CSC 466 Lab 6 Report

Authors:
Sophia Chung, spchung@calpoly.edu
Aditi Gajjar, agajjar@calpoly.edu
Anagha Sikha, arsikha@calpoly.edu

Introduction

Our main goal for this project is to examine the feasibility of using classification algorithms, particularly K-Nearest Neighbors (KNN) and Random Forests, to accurately determine the authorship of the Reuter's 50-50 dataset. This dataset contains text documents that are widely used in text mining and classification tasks. This study not only builds upon the methodologies utilized in Lab 3, but also serves as a standalone investigation into the effectiveness of these algorithms in the field of text classification and authorship attribution.

Methods

Text Vectorizer (Pre-Processing)

Preprocessing is crucial in shaping the dataset for effective analysis. To set up our data for our modeling methods, we followed the key steps mentioned below:

1. Data Collection and Organization: By utilizing the Reuter 50-50 dataset, comprising news stories by 50 authors, we ensure a balanced representation for each author. The dataset is neatly divided into training and test sets, which were given to us combined for a comprehensive analysis.

2. Text Vectorization: Each document is transformed into a vectorized tf-idf (Term Frequency-Inverse Document Frequency) representation. This process converts text data into a numerical format, enabling the application of machine learning algorithms.

3. Stopword Removal and Stemming: We used two techniques, stopword removal and stemming, to improve the quality of our dataset for classification algorithms. Stopword removal eliminates common and less meaningful words, reducing noise and focusing attention on significant terms. On the other hand, stemming condenses words to their root forms, ensuring uniformity and reducing word variation. The combination of these two techniques effectively reduces the complexity of the dataset, while maintaining essential content and computational efficiency. This dual approach significantly contributes to improved accuracy and processing speed in our text mining and classification tasks.

4. Creation of Ground Truth Data: For evaluation purposes, a ground truth file is prepared, mapping each document to its respective author. This aids in the later stages of assessing the accuracy of our classification models.

Random Forests

The RFAuthorship.py file implements the Random Forest classification algorithm for authorship identification. This program takes the file containing the vectorized tf-idf representations of the documents, m (the number of attributes in each tree), k (the number of data points used for constructing a decision tree), and N (the number of decision trees to build) as input parameters. For this case, Random Forests does not require cross-validation, as building individual trees with subsets of the data is sufficient to prevent overfitting. We chose to run the classifier following the 50-class classification way. This involved recognizing the individual authors directly, allowing a more comprehensive evaluation of the model's accuracy in attributing authorship across the entire dataset. The output of this program consists of the predicted authorship labels for each document, which is used to measure the algorithm's performance through various metrics.

KNN

The knnAuthorship.py file implements the K-Nearest Neighbors (KNN) algorithm for authorship attribution. We chose to implement a 50-class classification strategy. Our program takes the vectorized document representations file, k (number of nearest neighbors), and a specified similarity metric (cosine similarity or Okapi) as input parameters. Our approach involves testing two distance metrics—cosine similarity and Okapi distance. Cosine similarity distance excels in capturing directional similarity, making it ideal for discerning writing style nuances. For this metric, we convert our term frequency matrix into a TF-IDF matrix. For the Okapi distance metric , we incorporate term frequency-inverse document frequency term frequency (TF) representations, which represent the uniqueness of each author's vocabulary. By considering term rarity in the document corpus, Okapi distance provides a nuanced perspective on authorship.

Beyond distance metrics, we hyper-tune parameters like the number of neighbors (k) and vector parsing method. All-but-one cross-validation is employed to evaluate the KNN model, preventing overfitting. The 50-class classification strategy in the implementation enables the recognition of individual authors among the 50 present in the dataset. The program outputs predicted authorship labels for each document, ensuring an effective and concise authorship attribution process.

Classifier Evaluation

The classifierEvaluation.py file is used to evaluate the accuracy of the predictions made by either KNN or Random Forests algorithm. This program takes as input the output file generated by knnAuthorship.py or RFAuthorship.py and the ground truth file. It outputs detailed metrics for each author, including the total number of hits (correctly predicted documents), strikes (false positives predicted), and misses (documents written by the

author that were not attributed to the author). Precision, recall, and f1-score for each author were reported as well. Lastly, the overall accuracy of the predictions and a 50x50 confusion matrix are also provided.

