

Recommender System based on Customer Behaviour for Retail Stores

G. Krishna Kishore¹, D. Suresh Babu²

¹(Department of Computer Science Engineering, Velagapudi Rama Krishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India)

²(Department of Computer Science Engineering, Velagapudi Rama Krishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India)

Abstract: In today's fast moving world, shopping has become increasingly online. The benefits provided by online shopping outperform the need to shop in person. One such enticing benefit is the personalised recommendations provided specifically to each user. These recommendations guide the users in their shopping process and also unveil new range of products suiting their tastes. This system we intend to develop stores the user's purchase and rating data in a backend database. It also includes a smart beacon that connects to the offline shopper's mobile phone and gains data about the particular customer from backend database. First, users are clustered based on their demographic data. Recommender system employed analyses each cluster and finds out top rated items in each cluster. It then sends out customised offers for a particular customer. For our system, we wish to use a Hybrid recommender system as it avoids the cold start and scalability inherent with the other models of recommender systems .

Keywords: Clustering, Cold start problem, Demographic Data, Hybrid recommender systems, Recommender system

I. Introduction

A major challenge for large retailers is to address the needs of the consumers more effectively on a local level, while maintaining the efficiencies of central distribution. As the demand for mass customization by consumers grows, methods focused on store level optimization increase in value. The recent development in information technology and ubiquitous computing makes it feasible to move recommender systems from the ecommerce realm, where they are widely used, to retail stores

1.1 Nature of the Problem.

The world is moving towards online shopping due to the benefits it provides to the customers. Big e-commerce sites like Amazon, Flipkart etc. provide tailored recommendations to their customers based on their previous behaviour. These aid the customers during their shopping and provide satisfactory shopping experience. These recommendations are given by recommender systems employed by the sites. As these systems evolve and become more complex, more accurate predictions and recommendations are given to the users. If this continues, retail business might wane in the future. Hence, a system which uses cash receipt data to provide personalized recommendations to customers can help provide business advantage to the retail stores

1.2 Purpose

Main objective is

- To provide the customer a personalized experience.
- To build customer's trust and loyalty.
- To provide a better marketing strategy for offline retailers.

II. Literature Review

Recommender systems can now be found in many modern applications that expose the user to a huge collection of items. Such systems typically provide the user with a list of recommended items they might prefer, or predict how much they might prefer each item. These systems help users to decide on appropriate items, and ease the task of finding preferred items in the collection For example, the DVD rental provider Netflix displays ratings for every displayed movie in order to help the user decide which movie to rent. The online book retailer Amazon provides average user ratings for displayed books, and a list of other books that are bought by users who buy a specific book. Microsoft provides many free downloads for users, such as bug fixes, products and so forth. When a user downloads some software, the system presents a list of additional items that are downloaded together. All these systems are typically categorized as recommender systems, even though they provide diverse

services. In the past decade there has been a vast amount of research in the field of recommender systems, mostly focusing on designing new algorithms for recommendations. An application designer who wishes to add a recommendation system to her application has a large variety of algorithms at her disposal [1].

2.1 Different Approaches to Recommender Systems

The data captured is used by the recommender system to eventually provide the recommendations to the customer. Recommender systems can be present in all sorts of systems and situations, and thus can be implemented in many different ways [2].

There are six categories of recommendation approaches:

1. Non-personalized
2. Demographic-based
3. Collaborative filtering
4. Content-based
5. Knowledge-based
6. Hybrid

For our system, we wish to use a Hybrid recommender system as it avoids the cold start and sparsity problems inherent with the other models of recommender systems. A hybrid recommender system is one that combines multiple techniques together to achieve some synergy between them. Netflix is a good example of the use of hybrid recommender systems.

2.2 Analysis of Hybrid Online Movie Recommender System [3]

Here a hybrid recommender system which provides movie recommendations based on content and collaborative filtering and also using context for better recommendations. The methodology to recommend movies has the following steps with explanation:

1. Initially the relevant features have to be gathered used in movie recommendation.
2. After collecting data also including rating data and contextual information, we apply advance algorithm for relevant recommendation.

Step 1: Collect user information

For the new user, the system requests to register him an account to gather his personal information and also the ratings of a set of movies. New User sign in Store user information Fetch ratings of movies Movie Database Gather movie information Recommendations Neighbour set Movie recommender User database

Step 2: Create movie database

Collect the movie information such as title, actors, director, release date, ratings etc. for recommendations.

