## Summary of Findings

In this project, we applied various techniques to verify the authorship of Shakespeare’s poems and plays. Our techniques include (1) extracting features using TFIDF, n-gram, readability, and type-token ratio, (2) reducing dimensionality using PCA, (3) clustering plays and poems using K-Means, GMM, Hierarchical and DBSCAN and (4) visualize and analyze the data using MDS, LLE, Isomap, and others.

From our cluster analyses, we are suspicious of the authenticity of many scenes in historical plays. These plays are Anthony and Cleopatra, The 1st and 2nd Part of King Henry the IV, The Life of King Henry the V, The 1st, 2nd, and 3rd Part of King Henry the VI, The Life of King Henry the VIII, The Life and Death of King John, The Life and Death of Richard II, and The Life and Death of Richard the III. We also raised our suspicion in some parts of Romeo and Juliet due to its readability and the type token ratio. Specifically, Anthony and Cleopatra Act 4 and The Second Part of King Henry the Fourth Act 1, 2, and 5 are highly different from others.

Among the poems, we found that ‘a lover's complaint’, ‘the rape of lucrece’, and ‘venus and adonis’ are different from other poems and they might not be written by Shakespeare. Even among these works, they are not similar to each other, and might be written by three different persons.

We also took this project further by training a classifier to classify the genre of plays (i.e. historical, tragedy, or comedy plays). We trained Deep Belief Nets and Random Forest as classifiers. Our results show that both can classify plays with more than 50% (and up to 70%) accuracy with Random Forest performed better.

## Our Game Plan

We began with (1) scraping data and try to keep the details as much as possible. Here, we think the scene-level data should be appropriate because it contains more words than speech-level data but smaller enough to notice the difference among the scenes in the same plays (we don’t expect the whole plays to be written by other author without Shakespeare being part of it). Then, we (2) tried extracting features using different techniques and (3) explore the data using clustering and visualization techniques. Our plan is to focus mainly on straightforward features (e.g. TFIDF, n-gram) and use other advanced features (e.g. readability and type-token ratio) to support our findings.

## Web Scraping and Data preparation

(See folder /Data)

We scraped the data from <http://shakespeare.mit.edu> which published all Shakespeare’s works. We used Python 3’s ‘requests’ and ‘re’ to scrape the data (see Data/scraping.py). First we identified the html tags that related to the section that we want to scrape. Then, we scraped the speeches and categorized whether the scene is a plays or a poem. We also labelled the speaker names and the genre of the plays (i.e. tragedy, historical, or comedy). Because there are some pages that have different html tags that others (e.g. prologues and Henry VI act 5), we treated them separately. We formatted the data in pandas and stored them in tab-delimited .txt files.

We then preprocessed the data using Python 3’s nltk library (see Data/tokenStemRemove.py). First, we stripped out whitespace, punctuations, and common English stopwords. Then we stemmed the words using PorterStemmer() class in nltk. This stemmer allows us to group similar words such as delivering and delivery together by stemming them to just ‘deliv’. Lastly, we tokenized the speeches into words.

## Feature Extraction

(See FeatureExtraction\_TFIDF.ipynb, FeatureExtraction\_poems.ipynb, and Readability\_TTRatio\_FeatureExtraction.ipynb)

Before we can begin clustering, we need to extract features from the tokenized words data. The following types of features have been applied as style markers can be distinguished:

1. Term Frequency Inverse Document Frequency (TFIDF) and n-gram

We choose bag-of-words which is a sparse vector of occurrence counts of words as a representation of works, and we use TFIDF, an abbreviation of term frequency–inverse document frequency, which is a numerical statistic. TFIDF is intended to reflect how important a word is to a document in a collection or corpus.

We extracted features by counting unigram, 2-grams, 3-grams, 4-grams this way we capture phrases and multi-word expressions. The extracted features are exported as arrays of dimension (n x m), n is the number of plays or scenes, or poems, the m is the number of features, and we limit the number of features to 5000. The multiple vectors are saved into txt files (See folders /VectorizedFeatures and /N-gramFeatures and /Poems for data).

1. Type Token Ratio (TTR) - Richness of vocabulary

We believe different authors are likely to have different level of vocabulary richness reflected in the works, and therefore we extracted the type-token ratio as a measure of vocabulary richness from Shakespeare’s plays. Type-token ratio is calculated as V/N, where V represents the size of the vocabulary (unique tokens) of the sample, and N is the number of tokens. This TTR is used to support our results.

Reference: <http://www.lexically.net/downloads/version5/HTML/index.html?type_token_ratio_proc.htm>

1. Syntactic Features (Readability)

Syntactic features have been proposed as more reliable style markers since they are not under the conscious control of the author (Baayen et al., 1996; Diederich et al., 2000; Khmelev and Tweedie, 2001; Kukushkina et al., 2001; Stamatatos et al., 1999).

We counted the number of sentences, words, letters, average lengths of words, average lengths of sentences. We did this separately for each act, scene and play of Shakespeare’s plays. Besides, we also calculated the Coleman–Liau index (CLI), which is a readability test designed by Meri Coleman and T. L. Liau to gauge the understandability of a text (<https://en.wikipedia.org/wiki/Coleman%E2%80%93Liau_index>)

The Coleman–Liau index is calculated with the following formula:

CLI = 0.0588{L} - 0.296{S} - 15.8\,\!

L is the average number of letters per 100 words and S is the average number of sentences per 100 words. Those features above are exported as .txt files in Readability\_Features folder, and there sure will be some multicollinearity among them, and we will use PCA to remove the multicollinearity.

## Dimensionality Reduction – TFIDF and n-gram PCA

Even limit the number of words to the top 5000 frequently used words, we might still have too many features for clustering. Too many features make it difficult to visualize and analyze the clusters. As a result, we tried to use Latent Sematic Analysis (LSA) and Principal Component Analysis (PCA) to reduce the dimensions of the data.

First, Latent Semantic Analysis is applied then to reduce the dimensionality from 5000 to 2. However the explained variance of the LSA reduced model is only 5% to 10% which is quite low. So we tried applying PCA to find the number of components that we need to explain the variance. To use PCA, we normalized the vectors first. Then we applied PCA from n = 1 to n = number of original features.

