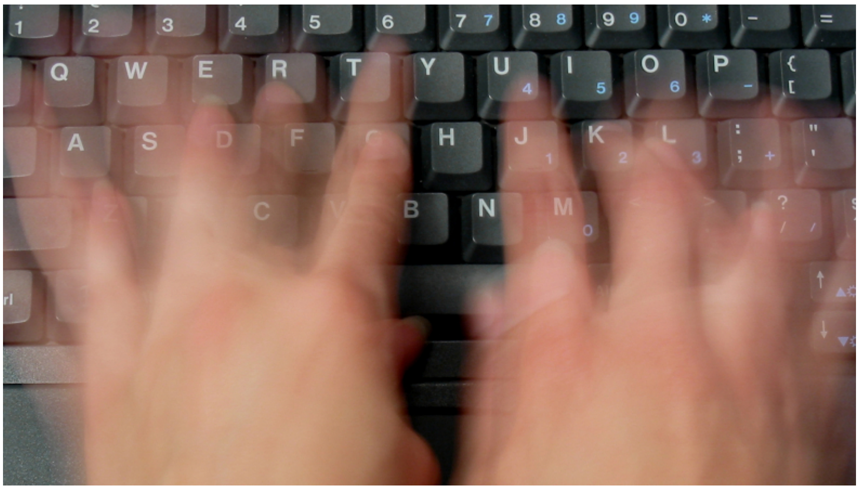
15.093 Machine Learning with An Optimization Lens

Fall 2023



Contextless Writing Quality Prediction

Lucas Goh, Kevin Sheng

Contents

1 Introduction

1.1 Problem Statement

Writers may use different techniques to plan and revise their work, especially in a test setting. They may demonstrate distinct pause patterns or allocate time strategically throughout the writing process. In reality, many of these typing behaviors may influence writing quality. However, most writing evaluation tools focus on only the final essay. This exposes the original work of the essay to a third party, let it be a tool or a human, exposing the author to privacy risks.

1.2 Purpose and Scope

In our project, we aim to explore the potential of writing quality prediction with contextless data. These models may help identify relationships between learners' writing behaviors and writing outcomes. Given that most current writing assessment tools mainly focus on the final written products, this may help direct learners' attention to their text production process and boost their metacognitive awareness in writing. Based on a deeper understanding of the writing process, ultimately, the learners can improve their writing quality.

2 Data

Our dataset originates from a competition held by Vanderbilt University and the Learning Agency Lab. The dataset comprises log inputs from about 2500 users (each user has around 3500 events), such as keystrokes and mouse clicks, taken during the composition of an essay. In total, there are approximately 8.4 million rows in our dataset.

Each essay was scored on a scale of 0 to 6. The goal is to predict the score of an essay only from its log of user inputs. To prevent reproduction of the essay text, all alphanumeric character inputs have been replaced with the "anonymous" character q; punctuation and other special characters have not been anonymized.

2.1 Data Description

The log data has the following entries:

- event_id - The index of the event, ordered chronologically
- down_time - The time of the down event in milliseconds
- up_time - The time of the up event in milliseconds
- action_time - The duration of the event (the difference between down_time and up_time)
- activity - The category of activity to which the event belongs to
  - Nonproduction - The event does not alter the text in any way
  - Input - The event adds text to the essay
  - Remove/Cut - The event removes text from the essay
  - Paste - The event changes the text through a paste input
  - Replace - The event replaces a section of text with another string
  - Move From $[x_1, y_1]$ To $[x_2, y_2]$ - The event moves a section of text spanning character index $[x_1, y_1]$ to a new location $[x_2, y_2]$
- text_change - The text that changed as a result of the event (in anonymized "q"s)

Understanding the data is critical for analysis. Specifically, a down event refers to an event that occurs when a key is pressed or "down." It indicates the initial action of pressing a key on the keyboard. An up event, similarly, refers to an event that occurs when a key that was previously pressed is released or "up." It indicates the action of releasing the key. These columns are critical to feature engineering later in the report.

