Unsupervised Capstone:

Classify authors based on the style of text

By Karen McGee

Problem Statement

- Build an unsupervised model that will classify authors based on the style of writing using natural language processing.

Research Questions:

- Are authors consistently grouped into the same cluster?
- Does your clustering on those members perform as you'd expect?
- Have your clusters remained stable or changed dramatically?
- Does our model provide a consistent performance?

Solution statement

- Import data from ten different authors of various writing styles
- Clean, tokenize and lemmatize the data
- Generate features using TFIDF
- Generate clusters (i.e. K-mean, MeanShift…etc.)
- Evaluate the performance of the clusters
- Generate models (i.e. Random Forest, logistic…etc.)
- Evaluate the performance of the models

Evaluation metrics of clusters and models

- Utilize the Random index (RI) adjusted score to evaluate the performance of each clusters.
- Use confusion matrix, classification report and accuracy score to evaluate the performance of each model.

Analysis – input of raw text file

Example of raw text file – Macbeth

"[The Tragedie of Macbeth by William Shakespeare 1603]\n\n\nActus Primus. Scoena Prima.\n\n
Thunder and Lightning. Enter three Witches.\n\n 1. When shall we three meet againe?\nIn Th
under, Lightning, or in Raine?\n 2. When the Hurley-burley's done,\nWhen the Battaile's lo
st, and wonne\n\n   3. That will be ere the set of Sunne\n\n   1. Where the place?\n 2. Vp
on the Heath\n\n   3. There to meet with Macbeth\n\n   1. I come, Gray-Malkin\n\n   All. Pa
dock calls anon: faire is foule, and foule is faire,\nHouer through the fogge and filthie a
yre.\n\nExeunt.\n\n\nScena Secunda.\n\nAlarum within. Enter King Malcome, Donalbaine, Lenox
, with\nattendants,\nmeeting a bleeding Captaine.\n\n King. What bloody man is that? he ca
n report,\nAs seemeth by his plight, of the Reuolt\nThe newest state\n\n   Mal. This is the
Serieant,\nWho like a good and hardie Souldier fought\n'Gainst my Captiuitie: Haile braue f
riend;\nSay to the King, the knowledge of the Broyle,\nAs thou didst leaue it\n\n   Cap. Do
ubtfull it stood,\nAs two spent Swimmers, that doe cling together,\nAnd choake their Art: T
he mercilesse Macdonwald\n(Worthie to be a Rebell, for to that\nThe multiplying Villanies o
f Nature\nDoe swarme vpon him) from the Westerne Isles\nOf Kernes and Gallowgrosses is supp
ly'd,\nAnd Fortune on his damned Quarry smiling,\nShew'd like a Rebells Whore: but all's to
o weake:\nFor braue Macbeth (well hee deserues that Name)\nDisdayning Fortune, with his bra
ndisht Steele,\nWhich smoak'd with bloody execution\n(Like Valours Minion) caru'd out his p
assage,\nTill hee fac'd the Slaue:\nWhich neu'r shooke hands, nor bad farwell to him,\nTill
he vnseam'd him from the Naue toth' Chops,\nAnd fix'd his Head vpon our Battlements\n\n   K

