

**Bachelor Thesis**  
**Electrical Engineering**  
**Thesis no: BEE**  
**December 2013**



## Facebook Blocket with Unsupervised Learning Filter

Mehmood ul Haq Minhas  
Khizer Amin

School of Engineering  
Blekinge Institute of Technology  
37179 Karlskrona  
Sweden

This thesis is submitted to School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical Engineering.

#### Contact Information

Authors:

1. Mehmood ul Haq Minhas  
e-Mail: rmh.minhas82@gmail.com
2. Khizer Amin  
e-Mail: khizer.amin83@gmail.com

Supervisor:

Raja M. Khurram Shahzad  
School of Computer Science and Communications,  
Blekinge Institute of Technology, Sweden  
E-mail: rks@bth.se

Examiner:

Sven Johansson  
School of Engineering,  
Blekinge Institute of Technology, Sweden  
E-mail: sven.johansson@bth.se

School of Engineering  
Blekinge Institute of Technology  
371 79 Karlskrona Sweden

Internet: [www.bth.se/ing](http://www.bth.se/ing)  
Phone: +46 455 385000  
Sweden

## Abstract

The Internet has become a valuable channel for both business-to-consumer and business-to-business e-commerce. It has changed the way for many companies to manage the business. Every day, more and more companies are making their presence on Internet. Web sites are launched for online shopping as web shops or on-line stores are a popular means of goods distribution. The number of items sold through the internet has sprung up significantly in the past few years. Moreover, it has become a choice for customers to do shopping at their ease. Thus, the aim of this thesis is to design and implement a consumer to consumer application for Facebook, which is one of the largest social networking website. The application allows Facebook users to use their regular profile (on Facebook) to buy and sell goods or services through Facebook. As we already mentioned, there are many web shops such as eBay, Amazon, and applications like blocket on Facebook. However, none of them is directly interacting with the Facebook users, and all of them are using their own platform. Users may use the web shop link from their Facebook profile and will be redirected to web shop. On the other hand, most of the applications in Facebook use notification method to introduce themselves or they push their application on the Facebook pages. This application provides an opportunity to Facebook users to interact directly with other users and use the Facebook platform as a selling/buying point. The application is developed by using a modular approach. Initially a Python web framework, i.e., Django is used and association rule learning is applied for the classification of users' advertisements. Apriori algorithm generates the rules, which are stored as a separate text file. The rule file is further used to classify advertisements and is updated regularly.



---

# Acknowledgments

---

First of all we thank Allah, the almighty, for granting us the strength and courage to complete our bachelor thesis. This project was carried out at the school of engineering, Blekinge Institute of Technology, Karlskrona, Sweden.

After that we would like to thank our supervisor Raja Muhammad Khurram Shahzad for his immense support during this thesis. Discussions with him has always been quite insightful and informative and helped us to re-organize ideas.

We would also like to give our sincere regards to Sven Johansson and Anders Hultgren for their support and help. Besides that we would like to say special thanks to our families for the encouragement and motivation they provided during the hard times.

Karlskrona 2013

Minhas Mehmood ul haq  
Khizer Amin



---

# Contents

---

<b>Acknowledgments</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	2
1.2 Motivation and Scope of the Thesis Work . . . . .	3
1.3 Thesis Overview . . . . .	3
<b>2 Technical Background</b>	<b>7</b>
2.1 Background . . . . .	7
2.2 State of Art Technologies related with Social Network . . . . .	8
2.2.1 Social Media and E-commerce . . . . .	8
2.3 Related Work in context of Unsupervised Filter . . . . .	14
2.3.1 Association Rules and Frequent Item sets . . . . .	14
<b>3 Web Technologies</b>	<b>17</b>
3.1 Programming Languages . . . . .	17
3.2 Data Base Management System . . . . .	18
3.3 Web Server Software . . . . .	20
3.4 Selected Technologies and Tools . . . . .	20
<b>4 Experiment and Results</b>	<b>23</b>
4.1 Facebook Application Implementation . . . . .	23
4.2 Web application on PythonAnywhere . . . . .	24

4.2.1	Project files Configuration . . . . .	26
4.2.2	Application Development and Logic . . . . .	26
4.3	Database . . . . .	26
4.3.1	Functions . . . . .	26
4.3.2	Mapping URLs to Views . . . . .	31
<b>5</b>	<b>Filter Implementation</b>	<b>33</b>
5.1	Implementation . . . . .	33
5.1.1	Rules Generation . . . . .	33
<b>6</b>	<b>Conclusions and Future Work</b>	<b>37</b>
	<b>Bibliography</b>	<b>39</b>
<b>7</b>	<b>Appendix</b>	<b>43</b>



# One

---

## Introduction

---

Current era is an era of Internet. E-business and web shops have changed the traditional way of buying and selling. Different models such business to business (B2B), business to consumer (B2C) and consumer to consumer (C2C) are followed. However, in first two models, i.e., B2B and B2C for online buying and selling, seller needs a platform for marketing to advertise their goods. On the other hand, in consumer to consumer business model, the link between the buyer and the seller can be helpful for buying and selling the things in local community. For the C2C model, Internet facilitates the direct marketing [1]. Social network platform can be an ideal place for such kind of activities, where people know each other, directly or indirectly and it is easy to approach the community of friends or friends of friends.

As important aspect of C2C is that sharing items for sale on social network and among community of friends can be approached without any cost. It is also a powerful way to drive traffic to your items. In the current web environment hundreds of websites are dedicated to online auctions. It is worth noting that to market the products and brand names many famous companies are using the social networks.

Our proposed application may be termed as Social Commerce that is similar to E- Commerce but with the involvement of social media. It includes collaborative e-commerce tools that enable a person to get



Figure 1.1: Consumer-to-Consumer<sup>2</sup>

advice from trusted individuals (friend list) on Facebook<sup>1</sup> to reach a fair bargain.

### 1.1 Problem Statement

The main objective of this project is to develop an e-commerce solution for social network users. This application makes it convenient to buy and sell product for the users from the users. Moreover, social networks provide a good marketing opportunity [2, 3]. Consumer to consumer is a business model where two individuals do business with each other directly. Like other buying and selling website, an intermediary/third party may be involved, which is application and platform provided. However, the purpose of the intermediary party is only to facilitate the transaction and provide a platform for the people to connect. The intermediary may receive a fee or commission on the sale, but is not liable for the products sold/exchanged. C2C normally takes the form of an auction where the bidding is done online. For example, eBay<sup>3</sup> and

---

<sup>1</sup><http://facebook.com>

<sup>2</sup>[http://www.mbaskool.com/images/stories/business\\_concepts/c2c.jpg](http://www.mbaskool.com/images/stories/business_concepts/c2c.jpg)

<sup>3</sup><http://ebay.com>

Amazon<sup>4</sup> are playing the role of intermediary. C2C reduces the cost due to direct interaction between persons and also eliminating the need of a physical store.

## 1.2 Motivation and Scope of the Thesis Work

The scope of this thesis is to develop a web store application in social media, i.e., Facebook. The users of social media are able to advertise product to sell or search the advertised product to buy. By using the unsupervised learning algorithm, authors will make sure that selling product followed the general ethical rule.

The main motivation of this thesis is to design and implement a C2C application for the social media, which give an opportunity to authors to learn about social network, application development for social media (Facebook), e-marketing and webpage development. We have used the Facebook which is the largest online social networking website. The role of application is to allow Facebook users to buy and sell goods or services through Facebook. There are different e-Commerce applications on Facebook, e.g., eBay and Amazon, however, none of them are directly interacting with the Facebook users, and all of them are using their own platform. Most of the applications in the Facebook, use notification method or they advertise their application on the Facebook page to introduce themselves in users. Our application uses the Facebook platform and network to spread the advertisement of a particular item from a particular user among the Facebook users and enables them for buy and selling.

## 1.3 Thesis Overview

This thesis report is split into different chapters. Each chapter will address specific aspect/s of the project. The summary of each chapter's content is as follows:

---

<sup>4</sup><http://amazon.com>

### **Chapter 1 - Introduction:**

This chapter gives an overview of the thesis. This also explains the aims and motivation of the project.

### **Chapter 2 - Technical Background**

This chapter covers social media and its effect on e-commerce and discussion or overview of different web store application's usability in Facebook. Furthermore, we also explain the difference between our applications and others web store applications.

### **Chapter 3 - Web Technologies**

This chapter covers the fundamental concepts of web programming, and introduces the technologies, which are used to develop web application. It also gives the detail description of Python, its web framework Django and database SQLite, which we selected for application development. In this chapter, we also discussed the concepts of Data Mining and classification.

### **Chapter 4 - Experiments and Results**

This chapter describes all the details of the engines which are included in the application along with detailed description of Facebook API's and its social plug-in, which we used in our application.

### **Chapter 5 - Filter Implementation**

In this chapter, we describe how we are handling the malicious advertisement by using APRIORI filter. In this stage, it implies that the database is connected online and all of the functionality will be ultimately tested and released.

## **Chapter 6 - Conclusions and Future Work**

This chapter concludes our thesis and present the future direction for extending the work.



# Two

---

## Technical Background

---

### 2.1 Background

With the growth and popularity of the social networking portals, the users (either individual or business) get the opportunity to create and maintain the network with friends, professionals, business colleagues, and alliances. Along with social interaction, social network provides them the business and professional opportunities. It is stated, "*The exponential growth of social media, from blogs, Facebook and Twitter to LinkedIn, offered organizations the chance to join a conversation with millions of customers around the globe every day*" [4].

In [2], the authors classified the types of social network sites and evaluate them in term of features and functionality. Moreover, in [5], authors discussed the opportunity of Internet marketing using social networking portals. Not only the networking and business opportunity of social networking sites was explored, also the researchers address the security issues of social networking [6].

The most popular online social networking website, Facebook gives the chance to an individual to advertise and market his product or service among the targeted group of people. Our application provides a platform for the users of Facebook to market their products or services. Moreover, it fulfill the community based buying and selling with the idea that don't discard anything might be someone else need it. It also

supports the concept of sustainability.

## 2.2 State of Art Technologies related with Social Network

Social Network Sites: A Social network site became a platform to build social networks or social relation among people who, for example shared interests, activities, backgrounds, or real-life connections [7].

### 2.2.1 Social Media and E-commerce

The usage of Internet has increased dramatically in the last decade, more and more companies found a new path of selling their products. Now these companies have explored new paths of business in the social media. The history behind the term Social media is that it is a group of Internet based applications that build on the ideological and technological foundations of Web 2.0, and allows the creation and exchange of user-generated content [8]. Web 2.0 is a platform for the social media.

Online social network sites, such as Facebook, MySpace<sup>1</sup>, and LinkedIn<sup>2</sup>, became worldwide communication tools that completely changed the communication paradigms.

### Facebook

Facebook was founded in 2004 by Mark Zuckerberg with his university friends. Facebook is an online social networking service. Its name comes from the conversational name of the book given to the students at the start of the academic year by some American university administrations to help students to know about each other. It was developed in PHP. In March 2013 Facebook had 1.15 billion active users. Discussed below are some interesting statistics of 2013 on Facebook [9]:

---

<sup>1</sup><https://myspace.com>

<sup>2</sup><http://www.linkedin.com/>



- Daily active users were 665 million on average for March 2013, an increase of 26% year-over-year.
- Monthly active users were 1.11 billion as of March 31, 2013, an increase of 23% year-over-year.
- Mobile Monthly active users were 751 million as of March 31, 2013, an increase of 54% year-over-year.

### **Facebook Applications**

Facebook application is one of the important feature that can be found on the Facebook website. More precisely, facebook applicaiton can be defined as "*An interactive software developed to utilize the core technologies of the Facebook platform to create an extensive social media framework for the app. Facebook Apps integrate Facebook's News Feed, Notifications, various social channels and other features to generate awareness and interest in the app by Facebook users*" [10]. To develop a Facebook application, developers' have to use Facebook Application programming interface (API's). API can be defined as "*A system of tools and resources in an operating system, enabling developers to create software applications*" [11]. Following are the names of some Facebook API's, which are commonly used to develop Facebook applications:

- Graph API
- FQL
- Open Graph
- Dialogs
- Chat
- Internationalization
- Ads

- Public Feed
- Keyword Insights

Our proposed application has used few of these APIs, which are discussed in detail, as follows:

### **Graph API**

According to [6], the Graph API is the primary way that data is retrieved from or posted to Facebook. The Graph API is a low-level HTTP-based API that can be used to query data, post new stories/posts, upload pictures and a variety of other tasks that an application might need to do. Facebook's open graph lets developer define new objects and actions in the social graph of people. Graph API facilitates the creation of new instances of actions and objects [6]. The social graph itself is a graph in the computer science domain, which consists of a series of nodes that are connect to each other. Understanding the differences between when developer needs a node and when developer needs to create a connection is an important distinction. The created connections let the developed application post advertisements to people's timeline, create posts with location tags or work with photos. Facebook always preserve the ID of object mapping. The Graph API is driven by HTTP requests. HTTP methods tend to map directly to actions on the graph. Some examples include "GET" for read, "POST" for modify and "DELETE" to remove the nodes.

It is important to know about an alternative Facebook API that also provides access to the social graph. The name of that API is FQL. FQL has functionality similar to the Graph API and provides a SQL-like interface. To use FQL, developer needs to know the basics of the Graph API, because FQL endpoints follow the Graph API. Thus, it is recommended to learn the use the Graph API, even if developer may prefer to use FQL.

### **Access token**

An access token is an opaque string, which is generated at the end of the authorization process. It represents a set of permissions that have been granted and can be used in the context of a particular application and for a particular person. Every (authenticated) request, that user make to the Graph API, will require passing along the access token. A few points about access tokens: They expire, so application has to manage their refreshing mechanism when time out occurs. Additionally, there is an access token that lets developer access a person's data. The developer may request an access token that operates on a page or an application. The page token is used to manage open graph data for a particular Facebook page. An application access token gives the application/developer access to application-specific data like application analytics. There are different types of access tokens to support different tasks [7], as follows:

- **User Access Token** - The user token is the most commonly used token. This token is needed when any application calls an API to read, modify or write a particular person's Facebook data on their behalf. User access tokens are generally obtained via a login dialog and require a person to permit the application to obtain one.
- **Application Access Token** - This kind of access token is required to modify and read the application settings. It can also be used to publish Open Graph actions. It is generated using a pre-agreed secret between the application and Facebook and is then used during calls that change application-wide settings. The developers may obtain an application access token via a server-to-server call.
- **Page Access Token** - These access tokens are similar to user access tokens, except that they provide permission to APIs that read, write or modify the data belonging to a Facebook Page. To obtain a page access token developer needs to obtain a user access token and ask for the manage pages permission. Once a developer has the user access token then he/she can get the page access token via the Graph API.

