1. Cluster the plays

Since Shakespeare’ s plays are generally categorized into three types: comedies, histories, and tragedies. I use a K-Mean method with 3 clusters set up to try to cluster those plays. The experiment result is showed below (color marks represent the ground truth: **yellow - history, red - tragedy, green - comedy**).

'Henry IV', 'Antony and Cleopatra', 'Coriolanus', 'Hamlet', 'Richard II', 'Henry VI Part 2', 'Henry VIII', 'Richard III', 'Henry V', 'Troilus and Cressida', 'Henry VI Part 3', 'Cymbeline', 'King Lear'	'A Midsummer nights dream', 'King John', 'macbeth', 'Timon of Athens', 'The Tempest', 'Julius Caesar', 'A Comedy of Errors', 'Two Gentlemen of Verona', 'Henry VI Part 1', 'Pericles', 'Titus Andronicus'	'Alls well that ends well', 'Loves Labours Lost', 'Taming of the Shrew', 'Merry Wives of Windsor', 'Romeo and Juliet', 'A Winters Tale', 'As you like it', 'Measure for measure', 'Much Ado about nothing', 'Twelfth Night', 'Merchant of Venice', 'Othello'

As a result, the K-Mean model works relatively well (against the random classifier). Apparently, we can tell that the third column is for the **Comedies**, and the first column is for either **Histories** or **Tragedies**.

There are also some interesting findings based on the result. Firstly, I find there is no intersection of yellow and green marks in either first or third cluster. However, there are intersections among red, yellow and green marks in the second clusters. A possible assumption is that maybe **Comedies** and **Histories** are very different in general so that their vectors are significantly away from each other. Secondly, **Tragedies** seems to appear in all three clusters. Maybe they share many similar characteristics with the other two types so it’s hard for the model to categorize them correctly.

2. Most similar characters

In the experiment, I load all the character names into the program and create a term-character matrix, which is similar to the term-document one. However, by using the similarity functions an the character vectors, I'm able to compute the similarities among characters. This makes sense in a way because the characters that always say similar words can be categorized as similar (or have a similar personality or something).

For this part, I use three of the similarity functions written before and compute the most/least similar character of each character in the plays. Since the computation process is slow, I store some of the matrices information and pre-load them to save time.

The result is shown in the code, here I only provide part of it since it's significantly long and not reader-friendly. Because I personally know little about Shakespeare's plays, no explanation will be provided here.

using compute_cosine_similarity the most similar character pairs (A, B) are: KING HENRY IV ------ KING HENRY V WESTMORELAND ------ KING HENRY IV FALSTAFF ------ BENEDICK
using compute_jaccard_similarity the most similar character pairs (A, B) are: KING HENRY IV ------ KING JOHN WESTMORELAND ------ VERNON FALSTAFF ------ HAMLET
using compute_dice_similarity the most similar character pairs (A, B) are: KING HENRY IV ------ KING JOHN WESTMORELAND ------ VERNON FALSTAFF ------ HAMLET

3. Cluster the characters

Moreover, I create a K-Mean model to categorize the characters into two clusters, hoping that it can distinguish the gender. And the result is shown here. There is no too much information shows the gender of a specific character, but we can still easily see that the first cluster contains some of the male names (green marks), while the second cluster contains female names (pink marks).

'KING HENRY IV', 'WESTMORELAND', 'POINS', 'EARL OF WORCESTER', 'NORTHUMBERLAND', 'BARDOLPH', 'First Traveller', 'Servant', 'Vintner', 'MORTIMER', 'GLENDOWER', 'VERNON', 'WORCESTER', 'ARCHBISHOP OF YORK', 'SIR MICHAEL'	'FALSTAFF', 'PRINCE HENRY', 'HOTSPUR', 'SIR WALTER BLUNT', 'First Carrier', 'Ostler', 'Second Carrier', 'GADSHILL', 'Chamberlain', 'PETO', 'Thieves', 'Travellers', 'LADY PERCY', 'FRANCIS', 'Hostess', 'Sheriff', 'Carrier', 'EARL OF DOUGLAS', 'Messenger', 'LANCASTER', 'BEDFORD'