Linking Writing Processes to Writing Quality

Matt Burns
College Of Science
Utah State University
Logan, Utah, USA
a02398138@usu.edu

Linhao Wu
College Of Science
Utah State University
Logan, Utah, USA
a02424787@usu.edu

Abstract

In the quest to link writing processes to writing quality, this study delves into the intriguing relationship between typing behavior and essay outcomes. Leveraging a dataset of approximately 5000 keystroke logs, this paper investigates the hypothesis that the micro-level actions inherent in writing—such as pausing patterns, revisions, and time allocations—affect overall writing quality. Previous research in this domain has often been constrained by small datasets and a narrow focus on a limited range of process features. Our work expands on this by employing advanced machine learning techniques, including Random Forest, CatBoost, and XGBoost algorithms, to analyze a large corpus of keystroke data. The evaluation of model performance is based on Root Mean Squared Error (RMSE) metrics, aiming to minimize prediction error. The models are further assessed for efficiency, considering computational cost—a crucial factor for their practical application in educational settings. The findings of this study are expected to contribute valuable insights for the development of automated writing evaluation tools and enhance the instructional practices surrounding writing education.

1. Introduction

The ability to effectively evaluate writing quality is crucial in educational settings, where the assessment of students' essays plays a pivotal role. Traditional grading processes are often subjective and labor-intensive, prompting a demand for more objective, efficient approaches. Recent advances in machine learning offer promising avenues to address these challenges, particularly through the analysis of keystroke logging data. Keystroke dynamics provide a rich, yet underutilized, dataset that can yield insights into the cognitive and mechanical aspects of writing processes, potentially correlating with the final quality of written essays.

This study capitalizes on the wealth of data derived from keystroke logs, exploring the relationship between the typing behaviors captured within these logs and the resultant essay quality. With a dataset comprising approximately 5000 logs, including various keystroke actions and temporal patterns, we seek to decode the underlying structure of writing processes that contribute to the production of high-quality essays.

Our research aims to transcend the boundaries of existing literature, which has largely been constrained by limited datasets and a narrow focus on few process features. By employing sophisticated machine learning algorithms, such as Random Forest, CatBoost, and XGBoost, we attempt to decode the nuanced patterns of writing behavior that influence essay quality. The performance of these models is meticulously evaluated using metrics such as the Root Mean Squared Error (RMSE), providing a robust assessment of their predictive accuracy.

Moreover, the models are scrutinized for their computational efficiency, a non-trivial consideration given the limited resources in educational contexts. This dual focus on accuracy and efficiency underscores the practical significance of our investigation, aiming to deliver actionable insights that can be readily deployed in educational technology platforms.

2. Accomplishments

We use a Linear Stacking Model to stack these three sub-models together, achieving better predictive performance.

Stacking is an ensemble learning technique that combines multiple base classifiers to obtain a more powerful overall predictive model. It employs three base classifiers (rf_model, cat_boost, xgb_boost) to generate prediction results and passes these predictions to the meta-classifier lr to produce the final classification outcome. This approach typically helps improve model performance, particularly in complex problems.

Our final score is 0.649 points, with only a difference of 0.076 points from the first place on the leaderboard.

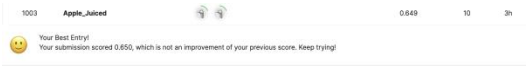

Figure 1. The score and ranking on Kaggle

3. Dataset analysis

3.1 Data Collection Procedure

For collecting keystroke Data, participants of this project were hired from Amazon Mechanical Turk, a crowdsourcing platform. They were invited to log onto a website that housed a demographic survey, a series of typing tests, an argumentative writing task, and a vocabulary knowledge test. Participants were required to use only computers with a keyboard.

During the argumentative writing task, participants were asked to write an argumentative essay within 30 minutes in response to a writing prompt adapted from a retired Scholastic Assessment Test (SAT) taken by high school students attempting to enter post-secondary institutions in the United States. To control for potential prompt effects, four SAT-based writing prompts were used and each participant was randomly assigned one prompt. Prior to the writing task, instructions were presented on the integral components in an argumentative essay (e.g., introduction, position, reasons and evidence) along with descriptions of their functions in argumentation. The instructions pages also introduced a set of rules for the writing task. These include that participants should write an essay of at least 200 words in 3 paragraphs and that they should not use any online or offline reference materials. To make sure participants stayed focused on the task during writing and to track behavior, the writing task page issued warnings whenever the participant was detected inactive for more than 2 minutes or moved to a new window in the process of writing. A screenshot of the writing task page is presented below.

