# JetBrains Writing Assistance Evaluation test task report

## 1. Introduction

This report evaluates three different models—DeBERTa, XLM-RoBERTa, and GPT-4o-mini—for their ability to predict formality in sentences from a dataset sourced from Huggingface: https://huggingface.co/datasets/osyvokon/pavlick-formality-scores. The dataset contains sentences from four genres: news, blogs, email, and QA forums, annotated for formality by human raters.

## 2. Approach

The models evaluated in this report are:

- **DeBERTa**: Initially designed for ranking, I utilized DeBERTa for formality prediction.
- **XLM-RoBERTa**: A multilingual transformer model, also used for formality classification.
- **GPT-4o-mini**: A language model from OpenAI, evaluated for formality detection based on its general language understanding capabilities.

The data preparation consists of:

- Normalizing formality scores to a 0-1 scale, as they were originally in a -3 to 3 scale.
- Testing the models on a limited subset of 100 sentences, shuffled for variability.

Here, a few example sentences from the processed dataset:

| | domain | avg_score | sentence |
|---|---|---|---|
| 0 | news | 0.400000 | Tang was employed at private-equity firm Fried... |
| 1 | news | 0.666667 | San Francisco Mayor Gavin Newsom's withdrawal ... |
| 2 | answers | 0.033333 | lol nothing worrying about that. |
| 3 | news | 0.500000 | She told Price she wanted to join the Police E... |
| 4 | news | 0.800000 | The prime minister is keen to use the autumn p... |

The evaluation involves:

- Measuring performance with standard metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and $R^2$.
- Presentation of scatter plots showing the relationship between predicted formality values and true values, as well as an MAE error histogram, where the bins represent intervals of true formality values.
- Lastly, converting continuous formality values to binary (1 for formal, 0 for informal) by setting a threshold = 0.5: if formality > 0.5, then 1; else, 0 and generating a confusion matrix, as well as calculating recall, precision, and plotting the ROC curve.

## 3. Model Evaluations

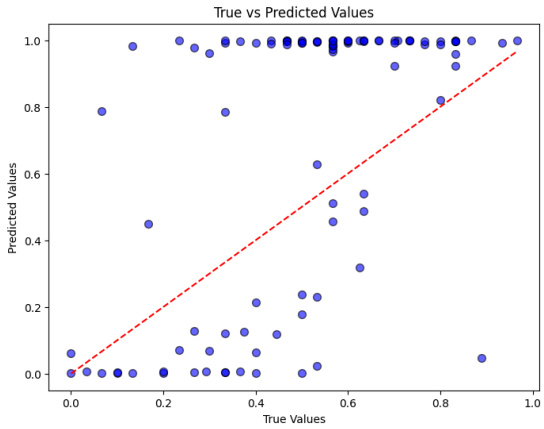
### 3.1 DeBERTa

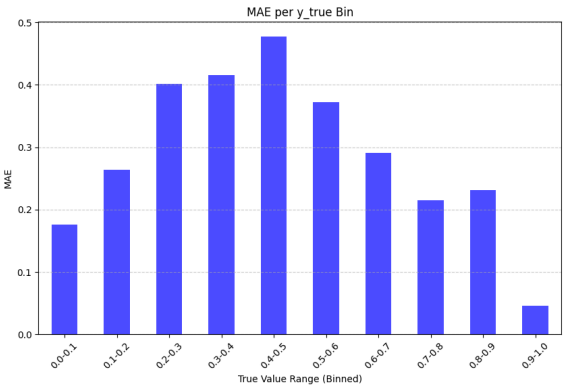
The first model evaluated is DeBERTa. While its primary use case is ranking, it was adapted for formality prediction. The predictions were compared against the true formality scores to evaluate accuracy.

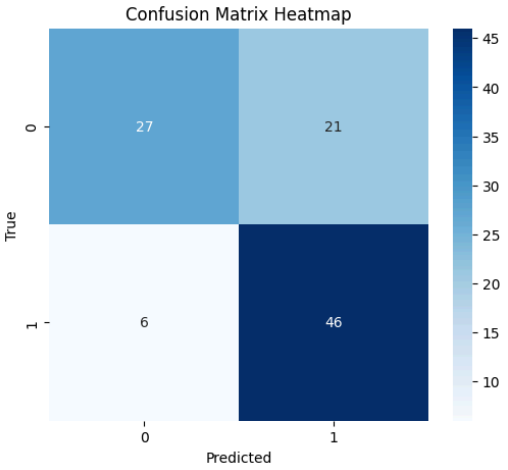
**Key Findings**:

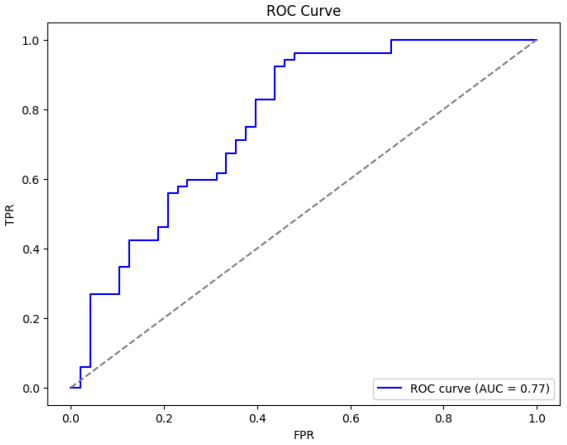
- DeBERTa performed well with very formal and very informal texts
- The model showed a tendency to overestimate formality in moderately formal sentences.
- It struggles with casual language, predicting extreme values (close to 0 or 1) for them.
- The model's predictions had significant error margins, reflected by an RMSE of 0.39 and MAE of 0.34. This led to a negative $R^2$ score (-1.98), indicating poor generalization.