## 2. TECHNICAL BACKGROUND

---

- **Client Token** - The client token is an identifier that a developer can embed into native mobile binaries or desktop applications to identify a particular application. The client token is not meant to be a secret identifier because it is embedded in applications. The client token is used to access application-level APIs, but only a very limited subset. The client's token may be found in application's dashboard.

### **Token Generation**

User Tokens: Although each platform generates access tokens through different APIs, however they follow the given basic strategy to get a user token [7]. There are a lot of information that is available about the graph API and Access token on Facebook developer page.

### **Data Mining**

In this section the theoretical background of filter for the application is discussed. The size of data and information is getting larger day by day. Thus, a lot of research has been done in Data Mining to address the new challenges raised by the increasing amount of data. The purpose of Data mining is to analyze data stored in different forms and at different places such as data warehouses. An example of such data is business data in an organization. The business data may come from all parts of business, from the production to the sales, and management. The result of data analysis can be used to decide marketing strategies for products. It is worthy nothing that our analysis strategy proposed in the thesis may also be suitable for the market base data analysis.

The main goal of applying data mining [12] and machine learning [13] is to identify patterns in the dataset and use those patterns for the prediction. This process is being followed for centuries but the advent of modern day computer technologies has managed to manipulate huge amounts of data, increasing power of computing has developed certain methods and techniques such as Support Vector Machines [14]

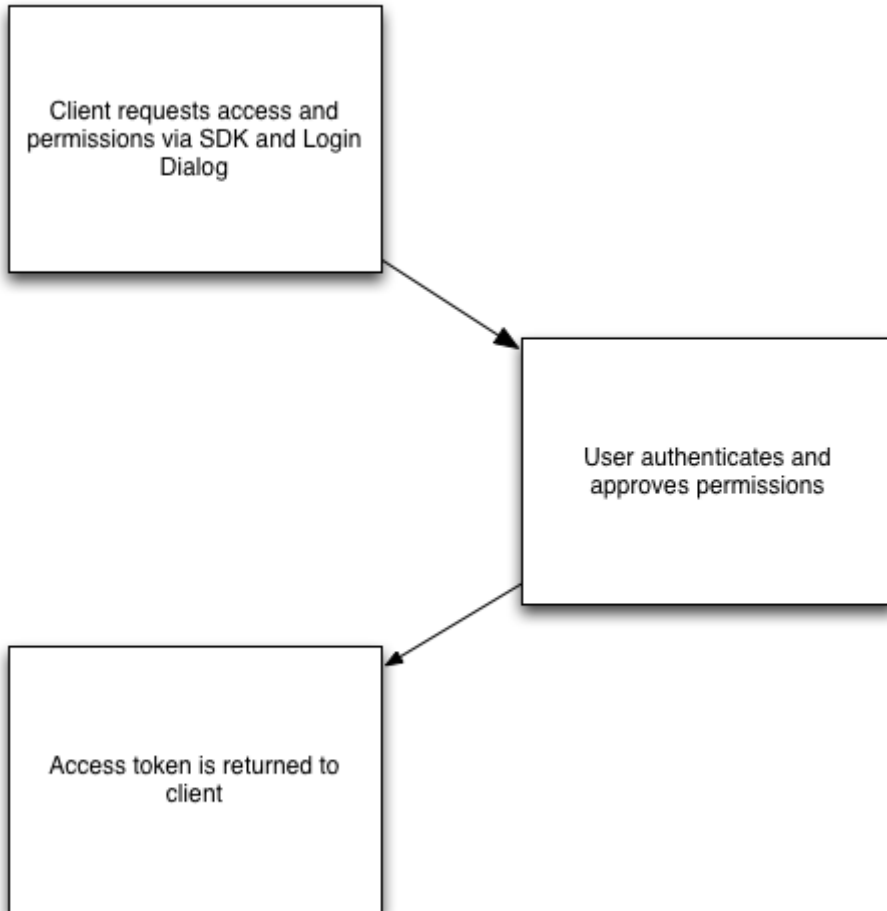


Figure 2.1: Generating Access Tokens

and Neural Networks [15]. to disclose hidden pattern in the large data sets.

### **Classification**

In Machine Learning, classification is a method to assign a class and categorize the new or unseen observations on the basis of a training data of observations [16]. There are many well-known algorithms to generate classifiers such as Support Vector Machines, Neural Networks K- Nearest Neighbors, etc [12]. The task of generated classifiers is to predict the future values based on the provided values. Filtering Spam emails from ham emails is an examples of this process in which a classifier may classify a given email into "*ham*" or "*spam*" classes. Another example is classification between malware and benign files [17,18,19,20,21].

### **Project Goals in context of Data Mining and Machine Learning**

The goal is to create a classification filter that can easily integrate itself into a Facebook application, which supports the users to post advertisements for their products to their friends and other users in order to sell or buy a product.

The user of the application is given a description box in order to enter some detail related to their ad post. The description is tokenized and classified by a dataset of rules in order to predict that web post is 'Approved' or 'Disapproved'. If the post is approved then that particular advertisement is published, and in case of disapproval the user will be informed by a message.

## **2.3 Related Work in context of Unsupervised Filter**

### **2.3.1 Association Rules and Frequent Item sets**

Since its inception in 1993, Association rule learning has been a highly acclaimed and well known research method for discovering interesting relations and patterns between variables in large databases [22]. This method is first introduced by Rakesh Agarwal for discovering patterns

and regularities in large scale databases in supermarkets [23] , for example , the rule onions ,potatoes => burger may be derived from a supermarket sales data, which would indicate that if a customer buys onions and potatoes together, that customer is likely to buy hamburger meat as well. This is also referred as market basket analysis, which indicate a retailer that customers usually purchase shampoo and conditioner together, so placing them both to promote at the same time would not create a significant increase in profit, while a promotion involving just one of the items would likely drive sales of the other.

Association rules learning is a process which require users to specify minimum support and confidence [22]. This minimum support will find the entire frequent item set in the database to further generate the rules from that database. There are many algorithms that do same job, some commonly proposed well known algorithms are, as follows:

### **Apriori algorithm**

Apriori<sup>3</sup> is a well-known algorithm for association rules mining, that requires minimum support and runs with a breadth first search strategy

### **Eclat algorithm**

Eclat is an algorithm that runs with a depth first search strategy, it analyzes the frequent item sets by observing the intersecting sets [24].

### **Related work**

Association rules analysis is an area in data mining that has acclaimed high attention of the research community, thus, there are many research papers which are published to explain the foundations and enhance it further. Some of this work is summarized, as follows:

---

<sup>3</sup>[http://en.wikipedia.org/wiki/Apriori\\_algorithm](http://en.wikipedia.org/wiki/Apriori_algorithm)

## 2. TECHNICAL BACKGROUND

---

1. Large database of customer transactions Reference [23], has used large databases of a big retailing company. They proposed an efficient algorithm for mining association rules and pruning large data items of a transactions list of a supermarket.
2. Comparisons of different association rules mining algorithms In [25], general behavior of Association rules are discussed. Authors have tested different algorithms strengths and weaknesses, and have carried out many runtime experiments. Authors concluded that almost all algorithms behave similarly if a market basket like data is provided to them.
3. Association analysis basic concepts and algorithms Reference is discuss association analysis on a market basket type transactional database [26]. Authors explain in details all the steps involve in rules mining and frequent item set generation. Authors present these concepts visually and with the help of tables. Furthermore, different types of efficient algorithms are also discussed in detail.



# Three

---

## Web Technologies

---

The backend to a website is pretty much everything the user cannot access. Generally, this means the programming that generates pages that the user views, or creating the "server-side" content of the site. To connect to the URL of a home page user need to send a request as a message across the Internet. If the system on the other side is working properly will respond the user request. The system on the other side, which responds the request, is known as Web servers and URL stands for Universal Resource Locator. A URL comes in three parts, i.e., `<method>://<host>/<absolutePathURL (apURL) >`.

### 3.1 Programming Languages

Choosing the right programming language for server-sides applications is an important issue and is not a simple job. Different languages have different capabilities and are suitable for different types of applications. Some languages are more commonly used in enterprise settings, while others are a staple of web applications.

#### C#

C# was developed by Microsoft<sup>1</sup> in 2000 as a fundamental part of its .NET framework. According to [9], C# ranks no.6 on the TIOBE Pro-

---

<sup>1</sup><http://microsoft.com>

gramming Community index<sup>2</sup> for September 2013. C# is an object-oriented, multi-paradigm programming language.

#### **PHP**

PHP (hypertext Processor) is a server-side scripting language that powers more than 240 million websites online. PHP is one of the younger languages on the list of languages. PHP is considered as easy to learn and can be readily embedded within HTML pages. PHP ranks no.2 on jobs Tractor<sup>3</sup>, and no.5 on the TIOBE index [9,27].

#### **Ruby**

Ruby was developed in 1995 by Yukihiro Matsumoto by combing elements from Perl and Lisp. Many students new to programming also find ruby comparatively easy to learn because of its simple syntax and English-like readability. Ruby is on TIBOE index rank no.11 and no.5 on jobs Tractor [9].

#### **Python**

Python is a dynamic programming language that is used in a wide variety of application domains. It was created in 1991 by Guido van Rossum. According to [27], Python ranks no.8 on the TIOBE index and on jobs Tractor its rank no.9.

## **3.2 Data Base Management System**

A database is a structured collection of data. It may be ranging from a simple shopping list to a picture gallery or the vast amounts of data in a corporate network. To determine, whether a website needs a database, the owner of the website need to answer different questions such as how frequently website is updated, how much contents are accepted from the

---

<sup>2</sup><http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

<sup>3</sup><http://jobstractor.com/>

visitors, what are contents, and many more. The data of the website will be stored on a computer either locally or remotely in a database.

To add, access, and process data stored in a computer database, you need a database management system such as Microsoft Access, MySQL<sup>4</sup>, SQLite<sup>5</sup>, and <sup>6</sup>. According to [28], Microsoft Access<sup>7</sup> is a database for limited purposes. According to [29], SQL stands for Structured Query Language, which is most common standardized language to access databases. Microsoft SQL is a database management system developed by Microsoft which uses SQL. MySQL is another example database management system. MySQL is a popular choice of database for web applications. It is world second most widely used open-source relational database management system. The official set of MySQL front-end tools, MySQL Workbench is actively built up by Oracle, and available free for use. Another popular database is SQLite. SQLite is an in-process library that implements a self-contained, server less, zero-configuration, transactional SQL database engine [30]. SQLite free for use for both commercial and private use. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite is different from most other SQL database. It is simple to use with following features:

- Simple and easy to use
- Simple to operate
- Simple to embed in a larger program
- Simple to maintain and customize

However, it is worth nothing that simplicity in a database engine can a weakness.

---

<sup>4</sup><http://mysql.com/>

<sup>5</sup><http://www.sqlite.org/>

<sup>6</sup>[www.postgresql.org/](http://www.postgresql.org/)

<sup>7</sup>[http://en.wikipedia.org/wiki/Microsoft\\_Access](http://en.wikipedia.org/wiki/Microsoft_Access)

## 3.3 Web Server Software

A web server is a program that serves contents upon the user request using the HTTP protocol [31]. Following are some well-known Web Server softwares.

### Internet Information Services (IIS)

According to [32], IIS<sup>8</sup> is a web server created by Microsoft for use with the Windows platforms. This webserver is flexible, secure and easy to manage for hosting.

### Apache http Server

Apache Http is an open source web server software. It has played a key role in the initial growth of the World Wide Web. It has been developed by an open source community-Apache software Foundation [33].

### Web hosting Service

A web hosting service is a Internet based service that allows individuals and organizations to make their website accessible via the Internet. Web hosts are companies that provide space on a server to deploy web applications/pages to an individual or organization.

## 3.4 Selected Technologies and Tools

### Python

We selected a python web development framework due to its advanced features. Google also uses python [34], which was one of the main motivations for authors to select this language as a tool. Python provides competitive advantages, which are, as follows:

---

<sup>8</sup><http://www.iis.net/>

- Multi-paradigms
- Scripting language
- Dynamic type system
- Large standard library
- Includes lists, dictionaries, sets, and much more as basic data types
- Automatic memory management
- Embedding into other languages
- Embeddable within applications as a scripting interface.
- Object oriented, imperative, functional
- Links to other dynamic languages
- A py2exe program that converts python scripts to executable for windows. Other platforms include it by default.
- Python is a web-friendly language
- Some implementations of python compile directly to machine code

#### **Django**

Django<sup>9</sup> is a prominent member of a new generation of web frameworks [35]. It is a free and open source web application framework, written in Python, that follows the model-view-controller architectural pattern. It is maintained by an independent organization, i.e., Django Software Foundation<sup>10</sup>. Django framework philosophies are, as follows:

- The more code, the more errors. This means that the amount of code should be minimized.

---

<sup>9</sup>[http://en.wikipedia.org/wiki/Django\\_\(web\\_framework\)](http://en.wikipedia.org/wiki/Django_(web_framework))

<sup>10</sup><https://www.djangoproject.com/foundation/>

### 3. WEB TECHNOLOGIES

---

- DRY principle, i.e., Don't repeat yourself tells that redundancy should be avoided. Each application's functionality should be only at one place, which not only reduces the amount of code, but it also contributes to the clarifications on the entire application.
- Explicit or in other words the framework should not make a great number of complicated tricks. Therefore, if a change in the core of the framework is needed, it can be done transparently.
- Model View Controller (MVC) is a way of developing software so that the code for defining and accessing data (the model) is separate from request routing logic (the controller), which in turn is separate from the user interface (the view) [36].

