**CHAPTER 1**

**INTRODUCTION**

**1.1 MACHINE LEARNING**

Machine learning is a branch of [artificial intelligence](http://en.wikipedia.org/wiki/Artificial_intelligence), concerns the construction and study of systems that can [learn](http://en.wikipedia.org/wiki/Learning) from data. For example, a machine learning system could be trained[4] on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam folders.

The core of machine learning deals with representation and generalization. Representation of data instances and functions evaluated on these instances are part of all machine learning systems. Generalization[6] is the property that the system will perform well on unseen data instances; the conditions under which this can be guaranteed are a key object of study in the subfield of [computational learning theory](http://en.wikipedia.org/wiki/Computational_learning_theory).

**1.1.1. ALGORITHM TYPES**

Machine learning [algorithms](http://en.wikipedia.org/wiki/Algorithm)[7] can be organized into a [taxonomy](http://en.wikipedia.org/wiki/Taxonomy_(general)) based on the desired outcome of the algorithm or the type of input available during training the machine

* [**Supervised learning**](http://en.wikipedia.org/wiki/Supervised_learning) algorithms are trained on labeled examples, i.e., input where the desired output is known. The supervised learning algorithm attempts to generalize a function or mapping from inputs to outputs which can then be used speculatively to generate an output for previously unseen inputs.
* [**Unsupervised learning**](http://en.wikipedia.org/wiki/Unsupervised_learning) algorithms operate on unlabelled examples, i.e., input where the desired output is unknown. Here the objective is to discover structure in the data (e.g. through a [cluster analysis](http://en.wikipedia.org/wiki/Cluster_analysis)), not to generalize a mapping from inputs to outputs.
* [**Semi-supervised learning**](http://en.wikipedia.org/wiki/Semi-supervised_learning) combines both labeled and unlabelled examples to generate an appropriate function or classifier.
* [**Transduction**](http://en.wikipedia.org/wiki/Transduction_(machine_learning)) tries to predict new outputs on specific and fixed (test) cases from observed, specific (training) cases.
* [**Reinforcement learning**](http://en.wikipedia.org/wiki/Reinforcement_learning) is concerned with how intelligent agents ought to act in an environment to maximize some notion of reward. The agent executes actions which cause the observable state of the environment to change. Through a sequence of actions, the agent attempts to gather knowledge about how the environment responds to its actions, and attempts to synthesize a sequence of actions that maximizes a cumulative reward.
* [**Learning to learn**](http://en.wikipedia.org/wiki/Learning_to_learn) learns its own [inductive bias](http://en.wikipedia.org/wiki/Inductive_bias) based on previous experience. Developmental, elaborated for [Robot learning](http://en.wikipedia.org/wiki/Robot_learning), generates its own sequences (also called curriculum) of learning situations to cumulatively acquire repertoires of novel skills through autonomous self-exploration and social interaction with human teachers, and using guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

**1.1.2. RELATED TOPICS**

**1.1.2.1. DATA MINING**

Data mining (the analysis step of the "Knowledge Discovery and Data Mining" process, or KDD), an interdisciplinary subfield of [computer science](http://en.wikipedia.org/wiki/Computer_science), is the computational process of discovering patterns in large [data sets](http://en.wikipedia.org/wiki/Data_set) involving methods at the intersection of [artificial intelligence](http://en.wikipedia.org/wiki/Artificial_intelligence), [machine learning](http://en.wikipedia.org/wiki/Machine_learning), [statistics](http://en.wikipedia.org/wiki/Statistics)[8], and [database systems](http://en.wikipedia.org/wiki/Database_system). The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

**1.1.2.2. ARTIFICIAL INTELLIGENCE**

Artificial intelligence (AI) is the intelligence exhibited by machines or software, and the branch of [computer science](http://en.wikipedia.org/wiki/Computer_science) that develops machines and software with human-like intelligence. Major AI researchers and textbooks define the field as "the study and design of intelligent agents", where an [intelligent agent](http://en.wikipedia.org/wiki/Intelligent_agent) is a system that perceives its environment and takes actions that maximize its chances of success. [John McCarthy](http://en.wikipedia.org/wiki/John_McCarthy_(computer_scientist)), who coined the term in 1955, defines it as "the science and engineering of making intelligent machines".

**1.1.2.3. STATISTICS**

Statistics is described as a mathematical body of science that pertains to the collection, analysis, interpretation or explanation, and presentation of [data](http://en.wikipedia.org/wiki/Data), or as a branch of [mathematics](http://en.wikipedia.org/wiki/Mathematics) concerned with collecting and interpreting data[8]. Because of its empirical roots and its focus on applications, statistics is typically considered a distinct mathematical science rather than as a branch of mathematics. Some tasks a statistician may involve are less mathematical; for example, ensuring that [data collection](http://en.wikipedia.org/wiki/Data_collection) is undertaken in a way that produces valid conclusions, coding data, or reporting results in ways comprehensible to those who must use them.

**1.1.2.4. DATA SCIENCE**

Data science is the study of the generalizable extraction of [knowledge](http://en.wikipedia.org/wiki/Knowledge) from [data](http://en.wikipedia.org/wiki/Data), yet the key word is science. It incorporates varying elements and builds on techniques and theories from many fields, including [signal processing](http://en.wikipedia.org/wiki/Signal_processing), [mathematics](http://en.wikipedia.org/wiki/Mathematics), [probability models](http://en.wikipedia.org/w/index.php?title=Probability_models&action=edit&redlink=1), [machine learning](http://en.wikipedia.org/wiki/Machine_learning), [computer programming](http://en.wikipedia.org/wiki/Computer_programming), [statistics](http://en.wikipedia.org/wiki/Statistics), [data engineering](http://en.wikipedia.org/wiki/Data_engineering), [pattern recognition and learning](http://en.wikipedia.org/wiki/Pattern_recognition_and_learning), visualization, uncertainty, [data warehousing](http://en.wikipedia.org/wiki/Data_warehousing), and [high performance computing](http://en.wikipedia.org/wiki/High_performance_computing) with the goal of extracting meaning from data and creating data products. Data science is a [buzzword](http://en.wikipedia.org/wiki/Buzzword), often used interchangeably with [analytics](http://en.wikipedia.org/wiki/Analytics) or [big data](http://en.wikipedia.org/wiki/Big_data), that is often abused for marketing anything involving data processing, in particular to re-brand existing [competitive intelligence](http://en.wikipedia.org/wiki/Competitive_intelligence) and analytics approaches. Data Science need not be always for big data, however, the fact that data is scaling up makes big data an important aspect of data science.

**1.1.2.5. BIG DATA**

Big datais the term for a collection of [data sets](http://en.wikipedia.org/wiki/Data_set) so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture, curation, storage, search, sharing, transfer, analysisand visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, determine quality of research, prevent diseases, [link legal citations](http://en.wikipedia.org/wiki/Legal_citation_analysis), combat crime, and determine real-time roadway traffic conditions."

**1.1.2.6. CLOUD**

In [computer networking](http://en.wikipedia.org/wiki/Computer_networking), cloud computing is [computing](http://en.wikipedia.org/wiki/Computing) that involves a large number of computers connected through a communication [network](http://en.wikipedia.org/wiki/Computer_network) such as the [Internet](http://en.wikipedia.org/wiki/Internet), similar to [utility computing](http://en.wikipedia.org/wiki/Utility_computing). In science, cloud computing is a synonym for [distributed computing](http://en.wikipedia.org/wiki/Distributed_computing) over a network, and means the ability to run a program or application on many connected computers at the same time.

