**Report**

**Date:** 15th July,2018

**Name**: Supreeth Umesh Kunder

**Assignment Number**: 2 (Role: Data Science)

**Title**:

Write a code to extract the keywords (like Inheritance, encapsulation, multithreading) from the document shared in the link http://bit.ly/epo\_keyword\_extraction\_document, and upload the code in Github and also mention the keywords in order of their weightages in a Google doc or excel sheet.

**Application Used**: Jupyter Notebook

**Modules Imported:**

* PyPDF2(converts PDF file to text readable format)
* Nltk(cleans and convert phrases into keywords)
* Collections(contains data types beyond built-in datatypes)
* Pandas(provides high performance and easy to use data structure and analysis)

**Working:**

There are many PDF(Portable Document Format) converter modules just like PyPDF2 which converts PDF format to text readable format. PyPDF2 fails to convert files containing scanned images, so we run the OCR library textract to convert scanned/image based PDF files into text.

Nltk(also called Natural Language Toolkit) provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

A tokenizer that divides a string into substrings by splitting on the specified string. We have used method tokenize() of class Tokenizer which is available in module nltk.tokenize.

The nltk module contains a list of stopwords, which are used to filter out unwanted texts from sentences. Basically, our motto is to find all keywords. Till date, the list is still incomplete as it fails to search words like The,etc.

So, we first open the pdf and then convert it to text. Through function word\_tokenize() we convert it to words or tokens. To acquire keywords, we must first create a list of punctuations and stopwords, which would not be included in the final list obtained. We will first convert all elements in lowercase and from module collections, we use constructor Counter which creates a dictionary of words and its occurences. Through generator function sorted, we would sort the key-value pairs in ascending order and convert it to a list of tuples in (key,value) format, which becomes much more easier to convert it to excel Spreadsheet format.

I have uploaded the assignment in my Github profile. Here is the link

<https://github.com/FuriousAlkay/Internship-Assignment>

**Start Date**: 14th July, 2018 **Completion Date**: 15th July,2018