**Predicting Bike Rental Count**

**Preeti Chauhan**

**30 November 2018**

1. INTRODUCTION
   1. **Problem Statement**

Here is a list of 154 sonnets from Shakespeare -

https://docs.google.com/document/d/10lgjloePyqrzSMqB3JkNMSZWsivUzlXL84q3pV2s954/edit

1. You need to tag each sonnet with a keyword(most important, and frequently occurring)

2. You need to build an algorithm in python to show related sonnets(recommendation engine sort of)

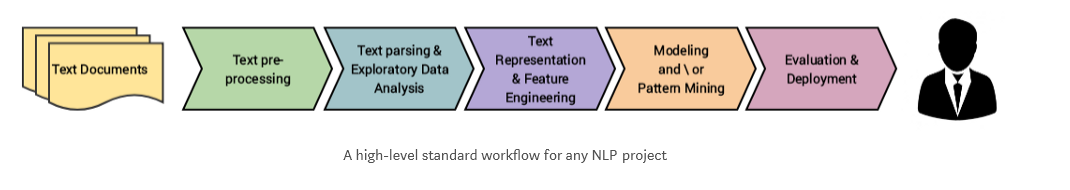
* 1. **Data**

Data here is a text file downloaded from the link given above Shakespeare\_Sonnets.txt

1. PROJECT APPROACH

To execute this project, I have gone with the general approach for most of the NLP projects(skipped few steps, wherever not required)

* 1. **General Approach**



1. DETAILED PROJECT IMPLEMENTATION

Below are the different phases of the implementation of project

* 1. **Define the Project ROADMAP**

I have sketched the plan to implement the project in different phases, taking cue from general approach.

The whole project is divided in 6 phases (and further subphases). Below are the phases defined.

* Define and categorize problem statement
* Gather the data
* Perform Exploratory Data Analysis
* Prepare data for consumption
* Finding most frequent/important words in all sonnets
* Building Content based recommender system
  1. **Implement the Project ROADMAP**

As per the above roadmap, let’s start the project, exploring each phase.

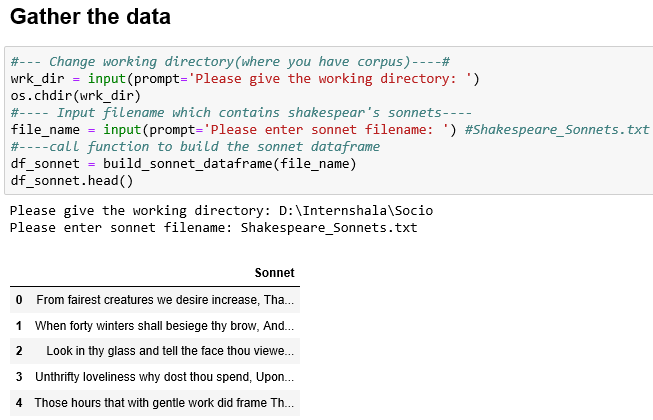
* + 1. **Categorize Problem**

As per the problem statement defined above, this is clearly an NLP problem, the end goal of which is to build a **content-based recommender system**.

* + 1. **Gather the Data**

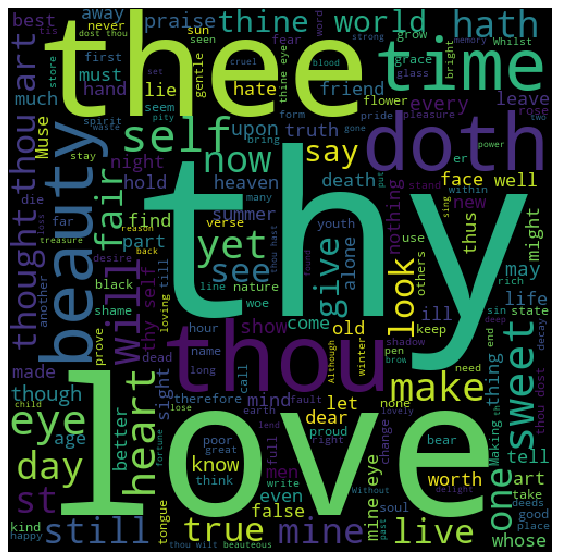
The data is given to us on platter. Simply import the data and have a glance.