# Four

---

## Experiment and Results

---

### 4.1 Facebook Application Implementation

The Facebook platform allows developers to create web applications that integrate with Facebook's social network and are delivered via the Facebook web site [37].

To develop a Facebook application developers have to sign up and authenticate a developer account. After signing, the developer may start to set up a new application on the developer page, i.e., <https://developers.facebook.com/applications>, see Figure 4.1. We

The screenshot shows the 'Create New App' interface on the Facebook developer platform. It features three main input fields: 'App Name' with the value 'GoodsSale' and a 'Valid' status indicator; 'App Namespace' with the value 'goodsbook' and an 'Available' status indicator; and 'App Category' with a dropdown menu set to 'Shopping' and a 'Choose a sub-category' dropdown. At the bottom, there is a 'Continue' button and a 'Cancel' button. A footer note states: 'By proceeding, you agree to the Facebook Platform Policies'.

Figure 4.1: Facebook Application Implementation

followed the procedure mentioned above and after receiving confirmation, we were forwarded to editing the application parameters. In this

field we were asked for the canvas URL that is a content pulled from this base URL to iframe on Facebook and Secure Canvas URL Content pulled from the secure base URL for users on HTTPS (required from October 1st, 2013), see Figure [4.2]. After October 2013, SSL is also a Facebook requirement. We used pythonanywhere<sup>1</sup> web hosting service to deploy our application because from 1st Oct Facebook application cannot be deployed on the local host unless developers have SSL certificate. The website "*pythonanywhere*" provided us in-browser access to the server-based Python and Bash Command-line interface, along with a code editor [31]. This page contains an Application ID and Application



The screenshot shows the 'App on Facebook' configuration interface. It includes a green checkmark icon and the title 'App on Facebook'. The settings are as follows:

Canvas Page: [?]	http://apps.facebook.com/goodsbook
Canvas URL: [?]	<input type="text" value="http://rajamehmood.pythonanywhere.com/"/>
Secure Canvas URL: [?]	<input type="text" value="https://rajamehmood.pythonanywhere.com/"/>
Canvas Width: [?]	<input checked="" type="radio"/> Fluid <input type="radio"/> Fixed (760px)
Canvas Height: [?]	<input checked="" type="radio"/> Fluid <input type="radio"/> Fixed at <input type="text" value="800"/> px
Unity Integration Enabled:	<input checked="" type="radio"/> Disabled <input type="radio"/> Enabled

Figure 4.2: Facebook Application Implementation-I

Secret which is used in our web application.

## 4.2 Web application on PythonAnywhere

To start the web application and login in to PythonAnywhere, we followed steps, as follows:

- Go to the "Web" tab.
- Select the "Add a new web application" option on the left-hand side.

---

<sup>1</sup><https://www.pythonanywhere.com/>



- If you have a Web Developer account, specify the domain you want your web application to appear on, then click "Next"
- Select the "Manual configuration" option from the list.
- Click "Next", and wait for the system to tell you that the web application has been created.

### Requirements

To develop our Facebook application, Django (version 1.3.7) and Facebook SDK (django-facebook) was required. To install Django 1.3.7 and django-facebook, virtual environment was setup. A virtual environment provided us a private Python environment. To install required packages we installed pip [35], which was a package management system. This package management system was used to install and manage software packages written in python [38]. It is worth noting that in pythonanywhere pip is already installed. After that, we installed Django 1.3.7 and django-facebook SDK. After installing our required packages we started our project and a new web application. Initially, Django created following given files to begin with:

*\_\_init.py*

*manage.py*

*settings.py*

*urls.py*

Later we executed following given command: *python manage.py startapplication myapplication*

We got following files after executing above-mentioned command:  
*\_\_init\_\_.py*

*models.py*

*tests.py*

*views.py*

### 4.2.1 Project files Configuration

To configure the django-Facebook SDK in our project we added some additional information in settings.py file about database and we also added our application and Facebook SDK in installed applications. In MIDDLEWARE\_CLASSES, we disabled the CsrfViewMiddleware, and added the FacebookMiddleware. FacebookMiddleware took care of adding different attribute, e.g., request.facebook.graph, which we was used to access the Facebook Graph API. At the end of settings.py file we included information about our Facebook application, i.e., Facebook application ID and its SECRET KEY. After giving all these information, we created and synchronized our database with django-admin.

### 4.2.2 Application Development and Logic

Views.py was one of the files, which Django generated for us when we ran the startup command as described above. A view function, or view is a simple Python function that takes a Web request and returns a Web response. This response can be HTML contents of a web page, or redirect, or a 404 error, or anything else. In our application file views.py we imported different classes, which we used in our project. We wrote different functions to handle application process. We also used views.py (instead of models.py) to manage the application database. We created a database manually using Firefox SQLite manager.

## 4.3 Database

### 4.3.1 Functions

We wrote @canvas\_only in views.py before first function to make sure that Facebook redirects the user to a screen where the users authorize

Columns (9)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	ID	INTEGER	1	null	1
1	USERID	INT	1	null	0
2	TITLE	TEXT	1	null	0
3	DESCRIPTION	TEXT	1	null	0
4	TAG	TEXT	0	null	0
5	CATEGORYID	INT	0	null	0
6	TYPE	INT	0	null	0
7	CITY	TEXT	0	null	0
8	COUNTRY	TEXT	0	null	0

Figure 4.3: ClassifiedPost

Columns (6)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	ID	TEXT	1	null	1
1	NAME	TEXT	1	null	0
2	COUNTRY	TEXT	1	null	0
3	CITY	TEXT	0	null	0
4	RANK	INT	0	null	0
5	Accesstoken	text	0	null	0

Figure 4.4: User

Columns (3)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	ID	INTEGER	1	null	1
1	texttag	TEXT	1	null	0
2	Post_ID	INTEGER	1	null	0

Figure 4.5: Tags

permission for the application, if application is loaded first time.

### Def home (request):

Our home function's first line was `my = request.Facebook.graph.get_` object ('me'), which we used to get access the Facebook graph API. The data can be used from Facebook with the help of graph API. This provides the information about user which can be friends information, relationships, wall posts, friendships, and etc. Later, we collected and stored user information, i.e., his name, location, number of friends in our

## 4. EXPERIMENT AND RESULTS

---

database. It is worth nothing that we were not only generating access\_token to use in the current session, if a user creates an ad to post, but we stored the access\_token in our database as soon as the user loaded our application. Later, we used this access\_token in another function to post relevant ad on his Facebook wall page. Same function was used to present top ten ads from our database to the current user according to his location, i.e., city if it exists in the database (i.e., given in Facebook information). If no ad already exists for that particular city, a message appeared that application don't have ads for this city, thus to view all ads posted in the application click home button.

### Def index (request):

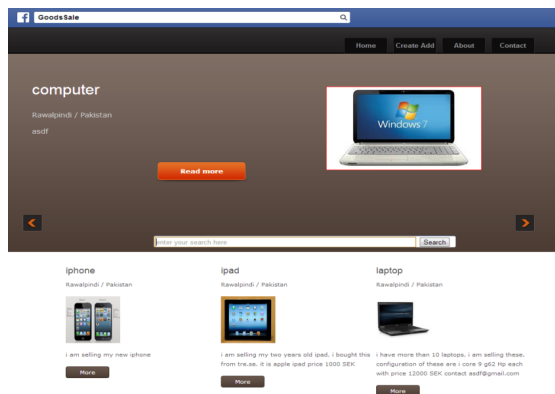


Figure 4.6: Index Function of Application

Behind the home button, we added another function named as index, which was almost similar to the home function but without oath token and without storing user's information in the database as we have already done this. This function was to show all ads from database.

### Def add\_new (request):

In this function we used user session token and if the user location exists than allow him to load "Create Ad" page, otherwise load another page,

Figure 4.7: Add\_New Function of Application

which contains an error message and ask for the permission to redirect the user to the Facebook page to add his location. This information was mandatory, if a user wants to create the ad.

#### **Def update\_status (request):**

This function was used to collect the ad information, which a user created to post. The information was collected on a html page from user and saved in our database. We also stored the picture given with the ad, if the user selected an imaged during the ad creation instead of a default picture from the database. In the same function we used filters to check the ad description for permitting or denying an ad. If filter approved the ad than we showed a success message to user that "your ad is posted" otherwise an error message is displayed with a message that "your ad cannot posted" please try again.

#### **Def search\_add (request):**

In this function we implemented keyword search. If the user search is not successful then application displayed an error message "Could not find" with home.html page instead of redirecting to new html page with

## 4. EXPERIMENT AND RESULTS

---

a message. If users search returned result successfully, then we load home.html with all ads containing keyword.

### Def more\_detail (request):



Figure 4.8: More\_Detail Function of Application

This function was used to display the advertisement's details, i.e., title and description. When a user clicked on the button, a new web page was loaded, which contained all detail of that particular ad. We allowed users to contact with seller in two ways, i.e., user can add comments on the specific ad and secondly, the user can send private message to seller by clicking on "Click here to contact seller" field.

### Def contact (request):

This function is responsible to redirect user to the new page which contain information to contact application developers.

### Def about (request):

This function is used to redirect user to a new page, which contains introduction of goodsbook application and about the developer team.

### 4.3.2 Mapping URLs to Views

Each view function returns an HTML page. To invoke a function application needed an URL, which tells the Django about the function to be invoked. "URLconf" is like a table of contents for Django application. Basically, it is mapping between URL patterns and the view functions that should call for those URL patterns.





# *Five*

---

## Filter Implementation

---

### **5.1 Implementation**

We used an algorithm for normal and abusive words proposed by [39]. We have modified the algorithm as per requirements and named it as '*Ads Filter Algorithm*'. The original algorithm was proposed for malware and benign classification. However, we used this algorithm for predicting about the type of advertisement, i.e., allowed or not allowed. The algorithm basically takes words (both abusive and normal) generate rules for them. Later, algorithm applies the rules over the tokenized stream of advertisement contents and returns the decision. It also updates the rules automatically.

#### **5.1.1 Rules Generation**

For the rule generation algorithm, please refer Algorithm 1.

## 5. FILTER IMPLEMENTATION

---

---

### Algorithm 1 Rule Generation

---

INPUT: A Collection of Files

Output: Normal words and Abusive words database

Read an abusive word file;

Run association rules analysis on the file with no support at all;

Output the generated rules to create the abusive words dataset;

Read a normal file;

Run association rules analysis on the file with no support;

Output the generated rules to create the normal words dataset;

**for** each file **do**

    Read the file;

    Run association rules analysis on the file with no support;

    Output the generated rules;

    Search the abusive database for the generated rules;

    Search the normal words for the generated rules;

**if** True Positive or True Negative **then**

        Goto next file;

**if** False Positive **then**

**if** Subject File is abusive **then**

            Remove the matching words from the abusive words database;

**end if**

**else**

**if** Subject File is a normal File **then**

            Remove the matching words from the normal words database;

**end if**

**end if**

**end if**

**if** False Negative **then**

**if** Subject File is a abusive **then**

        Add the new words to the abusive words database;

**else**

**if** Subject File is a normal File **then**

            Add the new words to the normal words database;

**end if**

**end if**

**end if**

<sup>34</sup>**end for**

---

Your Location : Stockholm, Sweden

Title:

Discription:

Tag:

Select product image:  No file chosen

Copyright © 2013 GoodsBook by [www.Goodsbook.com](http://www.Goodsbook.com)

Figure 5.1: Filter Output-I

Your Add Status

Approved

.....

Copyright © 2013 GoodsBook by [www.Goodsbook.com](http://www.Goodsbook.com)

Figure 5.2: Filter Output-II

As it can be seen in the Figure 5.1 a user types an advertisement in the description box and provides the other required information. The Figure 5.2 is the result of the previous one ,the advertisement is posted by the user which is approved by the application.

In the Figure 5.3 the user tries to post an advertisement that contain contents that are graphic in nature, the application disapproves this

## 5. FILTER IMPLEMENTATION

---

Your Location : Stockholm, Sweden

Title:

Discription:

Tag:

Select product image:  No file chosen

Copyright © 2013 GoodsBook by [www.Goodsbook.com](http://www.Goodsbook.com)

Figure 5.3: Filter Output-III

advertisement and notify the user to try.

# Six

---

## Conclusions and Future Work

---

This thesis presents a solution for a Facebook consumer to sale or purchase items from other consumers by using a retailing application. The application uses association rule to classify the advertisements as approved or disapproved. Facebook users can use this application as sellers and can create an advertisement along-with the description and picture of the product. The advertisement will be posted Facebook wall of the user and wall of its targeted customers, such as friends or people in the same locality. The buyer can send a private message to the seller and also can comment on the advertisement. The new buyer can also use keywords to search specific products. This thesis can be enhanced in the future in different direction, some possible future pointers are, as follows:

- **Authentication and authorization** Authentication is a procedure of informing the application about the user. This can be enhanced in a different ways such as:
  - By introducing the user name and password to sign in the application.
  - By sending SMS verification code to secure the sign in process.
- **Paypal** For secure monetary transaction, Paypal<sup>1</sup> can be integrated.

---

<sup>1</sup><http://paypal.com>

## 6. CONCLUSIONS AND FUTURE WORK

---

- **Google Map** To show seller's location Google map API can be integrated.
- **Auctions** Auction or Bidding system can also added for the user to make a bid on product.
- **Advertisement Editing** An advertisement editing function can be added, which allows a seller to edit his advertisement or delete his advertisement, and update new edited advertisement on Facebook wall page.
- **Seller Rank** In our application we are already ranking the seller by collecting his data from Facebook. However, this can be improved by allowing buyers to allocate a rank either positive or negative to the seller.
- **Filters** Currently, application is using Association rules based filter to check the description of an advertisement created by the seller. However, more filters can also be added to screen other contents of the advertisement such as image filter to check the picture selected by seller for his product at the time of creation of the ad. Moreover, a filter to avoid duplication of advertisement can also be added.