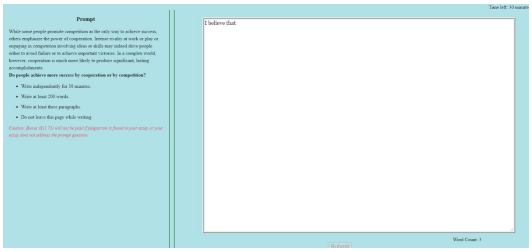

Figure 2. Writing task page

To collect participants' keystroke information during the argumentative writing task, a keystroke logging program was written in vanilla JavaScript and was embedded in the script of the website built for this project. The program listened to the keystroke and mouse events in the designated text input area using JavaScript's addEventListener method. It also logged the time stamp and cursor position information for each keystroke or mouse operation. Additionally, this program simultaneously aggregated the logged information and identified operation types (e.g., input, delete, paste, replace) and reported text changes in the writing process. The table below provides an example output of keystroke logging information reported by the program. As shown, Event ID indexes the keyboard and mouse operations in chronological order. Down Time denotes the time (in milliseconds) when a key or the mouse was pressed while Up Time indicates the release time of the event. Action Time represents the duration of the operation (i.e., Up Time - Down Time). Position registers cursor position information to help keep track of the location of the leading edge. Word Count displays the accumulated number of words typed in. Additionally, Text Change shows the exact changes made to the current text while Activity indicates the nature of the changes (e.g., Input, Remove/Cut).

An Example Dataframe of Keystroke Logging

Information:

Event ID	Down Time	Up Time	Action Time	Event	Position	Word Count	Text Change	Activity
1	30185	30395	210	Leftclick	0	0	NoChange	Nonproduction
2	41006	41006	0	Shift	0	0	NoChange	Nonproduction
3	41264	41376	112	I	1	1	I	Input
4	41556	41646	90	Space	2	1		Input
5	41815	41893	78	b	3	2	b	Input
6	42018	42096	78	e	4	2	e	Input
7	42423	42501	78	l	5	2	l	Input
8	42670	42737	67	i	6	2	i	Input
9	42873	42951	78	e	7	2	e	Input
10	43041	43109	68	v	8	2	v	Input
11	43289	43378	89	Space	9	2		Input
12	44560	44605	45	Backspace	8	2		Remove/Cut
13	44661	44762	101	e	9	2	e	Input
14	44954	45032	78	Space	10	2		Input
15	45325	45381	56	t	11	3	t	Input
16	45460	45538	78	h	12	3	h	Input
17	45640	45730	90	a	13	3	a	Input
18	45741	45808	67	t	14	3	t	Input
19	45933	46011	78	Space	15	3		Input

Figure 3. Keystroke Logging Information

3.2 Keystroke Measures

Production Rate

The rate of written language production can be measured by counting the number of characters, words, clauses, sentences, or T-units in the writing process or written product generated per unit of time. Example measures are as follows.

- number of characters (including spaces) produced per minute during the process
- number of characters (including spaces) produced per minute in the product

Pause

Pauses are generally defined as inter-keystroke intervals (IKI) above a certain threshold (e.g., 2000 milliseconds). The IKI refers to the gap time between two consecutive key presses typically expressed in milliseconds. To illustrate, suppose a writer types a character "A" at time 1 and then a character "B" at time 2. One can obtain the IKI between the two characters simply using the formula: IKI = Time2 - Time1. Global measures of pausing are usually associated with the duration and frequency of pauses calculated from different dimensions. Below are some typical pause measures.

- number of pauses (in total or per minute)
- proportion of pause time (as a % of total writing time)
- pause length (usually the mean duration of all pauses in text production)
- pause lengths or frequencies within words, between words, between sentences, between paragraphs, etc.

