Predicting ratings of AI generated presentations using Machine Learning

An exploration of predictive models and key factors influencing presentation quality

2-16-2025
Andreas Styliaras
2698459

1. Abstract

The developments in the field of generative AI have brought a lot of opportunities for companies, for instance to support educators and professionals in creating impactful presentation materials using interactive AI presentation tools. Sendsteps is an AI-powered platform that helps users create interactive presentations by automating content generation and enhancing audience engagement. This research investigates the predictive performance of linear and nonlinear machine learning models in classifying presentation ratings. Random Forest Classifier (RFC), Gradient Boosting Classifier (GBC), and a Feedforward Neural Network (NN) were evaluated using hyperparameter optimization techniques, including grid search for RFC and GBC, and Bayesian optimization for NN.

Extensive feature selection was performed, and synthetic oversampling (SMOTENC) was applied to address the high class imbalance. Additionally, an average image relevance score was created using OpenAI’s CLIP model to enrich the feature set.

The RFC achieved an average F1-score of 67%, and GBC yielded 66%, outperforming the NN, which obtained an accuracy of 54%. RFC and GBC demonstrated superior performance, particularly in classifying lower-rated presentations, while all models struggled with accurately predicting the highest rating class. This challenge arose because the highest-rated class contained only real samples, while synthetic data primarily supported the lower classes.

Feature importance analysis using Gini-importance revealed that the length of the initial prompt, average presentation element coordinates, and image relevance were the most influential factors in determining presentation quality. A Pearson correlation analysis further confirmed the significance of the length of the initial prompt and image relevance.

The results suggest that machine learning can effectively uncover patterns related to presentation quality; however, the subjective nature of user ratings and class imbalance remain key challenges. Future research should explore alternative techniques to handle class imbalance, as well as unbiased feature importance metrics. This study provides a foundational understanding for software platforms like Sendsteps.ai to leverage machine learning for optimizing presentation quality assessment.

2. Introduction

The rapid rise of Artificial Intelligence (AI) technologies has transformed industries worldwide, permeating fields as diverse as healthcare, finance, education, and content creation. Over the past few years, the emergence of large language models (LLMs), such as ChatGPT, has further accelerated the integration of AI into everyday applications. These powerful models have demonstrated remarkable capabilities in text generation, content summarization, and conversational AI, garnering attention from both academia and industry [1]. As a result, AI-driven content generation has emerged as a critical area of innovation, reshaping how businesses and individuals create, process, and interact with digital content.

This study represents a step toward bridging the gap between AI content generation and content evaluation. By exploring the relationship between presentation features and their perceived quality, the goal is to enhance the effectiveness of AI-driven platforms like Sendsteps.ai while contributing to the broader understanding of feature importance in machine learning.

2.1 Sendsteps.ai

Sendsteps.ai [2] exemplifies this evolution, serving as an AI-driven platform specifically designed to support educators and professionals in creating impactful presentation materials. The platform empowers users (e.g. university/primary professors, business officials, students, etc.) to transform and generating content into presentations and quizzes within minutes.

When creating a presentation with Sendsteps.ai, the user is presented a questionnaire that needs to be filled out. This questionnaire will then be processed and sent to the LLM as a structured prompt. In the questionnaire, the user can enter some keywords (maximum 3000 characters) or upload a document (PDF, PPTX, DOCX, TXT files). Additionally, you can choose the preferred language, the number of slides, the amount of text on the slides, generation of speaker notes, generation of interactive questions, who the presentation is aimed at (e.g. audience) and the tone of voice of the generated text. Finally, you can choose a preferred title for the presentation (based on the provided keywords) and select if you want to generate an outline first, allowing you to review and adjust the content, or immediately generate the presentation.

User data is collected through *Amazon Web Services* using *HeidiSQL*, a free Open-Source software that allows the modification of collected data and structures. One instance of user data is *presentation rating*: Sendsteps.ai are effectively collecting user feedback by actively giving users and clients the option to assign a grade to the presentation that has been generated for them specifically. Each presentation is given two distinct quality ratings, one for the presentation content and one for the presentation images. This grade ranges from **one to five**, one being the worst score and five the best score.

Alongside the rating assigned by users, specific characteristics of presentations are stored, offering a rich dataset that includes numerical features, binary indicators as well as prompt metadata, creating an extensive collection of presentation datapoints.

2.2 The problem

One of the main challenges that Sendsteps.ai faces is the difficulty in fully understanding the reasoning behind the ratings provided by users. While users are encouraged to provide feedback, the nature of this feedback can often be vague, subjective, or insufficiently detailed. In most cases, users simply assign a rating, with limited explanation regarding the reasons behind their choices. Occasionally, users may (by choice) leave a short description explaining their rating, but these comments often lack the depth and constructiveness needed to draw clear conclusions about the specific aspects of the presentation that influenced their judgment. For instance, when users assign a rating of "1" (the lowest rating), the reasons behind such a low score are not always clear—sometimes the comments are brief or do not focus on actionable areas for improvement. Similarly, high ratings (e.g., a "5") are also given with minimal clarification, leaving significant gaps in understanding why the presentation was perceived as excellent.

This lack of clear feedback makes it difficult for Sendsteps.ai to pinpoint the exact characteristics or content aspects that lead to positive or negative user experiences. While Sendsteps.ai collects data on key presentation characteristics, this alone does not provide a sufficient basis for understanding the complex relationships between presentation elements and user satisfaction. Without deeper insights into how specific aspect of the presentation content influence ratings, the platform struggles to improve its presentation generation process and optimize content directly based on user feedback.

2.3 Research goal

To address this gap, this research aims to leverage machine learning to bridge the divide between raw presentation features and user-assigned ratings, potentially providing Sendsteps.ai with valuable insights into what contributes to the perceived quality of a presentation.

The research goal is twofold:

> **Leverage machine learning to predict presentation ratings**: By predicting presentation ratings based on features and uncovering patterns of influence, the goal is to develop a model that can more accurately forecast how users will rate a presentation.
>
> **Identify which features are most strongly correlated:** Through explicit ranking of feature importance, this research will offer insights into which specific presentation features—whether related to content, images, or other aspects—affect ratings the most, both linearly and non-linearly.

This model can potentially be integrated into the presentation creation process. For any newly generated presentation, it will be passed through the predictive model, which can provide feedback before the presentation is shown to the user. This approach can potentially help prevent negative ratings by flagging presentations that are likely to receive a rating below a certain threshold (e.g. <3), prompting the system to generate a new version of the presentation.

Research question:

> *How do linear and non-linear predictive models perform in predicting the quality ratings of AI-generated presentations, and which features contribute most significantly to these predictions?*

3. Literature study

3.1 The use of Generative AI in education

Nowadays, the use of Artificial Intelligence (AI) in higher education has increased dramatically over the last five years [3]. A study introduces the GAIDE framework, which leverages Generative AI to enhance curriculum design, highlighting the potential of Generative AI to significantly reduce the workload of instructional staff among many other benefits in educational settings [4]. However, the study also highlights challenges, including a lack of genuine skill development, diminished problem-solving capabilities, and an over-reliance on the tool of non-experts (such as students), which can hinder self-sufficiency.

The rise of AI in higher education and other fields has introduced AI-powered presentation tools that enhance productivity, creativity, and customization through automation of tasks like design layout and data visualization or presentation generation altogether. Marketing outlets promise that "AI interactive presentation tools are transforming the global education landscape by changing traditional teaching methods and making learning more engaging and interactive for students" [5].

Among the biggest differentiations of generative AI applications are those that produce full presentations using prompts (Sendsteps.ai) versus those in which generative AI is integrated as a means of enhancing or modifying content (e.g., Google Slide plug-ins like MagicSlides, PowerPoint with Copilot integration) [6].

3.2 Predicting presentation quality

Numerous studies have been conducted on assessing the quality of presentations. A particular study uses multimodal sensing and machine learning techniques to evaluate and potentially help to improve the quality of the content and delivery of public presentations. The experiments suggest that multimodal cues can predict human scores on presentation tasks, and a scoring model comprising both verbal and visual features can outperform that using just a single modality [7]. Another study focuses on assessing the quality of presentations based on acoustic information and hand movements [8].

A more relevant study developed a comprehensive framework for information quality (IQ) for presentations and exploring the possibility of automatically detecting the IQ of slides. This study focuses on four main categorical taxonomies to evaluate the quality of presentation slides across three datasets: *Intrinsic* (e.g. accuracy, cohesiveness)*, Representational* (e.g. clarity, visual attraction)*, Contextual* (e.g. completeness, informativeness) and *Reputational* (e.g. author/institutional reputation). The results suggest that "better-quality slides have a tendency to contain more slides (pages), images, font colours, font names and highlights, contributing to better representational clarity and informativeness". Across the datasets, it was found that representational clarity, informativeness and visual attraction were the most effective features for ranking the quality of presentation slides, whereas completeness, ease of navigation and accuracy were relatively unhelpful [9].

Investigating the quality prediction of AI generated presentation is in some ways unprecedented, as virtually no similar research has been conducted. Predicting human assigned ratings of AI generated presentations in the specific use case of Sendsteps.ai is a niche subject. Nevertheless, it is of great interest to software companies offering similar AI presentation tools.

4. The solution

4.1 Data discovery

For this research, after consulting which information could be useful for my investigation with the Sendsteps development team, limited access to certain tables on the live database server was granted. These tables contain data of presentation characteristics directly linked to the generated presentations that users assigned feedback to. This access allowed the compilation of an informative dataset that connects multiple specific presentation attributes across a multitude of tables on the database server, which could play an important role in predicting quality ratings. The specific SQL query can be found in the appendix (*Figure 15*).

For each presentation to have a single rating score, the average of both the score of images and score of content is calculated, and thereupon rounded to the nearest integer. It is essential to round up, so the target variable '**presentation rating'** does not deviate from the initial five classes/grade ranges (1 to 5).

After extensive exploration of the database and experimentation with multiple potential datasets, the decision was made to stick to specific features that provide a good sample size of each class.