---

# Bibliography

---

- [1] B. Stone and R. Jacobs. *Successful Direct Marketing Methods*. McGraw-Hill Education, 2008.
- [2] D. Zarrella and A. Zarrella. *The Facebook Marketing Book*. O'Reilly Media, 2010.
- [3] J. Reynolds. *The Complete E-Commerce Book: Design, Build & Maintain a Successful Web-based Business*. Taylor & Francis, 2004.
- [4] The new conversation: taking social media from talk to action.
- [5] D. Chaffey, F. Ellis-Chadwick, R. Mayer, and K. Johnston. *Internet Marketing: Strategy, Implementation and Practice*. Financial Times. Financial Times Prentice Hall, 2009.
- [6] O. Nawaz, C. Gehrman, and M. Fiedler. Secure mobile social networks using usim in a closed environment. In *Internet Technology And Secured Transactions, 2012 International Conferece For*, pages 439–446, 2012.
- [7] Harsha Gangadharbatla. Facebook me: Collective self-esteem, need to belong, and internet self-efficacy as predictors of the igeneration’s attitudes toward social networking sites. *Journal of interactive advertising*, 8(2):5–15, 2008.
- [8] Access tokens - facebook developers. <https://developers.facebook.com/docs/facebook-login/access-tokens/>.

- [9] TIOBE software: Tiobe index. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
- [10] J. Feiler. *How to Do Everything: Facebook Applications*. How to do everything. McGraw-Hill Education, 2008.
- [11] Keywords scott and john chapter 5. [http://wiki.ggc.edu/wiki/Keywords\\_Scott\\_and\\_John\\_Chapter\\_5](http://wiki.ggc.edu/wiki/Keywords_Scott_and_John_Chapter_5).
- [12] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 2011.
- [13] Y. Kodratoff, R.S. Michalski, R.S. Michalski, J.G. Carbonell, and T.M. Mitchell. *Machine Learning: An Artificial Intelligence Approach*. Number v. 3 in Machine Learning. Morgan Kaufmann, 1990.
- [14] I. Steinwart and A. Christmann. *Support Vector Machines*. Information science and statistics. Springer, 2008.
- [15] K. Gurney. *An Introduction to Neural Networks*. Taylor & Francis, 2003.
- [16] S. Marsland. *Machine Learning: An Algorithmic Perspective*. Taylor & Francis, 2011.
- [17] Raja Khurram Shahzad, Syed Imran Haider, and Niklas Lavesson. Detection of spyware by mining executable files. In *International Conference on Availability, Reliability, and Security (ARES'10)*, pages 295–302. IEEE, 2010.
- [18] Raja Khurram Shahzad and Niklas Lavesson. Detecting scareware by mining variable length instruction sequences. In *Information Security South Africa (ISSA'11)*, pages 1–8. IEEE, 2011.
- [19] Raja Khurram Shahzad, Niklas Lavesson, and Henric Johnson. Accurate adware detection using opcode sequence extraction. In *Sixth International Conference on Availability, Reliability and Security (ARES'11)*, pages 189–195. IEEE, 2011.



- [20] Raja Khurram Shahzad and Niklas Lavesson. Veto-based malware detection. In *Seventh International Conference on Availability, Reliability and Security (ARES'12)*, pages 47–54. IEEE, 2012.
- [21] Raja Khurram Shahzad and Niklas Lavesson. Comparative analysis of voting schemes for ensemble-based malware detection. *Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 4, 2013.
- [22] C. Zhang and S. Zhang. *Association Rule Mining: Models and Algorithms*. Lecture Notes in Artificial Intelligence. Springer, 2002.
- [23] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993. <http://doi.acm.org/10.1145/170036.170072>.
- [24] O.Z. Maimon and L. Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer, 2006.
- [25] Jochen Hipp, Ulrich Guntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison, 2000.
- [26] P.N. Tan. *Introduction to Data Mining*. Pearson Education, Limited, 2013.
- [27] Best programming languages to learn in 2013: The elite eight. <https://www.udemy.com/blog/best-programming-language/>.
- [28] Home : The official microsoft IIS site. <http://www.iis.net/>.
- [29] Business intelligence | database management | data warehousing | microsoft SQL server. <http://www.microsoft.com/en-us/sqlserver/default.aspx>.
- [30] SQLite home page. <http://www.sqlite.org/>.

## BIBLIOGRAPHY

---

- [31] Category:Web server software - wikipedia, the free encyclopedia. [http://en.wikipedia.org/wiki/Category:Web\\_server\\_software](http://en.wikipedia.org/wiki/Category:Web_server_software).
  
- [32] Internet information services - wikipedia, the free encyclopedia. [http://en.wikipedia.org/wiki/Internet\\_Information\\_Services](http://en.wikipedia.org/wiki/Internet_Information_Services).
  
- [33] B. Laurie and P. Laurie. *Apache: The Definitive Guide*. O'Reilly Media, 2002.
  
- [34] Quotes about python. <http://www.python.org/about/quotes/>.
  
- [35] A. Holovaty and J. Kaplan-Moss. *The Definitive Guide to Django: Web Development Done Right*. Apresspod Series. Apress, 2009.
  
- [36] S. M. Niaz Arifin. A spatial agent-based model and a multi-dimensional data warehouse for malaria research.
  
- [37] ETS Hot Team. 7 things you need to know about facebook applications. [http://ets.tlt.psu.edu/wp-content/uploads/facebook\\_applications.pdf](http://ets.tlt.psu.edu/wp-content/uploads/facebook_applications.pdf).
  
- [38] Guide to setting up a virtualenv for pythonanywhere web apps. <https://www.pythonanywhere.com/wiki/VirtualEnvForNewerDjango>.
  
- [39] Muazzam Ahmed Siddiqui. *Data Mining Methods for Malware Detection*. University of Central Florida, 2008.

# Seven

---

## Appendix

---

Appendix A: settings.py

```
\# Django settings for mysite project.

DEBUG = True
$TEMPLATE\_DEBUG = DEBUG$

ADMINS = (
    $\# ('Your Name', 'your_email@example.com'),
)$

MANAGERS = ADMINS

DATABASES = {
    \='default': {
        $\'ENGINE': 'django.db.backends.sqlite3', \# Add '
        postgresql_psycopg2', $ 'postgresql', 'mysql', '
        sqlite3' or 'oracle'.
        \\'NAME': '/home/moodiraja/mysite/db.sqlite',
            \# Or path to database
            file if using sqlite3.
        \='USER': '', \# Not used
            with sqlite3.
        \='PASSWORD': '', \# Not used
            with sqlite3.
```

## 7. APPENDIX

---

```
\='HOST': '',                                \# Set to
    empty string for localhost. Not used with
    sqlite3.
'PORT': '',                                    \# Set to empty
    string for default. Not used with sqlite3.
}
}

\# Hosts/domain names that are valid for this site;
    required if DEBUG is False
\# See https://docs.djangoproject.com/en/1.3/ref/settings/#allowed-hosts
ALLOWED_HOSTS = []$

\# Local time zone for this installation. Choices can be
    found here:
\# http://en.wikipedia.org/wiki/List\_of\_tz\_zones\_by\_name$
\# although not all choices may be available on all
    operating systems.
\# On Unix systems, a value of None will cause Django to
    use the same
\# timezone as the operating system.
\# If running in a Windows environment this must be set to
    the same as your
\# system time zone.
TIME_ZONE = 'America/Chicago'$

\# Language code for this installation. All choices can be
    found here:
\# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = 'en-us'$
SITE_ID = 1$
\# If you set this to False, Django will make some
    optimizations so as not
\# to load the internationalization machinery.
USE_I18N = True$
```

---

```
\# If you set this to False, Django will not format dates,
    numbers and
\# calendars according to the current locale
$USE_L10N = True$

\# Absolute filesystem path to the directory that will
    hold user-uploaded files.
\# Example: "/home/media/media.lawrence.com/media/"
$MEDIA_ROOT = '/home/moodiraja/mysite/media'$

$\# URL that handles the media served from MEDIA_ROOT.
    Make sure to use a$
\# trailing slash.
\# Examples: "http://media.lawrence.com/media/", "http://
    example.com/media/"
$MEDIA_URL = '/media/'$

\# Absolute path to the directory static files should be
    collected to.
\# Don't put anything in this directory yourself; store
    your static files
$\# in applications' "static/" subdirectories and in
    STATICFILES_DIRS.$
\# Example: "/home/media/media.lawrence.com/static/"
$STATIC_ROOT = '/home/moodiraja/mysite/static'$

\# URL prefix for static files.
\# Example: "http://media.lawrence.com/static/"
$STATIC_URL = '/static/'$

\# URL prefix for admin static files — CSS, JavaScript
    and images.
\# Make sure to use a trailing slash.
\# Examples: "http://foo.com/static/admin/", "/static/
    admin/".
$ADMIN_MEDIA_PREFIX = '/static/admin/'$

\# Additional locations of static files
$STATICFILES_DIRS = ($
```

## 7. APPENDIX

---

```
\# Put strings here, like "/home/html/static" or "C:/
www/django/static".
\# Always use forward slashes, even on Windows.
\# Don't forget to use absolute paths, not relative
paths.
)

\# List of finder classes that know how to find static
files in
\# various locations.
$STATICFILES\_FINDERS = ($
    \='django.contrib.staticfiles.finders.
        FileSystemFinder',
    \='django.contrib.staticfiles.finders.
        ApplicationDirectoriesFinder',
\#     'django.contrib.staticfiles.finders.
        DefaultStorageFinder',
)

\# Make this unique, and don't share it with anybody.
$SECRET\_KEY = '5-vaoiw7cthe)hzo=7qs2=p@27*_ddb dh-(h4nmfv
(48_5*kj'$

\# List of callables that know how to import templates
from various sources.
$TEMPLATE_LOADERS = ($
    \='django.template.loaders.filesystem.Loader',
    $     'django.template.loaders.application_directories.
        Loader', $
\=\#     'django.template.loaders.eggs.Loader',
)

$MIDDLEWARE\_CLASSES = ($
    \='django.middleware.common.CommonMiddleware',
    \='django.contrib.sessions.middleware.
        SessionMiddleware',
    \='django.middleware.csrf.CsrfViewMiddleware',
    \='django.contrib.auth.middleware.
        AuthenticationMiddleware',
```

---

```

        \='django.contrib.messages.middleware.
            MessageMiddleware',
    $    'django_facebook.middleware.FacebookMiddleware', $
)

$ROOT\_URLCONF = 'mysite.urls'$

$TEMPLATE_DIRS = ($
    $# Put strings here, like "/home/html/
        django_templates" or "C:/www/django$templates".
    \=# Always use forward slashes, even on Windows.
    \=# Don't forget to use absolute paths, not relative
        paths.
)

$INSTALLED\_APPLICATIONS = ($
    \='django.contrib.auth',
    \='django.contrib.contenttypes',
    \='django.contrib.sessions',
    \='django.contrib.sites',
    \='django.contrib.messages',
    \='django.contrib.staticfiles',
    \=# Uncomment the next line to enable the admin:
    \=# 'django.contrib.admin',
    \=# Uncomment the next line to enable admin
        documentation:
    \=# 'django.contrib.admindocs',
    $'django\_facebook', $
    \='mysite.myapplication',
)

\# A sample logging configuration. The only tangible
    logging
\# performed by this configuration is to send an email to
\# the site admins on every HTTP 500 error.
\# See http://docs.djangoproject.com/en/dev/topics/logging
    for
\# more details on how to customize your logging
    configuration.

```

## 7. APPENDIX

---

```
$LOGGING = ${
    \='version': 1,
    $    'disable_existing_loggers': False,$
    \='handlers': {
    $        'mail_admins': ${
                \='level': 'ERROR',
                \='class': 'django.utils.log.
                    AdminEmailHandler'
            }
        },
    \='loggers': {
        \='django.request': {
    $            'handlers': ['mail_admins'],$
            \='level': 'ERROR',
            \='propagate': True,
        },
    }
}
$FACEBOOK\_APPLICATION\_ID = '249588381832401'$
$FACEBOOK\_SECRET\_KEY = 'eed8cf68caadf038b89fe0fd445cbab7
'$

$FACEBOOK\_CANVAS\_PAGE =$ 'https://applications.facebook.
    com/%s/' %

$FACEBOOK\_APPLICATION\_ID$

$FACEBOOK\_SCOPE = ['publish\_stream']$
```

Appendix B: manage.py

```
\#!/usr/bin/env python
$from django.core.management import execute_manager$
import imp
try:
    $imp.find_module('settings') \# Assumed to be in the
        same directory.$
except ImportError:
    \=import sys
```



---

```
\=sys.stderr.write("Error: Can't find the file '
    settings.py' in the directory containing %r. It
    applicationears you've customized things.\nYou'll
    have to run django-admin.py, passing it your
    settings module.\n" % __file__)
sys.exit(1)
```

```
import settings
```

```
$if \_name\_== "\_main\_":
    \= execute\_manager(settings)
```

Appendix C: urls.py

```
from django.conf.urls.defaults import patterns, include,
url

\# Uncomment the next two lines to enable the admin:
\# from django.contrib import admin
\# admin.autodiscover()
$urlpatterns = patterns('', $
    \# Examples:
    $url(r'^$', 'mysite.myapplication.views.home', name='
        home'), $
    url(r'^index$', 'mysite.myapplication.views.index',
        name='index'),
    url(r'^add_new$', 'mysite.myapplication.views.add_new
        ', name='add_new'),
    url(r'^update_status$', 'mysite.myapplication.views.
        update_status', name="update_status"),
    url(r'^search_add$', 'mysite.myapplication.views.
        search_add', name="search_add"),
    url(r'^more_ditail/(?P<id>(\d+))', 'mysite.
        myapplication.views.more_ditail', name="
        more_ditail"),
    url(r'^contact$', 'mysite.myapplication.views.contact',
        name='contact'), $
    url(r'^about$', 'mysite.myapplication.views.about',
        name='about'), $
```

## 7. APPENDIX

---

```
\#url(r'^save_uploaded_file', 'mysite.myapplication.
    views.save_uploaded_file', name="
    save_uploaded_file"),
\# url(r'^mysite/', include('mysite.foo.urls')),

\# Uncomment the admin/doc line below to enable admin
    documentation:
\# url(r'^admin/doc/', include('django.contrib.
    admindocs.urls')),

\# Uncomment the next line to enable the admin:
\# url(r'^admin/', include(admin.site.urls)),
\)
```

