**NATURAL LANGUAGE PROCESSING USING PYTHON**

**TABLE OF CONTENTS**

1. ***INTRODUCTION*…………………………………………………….3**
2. ***OBJECTIVE***……………………………………………………………….5
3. ***IMPORTANT TERMS***………………………………………………..6
4. Natural Language Processing……………………………….6
5. Data mining…………………………………………………………8
6. Machine Learning……………………………………………..10
7. ***USER STORY***…………………………………………………………..11
8. ***IMPLEMENTATION FOR EMAIL CLASSIFICATION***……12
9. Procedure and steps involved…………………………….12
10. ***SENTIMENT ANALYSIS***……………………………………………48
11. Simple text classification with textblob………..........49
12. Stopwords with NLTK……………………………………….…52
13. Training data for sentiment analysis using NLTK....54
14. ***SOFTWARE DEVELOPNMENT ENVIRONMENT***………..57
15. Python 3.5.0……………………………………………………….60
16. NLTK……………………………………………………………………63
17. Anaconda……………………………………………………………64
18. Spyder…………………………………………………………………65

**INTRODUCTION**:

Email has become one of the frequently used forms of communication. Everyone has at least one email account. Inflow of many messages is a major problem faced by the companies.

Currently there are many email classification techniques. The important algorithm namely, *Naïve Bayes Theorem* is the efficient way in classifying mails.

As, email is one of the most efficient and effective mode of communicating with one another and to classify the mails, need of effective filtering is a timely requirement.

Filtering is nothing but the method of arranging mails. Classification is the technique of data mining. Data mining is defined as discovering useful knowledge from large data.

Classification is a type of data analysis that extracts models describing important data classes or concepts.

Classification mainly consists of two steps.

***First is the learning step***: where a classification model is constructed and ***second step is the*** ***classification step:*** in this step the extracted model is used to predict the class labels for new data or unknown data depending on the learning step.

Machine learning algorithms are used for classification of objects of different classes. Such algorithms have proved to be efficient in classifying emails as sentiment, service or a complaint email.

**OBJECTIVE:**

*The objective of the project is to classify the emails based on feedback, service requests and complaints using python and NLP.*

**EMAILS**

**COMPLAINTS**

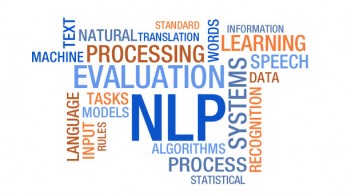
**SERVICE REQUESTS**

**FEEDBACK**

*Fig1. Classification of emails*

**IMPORTANT TERMS:**

**1. *NATURAL LANGUAGE PROCESSING***: Natural Language Processing is a field of computer science, artificial intelligence (also called Machine Learning), and linguistics concerned with the interaction between computers and human languages.



**WHY NATURAL LANGUAGE PROCESSING IS IMPORTANT?**

Applications for processing large amount of texts require NLP expertise.

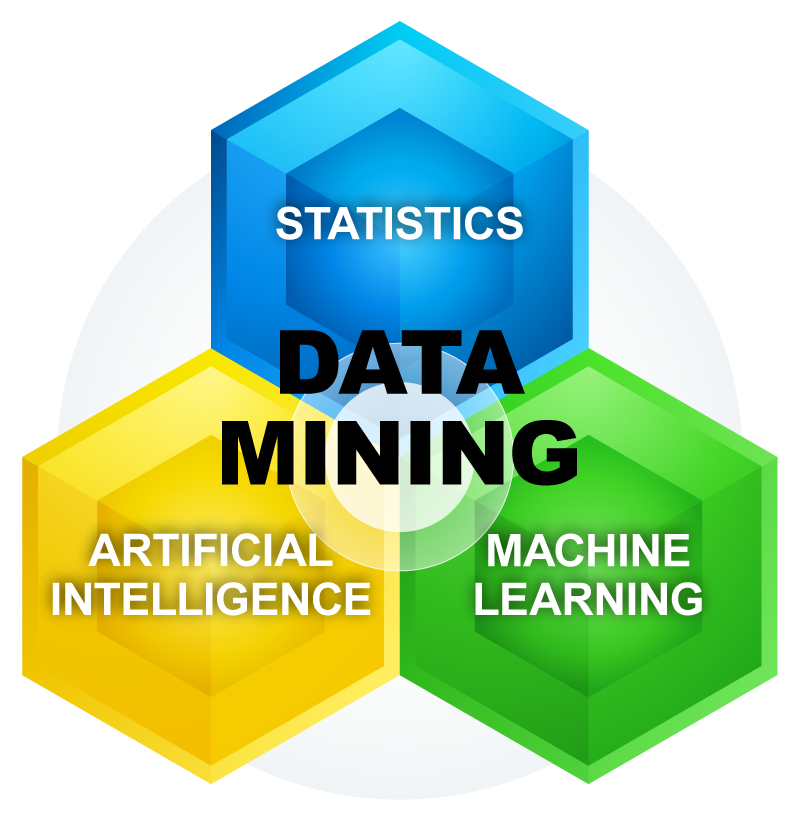
* ***Classify text into categories, index and search large texts***:

Classify documents by topics, language, author, spam filtering, information retrieval (relevant, not relevant), and sentiment classification (positive, negative).

* ***Extracting data from text***: converting unstructured text into data
* ***Information extraction***: discover names of people and events they participate in, from a document.
* ***Automatic summarization***: Condense 1 book to 1 page
* ***Speech processing, artificial voice***: E.g. Book hotel over a phone.
* ***Question answering***: find answers to natural language questions in a text collection or database.
* ***Spelling and Grammar Corrections***
* ***Plagiarism detection***
* ***Automatic translation***

***2. DATA MINING***: Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis.

*Data mining tools allow enterprises to predict future trends.*

****

**IMPORTANCE OF DATA MINING:**

* Uses sophisticated data manipulation technology.
* Deals with large databases.
* Identifies useful information.

***3. MACHINE LEARNING*:** Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed.

Machine learning focuses on the development of computer programs that can change when exposed to new data.

Machine learning systems are made up of three major parts, which are:

* ***Model:*** *the system that makes predictions or identifications.*
* ***Parameters:****the signals or factors used by the model to form its decisions.*
* ***Learner:*** *the system that adjusts the parameters — and in turns the model — by looking at differences in predictions versus actual outcome.*

**USER STORY**

*As a user of the system, I want to classify emails based on feedback, service requests and complaints.*