For plays, Figure 1 shows that for the TFIDF data and 2-gram, we don’t gain much explained variance after n > 100. For poems (See Figure 2), the explained variance is tailing off at n = 3 for TFIDF. So we decided to try the following numbers of components:

|  |  |  |
| --- | --- | --- |
| Feature Extraction | #Components  Plays | #Components  Poems |
|  |  |  |
| TFIDF | 100 | 3 |
| 2-gram | 100 | 3 |

By doing this, we limited our first exploration to only 4 models which are manageable.

## Clustering and Visualization – Poems

(See Clustering\_poem\_final.ipynb)

We first explored the poems using K-Means and GMM. Figure 3 shows the K-Means and GMM results using 3 PCA components and 2 clusters (We initialized 30 times and use 10,000 iterations). Using more clusters than 2 did not give a good separation as far as we tried it. Both show similar results with one large group of poems (green dots) having positive PCA1 and another small group (blue dots) with negative PCA1. GMM seems to provide a little better separation as seen with the clearer cut between groups. However, as we looked at the list of poems, they are mostly Sonnets and ‘a funeral elegy’. Because it is difficult to find a pattern here, we tried clustering using 2-gram to explore the data.

If we look at the 2-gram data (Figure 4), we can see that a lot of poems are clustered into one group (blue dots) and there are three poems that separated itself from others. We located these poems and they are ‘a lover's complaint’, ‘the rape of lucrece’, and ‘venus and adonis’. The PCA chart shows that the red, light-blue, and green dots are far away from the heavily clustered blue dots. As a result, we conclude that ‘a lover's complaint’, ‘the rape of lucrece’, and ‘venus and adonis’ might not be written by Shakespeare. And among these works, they might be written by three different persons.

## Clustering and Visualization (1st Explore) – Plays

(See Clustering\_play\_final.ipynb)

We first explored the plays at the scene level using K-Means and GMM. Figure 5 shows the K-Means and GMM results using 100 PCA components for TFIDF and 2-gram using 3 clusters. From TFIDF, the interesting group is the red dots on K-Means which have very low PCA1 and PCA2. These group separates itself from the others. So we suspected that this group of works might not be genuine. From K-means, these scenes are Cymbeline Act 3 Scene 1 and 2, most of Anthony and Cleopatra, and most of The Life and Death of Julius Caesar (See full list in Clustering\_play\_final.ipynb).

If we look at the K-Means clustering of 2-gram data (Figure 6), we can see that a lot of poems are clustered into one group (blue dots) and there are two smaller groups that are further away. We again suspected that these might not be genuine Shakespeare’s works. These works are Anthony and Cleopatra Act 4 and The Second Part of King Henry the Fourth Act 1, 2, and 5.

Given these results, we are suspicious of Anthony and Cleopatra and some of the historical plays. We conducted further investigations to confirm our speculation.

## Clustering and Visualization (Confirmatory) – Plays

(See Visualization\_Features\_Plays.ipynb)

##### Projections of Readability and Type Token Ratio

We now look at n-grams and readability features to dig deeper on the plays data. We projected the first two components and visualize what might be different from the majority (See Figure 7). In the PCA processed unigram vectorization of Shakespeare’s plays in each scene, we see that scenes from plays 20, 21, 22, 25, 27, 28 formed four clusters of their own which are separated from the rest of the scenes of plays. These plays are all historical plays (e.g. Henry VI, Richard, Anthony and Cleopatra, Coriolanus). Please refer to Table 1 for the full list of plays.

Looking at readability features (See Figure 8) of Shakespeare’s plays in each scene, we see that scenes from plays 22 (The Third Part of King Henry the VI) have a distinct level of readability (syntactic features) than other plays.

The Type Token Ratio feature is another way that we investigated the data (See Figure 9). TTR is an one column vector, and thus there is no need to use PCA to reduce dimensions to visualize, here the x-axis is the play number, and the Y- axis shows the Type Token Ratio, and we can see some scenes from plays 0,7,17,15,16,18,22,27,32 (e.g. All’s Well, Merchant of Venice, Two Gentlemen of Verona, Winter’s Tale, Henry VI, Anthony and Cleopatra, Macbeth) have higher Type Token Ratio, indicating more vocabulary richness than the rest of plays.

We can see that the authenticity of Shakespeare’s historical works are quite suspicious. We saw again that Anthony and Cleopatra and Henry VI are quite different from others.

##### Manifold projection on non-linear plane

We also tried manifold projection methods including LLE, LTSA, Hessian LLE, Modified LLE, ISOMAP, MDS, Spectral Embedding, t-SNE dimension reduction methods and visualize those projections.

In the act readability manifold projection graphs (Figure 10), we can see that scenes 3 and 4 from plays 22 (The Third Part of King Henry the VI), scene 0 from plays 19 (The Life of King Henry the V) and scene 2 and 3 from play 34 (Romeo and Juliet) appear quite a few times as outliers than the rest of the plays.

In the scene unigram vectorization manifold projection graphs (See Figure 11), we can see that some scenes from plays 11,15,19,22,23,24,34 quite a few times as outliers than the rest of the plays. These plays are The Taming, Two Gentlemen of Verona, The Life of King Henry V, The Third Part of King Henry the VI, The Life of King Henry the VIII, the Life and Death of King John, and Romeo and Juliet.

## Thoughts on Visualization and Clustering

We found PCA (both 2D and 3D) to be effective in identifying outliers. Also, we found that readability supports TDIDF results well. Our clustering shows consistent results for K-Means and GMM. DBSCAN, however, did not give a consistent result with others.

## Conclusion

From our clustering, we are suspicious of the authenticity of some scenes in historical plays. These plays are Anthony and Cleopatra, The 1st and 2nd Part of King Henry the IV, The Life of King Henry the V, The 1st, 2nd, and 3rd Part of King Henry the VI, The Life of King Henry the VIII, The Life and Death of King John, The Life and Death of Richard II, and The Life and Death of Richard the III. We also raised our suspicion in some parts of Romeo and Juliet due to its readability and the type token ratio.