Analysis – Text file cleaned, lemmatized and tokenized

Example of a cleaned, lemmatized and tokenized file - Macbeth

['Actus Primus Scoena Prima Thunder and Lightning Enter three Witches 1 When shall we three meet againe In Thunder Lightning or in Raine 2 When the Hurleyburleys done When the Battailes lost and wonne 3 That will be ere the set of Sunne 1 Where the place 2 Vpon the Heath 3 There to meet with Macbeth 1 I come GrayMalkin All Padock call anon faire is foule and foule is faire Houer through the fogge and filthie ayre Exeunt Scena Secunda Alarum within Enter King Malcome Donalbaine Lenox with attendant meeting a bleeding Captaine King What bloody man is that he can report As seemeth by his plight of the Reuolt The newest state Mal This is the Serieant Who like a good and hardie Souldier fought Gainst my Captiuitie Haile braue friend Say to the King the knowledge of the Broyle As thou didst leaue it Cap Doubtfull it stood As two spent Swimmers that doe cling together And choake their Art The mercilesse Macdonwald Worthie to be a Rebell for to that The multiplying Villanies of Nature Doe swarme vpon him from the Westerne Isles Of Kernes and Gallowgrosses is supplyd And Fortune on his damned Quarry smiling Shewd like a Rebells Whore but alls too weake For braue Macbeth well hee deserues that Name Disdayning Fortune with his brandisht Steele Which smoakd with bloody execution Like Valours Minion carud out his passage Till hee facd the Slaue Which neur shooke hand nor bad farwell to him Till he vnseamd him from the Naue toth Chops And fixd his Head vpon our Battlements King O valiant Cousin worthy Gentleman Cap As whence the Sunne gin

Analysis - combined document text, author and author code into a data frame.

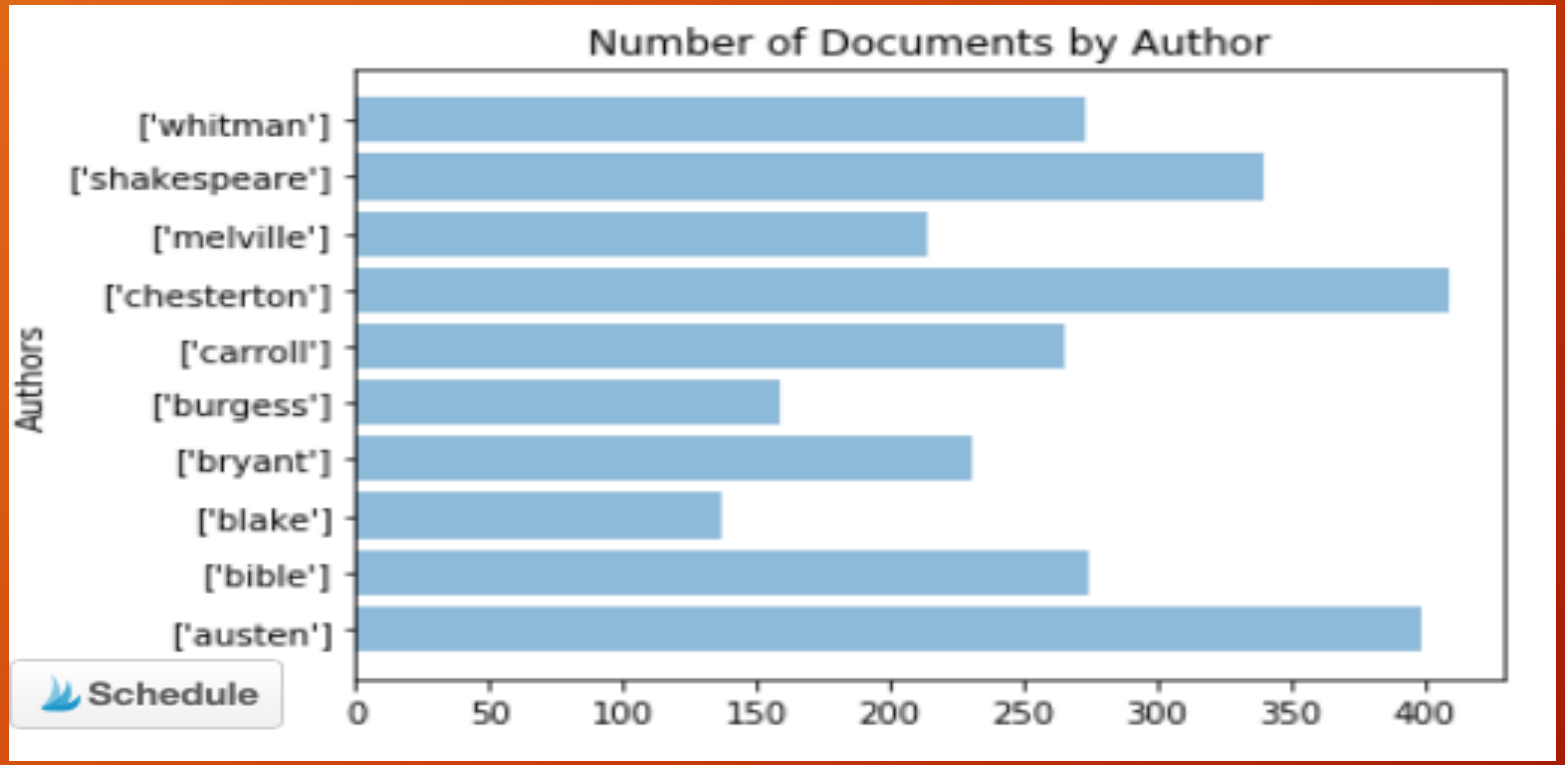
	text	authors	author_codes
0	Actus Primus Scoena Prima Enter Flauius Murell...	caesar	4
1	Forgets the shewes of Loue to other men Cassi ...	caesar	4
2	would not so with loue I might intreat you Be ...	caesar	4
3	tell you that Ile nere looke you ith face agai...	caesar	4
4	is for Romans now Haue Thewes and Limbes like ...	caesar	4