Revision

Revisions are operations of deletions or insertions in writing. A deletion is defined as the removal of any stretch of characters from a text whereas an insertion refers to a sequence of activities to add characters to a growing text (except the end). Below are some commonly used revision measures:

- number of deletions (in total or per minute)
- number of insertions (in total or per minute)
- length of deletions (in characters)
- length of insertions (in characters)
- proportion of deletions (as a % of total writing time)
- proportion of insertions (as a % of total writing time)
- product vs. process ratio (The number of characters in the product divided by the number of characters produced during the writing process)
- number/length of revisions at the point of inscription (i.e., at the current end of a text being produced)
- number/length of revisions after the text has been transcribed (i.e., at a previous point in the text)
- number of immediate revisions (the distance between the position of the flashing cursor and the revision point equal to zero)
- number of distant revisions (the distance between the position of the flashing cursor and the revision point larger than zero)

Burst

Bursts refer to the periods in text production in which stretches of texts were continuously produced with no pauses and/or revisions. There are mainly two types of bursts: P-bursts that refer to the written segments terminated by pauses, and R-bursts that describe the segments terminated by an evaluation, revision or other grammatical discontinuity.

- number of P-bursts (in total or per minute)
- number of R-bursts (in total or per minute)
- proportion of P-bursts (as a % of total writing time)
- proportion of R-bursts (as a % of total writing time)

- length of P-bursts (in characters)
- length of R-bursts (in characters)

Process Variance

Process variance attends to the dynamics of the writing process in relation to time and thus represents how the writer's fluency may differ at different stages.

Process variance is generally measured by first dividing the whole writing process into a certain number of equal time intervals (e.g., 5 or 10) and then calculating the total number of characters produced in the intervals (often normalized to the average number of characters per minute), or to make it more comparable among writers, the proportion of characters produced per interval. The standard deviation of characters produced per interval is also calculated from keystroke logs as an indicator of process variance.

3.3 File and Field Information

The keyboard log data during the composition process contains timestamps, implying the presence of temporal order information, as it records the sequence and time intervals of keypress events.

Each essay's writing process can be viewed as a sequence with time-related details, such as the duration of key presses and intervals between consecutive keystrokes.

The competition dataset comprises about 5000 logs of user inputs, such as keystrokes and mouse clicks, taken during the composition of an essay. Each essay was scored on a scale of 0 to 6. Your goal is to predict the score an essay received from its log of user inputs.

The files and features involved in the data set are as follows:

- **train_logs.csv** - Input logs to be used as training data. 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.
- id - The unique ID of the essay
- 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 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 [x1, y1] To [x2, y2] - The event moves a section of text spanning character index x1, y1 to a new location x2, y2
- down_event - The name of the event when the key/mouse is pressed
- up_event - The name of the event when the key/mouse is released
- text_change - The text that changed as a result of the event (if any)
- cursor_position - The character index of the text cursor after the event
- word_count - The word count of the essay after the event

Note that there may be events in the test set that do not occur in the training set. Your solution should be robust to unseen events.

Note: Key_down and key_up events may not necessarily occur in the same order as they are presented in the dataset. To illustrate, a writer may press down "a" and then press down "b" before he/she even releases "a". However, all the keystroke information about "a" comes before "b" in the dataframe.

- **test_logs.csv** - Input logs to be used as test data. Contains the same fields as train_logs.csv. The logs available in the public version of this file are only examples to illustrate the format.
- **train_scores.csv**

- id - The unique ID of the essay
- score - The score the essay received out of 6 (the prediction target for the competition)

- **sample_submission.csv** - A submission file in the correct format. See the Evaluation page for details.

In this Code Competition, we can get a few example logs in test_logs.csv to help us author our solutions. When our submission is scored, this example test data will be replaced with the full test set. There are logs for about 2500 essays in the test set.

Partial details of the dataset files are as follows:

id	score
3 unique values	3 total values
0000aaaa	1.0
2222bbbb	2.0
4444cccc	3.0

Figure 4. Detail of Sample_submission.csv

id	event_id	down_time	up_time	action_time
3 unique values	6 total values	6 total values	6 total values	6 total values
0000aaaa	1	338433	338518	85
0000aaaa	2	760073	760160	87
2222bbbb	1	711956	712023	67
2222bbbb	2	298502	298548	46
4444cccc	1	635547	635641	94
4444cccc	2	184996	185052	56