2.2 Exploratory Data Analysis (EDA)

As our data consists of logs of user inputs, it is not presented in a way that allows people to have a quick intuition about the writing process. Thus, our EDA here allows us to have some intuition about the data structure.

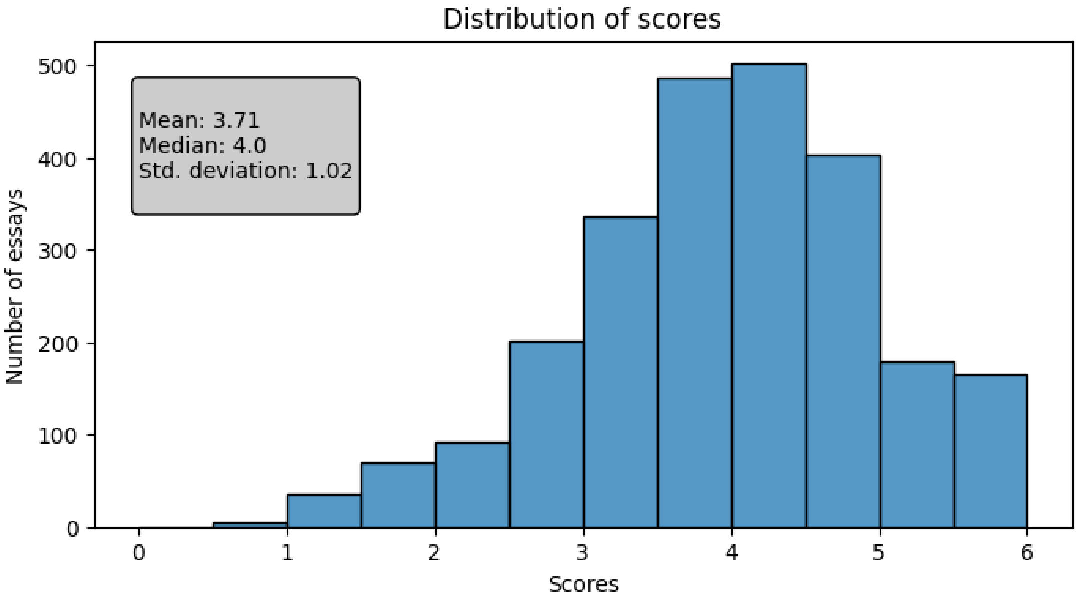


Figure 1: Score Distribution

Figure 1 displays the distribution of scores received by the essays users wrote. The distribution is relatively symmetric, slightly left skewed. A score of 3.5-4 is considered to be an average grade. There are lots of users who received scores between 3 to 5, while there are far fewer users who got a score higher than 5, which could be considered a high score. A few users get a score of 0.5, which is the lowest score awarded.

