Mini Project Report On

Recommender System for E-commerce portals.

Submission By:
Gaurav Gupta (RIT2012018)
Harshit Gupta (RIT2012048)
Kamal Soni (RIT2012076)
Nakshatra Maheshwari (RIT2012074)
Shashank Sharma (RIT2012075)
Shweta Choudhary (RIT2012025)

Under Guidance of
Dr. O.P. Vyas

Indian Institute of Information Technology, Allahabad

Jan - May 2015

CANDIDATE'S DECLARATION

We hereby declare that the project work entitled "**Comparison Shopping Trend Analyzer**" submitted to the IIIT Allahabad as 6th semester project, is a record of an original work done by us under the guidance of Dr. OP Vyas and has not been submitted to any other University or Institute or published earlier.

Place:
Date:

Gaurav Gupta (RIT2012018)
Harshit Gupta (RIT2012048)
Kamal Soni (RIT2012076)
Nakshatra Maheshwari (RIT2012074)
Shashank Sharma (RIT2012075)
Shweta Choudhary (RIT2012025)

CERTIFICATE FROM THE MENTOR

It is certified that this project report "**Comparison Shopping Trend Analyzer"** of mini project taken in 6th semester is the bona fide work of "Gaurav Gupta (RIT2012018), Harshit Gupta (RIT2012048), Kamal Soni (RIT2012076), Nakshatra Maheshwari (RIT2012074), Shashank Sharma (RIT2012075), Shweta Choudhary (RIT2012025)" who carried out the project work under my supervision.

Dr. O.P. Vyas

ACKNOWLEDGEMENTS

If words are considered as a symbol of approval and token of appreciation then let the words play the heralding role expressing my gratitude.

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project mentor Mr. O P Vyas for the guidance, inspiration and constructive suggestions that helpful us in the preparation of this project. We also thank our colleagues who have helped in our mini-project.

Place:
Date:

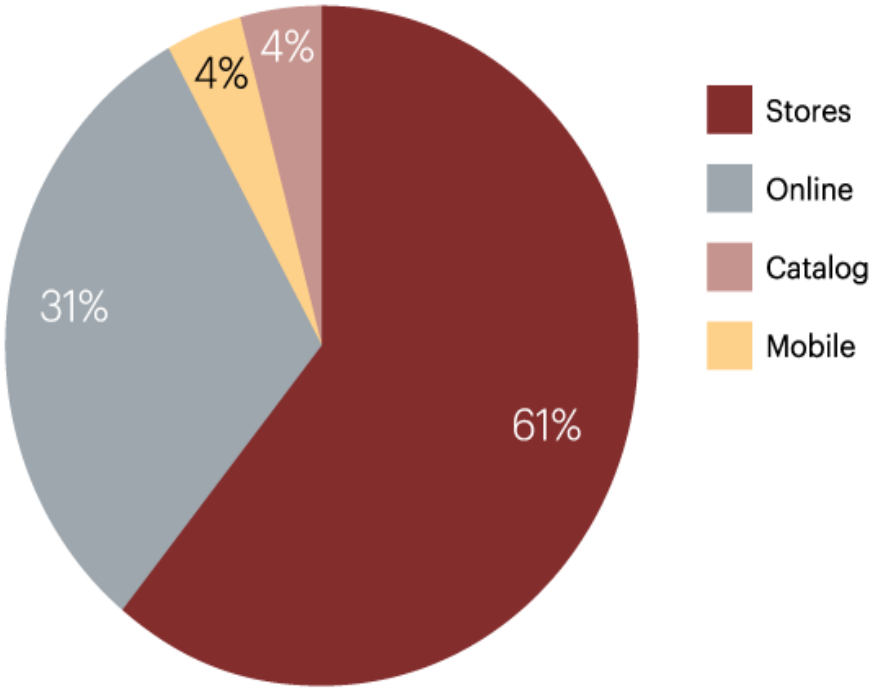
Gaurav Gupta (RIT2012018)
Harshit Gupta (RIT2012048)
Kamal Soni (RIT2012076)
Nakshatra Maheshwari (RIT2012074)
Shashank Sharma (RIT2012075)
Shweta Choudhary (RIT2012025)

