KEY TO QUALITY: DECODING THE LANGUAGE OF KEYSTROKES FOR PREDICTIVE INSIGHTS

Anuranjan Pandey, Dhruv Maheshwari, Kriti Jha, Nesar Nanjundaswamy & Pratyush Bhatnagar
Viterbi School of Engineering
University of Southern California
Los Angeles, CA 90089, USA
{akpandey, dhruvmah, kritijha, nnanjund, pb65357}@usc.edu

ABSTRACT

This paper presents a rigorous exploration of keystroke log data to predict overall writing quality by unraveling the intricate dynamics of the writing process. Writers utilize diverse techniques to plan, revise, and strategically allocate time throughout the writing process, with these nuanced actions influencing writing quality. In this paper, we explore a wide range of Machine learning models and their performances, ranging from the foundational Multinomial Logistic Regression to advanced models like LSTM and Transformers, navigating challenges such as model complexities and memory constraints. In our results, XGBoost Regression is the best-performing model with an RMSE of 0.6550. Beyond the competition, this work aspires to contribute to discussions on writing instruction, automated evaluation, and intelligent tutoring systems, envisioning a future where keystroke dynamics are pivotal in advancing automated writing assessment in educational technology.

1 INTRODUCTION

The evolving landscape of educational technology has prompted a shift in focus from the mere evaluation of final written products to a deeper understanding of the processes that govern writing. Traditional assessments, primarily fixated on the end product, often overlook the dynamic nature of the writing journey. This creates an opportunity for innovative methodologies that delve beyond surface-level evaluations. Our approach to writing assessment differs from conventional practices as we aim to explore the significance within each stage of the writing process.

This research stems from an immersive exploration into the domain of automated writing assessment, seeking to surpass traditional methodologies and delve into the nuanced behaviors inherent in the writing process. The research's significance lies in its holistic approach to predicting the writing assessment score just by using keystroke logs, without examining the actual essay or writing product, bridging the gap between the final product and the cognitive and behavioral processes involved in written work production.

To predict overall writing quality using keystroke logs as a rich source of information, we attempt to decode the language of keystrokes, revealing the interactions between learners' writing behaviors and their subsequent writing performance. The project [1] aims to provide a comprehensive understanding of writing quality assessment, nurturing autonomy, meta cognitive awareness, and self-regulation in writing.

The keystroke logs, obtained through a rigorous argumentative essay task, provide profound insights into participants' writing behaviors. Our feature engineering efforts focus on content and typing, editing, pauses, and cursor movement, creating 49 additional features. Encompassing aspects ranging from word length and typing speed to the frequency of distant revisions, these features transcend the traditional scope of writing assessment, promising a holistic representation of writing behaviors.

[1]The source code for the experiments is available at https://github.com/anuranjanpandey/Key-To-Quality/.

To achieve our objectives, our research employs diverse machine learning algorithms, showcasing a comprehensive approach to modeling. These algorithms include logistic regression, k-means clustering, naive Bayes, neural networks, and XGBoost, each contributing its unique strengths to the analytical framework. Our methodology is grounded in extensive feature engineering on keystroke data, which is the cornerstone for predictive modeling. By meticulously extracting insights from keystroke patterns, we not only capture the intricacies of writing behaviors but also lay the foundation for establishing meaningful connections between these behaviors and the overall quality of writing. This multi-faceted approach enriches the predictive capabilities of our models and, at the same time, sheds light on the dynamics that underlie effective written communication.

2 Related Work

A number of studies have explored using keystroke logs for various applications. Several of the most relevant are reviewed here.

Sinharay et al. (2019) analyzed the application of boosting, a data mining method, and Linear Regression to predict essay scores, while we propose a classification methodology with multiple models to provide nuanced insights complementing their work. Conijn et al. (2022) took a comparable approach, but we advance it by emphasizing time series data and neural networks like LSTMs and Transformers to capture evolving writing dynamics.

Malekian et al. (2019) characterized the writing processes in isolated writing phases, differing from our holistic perspective of spanning the entire writing process to comprehensively capture intricacies. Talebinamvar & Zarrabi (2022) aligned with our usage of PCA for feature extraction, and we build on this through additional analyses like heatmaps and SHAP for a thorough understanding. Hence, our approach further refines the methodology by incorporating advanced visualization techniques, thereby enriching the interpretability of the findings.