**1.2. OPINION MINING**

Opinion mining is a type of natural language processing for tracking the mood of the public about a particular product. Opinion mining, which is also called sentiment analysis, involves building a system to collect and examine opinions about the product made in blog posts, comments, reviews or tweets.  Automated opinion mining often uses machine learning, a component of artificial intelligence (AI). An opinion mining system is often built using software that is capable of extracting knowledge from examples in a dataset and incorporating new data to improve performance over time.  The process can be as simple as learning a list of positive and negative words, or as complicated as conducting deep parsing of the data in order to understand the grammar and sentence structure used.

The most common application of opinion is in the area of reviews of consumer products and services. There are many websites that provide automated summaries of reviews about products and about their specific aspects. A common practice for e-commerce web sites to enable their customers is to write reviews of products that they have purchased. Such reviews[2] provide valuable sources of information on these products. They are used by potential customers to find opinions of existing users before deciding to purchase a product. They are also used by product manufacturers to identify problems of their products and to find competitive intelligence information about their competitors.

**1.2.1 APPLICATIONS OF OPINION MINING**

Opinions are so important that whenever one needs to make a decision, one wants to hear others’ opinions. This is true for both individuals and organizations. The technology of opinion mining thus has a tremendous scope for practical applications.

* **Individual consumers**: If an individual wants to purchase a product, it is useful to see a summary of opinions of existing users so that he/she can make an informed decision. This is better than reading a large number of reviews to form a mental picture of the strengths and weaknesses of the product. He/she can also compare the summaries of opinions of competing products, which is even more useful.

* **Organizations and businesses**: Opinion mining is equally, if not even more, important to businesses and organizations. For example, it is critical for a product manufacturer to know how consumers perceive its products and those of its competitors. This information is not only useful for marketing and product benchmarking but also useful for product design and product developments.

**1.2.2 OPINION SPAM DETECTION**

Anyone from anywhere in the world can freely express his/her views and opinions without disclosing his/her true identify and without the fear of undesirable consequences. These opinions are thus highly valuable. However, this anonymity also comes with a price. It allows people with hidden agendas or malicious intentions[1] to easily game the system to give people the impression that they are independent members of the public and post fake opinions to promote or to discredit target products, services, organizations, or individuals without disclosing their true intentions, or the person or organization that they are secretly working for. Such individuals are called opinion spammers and their activities are called opinion spamming

**1.2.2.1 TYPES OF SPAM REVIEWS**

* Undeserving positive opinions to some target products in order to promote them and/or by giving unjust or malicious negative reviews to some other products in order to damage their reputation.
* The second type consists of non reviews (e.g., ads) which contain no opinions on the product.

**1.2.2.2 CHALLENGES FACED IN SPAM REVIEWS**

* The first is that a word that is considered to be positive in one situation may be considered negative in another situation[2]. Take the word "long" for instance. If a customer said a laptop's battery life was long, that would be a positive opinion.  If the customer said that the laptop's start-up time was long, however, that would be is a negative opinion. These differences mean that an opinion system trained to gather opinions on one type of product or product feature may not perform very well on another.
* A second challenge is that people don't always express opinions the same way. Most traditional text processing relies on the fact that small differences between two pieces of text don't change the meaning very much.  In opinion mining, however, "the movie was great" is very different from "the movie was not great".
* Finally, people can be contradictory in their statements. Most reviews will have both positive and negative comments, which is somewhat manageable by analyzing sentences one at a time. However, the more informal the medium (twitter or blogs for example), the more likely people are to combine different opinions in the same sentence.

**1.2.2.3 SPAM FILTER**

Spam blocker is the phrase used to describe a type of computer software that is designed to prevent [spam](http://www.webopedia.com/TERM/S/spam.html), or unsolicited mail sent in bulk to a large number of recipients, from going directly to the inbox of your [e-mail client](http://www.webopedia.com/TERM/E/e_mail_client.html).

Also known as spam filters, spam fighters or [anti-spam](http://www.webopedia.com/TERM/A/anti_spam.html) utilities, spam blocking [software](http://www.webopedia.com/TERM/S/software.html) has evolved from simply watching for keywords in the subject line of messages to using advanced technology like [Bayesian](http://www.webopedia.com/TERM/B/Bayesian_filter.html) and similar [heuristic](http://www.webopedia.com/TERM/H/heuristic_programming.html) filters[9] to help identify spam through suspicious word patterns or word frequency.

**1.3 EXISTING SYSTEM**

In the existing system, the spam reviews are identified by using heuristic rules where the helpfulness votes by the customer and rating deviation alone are considered. This limits the performance of the system.

Also the existing system uses naive bayes classifier for spam and non-spam classification where the accuracy is about 68% which may not provide the accurate results for the user.

**1.4 PROPOSED SYSTEM**

The system proposes to identify the spam reviews by considering not only the review related features in turn identifying spammers as well as opinion spam using two-view process. Since the features other than helpfulness and rating deviation like normalized length of review, date of posting, review similarity and ratio of personal and second pronouns are also considered , the accuracy of the identified fake reviews are ensured. Similarly in identifying spammers, features like range of brands reviewed by user, multiple reviews for the same product by user are considered.

The system attempts to classify the reviews obtained from freely available Flipkart datasets and the crawled Flipkart dataset with a greater accuracy using SVM algorithm. In order to improve the accuracy, the additional features like user rank, helpfulness[1] of the review, overall review of product also features[2] like similar id ,brand specific reviews, comparison of the sentiment of the review with the overall sales rank are used in addition to the review details.

A classifier is built based on the identified features from which each feature is assigned a weight depending on the classified training sets. Support Vector Machine Classifiers[2], which find a high dimensional separating hyper plane between two groups of data, is trained using the features extracted. Since the system results in a two-class classification (spam or non-spam) Linear Support Vector Machine is employed. Linear SVM Classifier is used to find out overall weight for a given review. Linear SVM classification[3] algorithm determines whether a given review is spam or non spam by mapping the calculated weights to a two dimensional space as vector points. Vector points which fall under the threshold are marked as spam reviews.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 INTRODUCTION**

One aspect of these user expressions is the reviews beneath a product in a specific e-commerce website (e.g. amazon and iOS app store). This kind of contents and the similar ones are called user-generated data (vs. contents provided by web site owners). Reviews contain rich user opinions on products and services. Potential customers read these reviews to find existing users’ opinions before deciding to purchase a product. They are also used by product manufacturers to identify product advantages and disadvantages and/or to find marketing intelligence information about their competitors [3].

It is now well understood that these data are very valuable and can be used for many applications [4]. So there is a great interest for extracting information from these data in recent years from both consumers and manufacturers. Unfortunately, most of the attempts in opinion mining have focused on the different algorithms and classification methods. In other words, the concentration is mostly on the application of opinion mining (e.g. [5], [6], [7]). Of course it is an important area, but trustworthiness of online opinions or opinion spam is rarely regarded. When we limit the area of research to product review spam, we understand that it is almost neglected so far except some few papers that are mostly done by analogous workgroups.

If the spammers write positive reviews, they may increase the chance of sale and if they write negative reviews, one will most likely buy a different product. The issue is more serious when the spammers write negative reviews beneath a good product and good ones beneath a bad or harmful product. It worth knowing that 40% of people in modern world rely on these reviews [8].

**2.2 NATURAL LANGUAGE PROCESSING**

The opinion mining problem is a mixed problem of information retrieval (IR) and Natural Language Processing (NLP) as are summarized below:

* **Analysis of linguistic resources for OM**

Knowing the linguistic terms helps getting the idea from the text. Classification of contents into positive and negative, and subjective and objective terms is the basic problem.

* **Text features identification and orientation**

Categorizing phrases as noun, verbs, adjectives and adverbs (part of speech). Some other tasks are stop words removal, fuzzy pattern matching.

* **Adjectives, noun, verbs and adverbs**

It discusses the importance of each part of a phrase to classify comparative sentences. The result shows the main focus is on adjective and adverbs.

* **Semantic orientation of text**

It is the classification of sentence according to its meaning and background knowledge. This is a supplement to the key role, syntactic analysis.