INDEX

1. INTRODUCTION

(a) Overview [1]:

- Price comparison have mushroomed over recent years and are now seen by many as a tool of consumer empowerment.
- Today's customer are shifting from brick and mortar shops to e-commerce.
- It has revolutionized the concept of modern business and changed the idea of locating on a certain territory and focus on a particular consumer audience.
- So there is definitely a must need to work on this system to provide customers with the best services and Present a unique opportunity for business to treat the market as a "conversation" between business and customers instead of traditional business-to-customer one way marketing.



Source: A.T. Kearney Future of Stores study, 2013

(b) Objective : *

a. To provide e-commerce customers a common platform where they can search for the best product they want to buy from various options (flipkart, amazon, ebay, snapdeal, etc.) .
b. How to select this best product?
c. Can we develop a system that can act as “intermediate” between the buyer and seller (various sites) and assists the buyer in selecting this best product based on buyer’s choice of preferences?
d. By this project- yes, you can!

* Only for Smart-Phones and Smart-TV's.

(c) Why this objective? :

Most people try to get best product under their affordable range irrespective of number of features.

2. Literature Survey

1. **Related Work done so far in this area** :-
    - Various comparison based websites are already available:-
        - Pricedekho.com
        - Junglee.com (An initiative of amazon).
        - Buyhatke.com
        - Mysmartprice.com
    - **Pros** :-
        - Price based comparisons.
        - Real-time updated data.
        - Data Clustering.
        - Specification Comparison.
    - **Cons** :-
        - Poor Keyword based searching :-
            1. To search for "**Lenovo A9**" shows following result.
                a. **Lenovo A9**
                b. **Lenovo A10** Quad Core processor Cortex **A9**
        - Different product in search results have similar names.
        - **Lack of Recommendations for Cold Start User.**

2. **Why our project :-**
    - Recommendation for Cold Start User.
    - Unrated products also gets pseudo ratings.
    - Searching and filtering is better than traditional Keyword based.

3. **Recommender System – An Ecommerce game changer**[1] :-

    - Recommender systems are an important part of the information and e-commerce ecosystem. They represent a powerful method for enabling users to filter through large information and product spaces.
    - With the increasing usage of the web, the daily life of user is revolving around the information available online. The utmost

need of hour is to surpass the invalid information and highlight the valid data.

