# Linking Writing Behavior with writing quality


*Group Members*
*Neeraj Gupta*
*Ayush Manojkumar Lodha*
*Dhiraj Pimparkar*


*Class Project: CSCI 57300*
*Under Guidance of*
***Prof. Dr. Mohammad Al Hasan***

# Problem Statement

- **We are solving the ongoing challenge on Kaggle hosted by the "The Learning Agency Lab".**
- **Competition Link:** *https://www.kaggle.com/competitions/linking-writing-processes-to-writing-quality/overview*
- **Objective**: To assess how typing behavior impacts the quality of written essays.
- **Method**: To build predictive models using a keystroke log dataset that captures various aspects of the writing process.
- **Challenge**: To comprehend the influence of diverse writing behaviors, including planning, revisions, and pause patterns, on the quality of written content.

# Dataset Description

| 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 |

*An Example Dataframe of Keystroke Logging Information*

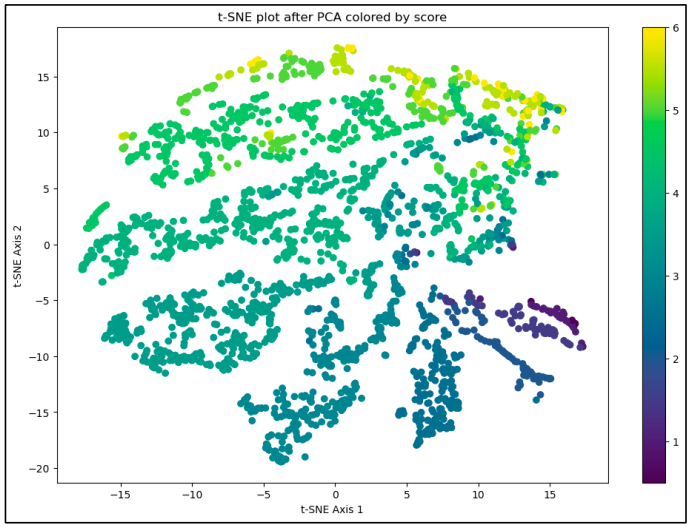
- Initial Data Contains **8,405,898** data instances and **11** variables.
- Dataset consists **2471** essays each having it's information in form of event ID as shown in the dataframe in the left.
- **Independent variables:**
  - **Categorical:** id, event_id, down_event, up_event, activity, text_change (6)
  - **Numerical:** down_time, up_time, action_time, cursor_position, word_count (5)
- **Dependent variables:** score
- Here the specific characters are hidden in form "q" instead of the real characters, which makes it impossible to recreate the essay and truly judge the essay on key log strokes information.

# Feature Description

| Feature Name | Feature Type (Categorical/ Numerical) | Description |
|---|---|---|
| **'id'** | Categorical | The unique ID of essay |
| **'event_id'** | Categorical | The index of event for user, order chronologically |
| **'down_event'** | Categorical | The name of the event when the key/mouse is pressed |
| **'up_event'** | Categorical | The name of the event when the key/mouse is released |
| **'down_time'** | Numerical | The timestamps of the down event in milliseconds |
| **'up_time'** | Numerical | The timestamps of the up event in milliseconds |
| **'action_time'** | Numerical | The duration of event<br>(difference between 'down_time' and 'up_time') |
| **'activity'** | Categorical | The category of activity which the event belongs to<br>o Nonproduction - The event does not alter the text in any way<br>o Input - The event adds text to the essay<br>o Remove/Cut - The event removes text from the essay<br>o Paste - The event changes the text through a paste input<br>o Replace - The event replaces a section of text with another string<br>o 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 |
| **'text_change'** | Categorical | The text that changed as a result of the event (if any) |
| **'cursor_position'** | Numerical | The character index of the text cursor after the event |
| **'word_count'** | Numerical | The word count of the essay after the event |
| **'score'** | Numerical – If Model is Regression<br>Categorical – If Model is Classification | Score of the essay from 0 to 6 |