Step3: Movie Recommender

This phase recommend movies to the user

2.3 Performance Analysis of Recommendation System Based On Collaborative Filtering and Demographics [4]

Here a proposed hybrid recommender system that combine prediction using item based collaborative filtering and demographic based user cluster in weighted scheme is used. To make system scalable, item similarity and user cluster are computed offline. To solve cold start problem, a new user is added to cluster with nearest centroid based on demographics. The system achieves lower MAE and higher coverage than traditional collaborative filtering algorithm.

2.4 A Movie Recommender System: MOVREC [5]

D.K. Yadav et al. introduced a movie recommendation system named MOVREC. It is based on collaborative filtering approaches that make use of information provided by users, analyses them and then recommends the movies that is best suited to user at that time. The 8 recommended movie lists is arranged according to ratings given to these movies by previous users and it uses K-means algorithm for this. The system is developed in PHP using Dreamweaver 6.0 and Apache Server 2.0. The proposed recommender system generates recommendation using data about users, the available items, and the previous transactions stored in customized databases. MovieREC allows a user to select the choices from a given set of attributes and then recommend a movie list based on the cumulative weight of different attributes and using K-means algorithm.

2.5 Clustering large Data Sets with Mixed Numeric and Categorical Values [6]

To overcome the problems in K-means while preserving its efficiency, K-prototypes were developed. Here a clustering algorithm is presented to solve data partition problems in data mining. The algorithm is based on the k-means paradigm but removes the numeric data only limitation whilst preserving its efficiency. The

algorithm clusters objects with numeric and categorical attributes in a way similar to k-means. Because objects are clustered against k prototypes instead of k means of clusters, it is called the k-prototypes algorithm. A method was developed to dynamically update the k prototypes in order to maximise the intra cluster similarity of objects. The object similarity measure is derived from both numeric and categorical attributes. When applied to numeric data the algorithm is identical to k-means. In testing with real data, this algorithm has demonstrated a capability of partitioning data sets in the range of a hundred thousand records, described by some 20 numeric and categorical attributes, into a hundred clusters in a couple of hours on a SUN Sparc 10 workstation.

2.6 Drawbacks of Existing Methods

- All the existing methods target online sites and customers online.
- No Definite System is developed for Customers who prefer to shop in physical Retail stores in person.
- K-means, while efficient at clustering, it works only for Numerical data. Thus, rendering it useless for categorical data or for mixed data types i.e., a mixture of both Numerical and Categorical data.

III. Proposed System

This chapter presents the detailed functionality of the proposed system.

3.1 Methodology

The system proposed, consists of three modules.

1. A device which acts as a hotspot and is used to create a private wireless network to its users. It also queries the database and pushes out specific offers to the user.
2. An app installed on the user's smartphone, connects to the store's private network via hotspot.
3. A simple database which stores the user's info connected to the store's network.

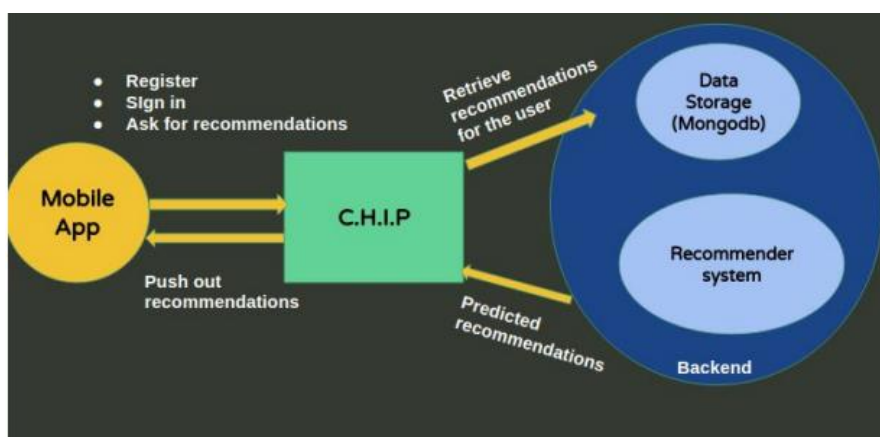


Fig. 3.1 Architecture of proposed system

3.1.1 Backend Process

A Recommender Engine/ System is developed to generate recommendations specific to an individual user and to store them in the Database.

