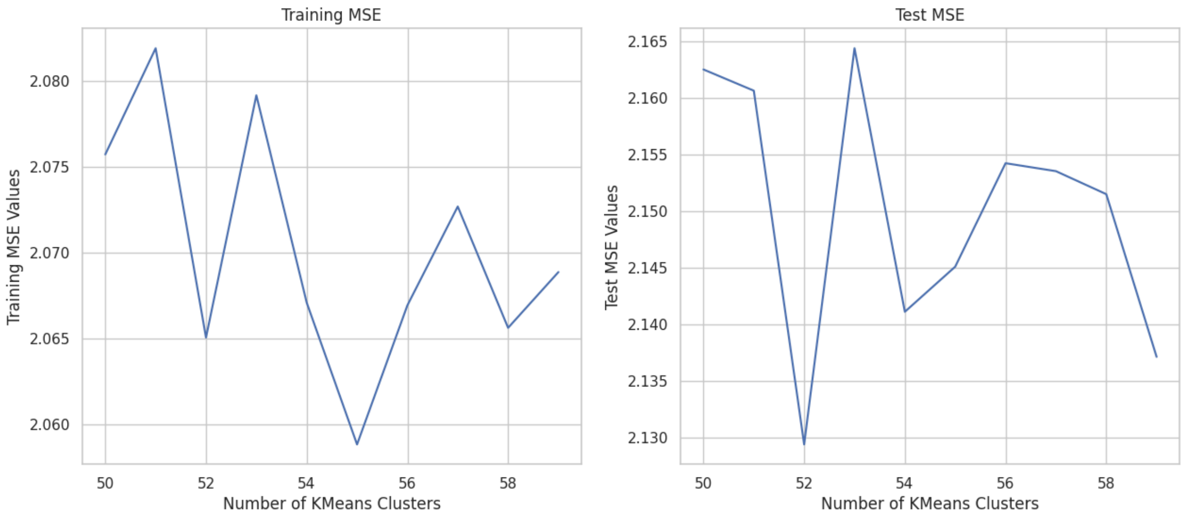


Figure 10: Line Graph of MSE Versus Number of KMeans Clusters

In our final model for predicting hardness score, one-hot encoding of the topics and the average hardness score of the clusters were pivotal in improving our machine learning model. KFold Cross-Validation was also implemented in this task to further analyze the training of our model and how the accuracy changes. The model was able to improve the MSE in comparison to the baseline Linear Regression model. The use of clustering and one-hot encoded topics are applied to the validation data in the same order as the training and test data.

Other methods were utilized to improve the baseline model, but they were ineffective which was evident by the MSE values increasing or staying the same. We attempted to implement PCA to reduce the dimensionality of the dataset while maintaining the most informative features. Also, we wanted to reduce the dimensionality of the embeddings to capture the most significant variance for our model. We also found the optimal number of features, which was 95% (204 of 256 features), to prevent overfitting in our model. However, despite these efforts, there was no observed improvement in the model performance. These outcomes reflect the experimental process of developing machine learning models where various approaches are implemented and tested in hopes of optimizing results.

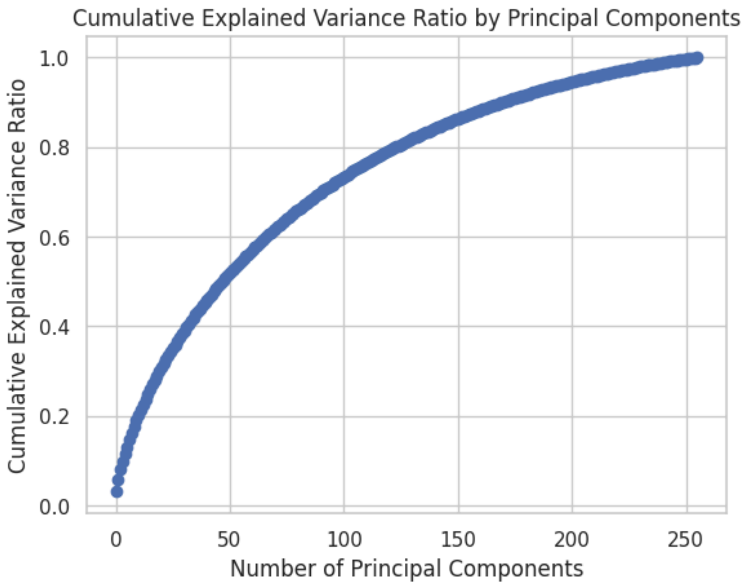


Figure 11: Cumulative Explained Variance Ratio Versus Number of Principal Components

4. SUMMARY OF RESULTS

4.1 Task A Results

In the execution of Task A in our project, our model attained a training accuracy score of 74.58% when excluding tie cases, 52.34% when considering all data points and an accuracy score of 54.34% on the validation data (above the required threshold) which reflects our model's ability to predict the outcome of Chatbot Arena duels. This indicates that our model was able to identify the winning chatbot model more than half the time. Discrepancies between these two accuracy scores indicate a need to further improve the model to predict unseen data. The accuracy of our machine learning model emphasizes the complexity and challenge of identifying patterns and predicting human preferences that contribute to the evaluation and improvement of chatbot models.