* **Ontology based learning**

The relationship between terms in text helps in understanding the background knowledge. Ontology, as a growing area, integrates the domain knowledge of individual words into the terms for learning and capturing concept from text.

**2.3 RELATED WORK**

Ramili M. and Parandini M. [10] implemented a client/server system that detects the spam comments. They categorized undesired messages into two categories

* Link Spam: Tries to convey advertisements or other messages, not related to the topic, to the reader.
* Comment Spam: Subtler, apparently innocuous messages that are injected to false the “real” feeling of the community on the topic.

Bing Liu and his workgroups are the most active ones in this field. It is believed that the reference [4], by Bing Liu and Nitin Jindal, in 2007 is the first published paper in review spam detection. Their investigation on the topic is based on Amazon dataset crawled from Amazon website (amazon.com). It has 5.8 million reviews and 2.14 million reviewers (members who wrote at least one review). In an analogous paper [11], they published their work in more detail in 2008. They mentioned the differences between email spam and web spam. Then they categorized spam review into three types and then followed a distinct strategy in each type for their detection.

* **Type 1 (untruthful opinions)**

Those that deliberately misdirect readers or OM systems by giving positive reviews to promote the objects (i.e. hyper spam ) and/or by giving unjust or malicious negative reviews to the objects to damage their reputation (i.e. defaming spam). This type is also known as fake reviews or bogus reviews.

* **Type 2 (reviews on brand only)**

Those that comment on the brand instead of commenting on the products. Although they may be useful but are often biased.

* **Type 3 (non-reviews)**

Those that are not reviews. It has two main sub-types:

(1) advertisements and

(2) other irrelevant reviews containing no opinions (e.g. questions and random texts).

The classification problem is different in type 1 and the other types. Because type 2 and type 3 are easily recognizable by humans, it can be classified by supervised classification. Type 3 is much harder to distinguish (even for humans). They used duplicate reviews as the main indicator for this type, since they found that untruthful reviews are likely to be reposted repeatedly.

A classifier for this kind of spam developed in [12]. They emphasized on psycholinguistic methods and text analysis. They also confirmed that deceptive spam is hardly identified by humans. Identifying spam in iOS app store is studied in [13]. The paper classifies spam app in a supervised setting with limited labeled data, and to cluster reviews in an unsupervised setting. They use simple Linear Gaussian parameterization on the labeled dataset.Instead of finding review spam [14] attempts to identify spammers (reviewers who spread spam) by looking at suspicious rating behavior. it believes that spammers, first, may target specific products and product groups in order to maximize their impact. Then they try to deviate from the other reviewers in their ratings. Researchers use scoring methods to measure the degree of spam.

Finding unusual review patterns using unexpected rules are checked in [15]. This paper represents suspicious behavior of reviewers. First they define the concept of “unexpectedness”. They mean it as the deviation from expectation. For example, if a reviewer wrote reviews that all of them are negative on products of a brand but other reviewers are generally positive about the brand; this reviewer is clearly a spam suspect.

**CHAPTER 3**

**REQUIREMENT SPECIFICATIONS**

**3.1 INTRODUCTION**

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements.

**3.1.1 HARDWARE SPECIFICATION**

* **Personal Computer**

Processor type : Intel Pentium series or I3, I5

Speed : 2.4 GHz

RAM : 2GB RAM

Hard disk : 20 GB HD (Minimum)

**3.1.2 SOFTWARE SPECIFICATION**

* Win2000/XP/Vista/7
* Python27
* IDE- Komodo 8
* Apache HTTP Server 2.2
* Sqlit

**3.2 TECHNOLOGIES USED**

**3.2.1 PYTHON**

Python is a widely used [general-purpose](http://en.wikipedia.org/wiki/General-purpose_programming_language), [high-level programming language](http://en.wikipedia.org/wiki/High-level_programming_language). Its design philosophy emphasizes code [readability](http://en.wikipedia.org/wiki/Readability), and its syntax allows programmers to express concepts in fewer [lines of code](http://en.wikipedia.org/wiki/Lines_of_code) than would be possible in languages such as [C](http://en.wikipedia.org/wiki/C_(programming_language)). The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple [programming paradigms](http://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](http://en.wikipedia.org/wiki/Imperative_programming) and [functional programming](http://en.wikipedia.org/wiki/Functional_programming) or [procedural](http://en.wikipedia.org/wiki/Procedural_programming) styles.

Scrapy is a web crawling framework with support for [web scraping](http://en.wikipedia.org/wiki/Web_scraping). It is [open-source](http://en.wikipedia.org/wiki/Open-source) and written in [Python](http://en.wikipedia.org/wiki/Python_(programming_language)). It is controlled using command line tools, that can be used to trigger the scrapers written in Python.

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers. Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

import smtplib

smtpObj = smtplib.SMTP( [host [, port [, local\_hostname]]] )

Here is the detail of the parameters:

* **host:** This is the host running your SMTP server. You can specifiy IP address of the host or a domain name like tutorialspoint.com. This is optional argument.
* **port:** If you are providing *host* argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.
* **local\_hostname**: If your SMTP server is running on your local machine, then you can specify just*localhost* as of this option.

An SMTP object has an instance method called **sendmail**, which will typically be used to do the work of mailing a message. It takes three parameters:

* The *sender* - A string with the address of the sender.
* The *receivers* - A list of strings, one for each recipient.
* The *message* - A message as a string formatted as specified in the various RFCs.

**3.2.2 SQL**

Structured Query Languageis a [special-purpose programming language](http://en.wikipedia.org/wiki/Special-purpose_programming_language) designed for managing data held in a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). SQL consists of a [data definition language](http://en.wikipedia.org/wiki/Data_definition_language) and a [data manipulation language](http://en.wikipedia.org/wiki/Data_manipulation_language). The scope of SQL includes data insert, query, [update and delete](http://en.wikipedia.org/wiki/Data_Manipulation_Language), [schema](http://en.wikipedia.org/wiki/Database_schema) creation and modification, and data access control.

The most common operation in SQL is the query, which is performed with the declarative [SELECT](http://en.wikipedia.org/wiki/Select_(SQL)) statement. SELECT retrieves data from one or more [tables](http://en.wikipedia.org/wiki/Table_(database)), or expressions. SQL is designed for a specific purpose: to query [data](http://en.wikipedia.org/wiki/Data) contained in a [relational database](http://en.wikipedia.org/wiki/Relational_database). SQL is a [set](http://en.wikipedia.org/wiki/Set_(computer_science))-based, [declarative](http://en.wikipedia.org/wiki/Declarative_programming) query language, not an [imperative language](http://en.wikipedia.org/wiki/Imperative_programming) like [C](http://en.wikipedia.org/wiki/C_(programming_language)) or [BASIC](http://en.wikipedia.org/wiki/BASIC_programming_language).

SQLite is a software library that implements a [self-contained](https://sqlite.org/selfcontained.html), [serverless](https://sqlite.org/serverless.html), [zero-configuration](https://sqlite.org/zeroconf.html), [transactional](https://sqlite.org/transactional.html) SQL database engine.

**3.2.3 HTML**

Hypertext Markup Language is the main [markup language](http://en.wikipedia.org/wiki/Markup_language) for creating [web pages](http://en.wikipedia.org/wiki/Web_page) and other information that can be displayed in a [web browser](http://en.wikipedia.org/wiki/Web_browser). HTML elements form the building blocks of all [websites](http://en.wikipedia.org/wiki/Website). HTML allows [images and objects](http://en.wikipedia.org/wiki/Img_(HTML_element)) to be embedded and can be used to create [interactive forms](http://en.wikipedia.org/wiki/Fieldset). It provides a means to create [structured documents](http://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](http://en.wikipedia.org/wiki/Semantic) for text such as headings, paragraphs, lists, [links](http://en.wikipedia.org/wiki/Hyperlink), quotes and other items. It can embed [scripts](http://en.wikipedia.org/wiki/Scripting_language) written in languages such as [JavaScript](http://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML web pages.