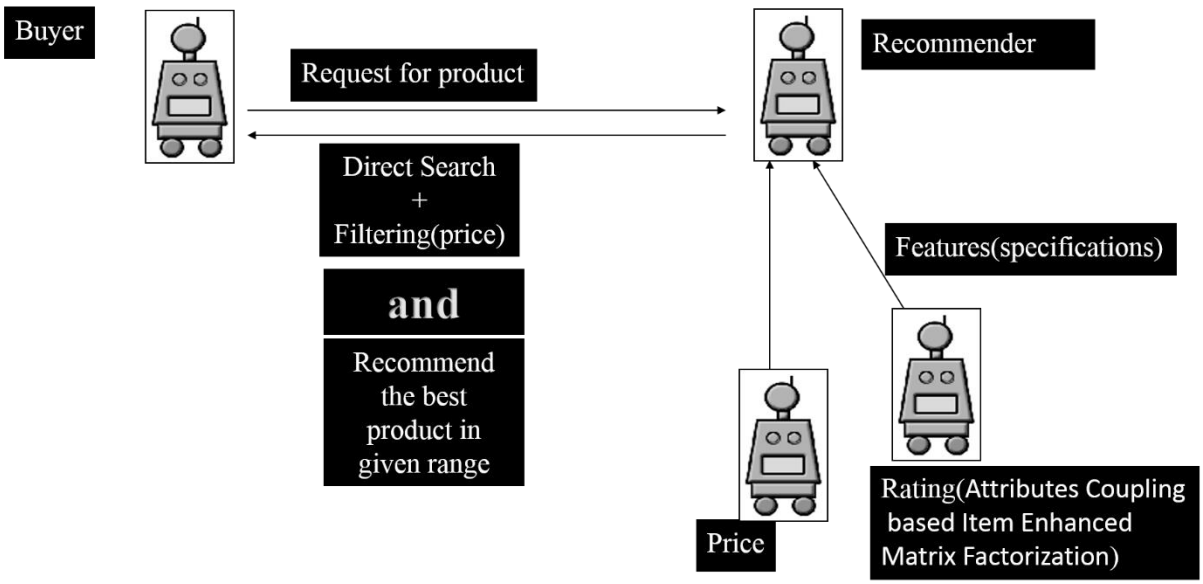
- The prediction algorithms then are of huge importance to online stores - the more accurate they are, the more the online store will sell.
- Types:
  - Collaborative Based Recommender Systems.
  - Content Based Recommender Systems.
- **Why is actually Recommendation System needed here**[1] :-
  - Electronic retailers and content providers offer a huge selection of products.
  - The users do not have to waste time searching for appropriate items.
  - Matching consumers with the most appropriate products is key to enhancing user satisfaction and loyalty.
  - Therefore, more retailers have become **interested in recommender systems,** which analyze patterns of user interest in products to provide personalized recommendations that suit a user's taste.
  - Good personalized recommendations can add another dimension to the user experience, enhancing the e-trade.
- **Collaborative Recommender System**[1] :-
  - This method is based on previous experience of users.
  - It requires user's history.
  - Algorithms :
    1. Kth Nearest Neighbor
    2. Pearson Correlation
  - Drawbacks:
    3. Cold Start Case is not handled
    4. Scalability is not feasible.
  - Examples:
    1. Last.fm
    2. Facebook
    3. LinkedIn
    4. Twitter

- **Content Based Recommender System** [3]:-
  - Creates a profile for each user or product to characterize its nature.
    1. Movie profile include attributes regarding its **genre, the participating actors, it's box office popularity.**
    2. User profile might include **demographic** information or **answers** provided on a suitable questionnaire.
  - Programs use these profiles associate users with matching products.
  - Algorithms:
    1. TF-IDF algorithm for calculating vector of weight
    2. Bayesian's Classifiers
    3. Cluster analysis
    4. Decision trees
    5. Artificial Neural Networks
  - Drawbacks:
    1. Based on Specific attributes cases like News may be useful but for videos it might not be useful.
  - Examples:
    1. Pandora Radio
    2. Internet Movie Database
- **Why we reject Collaborative approach and accept content based?** :-
  - Initially, every user who visits our platform will be cold start user so we need something which handles this problem.
  - If a product is at very high rating from last 2 years (say) then comparing it with brand new product with better features based on rating becomes ambiguous.
  - If the weights for different attribute assumed by our proposed algorithm becomes absolute, then our application can be trained (Machine learning).
  - The components can have binary, nominal or numerical attributes and are derived from either the content of the items or from information about the users' preferences. The task of the learning method is to select a function based on

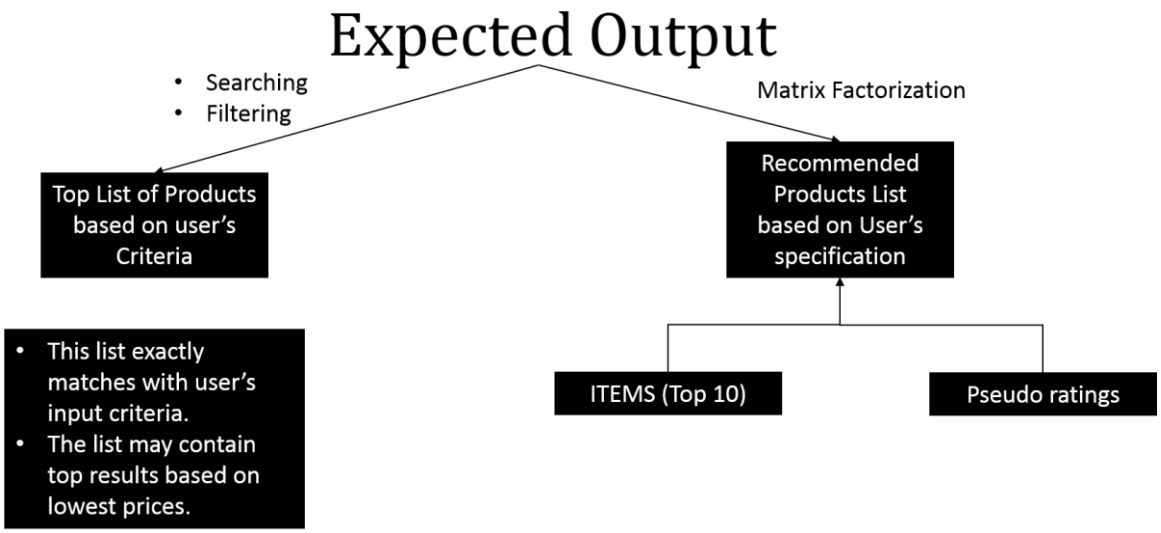
a training set of *m* input vectors that can classify any item in the collection.