For further clarification, within the SQL query some constraints are established to collect uniform data:

- Presentations with no rating (user feedback) <u>will not be considered</u>
- Any presentation (even with a rating) that has been deleted <u>will not be considered</u>
- Any presentation with no images <u>will not be considered</u>
- Any slides within presentations that have been deleted <u>will not be considered</u>
- Any shapes on slides that have been deleted <u>will not be considered</u>

Before processing the data, the dataset consists of 3099 presentations with 20 features:

<u>rating</u>, which represents the target variable **[1 to 5];**

<u>numSlides</u>, number of presentation slides **[Integer]**;

<u>numShapes</u>, number of presentation elements **[Integer]**;

<u>isEdited</u>, if *generated* presentation was edited or not **[Binary]**;

<u>hasVideo</u>, if presentation has a video or not **[Binary]**;

<u>hasWordcloud</u>, if presentation has a word-cloud **[Binary]**;

<u>hasMPC</u>, if the presentation has a quiz **[Binary]**;

<u>avgX</u>, average X coordinate of all shapes in presentation **[Continuous]**;

<u>avgY</u>, average Y coordinate of all shapes in presentation **[Continuous]**;

<u>avgWidth</u>, average width value of all shapes in presentation **[Continuous]**;

<u>avgHeight</u>, average height value of all shapes in presentation **[Continuous]**;

<u>presentationStyleId</u>, presentation template **[Integer]**;

<u>initialPromptTokens</u>, character length of initial prompt sent to LLM **[Integer]**;

<u>responseTokens</u>, character length of LLM response **[Integer]**;

length, structure of presentation content **[Text]**;

toneOfVoice, tone of presentation content **[Text]**;

backgroundOpacity, opacity of presentation background **[Continuous]**;

imgURLs, hyperlinks of images within the presentation **[Text]**;

subject, presentation subject **[Text]**;

language, presentation language **[Text]**

Evidently, the dataset features are represented by different datatypes: textual data, discrete data and continuous data.

In *Figure 1*, the overall target variable distribution can be observed.

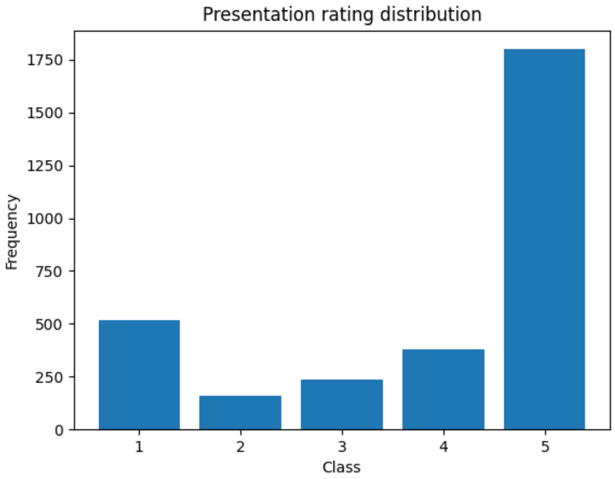


Figure 1

The class distribution of the target variable is highly imbalanced, with ratings 5 being dominating across presentations, comprising 58 % of the dataset. In contrast, the remaining classes are significantly underrepresented, particularly rating 2, which has the lowest proportion at merely 5%. Rating 1 is the second highest rating present in the dataset and accounts for 16 %. Rating 4 contributes 12 % and rating 3 marks 7 %.

The feature distributions before processing the data is found in *Figure 17* (appendix).

4.2 Data preparation

As the dataset includes some features with textual data (language, toneOfVoice, length), these need to be transformed into numerical data so the models can interpret them. The feature *toneOfVoice* can have four distinguished values (*Neutral, Casual, Intellectual* and *Persuasive)* and the feature *length* has three distinct categories (*Summarized, Informative, Detailed*).

Although there exist many encoding techniques, and it can be argued that *toneOfVoice* and *length* can be viewed as ordinal (e.g. there is a specific order to them), both features have been one-hot encoded. Even though this increases the dimensionality within the dataset, one-hot encoding can lead to better performance for linear and non-linear models as opposed to binary encoding or feature hashing [10].

At first glance, language seems to be a good addition to the feature set of this research. But upon further inspection, there is a high imbalance in language frequency, which could lead to improper interpretation of the model on the importance of language for the presentation rating. The data encompasses presentations across 42 languages, predominantly *English*, *Spanish*, and *Indonesian*. The remaining languages are represented by fewer than forty presentations, with some languages only being present in a single presentation, which may not provide sufficient data to draw reliable conclusions or enable the model to effectively capture language-related patterns. Eliminating outlying languages is an option, but it can be argued the specific language of the presentation does not significantly influence the presentation ratings. Not to mention it would increase the dimensions three-fold (after applying one-hot encoding). For these reasons, *language* was eliminated from the feature set.

Figure 16 (appendix) represents an initial correlation matrix using *Pearson's correlation coefficient* [11], which helps identifying which features have the strongest linear relationship with the target variable, as well as uncovering any potential redundancies or multicollinearity within the feature set.

As can be foremost observed, several features have a very high correlation between them. To remove redundancy, *numImgs* is eliminated. Furthermore, as *numShapes* is also very highly correlated to the number of slides per presentation, a new feature is created that represents the average shapes per slide (dividing the number of shapes by the number of slides) called *shapesPerSlide*. Moreover, *avgX* and *avgY*, denoting the averages X and Y coordinates of presentation elements, have a strong linear correlation (0.78). Both will be combined into a single feature *avgShapeCoordinates*, by calculating the mean of both values. *Length* has a positive correlation to *numSlides*, but both features will stay untouched, as although both have a strong linear association, they interact differently with other features. Not to mention *length* will undergo one-hot encoding, which should break up the high correlations. Lastly *avgWidth* has a high negative correlation with *avgX,* which will persist after combining both coordinates. Nevertheless, *avgWidth* will be included in the feature set.

There are other mild to medium correlations present, but no further modifications have been done to the features.

4.2.1 CLIP-score

Since images play a critical role in presentation quality, it was necessary to develop a method to assess their relevance to the presentation subject. CLIP [12] was selected for this task, as it enables the calculation of similarity scores between images and their respective presentation content. In Sendsteps.ai, images are sourced using the Serper API [13], which fetches Google search results based on the title. Each query retrieves two sets of approximately ten images, which users can select from for their presentation slides. All the images of a given presentation are stored in the feature *imgURLs*.

Originally, image relevance was intended to be evaluated based on slide content, assuming that textual context would provide a more meaningful basis for the relevance score. However, this proved unreliable, as many image-containing slides lacked informative text or there was no text altogether. Relevance will be computed based on the presentation title (*subject* feature) instead. CLIP assigns a similarity score to every image in each presentation**.** The average score across all images serves as the final relevance score. Since CLIP was primarily trained on English image-text pairs**,** all non-English presentation titles were translated into English. Additionally, to standardize input dimensions, all images were resized to 224×224 pixels, which led to slightly higher variance and standard deviation compared to using the original image sizes.

The average similarity scores across all presentations range from 12 to 38 (*Figure 2*), with some individual images exceeding 40 or scoring below 12. To test whether CLIP's scores held real statistical significance, a random (float) value between 12 and 38 was assigned to each presentation (*Figure 3*), serving as a baseline comparison to validate the relevance assessment methodology. Both features (*CLIP image relevance score* and *random control value*) are then compared using Pearson's correlation coefficient.

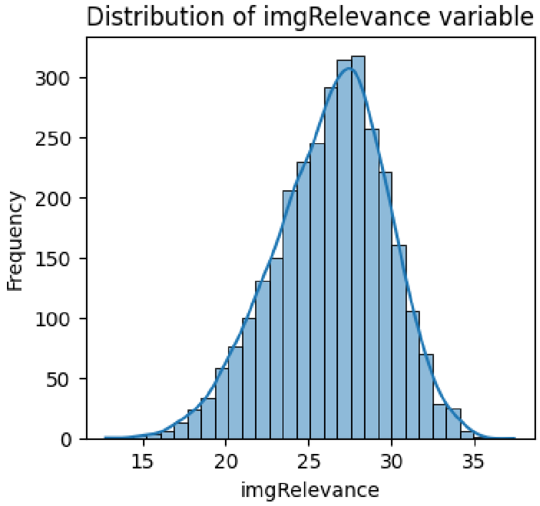


Figure 3

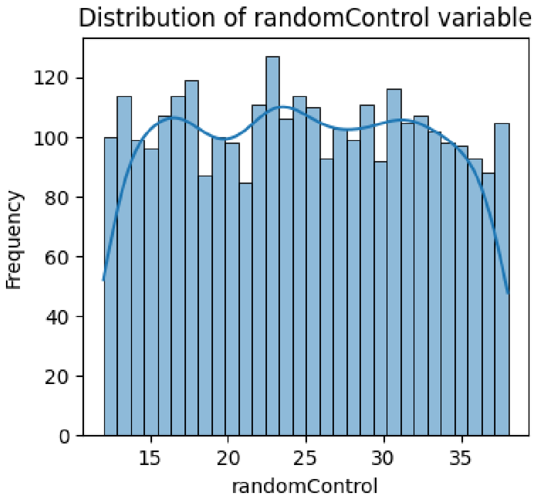
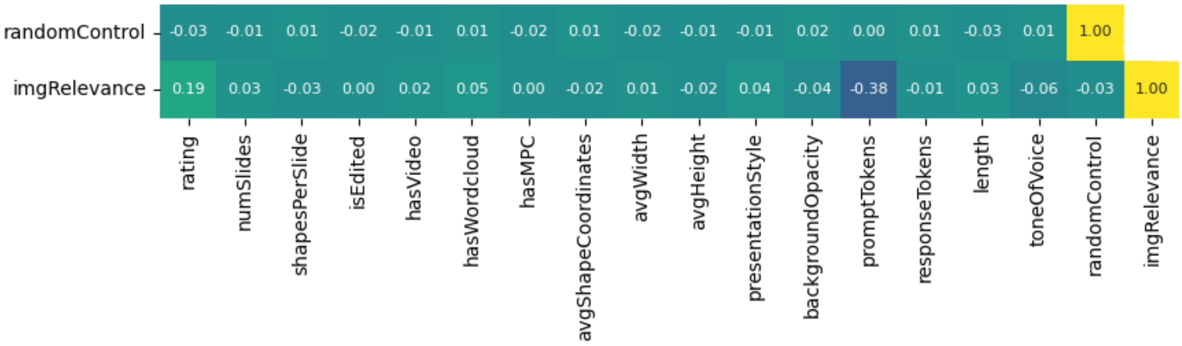


