# CSC 466 Lab 6 Report:
# Information Retrieval and Text Mining

Martin Solomon Hsu: mshsu@calpoly.edu
Rachel Roggenkemper: rroggenk@calpoly.edu
Instructor: Dr. Alexander Dekhtyar
CSC 466: Knowledge Discovery from Data, Fall 2023

### Abstract

In this lab, we will investigate authorship attribution in the Reuter 50-50 dataset using K-Nearest Neighbors (KNN) and Random Forest classifiers. After rigorous tuning and testing, the KNN was implemented with k = 1, while the Random Forest used 10 trees, 1000 word attributes, a sample size of 1000, and a threshold of 0.2. The study primarily evaluated precision, recall, and F1 scores for individual authors and overall model accuracy. The analysis revealed differences in the predictability of authors and the challenges of distinguishing authors with similar styles. Results showed that KNN significantly outperformed Random Forest, achieving 78.9% accuracy compared to 65.1%.

## I. Introduction

This lab report delves into the exploration of authorship attribution within the context of the Reuter 50-50 dataset, a diverse collection of news stories authored by 50 different writers. We will create vector space representations of the documents from this dataset using both stemming and stopword removal, and will use these representations to conduct a text mining study. Our investigation focuses on the application and efficacy of two machine learning algorithms: the K-Nearest Neighbors (KNN) and the Random Forest classifier. Each algorithm is meticulously tuned, tested, and examined to understand its ability to reliably establish the authorship of the text documents in the dataset.

## II. Dataset Description

We utilize the Reuter 50-50 dataset, which is a collection of text documents. The dataset consists of a collection of news stories published by the Reuters news agencies.The dataset was constructed to study machine learning algorithms for authorship attribution. It consists of a selection of 50 authors who published news stories with Reuters. For each author, exactly 100 news stories they authored are placed in the dataset.

# III. Methods

## Text Vectorization

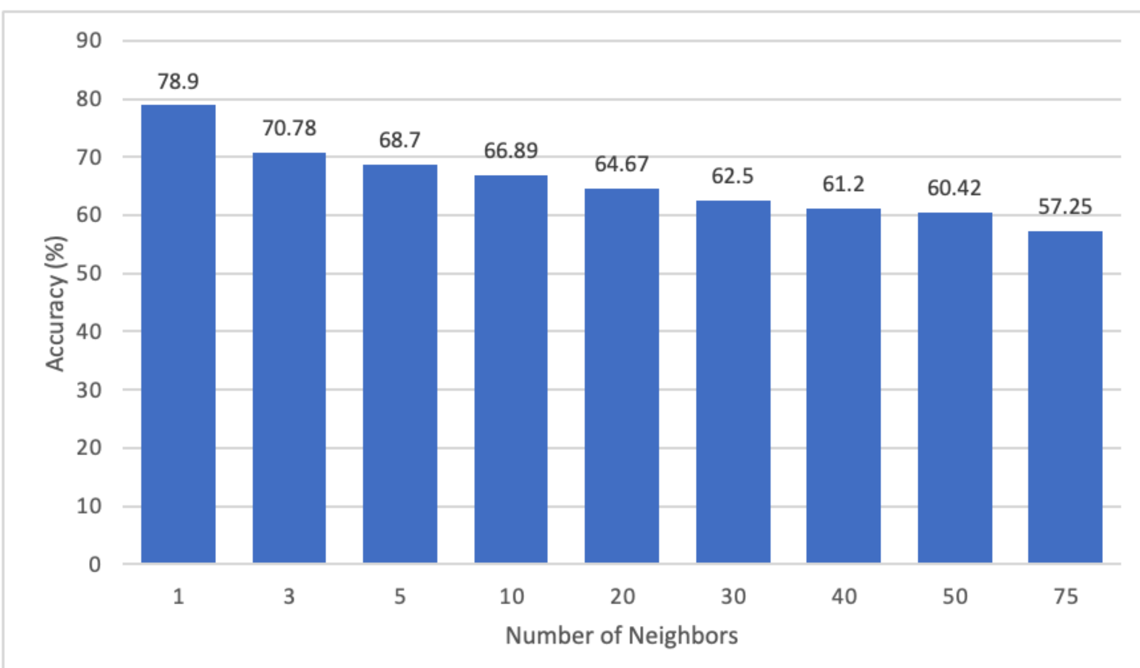
The first step of our text vectorization process was employing the TF-IDF (Term Frequency-Inverse Document Frequency) technique, emphasizing words unique to each document by accounting for their frequency and inverse document occurrence. Our preprocessing included the removal of stop words to eliminate common, less significant words, and the application of the Porter Stemmer algorithm for reducing words to their root forms, enhancing analytical efficiency. We calculated both cosine similarity and the Okapi metric for similarity measures but opted for cosine similarity due to observing higher effectiveness in capturing content similarity between document vectors. Notably, we refrained from thresholding, avoiding the exclusion of words based on their frequency to ensure a comprehensive analysis.

