NATURAL LANGUAGE PROCESSING, TOPIC MODELING

2018380130 - Khan Md Shahedul Islam

## Table of Contents

[**Abstract** 2](#_Toc8399)

[**Introduction** 2](#_Toc8400)

[**Requirements** 3](#_Toc8401)

[**IDE** 3](#_Toc8402)

[**Technical description** 4](#_Toc8403)

[**Working Method** 7](#_Toc8404)

[**Conclusion** 18](#_Toc8405)



# Abstract

Topic modeling is an unsupervised machine learning technique that’s capable of scanning a set of documents, detecting word and phrase patterns within them, and automatically clustering word groups and similar expressions that best characterize a set of documents.

This day we are hearing a lot about artificial intelligence, along with terms like machine learning and Natural Language Processing (NLP). Especially working in a company that processes hundreds, or even thousands of customer interactions every day. Data analysis of social media posts, emails, chats, open-ended survey responses, and more, is not an easy task, and less so when delegated to humans alone.

That’s why the world is excited about the implications artificial intelligence could have on day-to-day tasks, as well as on businesses as a whole. AIpowered text analysis uses a wide variety of methods or algorithms to process language naturally, one of which is topic analysis – used to automatically detect topics from texts.

By using topic analysis models, businesses are able to offload simple tasks onto machines instead of overloading employees with too much data. We can imagine the time a team could save and spend on more important tasks, if a machine was able to sort through endless lists of customer surveys or support tickets every morning.

# Introduction

On our project we have worked with datasets (performed three key algorithms that are widely used for topic analysis. We have used a dataset containing three famous writer’s sentences from their books (Edgar Allen Poe, H.P. Lovecroft, Mary Wollstonecraft Shelley). Then we have implied those three algorithms and got incredible (above 85th percentiles) results.

we have designed our algorithms (Naïve Bayes Classifiers, Logistic regression, xgboost classifiers) in an efficient and sophisticated way that this works with short memory and gpu power and generate great result. For data visualization we have used word cloud library. For

# Requirements

* high cuda cores processor
* dedicated graphics is a plus
* NLTK(should be pre included)
* xgboost library
* word cloud
* tensor flow
* matplotlib
* Keras
* Numpy and Pandas

# IDE

We have used anaconda and jupyterLab(Shahed)+Jupyter

Notebook(Amirbek) for writing our cell codes. And we used chrome to connect.

For downloading libs we used pip download.

* Anconda(3)
* Jupyter Notebook(py3)/JupyterLab(py3) - Chrome

# Technical description

**WordCloud** is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.

For generating word cloud in Python, modules needed are – matplotlib, pandas and wordcloud. To install these packages, we needed to run the following commands: **pip install matplotlib pip install pandas pip install wordcloud**

The dataset used for generating word cloud is collected from Kaggle data Repository.

**Python | Lemmatization with NLTK**

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.

Text preprocessing includes both Stemming as well as Lemmatization. Many times people find these two terms confusing. Some treat these two as same. Actually, lemmatization is preferred over Stemming because lemmatization does morphological analysis of the words.

**Applications of lemmatization are:**

* Used in comprehensive retrieval systems like search engines.
* Used in compact indexing

**Examples of lemmatization:**

-> rocks : rock

-> corpora : corpus

-> better : good

One major difference with stemming is that lemmatize takes a part of speech parameter, “pos” If not supplied, the default is “noun.”

**What is Vectorization ?**

*Vectorization* is used to speed up the Python code without using loop. Using such a function can help in minimizing the running time of code efficiently.

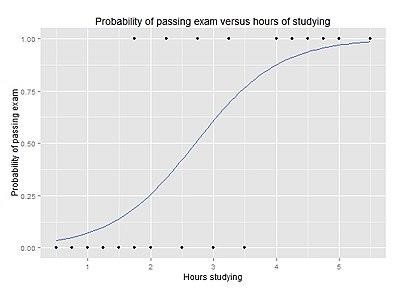
Various operations are being performed over vector such as dot product of vectors which is also known as scalar product as it produces single output, outer products which results in square matrix of dimension equal to length X length of the vectors, Element wise multiplication which products the element of same indexes and dimension of the matrix remain unchanged.

We will see how the classic methods are more time consuming than using some standard function by calculating their processing time.