Figure 2

As can be seen in *Figure 4*, the relevance score has some linear correlation (+0,19) with the target variable (*rating*). This indicates that if the overall image relevance for a presentation increases, so does the presentation rating. It has an even greater negative linear relationship with *promptTokens*. Based on this, this newly created featured using OpenAIs CLIP model is a considerable valuable addition to the feature set, compared to *randomControl* feature, which doesn't exceed a linear correlation stronger than 3%.



	rating	numSlides	shapesPerSlide	isEdited	hasVideo	hasWordcloud	hasMPC	avgShapeCoordinates	avgWidth	avgHeight	presentationStyle	backgroundOpacity	promptTokens	responseTokens	length	toneOfVoice	randomControl	imgRelevance
randomControl	-0.03	-0.01	0.01	-0.02	-0.01	0.01	-0.02	0.01	-0.02	-0.01	-0.01	0.02	0.00	0.01	-0.03	0.01	1.00	
imgRelevance	0.19	0.03	-0.03	0.00	0.02	0.05	0.00	-0.02	0.01	-0.02	0.04	-0.04	-0.38	-0.01	0.03	-0.06	-0.03	1.00

Figure 4

Handling imbalanced data

The distribution of the target variable '**presentation rating**' is highly imbalanced, with most presentations receiving the highest rating, while the other ratings are significantly underrepresented. Studies have shown that imbalanced data may lead to challenges in model performance (for example Convolutional Neural Networks [14]), as classifiers tend to be biased toward the majority class, potentially resulting in poor predictive accuracy for the minority classes. To mitigate this, Synthetic Minority Over-sampling Technique Nominal Continuous (SMOTENC) [15] is employed instead of traditional SMOTE, as the dataset comprises both, categorical and continuous features. Unlike standard SMOTE, which only works with continuous data, SMOTENC generates synthetic instances while preserving the relationships between categorical variables, ensuring a more appropriate augmentation of the minority classes. Instead of oversampling all minority classes to match the count of the majority class, a proportional discrepancy is maintained. This means that while minority classes are oversampled to improve representation, they are not all artificially increased to the same level as the majority class, preventing an unrealistic class balance. The new presentation rating distribution can be seen in *Figure 5*.

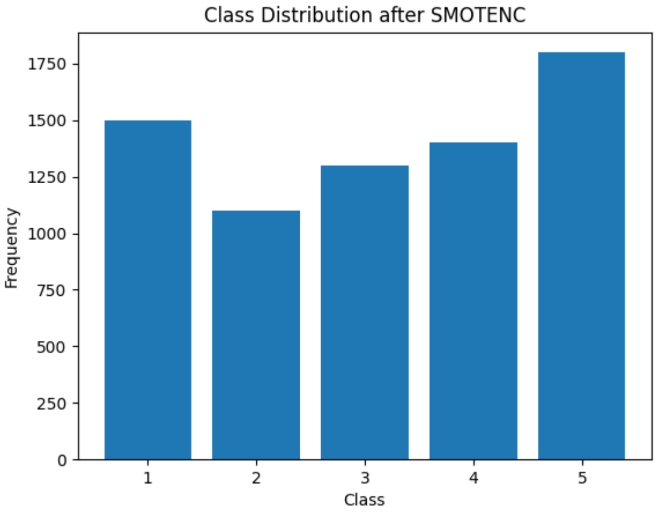
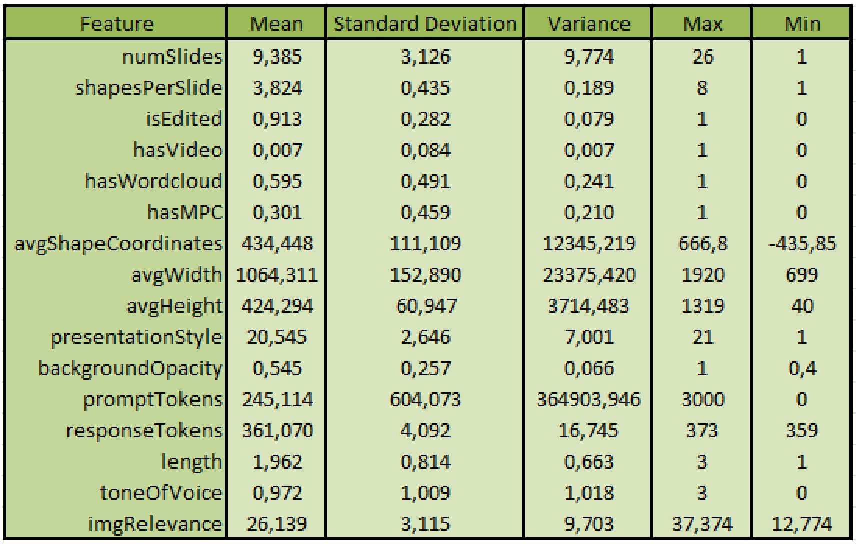


Figure 5

After applying the SMOTENC oversampling technique, presentation rating 1 increased to 1,500 samples, representing 21 % of the data, rating 2 was oversampled to 1,100 samples, accounting for 15% of the data, rating 3 was oversampled to 1,300 samples (18%) and lastly rating 4 increased to 1,400 samples (19%). Presentation rating 5 remained the majority class at 1,799 samples, making up 25 %.

The data now contains 7099 instances. This adjustment helps address the class imbalance present in the original dataset, improving the representation of minority classes and supporting more effective model training.

4.2.2 Feature scaling



Feature	Mean	Standard Deviation	Variance	Max	Min
numSlides	9,385	3,126	9,774	26	1
shapesPerSlide	3,824	0,435	0,189	8	1
isEdited	0,913	0,282	0,079	1	0
hasVideo	0,007	0,084	0,007	1	0
hasWordcloud	0,595	0,491	0,241	1	0
hasMPC	0,301	0,459	0,210	1	0
avgShapeCoordinates	434,448	111,109	12345,219	666,8	-435,85
avgWidth	1064,311	152,890	23375,420	1920	699
avgHeight	424,294	60,947	3714,483	1319	40
presentationStyle	20,545	2,646	7,001	21	1
backgroundOpacity	0,545	0,257	0,066	1	0,4
promptTokens	245,114	604,073	364903,946	3000	0
responseTokens	361,070	4,092	16,745	373	359
length	1,962	0,814	0,663	3	1
toneOfVoice	0,972	1,009	1,018	3	0
imgRelevance	26,139	3,115	9,703	37,374	12,774

Figure 6

As can be seen in *Figure 6*, there is a wide range of values across features.

The most common technique in feature scaling is standardization. Standardization changes the distribution of each feature such, that the mean of all features is zero. It makes all features play role on classification so that no feature impacts the model just because of their large magnitude [16]. Algorithms, such as neural networks, obtain better convergence with feature scaling than without it [17]. “All the tree-based algorithms however not require scaling. Performing feature scaling in these algorithms may not have significant effect” [16]. Either way, besides binary features and some features that are transformed using one-hot encoding (as described in section 3.2), standardization was applied to all features resulting in them having a mean of zero. The exception was *imgRelevance* which was scaled linearly (*normalization*), as the distribution of this feature resembled a normal distribution.

The feature distributions after one-hot encoding and feature scaling can be viewed in *Figure 19* (appendix).

4.3 Experimental Setup

4.3.1 Predictive models

In this study, two baseline linear models, Random Forest [18] and Gradient Boosting [19] have been employed to establish a foundational performance benchmark. These models serve as the baseline for comparison with a more sophisticated nonlinear approach, specifically a Neural Network [20]. This comparative analysis provides insight into the relative performance improvements that a more complex model can offer in this task of presentation quality prediction.

The data utilized throughout these experiments undergoes a systematic splitting process into three distinct sets: **training** (60%), **validation** (20%)**,** and **test** (20%). This is done to reduce the likelihood of overfitting and ensures a more reliable estimate of real-world model performance. Before splitting the dataset, shuffling is performed to make sure that any inherent order within the data does not influence the performance evaluation. Moreover, emphasis is put on maintaining the distribution of the target variable across the training, validation, and test sets, resulting in similar distributions of **'presentation ratings'** across splits.

The training set is used for model fitting, while the validation set is employed to fine-tune the model's hyperparameters. The final evaluation is conducted on the test set, which is never exposed to the model during the training or hyperparameter tuning phases, providing an unbiased measure of the model's true predictive power.

4.3.1.1 Baseline models

For the linear models, Random Forest Classifier (RFC) [21] and Gradient Boosting Classifier (GBC) [22], a brief grid search (27 combinations for RFC and 48 combinations for GBC) is initially performed to identify a set of optimal hyperparameters based on the validation accuracy. This done using the scikit-learn library [23].

The RFC hyperparameters include:

- **n_estimators:** The number of trees in the forest (100, 300, 500)
- **max_depth:** The maximum depth of the tree (*None*, 10, 20)

The GBC hyperparameters include:

- **n_estimators:** The number of boosting stages to perform (100, 300, 500, 700)
- **learning_rate:** Learning rate shrinks the contribution of each tree (0.01, 0.05, 0.1, 0.2)

There is a trade-off between *learning_rate* and *n_estimators* [22].