We examined the following classifiers with a leave-one-out methodology to determine if they can help us establish the authorship of the articles:

## A. K-Nearest Neighbors Classifier

The K-Nearest Neighbors (KNN) algorithm is a straightforward yet effective method for classification tasks. It operates on the principle of proximity in feature space, classifying a new data point based on the majority class among its k nearest neighbors. In our experiment, we implemented cosine similarity as our distance metric to measure the closeness of data points. The primary hyperparameter investigated in this method was the number of neighbors: k.

*Figure 1: K-Nearest Neighbors Parameter Tuning Results*



We investigated the impact of the number of neighbors on the classification accuracy as can be seen in Figure 1. The accuracy peaks at 78.9% when k = 1, suggesting that the closest neighbor offers the most significant predictive power in this dataset. As the number of neighbors increases, a general decline in accuracy is observed, reaching a low of 57.25% at k = 75. This trend indicates that the addition of more neighbors dilutes the relevance of the nearest points, potentially including neighbors from other classes that do not contribute positively to the decision-making process. Thus, after rigorous tuning and testing, we ultimately chose k = 1 for our analysis, focusing on the nearest neighbor to each test data point for class assignment.

**C. Random Forest Classifier**

Random Forests are an advanced ensemble learning technique based on decision trees. By building multiple decision trees on different subsets of the dataset and averaging their predictions, Random Forests aim to improve predictive accuracy and control overfitting. In our analysis, the Random Forest classifier was constructed with specific attention to several hyperparameters: the number of trees in the forest, the number of word attributes, the sample size for tree building, and the decision threshold. We fit the random forest classifier on the TF-IDF matrix, where each attribute is a word and each observation is a document. The class variable is the author. The primary hyperparameters investigated in this method were the number of trees in the forest, the number of word attributes, the sample size for tree building, and the decision threshold.

