# HW2: To be, or not to be

## Report

Pushkar Singh Negi

Ku ID: 2946319

EECS 731: Introduction to Data Science

## Notebook1: Deduced Additional Information, visualization and classification model: HW_2_Shakespeare_data_new.ipynb

**Input raw dataset** : Shakespeare_data.csv
**Processed dataset** : Shakespeare_ds_importantPlayer.csv,
Shakespeare_ds_numberOfWordsCol.csv,
Shakespeare_ds_Most_common_word.xlsx,
**Jupyter Notebook File name** : HW_2_Shakespeare_data_new.ipynb

1. Imported the Shakespeare_data.csv file using pandas in dataframe.
2. Checked the total non-null entries to handle missing values.
3. Replaced NaN values in Player column to Uknonw.
4. Analyzed the dataset with the help of various commands, such finding the total unique players in the Player column.
5. Next, I found out the **additional information#1** i.e. for **each Play, number of lines (PlayerLine) spoken by each Player** with the help of groupby feature.
6. Next, I applied various pandas feature to convert the above result dataset into a panda frame.

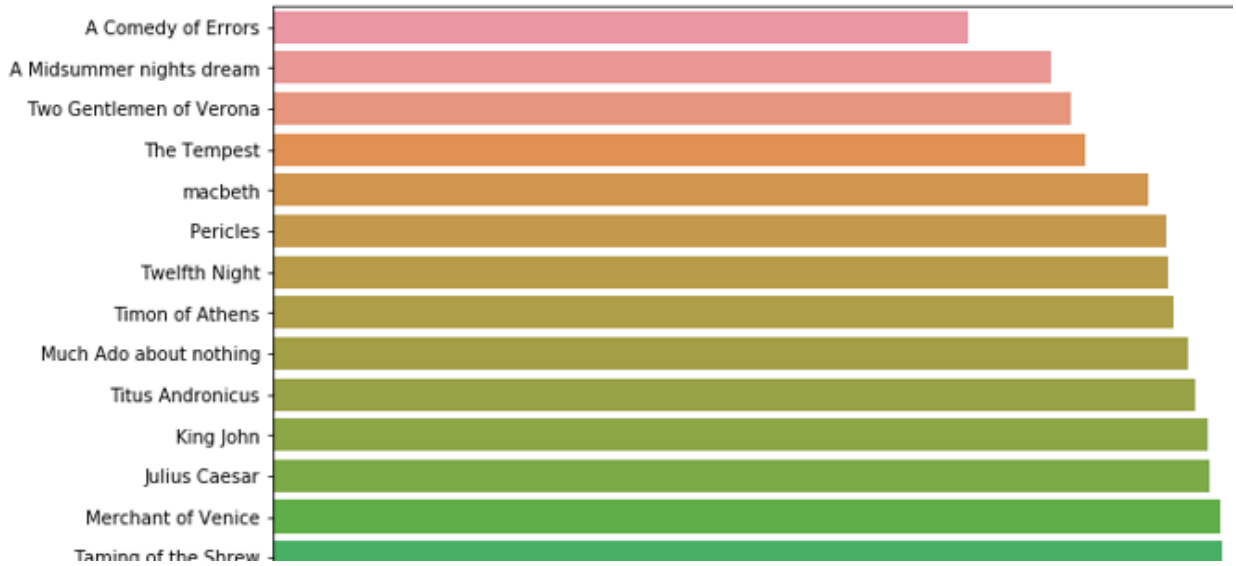
Out[16]:

| Play | Player | PlayerLine |
|---|---|---|
| A Comedy of Errors | ADRIANA | 284 |
| | AEGEON | 150 |
| | AEMELIA | 75 |
| | ANGELO | 99 |
| | ANTIPHOLUS | 6 |
| | BALTHAZAR | 31 |
| | Courtezan | 43 |
| | DROMIO OF EPHESUS | 191 |
| | DROMIO OF SYRACUSE | 323 |
| | DUKE SOLINUS | 97 |
| | First Merchant | 19 |

7. Next, I found out the **additional information#2** i.e. t**o count the number of PlayerLine corresponding to each Play** with the help of groupby feature.
8. Next, I applied indexing and converted the resulted dataset into a frame.
9. After sorting the values on the basis of PlayerLine in the above dataset, we got to know an additional information that which Play has the maximum number of line and which all play are more important than the other.

| | | |
|---|---|---|
| **A Winters Tale** | 3489 | A Winters Tale |
| **Troilus and Cressida** | 3711 | Troilus and Cressida |
| **Othello** | 3762 | Othello |
| **King Lear** | 3766 | King Lear |
| **Antony and Cleopatra** | 3862 | Antony and Cleopatra |
| **Richard III** | 3941 | Richard III |
| **Cymbeline** | 3958 | Cymbeline |
| **Coriolanus** | 3992 | Coriolanus |
| **Hamlet** | 4244 | Hamlet |

10. **Visualizations:** Plotted a graph to show: PlayerLine against Name of the Play