The chosen hyperparameters are those that yield the highest performance on the validation set. Once the hyperparameters are determined, a final evaluation is carried out on the test set, establishing a baseline for linear model performance. The GBC is evaluated on the test set exactly once. As the RFC is inherently stochastic, this classifier will be evaluated on the *test data* for 30 iterations

4.3.1.2 Neural Network

The neural network model (NN) follows a similar structure in terms of data splitting. However, a greater emphasis is placed on the optimization of hyperparameters. Once the data is split, the next step is to determine the optimal hyperparameters for the neural network. Initially, an extensive grid search was conducted to explore a wide range of potential configurations. However, this method quickly presented challenges. Given the vast number of hyperparameters to explore (over 4000 combinations), the grid search required an immense amount of time and computational effort. Therefore, it was decided to adopt a more efficient approach for hyperparameter tuning.

Bayesian optimization [24], implemented through the Optuna framework [25], aims to systematically search for the best hyperparameters by learning from past evaluations. The scope of the hyperparameters provided includes:

- **Batch size**: The number of samples processed before the model is updated, ranging from 32 to 512.
- **Hidden layer size**: The number of neurons in each hidden layer, ranging from 32 to 512.
- **Number of hidden layers**: The number of hidden layers in the network, varying from 1 to 7.
- **Dropout rate**: The fraction of units dropped during training to prevent overfitting, ranging from 0.1 to 0.5.
- **Learning rate**: The rate at which the optimizer updates the model, between 0.00001 and 0.01, using a logarithmic scale.
- **Activation function**: The activation function for each layer, with choices including ReLU, LeakyReLU, and Tanh.

This optimization process involves running 500 trials, where each trial corresponds to a unique combination of hyperparameters. The optimizer evaluates each combination and adjusts the search accordingly, in pursuit of the most effective configuration for the given dataset.

It is important to note that the learning rate hyperparameter, specifically refers to the Adam (Adaptive Moment estimation) optimizer, which is straightforward to implement, is computationally efficient and has little memory requirements [26].

The choice of activation functions [27] (ReLU, LeakyReLU, and Tanh) introduces nonlinearities into the model, which is essential for capturing complex patterns in the data. Additionally, the model architecture incorporates bias for each layer, as well as dropout for regularization.

Given the complexity of training a neural network, the number of epochs required for effective training is not fixed and can vary depending on the selected hyperparameters. To address this, the training procedure is capped at 1000 epochs, but early stopping is employed to prevent unnecessary computation. Early stopping monitors the loss on the *validation data*, and if the validation loss does not improve within a specified interval (20 epochs of patience), the training is stopped. This ensures that the model is not overfitted to the training data and saves computational resources by avoiding unnecessary epochs once the model has converged.

To assess the statistical reliability of the results, the number of test runs required is calculated based on the confidence interval of 95% and a margin of error of 5%. Each run, the *final neural* network with the best hyperparameters is retrained, with early stopping being employed based on validation loss, on the same training data and evaluated on the identical test set. To introduce some randomness, each run the neural network is initialized with different weights using a standard distribution with a standard deviation of 0,2. This ensures that the results are statistically significant and provides a robust estimate of the model's performance.

The results are presented in the following forms:

Confusion Matrix: A heatmap of the confusion matrix, averaged over all *test data*set runs, showing how often each class is predicted compared to its actual distribution.

Classification Report: The precision, recall, and F1-score for each class, averaged across all *test dataset* runs, offering a comprehensive view of model performance for each specific class.

Test Accuracy: The average accuracy (combined F1-scores) across all runs using the *test dataset*, serving as a measure of how well the model generalizes to new, unseen data.

This experimental setup is designed to rigorously evaluate the performance of both linear and nonlinear models, ensuring that the conclusions drawn from the study are both reliable and representative of the model's true capabilities.

4.3.2 Feature importance

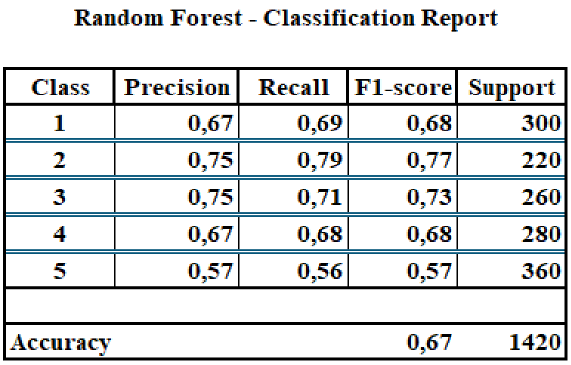
In the case of the linear models, Random Forest and Gradient Boosting, both offer built-in methods for ranking the importance of features. These rankings reflect the contribution of each feature in making accurate predictions, which can assist in identifying the most influential variables for predicting presentation ratings. These importance scores are calculated during model evaluation on the *test dataset* and are then visualized through feature importance plots.

Furthermore, to investigate potential linear relationships between features and the target variable, the Pearson correlation matrix is computed. This will measure the linear relationship between two continuous variables, providing a straightforward understanding of how features and the target variable are related.

4.4 Results

4.4.1 Baseline models

After the grid search (explained in section 4.3.1.1), a **max_depth** value of *None* and **n_estimators** value of *500* turned out to be the hyperparameters with the best accuracy on the *validation data*. Below (*Figure 7* and *Figure 8*) you can find the confusion matrix and classification report for the Random Forest Classifier, after 30 iterative evaluations on the *test data.*



Random Forest - Classification Report

Class	Precision	Recall	F1-score	Support
1	0,67	0,69	0,68	300
2	0,75	0,79	0,77	220
3	0,75	0,71	0,73	260
4	0,67	0,68	0,68	280
5	0,57	0,56	0,57	360
Accuracy			0,67	1420

Figure 7

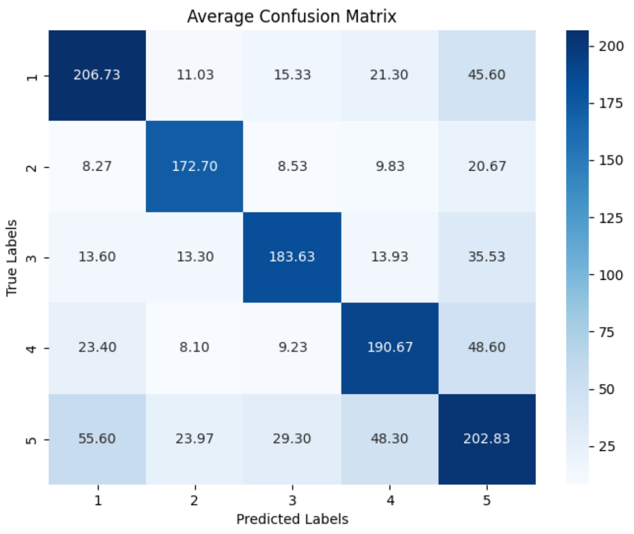
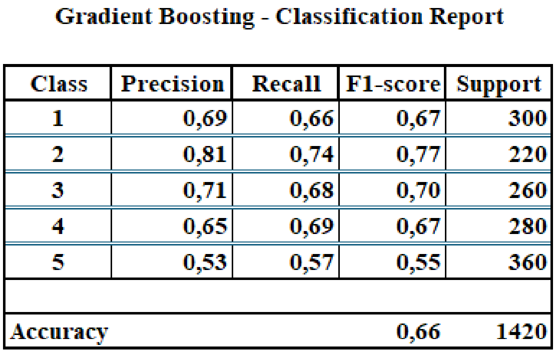


Figure 8 (RFC Confusion Matrix)

Looking at the classification report, the RFC correctly predicted the '**presentation rating**' 67% of the time. While this is a reasonable accuracy for multi-class classification, it indicates that the model is still making a fair number of misclassifications.
For the first class ('**presentation rating**' = 1), the model achieves a precision of 67% and a recall of 69%.
The second class ('**presentation rating**' = 2) exhibits the best performance among all classes, with a precision of 75% and a recall of 79%.
With a precision of 75% the model also performs well on the third class ('**presentation rating**' = 3). For class four ('**presentation rating**' = 4), the performance closely resembles that of class 1, with both precision and recall around 67% and 68%, respectively.
Class 5 ('**presentation rating**' = 5), however, has the lowest performance metrics across all classes. It achieves a precision of 57% and a recall of 56%.

The confusion matrix (*Figure 8*) reveals that actual presentations with a rating of 5 are often mistaken for other classes. Out of the actual 360 samples belonging to a rating of 5, on average, 55.6 samples were misclassified as having a rating of 1 and 48.3 were predicted to have a rating of 4. Confusing presentations with an actual rating of 5 to have a rating of 4 is arguably a less severe than misclassifying both ends of presentation ratings. Moreover, presentations are frequently misclassified as having a '**presentation rating**' of 5. Out of 300 samples, on average 45.6 presentations with an actual rating of 1 were misclassified as having a rating of 5.

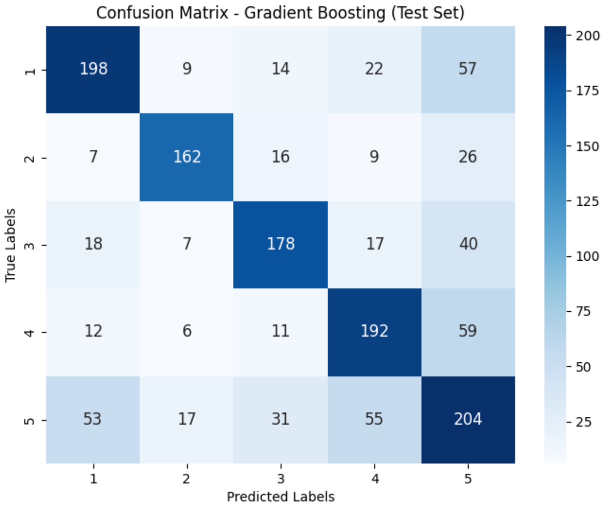
Furthermore, for the second decision-tree model (GBC), after the grid search (explained in section 4.3.1.1), a **n_estimators** value of *700* and **learning_rate** of 0.2 turned out to be the hyperparameters with the best accuracy on the *validation data*. *Figure 9* and *Figure 10* represent the confusion matrix and classification report for the Gradient Boosting Classifier, after evaluating it on the *test data*.