*Figure 2: Random Forest Parameter Tuning Results*

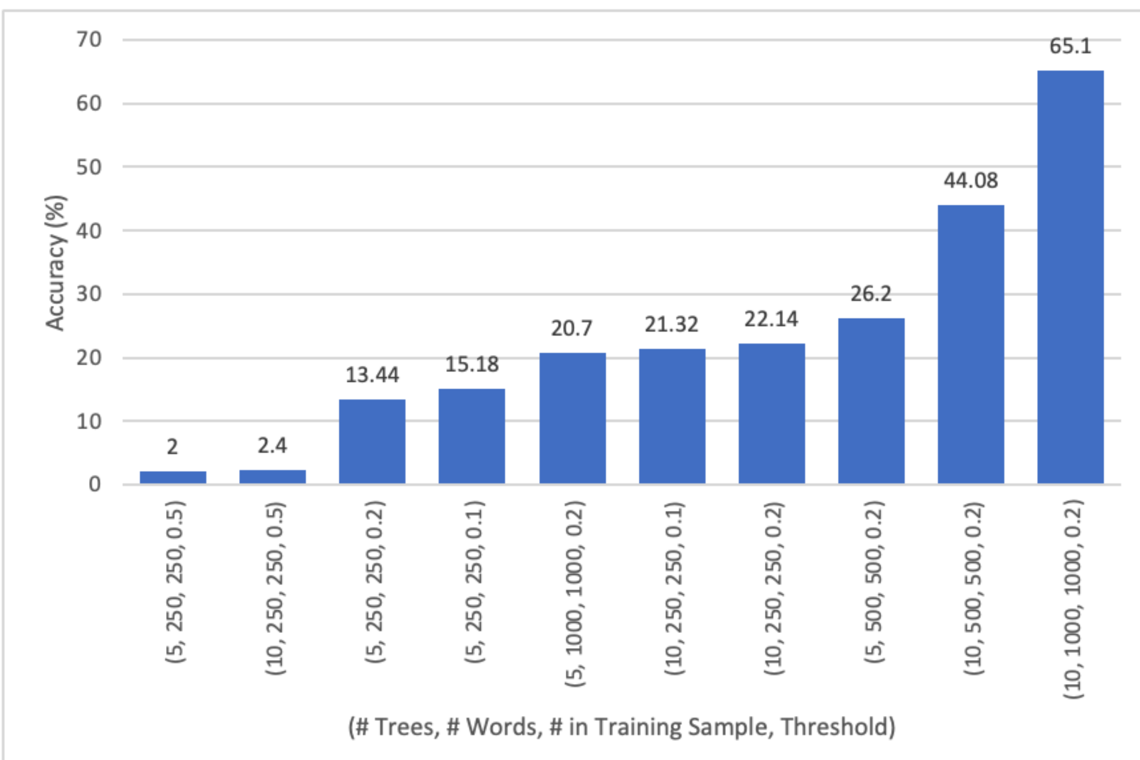


Figure 2 demonstrates the effects of varying the number of trees, the number of word attributes, the sample size, and the threshold for the random selection of features. The most successful configuration used 10 trees, with 1000 word attributes, a training sample size of 1000, and a threshold of 0.2, achieving an accuracy of 65.1%. The chart reveals that increasing the complexity of the model by using more word attributes and larger training samples enhances the classifier's performance. However, excessive complexity can lead to overfitting, as seen with smaller training sample sizes, which underscores the importance of balancing the model's complexity with the amount of available training data. Ultimately, after rigorous tuning and testing, we ultimately went with a configuration including 10 trees & 1000 word attributes & a sample size of 1000 & a threshold of 0.2.

## IV. Results

### A. K-Nearest Neighbors Classifier

The K-Nearest Neighbors Classifier, with k = 1, displayed strong performance in authorship attribution. The results, shown below in Table 1, demonstrate a high level of accuracy for most authors, with notable success in correctly classifying texts.

*Table 1: K-Nearest Neighbors Classifier Results*

| Author | Hits | Strikes | Misses | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| MatthewBunce | 100 | 3 | 0 | 0.970874 | 1 | 0.985222 |
| FumikoFujisaki | 98 | 4 | 2 | 0.960784 | 0.98 | 0.970297 |
| LynnleyBrowning | 98 | 6 | 2 | 0.942308 | 0.98 | 0.960784 |
| KarlPenhaul | 96 | 3 | 4 | 0.969697 | 0.96 | 0.964824 |
| RogerFillion | 94 | 10 | 6 | 0.903846 | 0.94 | 0.921569 |
| LynneO'Donnell | 91 | 24 | 9 | 0.791304 | 0.91 | 0.846512 |
| DarrenSchuettler | 90 | 17 | 10 | 0.841121 | 0.9 | 0.869565 |
| RobinSidel | 89 | 6 | 11 | 0.936842 | 0.89 | 0.912821 |
| JoWinterbottom | 89 | 24 | 11 | 0.787611 | 0.89 | 0.835681 |
| MichaelConnor | 89 | 8 | 11 | 0.917526 | 0.89 | 0.903553 |
| TimFarrand | 88 | 16 | 12 | 0.846154 | 0.88 | 0.862745 |
| JimGilchrist | 87 | 10 | 13 | 0.896907 | 0.87 | 0.883249 |
| HeatherScoffield | 87 | 10 | 13 | 0.896907 | 0.87 | 0.883249 |
| AlanCrosby | 87 | 19 | 13 | 0.820755 | 0.87 | 0.84466 |
| EdnaFernandes | 87 | 11 | 13 | 0.887755 | 0.87 | 0.878788 |
| PierreTran | 86 | 16 | 14 | 0.843137 | 0.86 | 0.851485 |
| AaronPressman | 86 | 20 | 14 | 0.811321 | 0.86 | 0.834951 |
| MarkBendeich | 85 | 22 | 15 | 0.794393 | 0.85 | 0.821256 |
| MarcelMichelson | 85 | 9 | 15 | 0.904255 | 0.85 | 0.876289 |
| ToddNissen | 85 | 13 | 15 | 0.867347 | 0.85 | 0.858586 |
| DavidLawder | 85 | 19 | 15 | 0.817308 | 0.85 | 0.833333 |
| KouroshKarimkhany | 84 | 32 | 16 | 0.724138 | 0.84 | 0.777778 |
| KeithWeir | 84 | 14 | 16 | 0.857143 | 0.84 | 0.848485 |
| JonathanBirt | 84 | 10 | 16 | 0.893617 | 0.84 | 0.865979 |
| SimonCowell | 82 | 9 | 18 | 0.901099 | 0.82 | 0.858639 |
| GrahamEarnshaw | 82 | 24 | 18 | 0.773585 | 0.82 | 0.796117 |
| JanLopatka | 81 | 15 | 19 | 0.84375 | 0.81 | 0.826531 |
| TheresePoletti | 81 | 33 | 19 | 0.710526 | 0.81 | 0.757009 |

