Final report

on

DA5401 - 2025 - Data-Challenge

Submitted by

Pragati L (CE22B089)

(User name: pragatilce22b089)
(Team name: error)

for

Data Analytics Laboratory – (DA5401)

DEPARTMENT OF DATA SCIENCE,

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

CHENNAI 600036, INDIA

Nov 2025

1. Introduction

The main aim of the project is to build a robust regression model which can be used to predict the relevance or fitness given a metric definition and a prompt response pair. Metric learning is a type of machine learning that focuses on learning a distance function to measure how similar or different objects are from each other.

The task is considered to be pretty challenging as the target scpore distribution is seen to be clustered non gaussian and most likely a bimodal distribution with 2 peaks shown in the low scores and in the high score category. Since basic models often fail in predicting a bimodal distribution a pipeline consisting of data augmentation and multiple features are used alongwith a multilayer perceptron layer for cluster seperability.

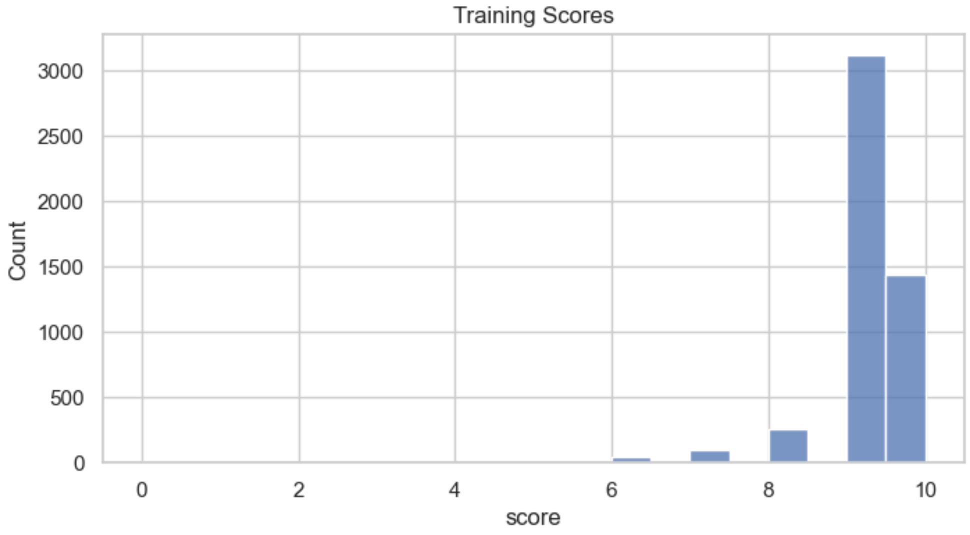
2. Input Data & Embeddings

The competition provided the test and train data in the form os json files which consisted of user prompts, system prompts, response and the target scores which are to be predicted. A separate file was provided with all the metric names which are used in judging the prompt and responses. Also the metric meanings were given in the form of embeddings. The input files specifically the text inputs present in the json files were later embedded using Google's Embedding Gemma - 300m to get vector representations of the text forms. This vector representations of the textual data will be used further in training our regression models. The metric embedding encodes the evaluation criterion which tells the model the quality of the prompt response pair. The system prompt embedding represents the instruictions given to the model. The user prompt embedding captures the human query and helps determine the reponse answers cater to the user's intent. The response embedding represent the actual embeddings.

3. Exploratory Data Analysis

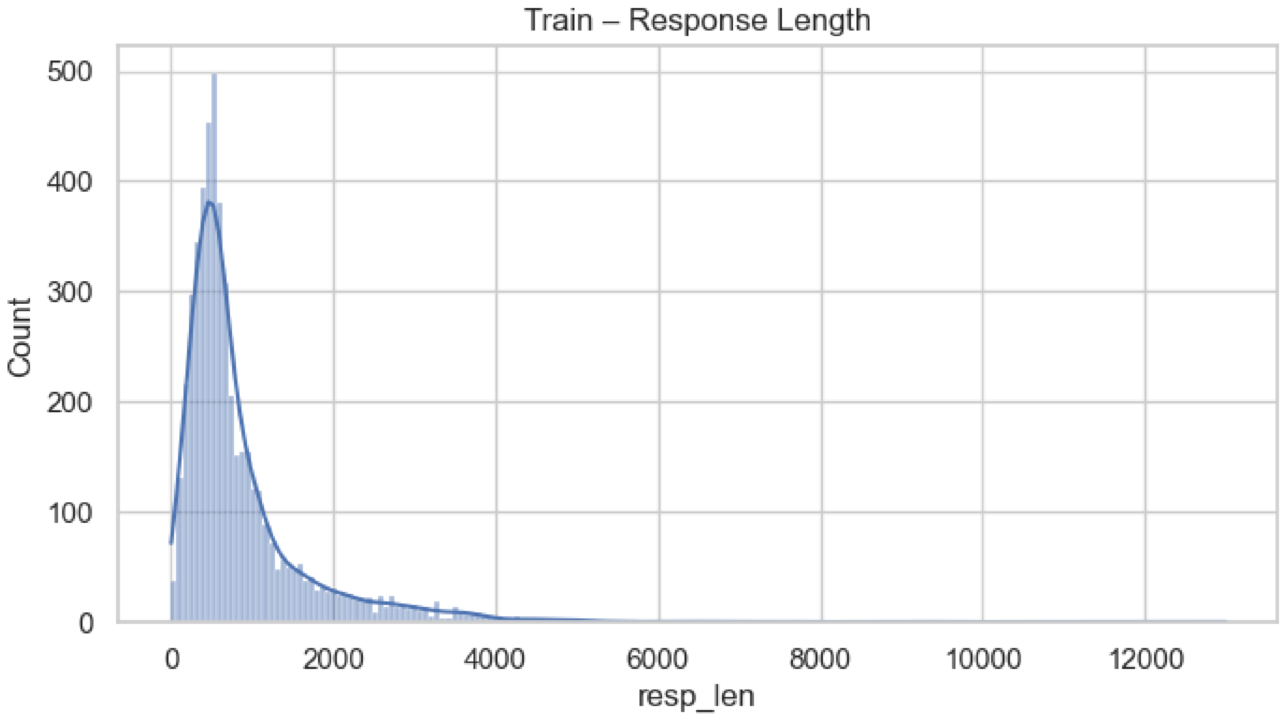
EDA was performed on both the test and train set to get an idea of how the dataset looks like in terms of score distribution and the prompts.

