Machine Learning with Shakespeare's Richard III

Introduction

William Shakespeare has captured the interests of countless theatergoers and readers for 400 years. I am no different after taking several courses on Shakespeare in my undergraduate and graduate careers. A play that has long been an interest of mine is King Richard III, the history of one of Britain's most notorious kings and his rise to the throne. Much of what makes Richard so villainous in this is play is Tudor propaganda created by Sir Thomas More in his unfinished History of King Richard the Third. However, it's this witty, fiendish depiction that adds so much flavor to the play.

I have spent a great deal of time with this play and wrote an M.A. thesis about it and Shakespeare's medieval influences. For this project, however, I have an opportunity to reexamine the play with machine learning algorithms and explore insights gleaned from classification and clustering techniques.

Data Collection

The Folger Shakespeare Library has digital copies of all of Shakespeare's works in numerous file types. I was able to use the XML file of Richard III by opening it in Excel. Excel created a schema for reading the file, and I worked with the resulting XML table to put it into a more workable format for CSV loading to Python.

I created a "sex" column and populated it with the gender of each speaker in order to perform some gender analysis.

```
In [116]: richardIII= pd.read_csv("richardIII_gender.csv")
          richardIII[185:200]
```

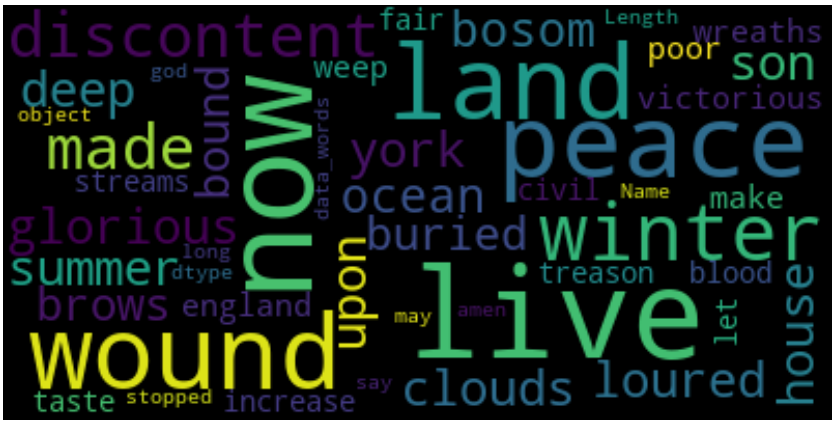
Out[116]:

	title	act_scene	folgerID	speaker	sex	prose_verse	lines
185	Richard III	1.1.108	Clarence_3H6	CLARENCE	male	verse	We know thy charge, Brakenbury, and will obey.
186	Richard III	1.1.109	RichardIII_R3	RICHARD	male	verse	We are the Queen's abjects and must obey.—
187	Richard III	1.1.110	RichardIII_R3	RICHARD	male	verse	Brother, farewell. I will unto the King,
188	Richard III	1.1.111	RichardIII_R3	RICHARD	male	verse	And whatsoe'er you will employ me in,
189	Richard III	1.1.112	RichardIII_R3	RICHARD	male	verse	Were it to call King Edward's widow sister,
190	Richard III	1.1.113	RichardIII_R3	RICHARD	male	verse	I will perform it to enfranchise you.
191	Richard III	1.1.114	RichardIII_R3	RICHARD	male	verse	Meantime, this deep disgrace in brotherhood
192	Richard III	1.1.115	RichardIII_R3	RICHARD	male	verse	Touches me deeper than you can imagine.
193	Richard III	1.1.116	Clarence_3H6	CLARENCE	male	verse	I know it pleaseth neither of us well.
194	Richard III	1.1.117	RichardIII_R3	RICHARD	male	verse	Well, your imprisonment shall not be long.
195	Richard III	1.1.118	RichardIII_R3	RICHARD	male	verse	I will deliver you or else lie for you.
196	Richard III	1.1.119	RichardIII_R3	RICHARD	male	verse	Meantime, have patience.
197	Richard III	1.1.120	Clarence_3H6	CLARENCE	male	verse	I must, perforce. Farewell.
198	Richard III	1.1.121	RichardIII_R3	RICHARD	male	verse	Go tread the path that thou shalt ne'er return.
199	Richard III	1.1.122	RichardIII_R3	RICHARD	male	verse	Simple, plain Clarence, I do love thee so

