

Predicting Authors

Olalekan Bello and Gaetano Dona-Jehan

May 11, 2021

Abstract

We employ both supervised and unsupervised learning methods to predict the authorship of philosophy text across four authors (Plato, Aristotle, Kant, Hume). For our supervised model, we use multinomial lasso regression using the tokens (words) as features and a line of text as the outcome. We get an accuracy of 67 percent with this technique. The unsupervised model uses topic modeling to give the probability that a document belong to a particular author. The results from both approaches are similar in that they perform well in distinguishing Kant and Hume from Plato and Aristotle but both models particularly fail to distinguish Plato from Aristotle.

Overview

Philosophy has an extensive history starting all the way from the era of the proto-philosophers such as Heraclitus and Thales to the early western philosophers like Plato and Aristotle to the later westerners such as Kant and Hume. Ideas are often transferred and refined from generation to generation and also within contemporaries. For this project, we aim to see if we could accurately distinguish ideas across a set of philosophers using both supervised and unsupervised machine learning tools. Our goal is to determine whether we can take a line from our set of authors and their works and accurately predict who wrote it.

We focus on four philosophers; Plato, Aristotle, Immanuel Kant and David Hume. We selected this group of philosophers for two reasons. One being that they are key figures in the history of philosophy. The second being that there exists relationships between the of authors that make the problem more interesting. Firstly, Plato and Aristotle are contemporaries as they both “published” works in the 4th century BC, whereas Kant and Hume published their essays in the 18th century. Given this, there might be a similarity that exists between authors in terms of writing styles and patterns that are common to their respective eras. There is also the issue of translations. Plato and Aristotle both originally wrote in Greek which has now been translated to English and Kant also originally wrote in German. Translations often contain the writing

mannerisms of the translator, meaning that, although the original works are from two different authors, the translated works might share similar writing patterns due to being translated by the same translator.

In terms of relationships, there also exists a mix of ideas between the authors. Plato is known as the father of western philosophy which all the authors belong to and was actually a direct teacher of Aristotle. Kant and Hume were also “competitors” in a sense in that they belonged to different schools of thought within Western philosophy. Kant even famously credited Hume for “awaking him from his slumber”.

Data and Models

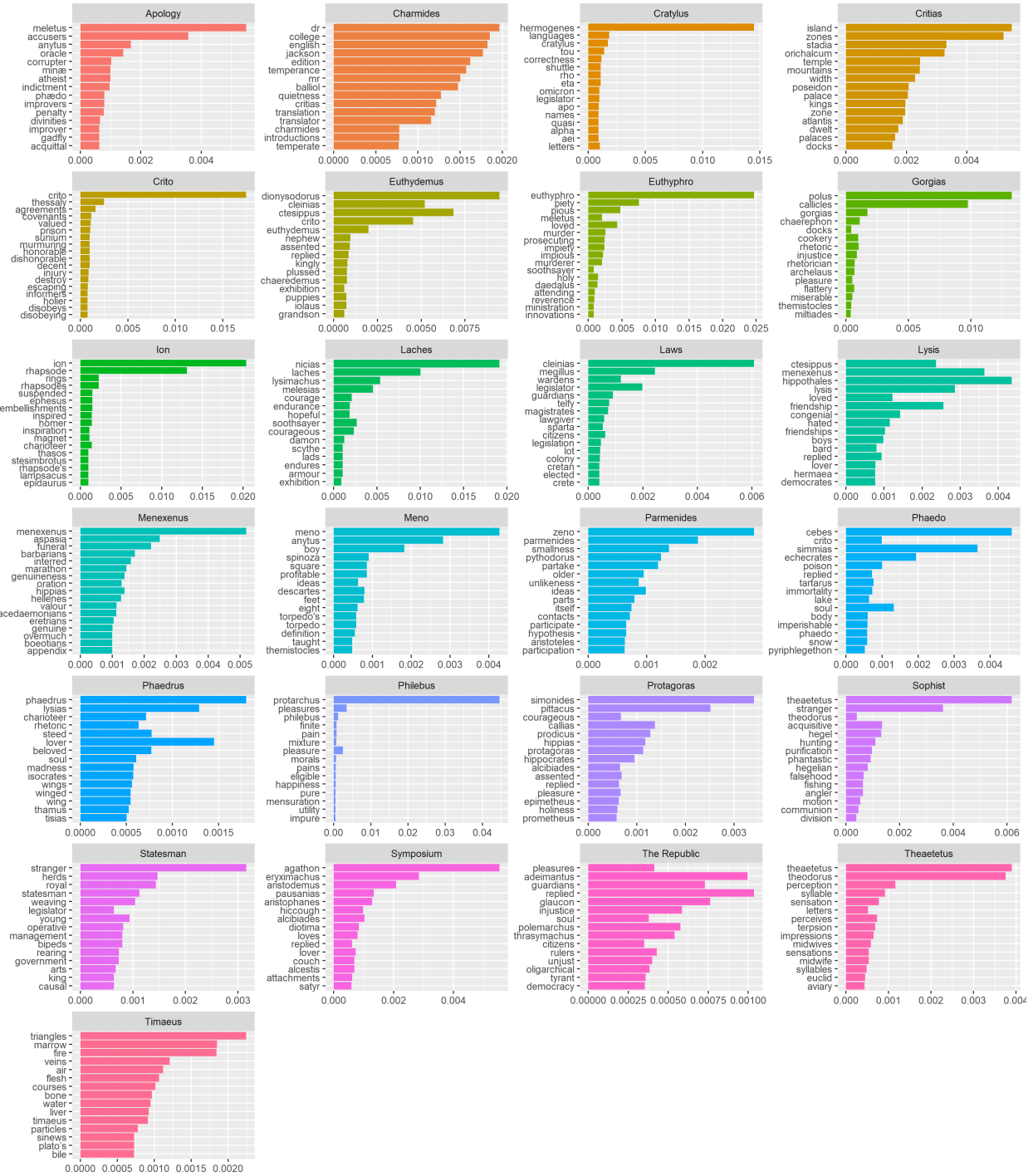
Our primary data source is the gutenbergr project ¹. The gutenbergr project is an online library of over 60,000 free e-books that span a long range of time and genres. We access this library using the “gutenbergr” package which allows us to directly download and process public domain works from the Gutenberg project collection as well as their metadata. The metadata that is given includes information about each work that we downloaded with their Gutenberg ID, title, information about the author, the language etc... We process this data using a number of text based packages in R. We break our text down into tokens whereby each token represents a feature of our dataset. Instead of using raw counts as our variables of interest, we use the slightly more sophisticated term-frequency inverse document frequency (tf_idf) ². Basically, the tf-idf is able to look within groups and identify the words that are most important to that particular group. This eliminates the need to manually remove stopwords such as “a, the, and, or” etc. because they would be weighted at 0 or very close to 0 because they are common across groups.