3.Project Model and Methodology

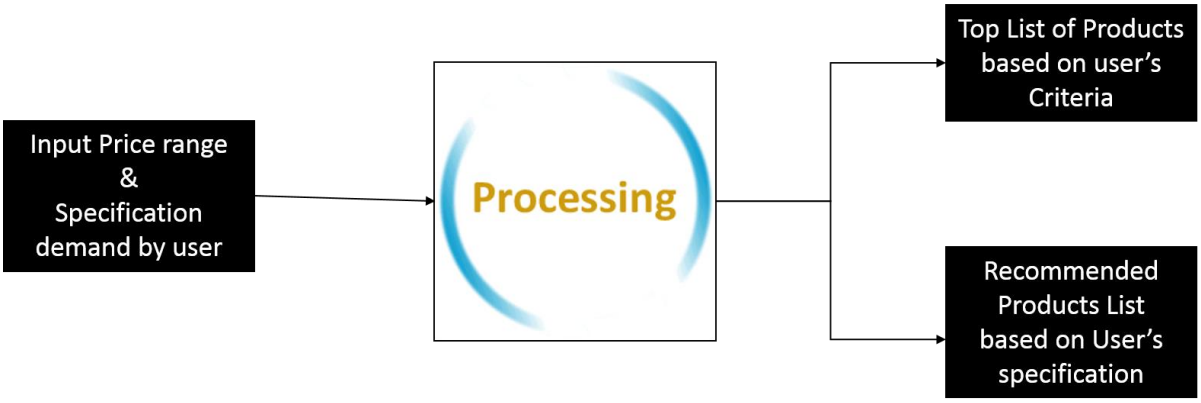
➢ **Project Model** :-



➢ **Expected Output :-**



- **Methodology :-**
  - **a. Overview :-**



Based on specification & ratings comparison of different models in that Price Range we will try to predict the best products.

Steps of Methodology

(a) Extraction of Data :

To extract data from various websites we used scrapy library of python to crawl through the resulting HTML pages on client side and extract information based on pattern of their HTML tags.

(b) Removing Noise :

Noise is some unwanted data. For example when collecting data via scrapy library, we may get some unwanted strings inside price.
Data Cleansing helps to resolve this problem.

- Let websites use the following variation in names for the product:
  - HTC One X+ (Black);
  - HTC One X+ Black;
  - HTC One X Plus;
  - HTC One X Plus, black;
- **Approaches in Data Cleaning :-**
  - Ignore everything that is in parenthesis.

- Breaking down into cases based on manufacturer, model number, etc.
- Define words automatically drop like "black", "blue" or "white".
- Make a dictionary of feature set (manufacturer, model number) for different catalog which should be same while declaring any product as common product.
- Compare the names via their Levenshtein distance [2] and use this distance for clustering.
- Heuristic Levenshtein distance [5].
- If scraped data has n entries, then overall complexity= ($O(n^2l^2)$)
- Where 1= product title length.

