Key to Quality: Decoding the Language of Keystrokes for Predictive Insights

Anuranjan Pandey, Dhruv Maheshwari, Kriti Jha, Nesar Nanjundaswamy, Pratyush Bhatnagar
CSCI 567 Machine Learning, Viterbi School of Engineering

Introduction

It's difficult to summarise the complex behavioral actions and cognitive activities in the writing process. Writers may use different techniques to plan and revise their work, demonstrate distinct pause patterns, or allocate time strategically throughout the writing process. Many of these small actions may influence writing quality. Even so, most writing assessments focus on only the final product. Data science may be able to uncover critical aspects of the writing process.

Past research explored some process features related to behaviors such as pausing, additions or deletions, and revisions. However, previous studies have used relatively small datasets.

Using feature engineering, we have tried to link keystroke logs to predict overall writing quality. These efforts may identify relationships between learners' writing behaviors and writing performance. This may help direct learners' attention to their text production process and boost their autonomy, metacognitive awareness, and self-regulation in writing.

Data Preprocessing

Dataset:

- Kaggle Competition – data contributed by Vanderbilt University
- Consists of 8,405,898 rows of 2471 unique essays
- Numerical columns (down_time, up_time, action_time, cursor_position, word_count)
- Categorical columns (activity, up_event, text_change)

Feature engineering:

- Implemented an additional 49 features on categories such as content and typing, editing, pauses and cursor movement.
- Few features
  - Content and typing (Word length, Non-production time, typing speed)
  - Editing (Word length, D/I Ratio, Major edits, Distant Revisions, typing speed)
  - Pauses (Total pause & time, IKI, pauses within sentence, pauses before sentences, Bursts, Bursts duration)
  - Cursor movement (Total movement, Distance moved)
- For time series:
  - One hot encoded 'activity' column keeping only 'Nonproduction', 'Input', 'Remove/Cut', 'Replace'
  - One hot encoded 'up_event' ∀ occur(symbols)>100,000
  - One hot encoded 'text_change' ∀ occur(symbols)>50,000
  - Padded sequences to make them of equal length and removed outliers

Modelling

Modelling:

Scores were converted to classes (labels) for classification and PCA and SHAP values were utilized to assess the impact of features on essay score, we implemented cross validation to tune hyper parameters.We also used scaler function

- Multinomial & Ordinal Logistic regression
- K-means using PCA
- Multinomial Naïve Baiyes
- Neural Network (4 hidden layers (64, 32, 16, 8) with dropout and ReLU, adam opt and MSE for loss)
- MLP Classifier (3 hidden layers (50,50,25), logistic, adam opt, adaptive lr, lr_initial = 0.01)
- XGBoost (classification: maxdepth: 4, α: 0.05, N-estimators: 90), (regressor: maxdepth: 4, α: 0.11, N-estimators: 66)
- LGBM

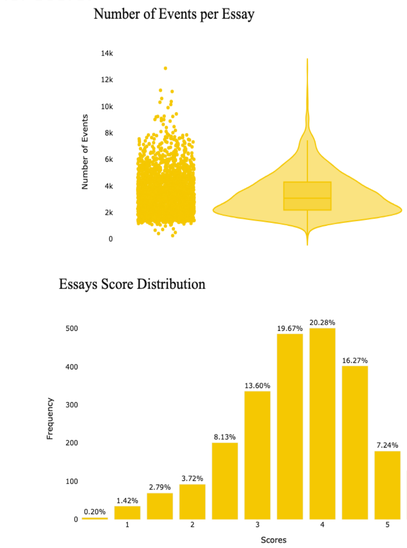
For time series analysis

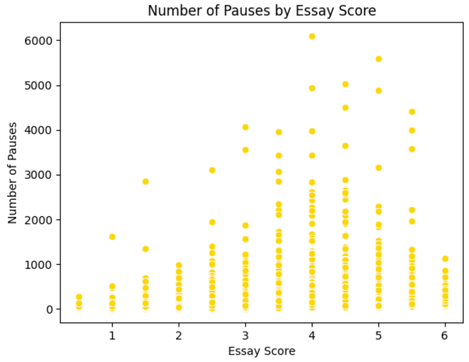
- LSTM (256 units, 3 hidden layers, 32 batch size)
- Transformer (8 embedding dimensions, 2 encoder layers, 2 heads, 8 batch size)

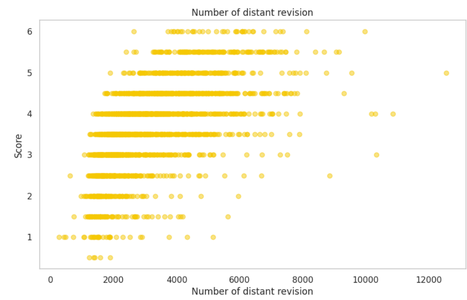
Observation

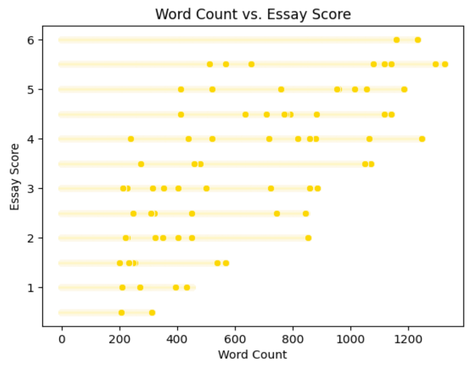
The following observations were made :

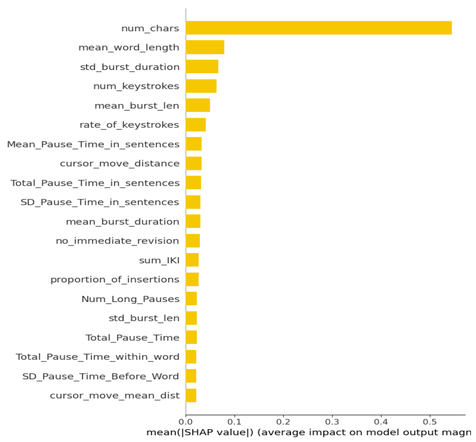
1. The average number of events per essay was around 3500.
2. The number of pauses was higher in essays scored between 3.5 to 5.5.
3. The more distant revisions, the better the score.
4. A higher number of words in the essay corresponds to higher score.











Results

ALGORITHM	RMSE
Multinomial Logistic Regression	0.684
Ordinal Logistic Regression	2.035
K-Means (K++)	2.0289
Multinomial Naïve Bayes	0.7198
Neural Network	0.7824
MLP Classifier	0.7063
XGBoost Regression	0.6550
XGBoost Classification	0.7286
LSTM	N/A
Transformer	N/A
LGBM Regression	0.6791

Challenges and future work

Challenges faced:

- LSTM : Large padding and sequence length made learning difficult.
- Transformer: Due to the large length of the essays, even the small transformer networks were running out of memory. i.e. even training upon 30 GB of RAM and P100 GPU,
- Address issues related to imbalanced class distribution. We can try and model essays with higher and lower score more using techniques like oversampling.

Future work:

- We could explore more features and models.
- Given access to more compute and memory, we can try even complex and bigger models like transformers.
- Explore AutoML (Automated Machine Learning) approaches to automate the process of model selection, hyperparameter tuning, and feature engineering. This can save time and resources in the model development pipeline.
- Optimize existing models for better computational efficiency, i.e.explore techniques like model pruning, quantization, or distillation to reduce the model size without significant loss of performance.
- Use of ensemble methods to combine predictions from multiple models to improve generalization and robustness.