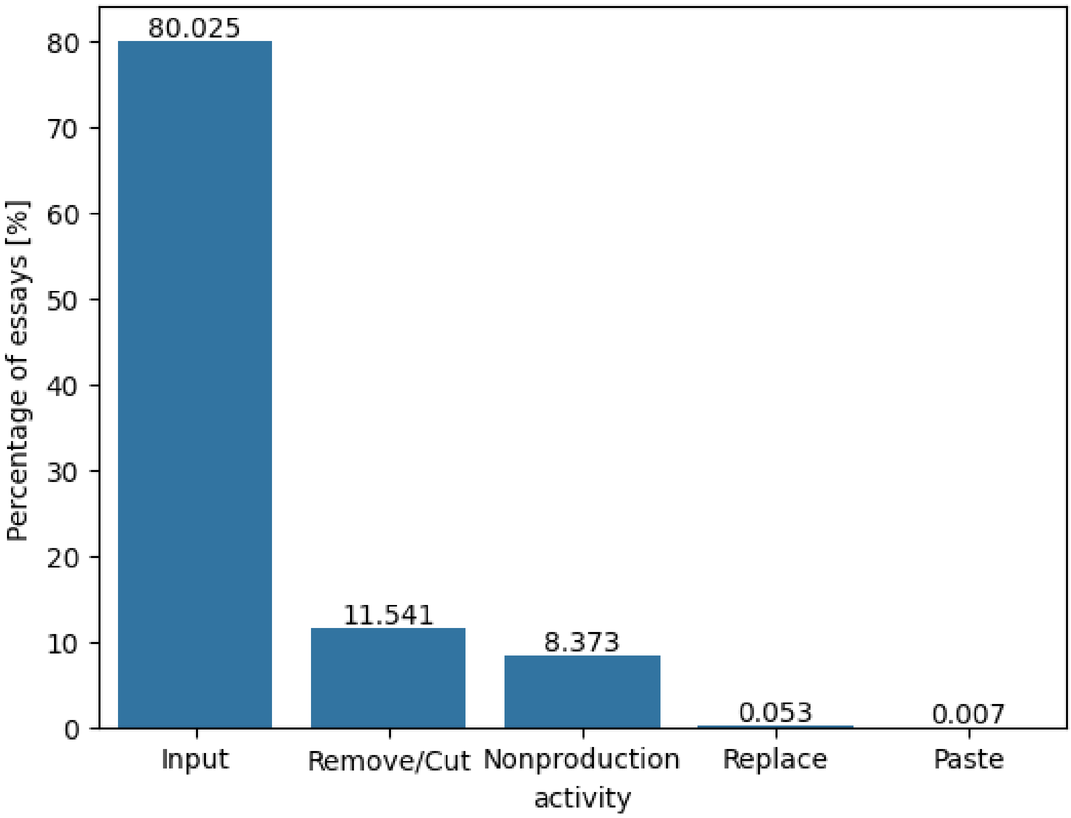


Figure 2: Activity Distribution

Figure 2 illustrates the distribution of activities. As expected, most of the activities are in the "Input" category. Surprisingly, there are considerable proportions of activities that fall under "Nonproduction", which indicates a potential for efficiency improvements.

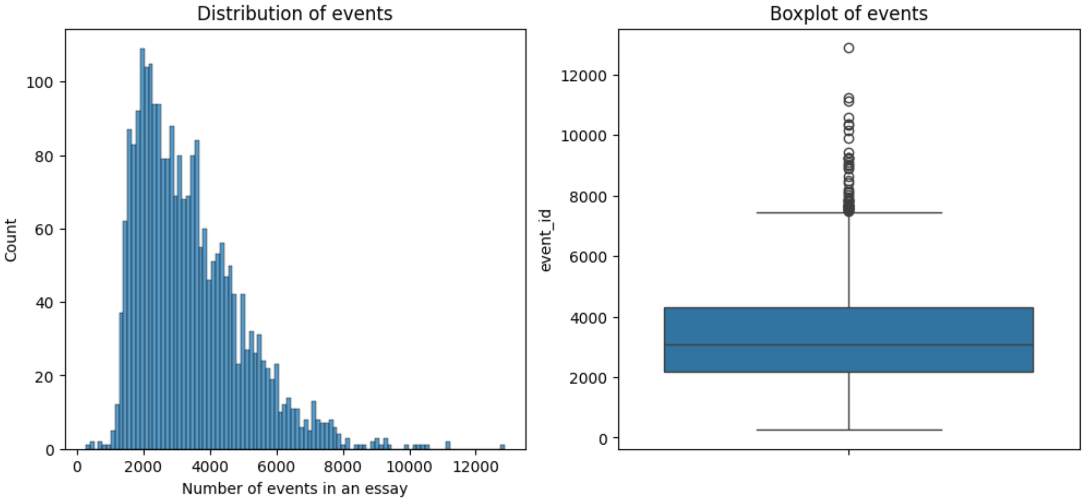


Figure 3: Events Distributions

Figure 3 shows a right-skewed distribution of events in essays, with most essays having around 2000 events. The boxplot reveals a median below 2000 events and outliers indicating significantly higher activity on some essays.