Store the text imported in to a dataframe.



* + 1. **Perform Exploratory Data Analysis**

Here, I am just checking the most frequent words in entire corpus(set of all sonnets) using wordcloud.



Seems like entire corpus has word frequently occurring words like ‘thee’, ‘thou’, ‘thy’ which does not define any given sonnet uniquely.

Hence, I’ll remove these words during preprocessing of corpus for better results.

* + 1. **Finding Most important/frequent words in a sonnet**
       1. **Data Preprocessing**

Let’s prepare the corpus for the above problem set by following below data preprocessing steps:

* Remove ‘numbers’ from the text
* Remove punctuation marks
* Remove stopwords.
* Remove any leading trailing spaces
* Remove common words like ‘thou’,’thee’,’thy’

**Note**: Here I am not using ‘Stemming’, ‘Lemmetization’ etc., because it is not required here.

Say , I have a sentence ‘Giving is the most satisfying thing ever.’. So, if I’ll have to title this text, I would rather title it as ‘Giving’ then ‘Give’(which is the stemmed word). So, I am leaving the words as such, instead of converting all words to base words.

**Note**: for the last step, i.e. to remove most common words which do not uniquely define any sonnet, we can follow another approach as well.

**Alternate approach:** Use nltk part of speech tagging to identify the part of speech in any sonnet and remove particular tagged words like verb etc. and retain only certain parts of speech like noun, adjectives which are more apt for a sonnet title.

I am just removing the hardcoded common words(obtained from wordcloud), to keep things simple.

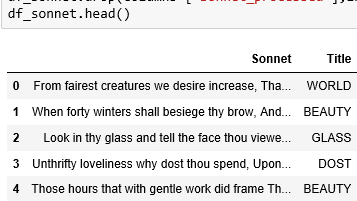
The processed text is stored in new column ‘Sonnet\_processed ‘

* + - 1. **Finding most frequent words**

Now we need to work on preprocessed text for further operations. Follow below steps:

* Convert the processed text to vectors(using TFIDF, since it is more relevant here)
* Create bag of words
* Add up the occurrences of each word from bag of words
* Sort the result by number of occurrences
* Select the top result and store it in ‘Title’

**Title here will be the answer to the first problem: that is, it is the most frequently occurring word in any sonnet which uniquely define that sonnet.**

****

* + 1. **Building Content based recommender system for Shakespeare Sonnets**

Lets build the recommender system now. Since we are recommending the sonnets which are similar to the sonnet selected, ie., we are building recommender system based on the similarity between the products/content, it is called Content based recommender system.

* Let’s not use the ‘processed\_text’ we have already used for previous use case.
* Here, I am using Rake() function from Rake\_nltk library to preprocess the corpus(sonnet)
* **Rake().extract\_keywords\_from\_text** will automatically remove punctuation, stopwords etc, and will give you set of words, which are actually **keywords** for that text.

**Note**: I have not used this function for previous use case because, it de-duplicate the words, which means not suitable where frequency of each word is important.

* Store the keywords in ‘Keyword’ column.
* Vectorize ‘Keywords’.
* Fit, transform the vector above to create a matrix.
* Use cosine\_similarity() function to create a similarity matrix for all the sonnet.
* So, if there are n num of sonnets then , the similarity matrix will of [n X n size), having similarity scores.
* To find the similar sonnets, pick the series from similarities matrix(for any sonnet) and sort it on scores.
* From above series, you can get the indexes of all similar sonnets sorted and can pick top n similar sonnets accordingly.

**This is how I build the basic content based recommender system for Shakespeare’s sonnet.**