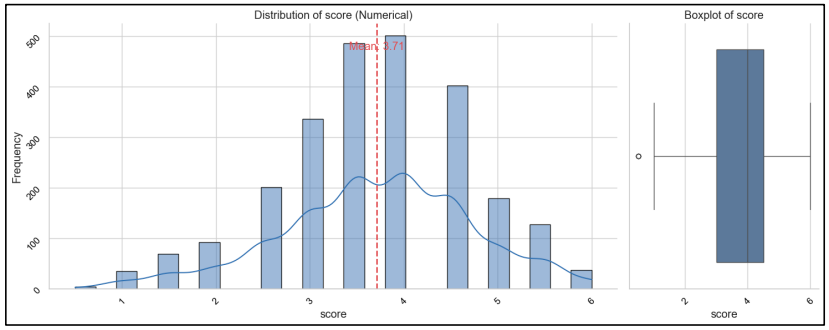
# Feature Engineering (Proposal)

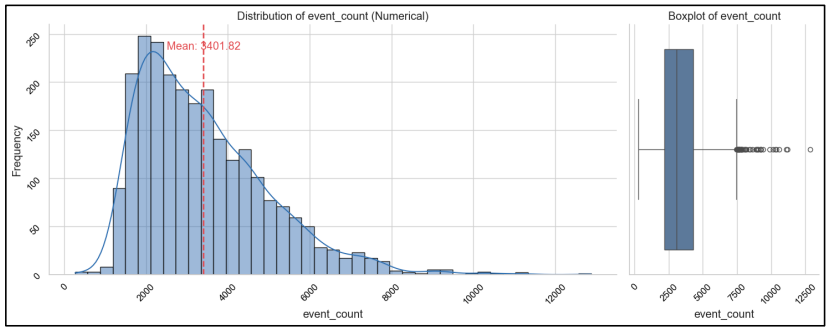
| Feature Name | Feature Type (Categorical/ Numerical) | Description |
|---|---|---|
| **'down_time_sum'** | Numerical | The sum of all 'down_time' values for each 'id'. Represents the total time of down events in milliseconds. |
| **'up_time_sum'** | Numerical | The sum of all 'up_time' values for each 'id'. Represents the total time of up events in milliseconds. |
| **'action_time_sum'** | Numerical | The sum of all 'action_time' values for each 'id'. Represents the total duration of all events for an 'id'. |
| **'cursor_position_max'** | Numerical | The maximum 'cursor_position' value for each 'id'. Indicates the farthest cursor position reached in the text by the user. |
| **'word_count_max'** | Numerical | The maximum 'word_count' value for each 'id'. Represents the highest word count reached in the essay at any point. |

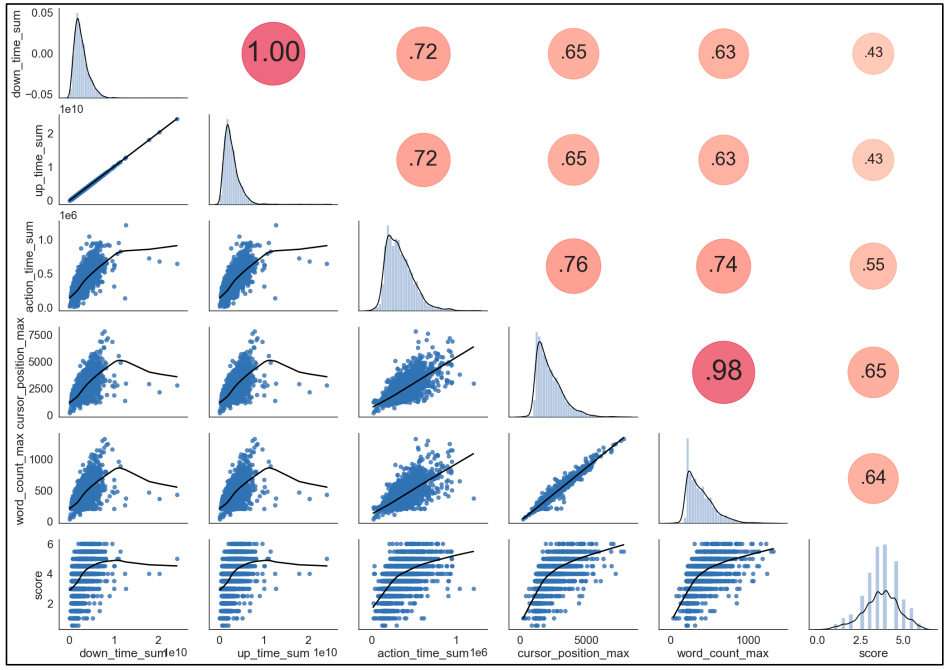



- We can see cluster variation by the score in the above plot; there is some separation between the clusters by scores, but not distinctly because of the huge variation in the data.