Analysis – display number of documents group by author codes and authors.

author_codes	authors	
0	austen	399
1	bible	274
2	blake	137
3	bryant	231
4	burgess	159
5	carroll	266
6	chesterton	410
7	melville	214
8	shakespeare	340
9	whitman	273

Name: authors, dtype: int64

Analysis –visual bar chart displaying number of documents group by authors



Analysis – Feature generation of text file

Features produce by TFIDF

woods	woof	wool	woollen	word	wordless	words	wore	work	workd	worke	worked	worker	working	working
0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.031282	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.013771	0.0	0.0	0.0	0.0	0.0	0.034234	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.023609	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.026765	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.033973	0.0	0.0	0.0	

5 rows × 12088 columns

Parameters used for TFIDF

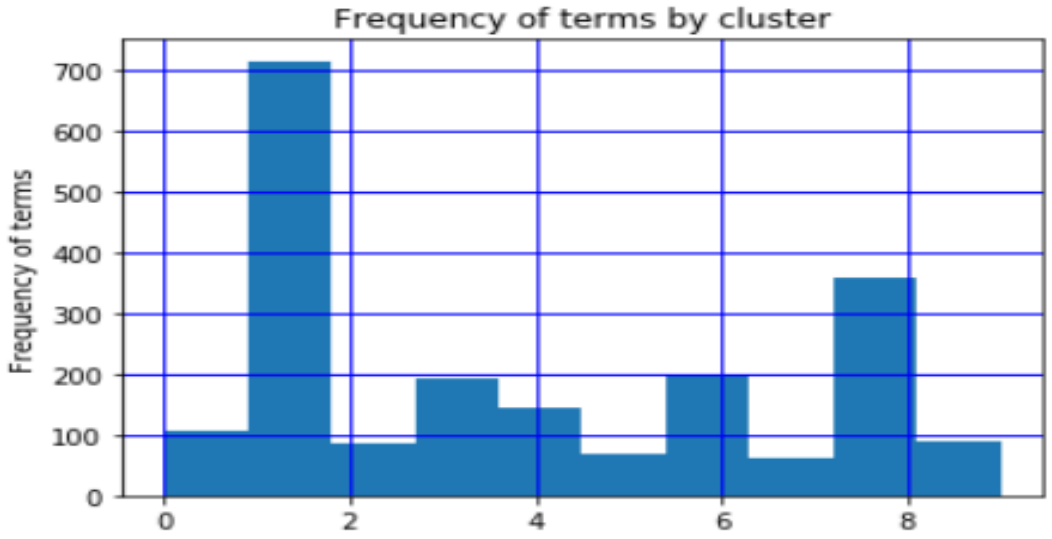
```
#Generate features using TFIDF

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_df=0.5, # drop words that occur in more than half the parag
                             min_df=3, # only use words that appear at least twice
                             stop_words=stopwords,
                             lowercase=True, #convert everything to lower case (since Alice
                             use_idf=True,#we definitely want to use inverse document freque
                             norm=u'l2', #Applies a correction factor so that longer paragra
                             smooth_idf=True #Adds 1 to all document frequencies, as if an e
                            )
```