## Additional Analyses (See RandomForest\_shakespear.py)

We also tried to classify the genre of the plays (i.e. historical, comedy, or tragedy). Our research question is “Can we teach a machine to understand comedy, historical, and tragedy plays?”

We obtain these labels from the MIT website. Since this is a classification problem, we tried two methods namely the Deep Belief Networks (DBN) and Random Forest (RF). In this case, our main focus is at the scene level. We first split the data (752 scenes) into 50% training and 50% testing data. Then we implemented DBN based on <https://github.com/larsmaaloee/deep-belief-nets-for-topic-modeling>. We trained a 2000-500-500-128 DBN using 100 epochs. The error rate for DBN is around 50-60% on the testing dataset which is quite good (See Figure 13).

For our random forest, we trained two RF models. One is using 10 classifiers and another one is a 100-classifier RF. The result in Figure 13 shows that 100-classifier RF is a little better than DBN (60-70% accuracy). The 10-classifier one performed much worse at 40-60%.

All results are quite good compared to a random pick at probability of 1/3. From what we looked at DBN, it has a potential to do better if we train with more epochs. However the algorithm is much slower and will be even slower if we use more epochs.

## Appendix 1: Figures

Figure 1: Number of components and explained variance ratio for Plays

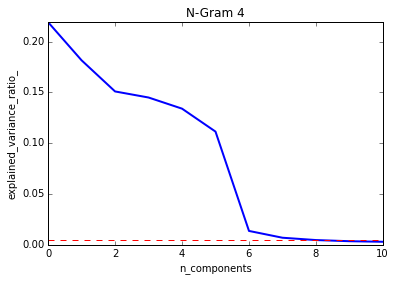
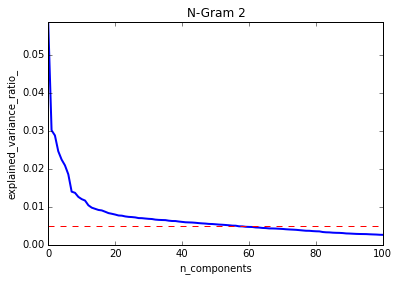
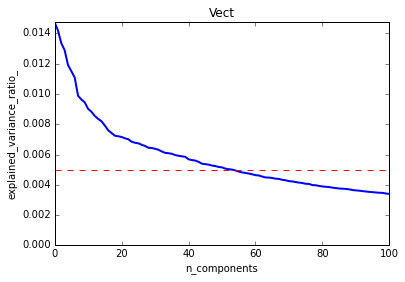


Figure 2: Number of components and explained variance ratio for Poems

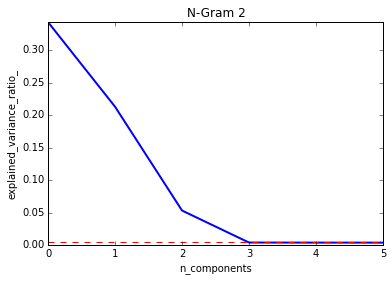
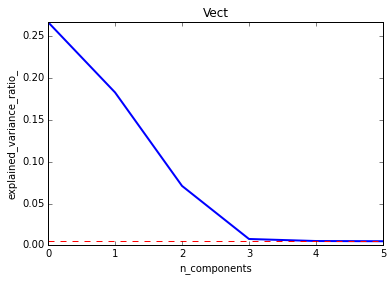


Figure 3: K-Means and GMM clustering (K = 2) of Poetry TFIDF

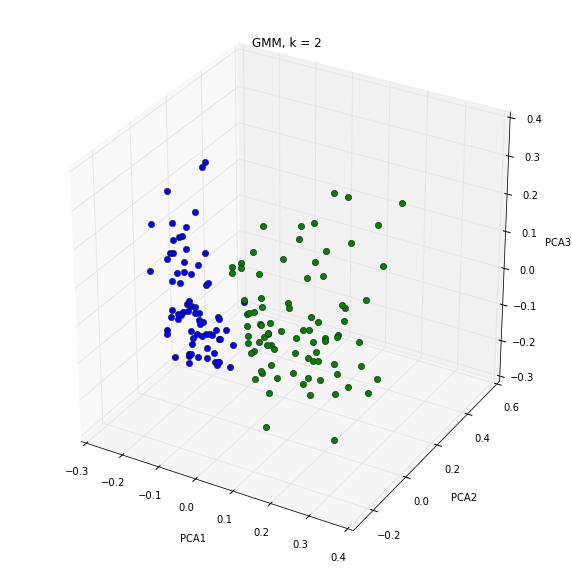
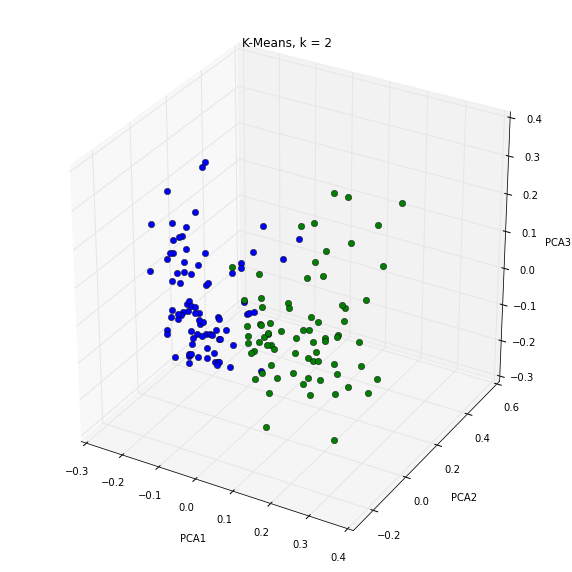