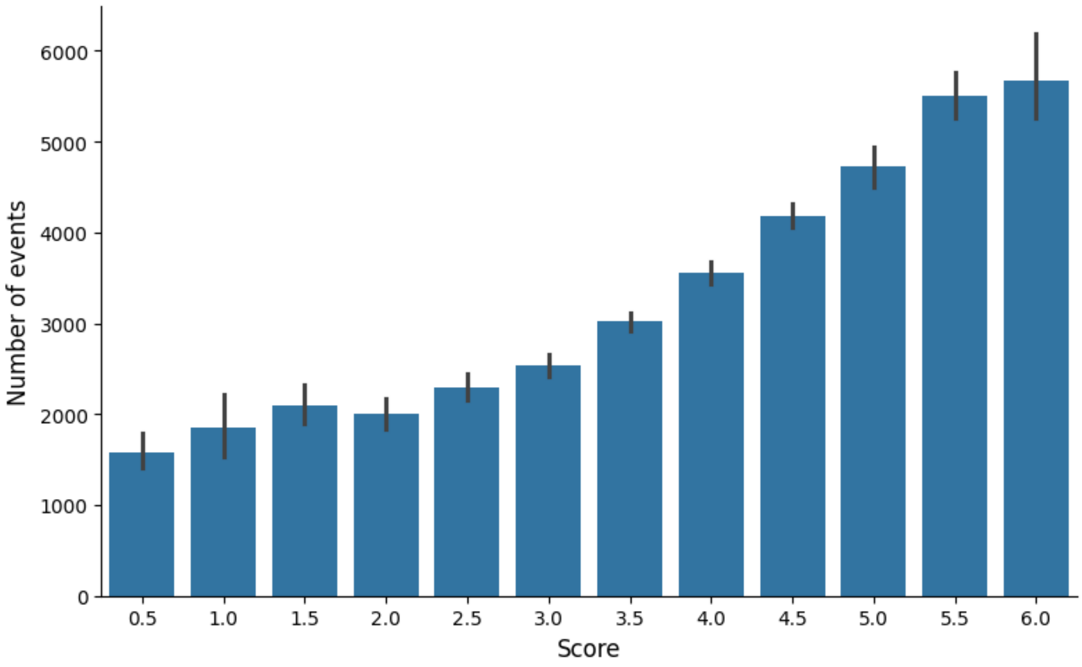


Figure 4: Event Distribution Against Writing Score

Finally, from Figure 4, it is notable that the number of events is positively correlated with final scores. Essays with more events are typically longer and edited more. This piece of insight makes intuitive sense and is used to formulate our baseline model.

3 Feature Engineering

One of the main challenges is to effectively engineer features from the log input data. Otherwise, it is nearly impossible to derive insights about the writing procedures. The feature engineering process involves three iterations: first, the data is aggregated to the per-user level using an array of summary statistics; second, to fully employ the time information of the log data, we engineered features related to latency and pauses; finally, to grasp the high-level structure of sentences and paragraphs, we recovered the structure of the essays from the log data and created features based on them. In total, we generated 394 features that would be used in the final model training.

3.1 Statistical Aggregation

The first iteration of feature engineering mainly involves using summary statistics to aggregate the characteristics of the essays. Specifically, we considered features related to activity counts, event counts, text change counts, punctuation counts, and word counts. These features set the basis of the project and will be used in Stage 1 of the modeling.

3.2 Time Series Features

Notably, the log data provides information on the timestamp of each action in the essay-writing process. In this step, we generated features related to the time and duration of events captured in log data. For example, the features can describe the duration and number of pauses in the writing process, indicating possible contemplation during the process. On the other hand, some features describe the lagging time in the keyboard inputs, potentially reflecting the user's typing skill proficiency.