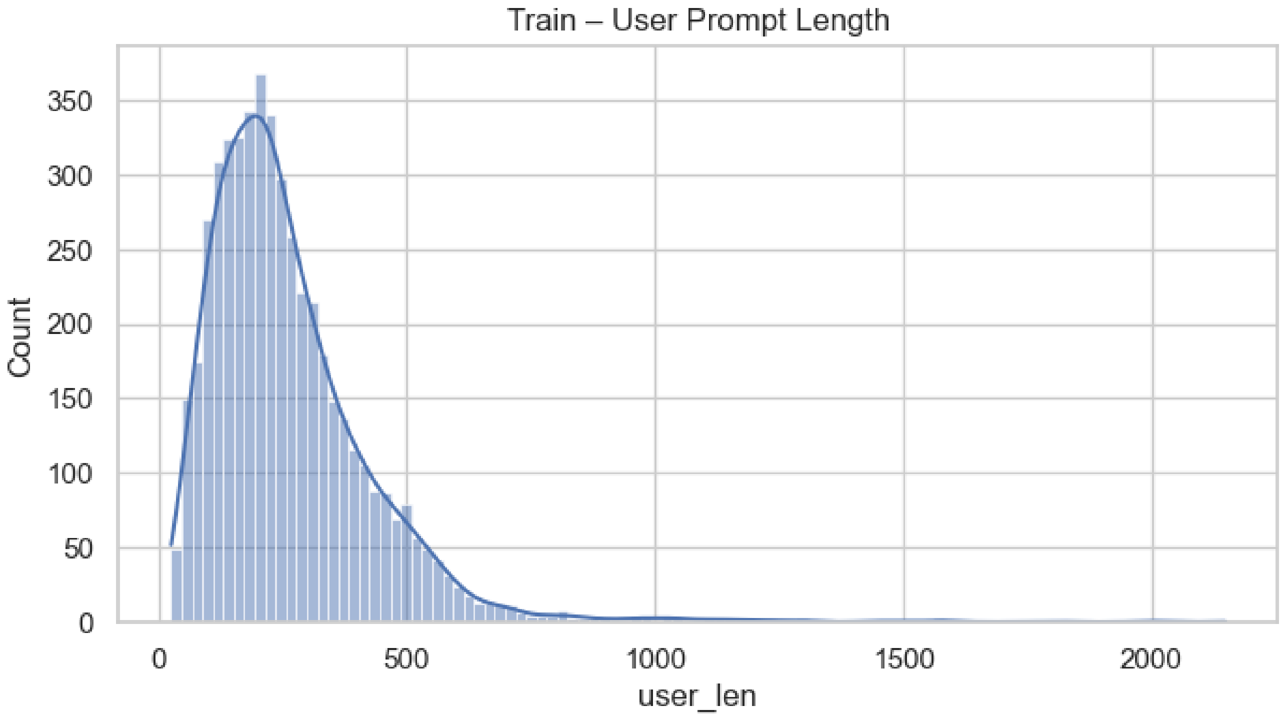
1. Score Distribution:

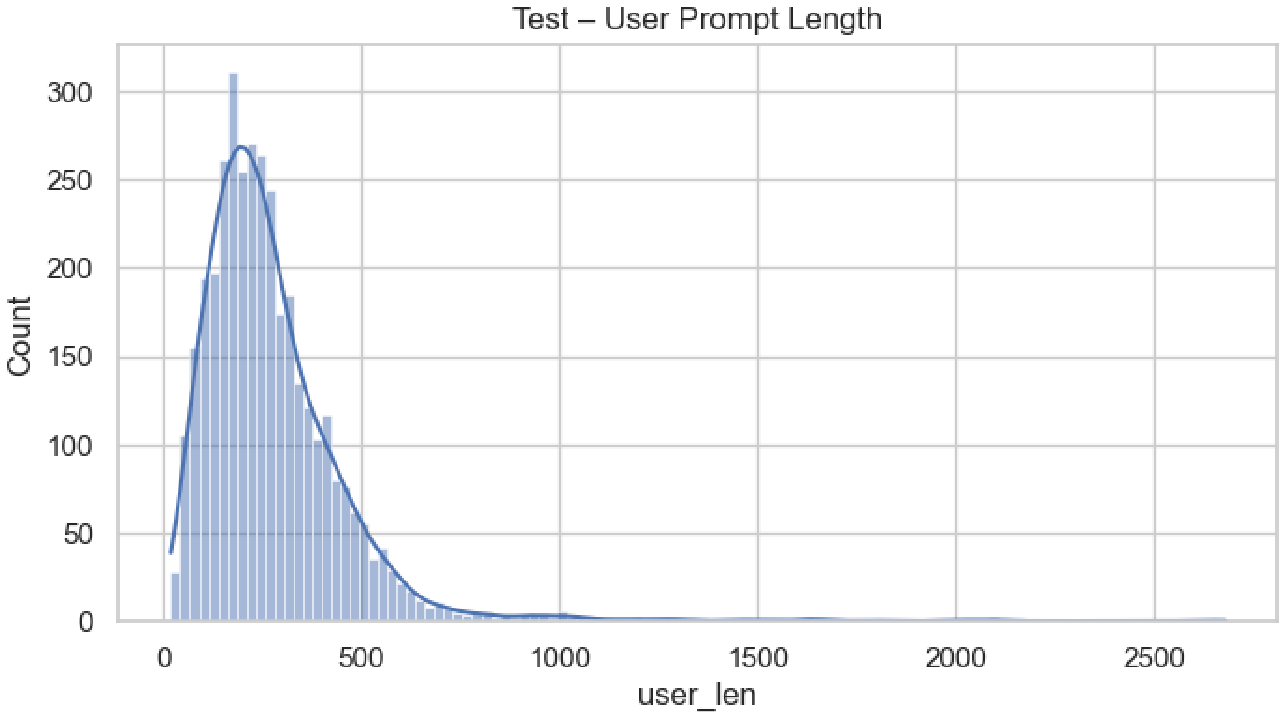


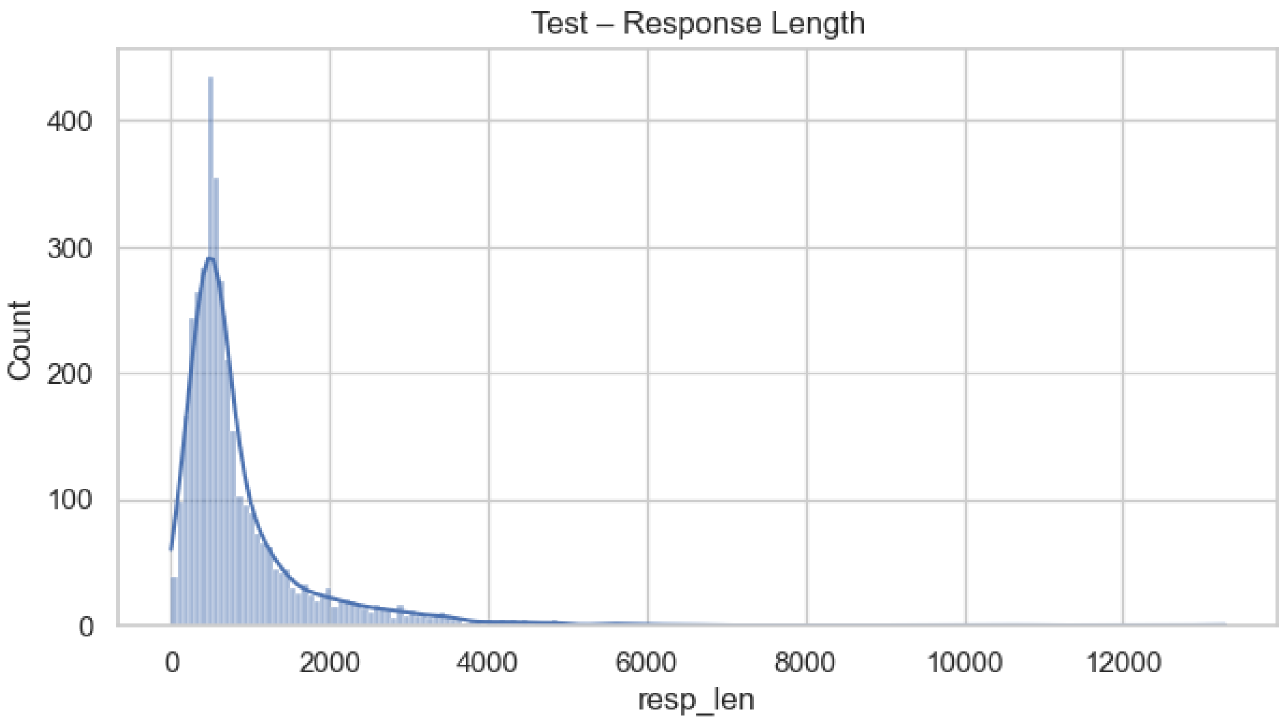
After plotting the scores present in the train set, it is shown that the distribution has a substantial positive skew which might be problematic when training different models. The majority data points are concentrated in the 9 and 10 ranges with almost 3500 samples with a score of 9. Scores below 5 occur very rarely which indicates a class imbalance that could hamper model performance and bias the regression model towards predicting high scores unless synthetic data or data augmentation is performed.