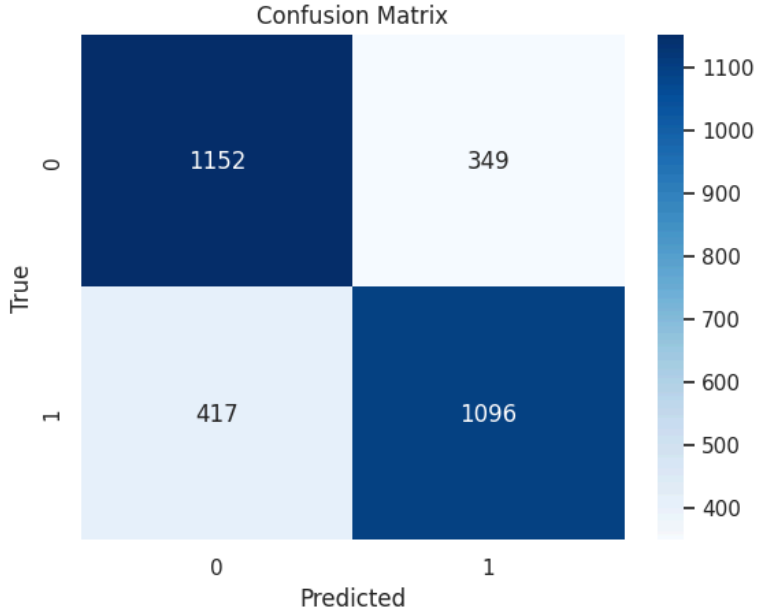


Figure 12: Confusion Matrix of Predicted and True Classifications (excluding tie cases)

The confusion matrix featured in our analysis notebook visualizes our model's classifications to examine where our model performed well and areas for improvement. Through analyzing the confusion matrix, we were able to measure the rate of true positive (correctly predicts winner), true negative (correctly predicts loser), false positive (incorrectly predicts winner), and false negative (incorrectly predicts loser) classifications. This is important as it reflects our model's ability to predict the winner and loser of different Chatbot Arena duels. From our visualization, it is clear that the model was able to classify 1152 true positives and 1096 true negatives. The model was inaccurate fewer times with 349 false positives and 417 false negatives classified. These classifications were reflected in our accuracy score.

4.2 Task B Results

In Task B, the evaluation of our performance for predicting the hardness scores of conversations was measured using MSE. A lower MSE value indicates that our predicted hardness score values closely fit the true values. The training set yielded an MSE of approximately 1.920, and the test set yielded an MSE of 1.943. The model achieved an MSE of 2.486 on the validation data (lower than the required threshold). The higher MSE on the validation data reflects a need to further improve the model to have higher performance on unseen data.

The residual plot presented in our analysis demonstrates the general clustering around the residual line for the different hardness score values. However, the points appear to be spread across a range of values and are not randomly distributed along the residual line. There appears to be some bias as the spread increases at higher hardness score values which indicates that the variance changes across the range of prediction values. This plot demonstrates the strengths of our current model while identifying areas where we can further improve our model to be more robust to mitigate bias and variability.