**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 INTRODUCTION**

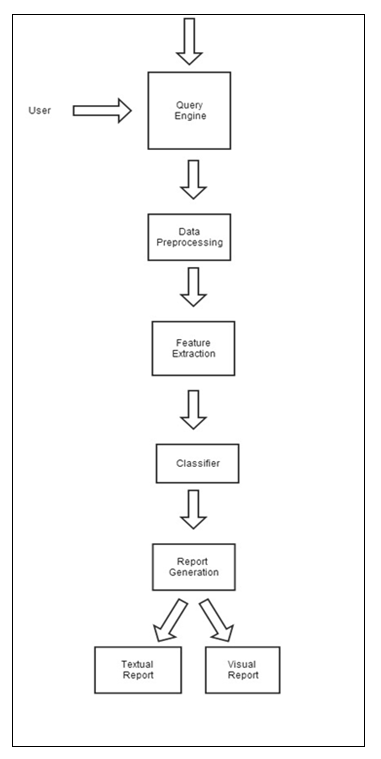
Systems design is the process of defining the architecture, components, modules, interfaces, and [data](http://en.wikipedia.org/wiki/Data) for a [system](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement). System design is a problem solving process in which the designer applies knowledge and experience to produce a conceptualization that defines and describes a solution to a problem. System design helps to produce a description of the software’s internal construction and describe the software’s architecture. System design also bridge the gap between software requirements and software code.

* Scope Definition  
                 The system provides services to the customers of the e-commerce websites who wish to buy a product and rely on the reviews of the product to make the decision of buying the product.
* Problem analysis  
   The system should be able to accurately classify the reviews as spam and non spam   
       The user and the product details are also to be analyzed in order to classify accurately
* Decision analysis  
                 The classification can be done by considering a number of review and reviewer features

**4.3 ARCHITECTURE DESIGN:**

The system architecture diagram enables graphically modeling the applications of a system. It specifies the components, interfaces and the interactions with other components.

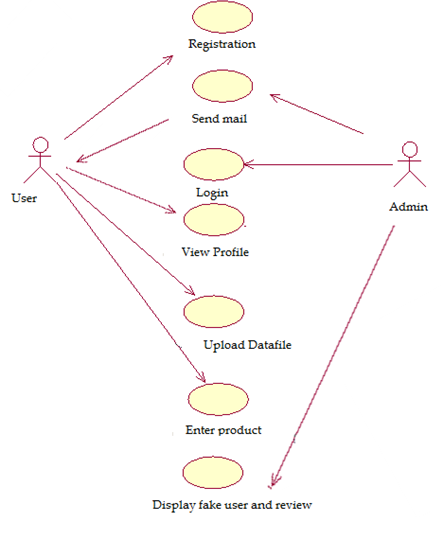
**4.3.1 SYSTEM ARCHITECTURE DIAGRAM**



**Fig.4.3.1 SYSTEM ARCHITECTURE DIAGRAM**

**4.4. USE CASE DIAGRAM**

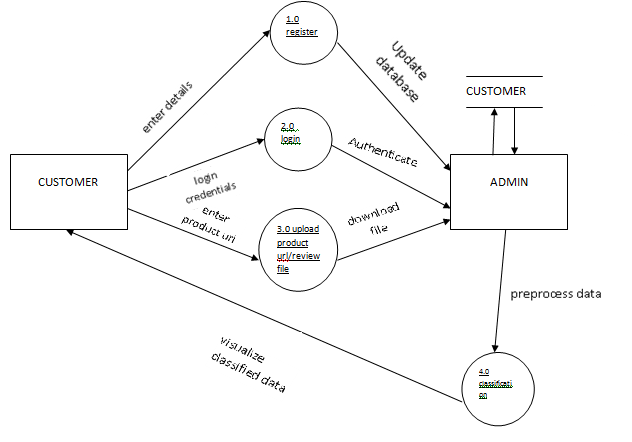
A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a [use case](http://en.wikipedia.org/wiki/Use_Case). A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.



**Fig.4.4 USECASE DIAGRAM**

**4.5. DATA FLOW DIAGRAM**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system), modeling its *process* aspects. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

****

**Fig.4.4 DATA FLOW DIAGRAM**

**CHAPTER 6**

**SYSTEM DESIGN**

**5.1 DESIGN OVERVIEW**

In order to provide a complete overview of the system, the system had to be the broken down to gain insight into its compositional sub-systems. Thus the system has been developed using Top down approach. Problem was analyzed and it was decided to identify one or two single minded function that can be seamlessly integrated to achieve the overall objective of the project. Some of the key functional requirements for the given problems are

* User should register before being able to avail the service of the system
* Input can be given as a data file in the user desktop or as a product  url
* Classify spam and non spam
* Recommend the reliable reviews to the user

So, it was decided to cull out the functionalities and their associated side effects as separate modules. Then modules were identified to satisfy the single entry style exit criterion.

After careful analysis and the application of top down strategy the following modules were identified which were expected to exhibit Functional coupling and unit cohesion.

* Data Preprocessing
* Feature Extraction
* Building Classifier
* Training and Visualization

**5.2 MODULE DESCRIPTION**

**5.2.1 DATA PREPROCESSING**

**5.2.1.1 Functionality**

Data preprocessing includes removal of irrelevant and redundant information present or noisy and unreliable data from the uploaded file or the file downloaded from the given URL.

* **Sentence tokenization**

The entire review is given as input and it is tokenized into sentences using NLTK package.

* **Removal of punctuation marks**

Punctuation marks used at the starting and ending of the reviews are removed along with the removal of additional white spaces. The URL links are also replaced with URL keyword.

* **Word Tokenization**

Each individual review is tokenized into words and stored in a list for

easier retrieval.

* **Removal of stop words**

Affixes are removed from the stem. For example, the stem of "cooking" is "cook", and the stemming algorithm knows that the "ing" *suffix* can be removed.

**5.2.1.2 Input Parameters**

The input to the data processing module can be given in two ways

* Uploading the review file
* Entering the URL of the product

**5.2.1.3 Output Parameters**

Irrelevant and unreliable data are removed thus generating noise free reviews.

**5.2.1.4 Exceptions**

Stemming and lemmatisation may generate words which have no meaning. The unmeaningful words are replaced using WordNet.

**5.2.2 FEATURE EXTRACTION**

**5.2.2.1 Functionality**

The preprocessed data is converted into a set of features by applying certain parameters. The following features are extracted:

* Content features

1. Normalized length of the review: Fake reviews tend to be of smaller length.

2. Ratio of personal pronouns vs second personal pronouns: Fake reviews provide more recommendation.

* Sentiment features

1. Subjective vs Objective ratio: Advertisers post fake reviews with more of objective information

2. Positive sentiment vs negative sentiment: The sentiment of the review is analyzed and it is compared with average sales rank.

* Product feature

1. Salerank: Salesrank of the fake review shows large deviation from the average salesrank.

* Metadata feature

1. Range of brands or products reviewed: Checks if user is interested only in the specific brand and product or other products too.

2. Reviewer ID: Same user posting review for the same product.

**5.2.2.2 Input Parameters**

* Content features

The preprocessed review is taken as input for extracting content features.

* Sentiment features

The preprocessed review is taken as input for extracting sentiment features.

* Product feature

The salesrank of each of the product in the review file and the average salesrank from the product info file is taken as input.

**5.2.2.3 Output Parameters**

The reviewer id along with the status about the feature, whether or not the feature is present in the review is entered into the database.

**5.2.2.4 Exceptions**

Reading helpfulness value from .csv file requires conversion from string (eg. “96%”) to float literal. CSV file stores all values as string literals.