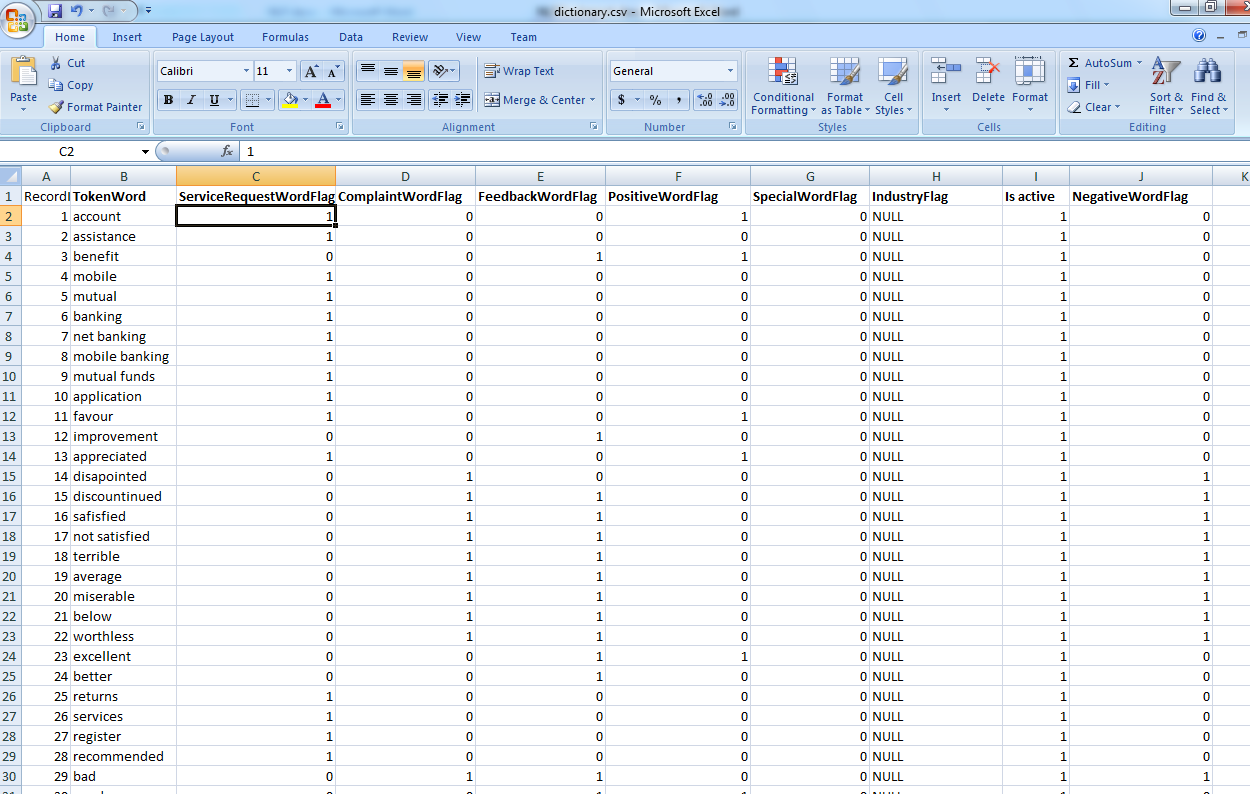
* *

***USER*** ***EMAILS***

** **

**IMPLEMENTATION FOR EMAIL CLASSIFICATION:**

1. **Create a dictionary**

****

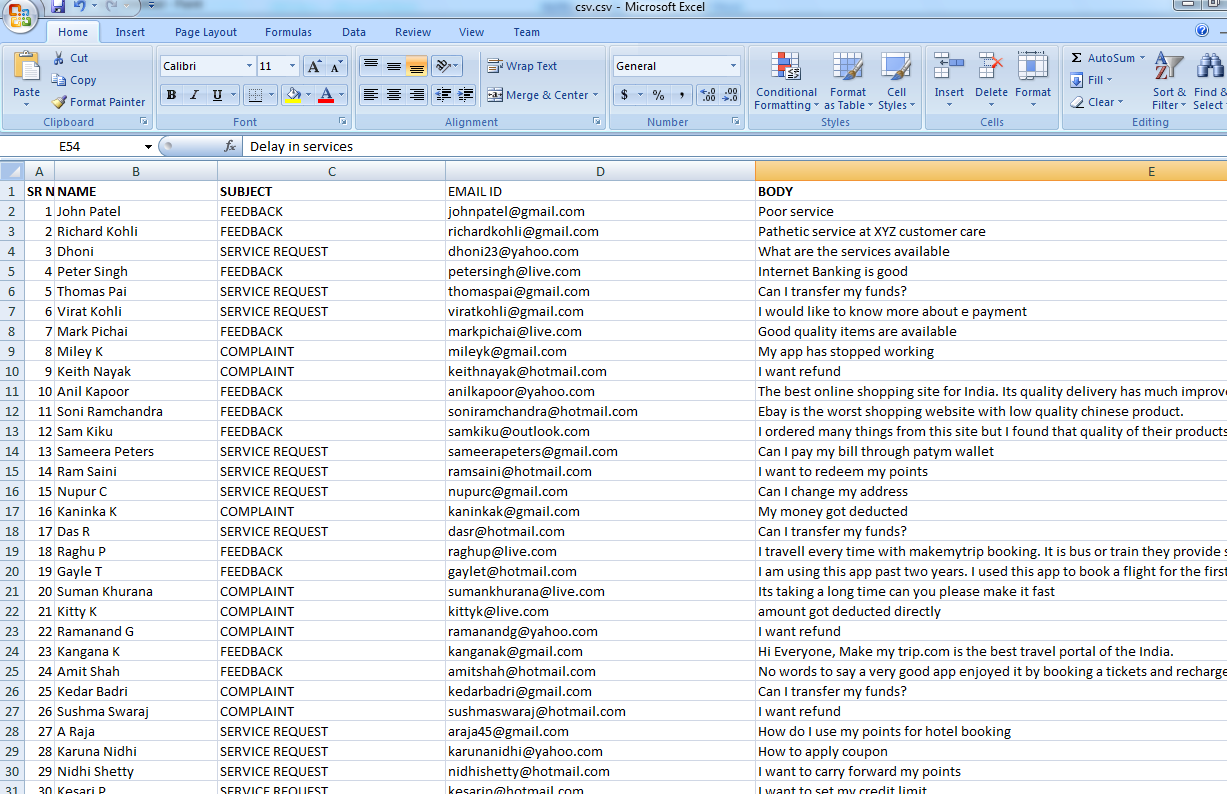
*Figure2. Creating a dictionary*

*The given csv extension file contains record ID, Token words i.e. The list of words, service requests, complaint words, feedback words, positive and negative words.*

*E.g. Terrible*

*It can be taken as a complaint, therefore its ‘1’.*

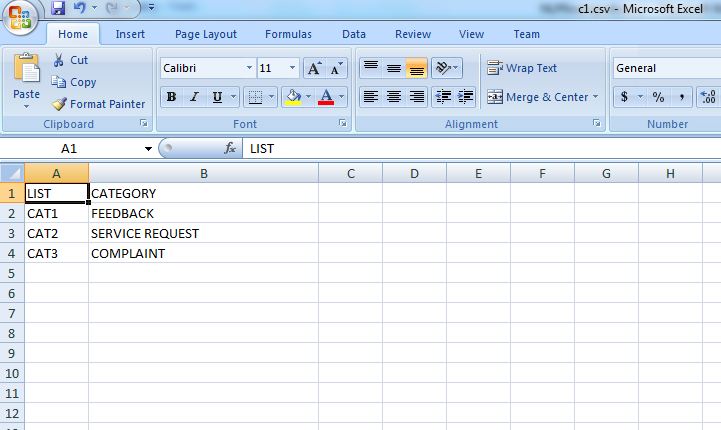
1. ***Creating a CSV file***

****

*Figure3. List of emails*

*Emails.csv contains all the emails with name, subject (feedback, service request, and complaint), Email Ids and body.*

1. ***Creating various categories***

****

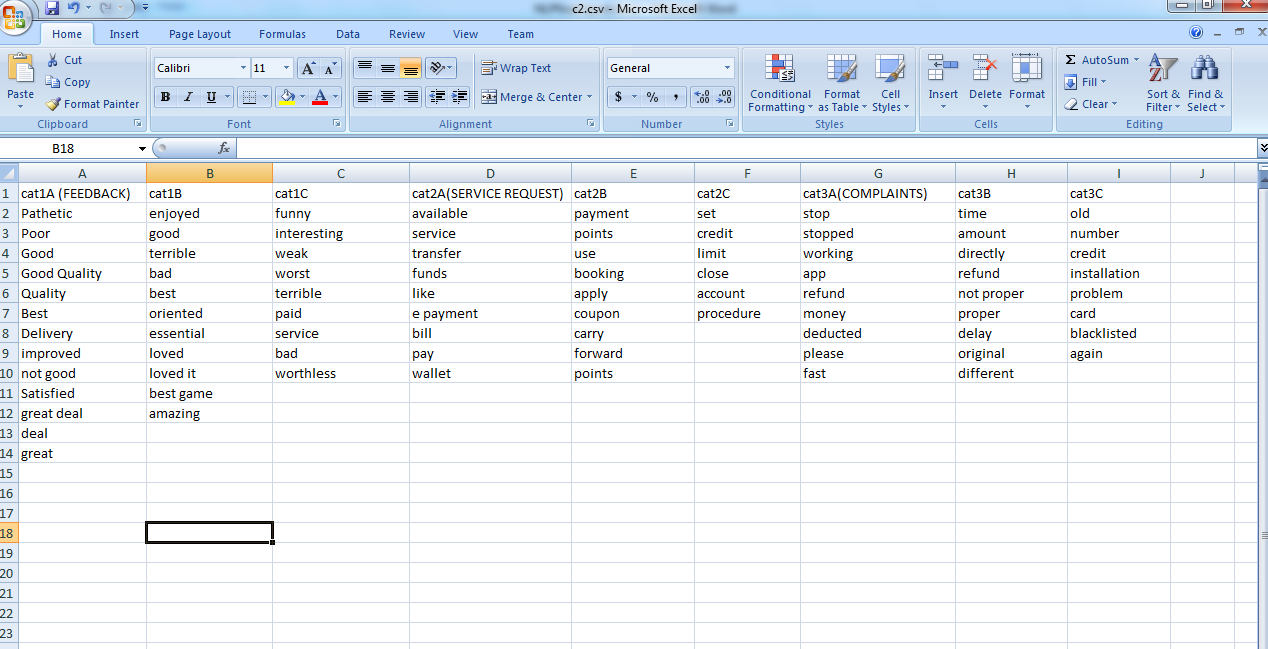
*Figure4. Creating various categories*

*For email classification, we have created following categories:*

*CAT1- Feedback*

*CAT2-Service Requests*

*CAT3-Complaint*

****

*Figure5. Categories*

*Another file contains classification of words*

* *CAT1A, CAT1B, CAT1C- Feedback*

*E.g. Great*

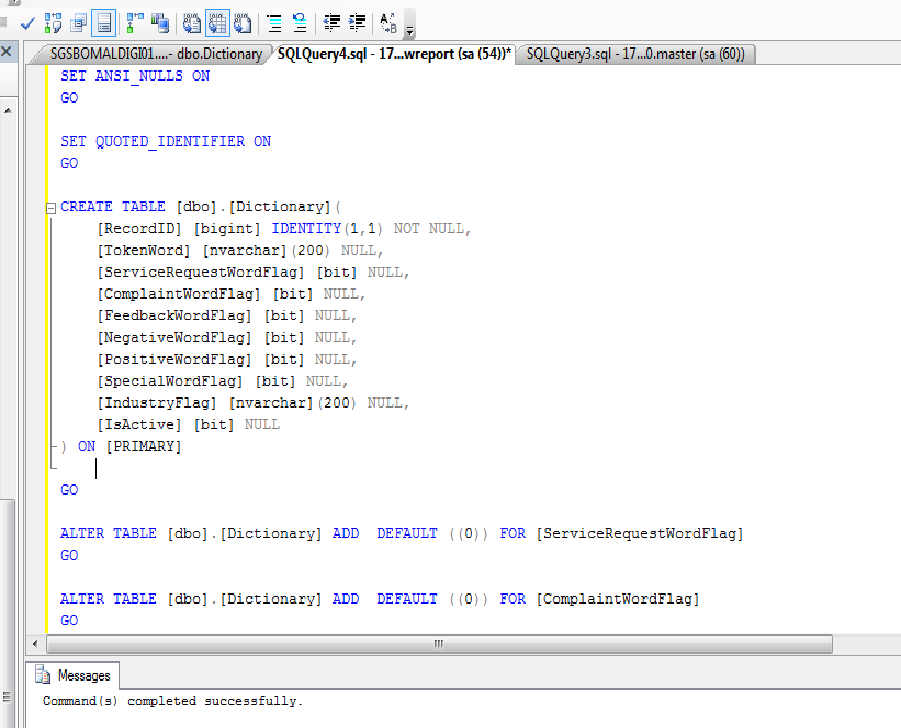
* *CAT2A, CAT2B, CAT2C- Service Requests*