Figure 4: K-Means and GMM clustering (K = 2) of Poetry 2-gram

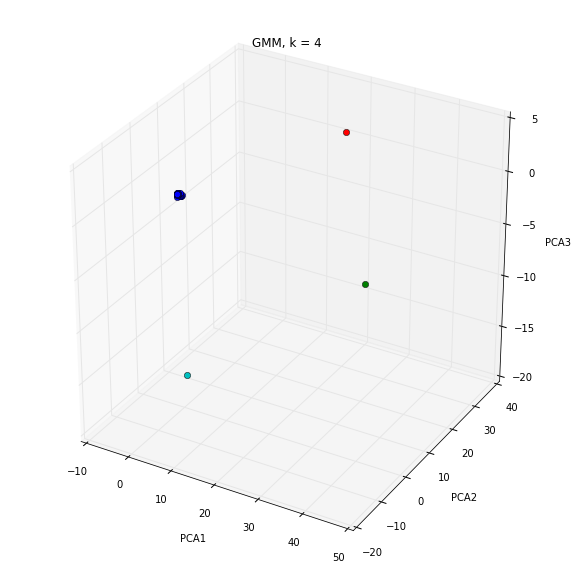
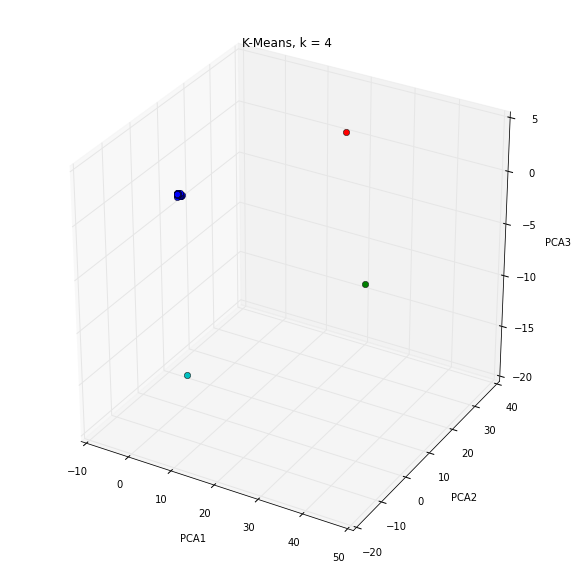


Figure 5: K-Means and GMM clustering (K = 3) of Plays TFIDF

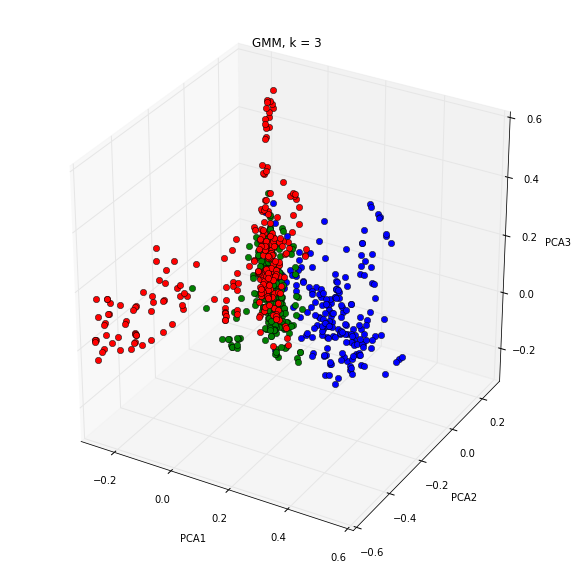
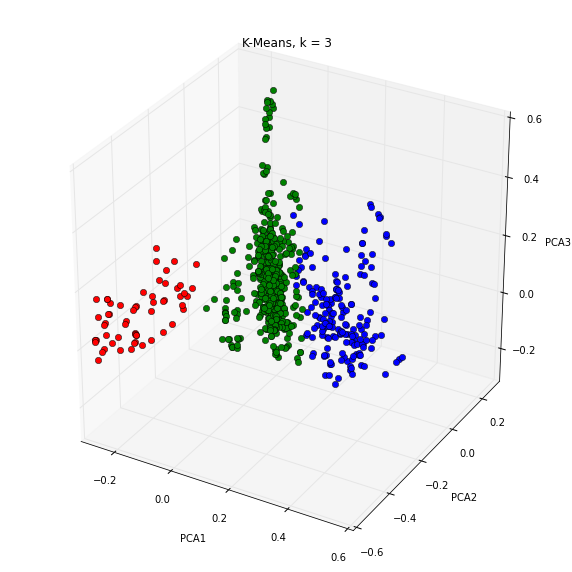