To start out, we take a look at the tf-idfs within Plato’s works, we see that the words that most uniquely identify Plato’s works are the names of characters which makes sense as Plato mostly expressed his philosophy through stories. Several of his works such as Euthyphro, Parmenides, Crito and Critias etc. are even named after the main character.

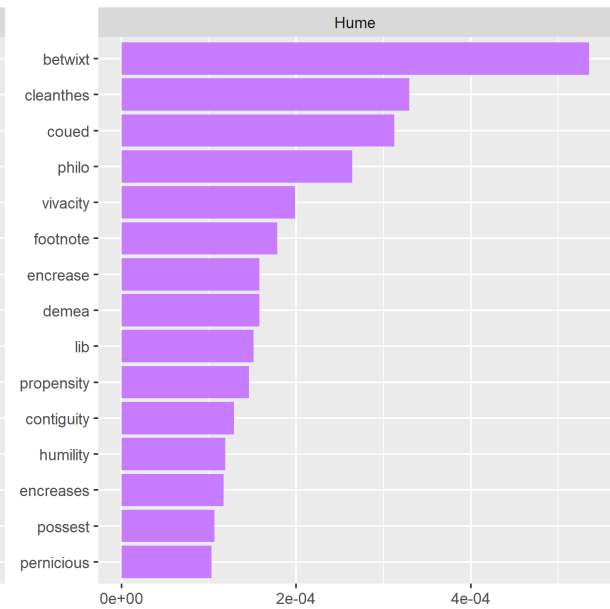
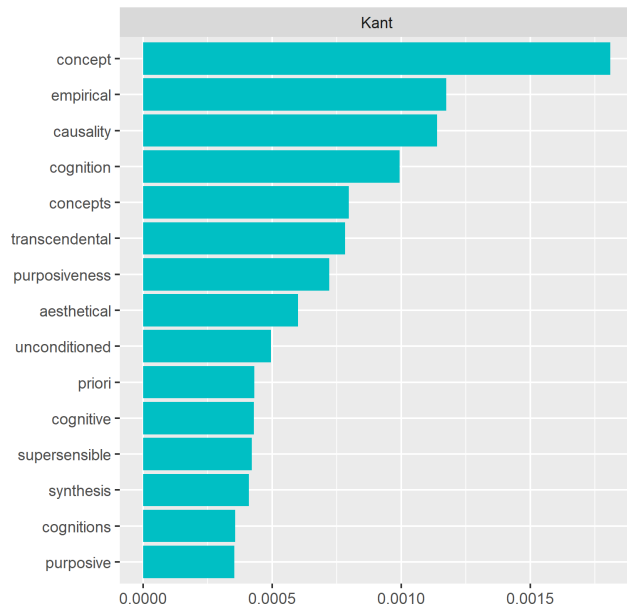
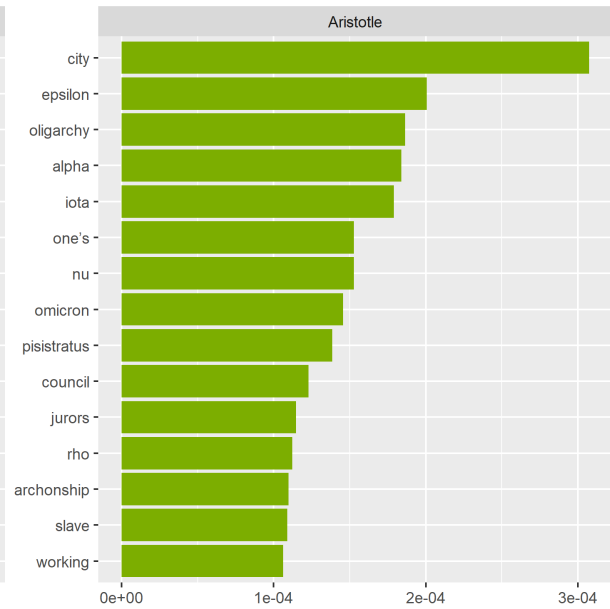
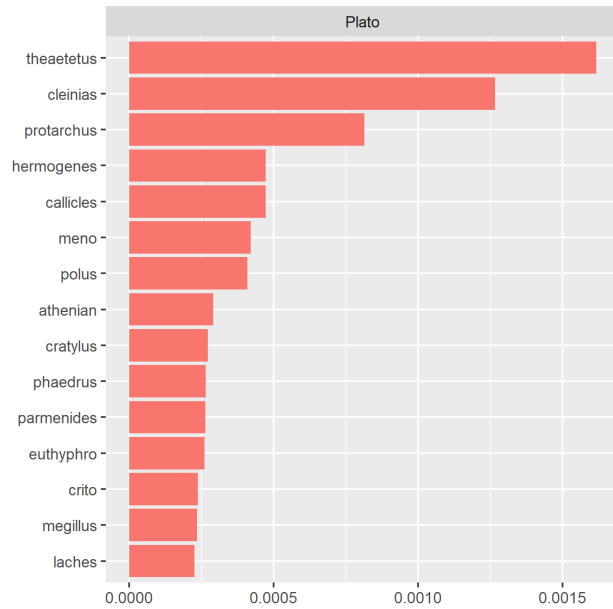
Next, we repeat the above step but this time across all authors. We see that Plato is mostly identified by names of characters as expected, Aristotle seems to be heavily identified by his use of greek letters, Kant looks to be identified by more broad philosophical ideas. It’s difficult to say how exactly Hume is identified but the rest of the terms associated with these authors makes sense.

¹<https://www.gutenberg.org/>

²More on tf-idf can be found here <https://www.tidytextmining.com/tfidf.html>



tf-idf



tf-idf

Supervised learning model

For our supervised learning model, we employ the use of a multinomial regression model with lasso regularization using the `glmnet` package in `R`. The tokens and their `tf-idfs` represent the features of our data. This method is well suited for text prediction as given the very high dimensional space of our features, the lasso regularization allows for the penalization of features (down to zero at times) and so the final model will only include features (tokens) that the algorithm considers to be key to prediction ³.

First, we split our data into a training and test set and examine the raw counts of lines by author. From the table below, we see that we do not have balance as Plato is overly represented in our training sample. To address this, authors with the majority of the lines are down-sampled after our text-preprocessing steps in order to achieve a balanced sample of authors to potentially improve our prediction. We also use a max of 1600 tokens ⁴ We explore different values for our penalty hyper-parameter and we use the one that returns the best results in terms of prediction accuracy as our final model. In dealing with the random variation in our train/test splits, we use `k-fold` cross validation with 10 folds.

```
## # A tibble: 4 x 2
##       n author
##   <int> <chr>
## 1 93319 Plato
## 2 41364 Kant
## 3 31689 Hume
## 4 23183 Aristotle
```

The table below reports the “best” model across a range of penalties. We see that our best model has an accuracy of about 67 percent. This is not “awesome” in terms of accuracy but is fairly reasonable. We are attempting to predict the author line by line within the same subject area of philosophy without any other predictors other than the text itself which is a fairly challenging task. There are also the aforementioned challenges of the connections that exist between the authors making the classification task even more difficult.