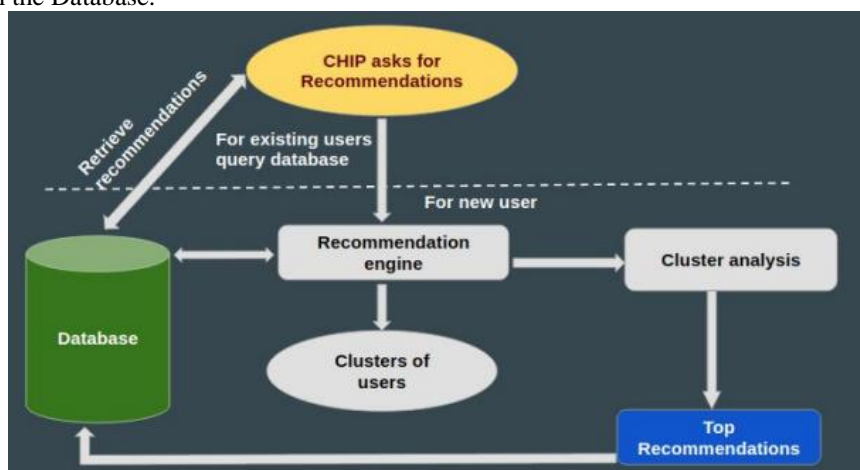


Fig. 3.2 Flow of process in Recommender System

3.1.1.1 Using K-Prototypes for clustering users

K-Prototypes algorithm is based on the k-means paradigm but removes the numeric data only limitation whilst preserving its efficiency. The algorithm clusters objects with numeric and categorical attributes in a way similar to k-means [6].

3.1.1.2 Mathematical Preliminaries

Let $X = \{X_1, X_2, \dots, X_n\}$ denote a set of n objects and $X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ be an object represented by m attribute values. Let k be a positive integer. The objective of clustering X is to find a partition which divides objects in X into k disjoint clusters. For a given n , the number of possible partitions is definite but extremely large. It is impractical to investigate every partition in order to find a better one for a classification problem. A common solution is to choose a clustering criterion to guide the search for a partition. A clustering criterion is called a cost function below.

3.1.1.3 Cost Function

The widely used cost function is the trace of the within cluster dispersion matrix. One way to define this cost function is

$$E = \sum_{l=1}^k \sum_{i=1}^n y_{il} d(X_i, Q_l) \quad (1)$$

Here, $Q_l = [q_{l1}, q_{l2}, \dots, q_{lm}]$ is the representative vector or prototype for cluster l , and y_{il} is an element of a partition matrix $Y_{n \times k}$

d is a similarity measure often defined as the square Euclidean distance.

Y has the following two properties, (1) $0 \leq y_{il} \leq 1$ and (2) $\sum_{l=1}^k y_{il} = 1$. Y is called a hard partition if $y_{il} \in \{0, 1\}$. Otherwise, it is a fuzzy partition. In a hard partition, $y_{il} = 1$ indicates that object X_i is assigned to cluster l by Y . We only consider hard partitions in this paper.

The inner term $E_l = \sum_{i=1}^n y_{il} d(X_i, Q_l)$ is the total cost of assigning X to cluster l , i.e., the total dispersion of objects in cluster l from its prototype Q_l .

E_l is minimised if

$$q_{lj} = (1/n_l) \sum_{i=1}^n y_{il} x_{ij} \quad \text{for } j=1, \dots, m \quad (2)$$

where $n_l = \sum_{i=1}^n y_{il}$ is the number of objects in cluster l . When X has categorical attributes, we can introduce a similarity measure as

$$d(X_i, Q_l) = \sum_{j=1}^{mr} (X_{ij}^r - q_{lj}^r)^2 + \gamma_1 \sum_{j=1}^{mc} \delta(X_{ij}^c, q_{lj}^c) \quad (3)$$

where $\delta(p, q) = 0$ for $p = q$ and $\delta(p, q) = 1$ for $p \neq q$. x_{ij}^r and q_{lj}^r are values of numeric attributes, whereas x_{ij}^c and q_{lj}^c are values of categorical attributes for object i and the prototype of cluster l . mr and mc are the numbers of numeric and categorical attributes. γ_1 is a weight for categorical attributes for cluster l .

We can rewrite E_l as

$$E_l = \sum_{i=1}^n y_{il} \sum_{j=1}^{mr} (X_{ij}^r - q_{lj}^r)^2 + \gamma_1 \sum_{i=1}^n y_{il} \sum_{j=1}^{mc} \delta(X_{ij}^c, q_{lj}^c) \quad (4)$$

where E_l^r is the total cost on all numeric attributes of objects in cluster l . E_l^r is minimised if q_{lj}^r is calculated. Let C_j be the set containing all unique values in the categorical attribute j and $p(c_j \in C_j | l)$ the probability of value c_j occurring in cluster l .