*E.g. Transfer*

* *CAT3A, CAT3B, CAT3C- Complaints*

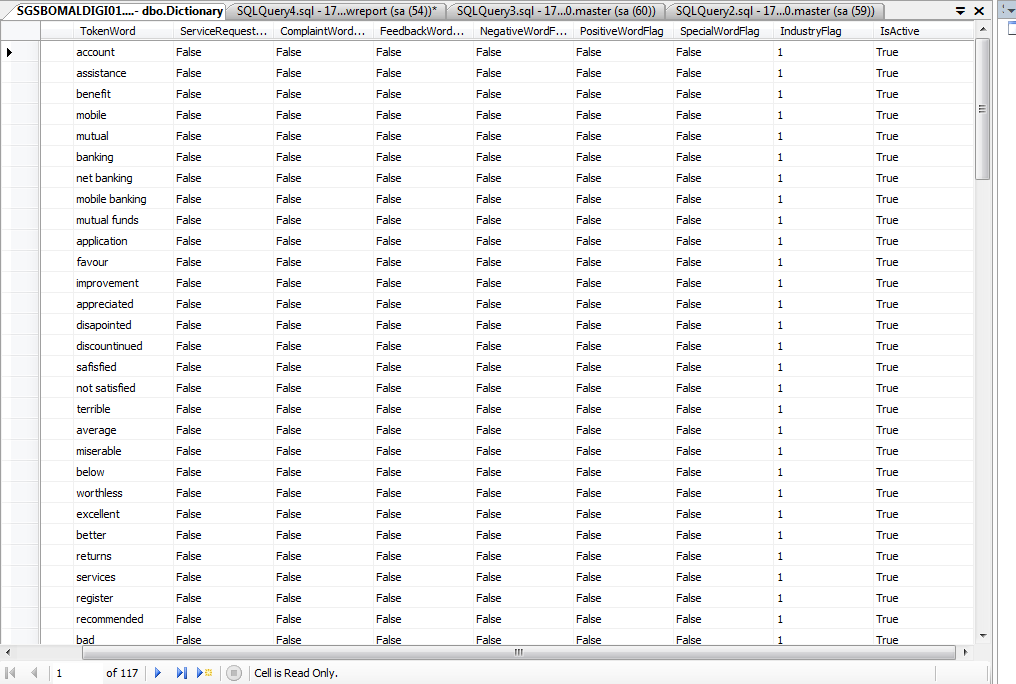
*E.g. Refund*

1. ***Using SQL***

****

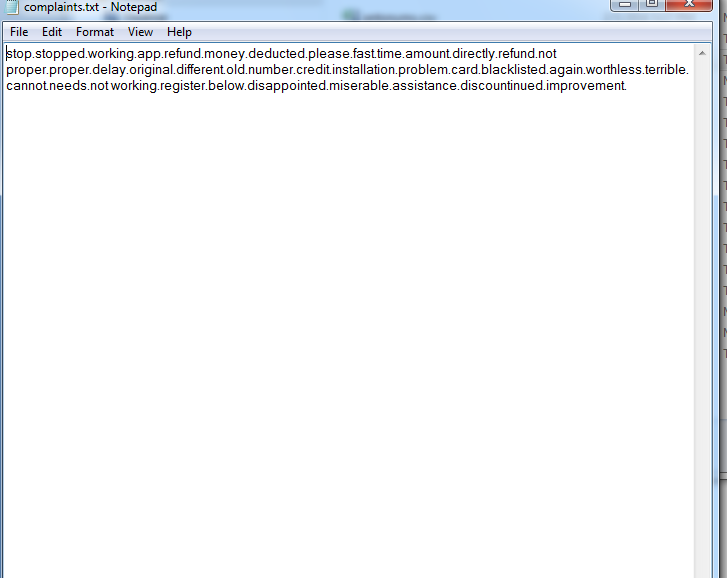
*Figure6. Using Sql server*

* *Creating a database*
* *Create table dbo [filename]*
* *Use Record ID, Token Word, Service Request, Complaint, Feedback, Negative and Positive flags from the dictionary file*
* *Alter each table*

**

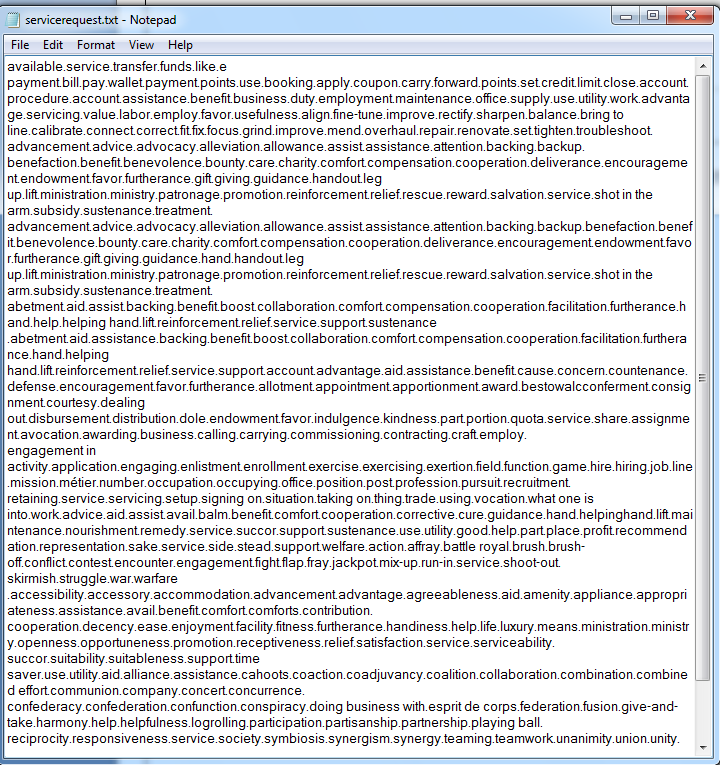
*Figure7. Database of dictionary*

1. ***Creating text files for various categories***

******

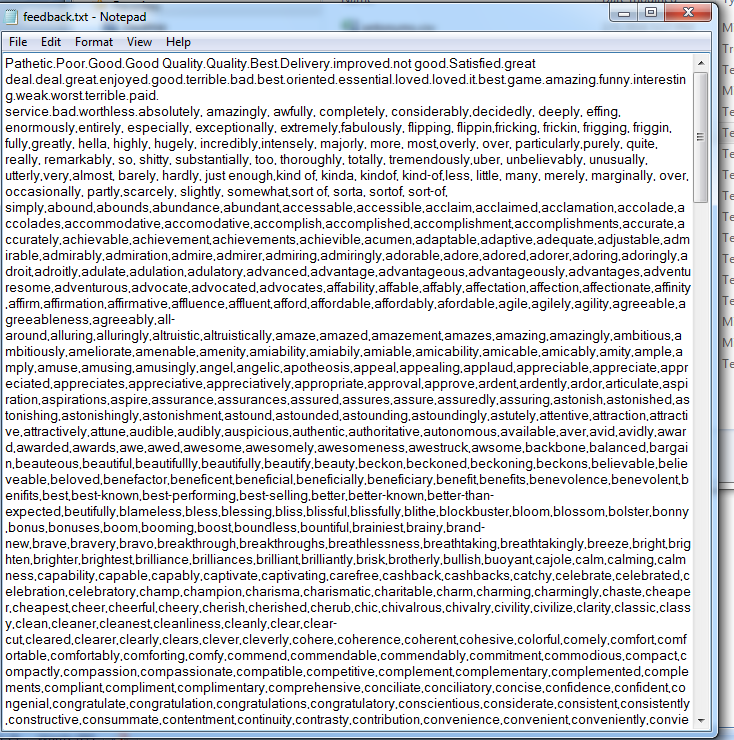
*Figure8. Creating text file*

*The file complaints.txt contains the list of words which come in CAT3 (complaints).*

****

*Figure9- Text file*

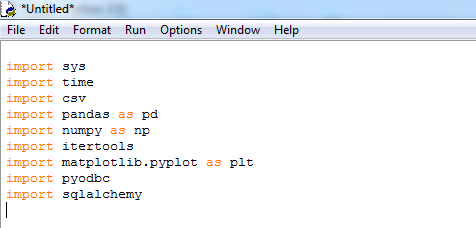
*The file servicerequest.txt contains the list of words which come in CAT 2 (service requests).*

****

*Figure10- Text file*

*The file feedback.txt contains the list of words which come in CAT 1(Feedback).*

1. ***Python code:***

****

1. ***import sys****- system specific parameters and functions.*

*This module provides access to some variables used or maintained by the*

*Interpreter and to functions that interact strongly with the interpreter.*

1. ***import time****- time access and conversions.*

*This module provides various time-related functions.*

1. ***import csv*** *– it imports a csv file.*
2. ***import pandas***- *It is a* [*Python*](http://www.python.org/) *package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.*

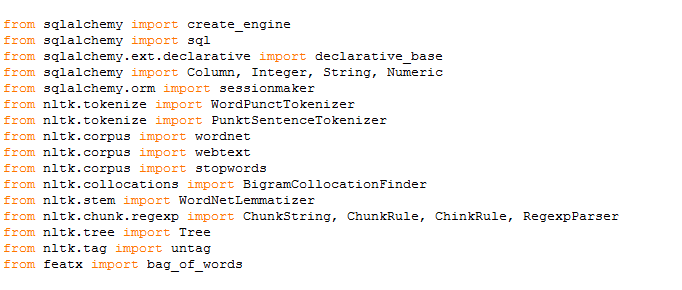
*It aims to be the fundamental high-level building block for doing practical,* ***real world*** *data analysis in Python.*

1. ***import numpy****- NumPy is the fundamental package for scientific computing with Python. It contains a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities*
2. ***import itertools****- it is a function for creating iterations for efficient looping. This module implements a number of* [*iterator*](https://docs.python.org/2/glossary.html#term-iterator) *building blocks*
3. ***import matplotlib.pyplot-*** *Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.*

*Matplotlib can be used in Python scripts, the Python and* [*IPython*](http://ipython.org/) *shell, the* [*jupyter*](http://jupyter.org/index.html) *notebook, web application servers, and four graphical user interface toolkits.*

1. ***import sqlalchemy****- SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.*