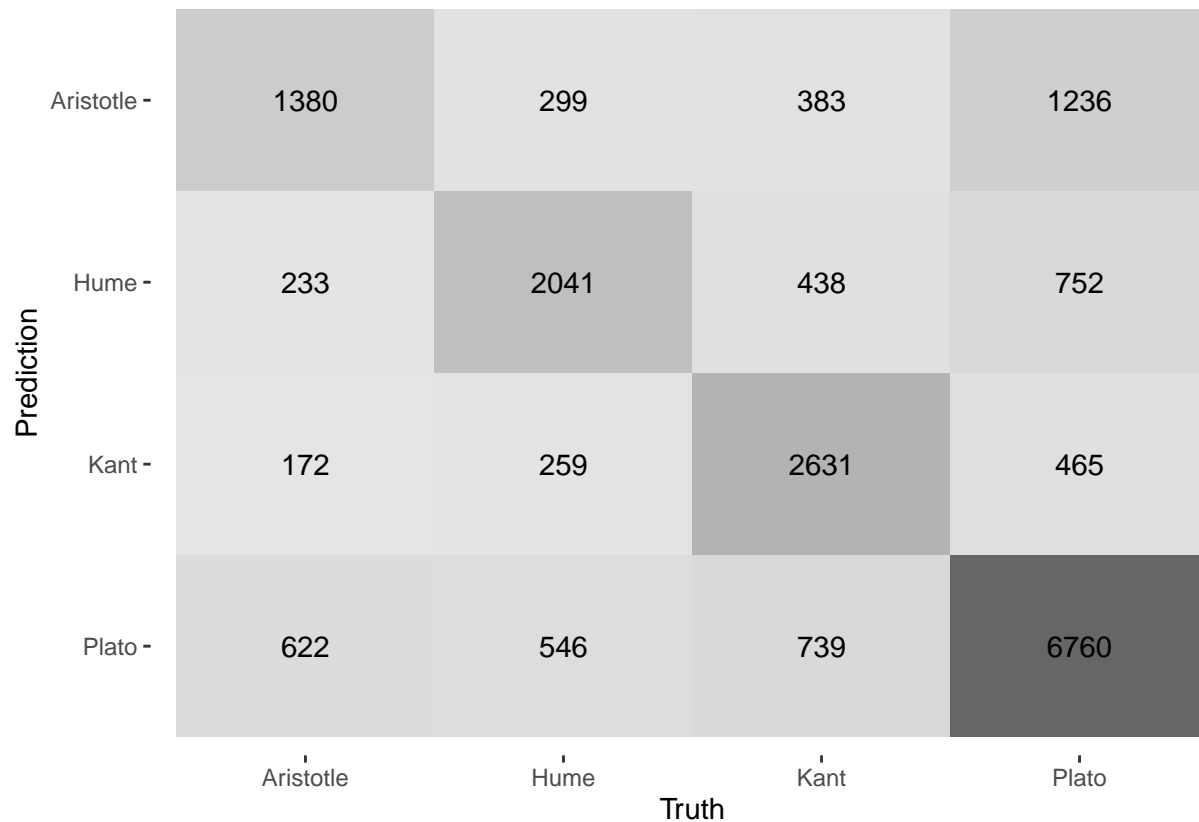
```
## # A tibble: 5 x 7
##   penalty .metric .estimator mean      n std_err .config
##   <dbl> <chr>      <chr>      <dbl> <int>    <dbl> <chr>
## 1 0.000464 accuracy multiclass 0.673     10 0.000977 Preprocessor1_Model104
## 2 0.000129 accuracy multiclass 0.673     10 0.00100  Preprocessor1_Model103
```

³More on lasso can be found here [https://en.wikipedia.org/wiki/Lasso_\(statistics\)](https://en.wikipedia.org/wiki/Lasso_(statistics))

⁴This is the largest value that maximizes predictive power with a reasonable computing time.

```
## 3 0.0000359 accuracy multiclass 0.673    10 0.00113 Preprocessor1_Model102
## 4 0.00001   accuracy multiclass 0.673    10 0.00117 Preprocessor1_Model101
## 5 0.00167   accuracy multiclass 0.664    10 0.00121 Preprocessor1_Model105
```

The confusion matrix below for the first fold shows exactly how the model classifies and mis-classifies. We see that the diagonal is well populated which is good. The model seems to do best at classifying Plato. This makes sense because as previously discussed, Plato uses the name of characters a lot in his works that would not be used by another author making it easier to identify that a line was written by him. With misclassification, we see that Plato is often mistaken for Aristotle which again makes is reasonable as Plato was Aristotle's teacher and so naturally there would be a transfer of ideas between their works.



Lastly, we evaluate our model on the test set. We have an accuracy of 66.7% which is very close to our in-sample accuracy.

```
## # A tibble: 2 x 4
##   .metric .estimator .estimate .config
##   <chr>   <chr>       <dbl> <chr>
## 1 accuracy multiclass    0.667 Preprocessor1_Model1
## 2 roc_auc  hand_till      0.860 Preprocessor1_Model1
```

Let's check and see which lines the model seems to do best at in the test set.