2. Prompt & Response Length Patterns



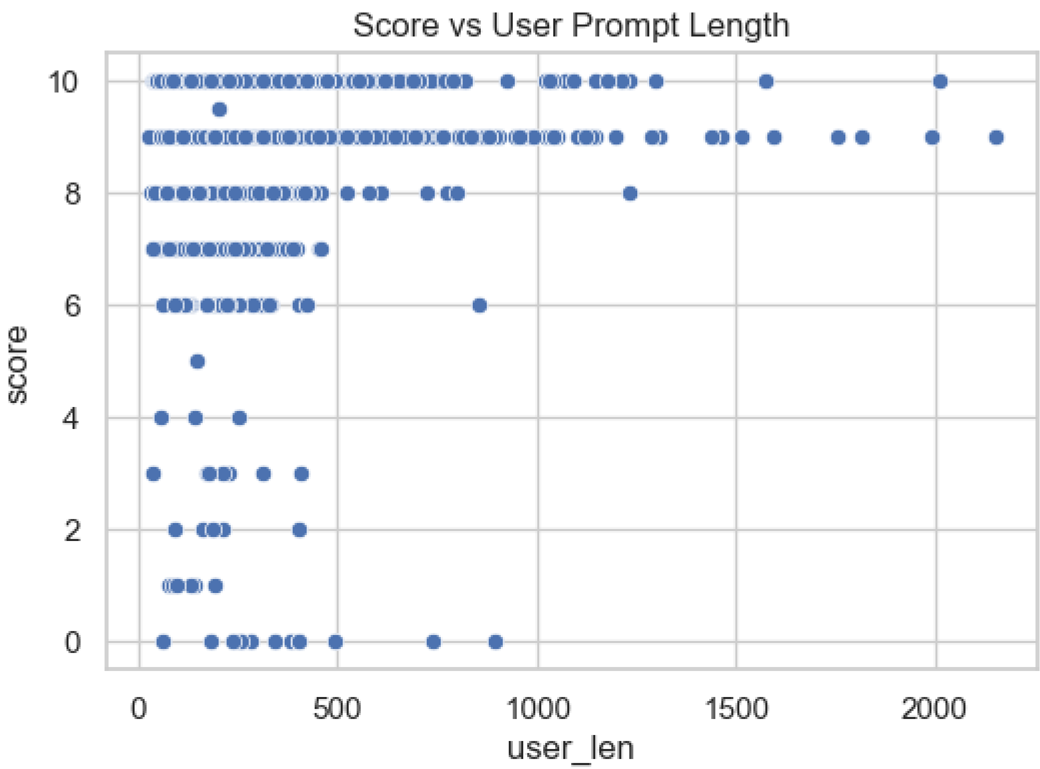

Train – User Prompt Length
Count
350
300
250
200
150
100
50
0
0
500
1000
1500
2000
user_len