Experiments

Method 1: KNN Classification

k	Distance Metric	Total Correct	Total Incorrect	Accuracy
10	Cosine similarity	1731	3269	0.3462
15	Okapi	1422	3578	0.2844

In this experiment, we tested different values of k to observe its impact on model performance. For the distance metrics, we evaluated both cosine similarity and Okapi distance to understand their effectiveness in capturing authorship patterns. The decision to increase or decrease the value of k was driven by a balance between overfitting and underfitting. Lower values of k, such as k = 10, tend to capture more local patterns but may be sensitive to noise, potentially leading to overfitting. However, higher values of k, like k = 15, capture more global patterns but might overlook local intricacies, potentially resulting in underfitting. We aimed to strike a balance between model complexity and predictive accuracy, with the data above displaying the trade-offs made. From the table above we see that with k=10 and Cosine similarity, the model achieved higher accuracy, whereas with k=15 and Okapi distance, the accuracy decreased.

Method 2: Random Forest

m	k	N	threshold	Total Correct	Total Incorrect	Accuracy
100	1000	50	0.000001	4707	293	0.9414
200	1000	100	0.000001	4515	485	0.903
250	500	150	0.0000001	3233	1767	0.6466

We explored different combinations to strike a balance between model complexity and predictive accuracy with the goal of optimizing the model's performance. The increase or decrease in m, k, N, and threshold was driven by observing the impact on the model's ability to accurately attribute authorship. For instance, when m = 100, k = 1000, N = 50, and threshold = 0.000001, the model achieved a high accuracy of 94.14%. However, as we

increased the number of attributes to 200 and 250, the accuracy slightly decreased, indicating a trade-off between model complexity and overfitting.

Results

Method 1: KNN Classification

The best results were with k = 10 and cosine similarity:

Author	Hits	Strikes	Misses	Precision	Recall	F1
RobinSidel	48	50	52	0.4898	0.48	0.4848
LynnleyBrowning	50	47	50	0.5155	0.5	0.5076
KouroshKarimkhany	50	63	50	0.4425	0.5	0.4695
MichaelConnor	43	43	57	0.5	0.43	0.4624
JoeOrtiz	38	45	62	0.4578	0.38	0.4153
EricAuchard	36	38	64	0.4865	0.36	0.4138
AaronPressman	49	46	51	0.5158	0.49	0.5026
SimonCowell	41	40	59	0.5062	0.41	0.453
LynneO'Donnell	49	47	51	0.5104	0.49	0.5
EdnaFernandes	41	45	59	0.4767	0.41	0.4409
KevinMorrison	36	40	64	0.4737	0.36	0.4091
SamuelPerry	44	47	56	0.4835	0.44	0.4607
PatriciaCommins	46	47	54	0.4946	0.46	0.4767
JohnMastrini	47	49	53	0.4896	0.47	0.4796
JanLopatka	48	51	52	0.4848	0.48	0.4824
KevinDrawbaugh	44	35	56	0.557	0.44	0.4916

KarlPenhaul	50	49	50	0.5051	0.5	0.5025
MartinWolk	45	43	55	0.5114	0.45	0.4787
ScottHillis	45	51	55	0.4688	0.45	0.4592
DavidLawder	46	46	54	0.5	0.46	0.4792
FumikoFujisaki	48	49	52	0.4948	0.48	0.4873
MarcelMichelson	48	50	52	0.4898	0.48	0.4848
NickLouth	46	44	54	0.5111	0.46	0.4842
DarrenSchuettler	46	42	54	0.5227	0.46	0.4894
WilliamKazer	34	37	66	0.4789	0.34	0.3977
TanEeLyn	48	44	52	0.5217	0.48	0.5
PierreTran	49	47	51	0.5104	0.49	0.5
HeatherScoffield	45	47	55	0.4891	0.45	0.4688
MureDickie	43	69	57	0.3839	0.43	0.4057
RogerFillion	47	144	53	0.2461	0.47	0.323
JimGilchrist	55	547	45	0.0914	0.55	0.1567
BradDorfman	45	144	55	0.2381	0.45	0.3114
AlanCrosby	49	148	51	0.2487	0.49	0.33
JonathanBirt	49	133	51	0.2692	0.49	0.3475
BenjaminKangLim	44	148	56	0.2292	0.44	0.3014
TheresePoletti	44	145	56	0.2328	0.44	0.3045
KeithWeir	45	147	55	0.2344	0.45	0.3082
JoWinterbottom	46	93	54	0.3309	0.46	0.3849
MarkBendeich	1	86	99	0.0115	0.01	0.0107