Best predicted lines

```
## # A tibble: 8 x 4
##   author      text                                title                                value
##   <chr>      <chr>                                <chr>                                <dbl>
## 1 Aristotle Self-Control.                    The Ethics of Aristotle              1.00
## 2 Aristotle Choice.                          The Ethics of Aristotle              1.00
## 3 Hume       fancy.                                A Treatise of Human Nature          1.00
## 4 Hume       considerable resemblance?                Dialogues Concerning Natu~          1.00
## 5 Kant       conditioned.                            The Critique of Pure Reas~          1.00
## 6 Kant       Judgement.                              Kant's Critique of Judgem~          1.00
## 7 Plato      CEPHALUS - SOCRATES - POLEMARCHUS        The Republic                        1.00
## 8 Plato      SOCRATES - CLEITOPHON - POLEMARCHU~    The Republic                        1.00
```

The lines are pretty short and most are even single words but the predictions make sense. As we see with Plato, lines with the character names are almost guaranteed to be written by him. Self-control is also a big theme within Aristotle work's as he was the one who heavily developed the idea of the golden-mean.

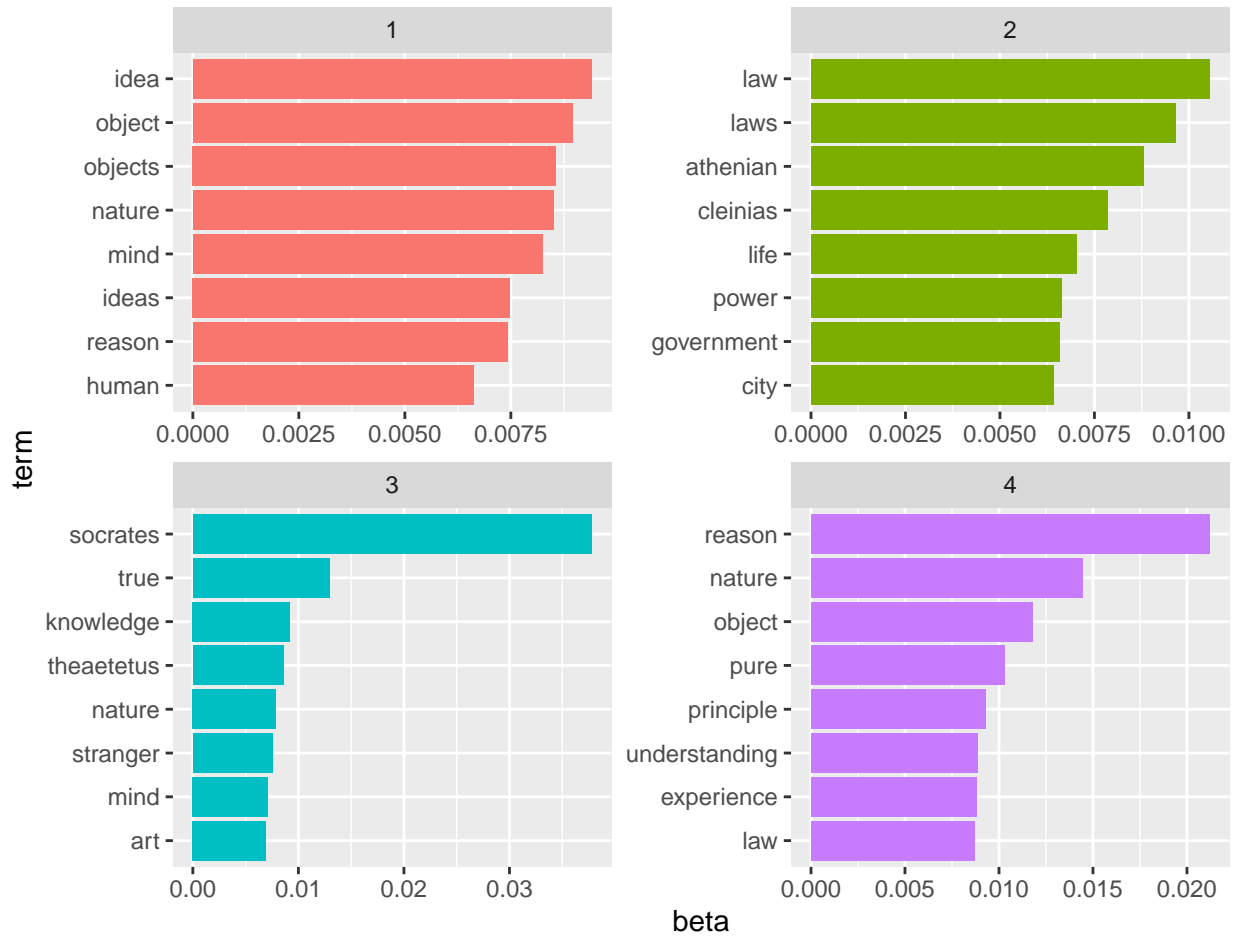
Unsupervised learning model

One of the most used models in text mining to analyze series of unlabeled, or even labeled, documents is topic models. Topic modeling is an unsupervised machine learning that “scans” these documents and finds natural cluster of words or phrases called topics. For example, say that we have two documents: an article about cars and an article about groceries. The common words in groceries would be “carrots”, “meat” and “milk” whereas the common words in cars would be “gas”, “horsepower” and “exhaust”. Consequently, each of these words would be respectively grouped together by the model creating two clusters. This might not seem

useful when we only consider two documents, but it becomes extremely useful when we have hundreds or more documents and we want to cluster them broadly on areas of interest. In fact, this process is most likely used by major news networks or social media outlets. For example, say that someone logs in to their New York Times (NYT) account and primarily reads articles related to international politics or economics. The NYT will then use this data to recommend other articles that have the international politics or economics topic. However, the question remains on how to create such a model.