Figure 6: K-Means and GMM clustering (K = 3) of Plays 2-gram

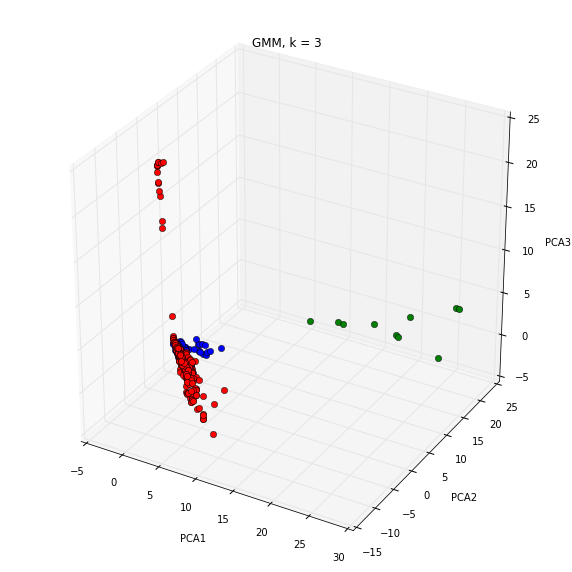
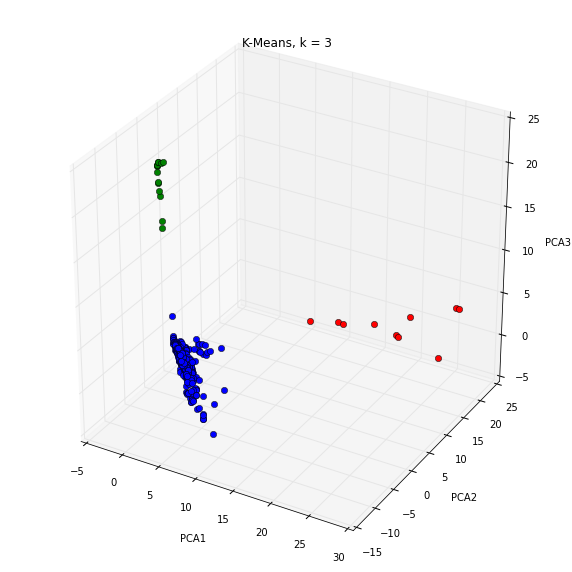
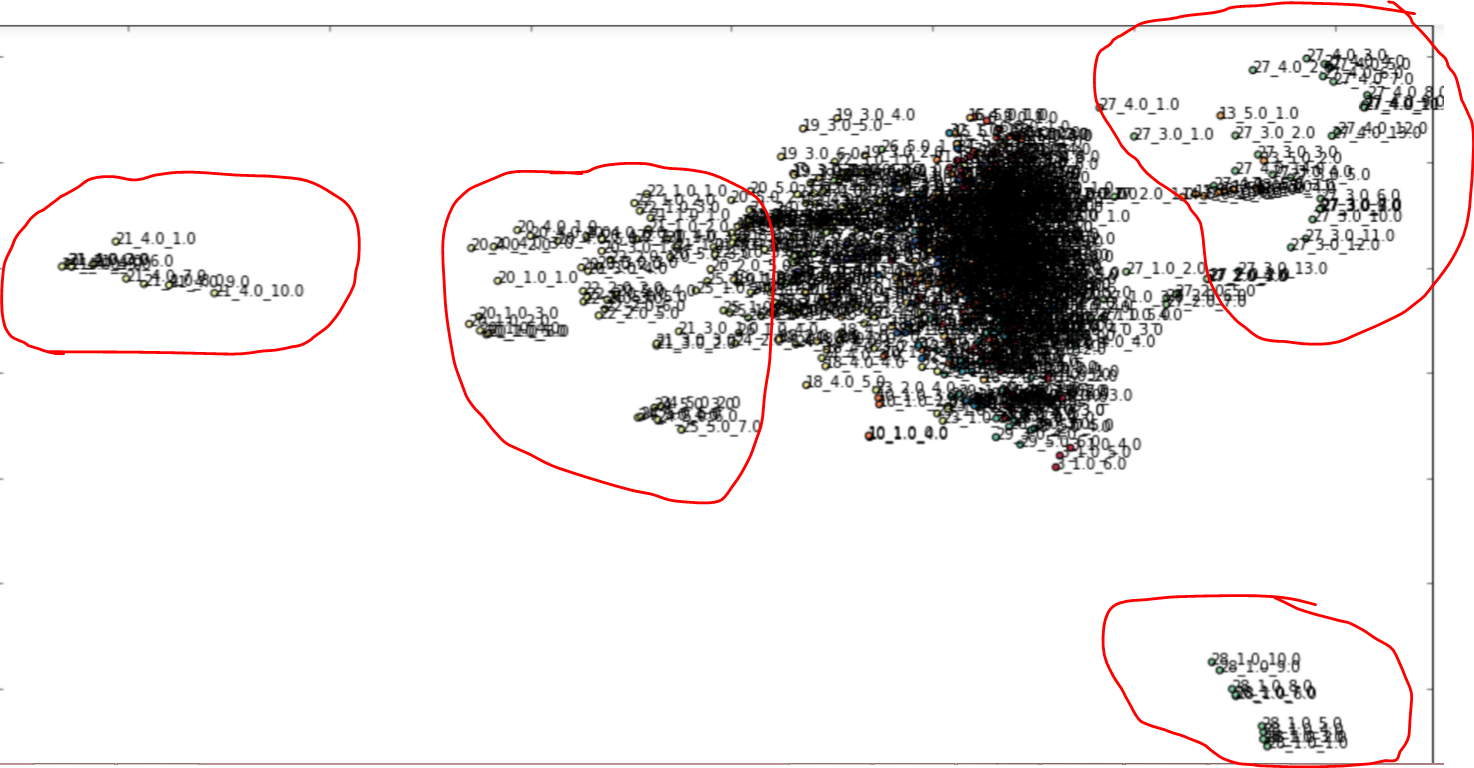