*It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.*

****

1. ***import create engine****- Construct a base class for declarative class definitions.*
2. ***sessionmaker****- establishes and maintains all conversations between the program and the databases. It represents an intermediary zone for all the Python model objects loaded in it.*

1. [***WordNet***](https://wordnet.princeton.edu/)*- It is a lexical database for the English language, which was created by Princeton, and is part of the NLTK corpus.*

*WordNet along with the NLTK module can be used to find the meanings of words, synonyms, antonyms, and more.*

1. ***The BigramCollocationFinder and TrigramCollocationFinder*** *- classes provide these functionalities, dependent on being provided a function which scores a ngram given appropriate frequency counts.*

*A number of standard association measures are provided in bigram\_measures and trigram\_measures.*

1. ***WordNet Lemmatizer****- It is a Part-of-Speech tagging project by first modifying each word in the training corpus to its stem (in place modification), and then training only on the new corpus.*

*For example, the word hopes is lemmatized to hope which is correct, but the word hoping remains hoping even after lemmatization.*

1. ***untag****- Given a tagged sentence, return an untagged version of that sentence. I.e., return a list containing the first elementof each tuple in \*tagged\_sentence\*.*

*E.g.*

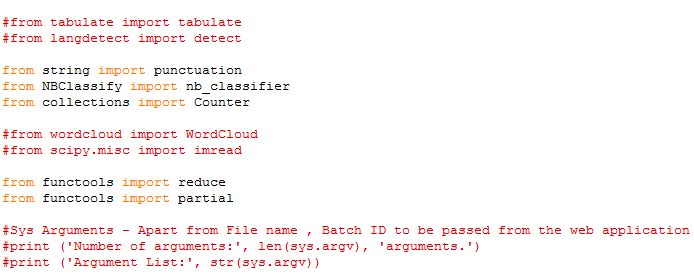
*>>> from nltk.tag.util import untag*

*>>> untag([('John', 'NNP'), ('saw', 'VBD'), ('Mary', 'NNP')]) ['John', 'saw', 'Mary'].*

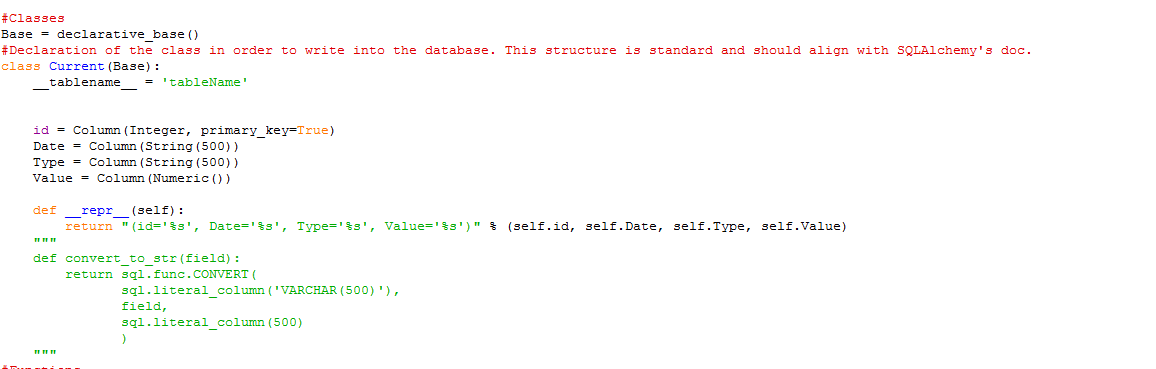
1. ***Tree*** *- trees are non-parametric supervised learning method used for* [*classification*](http://scikit-learn.org/stable/modules/tree.html#tree-classification) *and* [*regression*](http://scikit-learn.org/stable/modules/tree.html#tree-regression)*. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.*
2. ***Bag of Words****- bag of words model learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears. For example, consider the following two sentences:*

*Sentence 1: "The cat sat on the hat"*

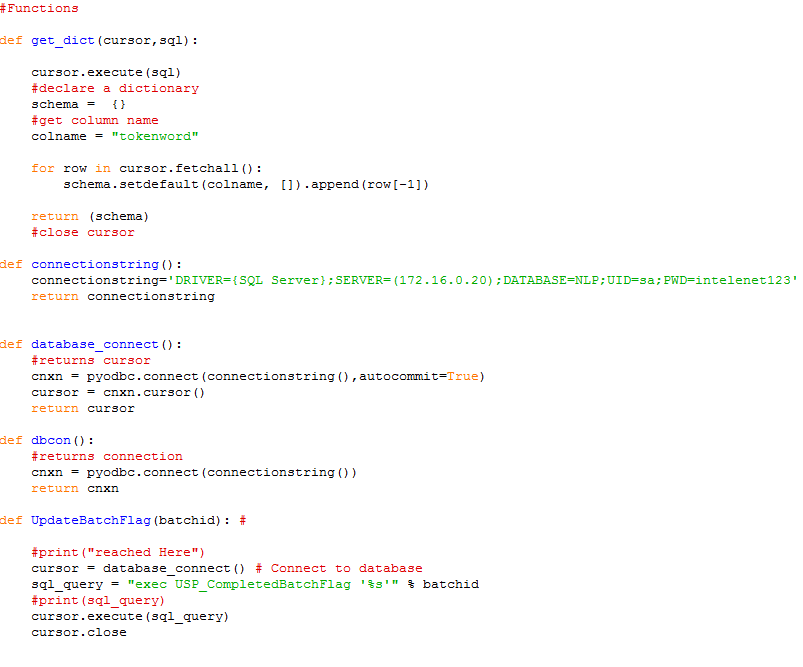
*Sentence 2: "The dog ate the cat and the hat"*

****

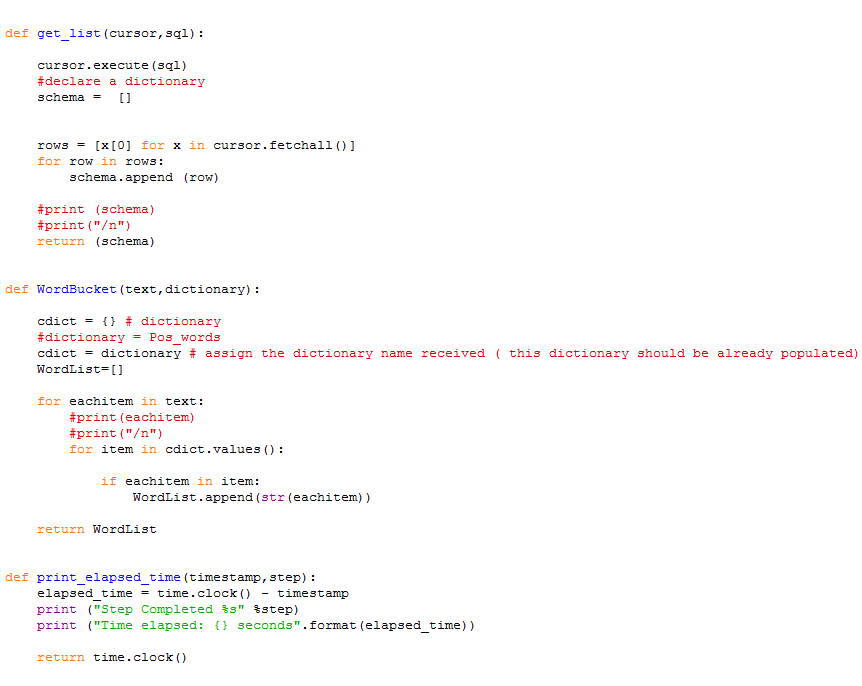
* *Import punctuation, NB classifier and counter from string, NB classify and collections respectively.*
* *From functools import reduce and partial.*

****

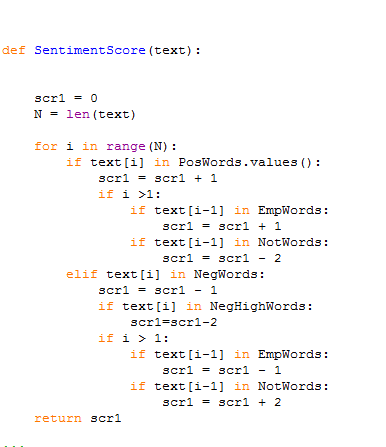
* *We declare a class in order to write into the database.*
* *We declare id, date, type and value.*

****

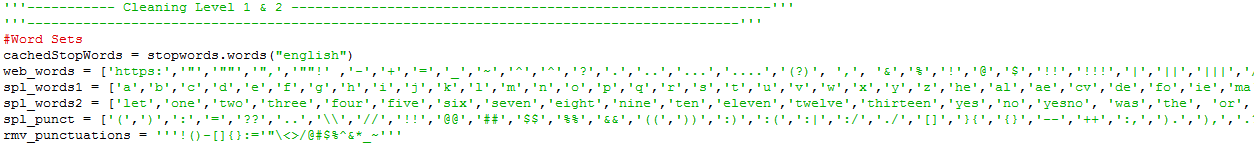
* *Function- using sql we define get\_dict from the sql server and we connect it to the database.*