Allen et al. (2016) combined text indices and keystroke logs to assess how the behavioral aspects influence the engagement and boredom during the writing, distinct from our quality prediction but sharing an emphasis on feature engineering. We introduce 49 features illuminating writing behaviors' connections to performance. Conijn et al. (2019) studied task impacts on keystroke features, motivating our comprehensive feature exploration, including revisions, typing, bursts, pauses, D/I ratio, etc.

Edwards et al. (2020) indicated keystroke data's utility for predicting programming outcomes, highlighting contextual impacts. We construct this using a more extensive dataset for deeper insights into writing processes and potentially enhanced prediction robustness. Zhu et al. (2019) examined keystroke patterns across proficiency/gender, driving our project's goals to provide richer understandings from large-scale data.

Zhang et al. (2019) research used Ward's minimum-variance method to cluster students into four groups based solely on four writing performance indicators: time spent writing, number of words written, number of keystrokes, and number of edits. We augment this through additional features to get a more comprehensive and nuanced understanding of the writing process. While Banerjee et al. (2014) did not directly link their research to predicting essay scores, their investigation of keystroke patterns in identifying deception within digital writing suggests untapped potential. Our study aims to explore this potential for predicting various writing-related outcomes, including essay scores.

These studies provide valuable insights into the potential of keystroke data for analyzing writing processes and related tasks. Our work builds upon this existing knowledge, contributing novel insights into the relationships between writing behaviors and writing quality and paving the way for further advancements in predicting and supporting effective writing.

3 Problem formulation

The core challenge lies in nuanced understanding and predictive modeling of writing quality using keystroke log data. Despite the wealth of information in keystroke dynamics, intricacies of the writing process - including planning, revisions, and temporal strategies - remain underexplored in assessments scrutinizing final products. The main question is: Can keystroke behaviors, ranging

from pausing patterns to editing strategies, be systematically linked to and predictive of the holistic quality of a written essay?

The dataset for this research is taken from an active Kaggle Competition Alex Franklin (2023), comprising keystroke logs of participants in a 30-minute argumentative writing task. Participants were Amazon Mechanical Turk users writing based on retired SAT-style prompts, adding a standardized structure to the task. The dataset has 8,405,898 rows representing writing instances for 2,471 unique essays, providing diverse samples. Numerical columns, including downtime, uptime, action time, cursor position, and word count, give additional context. Categorical columns such as activity, up event, and text change further contribute to richness. The main challenge is keystrokes are masked with a character 'q', preventing essay reconstruction. The desired model output is a score from 0.5 to 6, higher being better quality. Scoring is discretized with step size 0.5.

Lastly, since we did not have a solid baseline, we created one by training various models on this dataset, starting from conventional machine learning models like Multinomial Logistic Regression until experimenting with Deep learning models such as LSTM and Transformers.

3.1 Feature Engineering

Gaining a nuanced perspective on the writing process requires examining how various dimensions shape both efficiency and quality of output. Careful analysis of typing dynamics, content creation, and metrics around word length, downtime, speed, etc., provides a window into the interplay between production and the final product.

Editing proves a pivotal phase, going beyond surface-level changes to refine and enhance textual clarity, flow, and overall caliber. Tracking how writers manipulate factors like word length, deletions/insertions, substantive revisions, and typing rate reveals problem-solving strategies and the evolution of the piece.

Patterns around pausing also supply telling indicators of cognitive load and thought progression, as depicted in the associated graph. Quantifying total pause time, quickness between keystrokes, location of pauses, and such grants a lens onto the cadence of writing and mechanics of pause-and-process behind composed work. They provide a clearer picture of how writers distribute their cognitive efforts and manage time while composing text.

Furthermore, analyzing cursor movement dynamics is crucial for grasping the spatial dimension of writing. Total movement and distance metrics offer vital comprehension of how writers interact spatially with text, elucidating their navigation and manipulation techniques as shown in Fig. 2.