Figure 5. Detail of test_logs.csv

id	event_id	down_time	up_time	action_time
2471 unique values	1 – 12.9k	106 – 8.31m	252 – 8.31m	0 – 447k
001519c8	1	4526	4557	31
001519c8	2	4558	4962	404
001519c8	3	106571	106571	0
001519c8	4	106686	106777	91
001519c8	5	107196	107323	127
001519c8	6	107296	107400	104
001519c8	7	107469	107596	127
001519c8	8	107659	107766	107
001519c8	9	107743	107852	109
001519c8	10	107848	107978	130
001519c8	11	108008	108195	187
001519c8	12	108104	108259	155
001519c8	13	108229	108370	141
001519c8	14	108341	108486	145
001519c8	15	109296	109438	142
001519c8	16	109423	109559	136

Figure 6. Detail of train_logs.csv

id	score
2471 unique values	0.5 – 6
001519c8	3.5
0022f953	3.5
0042269b	6.0
0059420b	2.0
0075873a	4.0

Figure 7. Detail of train_scores.csv

4 Design and Implementation

4.1 Model selection

The selection of appropriate models was a pivotal step in our endeavor to correlate typing patterns with essay quality. Our goal was to utilize models that could capture the complexity and subtleties inherent in the keystroke data while being computationally efficient and robust against overfitting.

Initially, we considered a variety of models, including simple linear regressions, decision trees, and support vector machines. However, these models did not fully leverage the rich temporal and sequential nature of the keystroke data.

We settled on three primary models, each chosen for its unique strengths in handling large datasets and capturing intricate relationships within the data:

1) Random Forest Regressor: Known for its effectiveness in regression tasks, the Random Forest model was selected for its ability to handle the high-dimensional space and for its robustness to noise. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.[1]

2) CatBoost Regressor: CatBoost, a gradient boosting model, was chosen for its state-of-the-art performance in handling categorical features and its speed in processing large datasets. Two critical

algorithmic advances introduced in CatBoost are the implementation of ordered boosting, a permutation-driven alternative to the classic algorithm, and an innovative algorithm for processing categorical features.[2]

3) XGBoost Regressor: Tree boosting is a highly effective and widely used machine learning method.[3] XGBoost was particularly appealing due to its regularization features, which help in reducing overfitting, and its scalability, making it suitable for the extensive keystroke logs.

Gradient boosting of regression trees produces competitive, highly robust, interpretable procedures for both regression and classification, especially appropriate for mining less than clean data.[4] Each model was rigorously validated using cross-validation techniques to ensure its predictive performance generalized well to unseen data.

Through an iterative process of testing and tuning, we converged on a final ensemble approach that synergized the strengths of each individual model, culminating in a robust predictive framework that reliably assessed essay quality from keystroke dynamics.

We further validate the reasonableness and effectiveness of the models by calculating the Mean and Std. for each sub-model. The results are as follows:

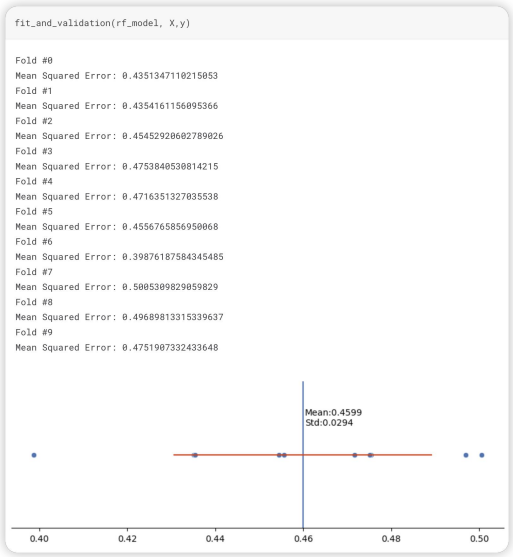

Figure 8. Performance of the Random Forest model

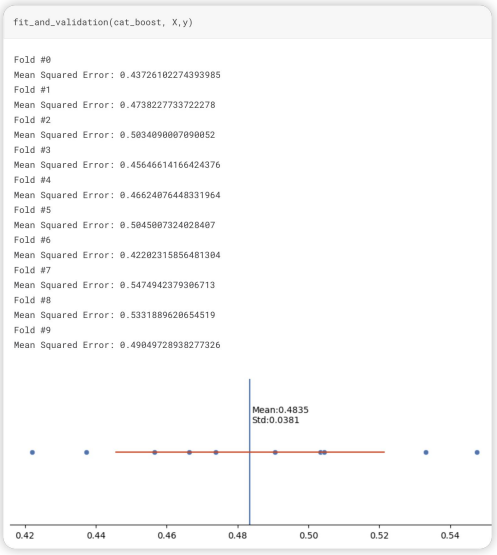

Figure 9. Performance of the Cat_boost

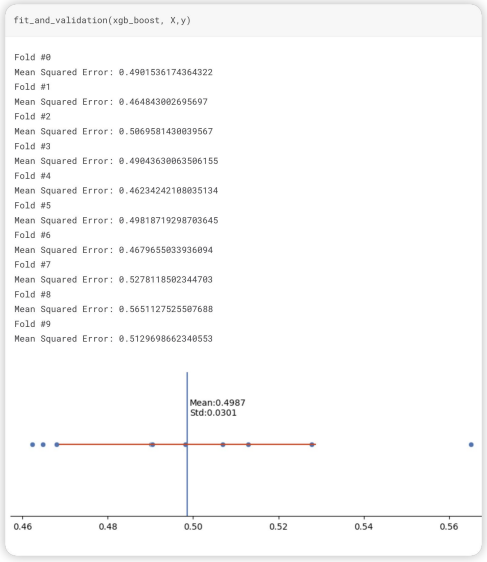

Figure 10. Performance of Xgb_boost

From the average MSE and standard deviation shown in the graph, it can be seen that the performance of all three sub-models is quite good.

We can observe that these models exhibit consistency and stability across different data splits, with minimal error fluctuations. Among them, the performance of the Random Forest model is the best.

4.2 Feature extraction

In the analytical journey to predict essay quality from keystroke data, feature extraction stands as a cornerstone. It is the process of

transforming raw data into a set of variables that can be effectively utilized by machine learning models. This phase is crucial, for the right features can significantly enhance model performance and interpretability.

Our approach to feature extraction was multifaceted, aiming to distill the essence of typing behavior into quantifiable metrics. We focused on both the temporal dynamics of typing and the text production patterns. The keystroke data provided rich information on the rhythm and cadence of writing, the frequency and types of revisions, and the overall writing speed.

By calculating the correlation between different features and the article scores, we obtained the following results:

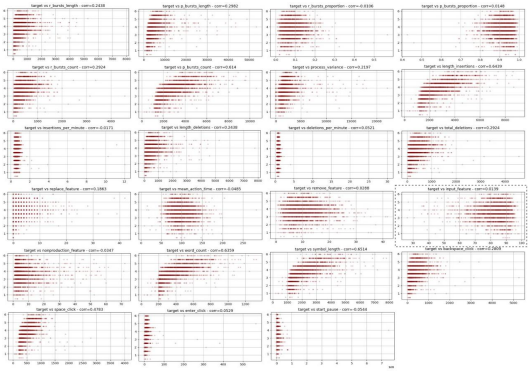

Figure 11. Correlation plots for selected features

The collection of scatter plots displayed in the image represents the correlation between different keystroke features extracted from essay writing data and the final essay scores. Each plot correlates a single feature, such as the count of specific keystrokes, the length of writing bursts, or the time-related aspects of the writing process, with the target variable, which is the essay score. A higher correlation coefficient (closer to 1 or -1) indicates a stronger linear relationship between the feature and the essay score, suggesting that the feature may be a good predictor of writing quality. Conversely, a correlation coefficient closer to 0 suggests little to no linear relationship. These plots are instrumental in understanding which features are most strongly associated with the quality of writing, and thus are valuable for refining predictive models in automated essay scoring systems.

By performing statistical analysis and summarization of these correlation data, we obtained more intuitive statistical results. Based on these results, we extract the features that have a significant impact on the article scores and discard unimportant features to reduce overfitting.