**What is the Logistic regression**?

**Logistic regression** is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

**Example:**

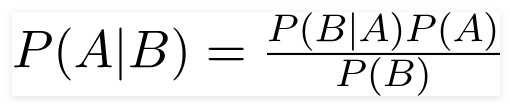


Graph of a logistic regression curve showing probability of passing an exam versus hours studying

**Principle of Naive Bayes Classifier:**

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem.

**Bayes Theorem:**



Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called **naive**.

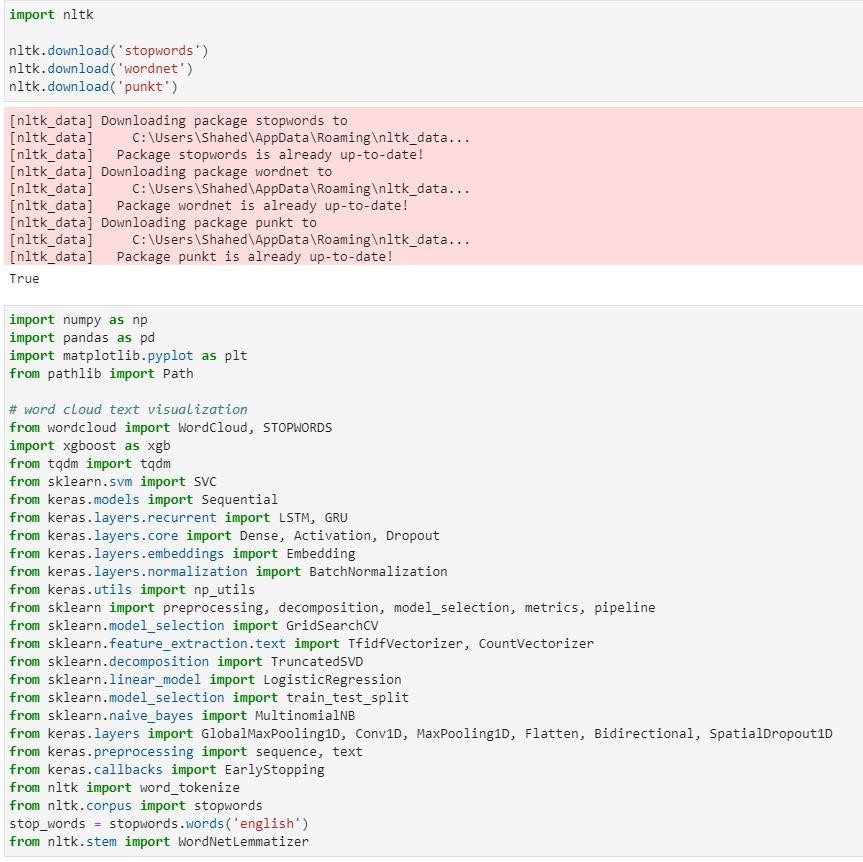
**XGBoost** is one of the most popular machine learning algorithm these days. Regardless of the type of prediction task at hand; regression or classification.

It is well known to provide better solutions than other machine learning algorithms. In fact, since its inception, it has become the "**state-of-the-art**” machine learning algorithm to deal with structured data.

# Working Method

For topic modelling/analysis the most used library is NLTK(we downloaded package-‘stopwords’,’wordcount’ and ‘punkt’). Then we have imported all the

libraries we need to use.



**Data Description**

The competition dataset contains text from works of fiction written by spooky authors of the public domain: Edgar Allan Poe, HP Lovecraft and Mary Shelley. The data was prepared by chunking larger texts into sentences using CoreNLP's MaxEnt sentence tokenizer, so we may notice the odd nonsentence here and there. Our objective is to accurately identify the author of the sentences in the test set.

**File descriptions** train.csv - the training set test.csv - the test set sample\_submission.csv - a sample submission file in the correct format

**Data fields**

* id - a unique identifier for each sentence
* text - some text written by one of the authors
* author - the author of the sentence (EAP: Edgar Allan Poe, HPL: HP