Appendix D: views.py

```
import facebook
import sqlite3
import random
import PIL
import json
from PIL import Image
from django.shortcuts import render
$from django_facebook.decorators import canvas_only$

import cgi
import cgitb; cgitb.enable()
import os, sys
$try: \# Windows needs stdio set for binary mode.$
    import msvcrt
    $msvcrt.setmode (0, os.O_BINARY) \# stdin = 0$
    $msvcrt.setmode (1, os.O_BINARY) \# stdout = 1$
except ImportError:
    pass

$UPLOAD\_DIR = ""$
```

---

```

$@canvas\_only$
def home(request):
    $me = request.facebook.graph.get_object('me')$
    userid= me['id']
    name = me['name']
    $friends = request.facebook.graph.get_connections("me
        ", "friends")$
    $rank =friends["data"]$
    if 'location' in me:
        location = me['location']['name']
        city = location.split(',')[0];
        country = location.split(',')[1];
        message=''
    else:
        message='your location is missing. Click home button
            to see all posts'
        city = ''
        country = ''
    $access_token= request.facebook.graph.access_token$
    $request.session['access_token'] = access_token$
    $request.session['me'] = me$

    connection = sqlite3.connect('/home/moodiraja/mysite/
        db.sqlite ')
    cursor = connection.cursor()
    try:
        $cursor.execute("INSERT INTO User (ID,NAME,COUNTRY,
            CITY,RANK,Accesstoken) VALUES (?, ?, ?, ?,?,?)",
            (userid , name, country , city ,len(rank) ,access_token
            ))$
    except:
        print 'welcome back'

    cursor.execute("select * from classifiedPost where
        CITY='" + city + "' and COUNTRY='" + country +"'")
    adds = cursor.fetchall();
    if len(adds)<1:
        if len(message)<1:

```

```
        message='No post from your city. Click home
            button to see all posts'
else:
    message=''
    cursor.execute("select * from classifiedPost order by
        id desc" )
    topadd = cursor.fetchall()[:10];
    cursor.execute("select * from Tag")
    tags = cursor.fetchall();
    connection.commit()
    connection.close()

    return render(request, 'home.html' , {'posts': adds, '
        top':topadd, 'tags':tags, 'message':message})
def index (request):

    connection = sqlite3.connect('/home/moodiraja/mysite/
        db.sqlite ')
    cursor = connection.cursor()
    cursor.execute("select * from classifiedPost")
    adds = cursor.fetchall();
    cursor.execute("select * from classifiedPost order by
        id desc" )
    topadd = cursor.fetchall()[:10];
    cursor.execute("select * from Tag")
    tags = cursor.fetchall();
    connection.commit()
    connection.close()

    return render(request, 'home.html' , {'posts': adds, '
        top':topadd, 'tags':tags })

$def add_new(request)$:
    $me = request.session['me']$
    \# mycity = request.session['myCity'];
    $access_token=request.session['access_token']$
    if 'location' in me:
        $return render(request, 'postNewAdd.html' , {'me':me,
            'access_token': access_token })$
```

---

```

    else:
        return render(request, 'error.html')
def condb():
    db = sqlite3.connect('data.db')
    c = db.cursor()
$def update_status(request):$
    $me = request.session['me']$
    connection = sqlite3.connect('/home/moodiraja/mysite/db
        .sqlite ')
    cursor = connection.cursor()

    request.POST.get('content', 'this is the default')
    add = request.POST.get('box', 'text missing' )
    Tag1 = request.POST.get('txtTag1', 'text missing' )
    Tag2 = request.POST.get('txtTag2', 'text missing' )
    Tag3 = request.POST.get('txtTag3', 'text missing' )
    userid = request.POST.get('userid', 0 )
    name = request.POST.get('username', 'N/A' )
    location = request.POST.get('userlocation', '0' )
    city = location.split(',')[0];
    country = location.split(',')[1];
    Tag = ""
    caption = request.POST.get('txtTitle', 'text missing' )
    picture = "http://moodiraja.pythonanywhere.com/media/
        noimage.gif"

\# data= source.file.read()
    $access_token = request.POST['access_token']$
    $graph = facebook.GraphAPI(access_token)$

    \#cursor.execute("select * from User where ID = ? ",
        (userid))
    \#check = cursor.fetchone()[0]
    \#if(check!= userid):
    try:
        $cursor.execute("INSERT INTO User (ID,NAME,COUNTRY,
            CITY,RANK,Accesstoken) VALUES (?, ?, ?, ?,?,?)
            ", (userid, name, country, city, 0, access_token))$
    except:

```

## 7. APPENDIX

---

```
    print 'welcome back'
    \#continue

cursor.execute("INSERT INTO classifiedPost (USERID ,
        TITLE ,DESCRIPTION ,TAG,CATEGORYID,TYPE,CITY,COUNTRY
        ) VALUES (?, ?, ?, ?, ?, ?,?,?)", (userid ,caption ,
        add,Tag,0,0 ,city ,country))

addid= cursor.lastrowid
if 'filename' in request.FILES:
    sources = request.FILES['filename'].read()
    fn = os.path.basename(str(addid)+".jpg")
    open('/home/moodiraja/mysite/media/' + fn, 'wb').
        write(sources)
    picture = "http://moodiraja.pythonanywhere.com/media
        /"+ str(addid)+".jpg"

$cursor.execute("INSERT INTO Tag (texttag , Post_ID )
    VALUES (?, ?)", (Tag1,addid))$
$cursor.execute("INSERT INTO Tag (texttag , Post_ID )
    VALUES (?, ?)", (Tag1,addid))$
$cursor.execute("INSERT INTO Tag (texttag , Post_ID )
    VALUES (?, ?)", (Tag3,addid))$
\#$cursor.execute("select id from User where city =
    '" + city + "'")
\#users = cursor.fetchall();$
\#$for id in users:
    \#graph.put_object(graph.get_object(id), "feed",
        message=add,description=add,picture=picture,name=
        caption)$
connection.commit()
connection.close()

$graph.put_object("me", "feed",message=add,description=
    add,picture=picture,name=caption)$

$return render(request, 'all_done.html',{ 'message' : "
    done"})$
```

---

```

def contact(request):
    return render(request, 'contact.html')
def about(request):
    return render(request, 'about.html')
$def search_add(request):$
    $search = request.POST.get('searchField','')$

    connection = sqlite3.connect('/home/moodiraja/mysite/
        db.sqlite')
    cursor = connection.cursor()

    message= ""

    cursor.execute("select * from classifiedPost where
        DESCRIPTION like '%" + search +"%' or title like '%"
        + search +"%' ")
    adds = cursor.fetchall();
    if len(adds) <1:
        message = "Could not find"
    cursor.execute("select * from Tag")
    tags = cursor.fetchall();
    connection.commit()
    connection.close()

    return render(request, 'home.html' , {'posts': adds, '
        tags':tags, 'message':message})
$def more_ditail(request, id):$
    \# $ me = request.facebook.graph.get_object('me')$

    connection = sqlite3.connect('/home/moodiraja/mysite/
        db.sqlite')
    cursor = connection.cursor()
    \# $ access_token= request.facebook.graph.access_token
    access_token= request.session['access_token']
    me =request.session['me']$

    \#
    cursor.execute("select * from classifiedPost where id
        =" + id)

```

## 7. APPENDIX

---

```
adds = cursor.fetchall();
cursor.execute("select * from Tag")
tags = cursor.fetchall();
connection.commit()
connection.close()

return render(request, 'add.html' , {'posts': adds, '
    tags': tags })
```

### Appendix E: home.html

```
\textless html xmlns="http://www.w3.org/1999/xhtml" \
    textgreater
\textless head \textgreater

\textless title \textgreater shop/title \textgreater
$ \textless link href="{{ STATIC_URL }}templatemo_style.
    css" rel="stylesheet" type="text/css" / \textgreater$
$ \textless link rel="stylesheet" type="text/css" href="{{
    STATIC_URL }}css/ddsmoothmenu.css" / \textgreater $
$ \textless script type="text/javascript" src="{{
    STATIC_URL }}js/jquery.min.js" \textgreater \textless /
    script \textgreater $
$ \textless script type="text/javascript" src="{{
    STATIC_URL }}js/ddsmoothmenu.js" \textgreater
    \textless /script \textgreater $
$ \textless script type="text/javascript" \textgreater $

ddsmoothmenu.init({
    $mainmenuid: "templatemo_menu", //menu DIV id
    orientation: 'h', //Horizontal or vertical menu:
        Set to "h" or "v"
    classname: 'ddsmoothmenu', //class added to menu's
        outer DIV
    //customtheme: ["\#1c5a80", "\#18374a"],$
    $contentsource: "markup" //"markup" or ["
        container_id", "path_to_menu_file"])$
```



---

```

\textless /script \textgreater

$ \textless link rel="stylesheet" type="text/css" media="
  all" href="{{ STATIC_URL }}css/jquery.dualSlider.0.2.
  css" / \textgreater $

$ \textless script src="{{ STATIC_URL }}js/jquery-1.3.2.
  min.js" type="text/javascript" \textgreater \textless /
  script \textgreater $
$ \textless script src="{{ STATIC_URL }}js/jquery.easing
  .1.3.js" type="text/javascript" \textgreater \textless
  /script \textgreater $
$ \textless script src="{{ STATIC_URL }}js/jquery.timers
  -1.2.js" type="text/javascript" \textgreater \textless
  /script \textgreater $
$ \textless script src="{{ STATIC_URL }}js/jquery.
  dualSlider.0.3.min.js" type="text/javascript" \
  textgreater \textless /script \textgreater $

\textless script type="text/javascript" \textgreater

  \$(document).ready(function() {

    \$("#carousel").dualSlider({
      auto:true,
      autoDelay: 6000,
      easingCarousel: "swing",
      easingDetails: "easeOutBack",
      durationCarousel: 1000,
      durationDetails: 600
    });

  });

\textless /script \textgreater
$ \textless link rel="stylesheet" href="{{ STATIC_URL }}
  css/slimbox2.css" type="text/css" media="screen" / \
  textgreater $

```

## 7. APPENDIX

---

```
$ \textless script type="text/JavaScript" src="{{
  STATIC_URL }}js/slimbox2.js" \textgreater \textless /
script>$
\textless /head \textgreater
\textless body\textgreater

$ \textless div id="templatemo_header_wrapplicationer" \
textgreater $

$ \textless div id="templatemo_header" class="
wrapplicationer" \textgreater $

$ \textless div id="templatemo_menu" class="
ddsmoothmenu" \textgreater $

$ \textless ul \textgreater $
  \textless li \textgreater \textless a
    href$="{% url index %}" \textgreater
    Home\textless /a \textgreater \textless
    /li \textgreater
  \textless li \textgreater \textless a
    href$="{% url add_new %}" \textgreater
    Create Add \textless /a \textgreater /
    li \textgreater
  \textless li \textgreater \textless a
    href$="{% url about %}" \textgreater
    About\textless /a \textgreater \
    textless /li \textgreater
  \textless li \textgreater \textless a href$
    ="{% url contact %}" \textgreater
    Contact \textless /a \textgreater \
    textless /li \textgreater
\textless /ul \textgreater
\textless br style$="clear: left" / \
textgreater
$ \textless /div \textgreater ... end of
templatemo_menu$
```

---

```

    $ \textless /div \textgreater $
\textless /div \textgreater
\textless div id="$templatemo_middle_wrapplicationer" \
textgreater $
    \textless div id="$templatemo_middle$" class="$"
        $wrapplicationer$" \textgreater
    \textless div id="$templatemo_slider$" \
        textgreater
        \textless div id="$scarousel$" \
            textgreater
        \textless div class="$panel" \textgreater
            \textless div class="$"
                $details_wrapplicationer$" \
                    textgreater
            \textless div class="$details$"
                \textgreater
            {%for postid , userid , title , dis , tag ,
                cat , type , city , country in top %}
                \textless div class="$detail"
                    \textgreater
                    \textless h2 \textgreater
                        \textless a href="$"#
                            v {{ title }} \
                                textless /a \
                                    textgreater \textless /
                                        h2 \textgreater \
                                            textless p \textgreater
                                                {{ city }} / {{country}}
                                                    \textless /p\
                                                        textgreater
                    \textless p \textgreater {
                        dis } \textless /p \
                            textgreater

                ( \textless a href=${%
                    url more_ditail postid
                    %} title="$Read more
                    " class="$slider_more
                    " \textgreater Read

```

```

        more \textless /a \
        textgreater \
    \textless /div\textgreater \
        textless !-- /detail --\
        textgreater
    {%endfor%}
    \textless /div \textgreater \
        textless !-- /details — \
        textgreater

$ \textless /div \textgreater \
    textless /details_wrapplicationer \
    textgreater $

\textless /div \textgreater \textless !-- /panel —
    \textgreater

    \textless a href="javascript:void(0);"
        class="previous" title="Previous"
    \textgreater Previous \textless /a
    \textgreater
    \textless a href="javascript:void(0)
        ;" class="next" title="Next" \
    textgreater Next \textless /a \
    textgreater

\textless div id="slider-image-frame" \
    textgreater
    \textless div class="backgrounds" \
    textgreater
        {%for postid , userid , title , dis , tag , cat
            , type , city , country in top %}
    \textless div class$=$" $item item_1$" \
    textgreater
\textless img src $=$" ${{ MEDIA_URL }}{{ postid }}.
    jpg$" \textgreater
    $ \textless /div \textgreater \
        textless !-- /item --\textgreater $
    {%endfor%}
```

---

```

        \textless /div \textgreater \textless !--
        /backgrounds --\textgreater
        \textless /div \textgreater
        \textless /div \textgreater
\textless /div \textgreater \textless $!-- END of
    templatemo_slider -- \textgreater $

```

```

\textless /div \textgreater

```

```

\textless form method="POST" id="search" action="{% url
    search_add %}" enctype = "multipart/form-data" \
    textgreater
\textless div id="searchbar" \textgreater
    <input value="" style="width:87%;"
        maxlength="100" placeholder="enter
        your search here" id="searchField"
        name="searchField">
        \textless button type="submit" \
            textgreater Search \textless /
            button \textgreater
\textless /div \textgreater
    \textless /form \textgreater