3.3 Essay Reconstruction

Although the essay content is anonymized, understanding the general structure at the sentence or paragraph level will provide valuable insights into the quality of writing. For example, the length of paragraphs can potentially indicate the complexity of the essay. The variation in sentence length, intuitively, can affect the rhythm and readability of the essay. Thus, we decided to reconstruct the essay with three key columns in the log data: "activity", "text_change", and "cursor_position". The "activity" column has a list of possible actions taken by the user, such as "Input", "Remove/Cut", "Paste", etc. For each type of activity, we were able to recover the essay changes incorporating information on the length of the changes and the place of the changes from the "text_change" and "cursor_position" columns. Then, we are able to generate sentence-level and paragraph-level features using s series of summary statistics such as mean, median, standard deviation, skewness, etc.

4 Methods

In this section, we delineate the three stages of modeling in the project. For the purpose of training and measuring the performance of our model, our data set is split into a training set and a testing set using an 80-20 split.

4.1 Baseline

From subsection 2.2, we observe that there appears to be a positive correlation between the total number of events and the scores received. Intuitively, more typing events should lead to longer essays and more edits during the process, which are common characteristics of essays with good quality. Thus, as a baseline model, we decided to run a simple linear regression using the number of events each essay has as the explanatory variable.

4.2 Modeling Without Aggregation Features

In the first iteration of model experimentation, we utilized only the features from summary statistics, without including features from time series information or essay reconstruction. The models applied in this stage are CART, Random Forest, and XGBoost. These models would be also applied using the full sets of features, allowing us to benchmark the effects of more complex feature engineering. All the models used 5-fold cross-validation to fine-tune the hyperparameters.

4.3 Modeling With All Features

In our final modeling stage, we used all the features obtained through feature engineering. For completeness purposes, we chose an extensive collection of both interpretable models and black-box models. The interpretable models applied using these features are CART, Optimal Regression Tree (OCT), Optimal Regression Tree with Hyperplanes (OCT-H), and Lasso Regression. We also applied less interpretable models aiming for a better prediction performance. They include Random Forest, XGBoost, LightGBM (LGBM), and CatBoost. All the models used 5-fold cross-validation to tune hyperparameters. OCT and OCT-H are also ran with sampling features selection to optimize performance. More precisely, a "brute-force" method is used by running the models on 1000 samples of feature set, with each sample selecting a feature with 0.2 probability. This is done to optimize the stability of the tree using a heuristic method.

5 Results

To measure the performance of our models, we use two main metrics:

- Root Mean Square Error (RMSE): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- R^2: $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$

RMSE measures how well the model's predictions are compared to the actual observed outcomes, while R^2 measures the proportion of variance in the dependent variable that is predictable from the independent variables.

Table 1 shows the model performance of Baseline, CART, Random Forest, and XGBoost using only the features engineered using the typing logs directly. We observe that all three models exceed the baseline model in both RMSE and R^2. In particular, Random Forest and XGboost have superior performances, lowering the test RMSE by more than 15% compared to the baseline.

Method	RMSE	R^2
Baseline	0.831	0.361
CART	0.733	0.502
Random Forest	0.694	0.554
XGBoost	0.692	0.597

Table 1: Results when Modeling without Aggregation Features

Table 2 displays the results from the final models with the full sets of the features. When compared to the results when modeling without aggregation features, results from our final models demonstrate that, by extracting features from the essay reconstructed, we improved the performance of all three models (namely, CART, Random Forest, and XGBoost). Intuitively, this means that sentence-and-paragraph-level features contribute to the understanding of the quality of the essays too, on top of the features of typing behaviors.

Method	RMSE	R^2
CART	0.716	0.525
Random Forest	0.642	0.618
OCT	0.694	0.554
OCT-H	0.728	0.509
Lasso	0.649	0.609
XGBoost	0.636	0.626
LGBM	0.636	0.626
CatBoost	0.633	0.629

Table 2: Modeling with All Features

Additionally, we observe that CatBoost outperforms all other models, lowering the test RMSE by 24.1% and improving the test R^2 by 97% when compared to the baseline model. Other gradient-boosted methods such as XGBoost and LGBM models has similar performance, and significantly outperform the other models. The comparison is illustrated in the figure below.

