# KEYSTROKE DYNAMICS AND ESSAY QUALITY

**Jonah Kim**
School of Engineering and Applied Science
University of Virginia
Charlottesville, VA 22904
crd3vm@virginia.edu

**Connor McCaffrey**
College of Arts & Sciences
University of Virginia
Charlottesville, VA 22904
cam7qp@virginia.edu

December 10, 2023

**ABSTRACT**

Students nation-wide struggle with English composition skills. While educational institutes provide writing centers and other resources, educators primarily focus on the essay itself rather than the writing process. However, through keystroke logs, an opportunity arises to analyze not only *what* a student writes but also *how* they write it. This could reveal valuable insights into an essay's quality not captured by present methods. Previous studies conducted in this field typically have employed small datasets and few features [1]. Due to the importance of data quantity and quality in statistical analysis, this could have negatively affected the studies' findings. This project aims to address this by using a recently published Kaggle dataset which contains eight times as many essays than previous studies [2, 3]. Experimenting with a variety of machine-learning methods, we hope to develop a regression model which predicts essay scores from keystroke dynamics.

***K*eywords** Applied Regression, Keystroke Dynamics, Essay Quality

## 1 Introduction:

**Motivation:** According to the National Assessment of Educational Progress (NAEP), approximately 20% of 12th-grade students face challenges in mastering basic English composition skills [4]. This can significantly hinder a student's future academic and career goals. While schools and universities offer writing assistance, these services focus on the final, written product rather than the writing process itself. The ability to assess not only *what* students write but also *how* they write could reveal insights into their composition's quality not captured by traditional means. Both educators and students could benefit from the development of an analysis tool. Teachers could use it to identify students who require additional support, and students could use it to evaluate themselves for self-study.

**Goal:** This project's primary objective is to explore the connections between keystroke dynamics and essay quality through the application of machine learning techniques. Our goal is to create a regression model capable of predicting the score essays based on its keystroke profile. Through this project, a future tool could be developed which records a student's keystrokes and provides predictions of the final essay's writing quality.

**Dataset:** For this project, we will be using a recently released dataset available through the "Linking Writing Processes to Writing Quality" competition on Kaggle. Collected by Vanderbilt University, the dataset contains over two thousand essay entries and over eight million key events in total. The collectors replaced the identity of alphanumerical key-events with the letter 'q' to ensure that any model developed used only the writing process itself (and not the essay's content) to make predictions. The participants of the study were hired from Amazon Mechanical Turk to write 30-minute argumentative essays based on past SAT prompts. Because of the prompts parallel those found on the SAT, an accurate model could provide a valuable resource for the millions of students preparing for examinations and seeking college admission each year.

## 2 Related Works:

**Datasets:** Previous studies related to our project suffer from small datasets and few features [1]. For example, studies published in the International Journal of Artificial Intelligence in Education and the Annual Meeting of the Cognitive Science Society used fewer than 300 participants [3, 2]. In contrast, our project's dataset contains eight times more essays than these studies. Due to importance of data quantity and quality to avoid bias, the small datasets used in the studies may have negatively affected their results. We hope that the size of our dataset will reduce the potential impact of collection variance.

**Previous Models:** The two studies listed above employed random-forest, radial-kernel SVM, naive Bayes, and hierarchical multiple regression models [3, 2]. Since these studies dealt with small to medium sized datasets, more complex models (i.e., neural networks) were unnecessary; however, our dataset may be sufficiently large to allow for their use and to develop more sophisticated and flexible models.

## 3 Methods:

**Data Processing:** Our dataset consists of multiple temporal sequences representing the keystrokes of an individual essay. Consequently, each sample holds a 2D matrix of values. While certain Keras layers accept sequential data (such as an LSTM), sequential data is incompatible with Sklearn's interface. Therefore, we could not train traditional models using the data in its original dimensionality. We solved this by extracting over 50 aggregate features through an R script. The features were engineered following the descriptions found in previous research [3]. This allowed us not only to train sci-kit-learn models but also to reduce the size and the dimensionality of our input.