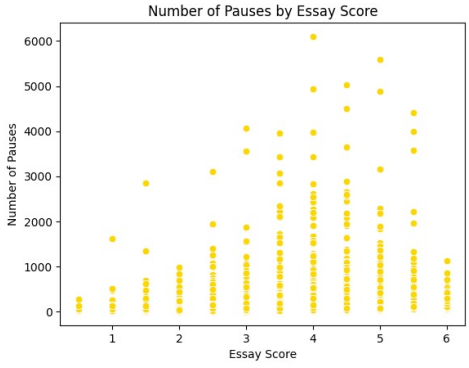

Figure 1: Number of Pauses vs Score

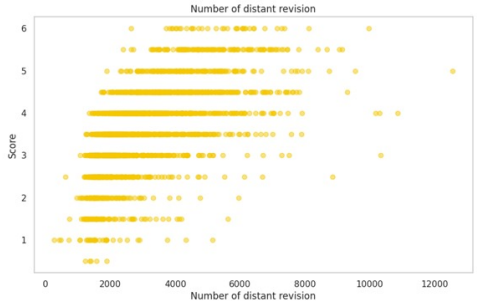

Figure 2: Number of distant revisions vs Score

4 Methodology

We decided to use classical machine learning models, both classification and regression models, on the feature-engineered dataset, and along with this, we also created a time series data that can be used to train our LSTM and Transformers model. The scores were label-encoded and converted to classes from 1-12. The dataset was split into 80:20::training:testing size. The following subsections

describe the various models trained and hyperparameters tuned to give the best possible RMSE. "Mean Squared Error" (MSE) was chosen as the loss function wherever applicable.

4.1 CLASSIC ML TECHNIQUES

We explored Multinomial and Ordinal Logistic Regression for classification. The Multinomial model utilized Adam and Cross-Entropy Loss, achieving a test RMSE of 0.684. This demonstrated effectiveness for multi-class tasks. The Ordinal model employed an ordinal loss function suited for ordered classes. Though the RMSE of 2.035 was higher, it showed promise for ordinal outcomes. By comparing precision and adaptability, we highlighted model selection factors. Further analysis can optimize each approach's strengths for given tasks.

We implemented the K-means model, incorporating PCA to reduce dimensionality and employing K++ for improved centroid initialization, facilitating faster convergence. Given the presence of 12 classes, the value of K was set as a constant at 12. The choice to utilize only one principal component aimed to strike a balance between dimension reduction and preserving variance in the data. However, the singular component captured a mere 46% of the variance. Attempts to enhance by increasing the number of components resulted in elevated RMSE.

We chose Multinomial Naive Bayes over Gaussian Naive Bayes to try Naive Bayes classification. Multinomial Naive Bayes has demonstrated effectiveness in capturing unique patterns of term frequencies, making it well-suited for our data having discrete data. Additionally, it is less sensitive to outliers compared to Gaussian Naive Bayes.Vijay & Verma (2023)

We also experimented with Complement Naive Bayes, which handles imbalanced datasets by adjusting feature likelihoods. However, our findings revealed that Multinomial Naive Bayes consistently outperformed it. This superiority can be attributed to its adept handling of text datasets' inherent sparsity and high dimensionality.

Taking a step further, we constructed an XGBoost model for both classification and regression tasks. This model incorporates regularization techniques to prevent overfitting, optimizes computational efficiency, and effectively handles missing data, rendering it a robust tool for addressing classification and regression problems. We experimented with varying values for key hyperparameters, specifically Max_depth (ranging from 4 to 16 with an increment of 1), N_estimators (ranging from 50 to 150 with an increment of 10), and Learning_rate (0.01, 0.41, 0.05, 0.11). Our exploration revealed that the optimal results were achieved with the following parameter values for XGBoost: For classification, Max_depth set to 4, learning rate set to 0.05, and N_estimators set to 90. For regression, the optimal values were Max_depth: 4, learning rate: 0.11, and N_estimators: 70.

We trained a Multi-Layer Perceptron (MLP) and employed Grid Search cross-validation to determine optimal hyperparameters. While we initially began with four hidden layers to align with our neural network design, we discovered that superior results were attained with only three hidden layers. Exploring hidden units in the range of 25 to 200, we identified that the configuration of (50, 50, 25) provided the most favorable outcomes when paired with the "logistic" activation function. The "Adam" optimization, featuring an adaptive learning rate and an initial learning rate set at 0.001, proved to be the most suitable choice.