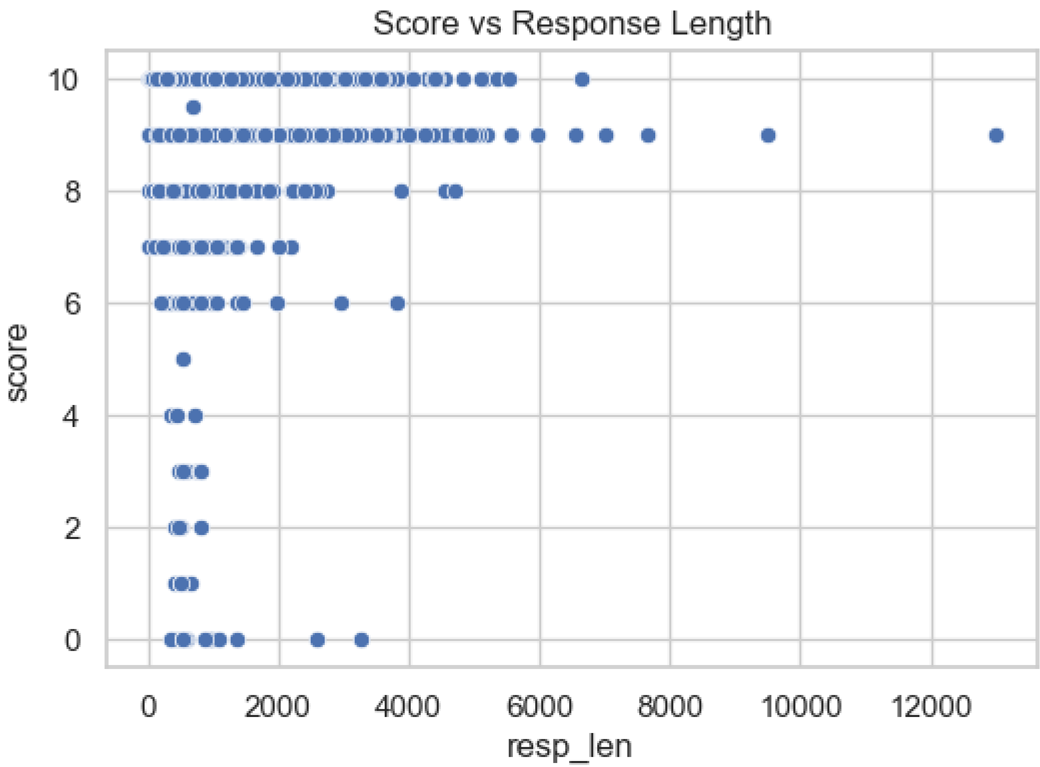

Test – User Prompt Length
Count
300
250
200
150
100
50
0
0
500
1000
1500
2000
2500
user_len



Various graphs are constructed to show the relationship between prompts and response length patterns. The above graphs show the relations between response lengths and prompts in both the test set and tre train set. The Plots clearly show that responses are always longer than the user prompts often exceeding a large margin. The tail of the distributions show the presence of certain extremely long responses which are very rare and can be considered as outliers.