JaneMacartney	1	94	99	0.0105	0.01	0.0103
MatthewBunce	0	78	100	0	0	0
ToddNissen	0	1	100	0	0	0
PeterHumphrey	0	4	100	0	0	0
TimFarrand	0	2	100	0	0	0
SarahDavison	0	0	100	0	0	0
GrahamEarnshaw	0	1	100	0	0	0
BernardHickey	0	1	100	0	0	0
KirstinRidley	1	0	99	1	0.01	0.0198
AlexanderSmith	0	0	100	0	0	0
LydiaZajc	1	2	99	0.3333	0.01	0.0194
Author	Hits	Strikes	Misses	Precision	Recall	F1

Document Statistics

Total number of documents with correctly predicted authors: 4707
Total number of documents with incorrectly predicted authors: 293
Overall accuracy: 0.9414

Method 2: Random Forest Classification

Our best results were with parameters m =100, k =1000, and N = 50:

Author	Hits	Strikes	Misses	Precision	Recall	F1
RobinSidel	97	0	3	1	0.97	0.98477
LynnleyBrowning	100	5	0	0.95238	1	0.97561
KouroshKarimkhany	100	5	0	0.95238	1	0.97561
MichaelConnor	94	1	6	0.98947	0.94	0.9641

JoeOrtiz	100	23	0	0.81301	1	0.89686
EricAuchard	89	0	11	1	0.89	0.9418
AaronPressman	98	3	2	0.9703	0.98	0.97512
SimonCowell	86	2	14	0.97727	0.86	0.91489
LynneO'Donnell	97	1	3	0.9898	0.97	0.9798
EdnaFernandes	87	1	13	0.98864	0.87	0.92553
KevinMorrison	85	1	15	0.98837	0.85	0.91398
SamuelPerry	92	0	8	1	0.92	0.95833
PatriciaCommins	91	1	9	0.98913	0.91	0.94792
JohnMastrini	99	0	1	1	0.99	0.99497
JanLopatka	100	3	0	0.97087	1	0.98522
KevinDrawbaugh	89	0	11	1	0.89	0.9418
KarlPenhaul	97	2	3	0.9798	0.97	0.97487
MartinWolk	95	0	5	1	0.95	0.97436
ScottHillis	83	3	17	0.96512	0.83	0.89247
DavidLawder	94	0	6	1	0.94	0.96907
FumikoFujisaki	95	2	5	0.97938	0.95	0.96447
MarcelMichelson	100	1	0	0.9901	1	0.99502
NickLouth	89	0	11	1	0.89	0.9418
DarrenSchuettler	89	0	11	1	0.89	0.9418
WilliamKazer	84	0	16	1	0.84	0.91304

TanEeLyn	96	17	4	0.84956	0.96	0.90141
PierreTran	96	3	4	0.9697	0.96	0.96482
HeatherScoffield	100	43	0	0.6993	1	0.82305
MureDickie	92	5	8	0.94845	0.92	0.93401
RogerFillion	100	1	0	0.9901	1	0.99502
JimGilchrist	100	101	0	0.49751	1	0.66445
BradDorfman	99	6	1	0.94286	0.99	0.96585
AlanCrosby	100	1	0	0.9901	1	0.99502
JonathanBirt	89	0	11	1	0.89	0.9418
BenjaminKangLim	94	5	6	0.94949	0.94	0.94472
ThereseePoletti	96	2	4	0.97959	0.96	0.9697
KeithWeir	97	12	3	0.88991	0.97	0.92823
JoWinterbottom	92	0	8	1	0.92	0.95833
MarkBendeich	82	2	18	0.97619	0.82	0.8913
JaneMacartney	94	5	6	0.94949	0.94	0.94472
MatthewBunce	99	0	1	1	0.99	0.99497
ToddNissen	96	4	4	0.96	0.96	0.96
PeterHumphrey	90	4	10	0.95745	0.9	0.92784
TimFarrand	93	5	7	0.94898	0.93	0.93939
SarahDavison	94	1	6	0.98947	0.94	0.9641
GrahamEarnshaw	96	4	4	0.96	0.96	0.96
BernardHickey	97	7	3	0.93269	0.97	0.95098