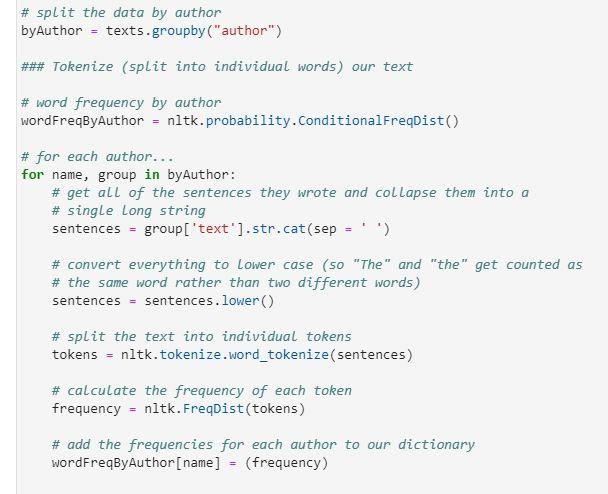
Lovecraft; MWS: Mary Wollstonecraft Shelley)

Explanatory Data Analysis:



Finding out how often which Authors used which words:

After performing tokenization we get a dictionary where each entry is the frequency distribution of words for a specific author.

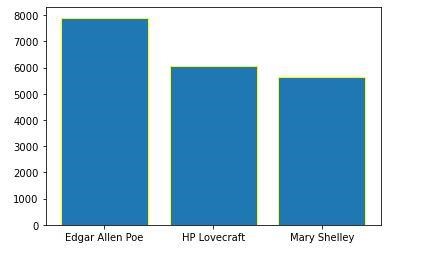


Now we can look at how often each writer uses specific words. Since we are students and, in my team, we are friends and we dream of a corona free world, how about 'student',' friends' and dream'?

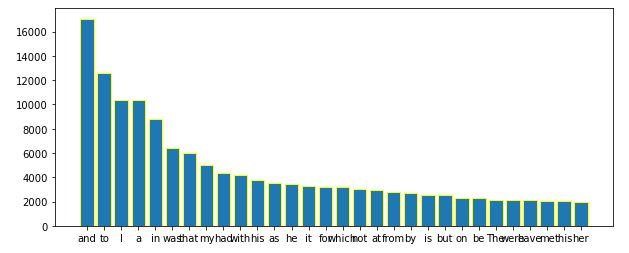


Let’s perform plot count(function to return a bar plot for each author count appears in the dataset-train.csv) and visualization with wordcloud.

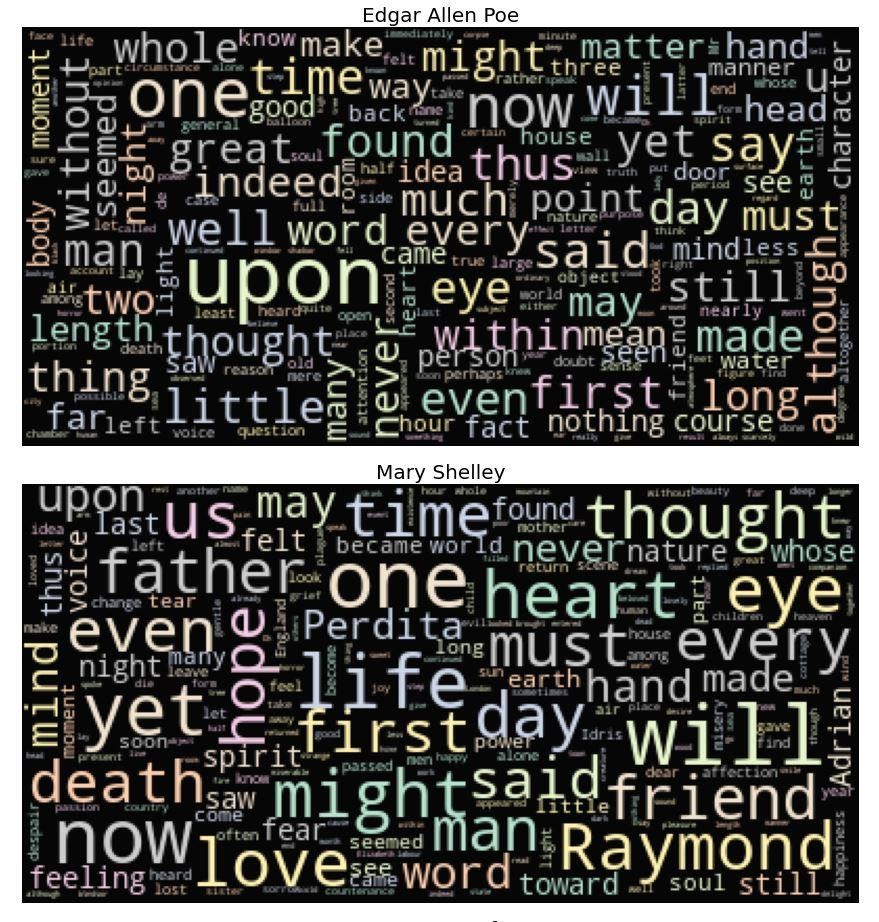
#def plot count():

