**Text Analysis**

* In this project, I performed ‘Text analysis’ of approx 170 articles available on the internet.
* Firstly, import the required libraries and modules like pandas, BeautifulSoap, nltk, gensim.
* Convert the input.xlsx file into a data frame and extract the needed data viz heading and body from the URL.
* Performed data cleaning by removing stop words and other necessary symbols.
* Created a dictionary of positive and negative words.
* Then performed sentimental analysis by calculating positive, negative, polarity, and subjectivity scores. Then performed data analysis by calculating [Analysis of Readability](https://docs.google.com/document/d/11FuBgszZwCSpVWekJ6rR5tBLjU--xfIC/edit#heading=h.z337ya), [Complex Word Coun](https://docs.google.com/document/d/11FuBgszZwCSpVWekJ6rR5tBLjU--xfIC/edit#heading=h.1y810tw)t, [Syllable Count Per Word](https://docs.google.com/document/d/11FuBgszZwCSpVWekJ6rR5tBLjU--xfIC/edit#heading=h.2xcytpi), etc.

Libraries used :

* **Pandas** - Pandas is mainly used for data analysis and associated manipulation of tabular data in Dataframes. Pandas allows importing data from various file formats such as comma-separated-values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel.
* **BeautifulSoap** - Beautiful Soup is a Python library that is used for web scraping purposes to pull the data out of HTML and XML files. It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.
* **Nltk** - NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc

## Text Processing steps:

1. Tokenization
2. Lower case conversion
3. Stop Words removal
4. Stemming
5. Lemmatization
6. Parse tree or Syntax Tree generation
7. POS Tagging

* **Gensim** - Gensim is implemented in Python and Cython for performance. Gensim is designed **to handle large text collections using data streaming and incremental online algorithms**, which differentiates it from most other machine learning software packages that target only in-memory processing.

**Text Analysis**

Objective of this document is to explain methodology adopted to perform text analysis to drive sentimental opinion, sentiment scores, readability, passive words, personal pronouns and etc.

Table of Contents

[**1**](#_35nkun2) **Sentimental Analysis** 2

[**1.1**](#_1ksv4uv) **Cleaning using Stop Words Lists** 2

[**1.2**](#_44sinio) **Creating dictionary of Positive and Negative words** 2

[**1.3**](#_2jxsxqh) **Extracting Derived variables** 2

[**2**](#_z337ya) **Analysis of Readability** 3

[**3**](#_3j2qqm3) **Average Number of Words Per Sentence** 3

[**4**](#_1y810tw) **Complex Word Count** 3

[**5**](#_4i7ojhp) **Word Count** 3

[**6**](#_2xcytpi) **Syllable Count Per Word** 4

[**7**](#_1ci93xb) **Personal Pronouns** 4

[**8**](#_3whwml4) **Average Word Length** 4

# **Sentimental Analysis**

Sentimental analysis is the process of determining whether a piece of writing is positive, negative or neutral. The below Algorithm is designed for use on Financial Texts. It consists of steps:

## **Cleaning using Stop Words Lists**

The Stop Words Lists (found [here](http://www3.nd.edu/~mcdonald/Word_Lists.html)) are used to clean the text so that Sentiment Analysis can be performed by excluding the words found in Stop Words List. Use this url if above does not work <https://sraf.nd.edu/textual-analysis/resources/>

## **Creating dictionary of Positive and Negative words**

The Master Dictionary (found [here](http://www3.nd.edu/~mcdonald/Word_Lists.html)) is used for creating a dictionary of Positive and Negative words. We add only those words in the dictionary if they are not found in the Stop Words Lists. Use this url if above does not work <https://sraf.nd.edu/textual-analysis/resources/>

## **Extracting Derived variables**

We convert the text into a list of tokens using the nltk tokenize module and use these tokens to calculate the 4 variables described below:

**Positive Score**: This score is calculated by assigning the value of +1 for each word if found in the Positive Dictionary and then adding up all the values.

**Negative Score**: This score is calculated by assigning the value of -1 for each word if found in the Negative Dictionary and then adding up all the values. We multiply the score with -1 so that the score is a positive number.

**Polarity Score**: This is the score that determines if a given text is positive or negative in nature. It is calculated by using the formula:

Polarity Score = (Positive Score – Negative Score)/ ((Positive Score + Negative Score) + 0.000001)

Range is from -1 to +1

**Subjectivity Score**: This is the score that determines if a given text is objective or subjective. It is calculated by using the formula:

Subjectivity Score = (Positive Score + Negative Score)/ ((Total Words after cleaning) + 0.000001)

Range is from 0 to +1

# **Analysis of Readability**

Analysis of Readability is calculated using the Gunning Fox index formula described below.

**Average Sentence Length** = the number of words / the number of sentences

**Percentage of Complex words** = the number of complex words / the number of words

**Fog Index** = 0.4 \* (Average Sentence Length + Percentage of Complex words)

# **Average Number of Words Per Sentence**

The formula for calculating is:

**Average Number of Words Per Sentence =** the total number of words / the total number of sentences

# **Complex Word Count**

Complex words are words in the text that contain more than two syllables.

# **Word Count**

We count the total **cleaned** words present in the text by

1. removing the stop words (using stopwords class of nltk package).
2. removing any punctuations like ? ! , . from the word before counting.

# **Syllable Count Per Word**

We count the number of Syllables in each word of the text by counting the vowels present in each word. We also handle some exceptions like words ending with "es","ed" by not counting them as a syllable.

# **Personal Pronouns**

To calculate Personal Pronouns mentioned in the text, we use regex to find the counts of the words - “I,” “we,” “my,” “ours,” and “us”. Special care is taken so that the country name US is not included in the list.

# **Average Word Length**

Average Word Length is calculated by the formula:

Sum of the total number of characters in each word/Total number of words