- **Heuristic Levenshtein distance[5] :-**
  - By considering Levenshtein distance between dictionary product title and the products' title we scraped, all 3 operations are considered:
  - 1:) DELETE OPERATION
  - 2:) SUBSTITUTION OPERATION
  - 3:) INSERT OPERATION
  - The optimal solution count consists of all the counts of respective operations.
  - If we consider only SUBSTITUTION OPERATION count from the optimal solution, then count comes nearly equal to 0.
  - Then we predict them similar.
  - Accuracy more than 50%

(c) Normalize the Schema for storing the above clean data in database :-

Products features and name remains same, its only price and ratings that varies with time.

(d) Create Adjacency Matrix of Product Ratings and use Matrix Factorization to get missing Values (Matrix Factorization):

Matrix factorization can be used to discover latent features underlying the interactions between two different kinds of entities. (Of course, you can consider more than two kinds of entities and you will be dealing

with tensor factorization, which would be more complicated.) And one obvious application is to predict ratings in collaborative filtering.

- **Algorithm:-**

i. Map both users and items to a joint latent factor space of dimensionality f .

$$\hat{r}_{ui} = q_i^T p_u.$$

ii. User-item interactions are modeled as inner products in that space
iii. Each item I is associated with a vector qi ,and each user u is associated with a vector pu,
iv. qi measures the extent to which the item possesses those factors
v. pu measure the extent of interest the user has in items
vi. the resulting dot product qipu captures the interaction between user u and item I - the user's overall interest in the item's characteristics
vii. The approximates user u's rating of item i, which is denoted by rui, leading to the estimate:
viii. Adding Biases :-
    a. However, typical collaborative filtering data exhibits large systematic
    b. tendencies for some users to give higher ratings than others
    c. And for some items to receive higher ratings than others
        i. Some products are widely perceived as better(or worse) than others
    d. It's unwise to explain the full rating value in this form.
    e. We should identify the user and item bias.
ix. A first-order approximation of the bias is as follows:

$$b_{ui} = \mu + b_i + b_u$$

x. The bias accounts for the user and item effects, where u denotes overall average rating and bu and bi indicates observed deviations of user u and item i.
xi. Example:
    a. Suppose we want to estimate user john's rating of the movie Titanic
    b. And the average rating over all movies is 3.7 stars

c. Titanic is better than an average movie, so it tends to be rated 0.5 starts above the average movie
d. John is a critical user, who tends to rate 0.3 stars lower than the average
e. Thus, the estimate for Titanic's rating by John would be (3.7+0.5-0.3).

xii. The estimate becomes :

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

xiii. Parameterizing the model :-

a. Rating scale of individual users
b. Ranking of individual items
c. User preferences

$$r_{ui}(t) = \mu + b_u(t) + b_i(t) + q_i^T p_u(t)$$

- **Attributes coupling based Item Enhanced Matrix Factorization Technique [3].**
  a. Matrix factorization technique is one of the most widely employed collaborative filtering techniques in the research of recommender systems due to its effectiveness and efficiency in dealing with very large user-item rating matrices.
  b. However, the majority focus on dealing with the cold start user problem and ignore the cold start item problem.
  c. In addition, there are few suitable similarity measures for these content enhanced matrix factorization approaches to compute the similarity between categorical items.
  d. Required attributes coupling based item enhanced matrix factorization method by incorporating item attribute information into matrix factorization technique as well as adapting the coupled object similarity to capture the relationship between items
  e. Item attribute information is formed as an item relationship regularization term to regularize the process of matrix factorization.
  f. Specifically, the similarity between items is measured by the Coupled Object Similarity considering coupling between items.

- **Ranking of individual items($b_i(t)$)** :-
    a. Antutu Benchmark [6].
    b. Ranking on the basis of specification score based on our proposed algorithm.
        i. Different features are given different weightage.
        ii. For each product, score is calculated for each feature on account of weight.
        iii. On the basis of Overall score, relationship between items is justified.
- **Antutu Benchmark [6]** :-
    a. Antutu comprehensively test certain aspects of a device, including UX, GPU, RAM, CPU, I/O. Each item is individually assessed and given a score and then used to rank the device among all other devices.
    b. Scores and rankings are easily available online.
    c. Cons :
        i. Compute score only on the basis of certain features.
        ii. Rest features remain intact (Supports memory card, NFC, etc.).