**5.2.3 CLASSIFICATION**

**5.2.3.1 Functionality**

Classification assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. Each data in the review file is assigned a weight and depending upon which it is classified into respective classes- Spam and non-spam.

**5.2.3.2 Input Parameters**

The database file generated in the feature extraction module is taken as input. Overall weight is computed for an individual review.

**5.2.3.3 Output Parameters**

If the overall weight is more than the threshold weight, the review is classified under the class spam else non-spam class. Fake users are also identified and displayed.

**5.2.3.4 Exception**

Threshold frequency computation requires removal of out layers which may otherwise lead to incorrect calculation of threshold frequency.

**5.2.4 TRAINING AND VISUALIZATION**

**5.2.4.1 Functionality**

Training involves training the built classifier with the already existing samples with known classes. The main steps involved in training are

* Determining the boundaries between each class
* Determine the threshold for which the boundaries return the most distinctive results when determining class membership.

There can be any number of classes. A term vector is a representation of all of the terms (as defined by the index options) in a node. Therefore, classes consist of sets of term vectors which have been deemed similar enough to belong to the same class. The samples for each class should be statistically relevant, and should have samples that include both solid examples of the class (that is, samples that fall well into the positive side of the threshold from the class boundary) and samples that are close to the boundary for the class. The samples close to the boundary are very important, because they help determine the best thresholds for your content.

Visualization involves plotting the classified data points on a two dimensional space considering only two features simultaneously in order to avoid cluttering based on the users choice. If many features are plotted simultaneously, it leads to visual cluttering whereby the threshold cannot be determined accurately.

**5.2.4.2 Input Parameters**

* **Feature file**

A file consisting of the collection of feature vectors

* **Feature vector**

A two dimensional vector representing the weights assigned to the two features selected by the user is given as input.

**5.2.4.3 Output Parameters**

Training and test data points are plotted on 2D planes. Each class is represented using different colors and threshold separating the spam and non-spam is marked on the plot.

**5.2.4.3 Exceptions**

Some of the plots which are non spam may fall into the plot region of spam which are the exception caused due to the feature points.

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 IMPLEMENTATION PLAN**

* New user has to register with the site in order to avail the features of Spam Fighter System.
* Confirmation mail is sent to the user after registration.
* Registered user can login to the website by providing username and password credentials.
* The home page consists of navigation to the profile, review file upload and URL link page.
* The profile page allows the user to check in their details entered during registration.
* The review file upload page allows the user to upload the review file for further processing.
* The URL page allows the user to enter the URL link of the product for which fake reviews have to be identified.
* Both the uploaded file and file generated from the URL is preprocessed to produce clean and noise free data.
* Feature are extracted from the file and weight is assigned to each feature .
* Overall weight for all users and reviews is computed.
* All users and reviews falling below a particular threshold are marked as fake users and fake reviews.
* Links are provided for accessing the list of fake users and fake reviews.
* Visualization plan is drawn for the generated dataset and the points are plotted on the plot.

**6.2. IMPLEMENTATION ISSUES**

**6.2.1 DATA PREPROCESSING**

**6.2.1.1 Problems Faced**

* The preprocessing of data includes word tokenization which breaks down the sentences to words. Stemming is a technique for removing affixes from a word, ending up with the stem. For example stemming of cooking is cook. The resulting stems are not always a valid word. For example, the stem of cookery is cookeri which doesn’t have any meaning.
* **Illegal values**

1. The dataset file uploaded may contain duplicate rows which needed to be eliminated and it had to be checked whether its only the reviews repeating or all the attributes.
2. The price of the product specifies in the dataset varied from float type to integer type
3. Certain values of the attributes were missing which created exceptions

* **Misspellings**

1. The reviews by the users contained certain typographical errors which could not be recognized by the system
2. The preprocessing of large dataset occupied more time and memory.

**6.2.1.2 Solution to Problem**

* **To overcome Illegal values:**

1. Duplicate rows are removed by considering the review id which is unique and the row containing the same review id is removed
2. The price attribute is normalized in such a way that all values are converted to floatdata type
3. The missing attribute values were replaced by null characters

* **To overcome misspellings**

1. The typographical errors were removed by spellcheck module which compared the words with the dictionary and replaced the wrongly spelled words with the rightly spelled words
2. The large data set is sampled into smaller instances in such a way that a majority of the instances are of low helpfulness review instances as the probability of being spam is more in such cases.

**6.2 FEATURE EXTRACTION**

**6.2.2.1 Problems Faced**

1. If the helpfulness value of the review is very less, it cannot be concluded that the review is spam because the user tend to rate only the first few reviews of the product.
2. In determining the polarity of the review, commonly used technique is comparing the ratio of positive words and negative words. However in sentences like "I refuse to accept it to be good" , though this technique may result in neutral polarity , it is a negative sentiment sentence.
3. In calculating the high similarity score ratio feature, the similarity count increases only if the words are repeated exactly however a spammer may intend to express the same opinion through different words depicting the same meaning

**6.2.2.2 Solutions to Problems**

1. The helpfulness value is assigned a relatively lower weight.
2. Part of speech tagging of the sentence is done and the following conditions are checked.

|  |  |  |
| --- | --- | --- |
| **Verb polarity** | **Adjective polarity** | **Overall sentiment** |
| Positive | Negative | Negative |
| Negative | Negative | Positive |
| Negative | Positive | Positive |

3. Word net is used and the words are replaced by their synonym, now the similarity score can be easily calculated.

**6.2.3 CLASSIFICATION**

**6.2.3.1 Problems Faced**

1. Since the reviews are classified based on the overall weights for a review instance however if a weight of one particular feature alone is considered the classification may differ

**6.2.3.2 Solution to Overcome**

1. The instances which result in outlays are removed to provide a clearer classification.

**6.2.4 TRAINING AND VISUALIZATION**

**6.2.4.1 Problems Faced**

1. If all the features are plotted on a 2D graph it will give a cluttered appearance and will be difficult to understand the classification .

**6.2.4.2 Solution to Overcome**

1. Only two features are considered at a time simultaneously , the two features are selected based on user preference

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENTS**

**7.1 CONCLUSION**

The spam filter designed for filtering fake reviews by using SVM classification provided a better accuracy of classification than the existing Naïve Bayes classification. This system was applied over other sampled instances of the dataset and the features identified contributed to the accuracy of the classification.

Also the system provides the user with a functionality to recommend the most truthful reviews to enable them to make decisions about the product.

**7.2 FUTURE ENHANCEMENT**

The system currently filters the fake reviews of flipkart website an can be extended to other product websites and also to eliminate other types of spams like blog spam. Also, the system can be converted into mashups and be used as a android application.