****

* *From the word bucket we take pos and neg words*
* *We print elapsed time.*

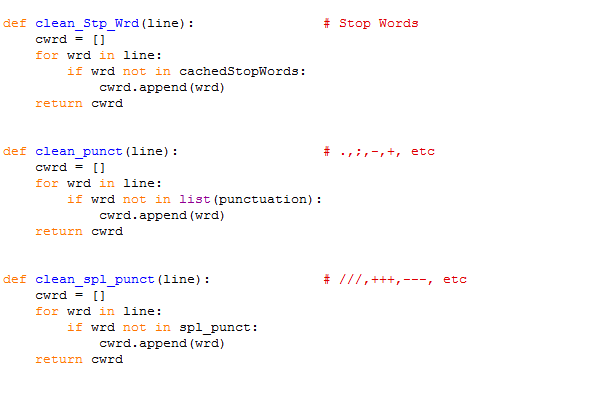
****

* *We define sentiment score of all the words.*

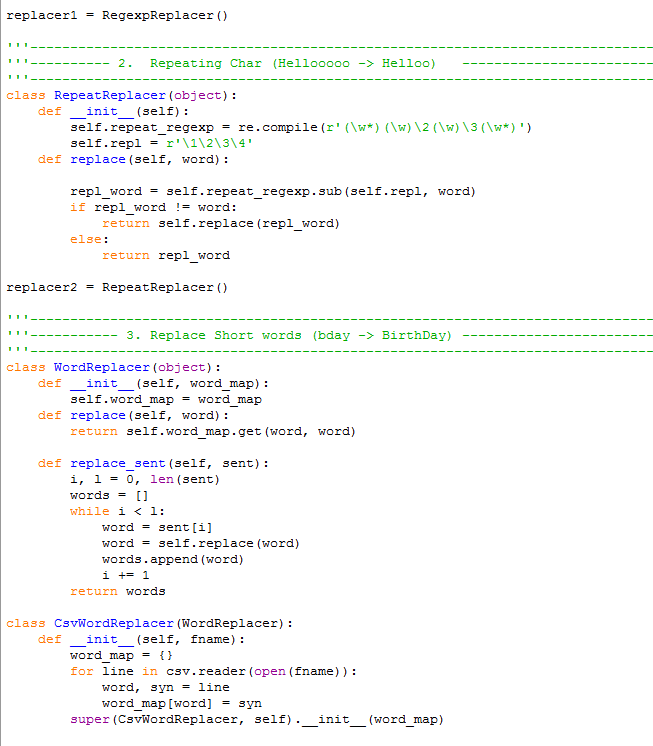
****

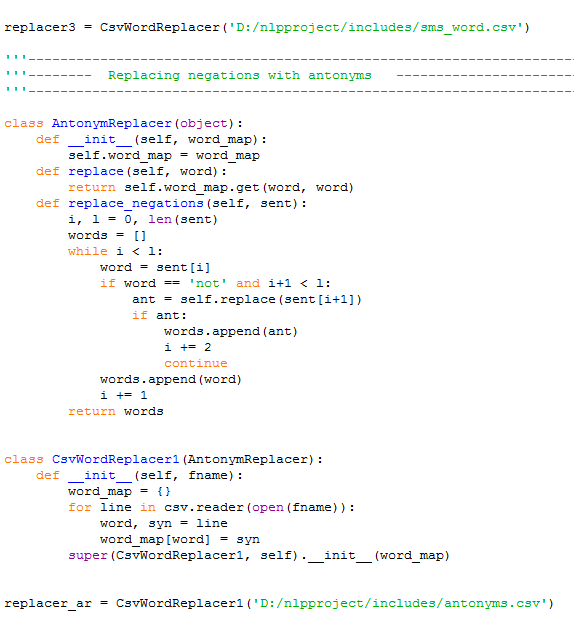
* *Word sets- we use stopwords.words(‘English’)*

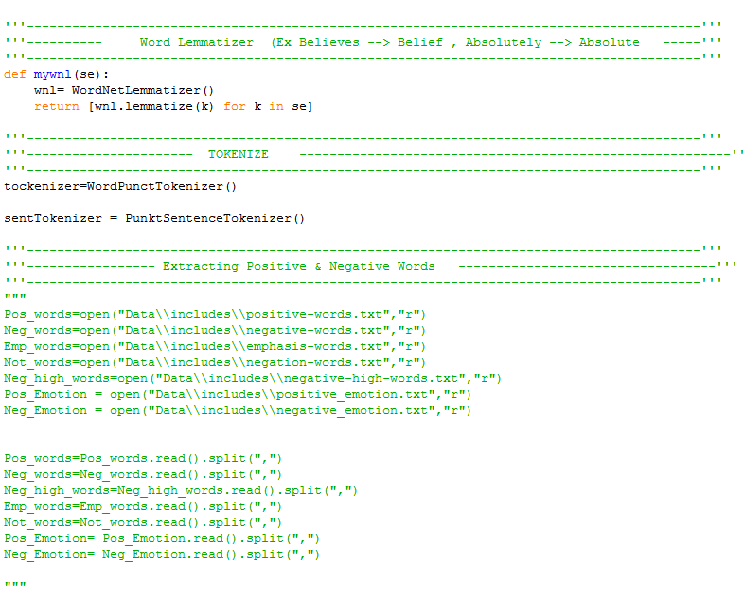
****

****

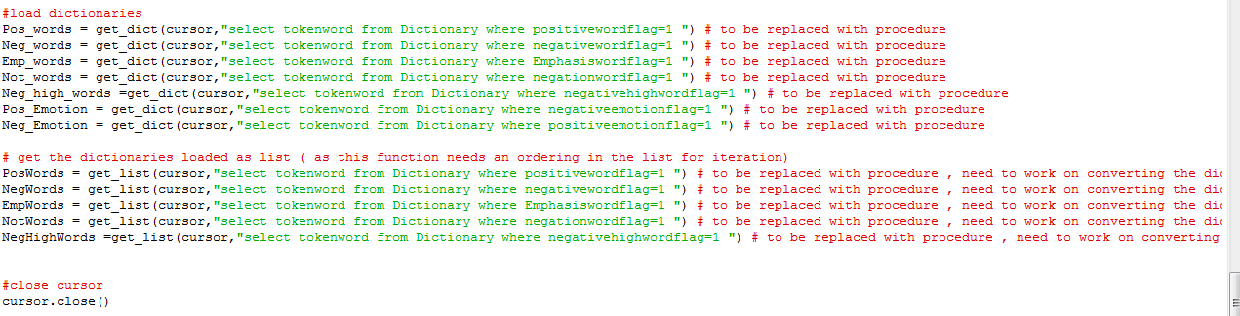
****

****

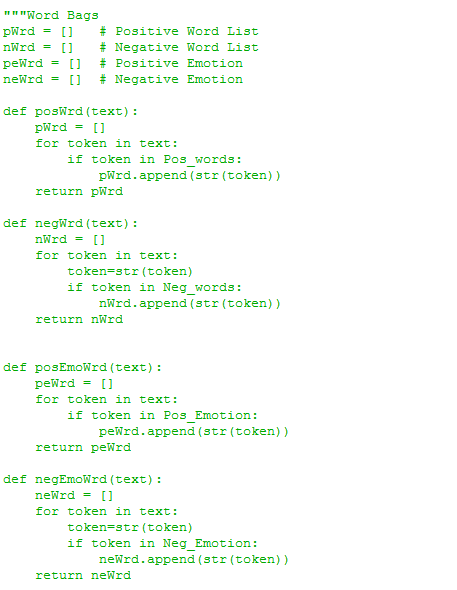
****

****

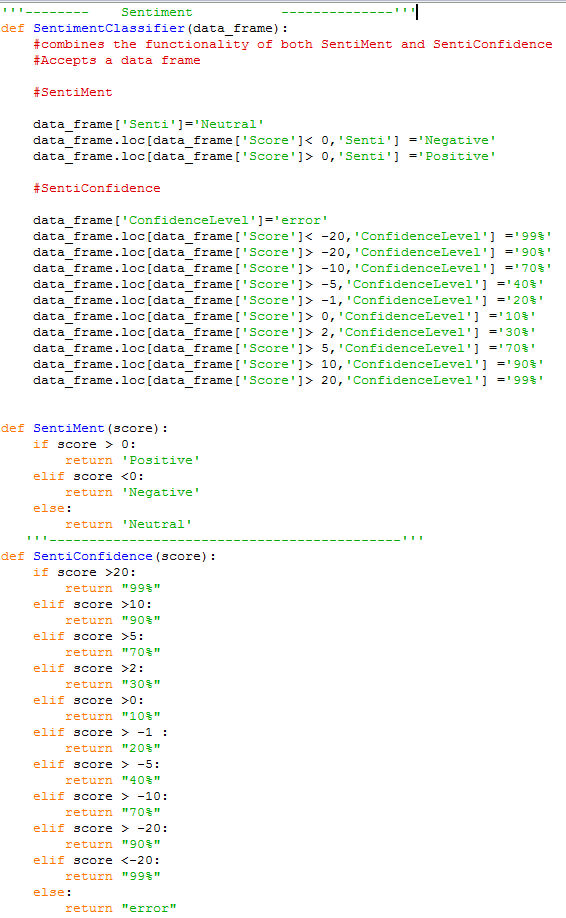
1. *Using wordlemmitizer*
2. *Tokenizing the sentences*
3. *Then extracting the words*