We also implemented deep neural networks, seeking to leverage their power to uncover meaningful patterns within the dataset. Our methodology involved applying cross-validation techniques to find optimal settings for key hyperparameters - the number of hidden layers, choice of activation function, units per layer, dropout rate, etc. After iterative experimentation, a four-hidden-layer architecture yielded the strongest outcomes, with the successive layers containing 64, 32, 16, and 8 units accordingly, followed by a single output layer. Applying Rectified Linear Unit (ReLU) activations in the hidden layers and a linear output function proved the most effective for the neural network. We progressively reduced dropout rates from 0.5 at the input layer down to 0.1 near the output layer to further minimize overfitting and improve model performance. This iterative adjustment of architectural variables and hyperparameters helped achieve heightened accuracy and new discoveries around writing quality predictors for this domain. The step-by-step neural network refinements proved critical for fully leveraging deep learning's potential on this distinct dataset.

4.2 SEQUENTIAL MODELS

To fully capture the temporal dynamics within the data, we leveraged advanced models like Long Short-Term Memory (LSTM) networks and Transformers that specialize in sequential data. For categorical features, one-hot encoding transformed these into representations better suited for machine learning - converting aspects like the 'activity' column into binary vectors highlighting the most common categories: 'Nonproduction,' 'Input,' 'Remove/Cut,' and 'Replace.' We similarly one-hot encoded the 'up_event' column but only for very frequent symbols, those appearing over 100,000 times. Similarly, the 'text_change' column was one-hot encoded for symbols occurring more than 50,000 times. This approach ensured that our model focused on the most significant and frequent categories, enhancing the accuracy and efficiency of our analysis.

Given the variable number of keystrokes in essays (ranging from 700 to 12,000), we zero-padded each essay to ensure equal length for input to the LSTM and Transformer models. Zero rows were added to match the highest number of rows minus the number of rows in the current essay. Outliers with a number of keystrokes ≥ 7500 were removed.

Initially, we designed the LSTM with three hidden layers, each comprising 256 units and a batch size of 64. However, due to computational constraints, we later adjusted the architecture to 2 layers, each with 128 units and a batch size of 32.

On a similar note, due to the substantial input size and limitations in memory usage, the Transformer model with eight embedding dimensions, two encoder layers, two heads, and a batch size of 8 experienced significant challenges impacting its learning process.

5 RESULTS AND DISCUSSION

In our pursuit to predict overall writing quality through the exploration of keystroke log data, we conducted an extensive analysis employing various machine learning algorithms. The results presented in Table 1 encapsulate the performance of diverse models and shed light on the intricate dynamics of the writing process. The performance metrics, as measured by RMSE, unveiled the nuanced effectiveness of these models.

ALGORITHM	**RMSE**
Multinomial Logistic Regression	0.684
Ordinal Logistic Regression	2.035
K-Means++	2.028
Multinomial Naïve Bayes	0.719
Neural Network	0.782
MLP Classifier	0.706
XGBoost Regression	0.655
XGBoost Classification	0.728
LGBM Regression	0.679
LSTM	N/A
Transformer	N/A

Table 1: Results

A thorough examination of various machine learning algorithms for predicting writing quality provides interesting insights. Multinomial Logistic Regression emerges as a robust baseline with an RMSE of 0.684. This shows that the model can capture the intricate patterns of the writing process, resulting in accurate prediction. However, Ordinal Logistic Regression has a significantly higher RMSE of 2.035, indicating difficulties in comprehensively capturing the nuanced variations within the dataset.

We also explored various clustering techniques; k-means++ yields an RMSE of 2.028, showcasing limitations in its ability to discern patterns relevant due to the curse of dimensionality. On the probabilistic side, Multinomial Naïve Bayes performs reasonably well with an RMSE of 0.719, positioning it favorably among the models. Neural Network and MLP Classifier exhibit competitive performance, boasting RMSE values of 0.782 and 0.706, respectively, underscoring the efficacy of neural network architectures in capturing the complexities of the writing process.

XGBoost Regression and XGBoost Classification demonstrate strong predictive capabilities, achieving RMSE values of 0.655 and 0.728, respectively. This reaffirms the versatility of the XGBoost algorithm in effectively handling the intricacies of predicting writing quality. To understand which features had the most significant role in classification, we used SHAP analysis (Appendix-Fig 6). LightGBM Regression further bolsters this trend, achieving a low RMSE of 0.679, emphasizing the effectiveness of gradient-boosting frameworks.

However, challenges are encountered with LSTM and Transformer models, as denoted by "N/A", primarily due to issues related to large padding, sequence length, and large memory requirements. In LSTMs, we tried padding the sequences, but this resulted in a large number of zeros for an average-length essay; then we also tried truncating the essay till only 5k events as the mean length of the essay were 3.5k and the maximum event were 12k, but again the loss was not decreasing. In the case of Transformers, even small architecture with two encoder blocks was not trainable on a single P100 GPU provided on Kaggle. Therefore, more computation is required to complete the experimentation.