Cluster analysis – Top terms identified for each KNN cluster and Author code

```
Top terms per cluster:
Cluster 0: Authour code: 0 ham haue lord king hor hamlet ophe laer qu selfe
Cluster 1: Authour code: 9 alice little like turnbull macian man could thing went would
Cluster 2: Authour code: 4 elinor marianne mrs could dashwood edward sister would jennings wil
loughby
Cluster 3: Authour code: 6 anne elliot mrs captain could mr wentworth charles lady would
Cluster 4: Authour code: 3 haue caesar bru macb brutus thou cassi enter cassius antony
Cluster 5: Authour code: 0 syme professor gregory man bull sunday like secretary dr anarchist
Cluster 6: Authour code: 8 unto shall lord thou ye thy god thee son israel
Cluster 7: Authour code: 7 berry jackal little mouse pail buster brahmin eat gingerbread big
Cluster 8: Authour code: 8 whale ship sea see old ahab boat upon yet long
Cluster 9: Authour code: 2 buster joe bear browns farmer little boy green pool otter
Prediction
```



Top terms from KNN clusters:

- 0,5
- 1
- 2
- 3
- 4
- 6,8
- 7
- 9

Associated with author codes:

- 0 - austen
- 9- whitman
- 4 – burgess
- 6- chesterton
- 3 -bryant
- 8 - shakesphere
- 7 - melville
- 2 - blake

Cluster visualizations: LSA Top terms by component documents

Graphically visualization of each cluster

	0	1	\
text			
carriage with four horse and with her own compl...	0.491164	-0.535324	
The guidon flag flutter gayly in the wind Bivou...	0.658404	-0.366064	
and Buster Bear had been fishing together in th...	0.142943	-0.204328	
one stroke I feel like that he said laughing bu...	0.567941	-0.594824	
the cruel order of her father and she said at o...	0.814644	-0.134597	

	2	3	\
text			
carriage with four horse and with her own compl...	0.572287	0.013016	
The guidon flag flutter gayly in the wind Bivou...	-0.389153	-0.092549	
and Buster Bear had been fishing together in th...	-0.264851	0.627365	
one stroke I feel like that he said laughing bu...	-0.371506	0.095320	
the cruel order of her father and she said at o...	0.154932	0.069491	

	4	5	\

Cluster visualizations: Kmeans crosstab results

Graphically visualization of each cluster

Comparing training k-means clusters against author codes:

col_0	0	1	2	3	4	5	6	7	8	9	Total
author_codes											
0	0	0	203	0	0	0	0	0	1	85	289
1	0	1	0	0	0	0	196	0	0	0	197
2	0	77	0	0	0	3	0	1	15	1	97
3	0	9	0	6	1	4	3	10	138	0	171
4	0	0	0	0	0	120	0	0	8	0	128
5	0	1	0	0	180	0	0	0	17	0	198
6	0	2	0	0	0	0	0	286	9	0	297
7	0	4	0	151	0	0	0	1	10	0	166
8	255	0	0	0	0	0	1	0	0	0	256
9	0	214	0	0	0	0	0	0	4	0	218
Total	255	308	203	157	181	127	200	298	202	86	2017

