Treating Code as Text: Code Authorship Assignment

Gabriel Krotkov, Zachary Strennen

December 2023

1 Introduction

In 1963, Mosteller and Wallace published a landmark paper in authorship analysis, *Inference in an Authorship Problem* [2], in which they built evidence that Madison was the likeliest author of the contested 12 Federalist Papers, with varying confidence for each of the individual papers. Their analysis was also insightful about shared authorship, in that for papers jointly authored, a binary model's level of confidence that the text was authored by one author seemed to reflect the size of that author's contribution to the joint authorship.

Could this analysis be applied to code? Fundamentally, coding is just another form of writing and many programmers have highly individual coding styles. In fact, some researchers have already shown that source code can be treated as a text for attributing authorship [3]. We are curious about the dynamics of code authorship analysis. In particular, we are interested in answering the following questions:

1. Can code be assigned authorship by the same methods that work for essays?
2. What features drive code authorship assignment between different pairs of authors?
3. Do the same features tend to be explanatory between different authors, or does each comparison have its own boutique most explanatory variables?
4. Do conglomerate authors (groups of experts that follow a consistent style) have identifiable traits similar to individual authors?

To answer these questions, we collected a corpus of texts grouped by author and applied a similar analysis to Mosteller and Wallace after tokenizing the code to account for key R operators. We considered key R-reserved words, operators, and major package namespaces as potential discriminators between pairs of authors, and then used Principal Component Analysis to summarize the whole matrix of pairwise author comparisons and draw insights about code authorship analysis in general.

2 Data

The data consists of a corpus of R code files read in as text. Most files were written by one individual author (usually an undergraduate student at CMU), but some text groups were written by a group of authors which conformed to a single style guide. Each group of R code files is assigned a unique id encoding whether it was authored jointly or individually, and each file is assigned a unique id of its own to distinguish it from other files by the same author. There are ten files per author and eight authors total.

3 Methods

3.1 Tokenization

The text from each R file is specifically transformed in order to be tokenized. Each space in the file is represented by the token "space_placeholder" and each new line is represented by the token "new_line_placeholder". Additionally, several non-alphanumeric operators unique to R, as well as common, punctuation-based operators are replaced with placeholder tokens. Here are a few examples of these placeholders:

R Operator	Tokenized
<-	assign_arrow_left_placeholder
->	assign_arrow_right_placeholder
%<%	pipe_arrow_left_placeholder
%>%	pipe_arrow_right_placeholder
::	namespace_placeholder
(	open_parentheses_placeholder
)	close_parentheses_placeholder
+	plus_sign_placeholder

Overall, there are 48 unique placeholder tokens used to transform each text. While not all placeholders will be present in each text, it can be hypothesized that the proportionality of these placeholders, along with the use of namespaces, can be used to identify the author of an R code file.

Once all of the files are loaded in and tokenized, a weighted document feature matrix is created for the loaded tokens separated by each file along with the associated author id. Reminiscent of the classification done by Mosteller and Wallace, lasso regression is used to determine authorship. Initially using all collected tokens for each file as predictor variables, we determine the probability of authorship between two students in question.

We used a 60% training data and 40% testing data split. The response variable is a probability to determine authorship. If the probability is closer to 0 authorship will be ascribed to one author and if the probability is closer to 1 authorship will be ascribed to the other author author, and we will select probability 0.5 as the dividing line. We repeated with process for several pairs of anonymous authors with the hopes of attaining high accuracy when predicting authorship.

3.2 Pairwise LASSO Authorship Prediction

For Mosteller & Wallace's analysis, they started with a list of words they expected would be key discriminators and started their modelling investigation with those words. For our purposes we need to generate a similar list of "good guesses" as to what tokens might be effective discriminators between code authors. To do this, we started with the key R operators that we tokenized in the previous section. However, those operators do not make an effective starting list of potentially explanatory discriminators - many key and potentially distincting features of R code are missing. To fill in this gap we added all the reserved words of the R programming language, as well as key namespace of frequently-used packages like "ggplot2", "dplyr", "tibble", and "roxygen2". Then we added the namespace of key packages that come built in with R, including "stats", "base", and "utils". All told, this resulted in a list of more than 6000 candidate discriminators, and we suspect it covered a good number of the most effective possible discriminators.