**APPENDICES**

**APPENDIX 1**

**SAMPLE CODING**

* **Python code for crawling product details from the entered URL Downloadprod.py**

#!c:\Python27\python.exe

#!/usr/bin/env python

import cgi, cgitb

form = cgi.FieldStorage()

file=open('name.txt','r')

for i in file:

a=i

link = form.getvalue('link')

print "Content-Type: text/html"

print

print """

<html>

<head>

<title>Spam Fighter</title>

<link href='/style.css' rel='stylesheet' type='text/css' />

</head>

<body>

<!--top part start -->

<div id="top">

<a href="/homepage.html"><img src='/spamlogo.jpg' alt='individual' width='286' height='66' border='0' /></a>

<ul>

<li><a href="#">home</a></li>

<li><a href="#">about us</a></li>

<li><a href="#">contact us</a></li>

</ul>

</div>

<!--top part end -->

<!--header start -->

<div id="header">

<h2><span>Why Spam Fighter???</span></h2>

<p>It has become a common practice for people to read online opinions/reviews for different purposes. For example, if one wants to buy a product, one typically goes to a review site (e.g., amazon.com) to read some reviews of the product. If most reviews are positive, one is likely to buy the product. If most reviews are negative, one will almost certainly not buy it. Positive opinions can result in significant financial gains and/or fames for busineses, organizations and individuals. This, unfortunately, gives strong incentives for opinion spamming.</p>

</div>

<!--header end -->

<!--body start -->

<div id="body" style="color:#000">

<br class="spacer" style="color:#000" />

<!--left panel start --><!--left panel end -->

<!--mid panel start --><!--mid panel end -->

<!--right panel start --><!--right panel end -->

<!--bodyBottom start --><!--bodyBottom end-->

<blockquote>

<blockquote>

<font color="#7642A3" size="+2"> <b>Hello """+a+"""</font></b></center><a href="/homepage.html" ><img src='/exit.jpg' align='right'></a><a href="/cgi-bin/backbutton.py"><img src='/back.jpg' align='right'></a>

<center><center><b><font color="#7642A3">File downloaded successfully!!</font></b></center>

<table width="200" border="0" cellpadding="25">

<tr>

<td><center><a href="/cgi-bin/urlfake.py" ><bold><font color="#7642A3">FAKE USERS(LIST)</font></bold></a></center></td>

<td><center><a href="/cgi-bin/urlfake.py" ><bold><font color="#7642A3">FAKE USERS(PLOT)</font></bold></a></center></td>

</tr>

<tr><td><center><a href="/cgi-bin/urlfakereview.py" ><bold><font color="#7642A3">FAKE REVIEWS(LIST)</font></bold></a></center></td>

<td><center><a href="/cgi-bin/plotreview.py" ><bold><font color="#7642A3">FAKE REVIEWS(PLOT)</font></bold></a></center></td> </tr>

</table>

</center>

<center>

</blockquote>

</blockquote>

<br class="spacer" />

<br class="spacer" />

<br class="spacer" />

<br class="spacer" />

<br class="spacer" />

<br class="spacer" />

</div>

<!--body end -->

<!--footer start --><!--footer end -->

</body>

</html>

"""

from twisted.internet import reactor

from scrapy import log, signals

from scrapy.crawler import Crawler

from scrapy.settings import Settings

from scrapy.xlib.pydispatch import dispatcher

import logging

from reviewitem import ReviewCrawler

from scrapy.spider import Spider

from scrapy.selector import HtmlXPathSelector

from scrapy.http import Request

import urlparse

from items import SampleCrawler

from reviewitem import ReviewCrawler

import csv

csk = csv.writer(open("OMREVIEW.csv", "wb"))

x=" "

class MySpiderd(Spider):

name = "saibaba123"

allowed\_domains=["flipkart.com"]

start\_urls=[link]

def parse(self,response):

cs = csv.writer(open("OM.csv", "wb"))

hxs=HtmlXPathSelector(response)

items=[]

item=SampleCrawler()

name=hxs.select('//div[@class="mprod-summary-title fksk-mprod-summary-title"]/h1')

item['pid']=hxs.select('//img/@data-pid').extract()

x=item['pid']

item['name']=name.select('normalize-space(./text())').extract()

item['avgrate']=hxs.select('//div[@class="pp-big-star"]/text()').extract()

item['listprice']=hxs.select('//span[@class="price list old-price"]/text()').extract()

item['salesprice']=hxs.select('//span[@class="fk-font-verybig pprice fk-bold"]/text()').extract()

br=hxs.select('//table[@class="fk-specs-type2"]')

brn=br[0].select('//td')

#l=br.index('Brand')

#item['brand']=br[l+1]

#for i in brn:

# print i.select('./text()').extract()

brand=brn.select('./text()').extract()

n=brand.index('Brand')

item['brand']=brand[n+1]

items.append(item)

cs.writerow([item['pid'][0], item['name'][0],item['brand'][0],item['listprice'][0],item['salesprice'][0],item['avgrate'][0]])

dum=[]

name=hxs.select('//div[@class="fclear line bmargin10"]/a/@href').extract()

print name

url=''.join(name)

abc="http://flipkart.com"+url+"&start=0&sort\_order=most-helpful"

yield Request(abc,self.par\_ur)

def par\_ur(self,response):

hxs=HtmlXPathSelector(response)

names=hxs.select('//a[@class="load-user-widget fk-underline"]')

items=[]

n=0

i=0

z=len(names)

while(i<9):

item=ReviewCrawler()

revstar=hxs.select('//div[@class="fk-stars"]')

p=hxs.select('//div[@class=" product-unit unit-3 browse-product "]/@data-pid')

item['pid']=p.extract()

item['rating']=revstar[i].select('normalize-space(@title)').extract()

revname=hxs.select('//a[@class="load-user-widget fk-underline"]')

item['userid']=revname[i].select('normalize-space(./text())').extract()

revdate=hxs.select('//div[@class="date line fk-font-small"]')

item['date']=revdate[i].select('normalize-space(./text())').extract()

revoverview=hxs.select('//div[@class="line fk-font-normal bmargin5 dark-gray"]')

item['title']=revoverview[i].select('normalize-space(strong/text())').extract()

revreview=hxs.select('//p[@class="line bmargin10"]')

item['rbody']=revreview[i].select('normalize-space(./text())').extract()

h=hxs.select('//div[@class="unit"]/strong/text()')

revid=hxs.select('//div[@class="review-list"]/a/@name')

item['reviewid']=revid[i].extract()

item['helpful']=h[n].extract()

n=n+1

item['totalreview']=h[n].extract()

n=n+1

items.append(item)

i=i+1 csk.writerow([item['pid'][0],item['rating'][0],item['userid'][0],item['date'][0],item['title'][0],item['rbody'][0],item['helpful'],item['totalreview'],item['reviewid']])

next\_page=hxs.select('//a[@class="nav\_bar\_next\_prev"]/@href').extract()

for n in next\_page:

if n:

yield Request(urlparse.urljoin("http://www.flipkart.com",n[1:]),callback=self.par\_ur)

def stop\_reactor():

reactor.stop()

dispatcher.connect(stop\_reactor, signal=signals.spider\_closed)

spider = MySpiderd()

crawler = Crawler(Settings())

crawler.configure()

crawler.crawl(spider)

crawler.start()

log.start(loglevel=logging.DEBUG)

log.msg('Running reactor...')

reactor.run() # the script will block here until the spider is closed

log.msg('Reactor stopped.')

* **Python code for preprocessing the input data for noise removal**

**Preprocessmodule.py**

#!/usr/bin/env python

import re

from nltk.corpus import stopwords

from nltk.corpus import webtext

from nltk.collocations import BigramCollocationFinder

from nltk.metrics import BigramAssocMeasures

from nltk.tokenize import word\_tokenize

from replacers import RegexpReplacer

replacer = RegexpReplacer()

import csv

from nltk.corpus import wordnet

from nltk.tokenize.punkt import PunktSentenceTokenizer

from replacers import SpellingReplacer

from nltk.stem import PorterStemmer

stemmer = PorterStemmer()

from nltk.tokenize.punkt import PunktSentenceTokenizer

from nltk.corpus import stopwords

english\_stops = set(stopwords.words('english'))

def sentence(s):

a= PunktSentenceTokenizer().tokenize(s)

return a

def filterreview(s):

review='\n'.join(s)

review = review.lower()

#Convert www.\* or https?://\* to URL

review = re.sub('((www\.[\s]+)|(https?://[^\s]+))','URL',review)

#Convert @username to AT\_USER

review = re.sub('@[^\s]+','AT\_USER',review)

#Remove additional white spaces

review = re.sub('[\s]+', ' ', review)

#Replace #word with word

review = re.sub(r'#([^\s]+)', r'\1', review)

review=re.sub('\.',' ',review)

review=re.sub('\!','',review)

review=re.sub('\,','',review)