6 Conclusion and future work

Our research culminated in an extensive examination of the dataset, leading to the creation of 49 novel features. This exploration set the stage for training a broad spectrum of machine learning models, encompassing everything from the basic Naive Bayes to more complex structures like LSTM and Transformers. The process of developing our essay scoring model involved overcoming numerous challenges. Employing LSTM architectures posed difficulties related to padding sequences to equal lengths and managing sequence complexity, which hindered learning. Despite having considerable computing power - 30GB RAM and a P100 GPU - memory constraints limited the complexity of Transformer models we could implement. Additionally, class imbalance in the dataset led us to explore techniques like oversampling to improve prediction accuracy across various score levels.

Going forward, further research could substantially improve the system's capabilities. Conducting research into extra features and testing various model types may significantly increase predictive accuracy. As more advanced computing platforms become accessible, investigating larger and more intricate models, especially Transformers, seems promising. Applying Automated ML to automate model selection, hyperparameter tuning, and feature engineering could revolutionize development, greatly reducing time and resource requirements. This technology could enable building more sophisticated models previously infeasible. Exploring optimization strategies like model pruning, quantization, and distillation could offer ways to enhance computational efficiency without sacrificing performance. Lastly, the implementation of ensemble methods, which amalgamate predictions from multiple models, could be key to achieving greater generalization and robustness in our models.

Author Contributions

All five authors have made equal contributions, from feature engineering to experimentation. Collectively, we have created 49 new features, implemented various machine-learning models, and fine-tuned them. The collaborative effort extends to addressing challenges with LSTMs and Transformers.

Acknowledgments

We are grateful to Prof. Dani Yogatama for his invaluable guidance and support throughout this research project. Special thanks to the dedicated TAs, Ting-Rui Chiang and Joshua Robinson, for their assistance and insightful contributions.

References

Maggie Demkin Perpetual Baffour Ryan Holbrook Scott Crossley Alex Franklin, Jules King. Linking writing processes to writing quality, 2023. URL https://kaggle.com/competitions/linking-writing-processes-to-writing-quality.

L. K. Allen, S. Crossley, C. Mills, S. D'Mello, M. E. Jacovina, and D. S. McNamara. Investigating boredom and engagement during writing using multiple sources of information: The essay, the

writer, and keystrokes. In *Proceedings of the 6th International Conference on Learning Analytics amp; Knowledge (LAK'16)*, pp. 114–123, 2016. doi: https://doi.org/10.1145/2883851.2883939.

R. Banerjee, S. Feng, J. S. Kang, and Y. Choi. Keystroke patterns as prosody in digital writings: A case study with deceptive reviews and essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1469–1473, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1155.

R. Conijn, J. Roeser, and M. van Zaanen. Understanding the keystroke log: The effect of writing task on keystroke features. *Reading and Writing*, 32:2353–2374, 2019. doi: 10.1007/s11145-019-09953-8.

R. Conijn, C. Cook, M. van Zaanen, and L. Van Waes. Early prediction of writing quality using keystroke logging. *International Journal of Artificial Intelligence in Education*, 32:835–866, 2022. doi: 10.1007/s40593-021-00268-w.

J. Edwards, J. Leinonen, and A. Hellas. A study of keystroke data in two contexts: Written language and programming language influence predictability of learning outcomes. In *The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*, Portland, OR, USA, March 11-14 2020. ACM. doi: 10.1145/3328778.3366863a.

D. Malekian, J. Bailey, G. Kennedy, P. de Barba, and S. Nawaz. Characterising students' writing processes using temporal keystroke analysis. *In Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, pp. 354–359, 2019.

S. Sinharay, M. Zhang, and P. Deane. Prediction of essay scores from writing process and product features using data mining methods. *Applied Measurement in Education*, 32(2):116–137, 2019.

M. Talebinamvar and F. Zarrabi. Clustering students' writing behaviors using keystroke logging: a learning analytic approach in efl writing. *Language Testing in Asia*, 12(6), 2022. doi: https://doi.org/10.1186/s40468-021-00150-5.

V. Vijay and P. Verma. Variants of naïve bayes algorithm for hate speech detection in text documents. In *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pp. 18–21, 2023. doi: 10.1109/AISC56616.2023.10085511.