➢ Ranking of Individual Items (Feature Based) :-

Feature	Type	Weight
Faster CPU	%	
Octa core CPU	%	
More RAM	%	
More Storage Capacity	%	
Has Dual Sim	B	
Longer Battery Life	%	
Better Screen resolution	%	
Better Camera resolution	%	
Bigger Battery	%	
Support memory card	B	
Support FM radio	B	
Support HDMI	B	
Has removable battery	B	
Longer battery standby time	%	
Bigger screen	%	
Light weight	%	
Slimmer	%	
Support NFC	B	

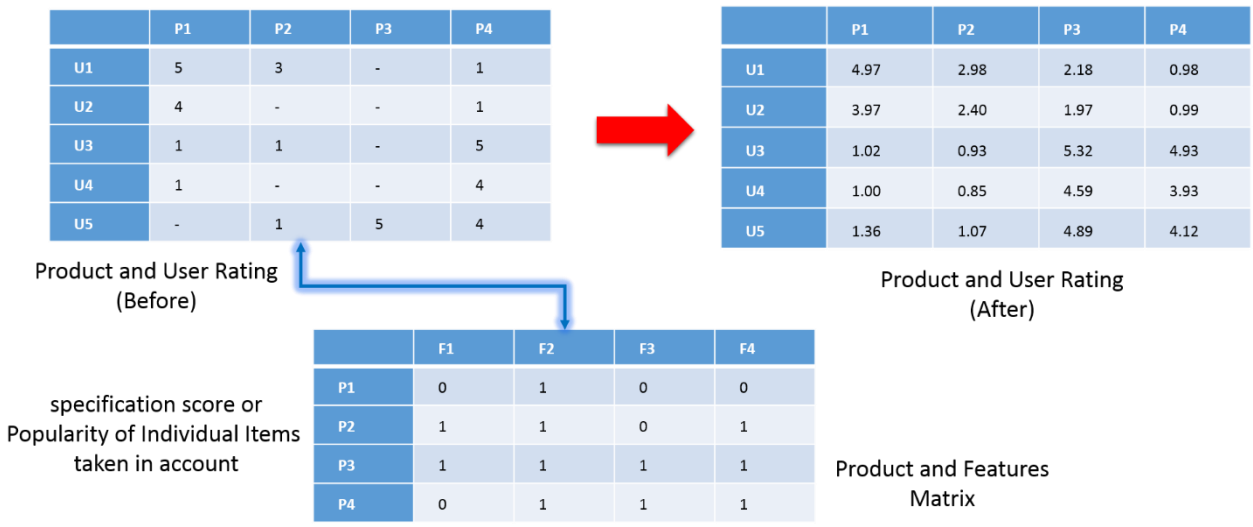
#For mobiles

Where,

- % = percentage difference of that feature between products
  - Ex: For, Faster CPU, feature_value(X) :
    - Galaxy S6 : 1.5 Ghz
    - Galaxy S4 : 1.2 Ghz
  - % = Galaxy S6 has 80% faster CPU than Galaxy S4
- And B = binary
  - Ex: Has Dual Sim:
  - Galaxy S6 : Yes(1)
  - Galaxy S4 : No(0)

- **Formulae used** :-
  a. **Score (for %), P1 = Σweight * (feature_value(X) / feature_max)**
  b. **Score (for B), P2 = Σweight * (Feature_value(X))**
  c. **Final Score(X) = P1 + P2**
  d. **Here, X = Specific product.**
  e. **feature_max = Upper bound of the feature value stored in Dictionary**.
  f. How to distribute weight??
  g. Known factor: Core features must be given more weight.