| | | | | | | |
|---|---|---|---|---|---|---|
| LydiaZajc | 81 | 4 | 19 | 0.952941 | 0.81 | 0.875676 |
| JoeOrtiz | 81 | 26 | 19 | 0.757009 | 0.81 | 0.782609 |
| NickLouth | 78 | 16 | 22 | 0.829787 | 0.78 | 0.804124 |
| KevinDrawbaugh | 77 | 17 | 23 | 0.819149 | 0.77 | 0.793814 |
| PatriciaCommins | 77 | 9 | 23 | 0.895349 | 0.77 | 0.827957 |
| KirstinRidley | 77 | 24 | 23 | 0.762376 | 0.77 | 0.766169 |
| MartinWolk | 76 | 18 | 24 | 0.808511 | 0.76 | 0.783505 |
| JohnMastrini | 76 | 23 | 24 | 0.767677 | 0.76 | 0.763819 |
| KevinMorrison | 76 | 29 | 24 | 0.72381 | 0.76 | 0.741463 |
| BradDorfman | 75 | 24 | 25 | 0.757576 | 0.75 | 0.753769 |
| BernardHickey | 70 | 18 | 30 | 0.795455 | 0.7 | 0.744681 |
| PeterHumphrey | 68 | 46 | 32 | 0.596491 | 0.68 | 0.635514 |
| EricAuchard | 68 | 35 | 32 | 0.660194 | 0.68 | 0.669951 |
| SarahDavison | 66 | 20 | 34 | 0.767442 | 0.66 | 0.709677 |
| AlexanderSmith | 65 | 23 | 35 | 0.738636 | 0.65 | 0.691489 |
| SamuelPerry | 65 | 27 | 35 | 0.706522 | 0.65 | 0.677083 |
| TanEeLyn | 61 | 37 | 39 | 0.622449 | 0.61 | 0.616162 |
| BenjaminKangLim | 58 | 58 | 42 | 0.5 | 0.58 | 0.537037 |
| JaneMacartney | 58 | 49 | 42 | 0.542056 | 0.58 | 0.560386 |
| WilliamKazer | 46 | 46 | 54 | 0.5 | 0.46 | 0.479167 |
| MureDickie | 45 | 46 | 55 | 0.494505 | 0.45 | 0.471204 |
| ScottHillis | 42 | 52 | 58 | 0.446809 | 0.42 | 0.43299 |

A key takeaway is the overall accuracy of 78.9%, with 3946 correctly classified and 1054 incorrectly classified instances. This excellent performance highlights the effectiveness of the KNN classifier in accurately identifying authorship.

**B. Random Forest Classifier**

The Random Forest Classifier, set with 10 trees, 1000 word attributes, a sample size of 1000, and a threshold of 0.2, showed a varied performance across different authors. As displayed in Table 2 below, the model's effectiveness fluctuated significantly, as indicated by the precision, recall, and F1 score for each author.