The main approach that is used is the Latent Dirichlet Allocation (LDA) method. This method assumes two things: similar topics use similar words and documents are a mixture of several topics. In the example above with the grocery and car articles, there are shared words between the two topics such as “budget”, “price” and “environment”. The advantage of using the LDA is that the documents that are studied can “overlap” with each other in terms of word or phrase usage which reflects the use of natural language. Since we are dealing with philosophical works, we decided to use the LDA as we want to allow for the possibility that the philosophers might use the same words or phrases in their works. The data preparation process is very similar to the supervised model, in which we break down our texts into tokens representing a particular word. The one difference that we make in the unsupervised learning model is that we are removing any words that appear less than 50 times in our tokenized dataset. The reason is that we want to see how well the model performs when there is a significant amount of overlap between the various works. This is because we believe that the words that are used less than 50 times are unique to a specific author, thus creating a clear distinction between them and the other authors.

To perform the LDA method in R, we are using the LDA command in the `topicmodels` package. This command highlights one of the key “issues” of topic modeling: the need to specify a number of topics. In the case of major news networks, one can see the various topics that describe an article (health, business, politics, etc. . .) that have been decided by these networks. The topic models do not inherently give names to the topics they simply cluster the documents based on the number of cluster points, i.e. topics, specified. However, since we know that our documents have been written by four authors, we specify to our model that there are four topics to cluster around.



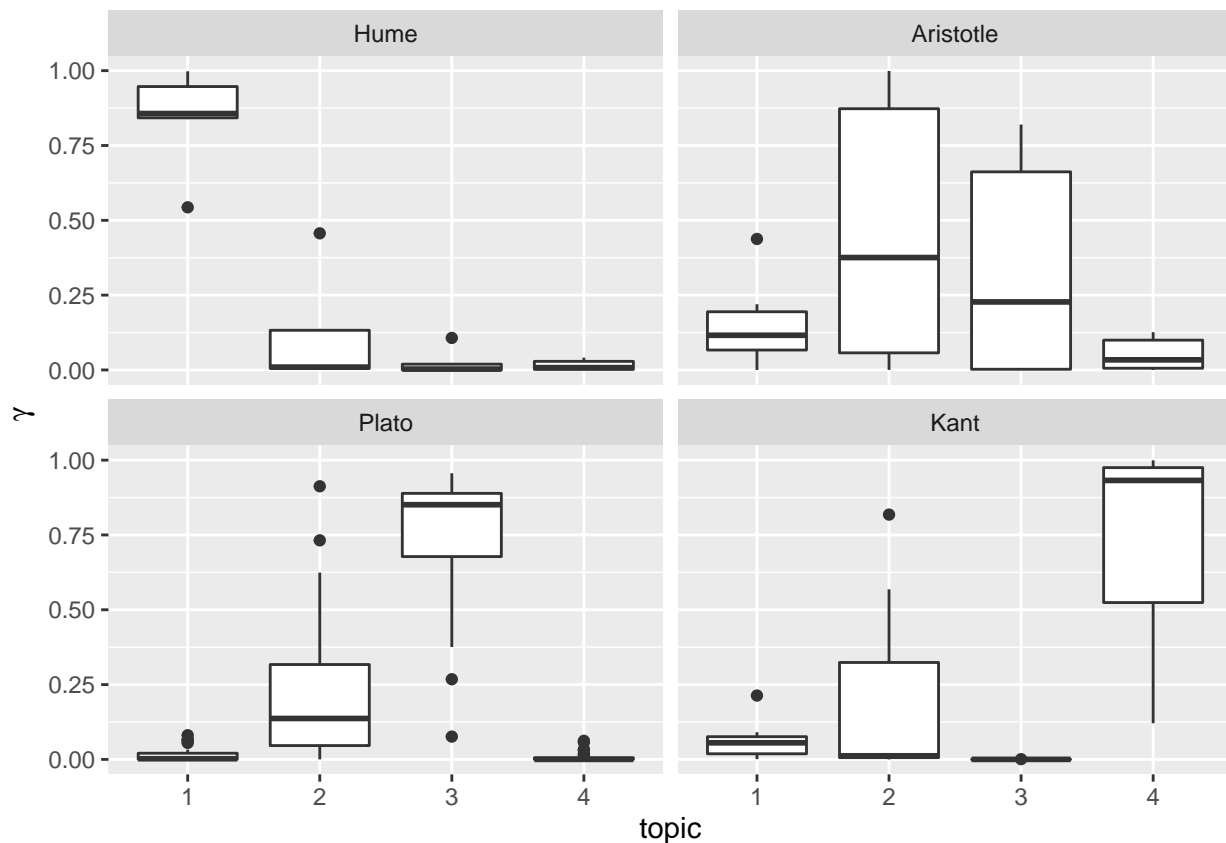
As we can see in the graph above, the issue of needing to specify the number and the name of topics is clear. The graph allows us to infer which topic belongs to which author, but it does not directly give us the names of the authors associated with each topic. We can conclude that topic 2 and 3 are associated with the Greek Philosophers due to the words Athenian, Cleinias, Socrates and Theaetetus, therefore leaving topics 1 and 4 to the more modern philosophers Hume and Kant. However, we are unable to confidently say which topic is specifically associated to a particular author. From term-frequency inverse document frequency within Plato's works, the words that most uniquely identify Plato's works are name of characters. Since topic 3 has two names that are mentioned, and that are in the top four most frequent words, we can conclude that topic 3 refers to Plato, and topic 2 to Aristotle. We believe that topic 1 represents Hume because we are somewhat familiar with Kant's philosophy about nature and principles, words that are found in topic 4.

```
## # A tibble: 172 x 4
```

```
##   Author    Title                                topic    gamma
##   <chr>    <chr>                                <int>    <dbl>
```