KirstinRidley	95	1	5	0.98958	0.95	0.96939
AlexanderSmith	93	0	7	1	0.93	0.96373
LydiaZajc	97	10	3	0.90654	0.97	0.9372
RobinSidel	97	0	3	1	0.97	0.98477
LynnleyBrowning	100	5	0	0.95238	1	0.97561
Document Statistics						
Total number of documents with correctly predicted authors: 1731 Total number of documents with incorrectly predicted authors: 3269 Overall accuracy: 0.3462						

Reflection

After analyzing the results of both the KNN and Random Forest classification methods on Reuter's 50-50 dataset, some intriguing observations emerge regarding their overall ability to accurately attribute authorship. The KNN classifier, with its best performance at k = 10 and using cosine similarity, displayed varying success across different authors. This suggests that certain authors, like Robin Sidel, Lynnley Browning, and Kourosh Karimkhany, had a balanced number of hits, strikes, and misses, indicating a moderate level of predictability in their writing styles. However, for authors such as Mark Bendeich, Jane Macartney, and others at the bottom of the table, the classifier struggled significantly, with almost no hits and a high number of misses. This performance disparity implies that the KNN method may be more sensitive to specific linguistic features or stylistic elements that are more pronounced in some authors' writings than in others. Interestingly, precision and recall varied significantly across authors, indicating that some authors' styles may be easier to predict but harder to distinguish from others - and vice versa.

After testing with optimal parameters (m = 100, k = 1000, N = 50), the Random Forest classifier demonstrated a notably superior overall accuracy compared to the KNN classifier. Notable authors, including Lynnley Browning, Kourosh Karimkhany, and Michael Connor, achieved high precision and recall scores, approaching perfect attribution. This method was found to be more resilient across a wider range of authors, indicating that the ensemble approach of Random Forests is better equipped to capture and generalize diverse stylistic patterns across multiple writers. The consistently high precision and recall scores for most authors suggest that the Random Forest model has a more uniform ability to capture distinct authorial signatures.

It was observed that authors with unique or distinct writing styles were easier to predict by both classifiers. For instance, authors with high precision and recall in the Random Forest method are likely to possess distinctive stylistic or thematic elements in their writing. On the other hand, authors with more generic or less distinctive writing styles presented a challenge, as evidenced by the low precision and recall scores in the KNN method and the few misses in the Random Forest method. The lower precision scores in the KNN classifier suggest that some authors were frequently confused with others, indicating similar writing styles or use of language.

Comparison/Conclusion

The comprehensive analysis of the Reuter's 50-50 dataset using both Random Forest and KNN classification methods reveals a distinct superiority of the Random Forest approach in authorship attribution tasks. With an impressive overall accuracy of 94.14%, the Random Forest method significantly outperforms the KNN classifier, which achieved a best-case accuracy of only 34.62%. This stark difference is further emphasized when examining precision and recall values, where Random Forest consistently demonstrates high scores across a diverse range of authors. For instance, authors like Lynnley Browning and Michael Connor saw near-perfect precision and recall in Random Forest, compared to much lower scores for authors like Robin Sidel and Joe Ortiz in the KNN model. This highlights Random Forest's robust capability to effectively navigate and distinguish between the intricate and varied writing styles present in the dataset. In contrast, the KNN classifier's performance was markedly uneven, often struggling to differentiate authors with similar styles and exhibiting significant variability across different authors. Overall, the Random Forest classification method's superior performance in accuracy, precision, and recall, along with its consistency and robustness in handling complex stylistic patterns, clearly establishes it as a more reliable and effective tool for text classification and authorship attribution in this context.