#trim

review = review.strip('\'"')

return review

def expansion(s):

q=replacer.replace(s)

return q

def words(s):

z= PunktSentenceTokenizer().tokenize(s)

for m in z:

b=word\_tokenize(m)

**APPENDIX 2**

**SNAPSHOTS**

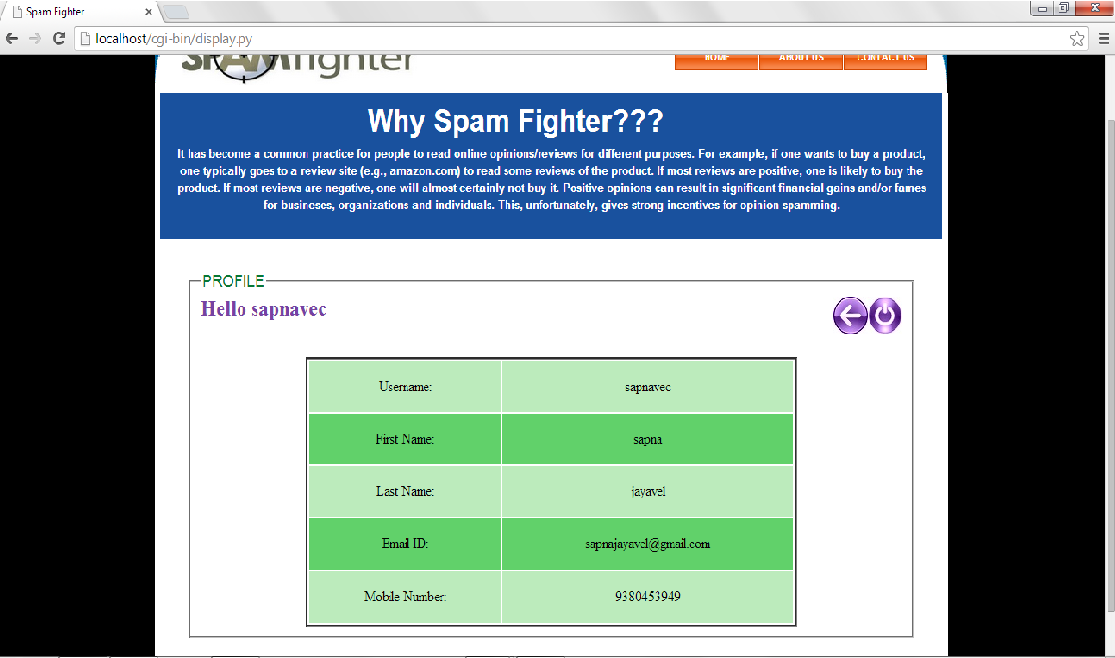
* **Homepage**

****

* **Profile page**

****

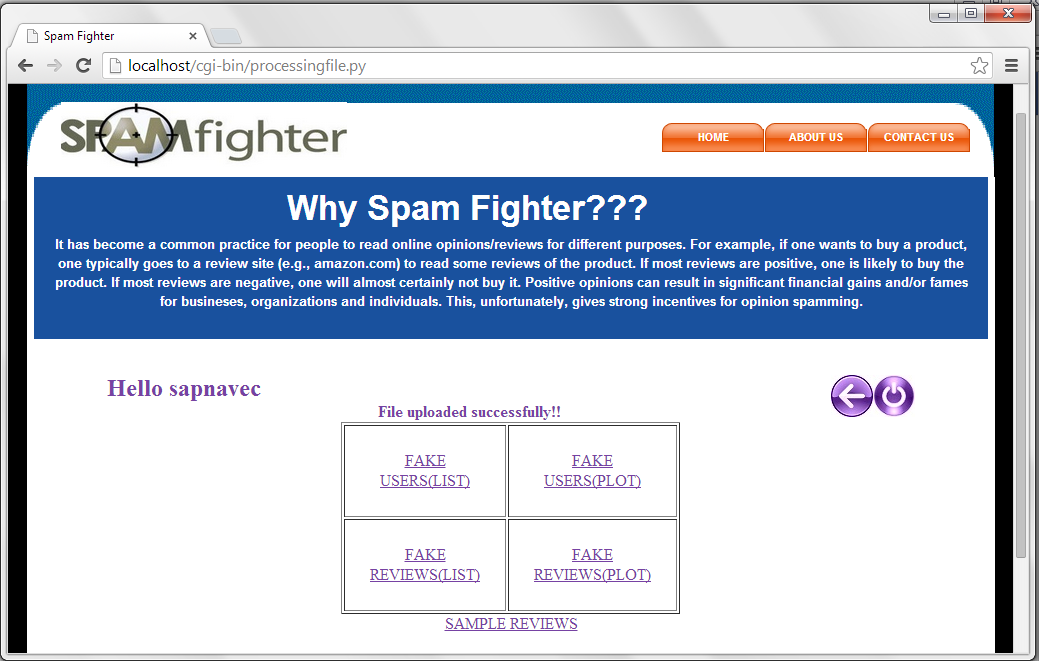
* **User details page**

****

* **Upload file**



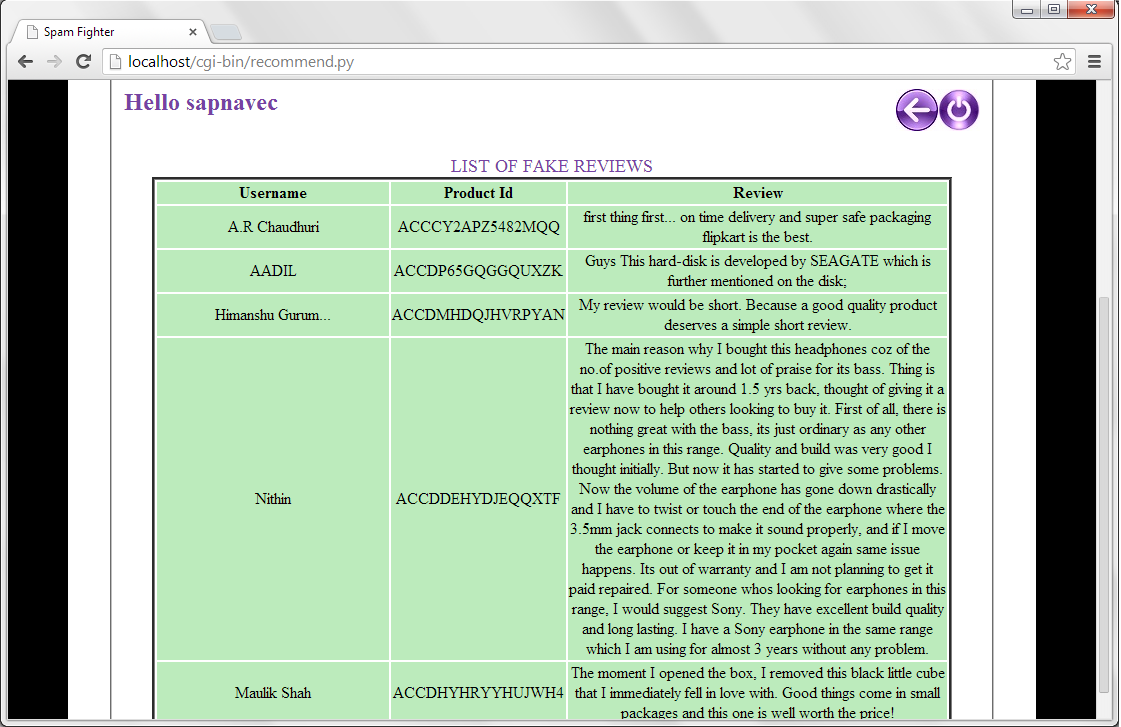
* **Processed file**

****

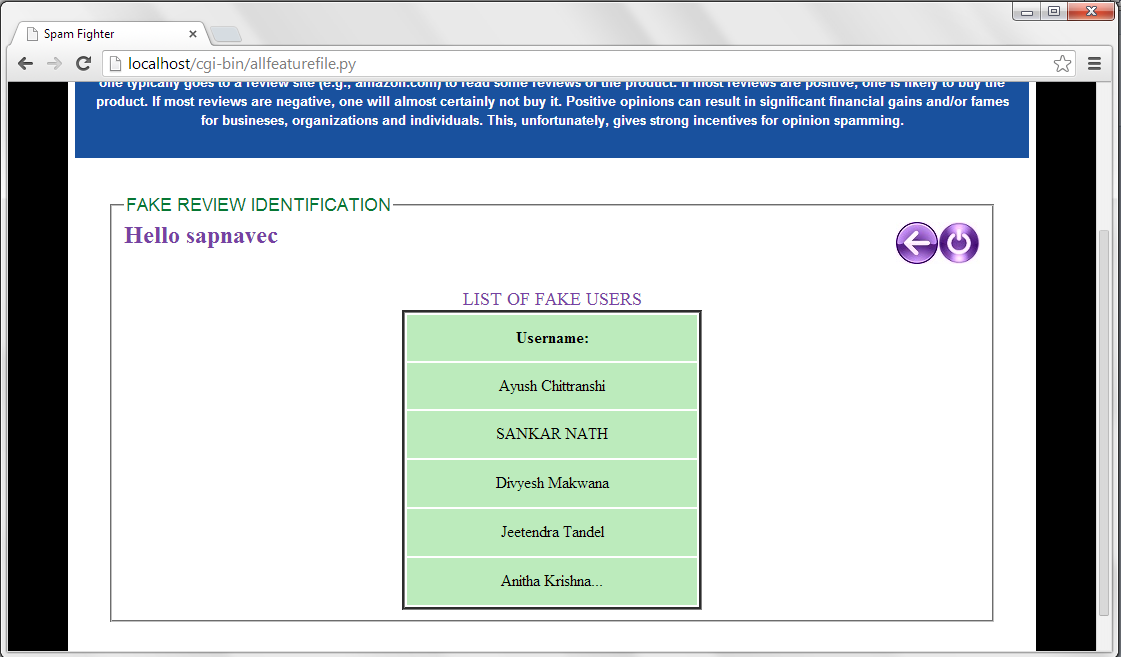
* **Upload reviews**

****

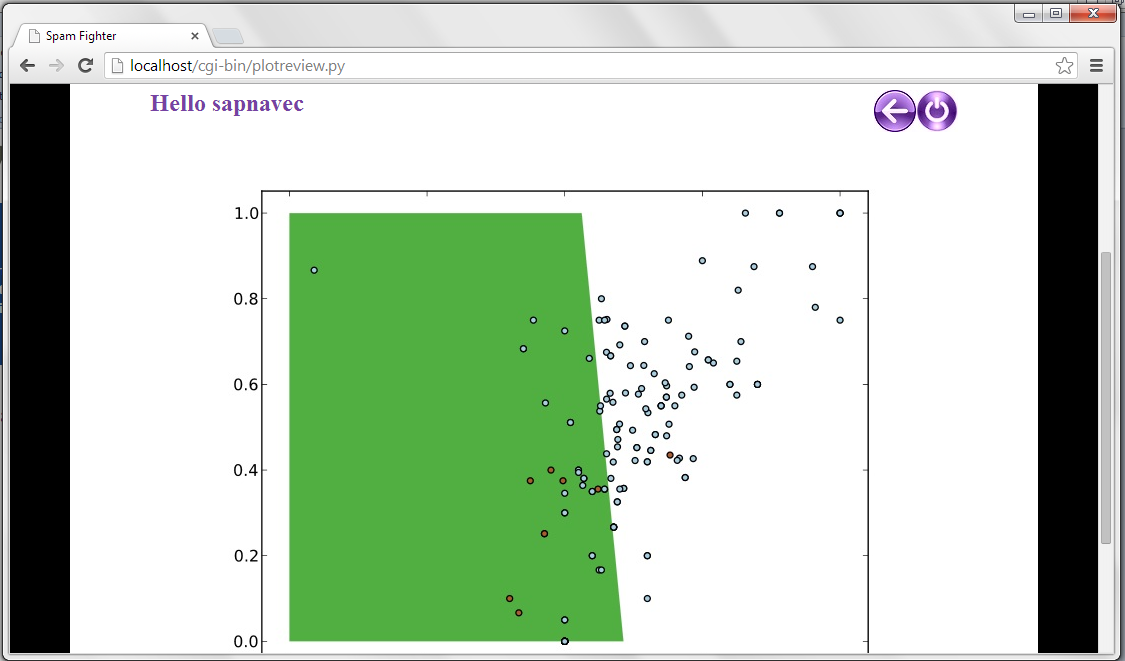
* **Fake Reviews**

****

* **Fake users**

****

* **Fake review plot**

****

**REFERENCES**

[1]Anindya Ghose and Panagiotis G. Ipeirotis, “Estimating the Helpfulness and Economic Impact of Product Reviews: Mining  
Text and Reviewer Characteristics,” in Proceedings of IEEE Transactions on Knowledge and Data Engineering, VOL. 23, NO. 10, OCTOBER 2012

[2] Fangtao Li, Minlie Huang, Yi Yang and Xiaoyan Zhu, “Learning to Identify Review Spam” in Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2008

[3] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery,” in *Proceedings of the twelfth international conference on World Wide Web - WWW ’03*, 2003

[4] N. Jindal, “Review spam detection,” *Proceedings of the 16th international conference on*, pp. 1189-1190, 2007

[5] R. Y. K. Lau, C. L. Lai, and Y. Li, “Leveraging the web context for context-sensitive opinion mining,” in *2009 2nd IEEE International Conference on Computer Science and Information Technology*, 2009

[6] W. Wang and Y. Zhou, “E-business Websites Evaluation Based on Opinion Mining,” in *2009 International Conference on Electronic Commerce and Business Intelligence*, 2009, pp. 87-90.