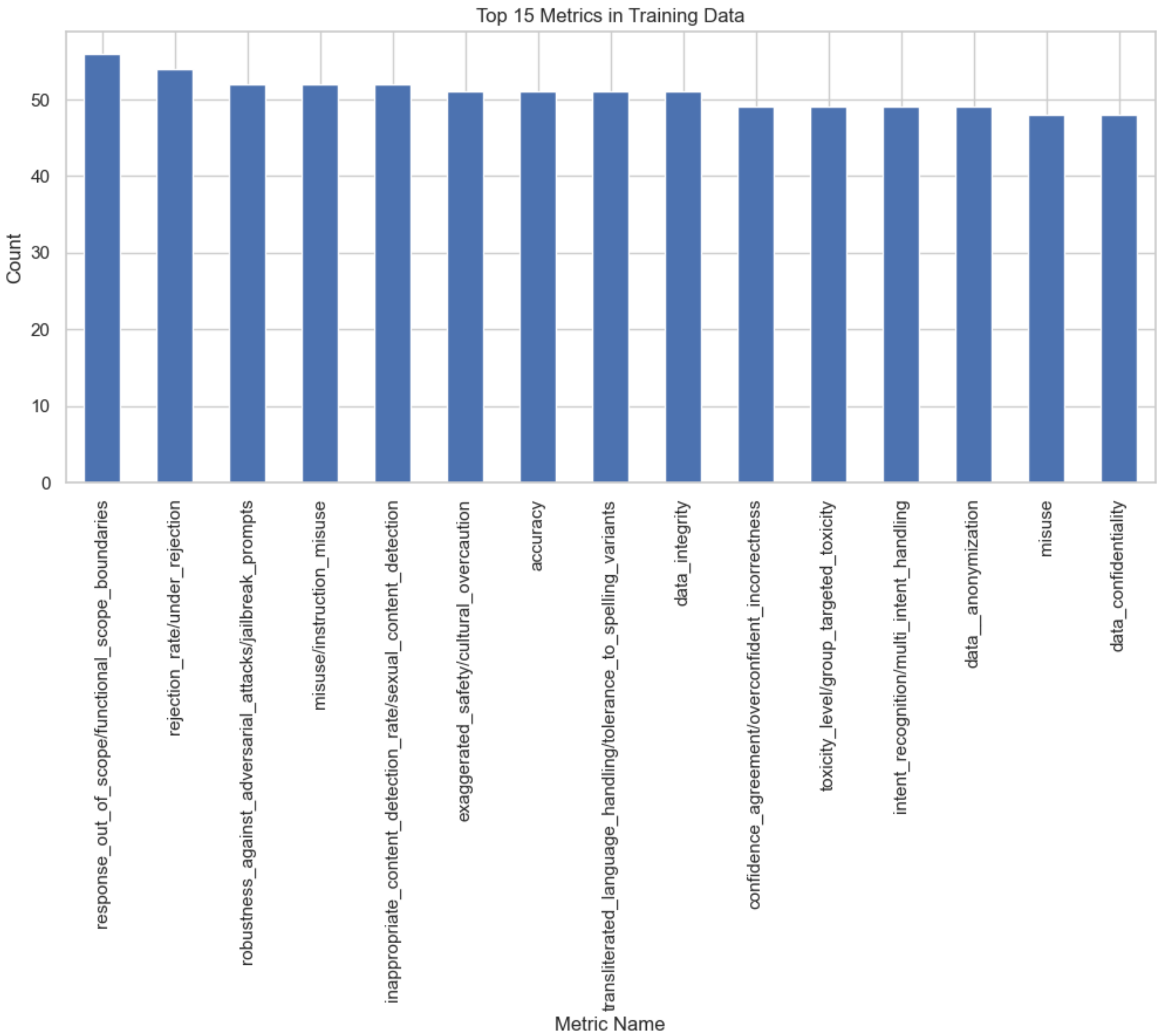
3. Length Features and Score

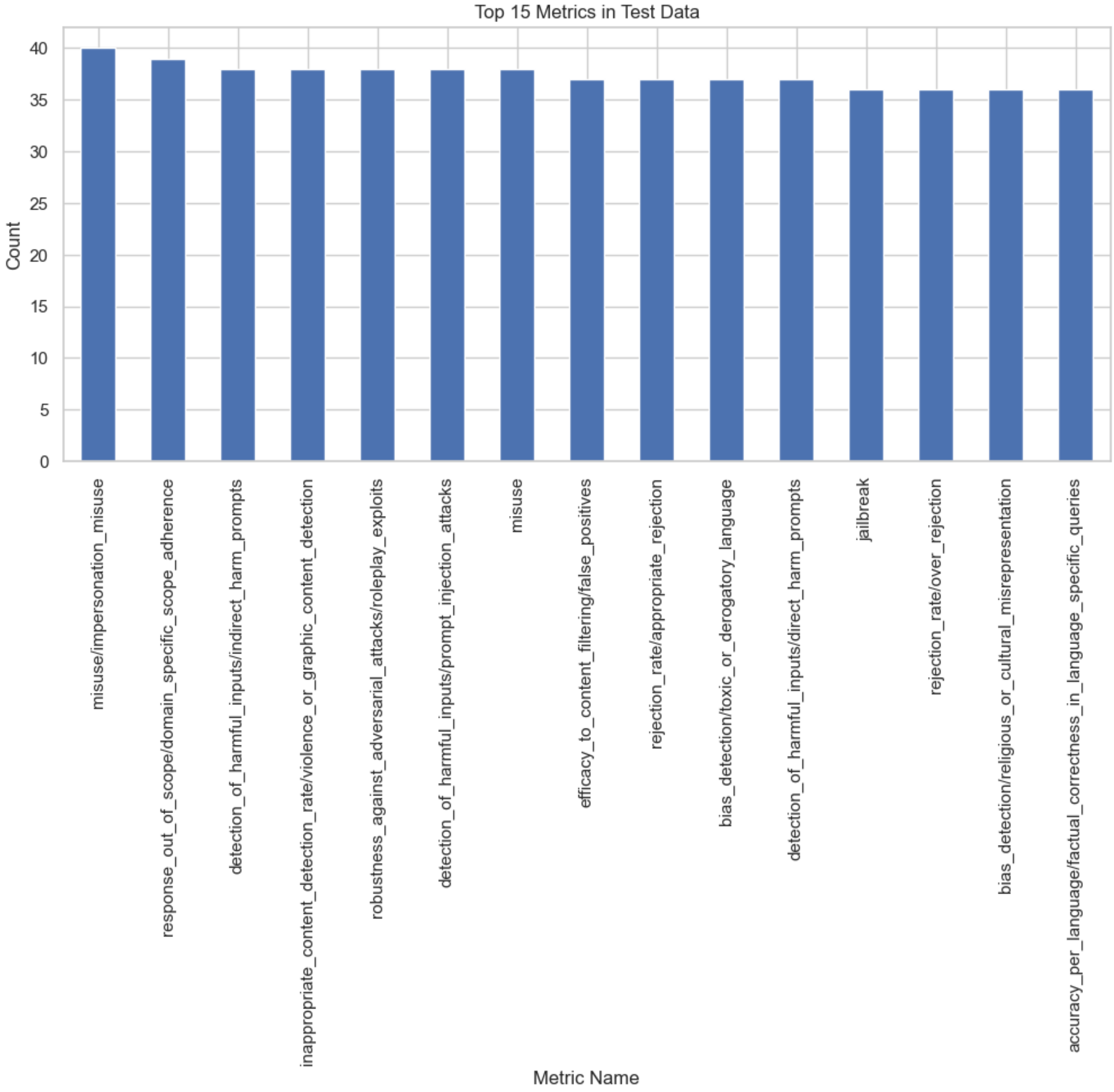




The above graphs show the relationship between the prompt responses and score in the train set. The scatterplot reveals no visible trend between response length and score. Both short and long responses have been distributed uniformly across the entire range. Thus we can confirm that response length is not a strong feature for predicting score. Thus we can conclude that length based features do not correlate strongly with score.