Discriminator Group	Source	Reasoning
R-Reserved Symbol	Tokenization process described in (3.1)	Reserved symbols are key to interpreting R code.
R-Reserved Words	Exhaustive List of R-reserved words	Reserved words are key to interpreting R code.
Tidyverse Namespace (dplyr, ggplot, etc.)	R Script Written to ID namespace of a package	The tidyverse includes some of the most frequently used R packages and so their usage is reasonable as a discriminator.
Base R Namespace (base, utils, stats, etc)	R Script written to ID namespace of a package	These packages come inbuilt on all R downloads and their use is frequently taught in introductory R courses.

To analyze the pairwise differences between groups of texts by two different authors we used LASSO regression (explained in more detail in Robert Tibshirani's paper [4]) to logistically predict

the likelihood of the authorship of a given text being attributed to the first author as opposed to the second author. Lambda was selected by cross validation for each pairwise comparision between two authors. We selected LASSO instead of other methods to reduce the number of variables retained in the model and to maintain some simplicity. The resulting model has coefficients which can be interpreted to mean "the average expected increase in the log odds of the first author being the true author of the paper if the given token happened to increase in proportion from 0 to 1."

3.3 Round Robin Authorship Analysis

One key difference between the analysis of Mosteller & Wallance and our analysis is that the Mosteller & Wallace study was focused on the attribution of specfic texts, for which a specific two candidates were known. Our interest is in code authorship more generally, and so our analysis needs to account for differences between a variety of authors. In order to achieve this, we designed a "round robin" series of pairwise author comparisons (referenced as author 1 against author 2) and an analysis to compile the results of many different pairwise comparisons. First, for each author we conducted a pairwise comparison as described in section 3.2, resulting in a model equation out of the LASSO with coefficients for each discriminator of significance.

To estimate the effectiveness of the whole procedure by cross validation we withheld 40% of the data from each model training procedure (so every model was tested the same amount) and took the average performance of each model on the withheld data as our cross validated metric of model performance.

Then, taking the vector of the coefficients as an embedding (and filling NULL spaces with 0 to make the number of columns match) we created a dataframe where each row represents a unique pairwise comparison of two different authors, and each column represents the token used as a discriminator between the two authors. The cell of a given row and column, then, is filled with the β coefficient indicating the contribution of that discriminator to the increasing likelihood of author 1 being the true author.

This leaves us with many more columns than can feasibly be analyzed, so we are interested in simplifying the information in the matrix to easy analysis. To achieve this, we performed dimension reduction using Principal Component Analysis to summarize the pairwise comparisons. A more in depth description of Principal Component Analysis can be found in Mackiewicz and Ratajczak's paper on the subject [1]. However, it can also be understood as a way to compress the relational information in a dataset into a small number of dimensions (usually 2) by creating vectors that account for the maximum amount of variance possible while being subject to the constraint of being orthogonal to the previous vectors ("principal component"). When visualizing the dimensions of our Principal Component Analysis, the data will be colored by the authorship comparison type: group to group, group to individual, and individual to individual.

4 Results

4.1 LASSO Authorship Prediction

In figure 1, you can see an example of the optimization of lambda for a single pairwise comparison between two authors. In our initial tests between sample code from CMU students, the LASSO approach showed its validity for pairwise authorship prediction, correctly classifying withheld validation texts with less than 10% cross validation error on average. This, at a baseline, answers our first research question - the LASSO approach to authorship attribution does work at least somewhat when applied to code. However, to answer our other research questions a method that considers information from many different pairwise comparisons is required.

Taken as a whole each individual LASSO model tended to achieve a reasonable performance on the withheld test data. In figure 2, you can see a plot of the cross validation accuracy rate of each LASSO model fit for the data. The models together averaged an 84% cross validation accuracy rate, indicating that in general the method is successful at identifying authorship broadly and correctly assigning it to data the model has not seen.