**Visualizations**:




**Precision and Recall**:

- Precision: 0.69
- Recall: 0.88 The model had a good recall but could misclassify many informal sentences.

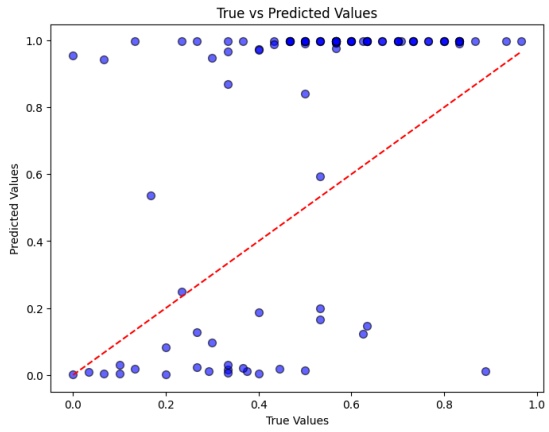
## 3.2 XLM-RoBERTa

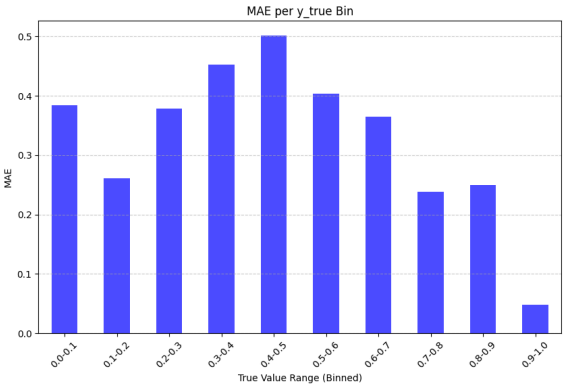
XLM-RoBERTa was the next model tested. Like DeBERTa, it showed a tendency to overestimate formality, especially for casual and informal sentences.

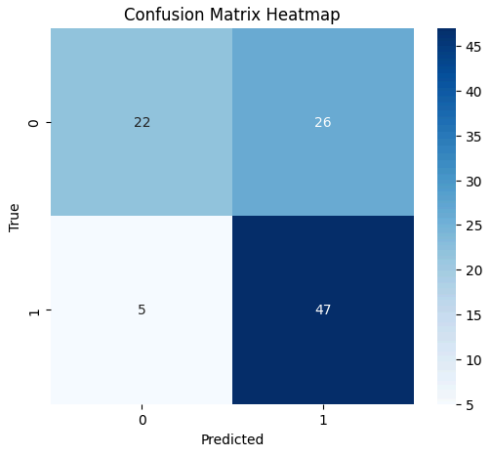
**Key Findings**:

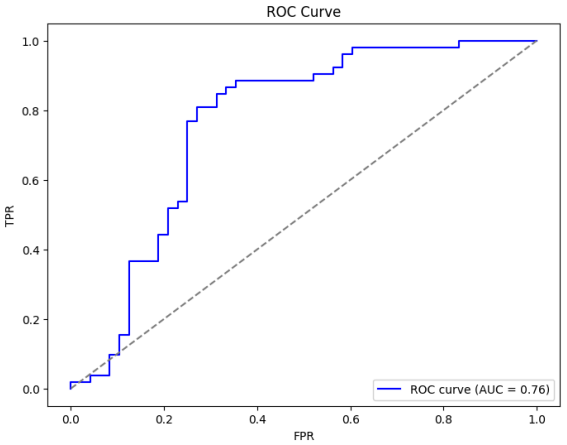
- The model struggled similarly to DeBERTa with neutral sentences, usually predicting extreme values as well.
- Results showed higher error rates than DeBERTa, with a smaller AUC (Area Under the Curve) and lower precision, but a higher recall.

**Visualizations**:




**Precision and Recall**:

- Precision: 0.64, lower than DeBERTa
- Recall: 0.90 slightly higher than DeBERTa

## 3.3 GPT-4o-mini (LLM as a judge approach)

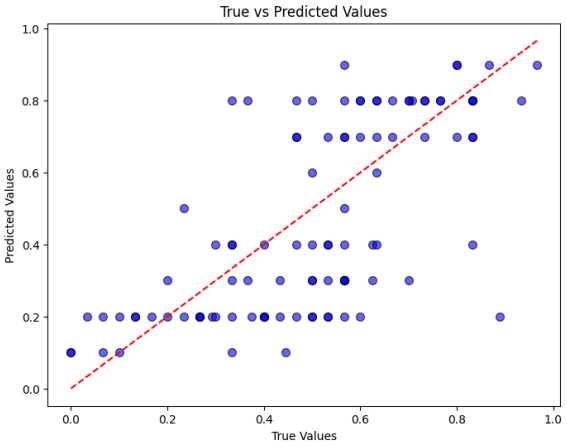
GPT-4o-mini demonstrated strong performance in detecting formality scores close to the real values, though with some limitations.