Gradient Boosting - Classification Report

Class	Precision	Recall	F1-score	Support
1	0,69	0,66	0,67	300
2	0,81	0,74	0,77	220
3	0,71	0,68	0,70	260
4	0,65	0,69	0,67	280
5	0,53	0,57	0,55	360
Accuracy			0,66	1420

Figure 9



Confusion Matrix - Gradient Boosting (Test Set)

True Labels \ Predicted Labels	1	2	3	4	5
1	198	9	14	22	57
2	7	162	16	9	26
3	18	7	178	17	40
4	12	6	11	192	59
5	53	17	31	55	204

Figure 10

Similar to Random Forest, the classification report indicates that the GBC correctly predicted the '**presentation rating**' 66% of the time.

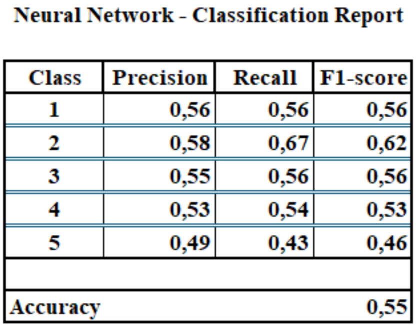
For the first class (‘**presentation rating**’ = 1), the model achieves a precision of 69% and a recall of 66%.
Again, the second class (‘**presentation rating**’ = 2) exhibits the best performance among all classes, with a precision of 81% and a recall of 74%.
With a precision of 71% and a recall of 68%, Gradient Boosting performs slightly worse on the third class (‘**presentation rating**’ = 3) than RFC.
For class four (‘**presentation rating**’ = 4), has a precision of 65% and a recall of 69%.
Also with the GBC, class 5 (‘**presentation rating**’ = 5) scores the lowest across all classes. It achieves a precision of 53% and a recall of 57%.

The confusion matrix (*Figure 10*) is showing similar results to Random Forest (*Figure 8*). Gradient Boosting shows slightly more misclassification across all classes compared to Random Forest, with marginally fewer correct predictions for classes 1, 2, and 3. Yet again, 156 presentations that have a rating of 5 are misclassified (highest being rating 4 with 55 instances and rating 1 with 53 instances). 182 presentations are incorrectly assigned a rating of 5.

4.4.2 Neural Network

After the extensive hyperparameter optimization search (using the Bayesian method), the following resulted in the highest accuracy on the *validation data*: **batch size:** *391*, **hidden layer size:** *249*, **number of hidden layers:** *2*, **dropout rate:** *0.162*, **learning rate:** *0.0036*, **activation function:** *Tanh*

Below (*Figure 11* and *Figure 12*) you can find the confusion matrix and classification report for the Feedforward Neural Network. Additionally, *Figure 20* (appendix) depicts the loss and accuracy across all runs on the test data.



Neural Network - Classification Report

Class	Precision	Recall	F1-score
1	0,56	0,56	0,56
2	0,58	0,67	0,62
3	0,55	0,56	0,56
4	0,53	0,54	0,53
5	0,49	0,43	0,46
Accuracy			0,55

Figure 11

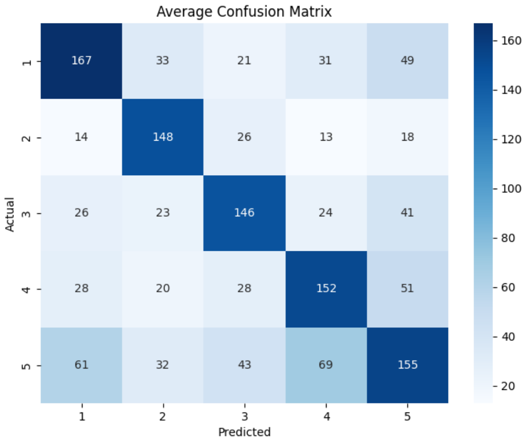


Figure 12

The classification report for the Neural Network model reveals an overall weaker performance compared to the previously evaluated linear models. The Neural Network achieves an accuracy of 54% with a standard deviation of 1%, and a loss (on the test data) of approximately 1,22 with a standard deviation of 1%.

Class 2 (‘**presentation rating**’ = 2) still emerges as the best-performing category, with an F1-score of 62%, though it is a clear decline from the 0.77 observed in both Random Forest and Gradient Boosting. The performance for class 5 (‘**presentation rating**’ = 5) continues to be the weakest, with an F1-score of 46%, crossing the 50% mark, further emphasizing the persistent difficulty in accurately classifying the highest presentation rating across all models. This is a decline from the 55% observed with Gradient Boosting and the 57% from Random Forest. Similarly, for classes 1, 3, and 4, the F1-scores hover around 53% to 56%, which is consistently lower than the performance levels achieved by the decision-tree models.

Compared to the Random Forest and Gradient Boosting models, this confusion matrix reflects greater dispersion across the classes. Misclassifications are more frequent and spread across multiple categories, particularly in class 5 (‘**presentation rating**’ = 5), where confusion with class 1 (‘**presentation rating**’ = 1) and class 4 (‘**presentation rating**’ = 4) is more pronounced. This aligns with the overall weaker performance shown in the classification report, supporting the conclusion that the Neural Network struggles more than the tree-based models to distinguish between the presentation ratings.

4.4.3 Feature importance

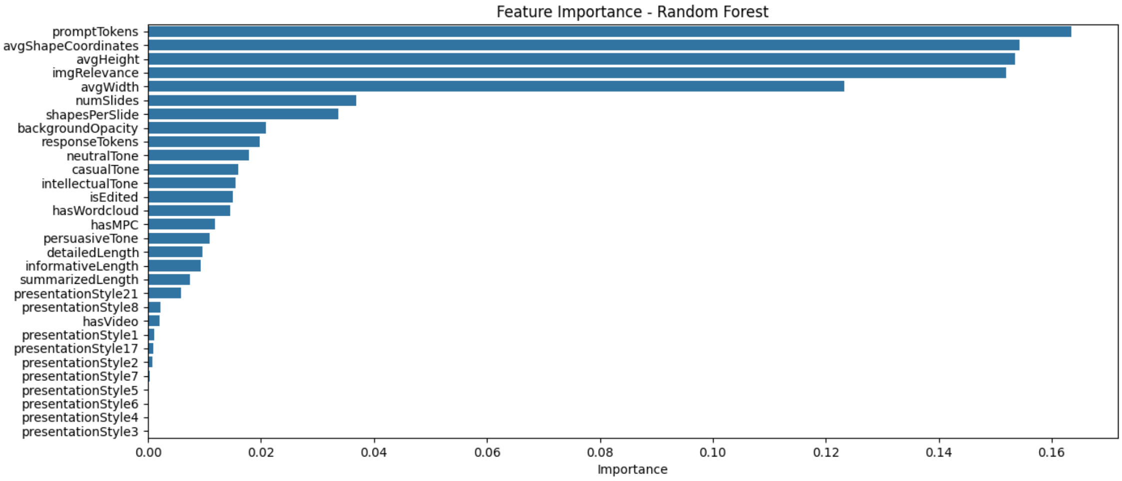


Figure 13

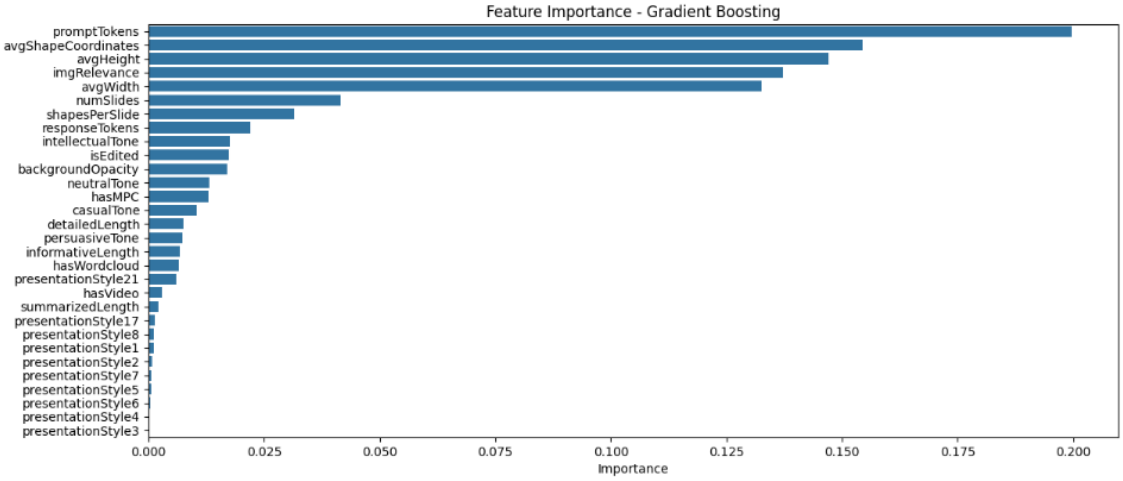


Figure 14

Both Random Forest (RFC) and Gradient Boosting (GBC) ranking plots (*Figure 13* and *Figure 14*), utilizing the built-in Gini-importance functionality, show the relative influence of various features during the decision-making process of the models. Both plots follow a very similar ranking hierarchy.

Both plots indicate that the most important feature is *promptTokens*, the length of the initial prompt given by the user. In the GBC ranking, the variable *promptTokens* substantially exceeds the influence of other features (around 20%) compared to the RFC ranking (around 16%). The average coordinates (combination of X and Y values) of presentation elements *avgShapeCoordinates* is ranked second, followed closely by *avgHeight,* the average height of presentations elements, and *imgRelevance,* using OpenAIs CLIP-model. The feature *avgWidth*, the average width of presentations elements, also shows notable importance. The number of slides and the average elements on a slide per presentation contribute to a lesser but still relevant extent. The tone of voice, which initially showed to have a high linear correlation with the number of slides, contributes less. It can be recognized that categorical (binary after one-hot encoding) features have less importance than continuous or discrete features, with a wider range of values, throughout this ranking.