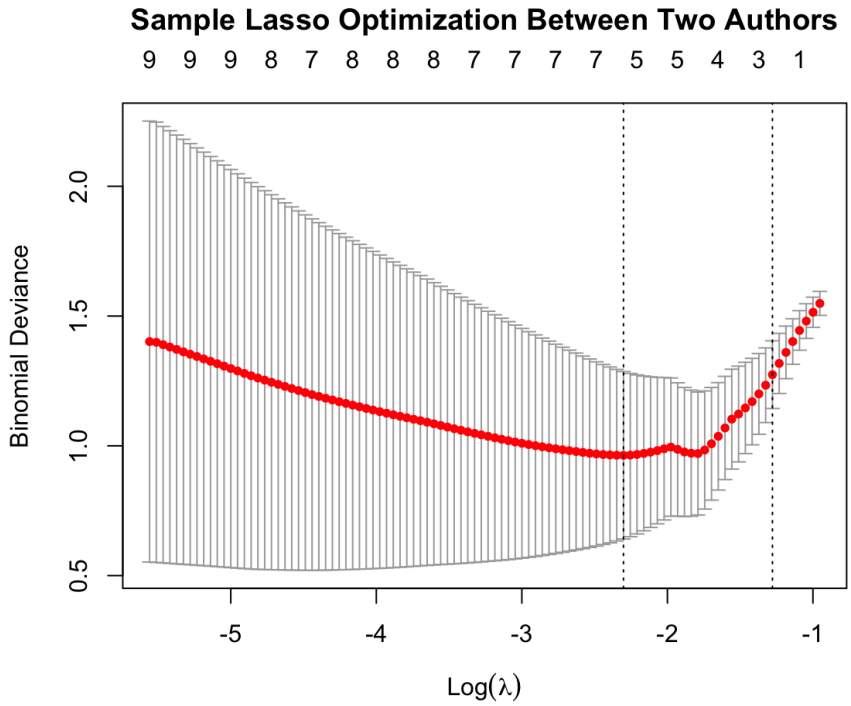

Figure 1: LASSO Pairwise Authorship Prediction

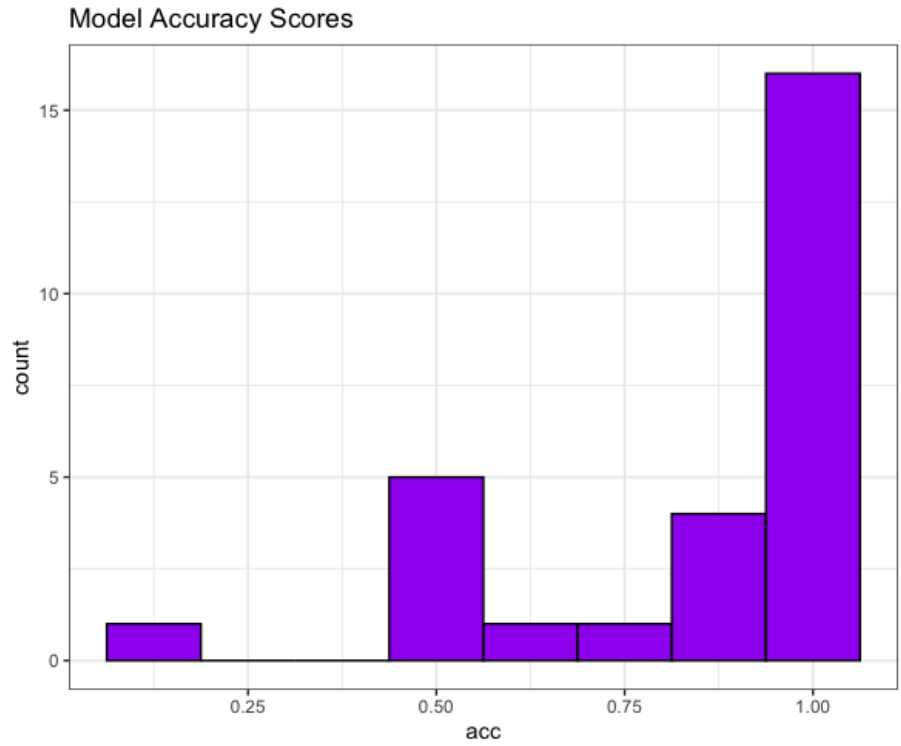

Figure 2: Model Accuracy Rates

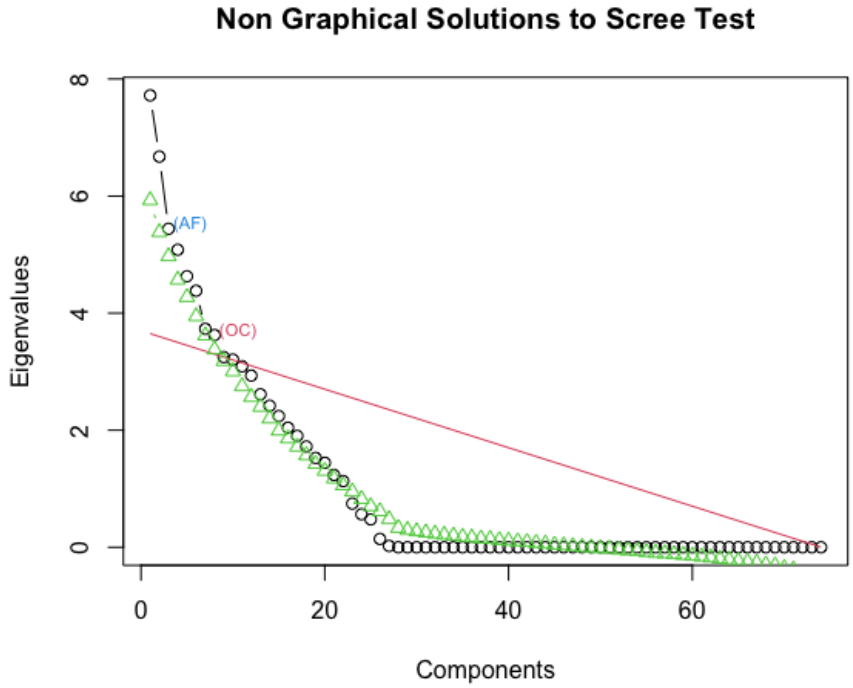

Figure 3: Scree Plot

4.2 Principal Component Analysis

The first step of our Principal Component Analysis was to make a scree plot to get a sense of the distribution of variance accounted for by each principal component. In our case, it appears that relatively many principal components explain meaningful amounts of variation in the data - the "elbow" of the scree plot doesn't really flatten out until the 25th component, indicating that the variance accounted for is relatively flatly decreasing from the 1st principal component to around the 20th. However, we can identify a "mini-elbow" between around the 8th and 9th principal components, indicating that the first 8 dimensions of the PCA matrix are the most explanatory.

We can see a similar dynamic in our contribution plot. Around 40 variables have a higher-than-expected contribution to PCA dimensions 1 through 8 (inclusive), with a very flat decrease of contribution to the expectation line. This indicates that many variables contribute nearly-equally to the key principal components that explain important variation in the data.

We cannot plot all 6 of the key principal components on 2 dimensions of a plot. However we plotted each PCA plot individually and identified the principal components that seemed the most explanatory for our variable of interest (comparison type), and included those (principal components 4 and 6) in the PCA plot in figure 5. Judging by the scree plot, these principal components still explain a great deal of variation in the data, and these components effectively group group-to-group comparisons away from other comparisons.

Another analysis we performed was looking at a biplot of the largest contributors to the key two principal components (4 and 6). The strongest contributors appear to be aligned along the 2nd quadrant to 4th quadrant diagonal. Comparing with the PCA plot, this appears to roughly be aligned with group-to-group comparisons. According to the biplot and PCA plot combined, it appears that parenthesis usage is much more explanatory when at least one individual is involved in the comparison, which makes intuitive sense, since many individuals have their own habits regarding parenthesis usage, while large groups of authors following a style guide together may have a specific behavior they are expected to follow when contributing code to the project. Similarly, the keyword "rename" is a much more effective discriminator when used in a group to group comparison context. This is perhaps picking up on tidyverse-obedient projects contrasted to non-tidyverse projects that don't use the rename structure.