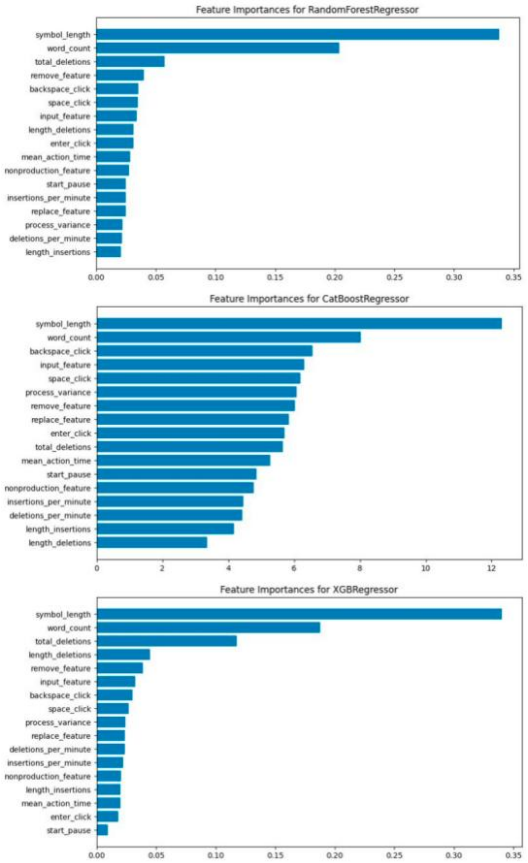

Figure 12. Correlation of data features

Based on the correlation, we have selected the following features:

- backspace_click
- text_length
- input_feature
- remove_feature
- mean_action_time
- replace_feature
- sentence_size_feature
- total_deletions
- r_bursts_count
- p_bursts_proportion

Based on these features, we build models for training and continuously add or remove features based on the training results.

4.3 Model stacking and training

Ensemble methods are considered the state-of-the art solution for many machine learning challenges. Such methods improve the predictive performance of a single model by training multiple models and combining their predictions. [5]

Stacking, as an ensemble learning technique, capitalizes on the diversity of multiple base classifiers to create a more robust and accurate predictive model.

In the context of this application, the three chosen base classifiers, namely rf_model, cat_boost, and xgb_boost, each contribute their unique strengths and perspectives to generate prediction results. These predictions are then cleverly combined by the meta-classifier lr, which acts as the "final decision-maker" in the ensemble. By aggregating the outputs of these diverse models, stacking has a track record of enhancing predictive performance, especially when tackling complex problems that may have non-linear relationships and intricate patterns in the data.

To implement this ensemble approach effectively, we employ a Linear Stacking Model. This stacking model seamlessly blends the predictions from the three sub-models, thus harnessing their collective intelligence to yield superior predictive accuracy and model performance. The synergy achieved through stacking often proves beneficial when aiming to extract the maximum predictive power from the underlying base models.

4.4 Evaluation

To accurately assess the performance of our predictive models, we employ the Root Mean Squared Error (RMSE) as our primary metric. This chapter outlines the importance of RMSE, its calculation, and its implications on model evaluation.

RMSE is a widely used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modeled. It serves as a way to quantify the variance in prediction errors, offering a clear standard for model performance.

The RMSE is mathematically represented as the square root of the average squared differences between the predicted and actual values. The formula is given by:

$$\text{RMSE} = \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right)^{1/2}$$

where $\hat{y}_i$ is the predicted value and y_i is the original value for each instance i over n total instances.

RMSE is particularly useful in scenarios where we need to compare the performance of various models or when fine-tuning the parameters of a single model. A lower RMSE value indicates better model performance, as it signifies smaller divergence from the actual values. It is sensitive to outliers, making it a rigorous measure that penalizes large errors more heavily.

In our study, RMSE provides a direct measure of model accuracy. By minimizing RMSE, we refine our models to better capture the underlying patterns within the keystroke data that correlate with the essay scores. The optimization processes involved in model training are iteratively adjusted, aiming to reduce the RMSE and thus enhance predictive accuracy.

By employing RMSE, our evaluation framework remains consistent and reliable. It allows us to effectively rank models, guide the model refinement process, and ensure the robustness of our predictive insights into the assessment of essay quality based on typing dynamics.

4.5 Result Analysis

The core objective of the analysis was to examine the predictive accuracy of the model developed for the assessment of essay quality. The results are encapsulated in a plot showcasing the relationship between the predicted scores by the model and the actual essay scores from the training data.

The plot features a scatterplot, with the actual essay scores on the x-axis and the predicted scores on the y-axis, juxtaposed with marginal histograms on both axes that illustrate the distribution of actual and predicted scores. Such a visualization aids in understanding the spread and concentration of predictions in comparison to true values.