```

```

\textless /div \textgreater

```

```

\textless div id$=$"templatemo_main$" class$=$"
    $wrapplicationer$" \textgreater
    {%for postid , userid , title ,dis ,tag ,cat ,type ,
        city ,country in posts %}
\textless div class$=$"$col one_third no_margin_right$" \
    textgreater

```

```

\textless h4 \textless {{ title }} \textless /h4\
    textgreater\textless p \textgreater {{ city }}/{{ country

```

```
}} \textless /p \textgreater
  \textless div class =$"$ img_nom$" \textgreater \
    textless span\textgreater\textless /span \
    textgreater
    \textless img src =$"${{ MEDIA_URL }}{{ postid
      }}.jpg" style=" height: 100px; width: 100px;
      $" \textgreater
    \textless /div \textgreater
  \textless p \textgreater {{ dis }} \textless /p \
    textgreater

  \textless a href$=${% url more_ditail postid %}
    class="$more$" \textgreater More \textless /a \
    textgreater
\textgreater /div \textgreater
{%endfor%}
\textless p \textgreater {{ message }}\textless / p \
textgreater

  \textless div class="clear" \textgreater \textless /
  div \textgreater
\textless /div \textgreater

\textless div id$=$"$templatemo_footer_wrapplicationer$" \
textgreater
  \textless div id$=$"$templatemo_footer$" class$=$"
  $wrapplicationer$" \textgreater
  Copyright Âl 2013 \textless a href$=$"$\#$" \
  textgreater GoodsBook \textless /a \textgreater
  by \textless a href="$\#$" target="$_parent$"
  title="$GoodsBook$" >Goodsbook.com \textless /a
  \textgreater
  \textless /div \textgreater
\textless /div \textgreater
```

---

```
\textless script type='text/javascript' src='js/logging.js
  \textgreater \textless /script\textgreater
\textless /body \textgreater
\textless /html \textgreater
```

Appendix F: postNewAdd.html

```
\textless html xmlns="http://www.w3.org/1999/xhtml"\
  textgreater
\textless head\textgreater

\textless title\textgreater shop\textless/title\
  textgreater
\textless link href$={{ STATIC_URL }}templatemo_style.
  css" rel="stylesheet" type="$text/css" /\textgreater
\textless link rel="$stylesheet" type="$text/css" href$=
  "${{ STATIC_URL }}css/ddsmoothmenu.css" /\textgreater
\textless script type="$text/javascript" src$={{
  STATIC_URL }}js/jquery.min.js"\textgreater\textless/
  script\textgreater
\textless script type="$text/javascript" src$={{
  STATIC_URL }}js/ddsmoothmenu.js"\textgreater
\textless/script\textgreater

\textless script type="$text/javascript"\textgreater

ddsmoothmenu.init({
  mainmenuid: "$templatemo_menu", //menu DIV id
  orientation: 'h', //Horizontal or vertical menu: Set
    to "$h" or "$v"
  classname: 'ddsmoothmenu', //class added to menu's
    outer DIV
  //customtheme: ["$#1c5a80", "$#18374a"],
  contentsource: "$markup" //"$markup" or ["
    $container_id", "$path_to_menu_file"]
})

\textless/script\textgreater
```

## 7. APPENDIX

---

```
\textless link rel$=$"stylesheet" type$=$"text/css"
  media$=$"all" href$=$"${{ STATIC_URL }}css/jquery.
  dualSlider.0.2.css" /\textgreater

\textless script src$=$"${{ STATIC_URL }}js/jquery-1.3.2.
  min.js" type$=$"text/javascript"\textgreater\
  textless/script\textgreater
\textless script src$=$"${{ STATIC_URL }}js/jquery.easing
  .1.3.js" type$=$"text/javascript"\textgreater\
  textless/script\textgreater
\textless script src$=$"${{ STATIC_URL }}js/jquery.timers
  -1.2.js" type$=$"text/javascript"\textgreater\
  textless/script\textgreater
\textless script src$=$"${{ STATIC_URL }}js/jquery.
  dualSlider.0.3.min.js" type$=$"text/javascript"\
  textgreater\textless/script\textgreater
\textless script type$=$"text/javascript"\textgreater
  (document).ready(function() {

      ("#carousel").dualSlider({
        auto: true,
        autoDelay: 6000,
        easingCarousel: "swing",
        easingDetails: "easeOutBack",
        durationCarousel: 1000,
        durationDetails: 600
      });

  });

\textless/script\textgreater
\textless link rel$=$"stylesheet" href$=$"${{ STATIC_URL }}
  css/slimbox2.css" type$=$"text/css" media$=$"screen" /\
  textgreater
\textless script type$=$"text/JavaScript" src$=$"${{
  STATIC_URL }}js/slimbox2.js"\textgreater\textless/
  script\textgreater
\textless/head\textgreater
```



---

```

\textless body\textgreater

\textless div id$=$"templatemo_header_wrapplicationer$"
  \textless div id$=$"templatemo_header$" class$=$"
    $wrapplicationer$" \textgreater
  \textless div id$=$"$site_title$" \textgreater \
    \textless a href$=$"http://www.templatemo.com" \
      \textgreater Free CSS Templates \textless /a \
        \textgreater \textless /div \textgreater
  \textless div id$=$"templatemo_menu$" class$=$"
    $ddsmoothmenu$" \textgreater
    \textless ul \textgreater
      \textless li \textgreater \textless a href$=
        "${% url index %}" \textgreater Home \
          \textless /a \textgreater \textless /li \
            \textgreater
      \textless }li \textgreater \textless a href$=$"
        {% url add_new %}" \textgreater Create Add \
          \textless /a \textgreater \textless /li \
            \textgreater
      \textless li \textgreater \textless a href $=$"
        {% url about %}" \textgreater About \
          \textless /a \textgreater \textless / li \
            \textgreater
      \textless }li \textgreater \textless a href $=$"
        {% url contact %}" \textgreater Contact \
          \textless /a \textgreater \textless /li \
            \textgreater
    \textless /ul \textgreater
    \textless br style$=$"$clear: left$" / \
      \textgreater
  \textless /div \textgreater \textless $- end of
    \textless /div \textgreater
  \textless form method$=$"POST" id$=$"$search$" action$=$"
    "${% url search_add %}" enctype $=$ "$multipart/
    form-data" \textgreater

```

## 7. APPENDIX

---

```
\textless/div\textgreater
\textless /div }id$=$"searchbar"\textgreater
\textless input value$=$" style$=$"width:87%;"$maxlength$=$
  "100"$placeholder$=$"enter your search here"$id$=$"
  searchField"$ name$=$"searchField"\textgreater
\textless button type$=$"submit"\textgreater Search \
  textless/button\textgreater
\textless /div \textgreater
\textless /form\textgreater
\textless span class$=$"$post-tag$"\textgreater
\textless br/\textgreater\textless br\textgreater\textless
  /br\textgreater
  \textless form method$=$"POST"id$=$"add"action$=$
    "{% url update\_status %}"enctype $=$ "
    multipart/form-data"\textgreater
\textless span }style$=$""\textgreater Your Location : {{
  me.location.name}}
  \textless table\textgreater
  \textless tr\textgreater\textless td\
    textgreater Title:\textless/td\textgreater
    \textless td\textgreater\textless input
    type$=$"text"name $=$"txtTitle"id $=$"
    txtTitle"/\textgreater\textless/td\
    textgreater\textless/tr\textgreater
  \textless tr\textgreater\textless td style$=$
    "vertical-align:text-top;"\textgreater
    Discription:\textless/td\textgreater
  \textless td\textgreater\textless textarea
    cols$=$"55"rows$=$"12"name$=$"box"id$=$"
    box"tabindex$=$"2"\textgreater \textless/
    textarea\textgreater\textless/td\
    textgreater
  \textless/tr\textgreater\textless tr\
    textgreater\textless td\textgreater Tag:\
    textless/td\textgreater\textless td\
    textgreater \textless input type$=$"text"
    name $=$"txtTag1"id $=$"txtTag1"/\
    textgreater\textless input type$=$"text"
    name $=$"txtTag2"id $=$"txtTag2"/\
```

---

```

        textgreater\textless input type$=$"text"
        name $=$"txtTag3" id $=$"txtTag3"/\
        textgreater\textless \hspace*{5mm}\
        textgreater\textless/tr\textgreater
\textless tr\textgreater\textless /\textgreater Select
    product image: \&nbsp;\textless/td\textgreater\textless
    \hspace*{5mm}\textgreater \textless input name$=$"
    filename" type$=$"file"/\textgreater\textless/td\
    textgreater\textless/tr\textgreater
\textless tr\textgreater\textless \hspace*{5mm}\
    textgreater
&nbsp;\textless/td\textgreater\textless \hspace*{5mm}
    style$=$"text-align:right"\textgreater\textless input
    type$=$"submit" class$=$"submit_btn" style$=$"float:
    right;" value$=$"Post"/\textgreater\textless/td\
    textgreater\textless/tr\textgreater
\textless/table\textgreater

\textless input type$=$"hidden" name$=$"quote" value$=$"{{
    quote}}"/\textgreater
\textless input type$=$"$hidden" name$=$"$access\_token"
    value$=$"{{access_token}}"/\textgreater
\textless input type$=$"hidden" name$=$"username" value$=$
    "{{me.name}}"/\textgreater
\textless input type$=$"hidden" name$=$"userid" value$=$"{{
    me.id}}"/\textgreater \textless input type$=$"hidden"
    name$=$"userlocation" value$=$"{{me.location.name}}"/\
    textgreater
\textless /form \textgreater
\textless /body \textgreater
\textless /html \textgreater

```

#### Appendix G: add.html

```

\textless html xmlns$=$"http://www.w3.org/1999/xhtml"\
    textgreater
\textless head\textgreater
\textless title\textgreater shop\textless /title\
    textgreater

```

## 7. APPENDIX

---

```
\textless link href$=${{ STATIC_URL }}templatemo_style.
  css" rel$=${"stylesheet" type$=${"text/css" /\textgreater
\textless link rel$=${"stylesheet" type$=${"text/css" href$=
  ${{ STATIC_URL }}css/ddsmoothmenu.css" /\textgreater
\textless script type$=${"text/javascript" src$=${{{
  STATIC_URL }}js/jquery.min.js"\textgreater \textless /
  script\textgreater
\textless script type$=${"text/javascript" src$=${{{
  STATIC_URL }}js/ddsmoothmenu.js"\textgreater
\textless /script\textgreater

\textless script type$=${"text/javascript"\textgreater

ddsmoothmenu.init({
  mainmenuid: "templatemo_menu", //menu DIV id
  orientation: 'h', //Horizontal or vertical menu:
    Set to "h" or "v"
  classname: 'ddsmoothmenu', //class added to menu's
    outer DIV
  //customtheme: ["\#1c5a80", "\#18374a"],
  contentsource: "markup" //"markup" or ["
    container_id", "path_to_menu_file"]
})

\textless /script\textgreater

\textless link rel$=${"stylesheet" type$=${"text/css" media$
  =${"all" href$=${{{ STATIC_URL }}css/jquery.dualSlider
  .0.2.css" /\textgreater

\textless script src$=${{{ STATIC_URL }}js/jquery-1.3.2.
  min.js" type$=${"text/javascript"\textgreater \textless
  /script\textgreater
\textless script src$=${{{ STATIC_URL }}js/jquery.easing
  .1.3.js" type$=${"text/javascript"\textgreater \
  textless /script\textgreater
\textless script src$=${{{ STATIC_URL }}js/jquery.timers
  -1.2.js" type$=${"text/javascript"\textgreater \
  textless /script\textgreater
```

---

```

\textless script src$=${{ STATIC_URL }}js/jquery.
  dualSlider.0.3.min.js" type$="text/javascript"\
  textgreater \textless /script\textgreater

\textless script type$="text/javascript"\textgreater

  $(document).ready(function() {

    $("#carousel").dualSlider({
      auto:true,
      autoDelay: 6000,
      easingCarousel: "swing",
      easingDetails: "easeOutBack",
      durationCarousel: 1000,
      durationDetails: 600
    });

  });

\textless /script\textgreater
\textless link rel$="stylesheet" href$=${{ STATIC_URL }}
  css/slimbox2.css" type$="text/css" media$="screen" /\
  textgreater
\textless script type$="text/JavaScript" src$=${{
  STATIC_URL }}js/slimbox2.js"\textgreater \textless /
  script\textgreater
\textless /head\textgreater
\textless body\textgreater

\textless div id$="templatemo_header_wrapplicationer"\
  textgreater
  \textless div id$="templatemo_header" class$="
    wrapplicationer"\textgreater
  \textless div id$="site_title"\textgreater \
    textless a href$="http://www.templatemo.com"\
    textgreater Free CSS Templates\textless /a\
    textgreater \textless /div\textgreater
  \textless div id$="templatemo_menu" class$="
    ddsmoothmenu"\textgreater

```

```
\textless ul\textgreater
  \textless li\textgreater \textless a
    href$=${"% url index %}"\textgreater
    Home\textless /a\textgreater \textless
    /li\textgreater
  \textless }li\textgreater \textless a
    href$=${"% url add_new %}"\textgreater
    Create Add\textless /a\textgreater \
    textless /li\textgreater
  \textless }li\textgreater \textless a
    href$=${"% url about %}"\textgreater
    About\textless /a\textgreater \
    textless /li\textgreater
  \textless }li\textgreater \textless a
    href$=${"% url contact %}"\textgreater
    Contact\textless /a\textgreater \
    textless /li\textgreater
\textless /ul\textgreater
\textless }br style$=${"clear: left" /\
  textgreater
\textless /div\textgreater \textless !-- end of
  templatemo_menu --\textgreater
\textless /div\textgreater
\textless form method$=${"POST" id$=${"search" action$=${
  "% url search_add %}" enctype $=${ "multipart/form-
  data" \textgreater

\textless /div\textgreater
\textless div }id$=${"searchbar"\textgreater
  \textless input value$=${"" style$=${"
    width:87%;" maxlength$=${"100"
    placeholder$=${"enter your search here
    " id$=${"searchField" name$=${"
    searchField"\textgreater
  \textless button type$=${"submit"\
    textgreater Search\textless /
    button\textgreater
\textless /div\textgreater
\textless /form\textgreater
```