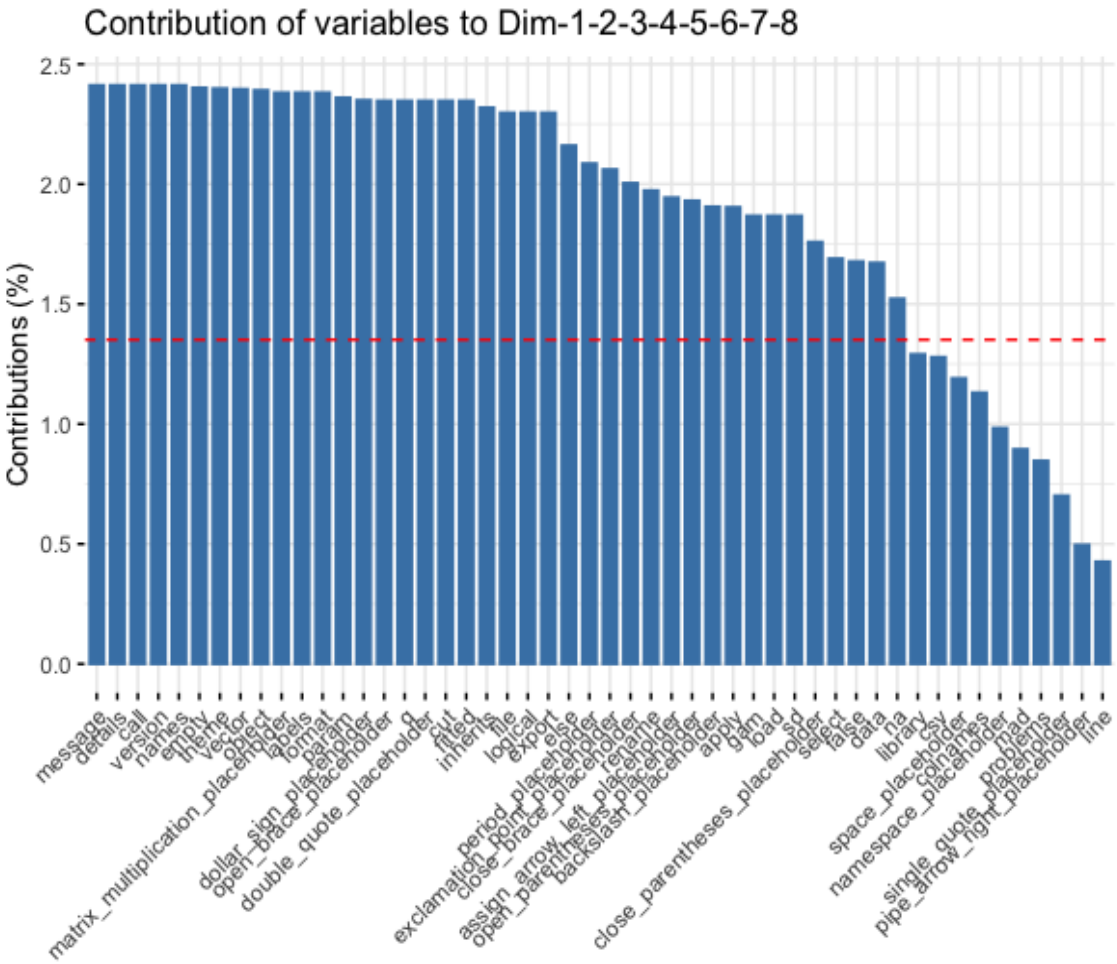

Figure 4: Variable Contributions

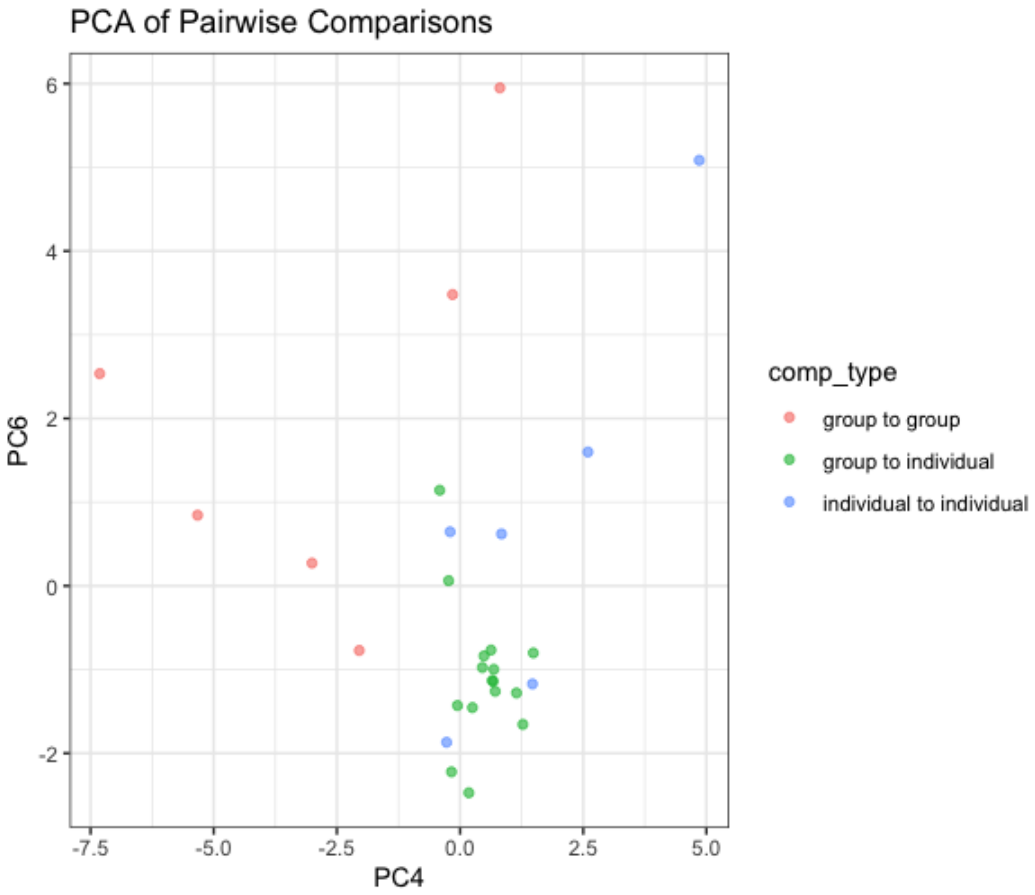

Figure 5: Principal Component Analysis

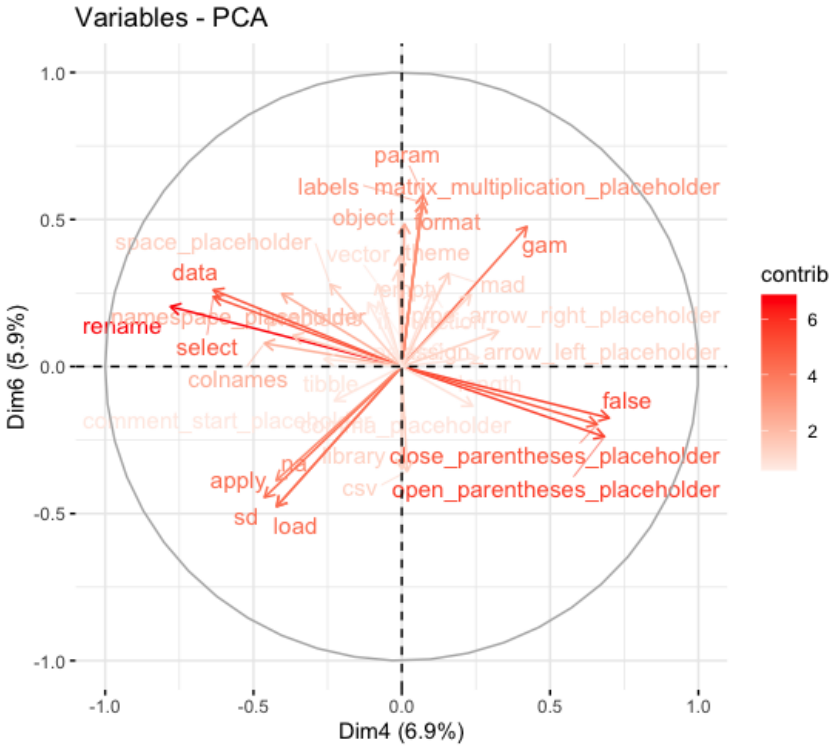

Figure 6: Biplot

5 Discussion

Our analysis identified that the procedure outlined by Mosteller & Wallace does in fact still perform well for code analysis, given a thoughtful tokenization of the code text. Individual LASSO models generally performed highly in the cross validation, earning a cross validation error rate of less than 16%. Furthermore, we were able to identify that these methods are still valid when aggregate authors are included in the analysis, but only considered aggregate authors that attempted to follow a consistent style guide. We also identified that in general the explanatory impact of each discriminator is highly variable (e.g. our highly spread contribution plot) between different pairwise comparisons, indicating that each pairwise comparision has its own key discriminators which are unlikely to be influential for another pairwise comparison.

It appears that aggregate authors do effectively have "code fingerprints" in the same way that individual authors do, but more analysis would be valuable to identify and quantify this. In particular, it would be valuable to gather more data so that the data set could be segmented further during cross validation to retrieve high quality estimates of the dropoff (if any) in model performance when comparing group authors to group authors as opposed to a comparison including an individual author.