Comparing testing k-means clusters against author codes:

col_0	0	1	2	3	4	5	6	7	8	9	Total
author_codes											
0	0	0	72	0	0	0	0	0	2	35	109
1	0	0	0	0	0	0	67	0	0	0	67
2	0	32	0	0	0	1	0	1	6	0	40
3	0	0	0	0	0	1	0	2	57	0	60
4	0	0	0	0	0	30	0	0	1	0	31
5	0	0	0	0	62	0	0	0	5	1	68
6	0	0	0	0	0	0	0	107	5	1	113
7	0	5	0	41	0	0	0	0	2	0	48
8	83	0	0	0	0	0	0	0	0	0	83
9	0	53	0	0	0	0	0	0	1	0	54
Total	83	90	72	41	62	32	67	110	79	37	673

Cluster visualizations: MeanShift crosstab results

Graphically visualization of each cluster

Comparing training meanshift clusters against authors:

col_0	0	1	2	3	4	5	Total
author_codes							
0	6	0	280	0	0	0	286
1	0	0	0	205	0	0	205
2	76	0	0	19	0	5	100
3	141	0	3	7	2	9	162
4	4	0	0	0	0	134	138
5	10	0	1	0	190	0	201
6	299	0	0	0	0	0	299
7	164	0	0	0	0	0	164
8	0	258	0	1	0	0	259
9	203	0	0	10	0	0	213
Total	903	258	284	242	192	148	2027

Comparing testing meanshift clusters against author codes:

col_0	0	1	2	3	4	5	Total
author_codes							
0	3	0	110	0	0	0	113
1	0	0	0	69	0	0	69
2	31	1	0	3	0	2	37
3	56	0	0	3	3	7	69
4	0	0	0	0	0	21	21
5	4	0	0	0	61	0	65
6	111	0	0	0	0	0	111
7	50	0	0	0	0	0	50
8	0	81	0	0	0	0	81
9	57	0	0	3	0	0	60
Total	312	82	110	78	64	30	676

The adjusted rand score is: 0.4741988155804718

Cluster visualizations: Spectral crosstab results

Graphically visualization of each cluster

Comparing training spectral clusters against authors:

col_0	0	1	2	3	4	5	6	7	8	9	Total
author_codes											
0	0	0	0	0	189	0	11	0	86	0	286
1	0	197	0	0	0	0	0	8	0	0	205
2	0	0	3	0	0	0	19	78	0	0	100
3	4	1	4	0	0	1	145	4	0	3	162
4	0	0	128	0	0	0	10	0	0	0	138
5	0	0	0	0	0	180	20	1	0	0	201
6	0	0	0	0	0	0	298	1	0	0	299
7	141	0	0	0	0	0	18	5	0	0	164
8	0	0	0	119	0	0	1	13	0	126	259
9	1	0	0	0	0	0	20	192	0	0	213
Total	146	198	135	119	189	181	542	302	86	129	2027

Comparing testing spectral clusters against author codes:

col_0	0	1	2	3	4	5	6	7	8	9	Total
author_codes											
0	4	0	0	75	0	34	0	0	0	0	113
1	0	55	0	0	0	0	12	0	2	0	69
2	10	0	0	0	1	0	0	0	26	0	37
3	56	1	2	0	1	0	6	0	3	0	69
4	0	0	0	0	21	0	0	0	0	0	21
5	8	0	0	0	0	0	1	56	0	0	65
6	111	0	0	0	0	0	0	0	0	0	111
7	3	0	44	0	0	0	0	0	3	0	50
8	1	0	0	0	0	0	21	0	1	58	81
9	6	0	0	0	0	0	0	0	54	0	60
Total	199	56	46	75	23	34	40	56	89	58	676

The adjusted rand score is: 0.6097229776789997

Cluster visualizations: Affinity crosstab results

Graphically visualization of each cluster

Comparing training affinity clusters against authors:

col_0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	\
author_codes																	
0	0	75	0	0	0	1	0	0	0	0	0	1	0	0	101	0	
1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	
2	0	0	2	6	6	1	21	1	0	9	0	8	0	8	0	0	
3	0	0	6	5	24	4	0	1	0	0	25	3	0	3	0	0	
4	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	
5	25	0	0	0	1	1	0	0	0	0	7	1	0	0	0	87	
6	0	0	48	0	3	32	0	45	0	0	1	3	0	1	0	0	
7	0	0	1	0	0	7	1	0	16	0	0	0	0	1	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	
9	0	0	0	0	2	0	15	0	0	7	0	39	0	41	0	0	
Total	25	75	57	25	36	46	38	47	16	17	33	55	13	54	101	87	