The extracted features encapsulate a diverse range of behaviors derived from the keystroke sequences:

- *Pauses*: The majority of features center around the timings of pauses, with specific attention to the Inter-Keystroke Interval (IKI) — the duration between consecutive key presses. Statistics such as the mean, median, standard deviation (SD), and maximum IKI provide insights into the distribution of this metric throughout the essay. Distinct features were crafted to distinguish mean and SD IKI values within words, between words, within sentences, and between sentences. Additionally, counts of the number of IKI values falling within specific length intervals were computed. Furthermore, the percentage of long pauses between words, calculated as pauses exceeding two standard deviations from the mean IKI, offers valuable insights into the distribution of pauses relative to one's own typical typing speed.

- *Revisions*: We counted the number of deletions and backspaces at the leading edge of the text and within the text body. Calculations included the mean and SD time spent in single and multiple backspace sequences. These serve as indicators of the writer's tendency to review and revise, shedding light on the iterative process of rereading and refining their written work.

- *Bursts*: Defined as uninterrupted sequences of text production, bursts are another important typing behavior to consider. Statistics such as the mean, SD, and maximum characters per burst, along with the total number of bursts, paint a vivid picture of typing dynamics. We took care to differentiate between revision, insertion, and production bursts to account for rapid additions and removals of text.

- *Verbosity Over Time*: To address how one's typing behavior changes as time elapses, we calculated the total keystrokes, characters, and words typed over 30-second intervals along with measures of the mean, SD, entropy, uniformity, and slope of those intervals.

This feature engineering process laid the groundwork for our subsequent predictive modeling efforts. Since we wanted to experiment with both sequential and traditional models, we processed both the temporal sequences and the aggregate features inside our data pipeline as shown in Figure 1. To ensure that the original distribution of essay scores was kept, we also applied a stratified split across the dataset before transforming the data.

**Models:** Due to the limited research surrounding this problem, we wanted to test a variety of models to determine which performed well. This also enabled us to see which models may be effectively combined using ensemble learning. To establish a baseline for comparison, we trained and tuned the same models used in the studies if they had regression implementations in Sklearn [3]. The models we found and employed for our baselines are the Support Vector Regressor and the Random Forest Regressor. We then compared a variety of Sklearn models and TensorFlow neural networks against them.

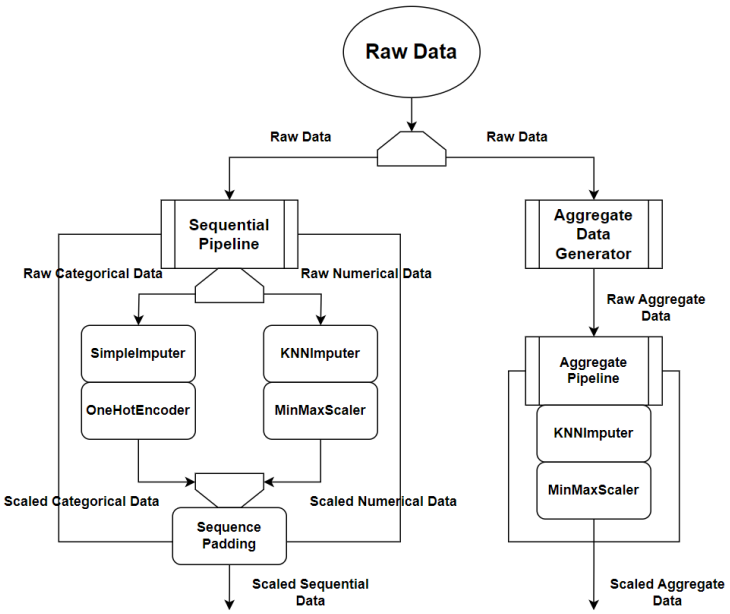


Figure 1: Project Pipeline

## 4 Experiments:

We scored our models using the Root Mean Squared Error (RMSE) metric against the predicted essay score and the actual value (ranges from 0 to 6 with 0.5 increments). Table 1 summarizes the initial trained models which performed with an error rate of less than 1 when evaluated using 5-fold cross validation. The Mean RMSE, STD, and median are calculated based on the five validation scores. Anything notated with (*nt*) was not tuned.