*Table 2: Random Forest Classifier Results*

| Author | Hits | Strikes | Misses | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| JohnMastrini | 89 | 285 | 11 | 0.237968 | 0.89 | 0.375527 |
| DavidLawder | 88 | 51 | 12 | 0.633094 | 0.88 | 0.736402 |
| TheresePoletti | 84 | 146 | 16 | 0.365217 | 0.84 | 0.509091 |
| JoWinterbottom | 82 | 29 | 18 | 0.738739 | 0.82 | 0.777251 |
| GrahamEarnshaw | 81 | 40 | 19 | 0.669421 | 0.81 | 0.733032 |
| AlanCrosby | 79 | 63 | 21 | 0.556338 | 0.79 | 0.652893 |
| KirstinRidley | 77 | 24 | 23 | 0.762376 | 0.77 | 0.766169 |
| JimGilchrist | 77 | 26 | 23 | 0.747573 | 0.77 | 0.758621 |
| BernardHickey | 76 | 67 | 24 | 0.531469 | 0.76 | 0.625514 |
| LynneO'Donnell | 76 | 17 | 24 | 0.817204 | 0.76 | 0.787565 |
| AlexanderSmith | 75 | 77 | 25 | 0.493421 | 0.75 | 0.595238 |
| MarcelMichelson | 75 | 17 | 25 | 0.815217 | 0.75 | 0.78125 |
| AaronPressman | 73 | 76 | 27 | 0.489933 | 0.73 | 0.586345 |
| BradDorfman | 72 | 59 | 28 | 0.549618 | 0.72 | 0.623377 |
| NickLouth | 71 | 35 | 29 | 0.669811 | 0.71 | 0.68932 |
| MatthewBunce | 70 | 3 | 30 | 0.958904 | 0.7 | 0.809249 |

| | | | | | | |
|---|---|---|---|---|---|---|
| EdnaFernandes | 70 | 56 | 30 | 0.555556 | 0.7 | 0.619469 |
| PierreTran | 69 | 21 | 31 | 0.766667 | 0.69 | 0.726316 |
| LydiaZajc | 69 | 12 | 31 | 0.851852 | 0.69 | 0.762431 |
| KevinMorrison | 68 | 19 | 32 | 0.781609 | 0.68 | 0.727273 |
| FumikoFujisaki | 67 | 39 | 33 | 0.632075 | 0.67 | 0.650485 |
| RobinSidel | 67 | 4 | 33 | 0.943662 | 0.67 | 0.783626 |
| PeterHumphrey | 67 | 16 | 33 | 0.807229 | 0.67 | 0.73224 |
| JonathanBirt | 66 | 25 | 34 | 0.725275 | 0.66 | 0.691099 |
| MarkBendeich | 65 | 22 | 35 | 0.747126 | 0.65 | 0.695187 |
| KouroshKarimkhany | 65 | 21 | 35 | 0.755814 | 0.65 | 0.698925 |
| BenjaminKangLim | 65 | 55 | 35 | 0.541667 | 0.65 | 0.590909 |
| JoeOrtiz | 65 | 30 | 35 | 0.684211 | 0.65 | 0.666667 |
| LynnleyBrowning | 64 | 11 | 36 | 0.853333 | 0.64 | 0.731429 |
| TanEeLyn | 63 | 50 | 37 | 0.557522 | 0.63 | 0.591549 |
| TimFarrand | 62 | 17 | 38 | 0.78481 | 0.62 | 0.692737 |
| KeithWeir | 62 | 10 | 38 | 0.861111 | 0.62 | 0.72093 |
| DarrenSchuettler | 62 | 35 | 38 | 0.639175 | 0.62 | 0.629442 |
| HeatherScoffield | 61 | 21 | 39 | 0.743902 | 0.61 | 0.67033 |
| MartinWolk | 60 | 14 | 40 | 0.810811 | 0.6 | 0.689655 |
| SimonCowell | 59 | 14 | 41 | 0.808219 | 0.59 | 0.682081 |
| RogerFillion | 59 | 6 | 41 | 0.907692 | 0.59 | 0.715152 |
| JaneMacartney | 59 | 40 | 41 | 0.59596 | 0.59 | 0.592965 |
| MichaelConnor | 58 | 8 | 42 | 0.878788 | 0.58 | 0.698795 |
| PatriciaCommins | 58 | 16 | 42 | 0.783784 | 0.58 | 0.666667 |
| EricAuchard | 57 | 35 | 43 | 0.619565 | 0.57 | 0.59375 |
| JanLopatka | 57 | 24 | 43 | 0.703704 | 0.57 | 0.629834 |
| ToddNissen | 56 | 9 | 44 | 0.861538 | 0.56 | 0.678788 |
| SarahDavison | 53 | 13 | 47 | 0.80303 | 0.53 | 0.638554 |
| ScottHillis | 49 | 20 | 51 | 0.710145 | 0.49 | 0.579882 |
| KevinDrawbaugh | 44 | 8 | 56 | 0.846154 | 0.44 | 0.578947 |
| MureDickie | 44 | 15 | 56 | 0.745763 | 0.44 | 0.553459 |
| KarlPenhaul | 43 | 14 | 57 | 0.754386 | 0.43 | 0.547771 |
| SamuelPerry | 42 | 17 | 58 | 0.711864 | 0.42 | 0.528302 |
| WilliamKazer | 36 | 12 | 64 | 0.75 | 0.36 | 0.486486 |