Figure 7: Two-Component PCA Projection of Scene TFIDF



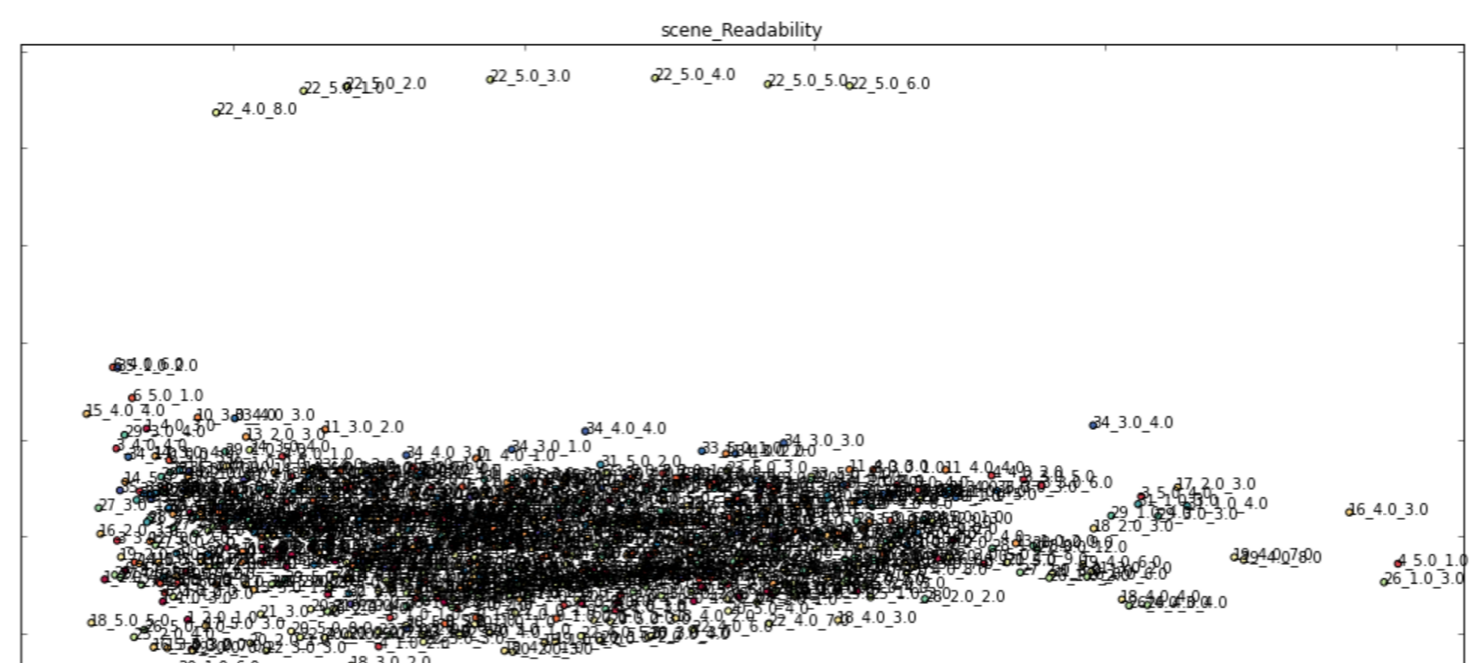
Some scenes from play 27

Some scenes from play 28

Some scenes from play 21

Some scenes from play 20,21,22,25

Figure 8: Two-Component PCA Projection of Scene Readability



Some scenes from play 22