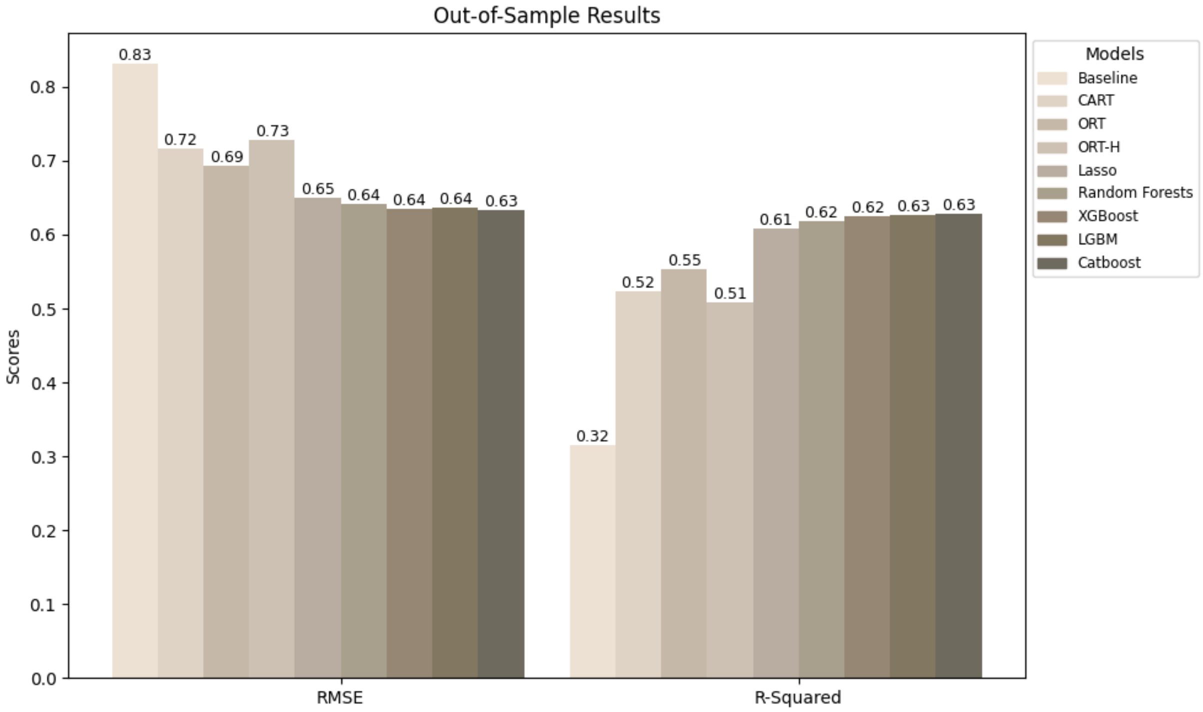


Figure 5: Models' Out-of-Sample Performances

6 Insights

In addition to prediction performance, our model also provides meaningful, interpretable insights into how writing processes are correlated to writing outcomes. SHAP (SHapley Additive exPlanations) values can be used to shed light on the models' decision-making processes. Here, we investigate the SHAP values of our best-performing model, CatBoost.