The overall accuracy was 65.1%, with 3256 correctly classified and 1744 incorrectly classified instances. The Random Forest model demonstrated a moderate level of accuracy. However, the precision for several authors was notably lower compared to the KNN classifier, reflecting the challenges in using Random Forest for this specific task of authorship attribution.

## V. Reflection on Methods

The exploration of authorship attribution using the K-Nearest Neighbors (KNN) and Random Forest classifiers not only highlighted the overall efficacy of these algorithms but also shed light on the varying predictability of different authors. The KNN classifier, particularly effective at k = 1, revealed that authors like Matthew Bunce, Fumiko Fujisaki, and Lynnley Browning were among the easiest to predict. Their high precision and recall rates suggest a distinct and consistent writing style that the KNN algorithm could readily identify.

Conversely, authors such as Scott Hillis, William Kazer, and Mure Dickie posed more significant challenges, as indicated by their lower precision and recall scores. This difficulty might be attributed to less distinctive writing styles or greater similarity in their use of language to other authors in the dataset.

Interestingly, the study also revealed instances of authors being frequently confused with each other, particularly in the case of the Random Forest classifier. Authors like Jane Macartney and Benjamin Kang Lim, who had lower precision rates, were often mistaken for other authors, implying a shared stylistic that the classifier could not differentiate.

## VI. Conclusions

This investigation into authorship attribution using K-Nearest Neighbors and Random Forest classifiers revealed significant insights into the performance of these algorithms and the predictability of individual authors. The K-Nearest Neighbors model, with its high accuracy, demonstrated its aptitude in discerning distinct authorial styles, easily predicting authors with unique writing styles, while struggling with those whose styles are less distinctive or similar to others. In contrast, the Random Forest's varied performance highlighted the complexities involved in such models for tasks requiring nuanced differentiation between classes. Ultimately, utilizing the K-Nearest Neighbors classifier, specifically with k = 1, proved to be more accurate compared to using Random Forest classifier.

## VII. Appendix

### A. K-Nearest Neighbors Classifier (k = 1)

```
Results Summary:
File: data_knn_results.csv

By Author:
             Author  Hits  Strikes  Misses  Precision  Recall        F1
       MatthewBunce   100        3       0   0.970874    1.00 0.985222
     FumikoFujisaki    98        4       2   0.960784    0.98 0.970297
    LynnleyBrowning    98        6       2   0.942308    0.98 0.960784
        KarlPenhaul    96        3       4   0.969697    0.96 0.964824
       RogerFillion    94       10       6   0.903846    0.94 0.921569
     LynneO'Donnell    91       24       9   0.791304    0.91 0.846512
  DarrenSchuettler    90       17      10   0.841121    0.90 0.869565
        RobinSidel    89        6      11   0.936842    0.89 0.912821
     JoWinterbottom    89       24      11   0.787611    0.89 0.835681
      MichaelConnor    89        8      11   0.917526    0.89 0.903553
         TimFarrand    88       16      12   0.846154    0.88 0.862745
       JimGilchrist    87       10      13   0.896907    0.87 0.883249
  HeatherScoffield    87       10      13   0.896907    0.87 0.883249
        AlanCrosby    87       19      13   0.820755    0.87 0.844660
```