Cluster visualizations: Affinity crosstab results continuation

Graphically visualization of each cluster

col_0	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	\
author_codes																	
0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	
1	47	73	0	0	0	0	0	0	0	0	32	0	0	0	0	51	
2	0	0	0	3	0	0	0	12	0	0	0	0	0	0	14	0	
3	0	0	0	26	2	2	0	0	0	3	2	0	0	29	2	0	
4	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0	0	
5	0	0	0	0	0	0	0	0	73	0	0	0	1	5	0	0	
6	0	0	0	0	0	1	0	0	0	58	0	0	0	0	0	0	
7	0	0	0	0	84	51	0	2	0	0	0	0	0	0	0	0	
8	0	0	33	0	0	0	57	0	0	0	1	35	0	0	0	0	
9	0	0	0	0	0	3	0	38	0	0	0	0	0	0	1	0	
Total	47	73	33	31	86	57	57	52	73	61	35	35	26	36	17	51	

Cluster visualizations: Affinity crosstab results continuation

Graphically visualization of each cluster

col_0	32	33	34	35	36	37	38	39	40	Total
author_codes										
0	0	83	0	0	0	0	0	0	0	286
1	0	0	0	0	0	0	0	0	0	205
2	0	0	1	0	0	0	0	0	8	100
3	24	0	0	0	1	0	0	0	0	162
4	0	0	0	0	46	0	74	0	0	138
5	0	0	0	0	0	0	0	0	0	201
6	0	1	106	0	0	0	0	0	0	299
7	0	0	0	0	0	0	0	0	1	164
8	0	0	0	40	0	24	0	56	0	259
9	0	0	0	0	0	0	0	0	67	213
Total	24	84	107	40	47	24	74	56	76	2027

Cluster visualizations: Affinity crosstab results continuation

Graphically visualization of each cluster

Comparing testing affinity clusters against author codes:

col_0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
author_codes																
0	0	0	34	0	0	0	0	41	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
2	0	9	0	0	0	0	7	0	1	9	0	1	5	1	0	0
3	1	0	0	0	0	2	1	0	9	1	0	1	23	8	1	0
4	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0
5	0	0	0	32	0	26	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	58	0	0	0	0	0	0	0	0	51	0	0
7	23	1	0	0	0	0	1	0	0	1	0	0	0	0	23	0
8	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0
9	0	8	0	0	0	0	22	0	0	14	0	0	0	0	1	0
Total	24	18	34	32	58	28	31	41	11	25	31	23	28	60	25	22

Cluster visualizations: Affinity crosstab results continuation

Graphically visualization of each cluster

col_0	16	17	18	19	20	21	22	23	Total
author_codes									
0	0	0	36	2	0	0	0	0	113
1	25	0	0	0	0	0	22	0	69
2	0	0	0	2	0	2	0	0	37
3	1	0	0	3	0	17	1	0	69
4	0	0	0	0	0	0	0	0	21
5	0	0	0	0	0	6	0	0	65
6	0	0	0	2	0	0	0	0	111
7	0	0	0	1	0	0	0	0	50
8	0	13	0	0	18	0	0	19	81
9	0	0	0	15	0	0	0	0	60
Total	26	13	36	25	18	25	23	19	676

Cluster evaluation results

Overall Spectral Clustering performed the best based on the Random Index (RI) Adjusted score.