****

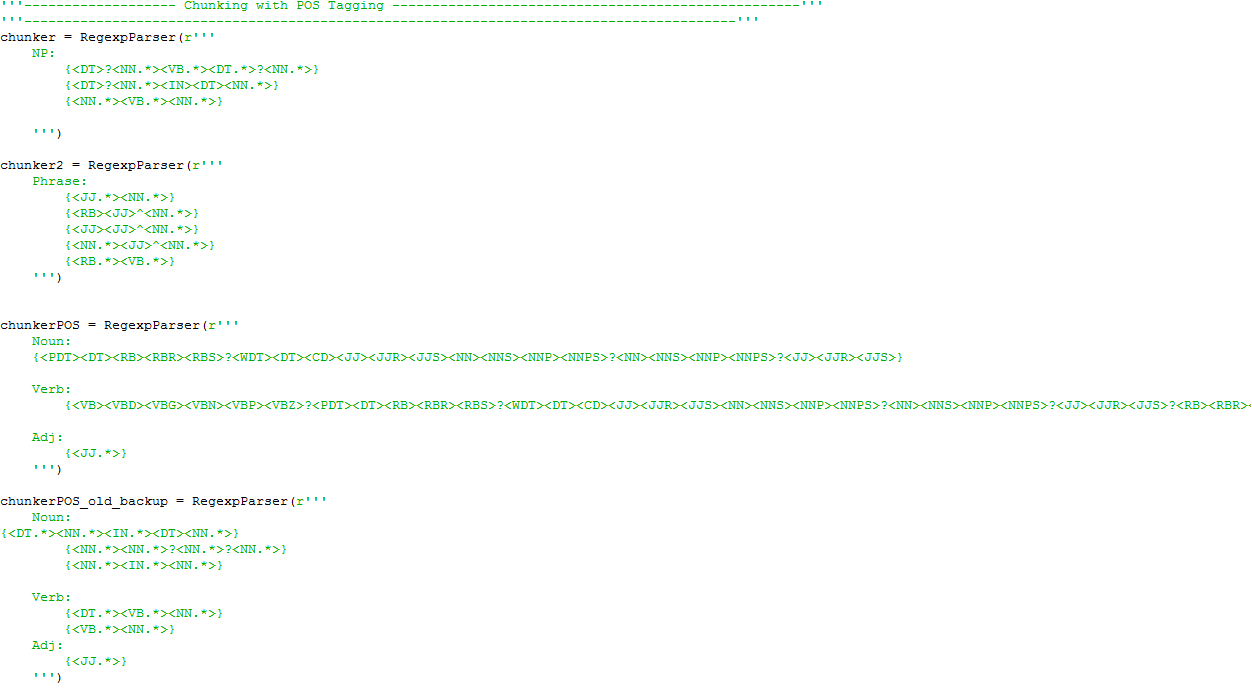
*Loading the dictionaries*

****

*From the word bag, extracting all the words according to the categories*

****

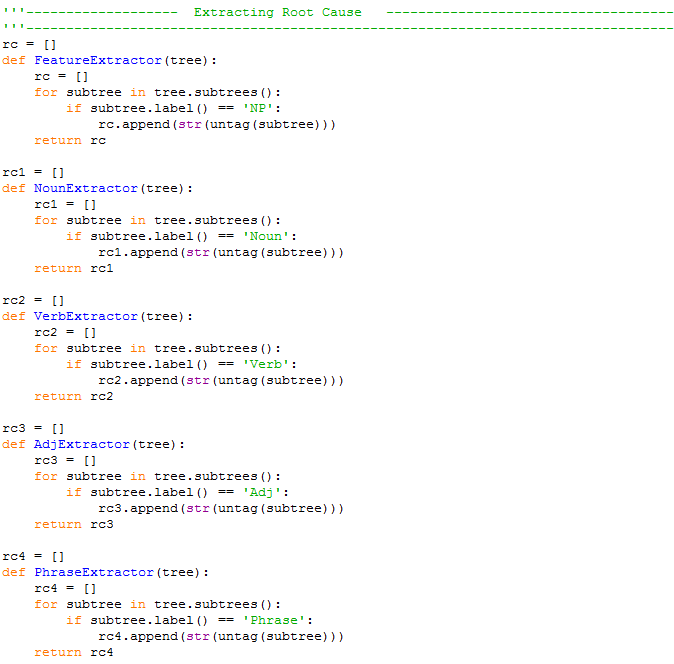
*Classifying the sentiments using sentiment classifier*

****

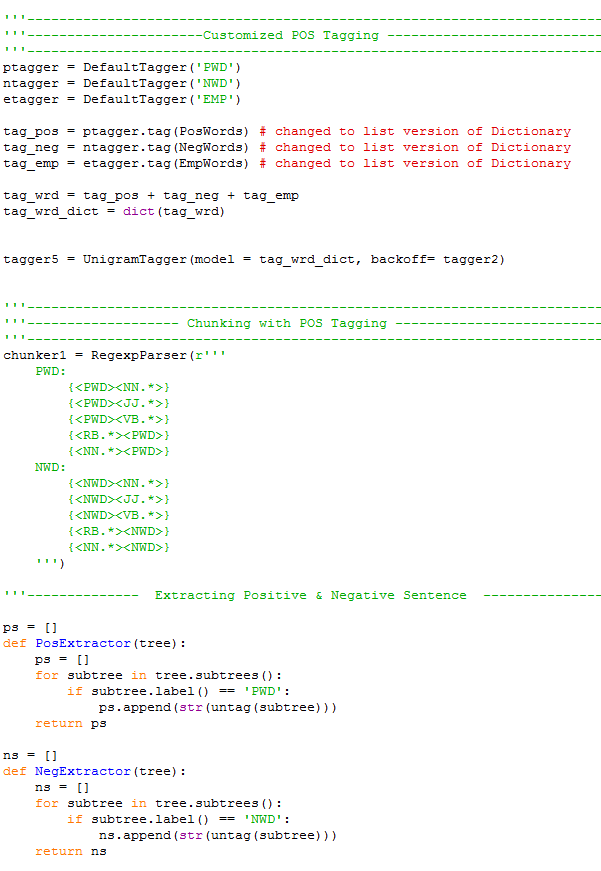
*Chunking with the POS tagger*

*POS tagging is a process deciding what is the type of every token from a text, e.g. NOUN, VERB, DETERMINER, etc.*

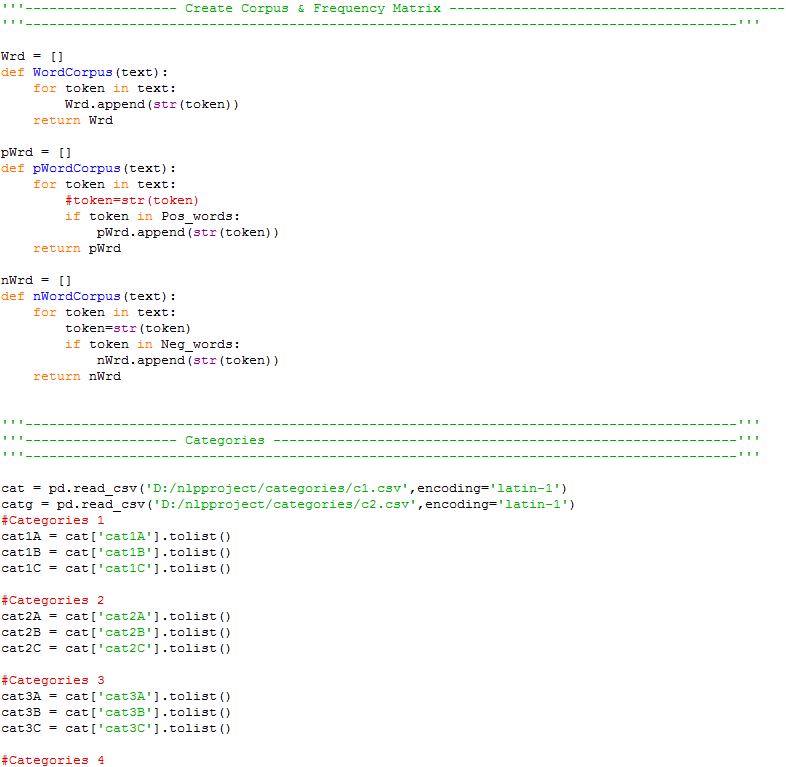
*Token can be word or punctuation*

****

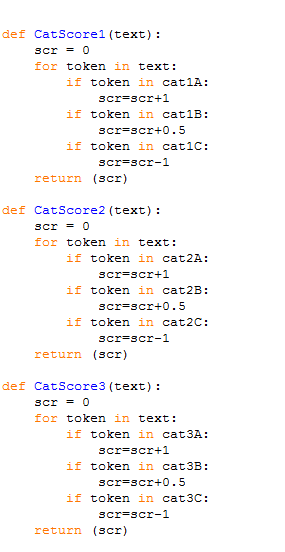
* *Extracting the rootcause, rc1, rc2, rc3 and rc4*