| MODEL NAME | TUNING | RMSE | STD | MEDIAN |
|---|---|---|---|---|
| SVR | **Tuned Baseline** | 0.6571 | 0.0224 | 0.6522 |
| GradientBoostingRegressor | *nt* | 0.6574 | 0.0218 | 0.6608 |
| StackingRegressor | *nt* | 0.6585 | 0.0270 | 0.6543 |
| BaggingRegressor(LGBMRegressor) | *nt* | 0.6587 | 0.0266 | 0.6616 |
| VotingRegressor | *nt* | 0.6615 | 0.0263 | 0.6586 |
| RandomForestRegressor | **Tuned Baseline** | 0.6616 | 0.0247 | 0.6584 |
| BaggingRegressor(ExtraTreesRegressor) | *nt* | 0.6633 | 0.0241 | 0.6620 |
| BaggingRegressor(XGBRegressor) | *nt* | 0.6637 | 0.0248 | 0.6520 |
| LinearRegression | *nt* | 0.6645 | 0.0263 | 0.6561 |
| ExtraTreesRegressor | *nt* | 0.6655 | 0.0265 | 0.6602 |
| BayesianRidge | *nt* | 0.6661 | 0.0270 | 0.6625 |
| Ridge | *nt* | 0.6666 | 0.0260 | 0.6654 |
| MLPRegressor | *nt* | 0.6691 | 0.0236 | 0.6632 |
| HistGradientBoostingRegressor | *nt* | 0.6759 | 0.0270 | 0.6687 |
| LGBMRegressor | *nt* | 0.6767 | 0.0281 | 0.6827 |
| XGBRegressor | *nt* | 0.6985 | 0.0243 | 0.6978 |
| SGDRegressor | *nt* | 0.7077 | 0.0274 | 0.7091 |
| KNeighborsRegressor() | *nt* | 0.7648 | 0.0280 | 0.7575 |
| PassiveAggressiveRegressor | *nt* | 0.8573 | 0.1728 | 0.7416 |
| DecisionTreeRegressor | *nt* | 0.9484 | 0.0387 | 0.9560 |
| ExtraTreeRegressor | *nt* | 0.9665 | 0.0292 | 0.9663 |

Table 1: Summary of Model Performance

Observing the table, we can see that the baseline models perform well, scoring at the top. The SVR model even out-performed the ensemble ones. However, some non-baseline models also performed well, such as the

GradientBoostingRegressor and even LinearRegression. Since most of the highest scoring models performed similarly, we decided to combine them using ensemble learning to see if each individual model could capture separate information and further reduce the error when combined. Using a StackingRegressor, we created a model composed of ExtraTrees, RandomForest, LGBM, XGB, LinearRegression, Ridge, BayesianRidge, SVR, and GradientBoosting. The model employed a Ridge meta regressor to combine the various models' outputs. This reduced the mean error and median to 0.6397 and 0.6377 respectively while slightly increasing the standard deviation to 0.0241. Overall, the ensemble stacking model outperformed all other regressors.

We also trained multiple neural networks with custom architectures using TensorFlow's functional API. We experimented with LSTM NNs, Convolutional NNs, Temporal Convolutional NNs, and Transformers. The best architecture we developed utilized a wide network consisting of a Temporal Convolutional neural network which processed our sequential data and a Dense neural network which handled our aggregate features. Both the Temporal Convolutional NN and the Dense NN employed skip connections to preserve the flow of gradients. The final layers of each network were concatenated together and fed into Dense layers which outputted the predictions using a scaled Sigmoid activation function. Although the network performed well ($\approx 0.68$ RMSE), it did not outperform the ensemble SciKit-Learn model.