	Cluster	Number of clusters	RI adjusted score
0	K-Means	10	0.816303
1	MeanShift	10	0.461811
2	SpectralClustering	10	0.628662
3	Affinity Clustering	22	0.389168

Cluster Summary

- **In summation based on my results table none of my clusters are predicting a 100% agreement between my ground truth and my solution but Kmeans Clustering is predicting the higher random index (RI) adjusted score and implies it is predicting the most accurate number of clusters.**

Model Performance – Random Forest

```
RFC Training mean set score: 0.8924398951302331
RFC Testing mean set score: 0.8301027409110038

 Random Forest confusion matrix
 [[110   0   1   2   0   0   0   0   0   0]
 [  0  69   0   0   0   0   0   0   0   0]
 [  0   0  32   3   0   0   0   0   0   2]
 [  3   0   8  46   2   5   2   0   2   1]
 [  0   0   0   0  21   0   0   0   0   0]
 [  0   0   2   2   0  60   1   0   0   0]
 [  5   0   0   6   0   4  95   0   0   1]
 [  1   0   0   1   0   0   1  47   0   0]
 [  0   0   0   0   0   0   0   0  81   0]
 [  0   0   3   4   0   1   5   0   0  47]]

 Random Forest classification report
               precision    recall  f1-score   support

            0       0.92      0.97      0.95       113
            1       1.00      1.00      1.00        69
            2       0.70      0.86      0.77        37
            3       0.72      0.67      0.69        69
            4       0.91      1.00      0.95        21
            5       0.86      0.92      0.89        65
            6       0.91      0.86      0.88       111
            7       1.00      0.94      0.97        50
            8       0.98      1.00      0.99        81
            9       0.92      0.78      0.85        60

    micro avg       0.90      0.90      0.90       676
    macro avg       0.89      0.90      0.89       676
weighted avg       0.90      0.90      0.90       676

Random Forest accuracy score: 0.8994082840236687
```

Model Performance - Knn

```
KNN Training mean set score: 0.9260875957840009
KNN Testing mean set score: 0.876811848551819

 KNN Confustion Matrix
 [[108   0   0   0   0   0   0   1   0   0]
 [  0  67   0   0   0   0   0   0   0   0]
 [  0  11  26   1   1   0   0   0   0   1]
 [  0   5   0  51   0   4   0   0   0   0]
 [  0   0   0   0  31   0   0   0   0   0]
 [  0   0   1   0   1  66   0   0   0   0]
 [  1   0   0   0   0   0 111   1   0   0]
 [  0   1   0   0   0   0   0  47   0   0]
 [  0   5   0   0   0   0   0   0  78   0]
 [  0  14   0   0   0   0   1   0   0  39]]

 KNN Classification Report
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       109
           1       0.65      1.00      0.79        67
           2       0.96      0.65      0.78        40
           3       0.98      0.85      0.91        60
           4       0.94      1.00      0.97        31
           5       0.94      0.97      0.96        68
           6       0.99      0.98      0.99       113
           7       0.96      0.98      0.97        48
           8       1.00      0.94      0.97        83
           9       0.97      0.72      0.83        54

   micro avg       0.93      0.93      0.93       673
   macro avg       0.94      0.91      0.91       673
weighted avg       0.94      0.93      0.93       673

KNN accuracy score: 0.9271916790490342
```

Model Performance – Gradient Boosting

```
Gradient Training mean set score: 0.9422917168977664
Gradient Testing mean set score: 0.8963054988206736

 Gradient Boosting confusion matrix
 [[107   0   0   5   0   0   0   0   0   1]
 [  0  69   0   0   0   0   0   0   0   0]
 [  0   0  33   3   0   0   0   0   0   1]
 [  0   1   3  64   0   0   0   0   0   1]
 [  0   0   0   0  21   0   0   0   0   0]
 [  0   0   2   4   0  59   0   0   0   0]
 [  0   0   0   1   0   0 110   0   0   0]
 [  0   0   0   0   0   0   0  50   0   0]
 [  0   0   0   3   0   0   0   0  78   0]
 [  0   0   0   1   1   1   0   0   0  57]]

 Gradient Boosting classification report
              precision    recall  f1-score   support

           0       1.00      0.95      0.97       113
           1       0.99      1.00      0.99        69
           2       0.87      0.89      0.88        37
           3       0.79      0.93      0.85        69
           4       0.95      1.00      0.98        21
           5       0.98      0.91      0.94        65
           6       1.00      0.99      1.00       111
           7       1.00      1.00      1.00        50
           8       1.00      0.96      0.98        81
           9       0.95      0.95      0.95        60

   micro avg       0.96      0.96      0.96       676
   macro avg       0.95      0.96      0.95       676
weighted avg       0.96      0.96      0.96       676

Gradient Boosting accuracy score: 0.9585798816568047
```