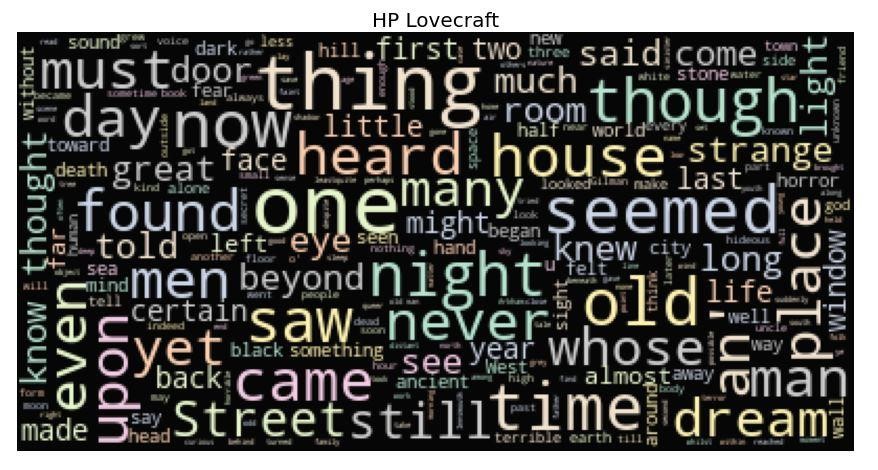


#def plot words count():This function will return a bar plot for first 30 words counts appears in dataset



# def word\_cloud\_viz(): This function will return wordcloud visualization of words for each author text.

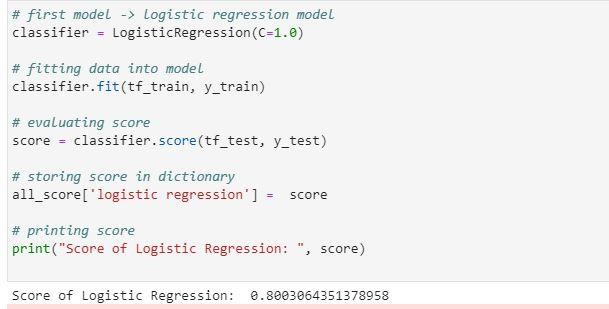




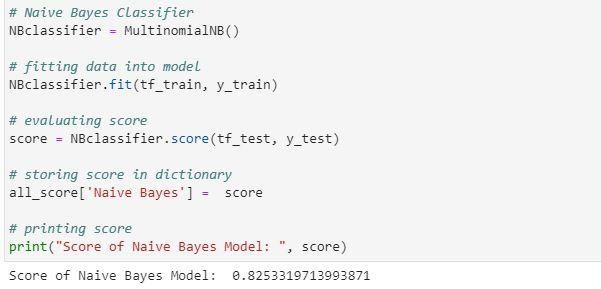
#invoking **Lemmatizer** from NLTK. Here we are customizing sklearn count vectorizer class. We are including lemmatizer along with it so that our preprocessing can be done with one place.



#Perrmoing **Logistic Regression**



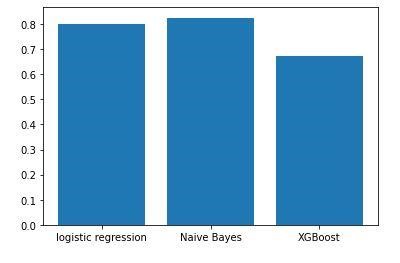
**#** Performing **Naïve Bayes Classifier**



**#**Performing xgboost classifier



Performance comparison between this three:



# Conclusion

This project was really fun to work with and at the same time it was really challenging. We have learned a lot from each other and explored the surface of the world of NLP. Being a team of machine learning enthusiast, we hope this project can be where we begin our journey. Thanks so much to our dear laoshi to take the time put the effort to explain everything to us clearly and always encouraging us.

**Thank you!**