| | | | | | | |
|---|---|---|---|---|---|---|
| EdnaFernandes | 87 | 11 | 13 | 0.887755 | 0.87 | 0.878788 |
| PierreTran | 86 | 16 | 14 | 0.843137 | 0.86 | 0.851485 |
| AaronPressman | 86 | 20 | 14 | 0.811321 | 0.86 | 0.834951 |
| MarkBendeich | 85 | 22 | 15 | 0.794393 | 0.85 | 0.821256 |
| MarcelMichelson | 85 | 9 | 15 | 0.904255 | 0.85 | 0.876289 |
| ToddNissen | 85 | 13 | 15 | 0.867347 | 0.85 | 0.858586 |
| DavidLawder | 85 | 19 | 15 | 0.817308 | 0.85 | 0.833333 |
| KouroshKarimkhany | 84 | 32 | 16 | 0.724138 | 0.84 | 0.777778 |
| KeithWeir | 84 | 14 | 16 | 0.857143 | 0.84 | 0.848485 |
| JonathanBirt | 84 | 10 | 16 | 0.893617 | 0.84 | 0.865979 |
| SimonCowell | 82 | 9 | 18 | 0.901099 | 0.82 | 0.858639 |
| GrahamEarnshaw | 82 | 24 | 18 | 0.773585 | 0.82 | 0.796117 |
| JanLopatka | 81 | 15 | 19 | 0.843750 | 0.81 | 0.826531 |
| TheresePoletti | 81 | 33 | 19 | 0.710526 | 0.81 | 0.757009 |
| LydiaZajc | 81 | 4 | 19 | 0.952941 | 0.81 | 0.875676 |
| JoeOrtiz | 81 | 26 | 19 | 0.757009 | 0.81 | 0.782609 |
| NickLouth | 78 | 16 | 22 | 0.829787 | 0.78 | 0.804124 |
| KevinDrawbaugh | 77 | 17 | 23 | 0.819149 | 0.77 | 0.793814 |
| PatriciaCommins | 77 | 9 | 23 | 0.895349 | 0.77 | 0.827957 |
| KirstinRidley | 77 | 24 | 23 | 0.762376 | 0.77 | 0.766169 |
| MartinWolk | 76 | 18 | 24 | 0.808511 | 0.76 | 0.783505 |
| JohnMastrini | 76 | 23 | 24 | 0.767677 | 0.76 | 0.763819 |
| KevinMorrison | 76 | 29 | 24 | 0.723810 | 0.76 | 0.741463 |
| BradDorfman | 75 | 24 | 25 | 0.757576 | 0.75 | 0.753769 |
| BernardHickey | 70 | 18 | 30 | 0.795455 | 0.70 | 0.744681 |
| PeterHumphrey | 68 | 46 | 32 | 0.596491 | 0.68 | 0.635514 |
| EricAuchard | 68 | 35 | 32 | 0.660194 | 0.68 | 0.669951 |
| SarahDavison | 66 | 20 | 34 | 0.767442 | 0.66 | 0.709677 |
| AlexanderSmith | 65 | 23 | 35 | 0.738636 | 0.65 | 0.691489 |
| SamuelPerry | 65 | 27 | 35 | 0.706522 | 0.65 | 0.677083 |
| TanEeLyn | 61 | 37 | 39 | 0.622449 | 0.61 | 0.616162 |
| BenjaminKangLim | 58 | 58 | 42 | 0.500000 | 0.58 | 0.537037 |
| JaneMacartney | 58 | 49 | 42 | 0.542056 | 0.58 | 0.560386 |
| WilliamKazer | 46 | 46 | 54 | 0.500000 | 0.46 | 0.479167 |
| MureDickie | 45 | 46 | 55 | 0.494505 | 0.45 | 0.471204 |
| ScottHillis | 42 | 52 | 58 | 0.446809 | 0.42 | 0.432990 |

```
Overall:
N Correctly Classified:   3946
N Incorrectly Classified: 1054
Accuracy:                 0.789
```

### B. Random Forest Classifier (10 trees & 1000 word attributes & sample size 1000 & threshold 0.2)

```
Results Summary:
File: data_rf_results.csv

By Author:
```