Model Performance – Logistic regression

```
LR Training mean set score: 0.9707663581669694
LR Testing mean set score: 0.8975607072562287

 Logistic regression confusion matrix
 [[109   0   0   0   0   0   0   0   0   0]
 [  0  67   0   0   0   0   0   0   0   0]
 [  0   0  29   3   1   0   0   0   0   7]
 [  0   0   0  59   0   0   1   0   0   0]
 [  0   0   0   0  31   0   0   0   0   0]
 [  1   0   0   1   0  66   0   0   0   0]
 [  0   0   0   0   0   0 113   0   0   0]
 [  0   0   0   0   0   0   0  48   0   0]
 [  0   0   0   0   0   0   0   0  83   0]
 [  0   0   0   0   0   0   0   0   0  54]]

 Logistic classification report
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       109
           1       1.00      1.00      1.00        67
           2       1.00      0.72      0.84        40
           3       0.94      0.98      0.96        60
           4       0.97      1.00      0.98        31
           5       1.00      0.97      0.99        68
           6       0.99      1.00      1.00       113
           7       1.00      1.00      1.00        48
           8       1.00      1.00      1.00        83
           9       0.89      1.00      0.94        54

   micro avg       0.98      0.98      0.98       673
   macro avg       0.98      0.97      0.97       673
weighted avg       0.98      0.98      0.98       673

Logistic accuracy score:  0.9791976225854383
```

Model Performance – Support Vector

```
SVC Training mean set score: 0.993090542307205 5
SVC Testing mean set score: 0.9776786951295511

 Support vector cufusion matrix
 [[109   0   0   0   0   0   0   0   0   0]
 [  0  67   0   0   0   0   0   0   0   0]
 [  0   0  38   2   0   0   0   0   0   0]
 [  0   0   0  60   0   0   0   0   0   0]
 [  0   0   0   0  31   0   0   0   0   0]
 [  0   0   1   0   0  67   0   0   0   0]
 [  0   0   0   0   0   0 113   0   0   0]
 [  0   0   0   0   0   0   0  48   0   0]
 [  0   0   0   0   0   0   0   0  83   0]
 [  0   0   0   0   0   0   0   0   0  54]]

 Support vector classification report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       109
           1       1.00      1.00      1.00        67
           2       0.97      0.95      0.96        40
           3       0.97      1.00      0.98        60
           4       1.00      1.00      1.00        31
           5       1.00      0.99      0.99        68
           6       1.00      1.00      1.00       113
           7       1.00      1.00      1.00        48
           8       1.00      1.00      1.00        83
           9       1.00      1.00      1.00        54

   micro avg       1.00      1.00      1.00       673
   macro avg       0.99      0.99      0.99       673
weighted avg       1.00      1.00      1.00       673

Support vector accuracy score: 0.9955423476968797
```

Summary results: Cluster and model performance

- Based on the crosstab results of the clusters, authors were not consistently grouped into the same cluster.
- I was expecting more clustering to occur on members who had more words per documents but the results of the clusters were not consistent with my expectation.
- Overall the clusters remained stable for every type of cluster (i.e. Kmeans, meanshift, spectral…etc.)
- Overall Model performance:
  - SVM and Logistic regression where consistent with each other with as their accuracy scores range from 99 and 98%
  - Random forest, KNN and Gradient boosting were a little lower in their accuracy scores range from 89, 93 and 95%