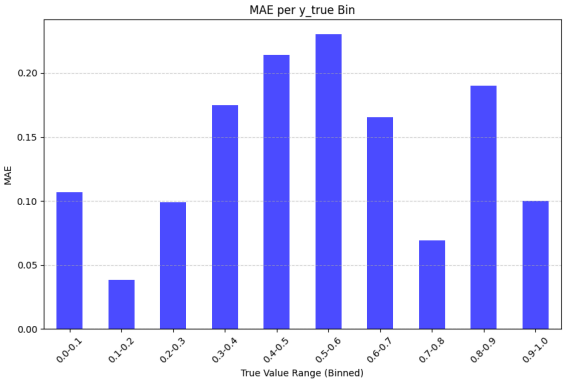
**Key Findings**:

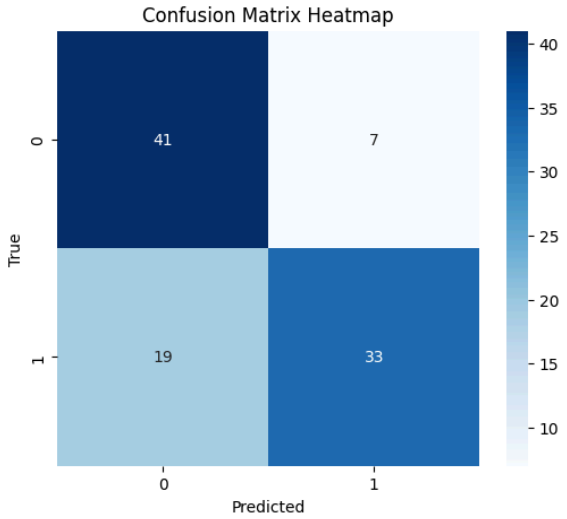
- The model showed improvement over both DeBERTa and XLM-RoBERTa, having better error rates with RMSE equal to 0.20489, MAE 0.162417 and R2 0.167764

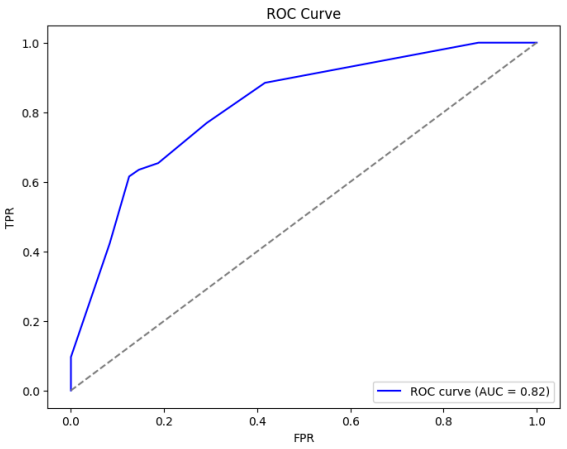
- The scatter plot below shows that its predictions align most closely with the true values.
- We can that It also struggles the most with moderately formal sentences however much less than other models
- A potential issue with GPT-4o-mini is its rounding of formality values to the nearest 0.1, which can impact precision; this could be changed by adjusting prompt.
- GPT-4o-mini has the highest AUC. However, it has the lowest recall, meaning it often fails to detect formal texts correctly.

**Visualizations**:




**Precision and Recall**:

- Precision:0.83 Higher than DeBERTa and XLM-RoBERTa.
- Recall: 0.63 Much lower than other models.

## 4. Discussion

Across the three models, a few consistent trends were observed:

- **Challenges with Neutral Texts**: All models struggled with texts that had neutral formality levels (e.g., a score close to 0.5). This was especially noticeable in the error histograms and ROC curves, where higher error rates were concentrated for these texts.
- **Overestimation of Formality**: Both DeBERTa and XLM-RoBERTa tended to overestimate formality for casual sentences, while GPT-4o-mini struggled with more informal language.
- **Model Selection**: Based on the evaluation metrics, GPT-4o-mini showed the best performance, but its sensitivity to less formal texts remains an area for improvement.

**Placeholder for suggestions on model improvements or further experiments**

## 5. Conclusion

In conclusion, while GPT-4o-mini emerged as the strongest performer, all models demonstrated specific challenges with neutral texts. Nevertheless, its predictions were still far from perfect. Further experimentation could help improve evaluation metrics. For example even though GPT had the worst Recall, adjusting the formality decision threshold could improve this. The issues with DeBERTa and XLM-RoBERTa could stem from the fact that they were trained on different datasets or designed for slightly different tasks, which is why they didn't perform as well in this context. The highest latency (2.5 minutes for 100 test sentences, compared to other models which took less than a minute) was observed with GPT, which is a downside, but this could be reduced by batching the inference.