TensorFlow neural networks proved difficult to train and evaluate. The networks which employed our sequential data (the primary reason we were experimenting with NNs) took many times longer to converge than SciKit-Learn's models since the Keras layers were operating on the full keystroke sequences rather than the summarized features. Additionally, because TensorFlow does not natively support 5-fold-cross-validation, we could not directly compare our neural networks to the models developed in SciKit-Learn. Its feasible that a neural network architecture exists which could outperform our ensemble model; however, since a simple model like LinearRegression scored well in our experiments, it seems likely that the employment of neural networks over-complicates the problem.

## 5 Results:

Our experiments show that clear correlations exist between a person's writing process and their final essay. Mapping our best model's RMSE onto a 0-100 point grade scale, the stacking model we developed can predict an essay's score within a $10.67 \pm 0.40$ (1 STD) point margin on average. Although this performance gap is relatively large, its not necessarily indicative of a flaw within the model itself. We all create and evaluate writing according to our own tastes and preferences. This creates noise within the data since the assigned training label might not be representative of its corresponding essay's quality. Additionally, as mentioned in the introduction, the dataset used in this project obscured the value of alphanumerical inputs, replacing all with the letter 'q'. This means that the model trained with only obscured keystroke logs and did not have access to the final text of an essay (i.e., what the graders were evaluating). If the original text was reintroduced, the error may decrease further with the correct model architecture.

## 6 Conclusion:

Using the results of our experiments, we can construct a predictive scoring tool which may benefit both Virginia educators and students. The final model developed uses only aggregate features from the keystroke logs. This means that it does not rely on a minimum or maximum number of keystrokes to make a prediction. By continually updating aggregate features based upon the latest keystroke pressed, the trained model could provide real-time essay score predictions. Alternatively, the logs could be saved and processed all at once for a more stable evaluation. Teachers could use this tool to quickly identify students in need of assistance while students could use this as an early grade predictor to estimate the quality of their essay.

It should be reiterated that our models use obscured keystrokes and operate based upon correlations not causation. For example, the feature within the dataset with the highest correlation to the essay score (correlation of 0.329) was the word count. The current data collection format makes an essay comprised of nonsense words and jumbles of letters appear identical to one with correct grammar (so long as the keystrokes profiles match). Consequently, this system should not be relied upon to provide final scores but as an intermediary tool in the writing process.

As a project bounded by a semester deadline, there are many places for improvement and additional experimentation. The aggregate features employed here can be further refined and tuned to eliminate false patterns and to generate new features. Additionally, future work may focus on developing language models which take into account the actual identity of each key press. This could allow for more complex evaluations based upon grammar, styling, and voice. If

pursued, careful work should be done to test the model on adversarial examples to ensure the model correctly evaluates noisy input (random keystrokes, key identity replacement, etc.).

## 7 Contributions:

- **Connor McCaffrey**: Wrote the aggregate feature extraction code in R (500+ lines of code), recorded/edited/rendered keystroke demonstration for video (1 min.).
- **Jonah Kim**: Created project library, applied visualization techniques, developed data pipelines, trained/developed/evaluated TensorFlow and SciKit-Learn models (≈750 lines of refactored code), made project presentation script, recorded (5 min.)/edited/rendered/uploaded project video.

## References

[1] Alex Franklin, Jules King, Maggie Demkin, Baffour Perpetual, Ryan Holbrook, and Scott Crossley. Linking Writing Processes to Writing Quality, 2023.

[2] Aaron Likens, Allen Laura, and Danielle McNamara. Keystroke Dynamics Predict Essay Quality. *Annual Meeting of the Cognitive Science Society*, July 2017.

[3] Rianne Conijn, Christine Cook, Menno Van Zaanen, and Luuk Van Waes. Early prediction of writing quality using keystroke logging. *International Journal of Artificial Intelligence in Education*, 32(4):835–866, December 2022.

[4] National Assessment of Educational Progress. Writing 2011: National Assessment of Educational Progress at Grades 8 and 12, September 2012.