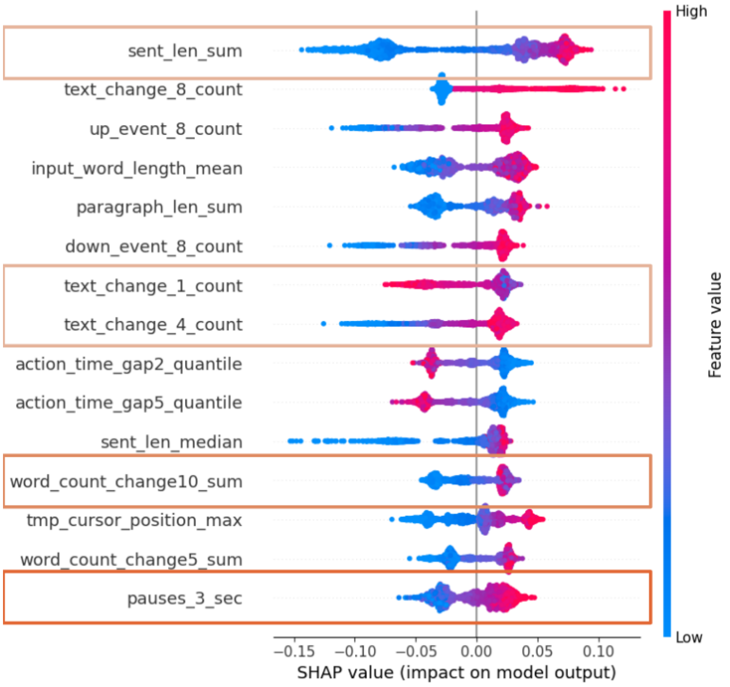


Figure 6: SHAP Plot of CatBoost

From Figure 6, we observe three key insights here. First of all, we observe that essays with longer sentences tend to lead to an essay of higher quality with a significant impact. This observation makes intuitive sense, as better writers may convey complex ideas or a series of thoughts that are related in longer sentences. Secondly, as *word_ count_ change10_ sum* here means the change in text count in 10 "events", we observe that the faster a writer writes, the better the writing quality is. This also makes sense, as more experienced writers spent considerable time honing their craft. This allows them to write more quickly because they are familiar with the structure, style, and nuances of good writing. Last but not least, we observe that the number of "at least 3-second" pauses also has an impact on the writing quality. However, the impact of the pauses on writing quality is more ambiguous here. While the effect on writing quality appears to be positive on average, we observe that a large number of them have a negative effect on writing quality as well. It is likely that the effect here is dependent on several other factors, such as writing speed after the pauses, pause location in writing, etc.

Together, the features above characterize the features of a good writer: one who can write long sentences, write fast, and pause to process and compose thoughts at certain times.

7 Conclusion

In this project, we set out to build a contextless writing prediction tool, a task that has garnered little effort before. Starting out with granular typing log features that have little usage on their own, we devised several methods to engineer a comprehensive set of 394 features, including reconstructing the essay structure and extracting key features from it.

For modeling, we experimented with an extensive collection of 8 prediction models. From our results, we observe that Catboost outperforms the others by lowering the test RMSE by 24.1% and improving the test R^2 by 97% compared to the baseline model.

Finally, the insights obtained from the models provide meaningful suggestions for learners to improve their writing qualities. Although it is noteworthy that correlation is not causation (for example, a longer essay does not necessarily lead to a better essay), the models do suggest that learners can reverse engineer the logic behind the correlations to enhance writing habits and skills.

To conclude, our novel approach of contextless writing quality prediction based on writing behavior and structure sheds light on the potential of writing quality prediction, even without text-based data. We show that these models have decent prediction power, and they have the potential to minimize privacy risks. In the future, one can possibly consider a model writing quality with both data on typing behavior and limited context (for ex. words with TF-IDF above a certain threshold), which can potentially achieve an optimal balance of accuracy and privacy.

8 Contribution

Our project was a collaborative effort at every stage. Initially, we jointly decided on the project topic and gathered the necessary data. Lucas then took the lead on exploratory data analysis and developed models using OCT, OCT-H, Lasso, LGBM, and CatBoost. Concurrently, Kevin concentrated on feature engineering and constructing models utilizing baseline, CART, Random Forest, and XGBoost techniques. For the final deliverables, both team members contributed equally to the development of the presentation slides and the final report. Specifically, Lucas focused more on crafting the slides, while Kevin dedicated his efforts to writing the detailed report.