Preprocessing and Exploration

I began by dropping several columns: title, act_scene, folgerID, and prose_verse. Using the "lines" attribute, I removed punctuation, transformed to lowercase, and removed stop words. Stop words come from a custom stop words list (see appendix). I used NLTK stop words as the base and added a number of words that are common to Shakespeare's language. Because I wanted to see the effect of words like "no" and "not" on bigram creation, I removed those words from the stop words list.

```
77                          [now, winter, discontent]
78                [made, glorious, summer, son, york]
79                      [clouds, loured, upon, house]
80                       [deep, bosom, ocean, buried]
81           [now, brows, bound, victorious, wreaths]
```



Experiments

Experiment 1: Topic Modeling

I thought it would be interesting to see how machine learning algorithms would determine topics for this play. This experiment was designed out of sheer curiosity, to see if and/or how algorithms would process

1) an unusual text format (unusual in that speaker is not associated with the text in the same way it is with most types of text, like articles or fiction)

2) Shakespeare's language structure. Because most lines are in verse form, syntax is quite different. Word order confuses many modern day readers.

3) line divisions. We do not have complete thoughts on each line. Sentences span numerous lines.

I utilized Gensim's LDA modeling for this experiment. Early experiments yielded results that were not useful. These results would likely have been much improved by utilizing named entity recognition (and I plan to experiment further to see how this might work, as I have not yet played with NER). I don't know if NER might have helped identify the primary players and locations and if it would affect how topics were clustered.

Experiment 2: Classification

Though I am not working with a traditionally labeled corpus, it (eventually) occurred to me that I do have labels: I have labels that identify the speaker for each line, and I have labels for the

gender of the speaker of each line. What if I could predict either the speaker for each line or the gender of the speaker?

I used Support Vector Machine algorithms to experiment with these classification problems.

Results

Experiment 1: Topic Modeling

Using Gensim's LDA modeling algorithms yielded interesting results. I call these results interesting because I had no idea what this output would be, and I'm fascinated by it. However, the results are not terribly informative. Though my domain knowledge is excellent, there are only couple of words in each cluster that do much to inform topic.

```
[(0,
  '0.013*"come" + 0.012*"lord" + 0.011*"good" + 0.009*"god" + 0.009*"death" + '
  '0.007*"edward" + 0.006*"grace" + 0.006*"go" + 0.006*"son" + 0.005*"done"'),
 (1,
  '0.019*"lord" + 0.010*"king" + 0.008*"like" + 0.006*"may" + 0.006*"day" + '
  '0.005*"time" + 0.005*"yet" + 0.005*"buckingham" + 0.005*"good" + '
  '0.004*"come"'),
 (2,
  '0.011*"lord" + 0.011*"say" + 0.010*"god" + 0.009*"let" + 0.008*"good" + '
  '0.008*"love" + 0.007*"soul" + 0.007*"much" + 0.005*"york" + 0.005*"father"'),
 (3,
  '0.016*"lord" + 0.014*"now" + 0.010*"king" + 0.008*"man" + 0.007*"good" + '
  '0.007*"edward" + 0.006*"clarence" + 0.006*"never" + 0.005*"come" + '
  '0.005*"heart"'),
 (4,
  '0.012*"upon" + 0.008*"die" + 0.007*"god" + 0.007*"queen" + 0.006*"think" + '
  '0.006*"go" + 0.006*"lord" + 0.006*"make" + 0.005*"cannot" + 0.005*"live"')]
```