M. Zhang, M. Zhu, P. Deane, and H. Guo. Identifying and comparing writing process patterns using keystroke logs. In *Educational Testing Service*, Princeton, NJ 08540, USA, 2019. doi: 10.1007/978-3-030-01310-3_32.

M. Zhu, M. Zhang, and P. Deane. Analysis of keystroke sequences in writing logs. Technical Report RR-19-11, Educational Testing Service, Princeton, NJ, 2019.

A Appendix

The below table provides a comprehensive list of features used in the analysis, along with their descriptions. It serves as a reference for understanding the various metrics calculated from the essay writing data and their respective roles in the predictive models.

Features	**Description**
Mean word length	The average length of words in an essay are calculated as the total number of characters divided by the total number of words.
Keystroke Rate	The rate at which keystrokes are made, computed as the total number of keystrokes divided by the total time taken to write the essay.
Number of keystrokes	The total count of keystrokes registered during the writing of an essay, including letters, punctuation, and functional keys.
Typing time	The aggregate amount of time spent typing, excluding any pauses or interruptions.

Number of characters	The total number of characters typed in an essay after accounting for insertions and deletions.
Non-production time	The total time spent on activities not directly related to text production, such as reviewing or reading the essay.
Delete/Insert Ratio	The ratio of deletion actions (like backspaces and cuts) to insertion actions (such as typing and pasting).
Number of revised insertions	The count of instances where text inserted into the essay was subsequently modified or corrected.
Discarded text	The total amount of text removed from the essay, including backspaces, cuts, and replacements.
Proportion of revised insertions	The proportion of insertions that were later revised, relative to the total number of insertions.
Number of major edits	The count of significant editing actions where two or more words were deleted in a single action.
Cursor move count	The number of instances where the cursor was repositioned during the essay writing process.
Major edit deletion time	The cumulative duration spent on making major edits, indicating the time invested in substantial revisions.
Cursor move distance	The cumulative distance the cursor was moved during editing, indicating the extent of navigation within the text.
Number of minor edits	The total number of instances where single words were deleted during the editing process.
Pauses in sentences	The count of pauses that occur within sentences, potentially indicating moments of reflection or revision.
Number of Cut/Copy/Paste	The combined total of cut, copy, and paste commands used, reflecting the degree of text manipulation.
Total pause time	The sum of the durations of all pauses taken during the essay writing process.
Number of backspaces	The count of backspace keypresses, reflecting the frequency of deletion during text production.
Pauses before sentence	The number of pauses taken before beginning a new sentence, which may indicate planning or organizing thoughts.
Duration of backspaces	The total time spent pressing the backspace key are indicative of the time consumed in correcting or revising text.
Pauses before word	The number of pauses that occur right before typing a new word, possibly reflecting word choice deliberation.
Proportion of deletions	The proportion of time spent deleting text compared to the overall time spent writing the essay.
Pause variance	The statistical variance of pause durations, giving a measure of how varied the pauses were in length.
Number of distant revisions	The count of revisions made at a distance from the current cursor position, reflecting more significant editing.
Long pause percentage	The percentage of pauses classified as long in duration out of the total number of pauses.
Number of immediate revisions	The count of revisions made at the cursor's position, reflecting on-the-spot editing.
Inter key interval	The average time interval between consecutive keystrokes, indicative of typing fluency and rhythm.
Number of bursts	The count of continuous typing sequences, separated by pauses shorter than 2 seconds.
Mean of burst length	The average length of continuous typing sequences, or bursts, measured in the number of characters.
Pauses within word	The number of interruptions that occur mid-word during typing, which may indicate hesitation or error correction.
Standard deviation of burst length	The standard deviation of the lengths of typing bursts, indicating the consistency of typing flow.

Mean of burst duration	The average time span of continuous typing bursts, reflecting the sustained attention during writing.
Standard deviation of burst duration	The standard deviation of the durations of typing bursts, showing the variability in sustained typing focus.