[7] Y.-Q. Xia, R.-F. Xu, K.-F. Wong, and F. Zheng, “The Unified Collocation Framework for Opinion Mining,” in *2007 International Conference on Machine Learning and Cybernetics*, 2007, pp. 844-850.

[8] “Latest spam statistics,” 2008. [Online]. Available: http://www.net-security.org/secworld.php?id=6056.

[9] K. Khan, B. B. Baharudin, a. Khan, and F. e-Malik, “Mining opinion from text documents: A survey,” *2009 3rd IEEE International Conference on Digital Ecosystems and Technologies*, pp. 217-222, Jun. 2009.

[10] M. Ramilli and M. Parandini, “Comment spam injection made easy,” in *Consumer Communications and Networking Conference*, 2009, pp. 1-5.

[11] N. Jindal, B. Liu, and S. M. Street, “Opinion Spam and Analysis,” 2008.

[12] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, “Finding Deceptive Opinion Spam by Any Stretch of the Imagination,” p. 11, Jul. 2011.

[13] R. Chandy and H. Gu, “Identifying spam in the iOS app store,” *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality - WebQuality ’12*, p. 56, 2012.

[14] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, “Detecting product review spammers using rating behaviors,” *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM ’10*, p. 939, 2010.

[15] N. Jindal, B. Liu, and E.-P. Lim, “Finding unusual review patterns using unexpected rules,” in *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM ’10*, 2010, p. 1549.

[16] Available at: www.cs.uic.edu/~liub/FBS/fake-reviews.html

Accessed on date 13.01.2014

### [17] Bing Liu, “Sentiment Analysis and Opinion Mining,” *Human Language Technologies,* Morgan & Claypool Publishers, May 2012

### [18]Toby Segaran, “Programming Collective Intelligence,” O'Reilly Media Inc., August 2007