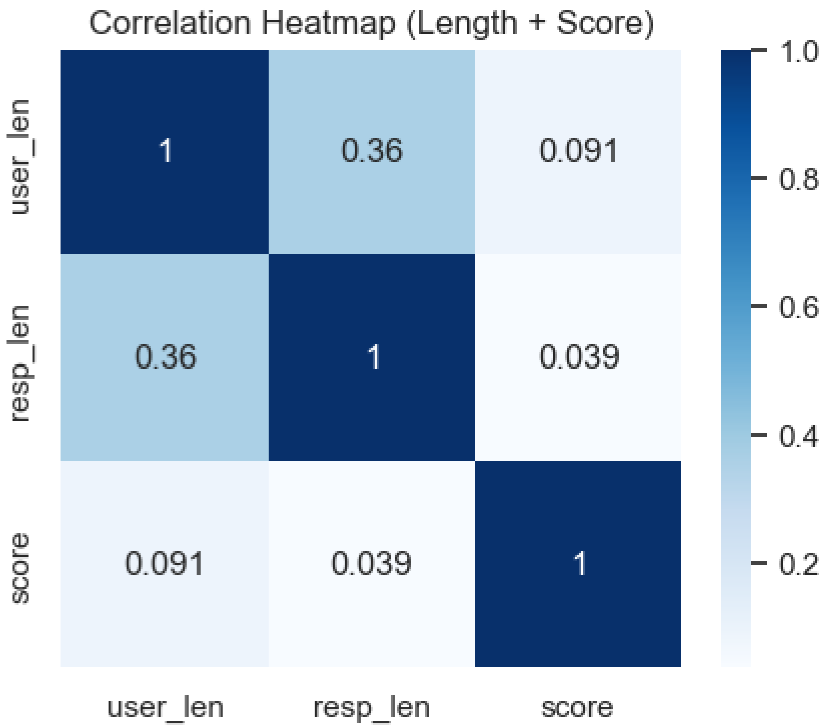
4. Metric wise distribution





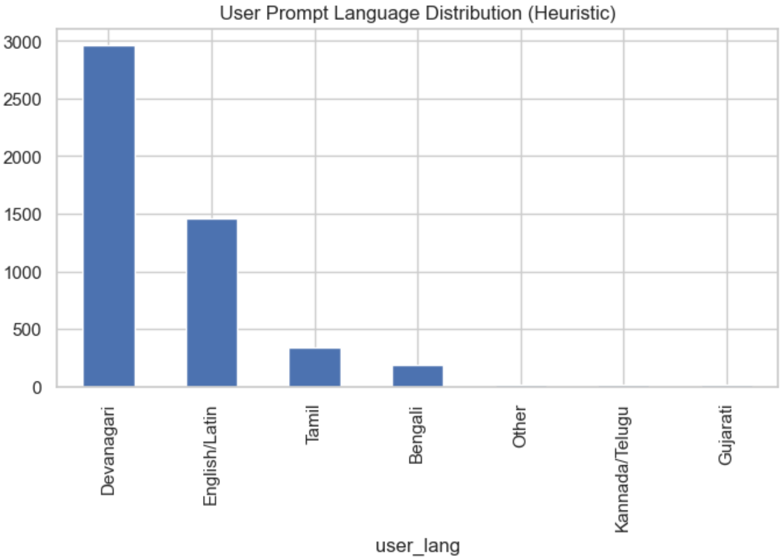
The above graphs show the top 15 distributed metric names across the test and train sets. The bar charts highlight that some metrics appear much frequently than others while few others have only a few samples. The top 15 metrics differ from the train set and test set.

5. Correlation Heatmap



The correlation heatmap shows that there are very weak correlations between prompt length, response length and score which indicates text length is not a reliable indicator in predictiong the final scores.

6. Prompt Language Distribution



Based on a simple Unicode script detection various scripts were detected in the prompts and the above chart shows the distribution. Majority of the promptys are shown to be from the Devanagari script and the next one is English text followed by Tamil and Bengali. The other category which do not match any category can be the ones from Bodo, Sindhi etc.

Thus we finally confirm that the csore distribution is extremely skewed and can assume that the test data can be bimodal.

4. Feature Preparation

To enhance coherence we merge the system, user and response embeddings into a uinified text embedding vector which enables the model to evaluate if the response aligns with the prompt and adheres to the metric meaning and metric name. This results in finally a 2304 dimensional text space per example for each of the 5000 samples.

5. Three Signal/Feature Respresentations

5.1 Feature 1 - Anomaly Detection via Isolation Forest

Isolation Forests typically helps in identifying outliers from the embedding vectors relative to the distribution of normal scores. A poor quality response might produce an embedding that deviated signifivcantly from the embedding space and can be essentially considered as an outlier. Isolation Forest works well in high dimensional spaces as it doesn't assume a uniform gaussian distribution always. The anomaly score is finally scaled between 0 and 1 which provides a normalized measure for further use.