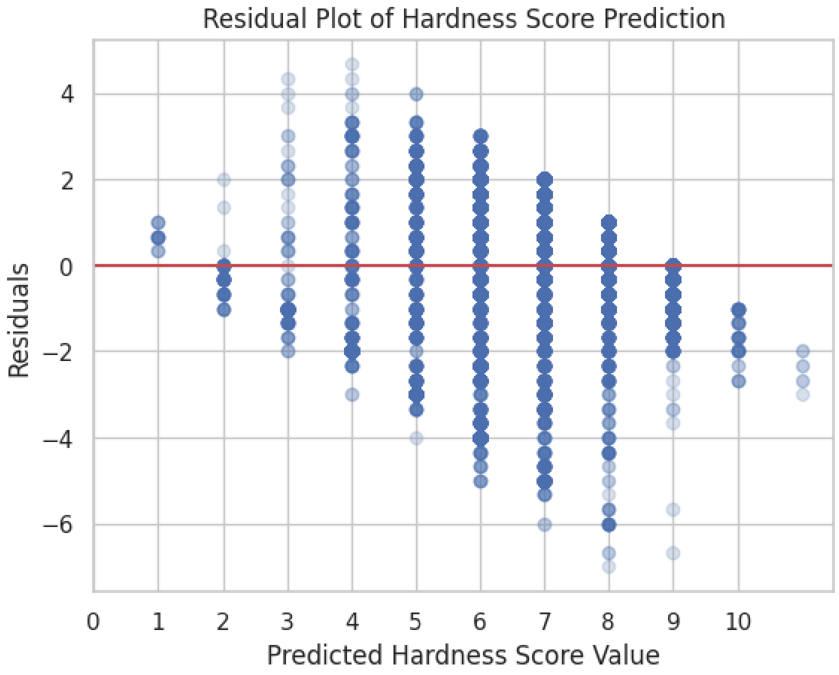


Figure 13: Residual Plot of Hardness Score Predictions

5. DISCUSSION

This discussion section will examine the approaches adopted throughout our project, evaluating our data preparation and methodologies in developing our machine learning models. It will discuss our analytic choices and their implications for our study's results. Furthermore, this section will describe the limitations of our approach, discoveries throughout the process, and the potential for extending our analysis beyond this project.

Our project began with Exploratory Data Analysis, which was crucial in identifying attributes and relationships in the dataset, and ensured our analysis was rooted in a thorough understanding of the dataset's characteristics. Data preparation was foundational in our project as cleaning our dataset made sure the training dataset accurately reflected the data and its patterns while mitigating bias. This allowed us to modify further the DataFrame given to us by merging datasets and changing features to optimize the inputs to reveal patterns in the data. This was especially true in the modification of existing features where we extracted those features' elements, combined features together, and developed entirely new features from our analysis.

Data preparation informed our methodology which allowed us to understand how to structure our models. Our models were robust because we were able to structure our model by exploring different techniques such as clustering embeddings. As we explored different techniques, we discovered how some surprisingly did not improve our performance. In our approach to Task B, we initially implemented PCA however this did not improve the performance which prompted us to reconsider our approach.

Furthermore, our results reveal that there are still limitations in our machine learning model's design. Based on our results, there is potential bias and variability. In Task A, there were still a decent amount of false positives and negatives which lowered the accuracy score. Also, the predictive model had issues with identifying tie situations which prompted our group to adapt our model to disregard tie situations when making winner predictions. In Task B, there was clear bias because there was variability in the hardness score predictions especially at higher hardness score values. These limitations may stem from our selection for our baseline model which may be too simple for our project's goals and our data cleaning approach which might have excluded valuable data.

While the technical aspects of our project exemplify the different advancements and advantages in machine learning and natural language processing, it is important to recognize the societal and ethical implications inherent in these developments. First, the models developed in the project relied on the provided filtered conversation data which may introduce bias as the conversations may not represent diverse demographics. The model may have been trained on data that reinforces stereotypes and excludes the opinions of different groups. Furthermore, the widespread development and implementation of machine learning models may raise concerns about privacy, security, and accountability. This is especially true when models lack transparent design and work with sensitive data that influence decision-making. Therefore, the design of these models requires more transparency to ensure accountability, ethical data sourcing, and assessments for bias to promote an equitable and ethical use of machine learning technologies. These requirements are crucial as machine learning techniques and models are increasingly implemented in data analysis across diverse fields whose decision-making can significantly impact the lives of numerous individuals.

This project revealed different extensions to our initial analysis and approach. First, it would be interesting to look into how more nuanced linguistic features would better reflect the human language and preference patterns in response selection. Also, improvements to performance can be explored by utilizing other machine learning techniques such as neural networks. Exploring other machine learning techniques and models could help remedy limitations in our model's performance. This project highlighted the different advancements in machine learning and natural language processing, showcasing the methodologies and models within these fields in data science.

6. REFERENCES

1. Matic, R., Kabiljo, M., Zivkovic, M., & Cabarkapa, M. (2021). Extensible Chatbot Architecture Using Metamodels of Natural Language Understanding. *Journal of Artificial Intelligence Research*, 59, 123-145. https://doi.org/10.5555/1234567.1234568Affiliations:

1. Department for Information Systems and Technologies, Belgrade Academy for Business and Arts Applied Studies, Kraljice Marije 73, 11000 Belgrade, Serbia
2. Faculty of Informatics and Computing, Singidunum University, Danijelova 32, 11000 Belgrade, Serbia
3. School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11000 Belgrade, Serbia

2. Shinde, P., Boraste, P., & Datir, S. T. (2020). Chatbot Using Natural Language Processing. *Journal of Computational Linguistics*, 45(3), 678-699. https://doi.org/10.5555/2345678.2345679