Figure 18 (appendix) indicates that there are very few features that have any significant linear association with the '**presentation rating**'. The only notable is the variable *promptTokens*, the length of the initial prompt given by the user, that has a negative correlation of -15% to the target variable. Another feature is *imgRelevance,* denoting the relevance of images within the presentation, which has a positive correlation of 21%. Honourable mentions are *presentationStyle21* (9% negative linear correlation), binary indicating if the presentation is of template number 21; *responseTokens* (8% positive linear correlation), the character length of the LLM response to the initial prompt of the user; *isEdited* (9% positive linear correlation), binary feature indicating if the presentation has been edited by the user; *detailedLength* (8% positive linear correlation), binary feature denoting if the user wanted the presentation content to be detailed.

All other features do not have a higher (positive or negative) association with the '**presentation quality**' than 0,06.

4.5 Discussion

4.5.1 Results

The linear models, with an average F1-score of 67% and 66%, outperform the more complex non-linear neural network that scores an accuracy of 54% with a standard deviation of 1%, and a loss (on the *test data*) of approximately 1,22 with a standard deviation of 1%. As the target variable distinguishes between five classes, random guessing would give you and accuracy of around 20% (1 in 5). Although, the respective performance of predictive models is satisfiable, there is definite space for improvement. The inherent subjectivity of user assigned '**presentation ratings'** could be a factor leading to the 'mediocre' performance of the models. Most of these ratings are assigned on a whim, without deeper contemplation, which could explain factor into the 'plateau-ing' of pattern recognition capability of the models.

For both linear models (RFC & GBC), presentation rating 2 scores the highest F1-score (77%) and presentation rating 5 scores the lowest F1-score (57% and 55%). It is important to note that in the raw data, without any processing, the presentation rating 2 class had 160 instances, which merely made up 5% of the whole dataset. After oversampling (SMOTENC), 940 synthetic instances were added resulting in 1100 total instances of the second class. In contrast, the class of presentation rating 5 has no synthetic values whatsoever, it constitutes only real-world examples.

Although other minority classes also have a high number of synthetic values, rating class 2 represents a very small proportion of real samples (160 out of 1100). This often leads to a more homogeneous and less noisy class structure, making it easier for the models to learn and generalize the patterns associated with it. This is reflected in both the Random Forest and Gradient Boosting models, which achieve the highest F1-scores (77%) for this class.

Comparatively, rating class 5 represents is composed entirely of authentic samples without synthetic interpolation. This class likely exhibits greater variability and complexity in its feature space, reflecting the diverse range of characteristics that define high-quality presentations. The models may struggle to capture these subtle distinctions, resulting in the lowest F1-scores (57% and 55%) across both models.

Even though overall performance decreases, the results of the neural network show a similar pattern: class 2 scores the highest F1-score (62%) and class 5 scores the lowest (46%).

The performance gap highlights the issue that synthetic oversampling can improve the imbalance of datasets but does not necessarily address the complexity or heterogeneity of naturally underrepresented but highly diverse classes. This reinforces the need for further investigation into the data preparation process, especially the usage of different oversampling techniques, taking the approach of undersampling the majority class, or even applying a combination of both undersampling of the majority class and less severe oversampling of minority classes as this study [28].

Everything considered, this study serves as a proof of concept that software companies like *Sendsteps.ai* can benefit from incorporating machine learning techniques to gain a better understanding of underlying patterns influencing presentation ratings (or presentation quality as a whole).

4.5.2 Limitations and Future research

The feature ranking (Gini-importance) within the Random Forest and Gradient Boosting Classifier are popular because they are simple and fast to compute. However, they are biased in favour of variables with many possible split points and high minor allele frequency. “Fast approaches have been developed to debias impurity-based variable importance measures” [29] [30]. Furthermore, investigating feature importance metrics for neural networks such as NormLine [31] could provide a deeper understanding of feature importance within the feedforward neural network applied during this research. Not to mention the existence of a wide variety of other feature importance metrics, which are useful to further investigate which specific presentation characteristics influence the ‘**presentation rating**’ , as well as help during the feature selection process (e.g. SHAP [32], permutation importance [33]). Due to time constraints, this was not considered during this research.

Additionally, the actual textual content of the presentations (e.g. the actual initial prompt of the user, the response of the LLM or the full presentation and slide content) was not included into the features. Short or even longer text can be processed into dense text embeddings [34] potentially adding a lot of valuable information to the feature set. One could also make use of sentiment analysis techniques [35] and assign a sentiment score, which can then be incorporated into the feature set.

As the dataset of this study comprised rudimentary data, which was inherently not collected with the purpose to train, *Sendsteps.ai* should investigate different ways of collecting user feedback [36] to first of all get a better understanding what specific user needs are, but more importantly to create a more informative dataset of presentation characteristics directly linked to user feedback, which could ultimately result in better pattern recognition capabilities of predictive models.

Within the features set utilised during this study, certain adjustments to the SQL query (*Figure 15* in appendix) can be made to extract even more descriptive feature. An example of this would be distinguishing between shape types when counting them, as opposed to not differentiating between any of them (*numShapes)*. This could lead to also distinguishing between the average coordinate values of shape types, instead of combining all X and Y values across all presentation elements.

Integration of the predictive models and insights from feature ranking into the generation process of new presentations (or maybe beyond) was not delved into. To effectively implement the knowledge gained throughout this research and enhance the quality of presentations, as well as maximize user satisfaction in the future, emphasis needs to be placed on methods to create a constructive feedback loop.

Lastly, the problem of presentation ratings was approached as an multiclassification problem, although other approaches are possible. First, by combining both rating values, as introduced in the beginning of this paper (e.g. *content rating* and *images rating*), a continuous value is produced. Instead of rounding this number up to the nearest integer, the problem of predicting ratings can be tackled using logistic regression models.

By differentiating between three (*bad, fair* and *good*) or only two (*bad* and *good*) presentation classes, one can reduce the dimensionality of the target variable to a binary classification problem.

Alternatively, the choice can also be made to separate both ratings (*content* and *images*) and create isolated feature sets for both. This would transform this problem into two distinctly separate projects with vastly different approaches.

References

[1] R. Bommasani, "On the Opportunities and Risks of Foundation Models," Stanford Institute for Human-Centered Artificial Intelligence, CRFM, Stanford University, 2022.

[2] "Sendsteps.ai," [Online]. Available: https://www.sendsteps.com/en/ai/.

[3] H.-C. Chu, "Roles and research trends of artificial intelligence in higher education: A systematic review of the top 50 most-cited articles," Australasian Journal of Educational Technology, Taiwan, 2022.

[4] E. Dickey and A. Bejarano, "GAIDE: A Framework for Using Generative AI to Assist in Course Content Development," West Lafayette, Indiana, USA, 2024.

[5] M. Coumans, "How AI Interactive Presentation Tools are Revolutionizing the Education Landscape," The Learning Counsel, 23 April 2023. [Online]. Available: https://thelearningcounsel.com/articles/how-ai-interactive-presentation-tools-are-revolutionizing-the-education-landscape/.

[6] B. M. D., "Emerging Presentation Technologies," *Nursing Educational Perspectives,* vol. 45, no. 4, p. 260, 2024.

[7] L. Chen, C. W. Leong, G. Feng and C. M. Lee, "Using Multimodal Cues to Analyze MLA'14 Oral Presentation Quality Corpus: Presentation Delivery and Slides Quality," Educational Testing Service, Princeton, NJ, USA.

[8] F. Haider, L. Cerrato, N. Campbell and S. Luz, "PRESENTATION QUALITY ASSESSMENT USING ACOUSTIC INFORMATION AND HAND MOVEMENTS".

[9] K. Seongchan, J.-G. Lee and M. Y. Yi, "Developing information quality assessment framework for presentation slides," *Journal of Information Science,* vol. I, no. 27, 2016.

[10] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing," Stockholm, Sweden, 2018.

[11] P. Sedgwick, "Pearson's correlation coefficient," BMJ, London, UK, 2012.

[12] A. Radford, "Learning Transferable Visual Models From Natural Language Supervision," 2021.

[13] "Serper," [Online]. Available: https://serper.dev/.

[14] P. Hensman and D. Masko, "The Impact of Imbalanced Training Data for Convolutional Neural Networks," KTH ROYAL INSTITUTE OF TECHNOLOGY, Stockholm, Sweden, 2015.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research,* vol. 16, p. 321–357, 2002.

[16] H. Alshaher, "Studying the Effects of Feature Scaling in Machine Learning," North Carolina A&T State University, 2021.

[17] X. Wang, "Influence of feature scaling on convergence of gradient iterative algorithm," 2019.

[18] A. Parmar, R. Katariya and V. Patel, "A Review on Random Forest: An Ensemble Classifier," in *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI)*, 2018.

[19] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics,* vol. 7, 2013.

[20] M. H. Sazli, "A Brief Review of Feed-Forward Neural Networks," Ankara University, Department of Electronics Engineering, Ankara, 2006.

[21] "scikit-learn Random Forest Classifier," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[22] "scikit-learn Gradient Boosting Classifier," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.

[23] "scikit-learn," [Online]. Available: https://scikit-learn.org/stable/index.html.

[24] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei and S.-H. Deng, "Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization," *Journal of Electronic Science and Technology,* vol. 17, no. 1, pp. 26-40, 2019.

[25] "OPTUNA," [Online]. Available: https://optuna.org.

[26] D. P. Kingma and J. L. Ba, "ADAM: A Method for stochastic optimization," 2017.

[27] S. Sharma, S. Sharma and A. Athaiya, "Activation functions in neural networks," *International Journal of Engineering Applied Sciences and Technology,* vol. 4, no. 12, pp. 310-316, 2020.

[28] R. Mohammed, J. Rawashdeh and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," Jordan University of Science and Technology, Irbid, Jordan.

[29] S. Nembrini, I. R. Koenig and M. N. Wright, "The revival of the Gini importance?," 2018.

[30] C. Strobl, A.-L. Boulesteix, A. Zeileis and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics,* 2007.