5.2 Feature 2 - Metric - Text Compatibility:

This is the most important feature of the model. A neural network is basically used here to predict whether a metric embedding and text embedding belong together in the same embedding space or not. True metric text pairs are taken to be the true positives and for the negatives randomly shuffled text are paired up with metrics. This ensures that the classifier learns strict boundaries between the scores. Such a contrastive mismatch model works primarily because metrics often apply constraints like factuality, tone etc. Embeddings for metric and text should align when the response satisfies the metric. So when we mix and match the pairs we automatically assign a very low score to this particular data point which later helps the model to learn to predict low scores as well. The MLP model learns a non linear manifold because of this data augmentation performed.

5.3 Feature 3 – Semantic Coherence via Coherence Similarity

Cosine similarity between user and response embeddings are used which tell us how well the response addressed the user’s query. A high similarity indicates that the prompt and response are pretty similar and match the user query. A low similarity score indicates an irrelevant or off topic reply.

6. Data Augmentation

The main aim of the regressor is to differentiate accurately between strong and weak samples. The real train data alone is not enough to do this as the data is pretty skewed with very less low score samples. This introduces the need for a data augmentation technique. A synthetic mismatch augmentation is done where metric embeddings remain unchanged while the user system and response embeddings are shuffled and these incorrect pairs resemble failed outputs and the synthetic samples are uniformly assigned a core between 0 to 3.5 which represents poor performance. This teaches the model low signal patterns which predicts the LightGBM model from overpredicting and thus increases the diversity in predictions. Thus the augmented and real data together doubles training size and thus stabilizes the regressor model.

7. Computing Final Signals

After augmentation the system finally computes F1, F2, F3 for both the real and synthetic data after which Min Max scaling is applied which ensures uniform numeric ranges. This helps LightGBM in converging fatser and also prevents any kind of domination from just a single signal. Each signal captures a different dimension of the data like anomalies in the embeddings, text similarity etc. Thus this feature engineering captures all the details and nuances of the dataset and helps the model predict accurately.

8. Feature Interaction/ Feature Engineering

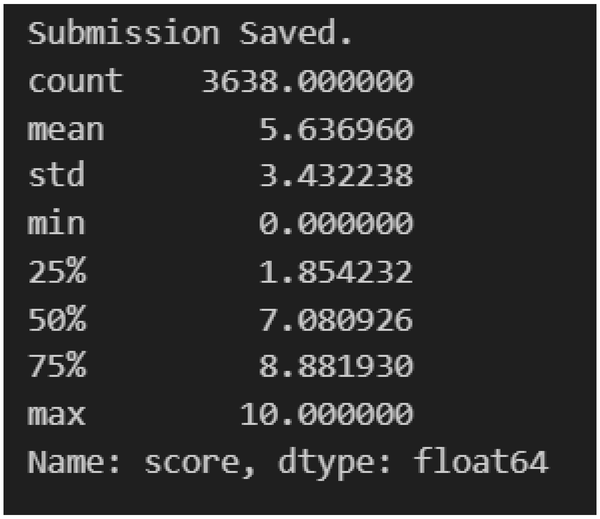
The regression features are increased using few interactyion terms like S1 x S2, S2 X S3, S1 x S3 and S1 x S2 x S3. These interaction features matter a lot as the score distributions are non linear and multimodal. The test set is mostly assumed to be bimodal after analysing through multiple submissions. These interaction terms allow LightGBM to identify different clusters presnt in the data and outlier cases where signals disagree and complex patterns between anomaly scores, text similarity etc. Thus, a structured feature space is constructed using feature engineering.

9. Final Regression with LightGBM

LightGBM is chosen for its abiluity to model non linear interactions and capture sharp decision boundaries. It is capable of handling dense tabular features and also can handle high dimensional features. The paremeters are carefully chosen after multiple iterations. Thus Finally we get a regressor which is capable of distinguishing subtle semantic behaviours.

10. Inference and Output

For each test sample we thus compute **S1, S2, S3** and generate interaction features. We finally apply LightGBM to predict the raw score and clip the score to [0,10] range. We finally export the results to a csv file with IDs. The final outputs retain a high variance of standard deviation of around 3.43 and strong cluster modeling. The decription of the final csv shows that the model is performing well as we can see the scores being distributed evenly and not just clustered toward the extreme ends.



```
Submission Saved.
count    3638.000000
mean        5.636960
std         3.432238
min         0.000000
25%         1.854232
50%         7.080926
75%         8.881930
max        10.000000
Name: score, dtype: float64
```

11. Other Approaches

Other approaches such as finetuning different LLMs were also tried out with the train data set. Specifically, the Roberta XLS model known for its multilingual capacities was tried out. This approach didn’t work out again because data augmentation techniques and fetutre engineering techniqiyes weren’t used and the Roberta model was finetuned purely on the train set which contains highly imbalanced data.

12. Conclusion

Thus this approach consists of a well balanced pipeline which integrates the embeddings into a unified model to predict final scores. It specifically constriucts a deep learning MLP based model for detection of anomaly based signals and amplifies signal information through strategic data augmentation and feature engineering techniques. It also leverages LightGBM which a a fast and easy to use cluster aware regression model. We can see this model performs best compared to many other baseline models like XG Boost etc.