# Exploratory Data Analysis (EDA) - I



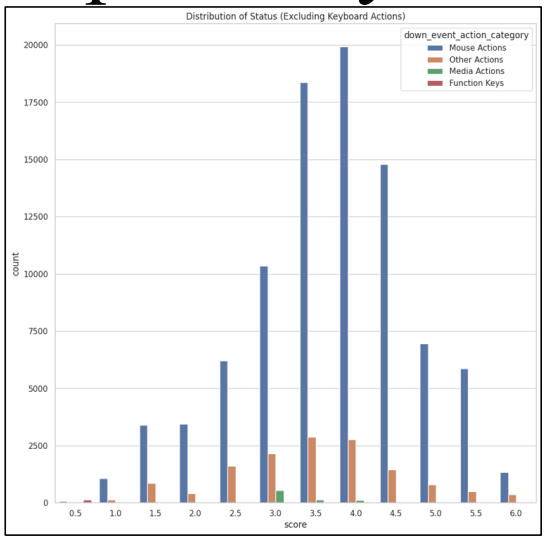



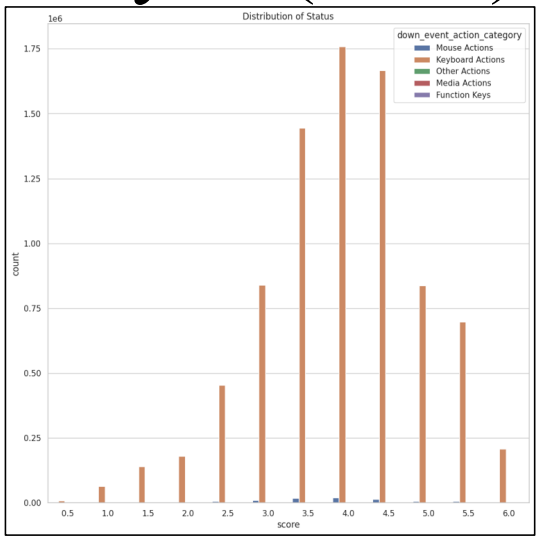



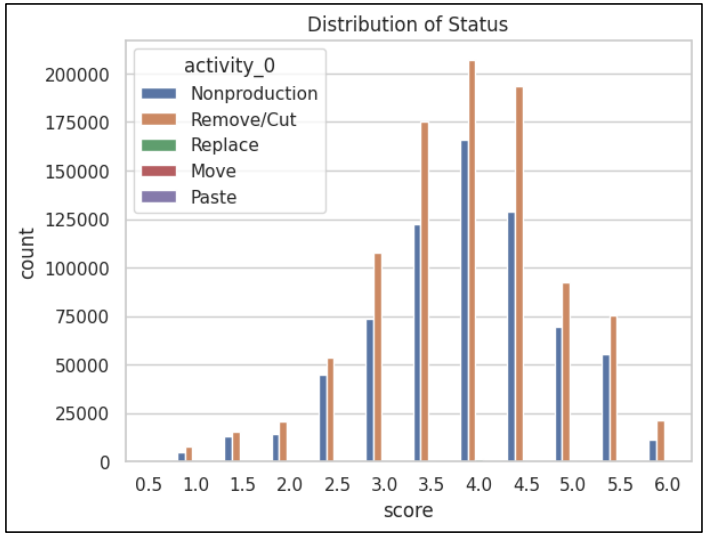

**Findings**

- **Score Trend:** Essays typically score between 2 to 5.
- **High Correlation:** 'Down_time' and 'up_time' are closely linked, suggesting possible multicollinearity.
- **Data Characteristics:** Noticeable skewness and several outliers in 'up_time' and 'down_time' distributions.
- **Preprocessing Need:** To address the correlation and outliers, preprocessing, like log transformation or normalization, is needed, which is further proven as the important recommendation from the EDA.