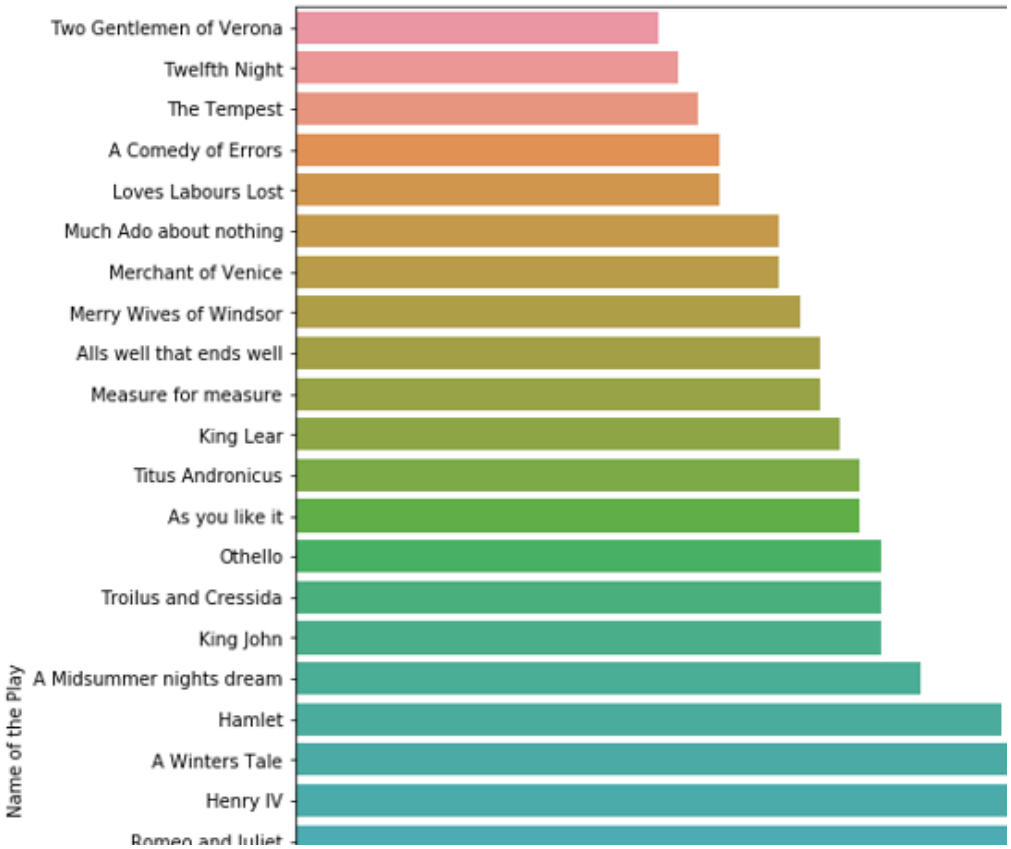



11. Next, I found out the **additional information#3** i.e. **Number of Players corresponding to each Play**.
12. Applied, transformations and found an additional information: play that has maximum and minimum player

Out[38]:

| | Number of Players | Name of the Play |
|---|---|---|
| 0 | 18 | Two Gentlemen of Verona |
| 1 | 19 | Twelfth Night |
| 2 | 20 | The Tempest |
| 3 | 21 | A Comedy of Errors |
| 4 | 21 | Loves Labours Lost |
| 5 | 24 | Much Ado about nothing |
| 6 | 24 | Merchant of Venice |
| 7 | 25 | Merry Wives of Windsor |
| 8 | 26 | Alls well that ends well |
| 9 | 26 | Measure for measure |
| 10 | 27 | King Lear |
| 11 | 28 | Titus Andronicus |
| 12 | 28 | As you like it |

**13. Visualizations: Plotted a graph between: Name of the Play and Number of Players**



14. **Additional Information #4:** On the basis of number of words in each PlayerLine correspondong to each Player, found the the **Player that spoke the maximum number of words**, and hence is the **important/ or the player that has spent most time in the play**.

15. **My findings --> Player named GLOUCESTER has maximum number of 14319 total words** in all the PlayerLine, and hence the **important/main.**

16. **Visualization:** Plotted a graph between Number of words in the PlayerLine against Player.

17. **Saved** the resulted dataframe in .csv and have checked in git repo.

18. **Additional Information #5: Made a list of most frequent distinct words used in the play by their occurance.**

```
Out[79]: dict_items([('act', 249), ('i', 18949), ('scene', 755), ('i.', 257), ('london.', 45), ('the', 26991), ('palace.', 116), ('ent
         er', 1953), ('king', 888), ('henry,', 54), ('lord', 782), ('john', 122), ('of', 15697), ('lancaster,', 29), ('earl', 82), ('w
         estmoreland,', 14), ('sir', 439), ('walter', 17), ('blunt,', 19), ('and', 24245), ('others', 118), ('so', 3645), ('shaken',
         3), ('as', 5441), ('we', 3172), ('are,', 93), ('wan', 2), ('with', 7342), ('care,', 32), ('find', 474), ('a', 13907), ('tim
         e', 664), ('for', 7034), ('frighted', 12), ('peace', 162), ('to', 18129), ('pant,', 1), ('breathe', 48), ('short-winded', 1),
         ('accents', 5), ('new', 158), ('broils', 5), ('be', 6260), ('commenced', 2), ('in', 10212), ('strands', 1), ('afar', 19), ('r
         emote.', 1), ('no', 2753), ('more', 1763), ('thirsty', 6), ('entrance', 13), ('this', 5704), ('soil', 13), ('shall', 3351),
         ('daub', 3), ('her', 3001), ('lips', 65), ('own', 607), ("children's", 12), ('blood,', 170), ('nor', 914), ('trenching', 1),
```

19. Saved the resulted dataframe in .csv and have checked in git repo.

20. Did **visualization for each player against number of lines.**

21. **LOGISTIC REGRESSION:** In order to apply logistic regression model, changed the datatype of all attribute to int.

22. Splited the data into testing and training data with the help of sklearn.model_selection import train_test_split

23. Fitted the X_train and y_train set

24. Applied the predictions and calculated the accuracy.

## Notebook2: Deduced Additional Information, visualization and classification model: Shakespeare_notebook2.ipynb

In this notebook also I have continued working on the training and test dataset and applied logistic regression approach and for various combination of features (attributes) and also used additional information for the logistic regression model.