---

```

\textless table style$=$" height: 300px; width: 100%;"
border$=$"1" \textgreater
{%for postid , userid ,title ,dis ,tag ,cat ,type ,city ,
country in posts %}
\textless tr\textgreater \textless td\textgreater
\textless table }style$=$" width: 100%;" \
textgreater
\textless tr\textgreater \textless td colspan$=$
"2"\textgreater \textless h3 style$=$"text-
align:center;"\textgreater {{ title }}\
textless /h3\textgreater \textless /td\
textgreater \textless /tr\textgreater

\textless tr\textgreater \textless td colspan$=$
"2"\textgreater \textless p\textgreater {{
dis }}\textless /p\textgreater \textless /td\
textgreater \textless /tr\textgreater

\textless tr\textgreater \textless td \
textgreater (Post Location : {{ city }},{
country })
\textless /td\textgreater \textless td style$=$"text-
align:right;"\textgreater
\textless div class$=$"item item_1"\textgreater
\textless img src $=$"{{ MEDIA_URL }}{{ postid }}.
jpg"\textgreater
\textless /div\textgreater

\textless /td\textgreater \textless /tr\
textgreater
\textless /table\textgreater
\textless div id$=$"fb-root"\textgreater \textless /
div\textgreater
\textless script\textgreater (function(d, s, id) {
}var js , fjs $=$ d.getElementsByTagName(0);
if (d.getElementById) return;
js $=$ d.createElement(s); js.id $=$ id;
js.src $=$ "//connect.facebook.net/en_US/all.js#\xfbml$=
$1 \&applicationId$=$249588381832401";

```

## 7. APPENDIX

---

```
fjs.parentNode.insertBefore(js, fjs);
(document, 'script', 'facebook-jssdk');\textless /script\
textgreater
\textless div class$=$"fb-comments" data-href$=$"{{
MEDIA_URL }}{{ postid }}.jpg" data-width$=$"735"\
textgreater \textless /div\textgreater

{%endfor%}
\textless /table\textgreater
\textless /body\textgreater
\textless /html\textgreater

}
```

Appendix H: all\_done.html

```
\textless html xmlns$=$"http://www.w3.org/1999/xhtml"\
textgreater
\textless head\textgreater

\textless title\textgreater shop\textless /title\
textgreater
\textless link href$=$"{{ STATIC_URL }}templatemo_style.
css" rel$=$"stylesheet" type$=$"text/css" /\textgreater
\textless link rel$=$"stylesheet" type$=$"text/css" href$=
$"{{ STATIC_URL }}css/ddsmoothmenu.css" /\textgreater
\textless script type$=$"text/javascript" src$=$"{{
STATIC_URL }}js/jquery.min.js"\textgreater \textless /
script\textgreater
\textless script type$=$"text/javascript" src$=$"{{
STATIC_URL }}js/ddsmoothmenu.js"\textgreater
\textless /script\textgreater

\textless script type$=$"text/javascript"\textgreater

ddsmoothmenu.init({
mainmenuid: "templatemo_menu", //menu DIV id
```



---

```

orientation: 'h', //Horizontal or vertical menu: Set
to "h" or "v"
classname: 'ddsmoothmenu', //class added to menu's
outer DIV
//customtheme: ["\# 1c5a80", "\# 18374a"],
contentsource: "markup" //"markup" or ["
container_id", "path_to_menu_file"]
})

\textless /script\textgreater

\textless link rel$="$stylesheet" type$="$text/css" media$
=$"all" href$="$"{{ STATIC_URL }}css/jquery.dualSlider
.0.2.css" /\textgreater

\textless script src$="$"{{ STATIC_URL }}js/jquery-1.3.2.
min.js" type$="$text/javascript"\textgreater \textless
/script\textgreater
\textless script src$="$"{{ STATIC_URL }}js/jquery.easing
.1.3.js" type$="$text/javascript"\textgreater \textless
/script\textgreater
\textless script src$="$"{{ STATIC_URL }}js/jquery.timers
-1.2.js" type$="$text/javascript"\textgreater \textless
/script\textgreater
\textless script src$="$"{{ STATIC_URL }}js/jquery.
dualSlider.0.3.min.js" type$="$text/javascript"\
textgreater \textless /script\textgreater

\textless script type$="$text/javascript"\textgreater

$(document).ready(function() {

    $("# carousel").dualSlider({
        auto:true,
        autoDelay: 6000,
        easingCarousel: "swing",
        easingDetails: "easeOutBack",
        durationCarousel: 1000,
        durationDetails: 600
    })
})

```

```
    });  
  
  \});  
  
  \textless /script\textgreater  
  \textless link rel$=$"stylesheet" href$=$"{{ STATIC_URL }}  
    css/slimbox2.css" type$=$"text/css" media$=$"screen" /\textgreater  
  \textless script type$=$"text/JavaScript" src$=$"{{  
    STATIC_URL }}js/slimbox2.js"\textgreater \textless /  
    script\textgreater  
  \textless /head\textgreater  
  \textless body\textgreater  
  
  \textless div id$=$"templatemo_header_wrapplicationer"\textgreater  
    \textless div id$=$"templatemo_header" class$=$"  
      wrapplicationer"\textgreater  
      \textless div id$=$"site_title"\textgreater \textless a href$=$"http://www.templatemo.com"  
        \textgreater Free CSS Templates\textless /a\textgreater  
        \textless /div\textgreater  
      \textless div id$=$"templatemo_menu" class$=$"  
        ddsmoothmenu"\textgreater  
        \textless ul\textgreater  
          \textless \}li\textgreater \textless a href$=$"{% url index %}"\textgreater  
            Home\textless /a\textgreater \textless  
            /li\textgreater  
          \textless }li\textgreater \textless a href$=$"{% url add_new %}"\textgreater  
            Create Add\textless /a\textgreater \textless  
            /li\textgreater  
          \textless }li\textgreater \textless a href$=$"{% url about %}"\textgreater  
            About\textless /a\textgreater \textless  
            /li\textgreater
```

---

```

        \textless }li\textgreater \textless a
            href$=${"% url contact %}"\textgreater
            Contact\textless /a\textgreater \
            textless /li\textgreater
    \textless /ul\textgreater
    \textless }br style$=${"clear: left" /\
        textgreater
    \textless /div\textgreater \textless !-- end of
        templatemo_menu --\textgreater
\textless /div\textgreater
\textless form method$=${"POST" id$=${"search" action$=${
    "{% url search_add %}" enctype $=${ "multipart/form-
    data" \textgreater

\textless /div\textgreater
\textless div id$=${"searchbar"\textgreater
    \textless input value$=${"" style$=${"
        width:87%;" maxlength$=${"100"
        placeholder$=${"enter your search here
        " id$=${"searchField" name$=${"
        searchField"\textgreater
    \textless button type$=${"submit"\
        textgreater Search\textless /button
        \textgreater
\textless /div\textgreater
    \textless /form\textgreater

\textless span class$=${"post-tag"\textgreater

\textless p\textgreater \textless br\textgreater \textless
    /br\textgreater \textless br\textgreater \textless /br
    \textgreater
    Your Add has been succussfully published.....

\textless /p\textgreater
\textless /body\textgreater
\textless /header\textgreater
\textless /html\textgreater

```

## Appendix I: error.html

```
\textless html xmlns$=$"http://www.w3.org/1999/xhtml"\
  textgreater
\textless head\textgreater

\textless title\textgreater shop\textless /title\
  textgreater
\textless link href$=$"{{ STATIC_URL }}templatemo_style.
  css" rel$=$"stylesheet" type$=$"text/css" /\textgreater
\textless link rel$=$"stylesheet" type$=$"text/css" href$=
  $"{{ STATIC_URL }}css/ddsmoothmenu.css" /\textgreater
\textless script type$=$"text/javascript" src$=$"{{
  STATIC_URL }}js/jquery.min.js"\textgreater \textless /
  script\textgreater
\textless script type$=$"text/javascript" src$=$"{{
  STATIC_URL }}js/ddsmoothmenu.js"\textgreater
\textless /script\textgreater

\textless script type$=$"text/javascript"\textgreater

ddsmoothmenu.init({
  mainmenuid: "templatemo_menu", //menu DIV id
  orientation: 'h', //Horizontal or vertical menu: Set
    to "h" or "v"
  classname: 'ddsmoothmenu', //class added to menu's
    outer DIV
  //customtheme: ["\#1c5a80", "\#18374a"],
    contentsource: "markup" //"markup" or ["
      container_id", "path_to_menu_file"]
})

\textless /script\textgreater

\textless link rel$=$"stylesheet" type$=$"text/css" media$
  =$"all" href$=$"{{ STATIC_URL }}css/jquery.dualSlider
  .0.2.css" /\textgreater
```

---

```

\textless script src$="$"{{ STATIC_URL }}js/jquery-1.3.2.
  min.js" type$="$"text/javascript"\textgreater \textless
  /script\textgreater
\textless script src$="$"{{ STATIC_URL }}js/jquery.easing
  .1.3.js" type$="$"text/javascript"\textgreater \textless
  /script\textgreater
\textless script src$="$"{{ STATIC_URL }}js/jquery.timers
  -1.2.js" type$="$"text/javascript"\textgreater \textless
  /script\textgreater
\textless script src$="$"{{ STATIC_URL }}js/jquery.
  dualSlider.0.3.min.js" type$="$"text/javascript"\
  textgreater \textless /script\textgreater

\textless script type$="$"text/javascript"\textgreater

  $(document).ready(function() {

    $("#carousel").dualSlider({
      auto:true,
      autoDelay: 6000,
      easingCarousel: "swing",
      easingDetails: "easeOutBack",
      durationCarousel: 1000,
      durationDetails: 600
    });

  });

\textless /script\textgreater
\textless link rel$="$"stylesheet" href$="$"{{ STATIC_URL }}
  css/slimbox2.css" type$="$"text/css" media$="$"screen" /\
  textgreater
\textless script type$="$"text/JavaScript" src$="$"{{
  STATIC_URL }}js/slimbox2.js"\textgreater \textless /
  script\textgreater
\textless /head\textgreater
\textless body\textgreater

```

```
\textless div id$=$"templatemo_header_wrapplicationer"\
textgreater
  \textless div id$=$"templatemo_header" class$=$"
  wrapplicationer"\textgreater
  \textless div id$=$"site_title"\textgreater \
  textless a href$=$"http://www.templatemo.com"\
  textgreater Free CSS Templates\textless /a\
  textgreater \textless /div\textgreater
  \textless div id$=$"templatemo_menu" class$=$"
  ddsmoothmenu"\textgreater
  \textless ul\textgreater
    \textless li\textgreater \textless a href$
    =${"% url index %}"\textgreater Home\
    textless /a\textgreater \textless /li\
    textgreater
  \textless }li\textgreater \textless a
  href$=${"% url add_new %}"\textgreater
  Create Add\textless /a\textgreater \
  textless /li\textgreater
  \textless }li\textgreater \textless a
  href$=${"% url about %}"\textgreater
  About\textless /a\textgreater \textless
  /li\textgreater
  \textless }li\textgreater \textless a
  href$=${"% url contact %}"\textgreater
  Contact\textless /a\textgreater \
  textless /li\textgreater
  \textless /ul\textgreater
  \textless }br style$=$"clear: left" /\
  textgreater
  \textless /div\textgreater \textless !-- end of
  templatemo_menu --\textgreater
\textless /div\textgreater
\textless form method$=$"POST" id$=$"search" action$=$
"{% url search_add %}" enctype $=$ "multipart/form-
data" \textgreater

\textless /div\textgreater
\textless div }id$=$"searchbar"\textgreater
```

---

```

        \textless input value$=$"" style$=$"
            width:87%;" maxlength$=$"100"
            placeholder$=$"enter your search here
            " id$=$"searchField" name$=$"
            searchField"\textgreater
        \textless button type$=$"submit"\
            textgreater Search\textless /button
        \textgreater
    \textless /div\textgreater
    \textless /form\textgreater

\textless span class$=$"post-tag"\textgreater

\textless p\textgreater \textless br\textgreater \textless
    /br\textgreater \textless br\textgreater \textless /br
    \textgreater
    Please set your location in facebook profile to post an
    ad.

\textless /p\textgreater
    \textless /body\textgreater
\textless /header\textgreater
\textless /html\textgreater

```

\$Appendix M: Rules\\_Filter.py

```

from assoc import Apriori , printResults
import os

def datafile(name):
    ifile \&= open (name)
    for line in ifile:
        ret \&= (line)
    yield ret

\# clean\_data $=$ '/home/khizer/Machine Learning
    Algorithms/Amazon Data/clean/c2.csv'

```

## 7. APPENDIX

---

```
\# clean\_rule\_data \&= '$clean\_rule2.txt$'
\# minSupport      $=$ 0.0
\# infile\_clean \&= datafile(clean\_data)
\# cleanrules     \&= Apriori(infile\_clean, minSupport)
\# get\_clean\_rules \&= printResults(cleanrules)
\# text\_file \&= open(clean\_rule\_data, 'a')
\# text\_file.write(get\_clean\_rules)

\# unclean\_data \&= '/home/khizer/Machine Learning
  Algorithms/Amazon Data/unclean/u2.csv'
\# unclean\_rule\_data \&= 'urule2.txt'
\# minSupport      \&= 0.0
\# infile\_unclean \&= datafile(unclean\_data)
\# uncleanrules    \&= Apriori(infile\_unclean, minSupport
  )
\# get\_unclean\_rules \&= printResults(uncleanrules)
\# text\_file \&= open(unclean\_rule\_data, 'a')
\# text\_file.write(get\_unclean\_rules)

def search\_clean(infile\_clean, infile\_for\_clean):
    genrules \&= []
    comprules \&= []
    lastrule \&= []
    state \&= False
    for items in infile\_clean:
        genrules.applicationend(items)

    for item\_curr in infile\_for\_clean:
        comprules.applicationend(item\_curr)

    for rule1 in genrules:
        for rule2 in comprules:
            if rule2 $==$ rule1:
                state \&= True
                lastrule.applicationend(
                    rule2)
```