# Exploratory Data Analysis (EDA) - II

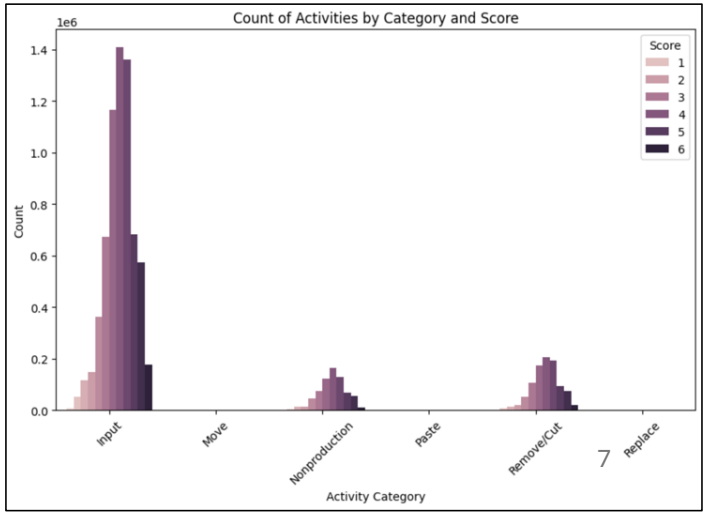



## Findings

- The score distribution, ranging from 0 to 6, shows a normal spread.
- Users frequently engaging in media actions, possibly indicating distraction, usually score around 3.0, hinting at a potential link between media actions and lower scores.
- A decrease in the Non Production vs. Remove/Cut ratio correlates with higher essay scores, suggesting less non-productive activity leads to better performance.
- Compared to Input and Non Production actions, Replace, Paste, and Move actions are less common among users while writing essays.



# Feature Engineering and Experimental setup

- Further, we engineered more features to detect intricacies.
- Assumption for designing these features: Average length of sentence of 50 characters, according to surveys.

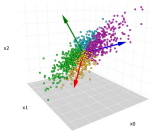
| Feature Name | Feature Type (Categorical/ Numerical) | Description |
|---|---|---|
| **unsurity** | Categorical (binary: 0 or 1) | Indicates large text removal (>50 characters) during 'Remove/Cut' activity. 1 for unsurity, 0 otherwise. |
| **structural_change** | Categorical (binary: 0 or 1) | Shows if a large text change (>50 characters) was a 'Replace' activity. 1 for structural change, 0 otherwise. |
| **long_paste** | Categorical (binary: 0 or 1) | Flags large text addition (>50 characters) due to 'Paste' activity. 1 for long paste, 0 otherwise. |
| **unproductive_time** | Numerical | Total time spent in 'Nonproduction' activities, quantifying unproductive duration. |
| **external_help** | Categorical (binary: 0 or 1) | Indicates reliance on external help: 'Paste' activity when word count < 10. 1 for external help, 0 otherwise. |
| **pasted_words_number** | Numerical | Number of words added during 'Paste' activities, quantifying the extent of text addition. |
| **large_changes** | Categorical (binary: 0 or 1) | Reflects whether a text change was over 50 characters, regardless of activity type. 1 for large changes, 0 otherwise. |

- Further, we aggregate these features for each users resulting in count or sum of these things happening across the duration of essay writing.

# Feature Engineering and Experimental setup

- In addition to previous feature engineering, we need to capture the things that a person do while writing an essay. This includes what type of keys the writer pressed while working on the essay.
- For this, we created took the "up_event" column, which basically tells from which key the writer lifted the fingers from, for example, Enter, Space, F1, F2, etc and created a user level event counting.
- By going to user level, we went from **8,405,898** data instances to **2,471** data points
- By doing feature engineering we went from **11** features to **146** features

- Going further we kept two versions of data, PCA and Non-PCA(original data)
- By applying PCA (Principal Component Analysis), the features were reduced to 30, capturing 95% of the variance.
- These dataset were used in all further experiments, both regression and classification.



## With PCA ( 95% variance covered )

| | Training | Validation | Testing |
|---|---|---|---|
| X | (2000,30) | (223,30) | (248,30) |
| Y | (2000,1) | (223,1) | (248,1) |

## Without PCA

| | Training | Validation | Testing |
|---|---|---|---|
| X | (2000,146) | (223,146) | (248,146) |
| Y | (2000,1) | (223,1) | (248,1) |

# Benchmarking according to Modelling Strategies

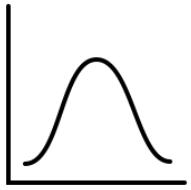
- There are 12 score which are assigned to the users: 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6
- In our proposal, the exploratory data analysis (EDA) provided insight that the 'score' label for supervised learning could be addressed with both classification and regression strategies.