Table 2: Feature Descriptions

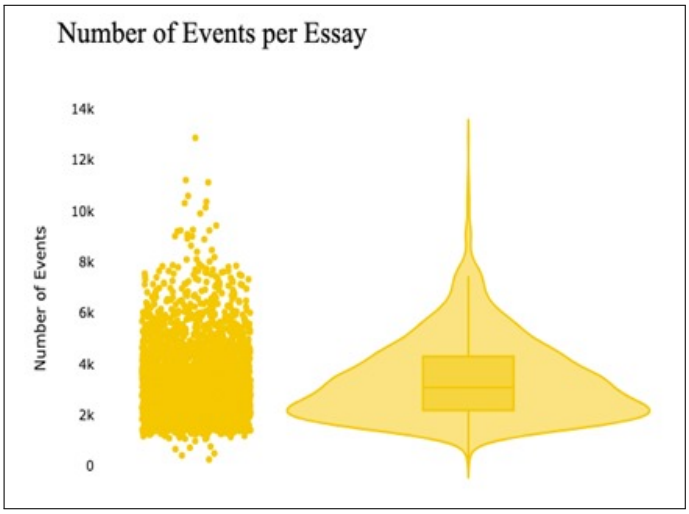

Figure 3: Number of events per essay

Fig 3 illustrates the distribution of writing events for each essay, depicting a wide range in the number of interactions (such as keystrokes, edits, and cursor movements) that occur during the essay writing process. The violin plot alongside the scatter provides a density estimation of the events, offering insight into the commonality of event frequencies per essay.

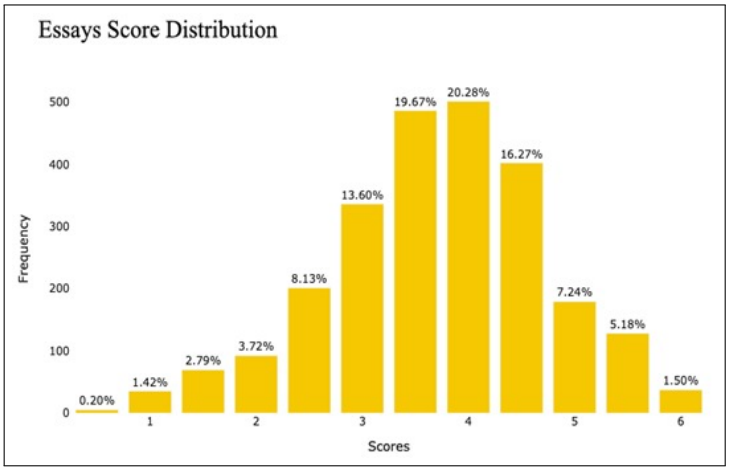

Figure 4: Essay Score Distribution

The bar chart in Fig 4 represents the frequency distribution of scores across all essays evaluated. It highlights the prevalence of each score category, allowing for the observation of grading trends and the distribution of writing proficiency within the dataset.