Figure 9: Projection of Scene Type Token Ratio

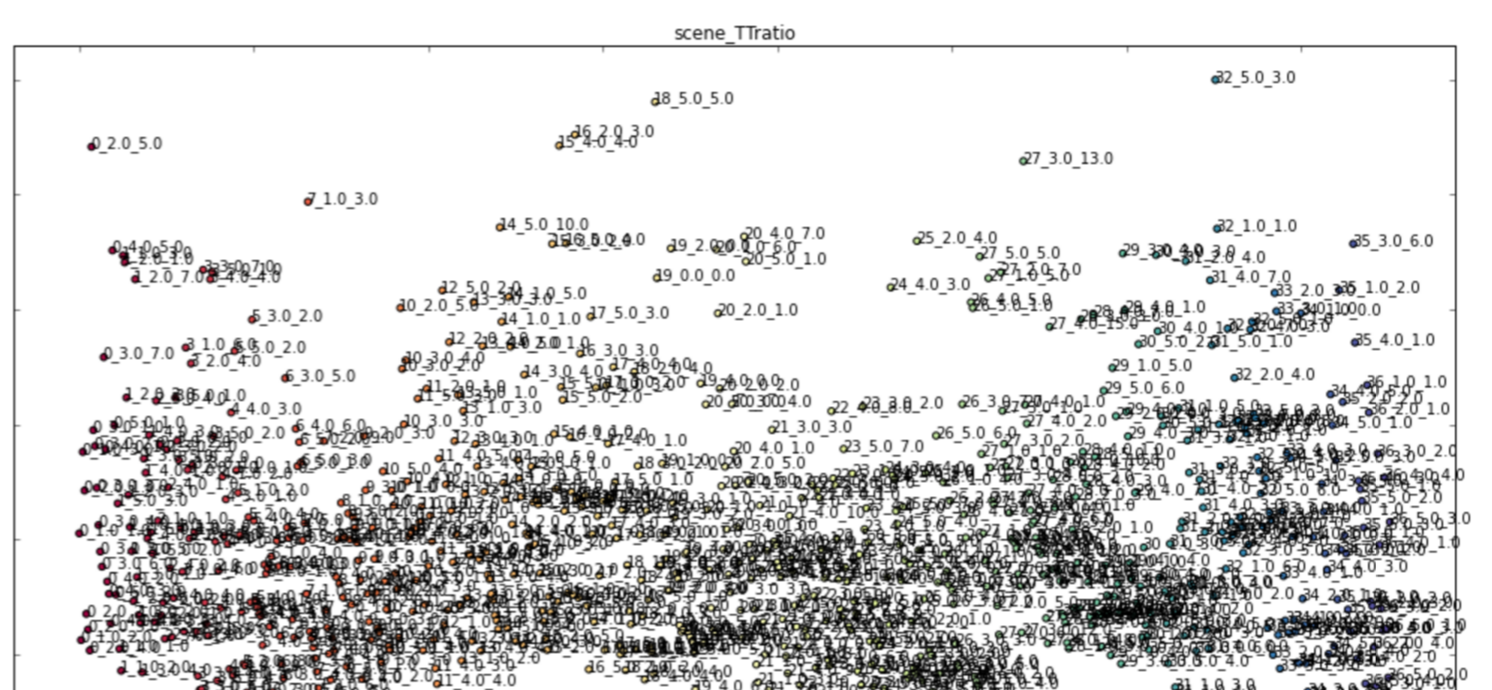


Figure 10: Various Manifold Projections of Act Readability

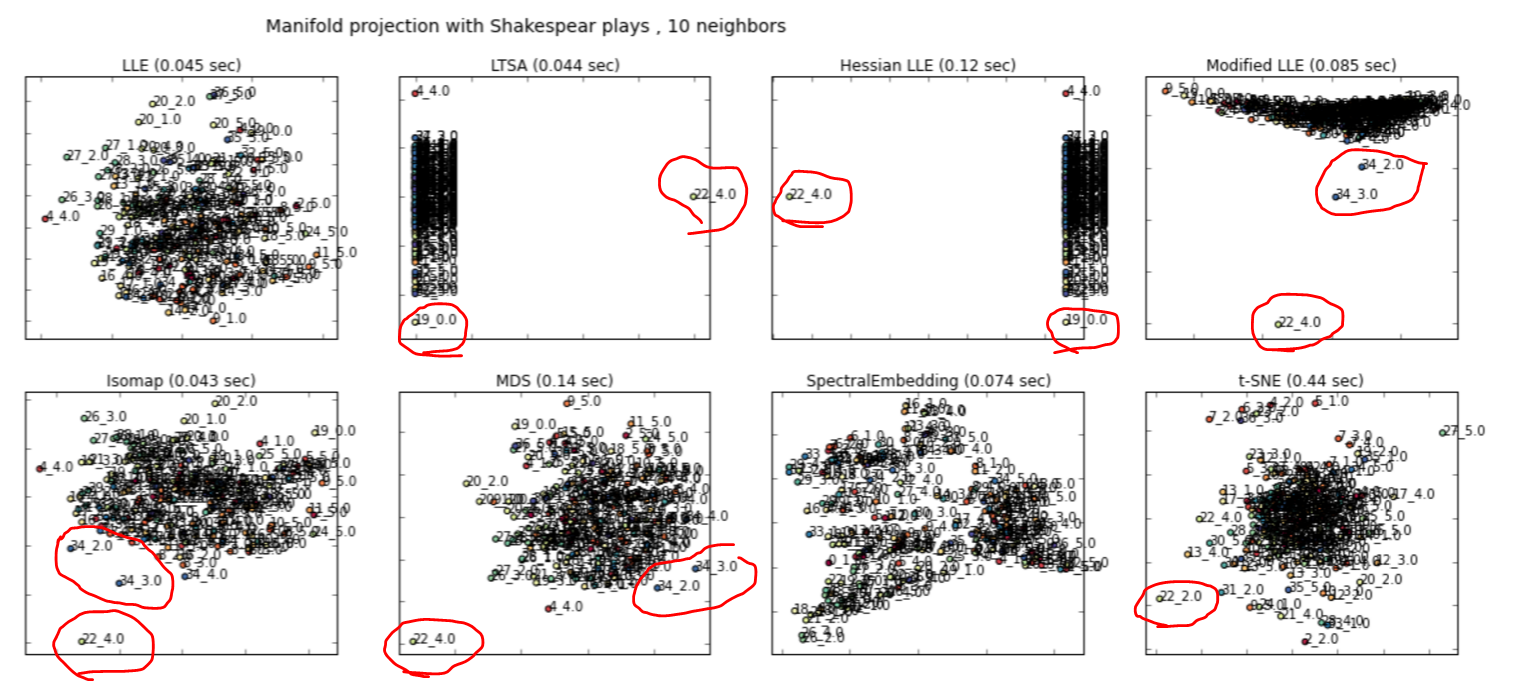


Figure 11: Various Manifold Projections of Act TFIDF

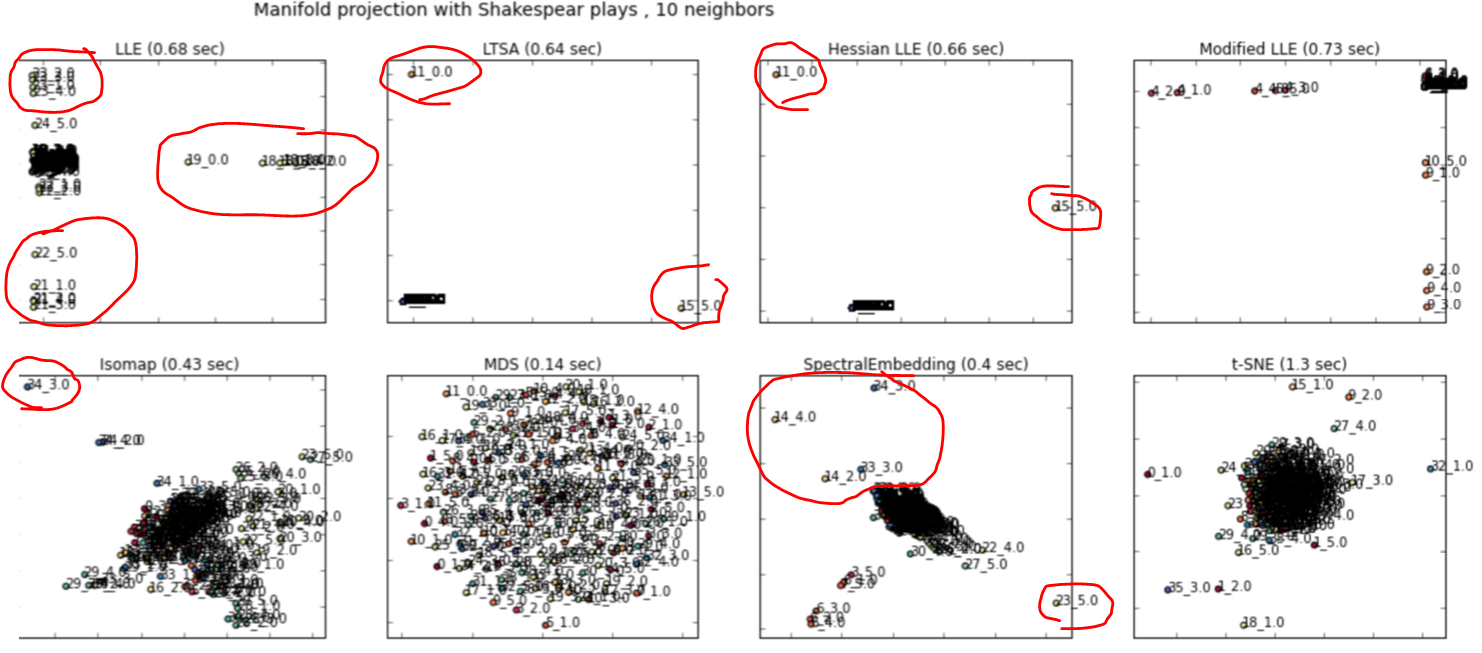
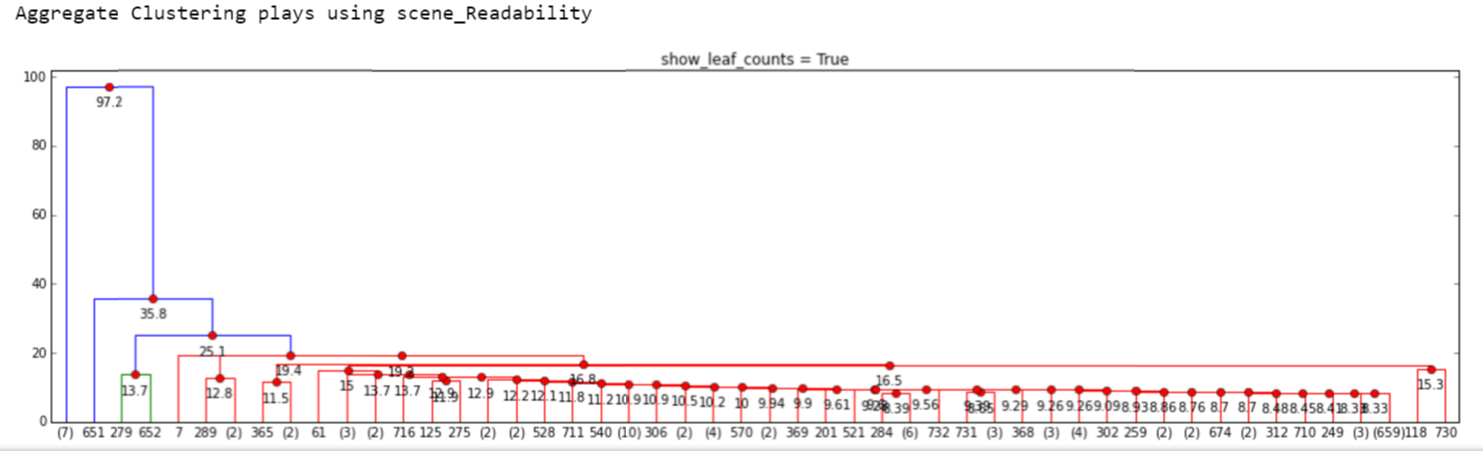