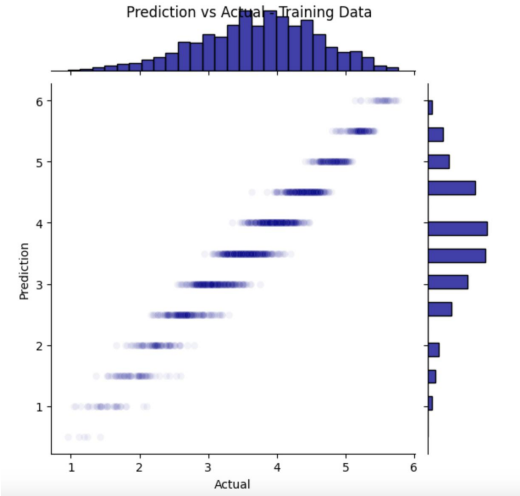

Figure 13. Prediction vs Actual Training Data

Upon close observation of the scatterplot, we can discern a pattern that suggests a linear relationship between the predicted and actual scores, albeit with some variance. This linear trend indicates that the model, to a certain extent, is capable of capturing the underlying assessment criteria that determine essay quality. However, the degree of scatter suggests room for model refinement.

The histograms provide additional insight into the frequency distribution of scores. They reveal the concentration of predictions around certain score ranges, which can help identify areas where the model's predictive power is strong or where it may falter.

In sum, the results demonstrate a promising but imperfect predictive capability of the model. The linear relationship between actual and predicted scores is evident, yet not absolute, indicating potential areas for further model improvement and adjustment. This analysis forms the basis for subsequent optimization efforts aimed at enhancing the model's precision and reliability in scoring essays.

5. Summary

The project aimed to establish a robust link between the processes involved in essay writing and the resulting quality of the essays. This was pursued by analyzing keystroke logs with advanced machine learning models to predict essay scores.

A comprehensive dataset containing approximately 5000 keystroke logs was the foundation for our analysis. These logs provided granular details about typing behavior, including pauses, revisions, and temporal patterns—all of which were hypothesized to influence essay quality.

We deployed a suite of machine learning models, including Random Forest, CatBoost, and XGBoost, each of which was meticulously trained and tuned on the training dataset. The models were evaluated based on their predictive performance, using metrics such as the Root Mean Squared Error (RMSE), and were further assessed for their efficiency and computational feasibility for practical educational applications.

A critical phase of the project involved feature selection, where we determined which aspects of the typing data were most indicative of essay quality. This process was informed by examining the correlation between features and essay scores, as well as by evaluating feature importances generated by the models.

The results depicted a generally linear relationship between predicted and actual scores, suggesting that the models captured key assessment criteria to a reasonable extent. However, the presence of variance in the predictions pointed to opportunities for model enhancement.

In conclusion, this study successfully demonstrated the potential of machine learning models to predict essay quality from typing data. The findings offer valuable insights that can contribute to the development of automated writing evaluation tools, ultimately aiming to improve instructional practices in writing education. The project also highlighted the importance of feature

selection and model tuning in the development of predictive models. Going forward, there is scope for refining the models further, particularly by integrating more nuanced linguistic and syntactical features to improve the accuracy of essay quality assessment.

6. Future Work

The current machine learning models offer a solid baseline for essay quality prediction. However, future work could involve integrating more sophisticated natural language processing techniques, such as sentiment analysis and syntactic complexity measures. Additionally, exploring model interpretability through techniques like SHAP values could provide deeper insights into feature contributions and model decisions.

To refine the predictive power of the models, additional data could be collected. This might include more diverse keystroke logs, capturing broader writing styles and cognitive processes. Incorporating demographic and educational background information could also enable a more nuanced understanding of writing performance.

An exciting avenue for future work is the development of a real-time analysis tool that could provide writers with instant feedback on their writing process and suggest improvements, thereby serving both educational and professional development purposes.

Conducting longitudinal studies to track the evolution of individual writing styles and their correlation with essay quality over time would be beneficial. This could help in understanding how writing processes mature and what interventions are most effective at different learning stages.

It is crucial for future work to consider ethical implications and potential biases in automated scoring systems. Ensuring that models do not perpetuate or amplify existing biases in writing assessment is essential for their equitable application.

In summary, future work will aim to build upon the current project's foundations, expanding its scope, improving model performance, and ensuring that the benefits of such technology can be ethically and effectively integrated into educational and professional domains.

References

[1] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://link.springer.com/content/pdf/10.1023/a:1010933404324.pdf

[2] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems, 31. https://papers.nips.cc/paper_files/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://dl.acm.org/doi/10.1145/2939672.2939785

[4] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 29(5), 1189-1232. https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full

[5] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1249. https://www.researchgate.net/publication/323424342_Ensemble_learning_A_survey