Though I was unable to use LDA Mallet's algorithm, I decided to let Mallet process the text, and I fed the Mallet-formatted text back into Gensim's LDA. The vastly different results again fascinated me but are again not terribly informative. I note them here because Mallet's processing effected one significant change in the results: the proper nouns in the above output (Edward, Buckingham, York, God, and perhaps king and queen) are not included in the new topic results.

```
[(0,
  '0.011*"butchered" + 0.011*"left" + 0.009*"perhaps" + 0.008*"exempt" + '
  '0.006*"waiting" + 0.006*"old" + 0.006*"gave" + 0.005*"claim" + '
  '0.005*"flout" + 0.005*"bottom"'),
 (1,
  '0.035*"claim" + 0.013*"oratory" + 0.013*"odd" + 0.008*"forthwith" + '
```

```
 '0.006*"changing" + 0.006*"greatness" + 0.006*"naked" + 0.005*"old" + '
 '0.005*"rely" + 0.005*"honor"'),
(2,
 '0.015*"claim" + 0.012*"changing" + 0.011*"every" + 0.010*"old" + '
 '0.010*"oratory" + 0.009*"exempt" + 0.007*"sojourn" + 0.007*"resolved" + '
 '0.006*"swine" + 0.006*"rashly"'),
(3,
 '0.032*"changing" + 0.009*"resolved" + 0.008*"sojourn" + 0.007*"tackling" + '
 '0.006*"claim" + 0.006*"nights" + 0.005*"famously" + 0.005*"greatness" + '
 '0.005*"hot" + 0.004*"forthwith"'),
(4,
 '0.015*"resolved" + 0.009*"old" + 0.008*"claim" + 0.008*"waiting" + '
 '0.007*"troops" + 0.007*"naked" + 0.007*"sojourn" + 0.006*"level" + '
 '0.006*"oratory" + 0.005*"changing"')]
```

As I continued to experiment, I added bigrams and stemming. I don't know if the following links will work, but here are the visualizations for topic modeling for the entire play: LDA_viz_5

I also performed LDA topic modeling for just the character Richard III: Richard_LDA_viz_5

Experiment 2: Classification

I began my experiments with Linear SVM for both classifying by speaker and by gender. After becoming more familiar with expected outputs, I ran GridSearchCV on each to determine the best kernel and cost for each.

Classification by speaker performed best with the Radial Basis Function and a cost of 1. Early results with Linear SVM produced accuracy scores around 25%. However, RBF produced an accuracy score of 32.5%. Though this is not a particularly inspiring result, in many ways this is still not terrible. Among the many complications for this task, there is the fact that there are 53 classes to predict. (I plan to further experiment by reducing the number of speakers to 10 by filtering the top 10 speakers.) Additionally, the task is complicated by the fact that Shakespeare wrote all the lines; the speakers do not have the kind of nuanced speech that real individuals would have, the nuances that might make individual speech patterns easier to discern.

Classification by gender, however, was rather successful. The dataset is quite unbalanced.

```
[['female' 571]
 ['male' 2064]]
```

However, the polynomial kernel with a cost of 1 performed quite well in predicting female lines. The overall accuracy result of 78% were considerably higher than I had expected.

```
[[782  89]
 [170  89]]
```

	precision	recall	f1-score	support
male	0.68	0.10	0.18	259
female	0.79	0.99	0.88	871
accuracy			0.78	1130
macro avg	0.73	0.54	0.53	1130
weighted avg	0.76	0.78	0.72	1130

Appendix

Custom Stop Words

'i',	'themselves',	'as',	'when',	'thou',
'me',	'what',	'until',	'where',	'thy',
'my',	'which',	'while',	'why',	'thine',
'myself',	'who',	'of',	'how',	'hath',
'we',	'whom',	'at',	'all',	'would',
'our',	'this',	'by',	'any',	'shall',
'ours',	'that',	'for',	'both',	'tis',
'ourselves',	"that'll",	'with',	'each',	'well',
'you',	'these',	'about',	'few',	‘thus’
"you're",	'those',	'against',	'more',	
"you've",	'am',	'between',	'most',	
"you'll",	'is',	'into',	'other',	
"you'd",	'are',	'through',	'some',	
'your',	'was',	'during',	'such',	
'yours',	'were',	'before',	'only',	
'yourself',	'be',	'after',	'own',	
'yourselves',	'been',	'above',	'same',	
'he',	'being',	'below',	'so',	
'him',	'have',	'to',	'than',	
'his',	'has',	'from',	'too',	
'himself',	'had',	'up',	's',	
'she',	'having',	'down',	'can',	
"she's",	'do',	'in',	'will',	
'her',	'does',	'out',	'just',	
'hers',	'did',	'on',	'should',	
'herself',	'doing',	'off',	"should've",	
'it',	'a',	'over',	'd',	
"it's",	'an',	'under',	'll',	
'its',	'the',	'again',	'm',	
'itself',	'and',	'further',	'o',	
'they',	'but',	'then',	're',	
'them',	'if',	'once',	've',	
'their',	'or',	'here',	'y',	
'theirs',	'because',	'there',	'thee',	