Figure 12: Dendrogram of Hierarchical Clustering of Scene Readability and Type Token Ratio



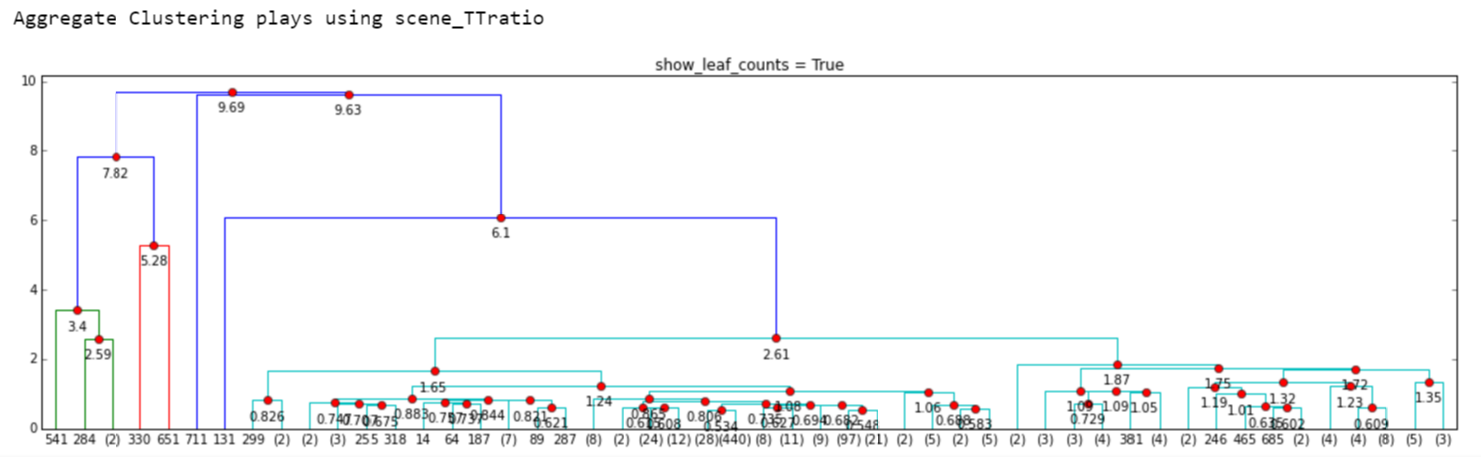


Figure 13: Accuracy of Classifying Shakespeare’s Plays (DBN vs RF)



## Appendix 2: Tables

Table 1: List of Plays

|  |  |
| --- | --- |
| 0 | all's well that ends well |
| 1 | as you like it |
| 2 | the comedy of errors |
| 3 | cymbeline |
| 4 | loves labours lost |
| 5 | measure for measure |
| 6 | the merry wives of windsor |
| 7 | the merchant of venice |
| 8 | a midsummer night's dream |
| 9 | much ado about nothing |
| 10 | pericles, prince of tyre |
| 11 | the taming of the shrew |
| 12 | the tempest |
| 13 | troilus and cressida |
| 14 | twelfth night |
| 15 | two gentlemen of verona |
| 16 | winter's tale |
| 17 | the first part of king henry the fourth |
| 18 | the second part of king henry the fourth |
| 19 | the life of king henry the fifth |
| 20 | the first part of king henry the sixth |
| 21 | the second part of king henry the sixth |
| 22 | the third part of king henry the sixth |
| 23 | the life of king henry the eighth |
| 24 | the life and death of king john |
| 25 | the life and death of richard the second |
| 26 | the life and death of richard the third |
| 27 | antony and cleopatra |
| 28 | the tragedy of coriolanus |
| 29 | the tragedy of hamlet, prince of denmark |
| 30 | the life and death of julius caesar |
| 31 | king lear |
| 32 | the tragedy of macbeth |
| 33 | othello, the moore of venice |
| 34 | romeo and juliet |
| 35 | timon of athens |
| 36 | titus andronicus |

## Appendix 3: Reproduction

## Technical Notes

The codes in this project are based on standard Python 3 libraries. We use Scikit-learn and nltk libraries for machine learning and natural language processing. If you find a difficulty running the codes, please visit the project Github at <https://github.com/dingchaoz/machine_learning/tree/master/HW4> and download the entire project.

## Data Folder

Please unzip the project in the same folder to prevent errors.