[31] "NORMLIME: A NEW FEATURE IMPORTANCE METRIC FOR EXPLAINING DEEP NEURAL NETWORKS," in *ICLR*, 2020.

[32] H. Wang, Q. Liang, J. T. Hancock and T. M. Khoshgoftaar, "Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods".

[33] A. Altmann, L. Tolo, O. Sander and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Data and text mining,* vol. 26, no. 10, p. 1340–1347, 2010.

[34] J. X. Morris, V. Kuleshov, V. Shmatikov and A. M. Rush, "Text Embeddings Reveal (Almost) As Much As Text," Department of Computer Science, Cornell University, 2023.

[35] D. M. D, S. C and A. Ganesh, "Sentiment Analysis: A Comparative Study On Different Approaches," Department of CSE, Vidya Academy of Science and Technology, Thrissur 680501, India.

[36] A. Fabijan, H. H. Olsson and J. Bosch, "Customer Feedback and Data Collection Techniques in Software R&D: A literature review," Chalmers University of Technology, Department of Computer Science & Engineering, Malmö, Sweden, 2015.

Appendix

```sql
SELECT
        F.createdAt,
        ROUND(AVG(F.rating), 0) AS rating,
        COUNT(DISTINCT S.slideIndex) AS numSlides,
        COUNT(distinct SS.id) AS numShapes,
        COUNT(distinct S.filename) AS numImgs,
        P.isEdited,
        (EXISTS (SELECT * FROM feedback z WHERE z.presentationId = F.presentationId
        AND z.text IS NOT NULL)) AS hasExplanation,
        (EXISTS (SELECT * FROM slide_shapes xx JOIN slides x ON xx.slideId = x.id
                WHERE x.presentationId = F.presentationId AND xx.deletedAt IS NULL AND xx.type = 'video')) AS hasVideo,
        (EXISTS (SELECT * FROM slide_shapes xx JOIN slides x ON xx.slideId = x.id
                WHERE x.presentationId = F.presentationId AND xx.deletedAt IS NULL AND xx.type = 'wordcloud')) AS hasWordcloud,
        (EXISTS (SELECT * FROM slide_shapes xx JOIN slides x ON xx.slideId = x.id
                WHERE x.presentationId = F.presentationId AND xx.deletedAt IS NULL AND xx.type = 'mpc_options')) AS hasMPC,
        ROUND(AVG(SS.x), 1) AS avgX,
        ROUND(AVG(SS.y), 1) AS avgY,
        ROUND(AVG(SS.width), 1) AS avgWidth,
        ROUND(AVG(SS.height), 1) AS avgHeight,
    P.presentationStyleId,
    S.backgroundOpacity,
    #COUNT(distinct PC.id) AS aiCalls,
        #SUM(prompt_tokens) AS sumPrompt_tokens,
    #SUM(response_tokens) AS sumResponse_tokens,
    #SUM(PC.total_tokens) AS sumTotalTokens,
    CHAR_LENGTH(PA.initialPrompt) AS initialPromptTokens,
    CHAR_LENGTH(PA.response) AS responseTokens,
    PA.`language` AS 'language',
    PA.`length`,
    PA.toneOfVoice,
    GROUP_CONCAT(distinct S.filename SEPARATOR '||' ) AS imgURLs,
    PA.`subject`
FROM feedback F
LEFT JOIN slides S ON F.presentationId = S.presentationId
JOIN slide_shapes SS ON S.id = SS.slideId
JOIN presentations P ON F.presentationId = P.id
#JOIN presentation_ai_calls PC ON PC.presentationId = F.presentationId
JOIN presentation_ai_jobs PA ON PA.presentationId = F.presentationId
WHERE F.rating IS NOT NULL
AND S.isDeleted = 0
AND P.isDeleted = 0
AND SS.deletedAt IS NULL
AND S.filename IS NOT NULL
AND PA.toneOfVoice IS NOT NULL
GROUP BY F.presentationId
```

Figure 15 (SQL Query)

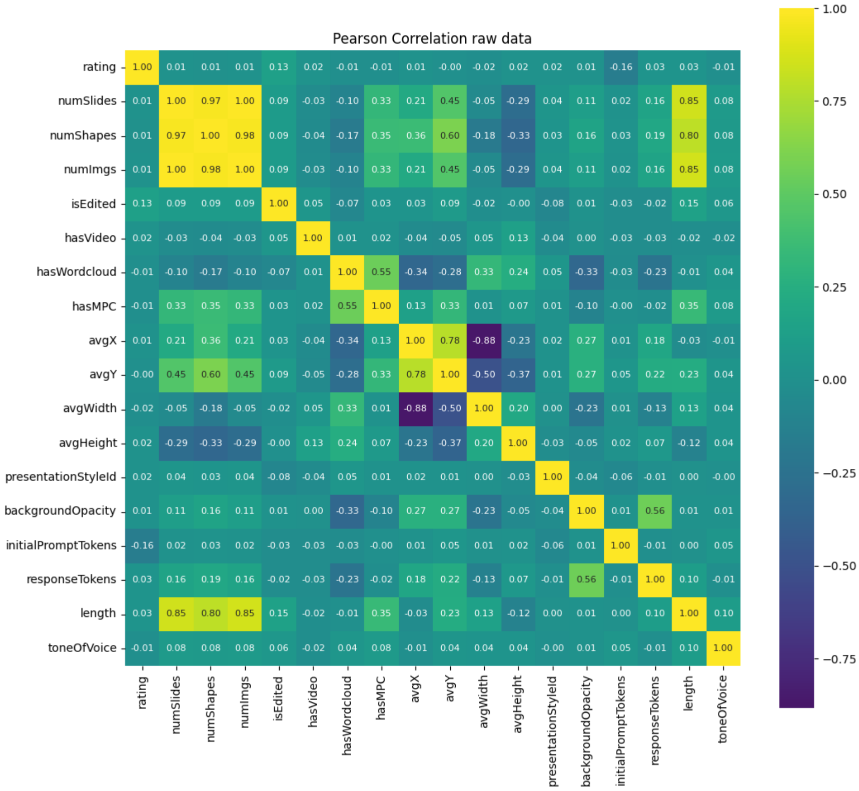


Figure 16 (Pearsons's correlation matrix of unprocessed data)

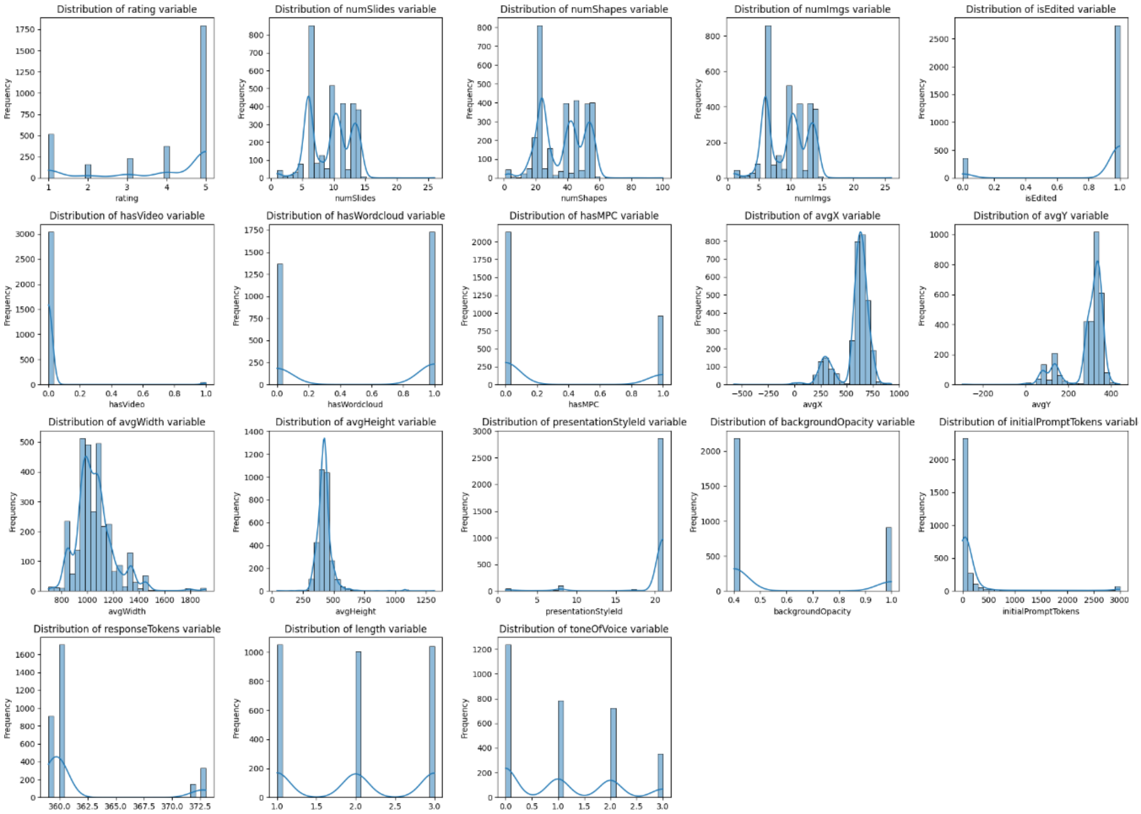


Figure 17 (feature distribution of unprocessed data)

