**Some tools used**

**Python**

**Python** is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python is **multipurpose**: it is not specialized to a specific target of users (like R for statistics, or PHP for web programming). It is extended through modules and libraries that hook very easily into the C programming language. For our purpose, we used a bunch of python libraries like tweepy, tkinter etc. a brief explanation for all of them has been given below. Python can be used for any programming task, from GUI programming to web programming with everything else in between. It's quite efficient, as much of its activity is done at the C level. For our purpose we have used the Python 3.4.3.

**Python csv module**

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. There is no “CSV standard”, so the format is operationally defined by the many applications which read and write it. The lack of a standard means that subtle differences often exist in the data produced and consumed by different applications. These differences can make it annoying to process CSV files from multiple sources. Still, while the delimiters and quoting characters vary, the overall format is similar enough that it is possible to write a single module which can efficiently manipulate such data, hiding the details of reading and writing the data from the programmer.

The [**csv**](https://docs.python.org/2/library/csv.html#module-csv) module implements classes to read and write tabular data in CSV format. It allows programmers to say, “Write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

The [**csv**](https://docs.python.org/2/library/csv.html#module-csv) module’s [**reader**](https://docs.python.org/2/library/csv.html#csv.reader) and [**writer**](https://docs.python.org/2/library/csv.html#csv.writer) objects read and write sequences. Programmers can also read and write data in dictionary form using the [**DictReader**](https://docs.python.org/2/library/csv.html#csv.DictReader) and **[DictWriter](https://docs.python.org/2/library/csv.html" \l "csv.DictWriter" \o "csv.DictWriter)** classes.

**PyCharm: Integrated Development Environment**

**PyCharm** is an Integrated Development Environment (IDE) used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django. PyCharm is developed by the Czech company Jet Brains.

It is cross-platform working on Windows, Mac OS X and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License. PyCharm Community Edition is less extensive than the Professional Edition.

**NLTK**

**NLTK** 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](http://nltk.org/nltk_data/) such as WordNet , along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, and an active [discussion forum](http://groups.google.com/group/nltk-users).NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

In our project, we used NLTK in cleaning module as well as Part of Speech Tagging.NLTK helped in data cleaning by providing the list of stopwords.NLTK further helped in the tagging of Part of speech such as verbs , adjectives etc.NLTK provides several modules for these type of tasks, such as Punkt Tokenizer module for tokenizing the given string.

**Tkinter**

The **Tkinter** module (“Tk interface”) is the standard Python interface to the Tk GUI toolkit.Both Tk and Tkinter are available on most Unix platforms, as well as on Windows and Macintosh systems. Starting with the 8.0 release, Tk offers native look and feel on all platforms.Tkinter consists of a number of modules. The Tk interface is provided by a binary extension module named **\_tkinter**. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.

The public interface is provided through a number of Python modules. The most important interface module is the **Tkinter** module itself. To use Tkinter, all we need to do is to import the **Tkinter** module. We made extensive use of Tkinter to provide an interactive graphical user interface to the user.

**Tweepy**

The python tweepy library has been used for interacting with the twitter and to download the twitter data.It is basically a data streaming library for tweeter.

Tweepy supports oauth authentication. Authentication is handled by the tweepy.AuthHandler class.

Tweepy provides access to the well documented Twitter API. With tweepy, it's possible to get any object and use any method that the official Twitter API offers.

Main Model classes in the Twitter API are Tweets, Users, Entities and Places. Access to each returns a JSON-formatted response and traversing through information is very easy in Python.

Tweepy Streaming API

One of the main usage cases of tweepy is monitoring for tweets and doing actions when some event happens. Key component of that is the StreamListener object, which monitors tweets in real time and catches them.

Tweepy REST API

Twitter provides the REST search API for searching tweets from Twitter’s search index. This is different than using the streaming filter API, in that the later is real-time and starts giving you results from the point of query, while the former is retrospective and will give you results from past, up to as far back as the search index goes (usually last 7 days). While the streaming API seems like the thing to use when you want to track a certain query in real time, there are situations where you may want to use the regular REST search API. We may also want to combine the two approaches, i.e. start 2 searches, one using the streaming filter API to go forward in time and one using the REST search API to go backwards in time, in order to get some on-going and past context for your search term.