## Classification

- To calculate the baseline accuracy, you can use the following formula:

$$Baseline\ Accuracy = \frac{Number\ of\ Instances\ with\ most\ frequent\ class}{Total\ Number\ of\ Instances}$$

- Baseline Accuracy for Classification = 0.2027
- We will be treating this as the benchmarking accuracy for model comparison and conclusions.
- If a model has more accuracy than the limiting accuracy, we will accept that model.

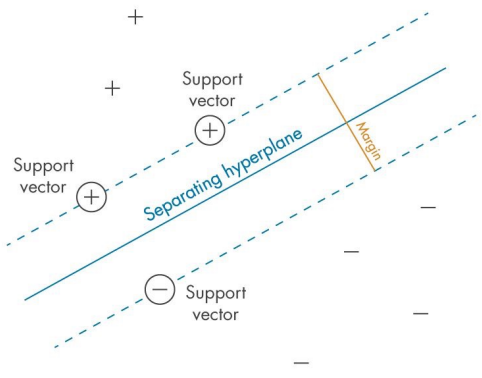


## Regression

- Consider here that, the grader grades each essay by randomly picking the score from normal distribution which has the mean and standard deviation.
- Taking Root Mean Square of scores generated by the normal distribution = 1.4317
- This would be our baseline for the regression models

- Note: Taking RMSE of by considering each user/essay gets the mean value = 1.0246

# Classification using SVC (Support Vector Classifier)




- Support Vector Classification (SVC) is specifically designed for classification tasks; it works by finding the hyperplane that best separates different classes in the feature space.
- We used hyperopt library to find the best hyperparameters that fit the training dataset.
- Without PCA: Kernel = 'rbf'; C = 6.9580 ; gamma = 0.023
- With PCA: Kernel = 'rbf'; C = 0.7204; gamma = 0.053(rbf - radial basis function)

## With PCA

- **Validation Accuracy: 0.2960**
- **Test Accuracy: 0.2984**

## Without PCA

- **Validation Accuracy: 0.3094**
- **Test Accuracy: 0.2863**



*References:* https://uk.mathworks.com/discovery/support-vector-machine.html

# Regression using LGBRegressor

- The LGB (Light Gradient Boosting) Regressor belongs to the gradient boosting family, where new models are added to correct the errors made by existing models. Additionally, it supports regularization to avoid overfitting.
- We used hyperopt library in python to get the best performing parameters on the validation set, which were then used to learn a new LGBM regressor on training data.
- LGBM trained with best parameter was used to do prediction on the un-touched test
- The RMSE obtained by the LGBRegressor is 1.5 times better than our benchmark

## With PCA

- **Validation RMSE: 0.6949**
- **Test RMSE: 0.7217**

## Without PCA

- **Validation RMSE: 0.6248**
- **Test RMSE: 0.6255**

# Deep Learning – Multilinear Perceptron (MLP)

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_4 (Dense)             (None, 128)               18816

 dense_5 (Dense)             (None, 64)                8256

 dense_6 (Dense)             (None, 16)                1040

 dense_7 (Dense)             (None, 1)                 17

=================================================================
Total params: 28129 (109.88 KB)
Trainable params: 28129 (109.88 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- Multi layer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers—the input layer, output layer and hidden layer.
- Optimizer: Adam
- Loss: Mean Squared Error
- Activation function: ReLU (for the hidden layers)