One limitation of our analysis is that it does not make a distinction between an evaluation of a jointly authored text and evaluation of an individually authored text when modeling. Instead, only pairwise comparisons are made. Future analysis could explicitly focus on deciding the percentage of authorship of a jointly authored text given a list of contributors and texts authored individually by each of those contributors to form a baseline. However, this analysis does not explore that space.

Another future improvement worth pursuing is to consider a random forest classifier instead of a LASSO model approach. This change could be more effective in considering all the possible tokens to accurately predict authorship, but would trade off some of the interpretability of the LASSO model, since it does not produce a convenient vector of coefficients like LASSO does to compress into two dimensions and plot with PCA.

Once again, the data in this analysis are very limited. To produce a more succinct accuracy rate, more authors could have been included beyond 8. More texts from each author would have been beneficial to the lasso as well. While a 60% training 40%, proved beneficial for this model, this parameter could have been tuned more succinct.

Ultimately, we found that code can loosely be assigned authorship by the same methods that work for essays given respect to specific attributes of the code. The features that drive code authorship assignment are very dynamic across each pairwise comparison. However, given the PCA, we can see that conglomerate authors can be differed from individual authors.

Going forward, we would like to continue the relationship of the number of unique authors and the number of high-contributing discriminators as observed in the results section. If the two quantities grow in proportion, that could indicate that each pairwise comparison is boutique and has its own set of key discriminators rather than sharing them with other comparisons.

References

[1] Waldemar Ratajczak Andrezj Mackiewicz. "Principal Components Analysis". In: *Computers & Geosciences* (1993). URL: https://www.sciencedirect.com/science/article/pii/009830049390090R (visited on 12/10/2023).

[2] David L. Wallace Frederick Mosteller. "Inference in an Authorship Problem". In: *Journal of the American Statistical Association* (1963). URL: https://www.jstor.org/stable/2283270 (visited on 12/10/2023).

[3] Stefanos Gritzalis Georgia Frantzeskou Efstathios Stamatatos. "Identifying Authorship by Byte-Level N-Grams: The Source Code Author Profile (SCAP) Method". In: *International Journal of Digital Evidence* (2023). URL: https://www.utica.edu/academic/institutes/ecii/publications/articles/B41158D1-C829-0387-009D214D2170C321.pdf (visited on 11/28/2023).

[4] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society* (1996). URL: https://www.jstor.org/stable/2346178 (visited on 12/10/2023).