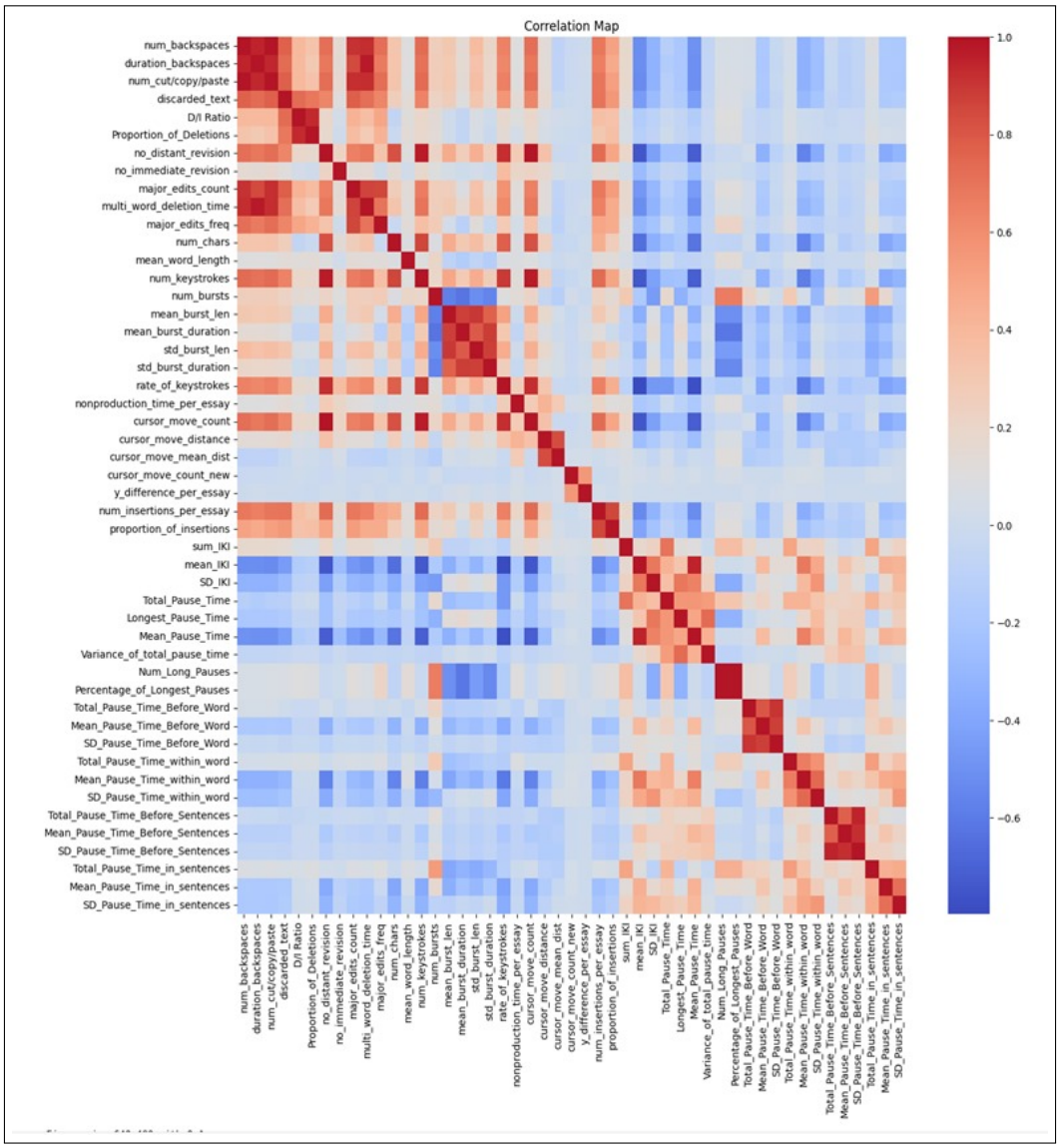

Figure 5: Correlation between features

In Fig 5, a heatmap is showcasing the correlations between different features extracted from the essay writing process. Darker shades represent stronger relationships, with red indicating positive and blue representing negative correlations. This aids in visually identifying features that might influence one another, thereby informing feature selection for modeling purposes.

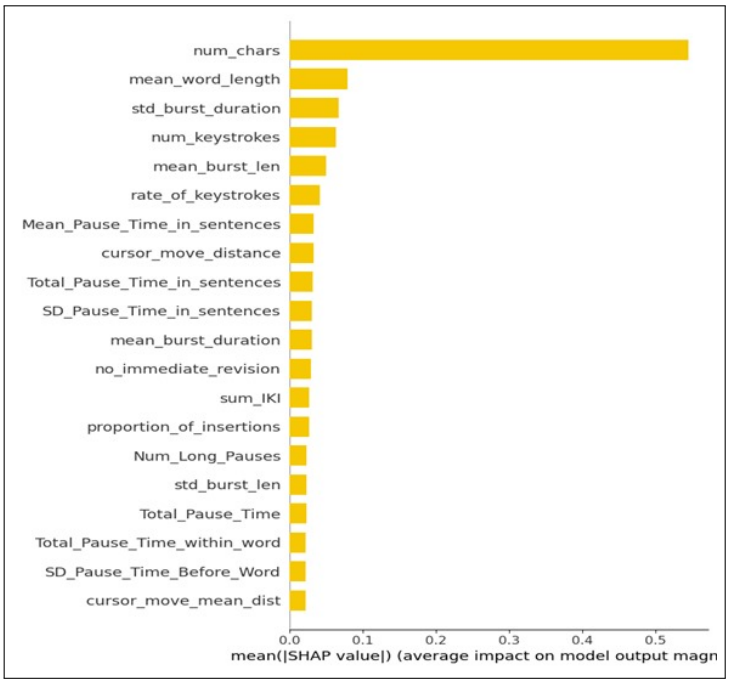

Figure 6: Feature importance using SHAPE

The bar chart in Fig 6 quantifies the importance of various features in our best-performing model, i.e., the XGBoost model using SHAP values. Longer bars denote greater impact, offering a clear visual hierarchy of features that most influence model output and, ultimately essay scores.