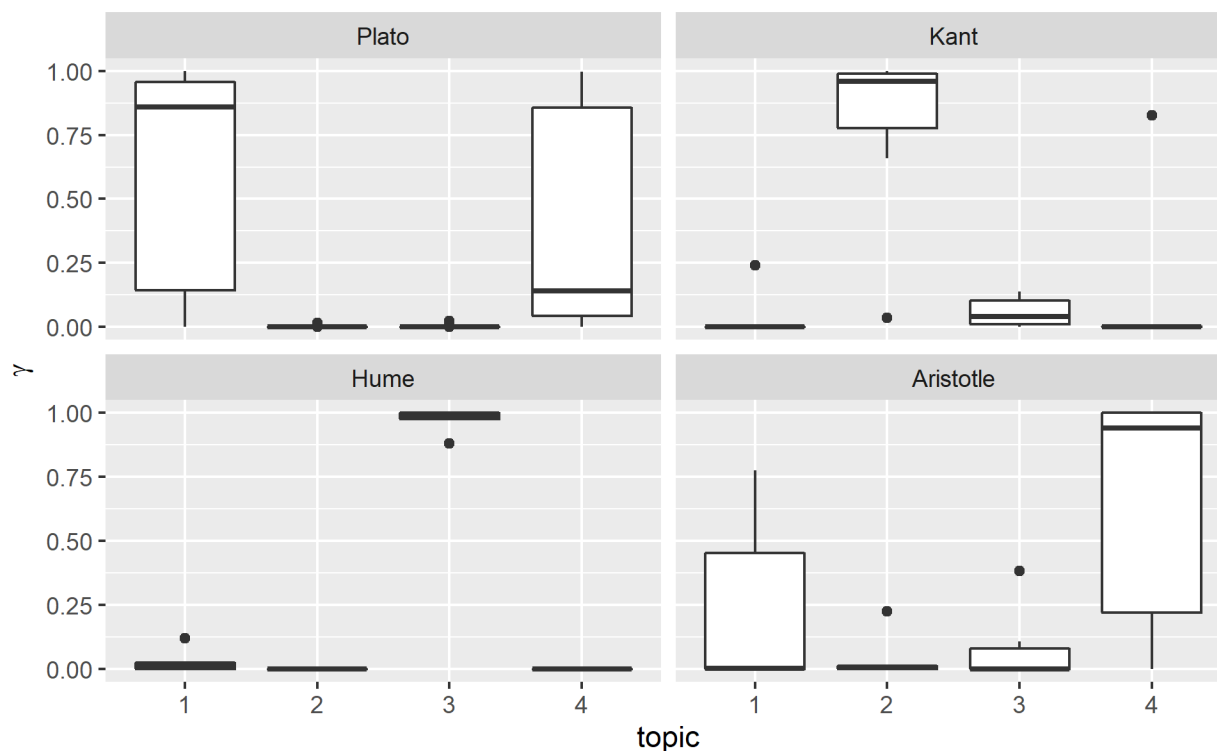
```
## 1 Aristotle Aristotle on the art of poetry          1 0.112
## 2 Aristotle Politics: A Treatise on Government      1 0.0513
## 3 Aristotle The Athenian Constitution              1 0.0000304
## 4 Aristotle The Categories                         1 0.438
## 5 Aristotle The Ethics of Aristotle                1 0.220
## 6 Aristotle The Poetics of Aristotle               1 0.119
## 7 Hume      A Treatise of Human Nature              1 0.998
## 8 Hume      An Enquiry Concerning Human Understanding 1 0.947
## 9 Hume      An Enquiry Concerning the Principles of Morals 1 0.857
## 10 Hume     Dialogues Concerning Natural Religion    1 0.842
## # ... with 162 more rows
```

However, this is not enough. We are interested in seeing the document per topic probability meaning that we want to see what is the probability that a document is situated in a particular topic. For example, Aristotle's book "Aristotle on the Art of Poetry" has an 11% chance of coming from topic 1, which we assume to be Hume. In order to get a better idea, we look at the boxplot that represents the probability, denoted as γ , that the author is in the respective topic.