## With PCA

- **Validation RMSE: 0.7651**

## Without PCA

- **Validation RMSE: 0.7045**

# Deep Learning - Transformers

| event_id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 990 | 991 | 992 | 993 | 994 | 995 | 996 | 997 | 998 | 999 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| id | | | | | | | | | | | | | | | | | | | | | |
| 001519c8 | Leftclick | Leftclick | Shift | q | q | q | q | q | q | Space | ... | q | q | q | Space | q | q | q | Space | q | q |
| 0022f953 | Leftclick | Shift | q | q | q | q | Space | q | q | Space | ... | Backspace | , | Space | q | q | q | Space | Backspace | q | , |
| 0042269b | Leftclick | Shift | q | q | q | q | q | q | q | Space | ... | q | q | Space | q | q | q | q | q | q | q |
| 0059420b | Leftclick | Leftclick | Shift | Shift | Shift | Shift | Shift | Shift | Shift | Shift | ... | q | q | q | q | q | Space | q | q | q | Space |
| 0075873a | Leftclick | Shift | q | q | q | q | q | q | q | q | ... | q | q | q | q | Space | q | q | Space | q | q |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ffb8c745 | Leftclick | Tab | Leftclick | Space | Space | Space | Space | Space | Shift | q | ... | q | q | . | Space | Space | Shift | q | q | q | q |
| ffbef7e5 | Leftclick | Leftclick | Shift | q | q | q | q | Space | q | q | ... | q | q | q | q | Space | q | q | Space | q | q |
| ffccd6fd | Leftclick | Leftclick | q | q | q | q | q | q | Space | q | ... | ArrowLeft | Backspace | ArrowDown | q | q | q | q | q | Space | q |
| ffec5b38 | Leftclick | Shift | q | q | q | q | q | q | q | q | ... | q | q | Space | q | q | q | Space | q | q | q |
| fff05981 | Leftclick | Leftclick | q | q | Space | q | q | q | q | Space | ... | q | q | q | q | q | q | q | q | q | q |

2471 rows × 999 columns

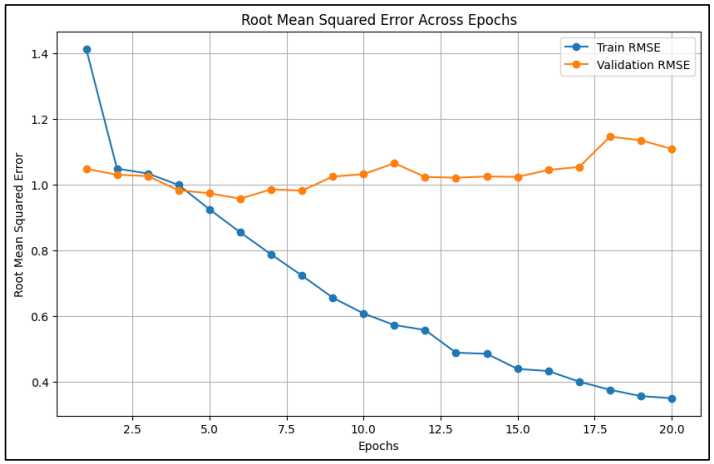
**Model: "functional_1"**

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_layer (InputLayer) | (None, 1000) | 0 |
| token_and_position_embedding (TokenAndPositionEmbedding) | (None, 1000, 32) | 36,160 |
| transformer_encoder (TransformerEncoder) | (None, 1000, 32) | 12,442 |
| transformer_encoder_1 (TransformerEncoder) | (None, 1000, 32) | 12,442 |
| transformer_encoder_2 (TransformerEncoder) | (None, 1000, 32) | 12,442 |
| get_item (GetItem) | (None, 32) | 0 |
| dense (Dense) | (None, 1) | 33 |

**Total params:** 73,519 (287.18 KB)
**Trainable params:** 73,519 (287.18 KB)
**Non-trainable params:** 0 (0.00 B)

- Integer Vectorization using TextVectorization layer from keras: by representing each word in the input text with a unique integer index.
- TokenAndPositionEmbedding layer: Embeddings created by this layer are not explicitly pre-trained on external data. Instead, they are likely initialized randomly and fine-tuned during the training of the entire model.
- Sequence Length: 1000 (The first 1000 events are considered per user/essay).
- We observe that the model overfits on the training data from the train_rmse and val_rmse plots.
- Overfitting might be attributed to the limitation of training data. (~2000 training samples.)

**Val RMSE: 1.1086**



# Conclusion & Future Work

- Through our exploration of both regression and classification machine learning models, we've identified our best options as 'LGB' for regression and 'SVC' for classification.
- Regression is our top choice for this task, as the competition's host page specifies 'quality' as the evaluation metric, which is associated with RMSE (Root Mean Square Error).
- Another reason for favoring regression is the large number of target variable 'score' classes.

- Optimization of the Transformer: The transformer model's optimization is pending due to our limited knowledge of transformers.
- Sequence Length Adjustment: Increase the sequence length as memory allocation issues on Google Colab have been resolved.
- Advanced Feature Engineering: Implement more sophisticated feature engineering based on domain experts' recommendations.

# Appendix – Correlation map of engineered features