- **Working module** (Flow charts) :-



Flowchart: 1

Say, P4 not available
or not rated in Flipkart

	P1	P2	P3	P4
U11	4.97	2.98	2.18	--
U12	3.97	2.40	1.97	--
U13	1.02	0.93	5.32	--
U14	1.00	0.85	4.59	--
U15	1.36	1.07	4.89	--

Matrix from Flipkart(say)

+

	P1	P2	P3	P4
U21	4.97	2.98	2.18	0.98
U22	3.97	2.40	1.97	0.99
U23	1.02	0.93	5.32	4.93
U24	1.00	0.85	4.59	3.93
U25	1.36	1.07	4.89	4.12

Matrix from Amazon(say)

=

	P1	P2	P3	P4
Flipkart	2.46	1.65	3.79	--
Amazon	2.46	1.65	3.79	2.99

Flowchart: 2

	P1	P2	P3	P4
Flipkart	2.46	1.65	3.79	--
Amazon	2.46	1.65	3.79	2.99

→

	P1	P2	P3	P4
Flipkart	2.46	1.65	3.79	1.50
Amazon	2.46	1.65	3.79	2.00

Solved case of *Cold Start*

Flowchart: 3

(e) Select Best product as one with maximum ratings :

Now since we have ratings for all products sum it up and the product with maximum ratings can be considered the best.

4. Hardware and Software Requirements

(a) Hardware Requirements:

None

(b) Software Requirements:

- Python Compiler
- PIP
- Scrappy Library,
- Selenium Webdriver (for multipages scroll and AJAX Websites)
- MYSQL (to store scrapped values),
- Notepad++ (Any IDE will be suitable), Dev C++ (gcc based IDE).
- Virtual Box and Ubuntu 14.04, LAMP (Linux Apache Mysql PHP) -Server for web hosting.

5. Activity Time Chart

Time	Activity
January	Learning about Data Collection, python Language, php language.
February	Learning about Data Cleansing, Filling of missing ratings by Matrix Factorization, regular Expressions to remove noise.
March	Implementation of scrapy, removing noise from data using regulat expressions/Data Cleansing, matrix factorization and finally php based User Inerface to Query with MySQL database.
April	More Literature Survey on Different forms of Recommender System and Matrix Factorization. **Discussion on this idea and algorithms with Mr. Abhishek Kumar (Assistant of Dr. Artus, University of Paderborn, Germany).**
May	Implementation of all Suggestions mentioned mid-semester evaluation and also design and algorithm implementation for Recommender systems (Algorithms + Backend + GUI).

(a) Work done before mid-semester:

a. Currently we are in success to extract data from websites

b. Removing of the noise.

c. Successfully apply matrix factorization on ratings*.

d. PHP based UI for User Query.

e. VB Scripts to crawl data at regular intervals.

(b) Work done after Mid-Semester:

a. Better Extraction of data and Prices **(Cart Prices including delivery charges)**.

b. Implementation of Better Matrix Factorization for Recommender Systems (includes feature matrix into account).

c. Better Handling of large Database **(Normalization of schema up to 3rd Normal Form).**

d. More secure backend and user interface (authorization of user, resistant to SQL injection attacks and penetration attacks).

*** Ratings of product is extracted as overall rating per website as it was not possible to extract individual users' rating for every product and adjacency is created for websites vs Product.**

6. Limitations

- In this dynamic world, websites keeps changing. So scraping scripts needs to be changed.
- Large database required (Normalization needed).
- Data Noise can never be completely removed.
- Large Scale catalog requires large computation time.
    - Our Algorithms may struck in hours' computation.

7. Refrences

- **[1] Hsinchun, Roger, Veda. 2012. "Business Intelligence and analytics: From Big data to Big Impact," MIS Quaterly Vol. 36 No. 4, pp. 1**
- **[2] Joeran, Stefan, Marcel, Bela, Corinna, Andreas. 2013. "Recommender System Evaluation: A Quantitative Literature Survey," ACM International Conference.**
- **[3] Jennifer, Mu Zhu. 2013. "Content Boosted Matrix Factorization Technique for Recommender Systems," Cornell University.**
- **[4] Matrix factorization techniques for recommender systems, Yehuda Koren, 2009.**
- **[5] (http://en.wikipedia.org/wiki/Levenshtein_distance)**
- **[6] (http://www.antutu.com/en/Ranking.shtml)**

8. Remarks