Unfortunately, our model does not have a very good predictive capability, although it is important to note that we removed words that were used less than 50 times, which should affect the results. Considering this limitation, the model does a relatively good job at distinguishing works written by Hume and Kant from work written in a different era, written by Aristotle and Plato. Furthermore, Hume’s works, topic 1, have almost been perfectly associated with being written by Hume. Kant’s association with topic 4 is also high. However, the model is terrible at distinguishing works between Plato and Aristotle. Both authors have written in the same language and have both been translated. As mentioned earlier, translations often contain the writing mannerisms of the translator, meaning that translated works might share similar writing patterns as they might be translated by the same translator. In fact, Plato’s works have been mostly attributed to Aristotle as an author, as shown by the table in the appendix, possibly because Aristotle was Plato’s student.

This seems to be supported by the figure below where we reran LDA model without excluding the words that have appeared 50 times or less. As we can see, both the prediction accuracy for Hume and Kant have significantly increased, whereas the predictions for Plato and Aristotle have marginally gotten better. In fact the medians are closer to where they should be. Closer to 1 if the topic is associated with the author and closer to 0 when the topic is not associated with the author. However, there is a big range in the probabilities between the first and third quartiles. However, our findings with the unsupervised models are consistent with the results of the supervised models.



Conclusion

Although we used a supervised and unsupervised models, it is very interesting that we reach the same conclusion: Plato's works are more likely to be attributed to Aristotle. One possible reason is that there is an issue of the works being translated by the same author. The second reason is that Plato's works are in fact similar to Aristotle's and it would be difficult to distinguish them without a neural network. To verify if there is an issue of translation, we would suggest considering other books from the Classical Era that have also been translated into English, assuring that there is no connection between the authors. If our models give similar results, where they confuse one author for another, then there is a translation issue. However, if the models are able to make a clear distinction between the two, then only logical explanation is that the relationship between Aristotle and Plato is creating issues for our models.

Disregarding this issue, it seems that the our models have a relatively high change of predicting the author of philosophical works. Since philosophy tends to build on itself, therefore adding a level of complexity, we are certain that our models can be applicable to other genres of literature and with more than four authors.

Appendix

The table below shows the which authors have misidentified by the model. The Author column indicates the correct author and the consensus column indicates what the model predicted the author to be. As we can see, three of Aristotle's work have been attributed to Plato, two of Kant's works have been attributed to Aristotle. The remaining rows are all the work written by Plato that have mis-attributed to Aristotle.

```
## # A tibble: 30 x 5
##   Author Title topic gamma consensus
##   <chr> <chr> <int> <dbl> <chr>
## 1 Aristot~ "Aristotle on the art of poetry" 3 0.734 Plato
## 2 Aristot~ "The Categories" 3 0.445 Plato
## 3 Aristot~ "The Poetics of Aristotle" 3 0.820 Plato
## 4 Kant "Of the Injustice of Counterfeiting Books\r\n~ 2 0.818 Aristotle
## 5 Kant "Perpetual Peace\nA Philosophical Essay" 2 0.568 Aristotle
## 6 Plato "Apology" 3 0.501 Aristotle
## 7 Plato "Charmides" 3 0.902 Aristotle
## 8 Plato "Cratylus" 3 0.946 Aristotle
## 9 Plato "Critias" 2 0.732 Aristotle
## 10 Plato "Crito" 3 0.561 Aristotle
## # ... with 20 more rows
```

Resources used

CRAN for gutenbergr

<https://cran.r-project.org/web/packages/gutenbergr/vignettes/intro.html>

Jane Austen vs Wells classification

<https://julasilge.com/blog/tidy-text-classification/>

Federalist papers classification

<https://www.hvitfeldt.me/blog/predicting-authorship-in-the-federalist-papers-with-tidymodels/>

<https://www.hvitfeldt.me/blog/authorship-classification-with-tidymodels-and-textrecipes/>

The Office classification

<https://www.hvitfeldt.me/blog/tidytuesday-pos-textrecipes-the-office/>

Term frequency guide

<https://www.tidytextmining.com/tfidf.html>

Topic modeling

<https://www.tidytextmining.com/topicmodeling.html>

Multiclass classification

<https://smltar.com/mlclassification.html#mlmulticlass>