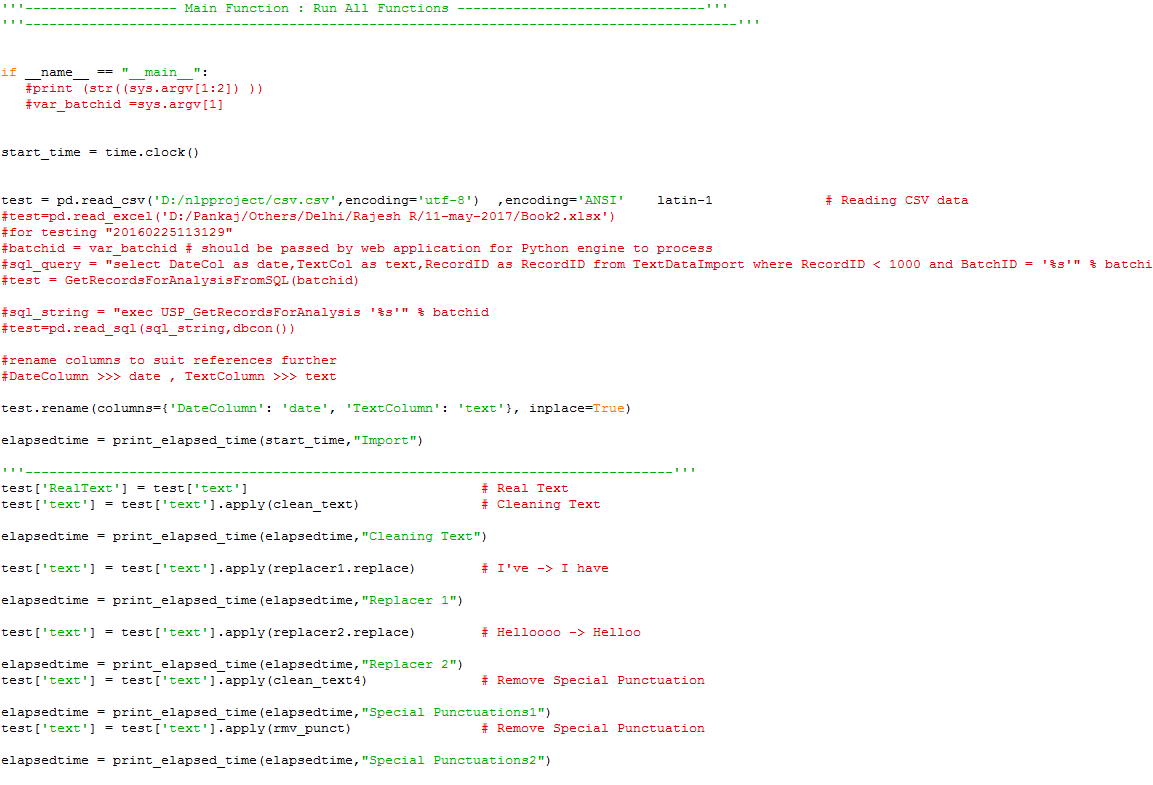
****

* *Customizing POS tagger, pos an nos.*
* *Chunking with the POS tagger*

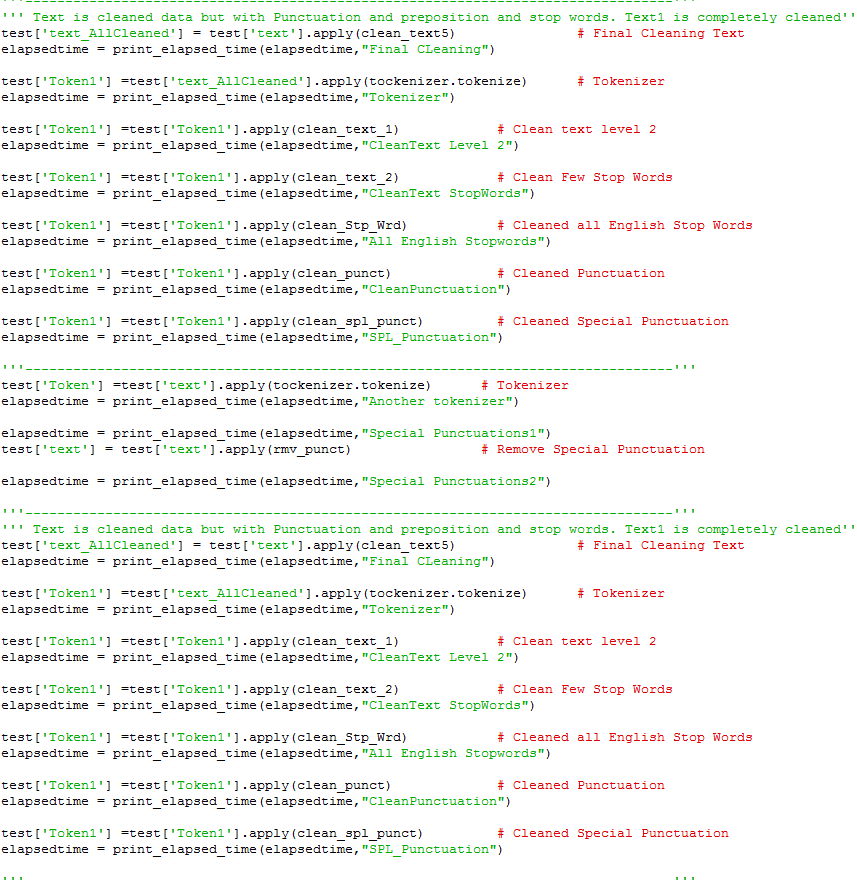
****

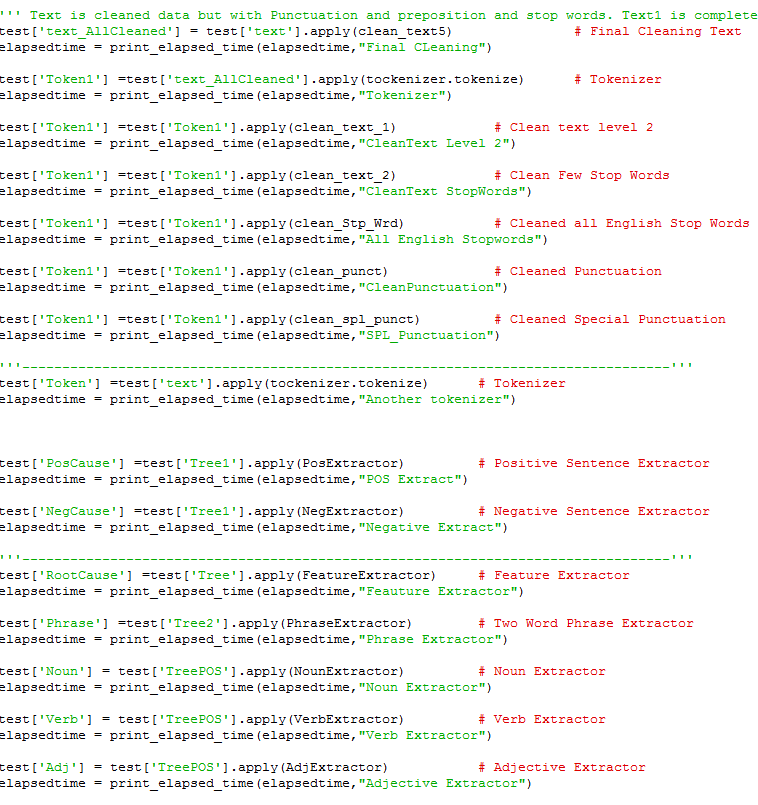
* *We create corpus and frequency matrix.*
* *Reading categories via CSV file.*

****

****

* *Reading the csv file and then testing it.*

****

****

* *Text is cleaned with punctuation, preposition and stop words*

**STEPS INVOLVED:**

*Creating a folder “NLP project”*

*Creating a dictionary*

*Creating emails csv file*

*Creating text files for various categories*

*Using SQL for dictionary database*

*Using Python code*

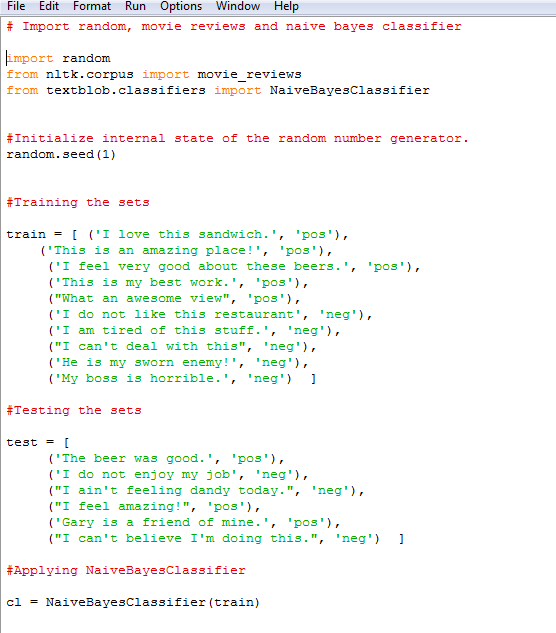
*Emails classified*

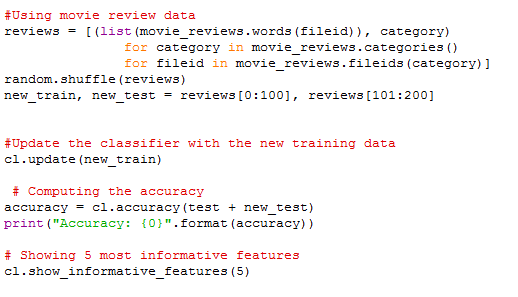
**SENTIMENT ANALYSIS**

****

* *Sentiment analysis is the classification of the polarity of a given text in the document, sentence or phrase.*
* *The goal is to determine whether the expressed opinion in the text is positive, negative or neutral.*

***SIMPLE TEXT CLASSIFICATION WITH PYTHON AND TEXTBLOB***

******

****

***EXPLAINATION***

1. *The first classifier will be a simple sentiment analyzer trained on a small dataset*
2. *We will import the* ***textblob.classifiers*** *and create training and testing data.*
3. *We create a new classifier by passing training data into the constructor for a* ***NaiveBayesClassifier****.*
4. *We then classify arbitrary text using the* ***NaïveBayesClassifier.classify (text)*** *method.*
5. *We then check the accuracy of the test set.*

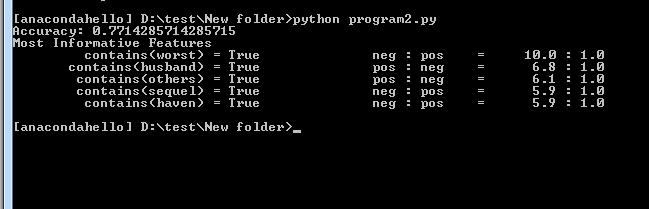
***e.g. cl.accuracy (test)***

***#0.78***

1. *We can then find informative features*

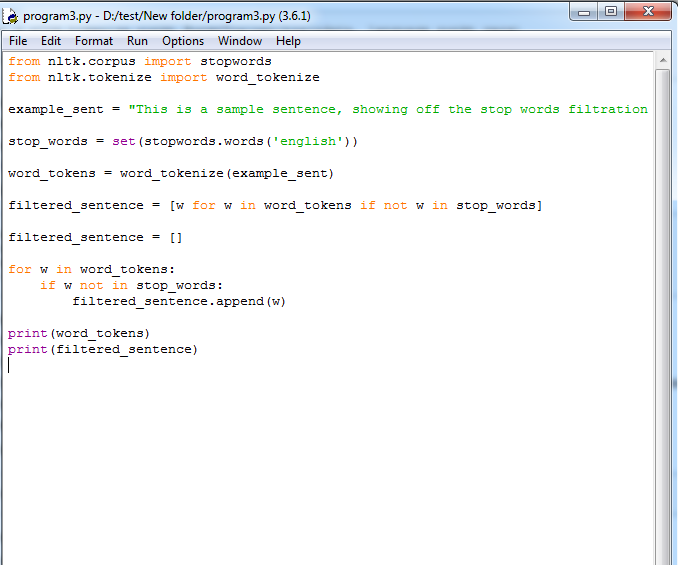
***cl.show\_informative\_features (5)***