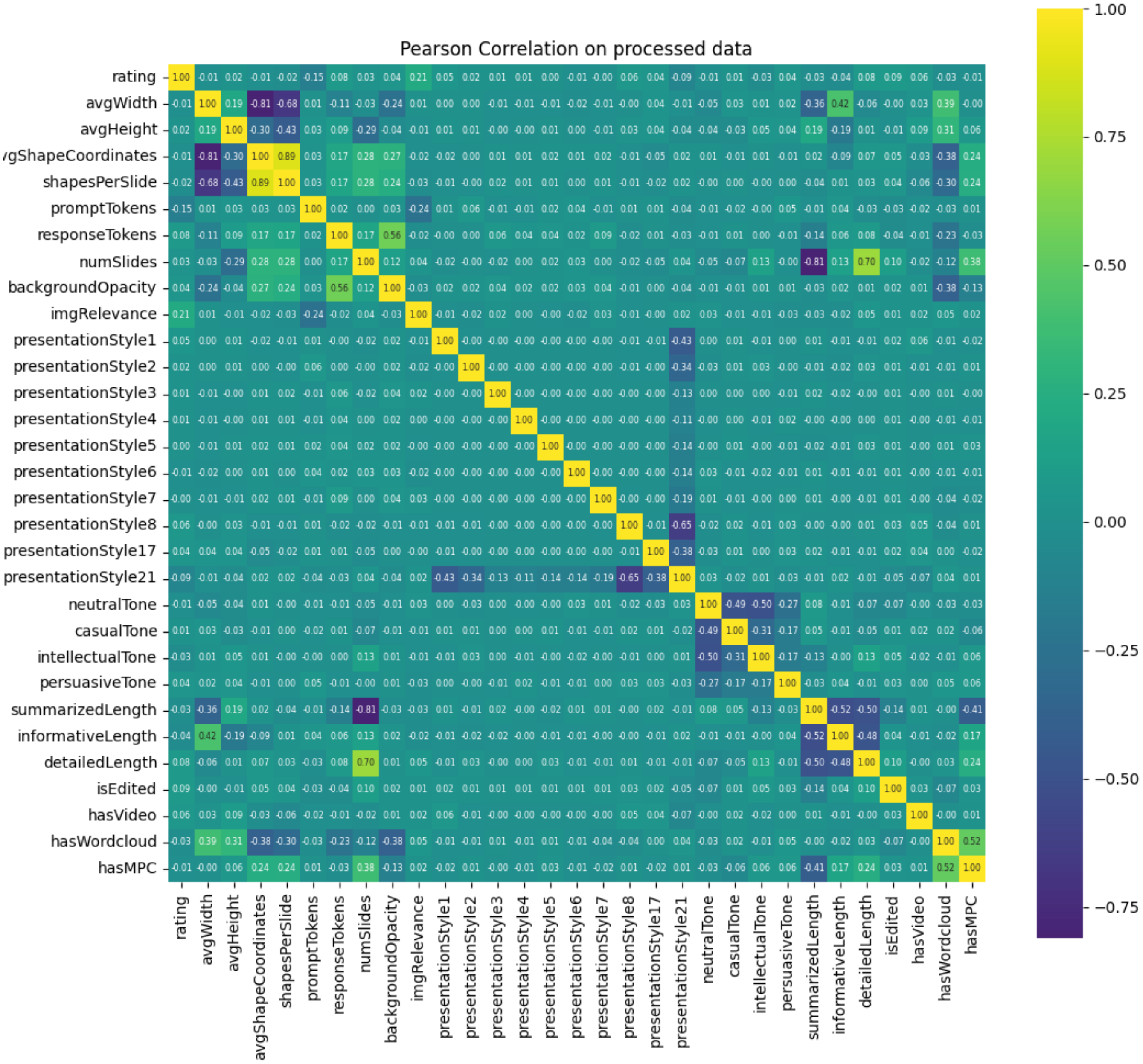
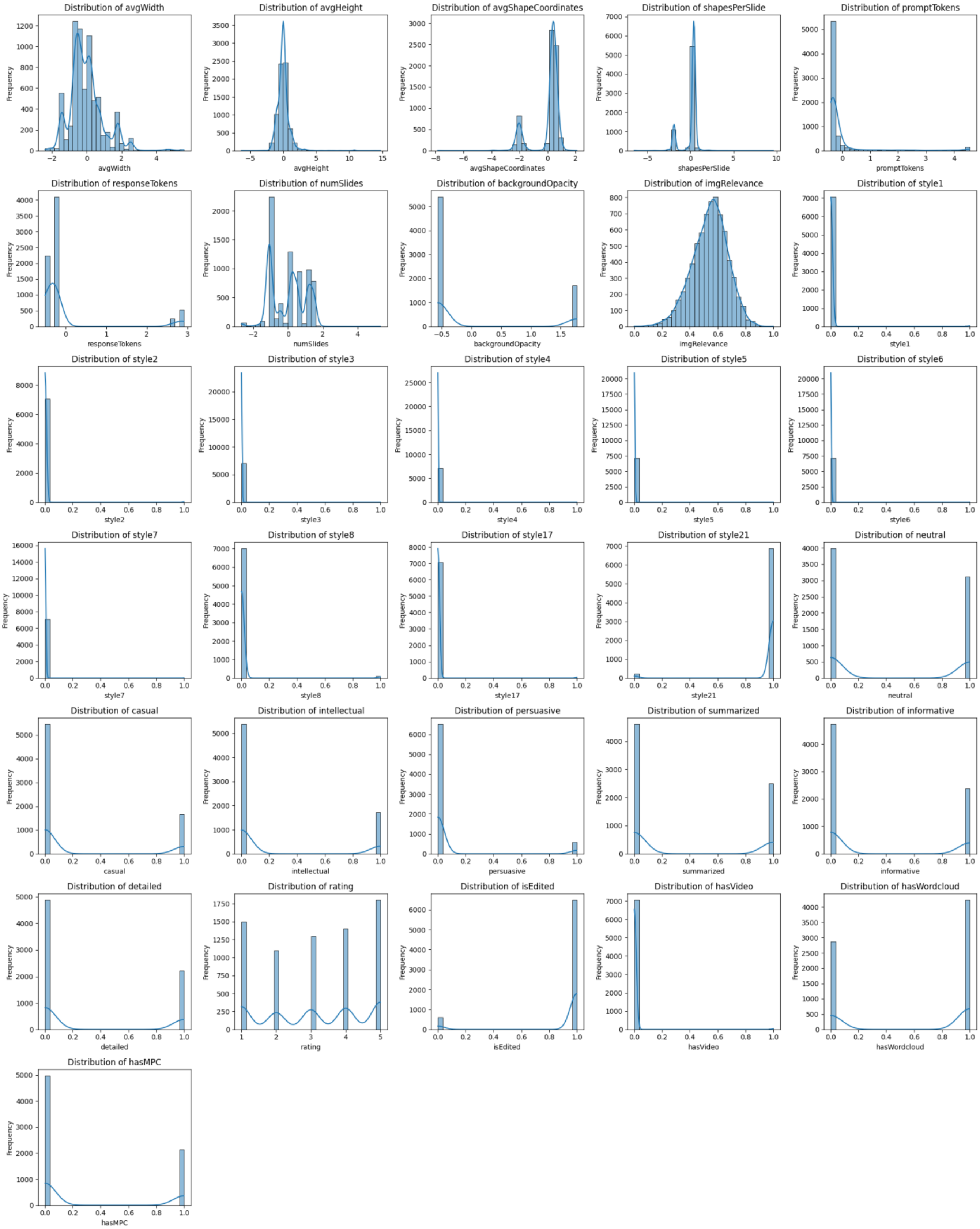


Figure 18 (Pearsons's correlation matrix of processed data)



Distribution of avgWidth

Distribution of avgHeight

Distribution of avgShapeCoordinates

Distribution of shapesPerSlide

Distribution of promptTokens

Distribution of responseTokens

Distribution of numSlides

Distribution of backgroundOpacity

Distribution of imgRelevance

Distribution of style1

Distribution of style2

Distribution of style3

Distribution of style4

Distribution of style5

Distribution of style6

Distribution of style7

Distribution of style8

Distribution of style17

Distribution of style21

Distribution of neutral

Distribution of casual

Distribution of intellectual

Distribution of persuasive

Distribution of summarized

Distribution of informative

Distribution of detailed

Distribution of rating

Distribution of isEdited

Distribution of hasVideo

Distribution of hasWordcloud

Distribution of hasMPC

Figure 19 (feature distributions of processed data)

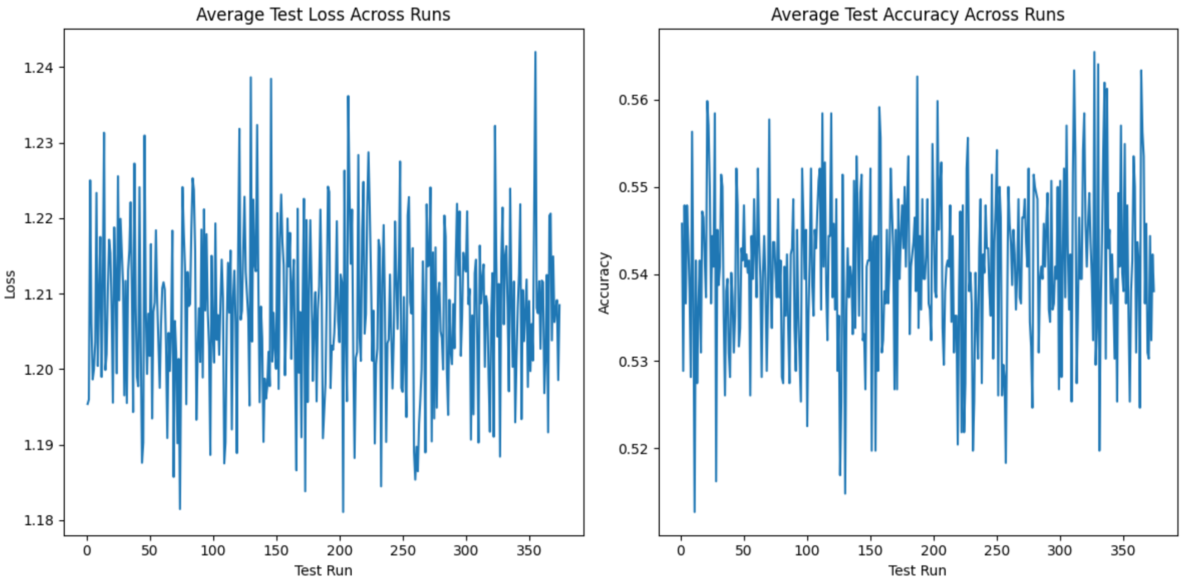


Figure 20 (Average Loss and Accuracy of Neural Network on Test data)