| Author | Hits | Strikes | Misses | Precision | Recall | F1 |
|---|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| JohnMastrini | 89 | 285 | 11 | 0.237968 | 0.89 0.375527 |
| DavidLawder | 88 | 51 | 12 | 0.633094 | 0.88 0.736402 |
| TheresePoletti | 84 | 146 | 16 | 0.365217 | 0.84 0.509091 |
| JoWinterbottom | 82 | 29 | 18 | 0.738739 | 0.82 0.777251 |
| GrahamEarnshaw | 81 | 40 | 19 | 0.669421 | 0.81 0.733032 |
| AlanCrosby | 79 | 63 | 21 | 0.556338 | 0.79 0.652893 |
| KirstinRidley | 77 | 24 | 23 | 0.762376 | 0.77 0.766169 |
| JimGilchrist | 77 | 26 | 23 | 0.747573 | 0.77 0.758621 |
| BernardHickey | 76 | 67 | 24 | 0.531469 | 0.76 0.625514 |
| LynneO'Donnell | 76 | 17 | 24 | 0.817204 | 0.76 0.787565 |
| AlexanderSmith | 75 | 77 | 25 | 0.493421 | 0.75 0.595238 |
| MarcelMichelson | 75 | 17 | 25 | 0.815217 | 0.75 0.781250 |
| AaronPressman | 73 | 76 | 27 | 0.489933 | 0.73 0.586345 |
| BradDorfman | 72 | 59 | 28 | 0.549618 | 0.72 0.623377 |
| NickLouth | 71 | 35 | 29 | 0.669811 | 0.71 0.689320 |
| MatthewBunce | 70 | 3 | 30 | 0.958904 | 0.70 0.809249 |
| EdnaFernandes | 70 | 56 | 30 | 0.555556 | 0.70 0.619469 |
| PierreTran | 69 | 21 | 31 | 0.766667 | 0.69 0.726316 |
| LydiaZajc | 69 | 12 | 31 | 0.851852 | 0.69 0.762431 |
| KevinMorrison | 68 | 19 | 32 | 0.781609 | 0.68 0.727273 |
| FumikoFujisaki | 67 | 39 | 33 | 0.632075 | 0.67 0.650485 |
| RobinSidel | 67 | 4 | 33 | 0.943662 | 0.67 0.783626 |
| PeterHumphrey | 67 | 16 | 33 | 0.807229 | 0.67 0.732240 |
| JonathanBirt | 66 | 25 | 34 | 0.725275 | 0.66 0.691099 |
| MarkBendeich | 65 | 22 | 35 | 0.747126 | 0.65 0.695187 |
| KouroshKarimkhany | 65 | 21 | 35 | 0.755814 | 0.65 0.698925 |
| BenjaminKangLim | 65 | 55 | 35 | 0.541667 | 0.65 0.590909 |
| JoeOrtiz | 65 | 30 | 35 | 0.684211 | 0.65 0.666667 |
| LynnleyBrowning | 64 | 11 | 36 | 0.853333 | 0.64 0.731429 |
| TanEeLyn | 63 | 50 | 37 | 0.557522 | 0.63 0.591549 |
| TimFarrand | 62 | 17 | 38 | 0.784810 | 0.62 0.692737 |
| KeithWeir | 62 | 10 | 38 | 0.861111 | 0.62 0.720930 |
| DarrenSchuettler | 62 | 35 | 38 | 0.639175 | 0.62 0.629442 |
| HeatherScoffield | 61 | 21 | 39 | 0.743902 | 0.61 0.670330 |
| MartinWolk | 60 | 14 | 40 | 0.810811 | 0.60 0.689655 |
| SimonCowell | 59 | 14 | 41 | 0.808219 | 0.59 0.682081 |
| RogerFillion | 59 | 6 | 41 | 0.907692 | 0.59 0.715152 |
| JaneMacartney | 59 | 40 | 41 | 0.595960 | 0.59 0.592965 |
| MichaelConnor | 58 | 8 | 42 | 0.878788 | 0.58 0.698795 |
| PatriciaCommins | 58 | 16 | 42 | 0.783784 | 0.58 0.666667 |
| EricAuchard | 57 | 35 | 43 | 0.619565 | 0.57 0.593750 |
| JanLopatka | 57 | 24 | 43 | 0.703704 | 0.57 0.629834 |
| ToddNissen | 56 | 9 | 44 | 0.861538 | 0.56 0.678788 |
| SarahDavison | 53 | 13 | 47 | 0.803030 | 0.53 0.638554 |
| ScottHillis | 49 | 20 | 51 | 0.710145 | 0.49 0.579882 |
| KevinDrawbaugh | 44 | 8 | 56 | 0.846154 | 0.44 0.578947 |
| MureDickie | 44 | 15 | 56 | 0.745763 | 0.44 0.553459 |
| KarlPenhaul | 43 | 14 | 57 | 0.754386 | 0.43 0.547771 |
| SamuelPerry | 42 | 17 | 58 | 0.711864 | 0.42 0.528302 |

```
     WilliamKazer    36       12      64   0.750000    0.36 0.486486

Overall:
N Correctly Classified:   3256
N Incorrectly Classified: 1744
Accuracy:                 0.651
```