***OUTPUT***

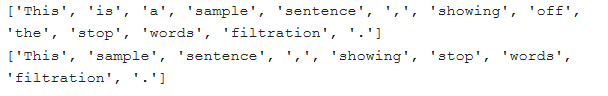
******

*Figure12. Output*

***STOP WORDS WITH NLTK:***

****

*Output*

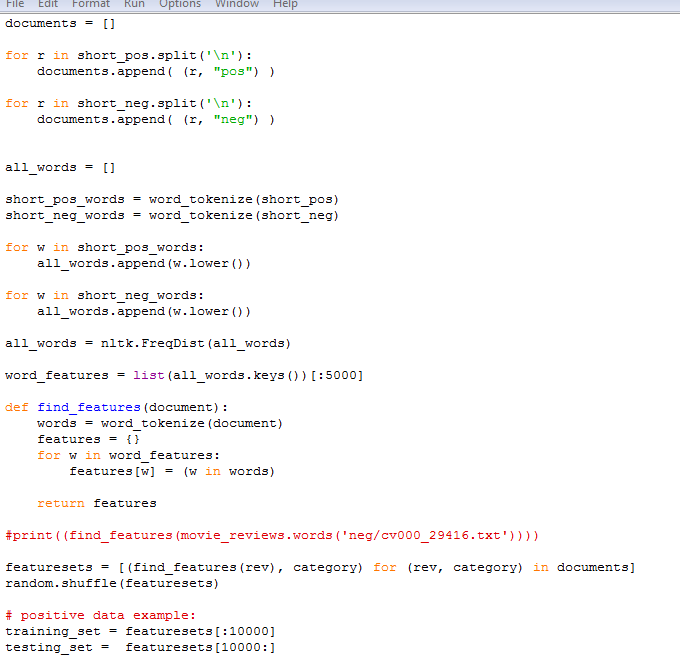
****

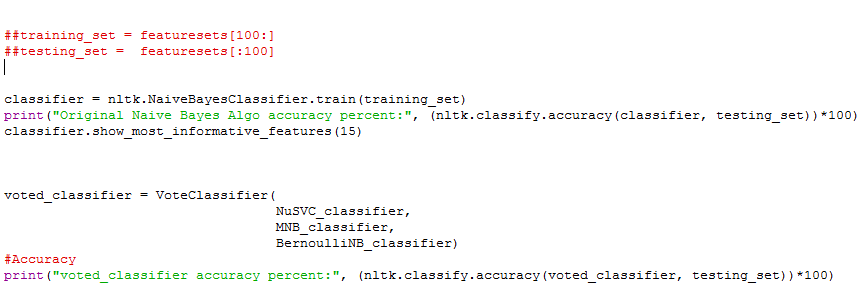
***EXPLAINATION***

1. *Stop\_words are used to remove words which have no meaning.*
2. *For example, we may wish to completely cease analysis if we detect words that are commonly used sarcastically, and stop immediately.*
3. *Sarcastic words or phrases are going to vary by lexicon and corpus.*
4. *For now, we'll be considering stop words as words that just contain no meaning, and we want to remove them.*

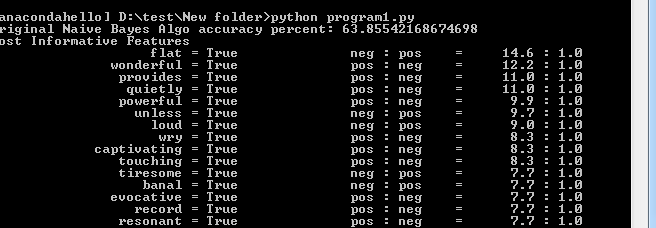
**TRAINING DATA FOR SENTIMENT ANALYSIS USING NLTK**

****

****

****

***Output***

****

*Figure4. Output*

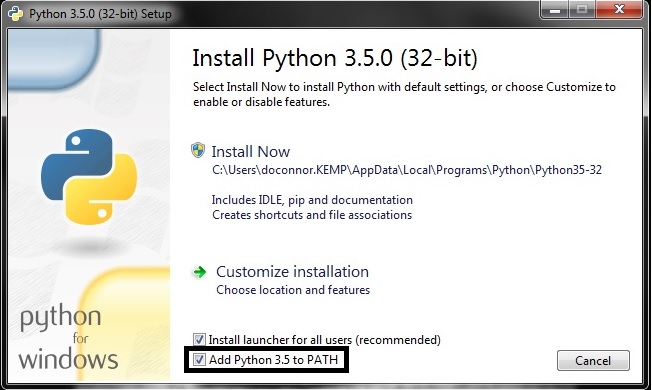
***EXPLAINATION***

1. *We train our new data set as our goal is to do email sentiment.*
2. *We keep shorter negative and positive data set, to get more accuracy*

**SOFTWARE DEVELOPMENT ENVIRONMENT**

1. ***Python 3.5.0***

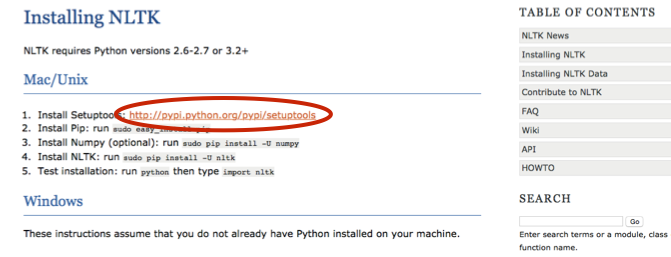
*Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.*

**

1. ***NLTK:***

*We install NLTK tools using pip*

*Run sudo pip install -U nltk*

******

**NATURAL LANGUAGE TOOL KIT:**

*Natural Language Tool Kit is a leading platform for building Python programs to work with human language data.*

*It 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, and tagging, parsing, and semantic reasoning, wrappers for industrial strength NLP libraries.*

***IMPORTANCE OF NLTK***

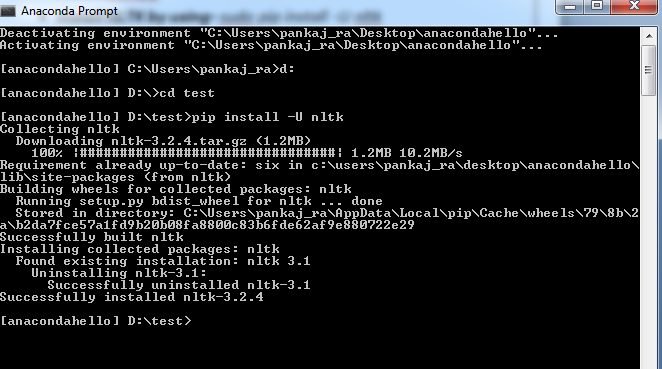
* *The NLTK module is a massive tool kit, aimed at helping with the entire Natural Language Processing (NLP) methodology.*
* *NLTK will aid with everything from splitting sentences from paragraphs, splitting up words, recognizing the part of speech of those words, highlighting the main subjects*
* *It also helps the machine to understand what the text is all about.*
* *NLTK can be installed using pip or from nltk website.*
* *The easiest way to download NLTK is by using pip.*

***INSTALLING NLTK***

NLTK requires Python versions 2.7, 3.4, or 3.5.

***FOR WINDOWS (32bit preferred)***

* ***Install Python 3.5***
* ***Install NLTK by using-*** *sudo pip install -U nltk*



*Fig2. Successfully installed NLTK*

1. ***INSTALLATION PACKAGES*:**

* ***Installing Scikit Learn****: It is a simple and efficient tool for data mining and data analysis.*
* ***Installing Pandas:*** ***pandas*** *is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical,* ***real world*** *data analysis in Python.*
* ***Installing Numpy:*** *NumPy is a general-purpose array-processing package designed to efficiently manipulate large multi-dimensional arrays of arbitrary records without sacrificing too much speed for small multi-dimensional arrays. NumPy is built on the Numeric code base and adds features introduced by numarray as well as an extended C-API and the ability to create arrays of arbitrary type which also makes NumPy suitable for interfacing with general-purpose data-base applications.*

*There are also basic facilities for discrete Fourier transform, basic linear algebra and random number generation.*

* ***Installing Textblob:***

TextBlob *is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.*

* ***Installing stopwords:*** *contains list of common stop words in various languages in Python.*
* ***Installing porterstemmer:*** *Stemming algorithms attempt to automatically remove suffixes (and in some cases prefixes) in order to find the “root word” or stem of a given word. This is useful in various natural language processing scenarios, such as search.*
* ***Installing punksentencetokenizer:*** *It is an abstract class for the default sentence tokenizer.*
* ***Installing NaiveBayesClassifier-****A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes’ theorem with strong (naive) independence assumptions.*

1. ***ANACONDA***

* *Anaconda is the leading open data science platform powered by Python.*
* *The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular Python, R and Scala packages for data science.*
* *Additionally, it has access to over 720 packages that can easily be installed with conda, it is a renowned package, dependency and environment manager, that is included in Anaconda.*
* *We use Anaconda 4.4.0 for 32 bit Windows*

1. ***SPYDER***

* *Spyder (formerly Pydee) is an* [*open source*](https://en.wikipedia.org/wiki/Open-source_software) *cross-platform* [*integrated development environment*](https://en.wikipedia.org/wiki/Integrated_development_environment) *(IDE) for scientific programming in the* [*Python language*](https://en.wikipedia.org/wiki/Python_(programming_language))*.*
* *Spyder integrates* [*NumPy*](https://en.wikipedia.org/wiki/NumPy)*,* [*SciPy*](https://en.wikipedia.org/wiki/SciPy)*,* [*Matplotlib*](https://en.wikipedia.org/wiki/Matplotlib) *and* [*IPython*](https://en.wikipedia.org/wiki/IPython)*, as well as other open source software.*
* *Spyder is extensible with plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as* [*Pyflakes*](https://en.wikipedia.org/w/index.php?title=Pyflakes&action=edit&redlink=1)*,* [*Pylint*](https://en.wikipedia.org/wiki/Pylint)*and Rope.*
* *It is available cross-platform through* [*Anaconda*](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))*, on Windows with WinPythonand Python(x,y).*

**THE END**