---

```

percentage \&= (100*(float(len(lastrule))/float(len
(genrules))))
return\_num \&= return\_percentage(percentage)

if return\_num $==$ 1:
    state \&= True
elif return\_num $==$ 0:
    state \&= False
elif return\_num $==$ 2:
    state \&= True
elif return\_num $==$ 3:
    state \&= False

return state, ''.join(lastrule), ''.join(genrules)

def search\_unclean(infile\_unclean, infile\_for\_unclean):
    genrules \&= []
    comprules \&= []
    lastrule \&= []
    for items in infile\_unclean:
        genrules.applicationend(items)

    for item\_curr in infile\_for\_unclean:
        comprules.applicationend(item\_curr)

    for rule1 in genrules:
        for rule2 in comprules:
            if rule2 $==$ rule1:
                lastrule.applicationend(
                    rule2)

percentage \&= (100*(float(len(lastrule))/float(len
(genrules))))

return\_num \&= return\_percentage(percentage)

if return\_num $==$ 1:
    state \&= True
elif return\_num $==$ 0:

```

```
        state \&= False
    elif return\_num $==$ 2:
        state \&= True
    elif return\_num $==$ 3:
        state \&= False

    return state , ''.join(lastrule) , ''.join(genrules)

def Tester(fileclass , return\_Cstate , return\_Ustate):
    label \&= 'None'
    if (return\_Ustate $==$ True or return\_Cstate $==$
        $ False):
        Label \&= 'TP'
    if (return\_Ustate $==$ False or return\_Cstate $
        ==$ True):
        Label \&= 'TN'
    if (return\_Ustate $==$ False or return\_Cstate $
        ==$ False):
        if fileclass $==$ 'unclean':
            Label \&= 'FPC'
        if fileclass $==$ 'clean':
            label \&= 'FPU'
    if (return\_Ustate $==$ True or return\_Cstate $==$
        $ True):
        if fileclass $==$ 'unclean':
            Label \&= 'FNC'
        if fileclass $==$ 'clean':
            label \&= 'FNU'

    return Label

def return\_percentage(percentage):
    if percentage $\leq$ 10:
        num \&= 0
    elif percentage $\geq$ 90:
        num \&= 1
    elif percentage $\geq$ 50 or percentage $\leq$
        89:
        num \&= 2
```

---

```

else:
    num \&= 3

return num

def \_ \_ Operation \_ \_ (Label, return \_ Crules, return \_ Urules,
    return \_ gen \_ CRule, return \_ gen \_ URule, curr \_ file):

    """ Clean files Stuff """
    clean \_ data \&= 'clean \_ rule \_ database.txt'
    update \_ clean \&= datafile(curr \_ file)
    Add \_ clean \_ file \&= open( clean \_ data , "a")
    filename \_ C \&= 'Updated \_ C \_ data.txt'
    Rem \_ clean \_ file \&= open(filename \_ C, 'w')
    Rem \_ list \_ clean \_ 1 \&= []
    Rem \_ list \_ clean \_ 2 \&= []

    """ Unclean files Stuff """
    unclean \_ data \&= 'unclean \_ rule \_ database.txt'
    update \_ unclean \&= datafile(curr \_ file)
    Add \_ unclean \_ file \&= open( unclean \_ data , "a")
    filename \_ U \&= 'Updated \_ U \_ data.txt'
    Rem \_ unclean \_ file \&= open( filename \_ U , "w")
    Rem \_ list \_ unclean \_ 1 \&= []
    Rem \_ list \_ unclean \_ 2 \&= []

    if Label $==$ 'TP':
        statement \&= 'go to nxt file'

    elif Label $==$ 'TN':
        statement \&= 'go to nxt file'

    elif Label $==$ 'FPU':
        statement \&= 'Removed'
        for \_ rem \_ rule \_ in return \_ Urules:
            Rem \_ list \_ unclean \_ 1.
            applicationend(\_ rem \_ rule \_)
        for newR in return \_ gen \_ URule:

```

```
        Rem\_list\_unclean\_2.  
        applicationend(newR)  
for x in Rem\_list\_unclean\_1:  
    for y in Rem\_list\_unclean\_2:  
        if x not in y:  
            Rem\_unclean\_file  
            .write(y)  
        break  
  
elif Label $==$ 'FPC':  
    statement \&= 'Removed'  
    for \_rem\_rule\_ in return\_Crules:  
        Rem\_list\_clean\_1.applicationend  
        (\_rem\_rule\_)  
    for newR in return\_gen\_CRule:  
        Rem\_list\_clean\_2.applicationend  
        (newR)  
    for x in Rem\_list\_clean\_1:  
        for y in Rem\_list\_clean\_2:  
            if x not in y:  
                Rem\_clean\_file.  
                write(y)  
            break  
  
elif Label $==$ 'FNC':  
    statement \&= 'Added'  
  
    for \_\_additem\_\_ in update\_clean:  
        if \_\_additem\_\_ !\&= return\  
        \_gen\_CRule:  
            Add\_clean\_file.write(\_  
            \_additem\_)\_)  
  
elif Label $==$ 'FNU':  
    statement \&= 'Added'  
  
    for \_\_additem\_\_ in update\_unclean:  
        if \_\_additem\_\_ !\&= return\  
        \_gen\_URule:
```

---

```
Add\_unclean\_file.write(\
\_additem\_)
```

```
return statement
```

```
def main():
```

```
clean\_rule\_data \&= 'clean\_rule\_database.txt '  
unclean\_rule\_data \&= 'unclean\_rule\_database.  
txt '  
curr\_file \&= '/home/khizer/Machine Learning  
Algorithms/Amazon Data/Data/unclean\_rule1.txt '
```

```
\#os.chdir("/home/khizer/Machine Learning  
Algorithms/Amazon Data")
```

```
\#for file in os.listdir("/home/khizer/Machine  
Learning Algorithms/Amazon Data/Data"):
```

```
if curr\_file.endswith(".txt"):  
fileclass \&= 'unclean'
```

```
elif curr\_file.endswith(".csv"):  
fileclass \&= 'clean'
```

```
infile\_clean \&= datafile(clean\_rule\_data)  
infile\_unclean \&= datafile(unclean\_rule\_data)
```

```
infile\_for\_clean \&= datafile(curr\_file)  
infile\_for\_unclean \&= datafile(curr\_file)
```

```
""" Search for the match in their respective  
databases """
```

```
return\_Cstate, return\_Crules, return\_gen\_CRule  
\&= search\_clean(infile\_clean, infile\_for\_  
_clean)
```

```
return\_Ustate, return\_Urules, return\_gen\_URule  
\&= search\_unclean(infile\_unclean, infile\_for  
\_unclean)
```

```
""" Return the Verdict """
```

```

returned\_label \&= Tester(fileclass , return\_
    \_Cstate , return\_Ustate)
print returned\_label

""" Implementing the Verdict """
return\_statement \&= \_\_Operation\_\_(returned\_
    \_label , return\_Crules , return\_URules , return\_
    \_gen\_CRule , return\_gen\_URule , curr\_file)
print return\_statement

```

```

if \_\_name\_\_ $==$ "\_\_main\_\_":
    main()

```

\$Appendix N: Search\_Filter.py

```

from stopwords import stopwords_filter

def Add_Data(Cdata , words):
    state \&= False
    new_data \&= 'pre_rules_database.txt'
    Add_clean_file \&= open( new_data , "a")
    for nwords in words:
        for cwords in Cdata:
            if nwords not in cwords:
                Add_clean_file.write(str(
                    nwords.split()))
                state \&= True
            else:
                state \&= False

    return state

def search_data(Dataset , words):

```

---

```

state \&= False
list_1 \&= []
list_2 \&= []
Message_2 \&= []
msg_out \&= "Initializing ..."
for data in Dataset:
    list_1.applicationend(data)
for keys in words:
    list_2.applicationend(keys)
\#print words
for k_comp in list_2:
    for d_comp in list_1:
        if k_comp.lower() in d_comp.lower():
            state \&= True
            Message_2.applicationend(k_comp)
            msg_out \&= 'The word %s is infamous , try
                using different word' % str(Message_2)

return state , msg_out

```

```
def Run_Filter(Uinput):
```

```

\#state \&= False
unclean_rule_data \&= '/home/rmhminhas/Assoc_Filter/
    bad_words.txt '
infile_unclean \&= open( unclean_rule_data , "r")
clean_data \&= '/home/rmhminhas/Assoc_Filter/
    clean_rule_database.txt '
infile_clean \&= open( clean_data , "r")
UserInput \&= Uinput.split()
getwords \&= stopwords_filter(UserInput)
state , message \&= search_data(infile_unclean , getwords)

if state $==$ True:
    print 'Disapplicationroved '
    final_message \&= message
    nstate \&= True

```

## 7. APPENDIX

---

```
elif state==$ False:
    final_message \&= "Applicationroved"
    nstate \&= False
    dstate \&= Add_Data(infile_clean ,getwords)
    if dstate==$ True:
        print "The word is Added to the
            database"++
    elif dstate==$ False:
        print "The word already exists in the database
            "
    return nstate , final_message

if \_name\_==$ "\_main\_":

    while True:
        Uinput \&= raw\_input('Enter the string of words :
            ')
        a,b\&=Run\_Filter(Uinput)
        print a,b

$
```

### Appendix O: Stopwords.py

```
def datafile(name):
    ifile==$ open (name,'rU')
    for line in ifile:
        ret==$ frozenset(line.split())
        yield ret

get\_words==$ []
def stopwords\_filter(user\_input):
    arr1==$[]
    arr2==$[]
    Dataset==$ [line.strip() for line in open('/home/
        rmhminhas/Assoc\_Filter/english.txt', 'r')]
    for data in Dataset:
        arr1.applicationend(data)
```



---

```

for data\_2 in user\_input:
    arr2.applicationend(data\_2)

for x in arr1:
    for y in arr2:
        if y $==$ x:
            arr2.remove(x)

return arr2

def main():
    while True:
        user\_input $=$ raw\_input('enter ')
        \_\_UserInput\_\_ $=$ user\_input.split()
        getwords $=$ stopwords\_filter(\_\_UserInput\_\_)
        print getwords

if \_name\_ $==$ "\_main\_":
    main()

```

Appendix P: assoc.py

```

from itertools import chain, combinations          \# for
    combining datasets
from collections import defaultdict              \#
    library which will be used for counting occurrence
from optparse import OptionParser              \#
    powerful library for parsing command-line options

def subsets(arr):
    """ Returns non empty subsets of arr """
    return chain(*[combinations(arr, i + 1) for i, a in
        enumerate(arr)])

```

```
def testing\_minsupport(itemSet, transactionList,
    minSupport, freqSet):

    \_\_itemSet\_\_ $=$ set()
    localSet $=$ defaultdict(int)

    for item in itemSet:
        for transaction in transactionList:
            if item.issubset(transaction):
                freqSet[item] += $ 1
                localSet[item] += $ 1

    for item, count in localSet.items():
        support $=$ float(count)/len(transactionList)

        if support >=$ minSupport:
            \_\_itemSet\_\_.add(item)

    return \_\_itemSet\_\_

def joinSet(itemSet, length):
    """Join datasets """
    return set([i.union(j) for i in itemSet for j in
        itemSet if len(i.union(j))==$ length])

def Apriori(data, minSupport):
    transactionList $=$ list()
    itemSet $=$ set()
    for record in data:
        transaction $=$ frozenset(record)
        transactionList.append(transaction)
        for item in transaction:
            itemSet.add(frozenset([item]))        \# Generate
            1-itemSets

    freqSet $=$ defaultdict(int)
```

---

```

largeSet      $$ dict()          \# Global
              dictionary which stores (key=$n-itemSets ,value=$
              $support) which satisfy minSupport

initial\_set   $$ testing\_minsupport(itemSet ,
              transactionList , minSupport, freqSet)

current\_data\_Set $$ initial\_set
k $$ 2
while(current\_data\_Set !=$ set ( [] ) ):
    largeSet[k-1]  $$ current\_data\_Set
    current\_data\_Set  $$ joinSet(current\_data\_
    _Set,k)
    updated\_set      $$ testing\_minsupport(current\_
    _data\_Set, transactionList , minSupport,
    freqSet)
    current\_data\_Set  $$ updated\_set
    k $$ k + 1

genRules=$$ []
for key,value in largeSet.items() [1:]:
    for item in value:
        \_subsets $$ frozenset(x for x in subsets(
        item))
        for element in \_subsets:
            remain $$ item.difference(element)
            if len(remain)>0:
                genRules.applicationend(( list (element)
                ,list (remain)))

return genRules

def printResults(rules):
    """prints the generated rules in a file :"""
    output $$ []
    for rule , extra in rules:

```

## 7. APPENDIX

---

```
        mal\_liklihood\_1, mal\_liklihood\_2 $=$ rule ,
        extra
        finalfile $=$ " %s ----> %s " % (str(mal\_liklihood\_1), str(mal\_liklihood\_2))
        \# text\_file.write(finalfile)
        \# text\_file.write('')
        output.applicationend(finalfile)
return ''.join(output)

def datafile(name):
    ifile $=$ open (name)
    for line in ifile:
        ret $=$ frozenset(line.split())
        yield ret

if \_\_name\_\_ $==$ "\_\_main\_\_":
    """
    Parsing command-line options
    """

    optparser $=$ OptionParser()
    optparser.add\_option('-f', '--inputFile', dest $=$ 'input', help $=$ 'filename', default$=$None)
    optparser.add\_option('-s', '--minSupport', dest$=$'minS', help $=$ 'minimum support value', default$=$0, type$=$'float')

    (options, args) $=$ optparser.parse\_args()

    inFile $=$ options.input

    if inFile is not None:
        inFile $=$ datafile(options.input)

    minSupport $=$ options.minS
    rules $=$ Apriori(inFile, minSupport)
```

---

```
\#filename $=$ raw\_input('Enter the file name :' )
                                \# Ask for the file name
\#text\_file $=$ open(filename,'a')

printResults(rules)                \# An
  output method
```

.....