E_l^c can be rewritten as

$$E_l^c = \gamma_1 \sum_{j=1}^{mc} n_l (1 - p(c_j \in C_j | l)) \quad (5)$$

3.1.1.4 Similarity Measure

The cost function defined is a combined similarity measure on both numeric and categorical attributes between objects and cluster prototypes. The similarity measure on numeric attributes is the square Euclidean distance whereas the similarity measure on categorical attributes is the number of mismatches between objects and cluster prototypes. Weight γ_1 is introduced to avoid favouring either type of attribute.

If $\gamma_1 = 0$, clustering only depends on numeric attributes, i.e., locations of the objects. The result will be the two clusters separated by the vertical dashed line. If $\gamma_1 > 0$

3.1.1.5 K-prototypes Algorithm

The k-prototypes algorithm can be described as follows.

- 1) Select k initial prototypes from a data set X , one for each cluster.
- 2) Allocate each object in X to a cluster whose prototype is the nearest to it according to cost function. Update the prototype of the cluster after each allocation.
- 3) After all objects have been allocated to a cluster, retest the similarity of objects against the current prototypes. If an object is found such that its nearest prototype belongs to another cluster rather than its current one, reallocate the object to that cluster and update the prototypes of both clusters.
- 4) Repeat (3) until no object has changed clusters after a full cycle test of X . The algorithm is built upon three processes, initial prototypes selection, initial allocation, and reallocation.

3.1.1.6 Handling Missing Data

- The algorithm treats any missing / unseen data as matching with each other but mismatching with non-missing / seen data when determining similarity between points.
- When predicting, the model treats any values in matrix that
 - It has not seen before during training
 - Are missing, as being a member of the "unknown values" category.
- K-prototypes accepts np.NaN values as missing values in the Matrix for categorical values and for numerical values, it does not accept missing values
- K-modes algorithm accepts np.NaN values as missing values in the matrix, fitting them in their own category.

3.1.1.7 Cluster selection for target user

To assign target user p to a cluster, we compute the Euclidean distance between p and each of the cluster centroids.

We choose that cluster whose centroid has the least Euclidean distance, and we add p to that cluster provided the Euclidean distance is less than a threshold.

3.2 Implementation

3.2.1 Creating a hotspot using C.H.I.P

C.H.I.P can be configured to act as a hotspot which provides a secure network for the user to connect to. It allows requests the backend for recommendations for the user based on his user id and takes the response and pushes out the recommendations to the mobile phone.

3.2.2 Mobile Application

3.2.2.1 Methodology involved in mobile application

1. The mobile application, when launched, automatically enables WiFi option in the smartphone and scans for the store's hotspot.
2. If it is available, it disconnects from any other network to which it is already connected and connects to the hotspot.
3. Else, it displays a message to the user that the store's network is not available.
4. If connected, the hardware device fetches recommendations for the user and pushes t to the application. Application displays the recommendations to the user.

3.2.3 Implementation of Recommender System

3.2.3.1 Clustering Users

Clustering of Users is done using K-Prototypes algorithm.

K-Prototypes algorithms randomly allocate the initial centroids.

Number of clusters that could be made starts from a minimum value of 3. Optimal solution for the number of clusters the data should be divided into is determined using the cost function. A dramatic drop in the cost function is observed when an optimal number of clusters are used.

For our data a total of 10 clusters were formed.

A scatter plot was made with users on the X-axis and Clusters on Y-axis.

Users and their respective clusters are written into a file and this file forms an input for the next step.

3.2.3.2 Merging users of Clusters with ISBNs of the Books they rated

From the previous step, we get a file which consists of UserID and their respective cluster number.

Book Ratings file has UserID, ISBN of the book and the rating

These 2 documents are merged to get userID, ISBN of the book they rated, the rating and their cluster into a single file.

3.2.3.3 Finding Popular books in each cluster

Users are sorted according to their cluster numbers.

Users belonging to same cluster are placed in one file.

Books which are rated above 5 are taken to retrieve popular books belonging to each cluster.

3.2.3.4 Generating Recommendations

An existing user gets recommended 5 best rated books; which he/she hasn't read already.

If the number of best rated books is below 5 in a given cluster, all the books are recommended.

For a new user, he is first assigned to a cluster and then top rated books in that cluster are recommended to him.

3.2.4 MongoDB Documents

MongoDB stores data records as BSON documents. BSON is a binary representation of JSON documents, though it contains more data types than JSON. MongoDB documents are composed of field-and-value pairs and have a unique id for each and every document. The value of a field can be any of the BSON data types, including other documents, arrays, and arrays of documents. Following image shows few documents of User's data collection.

```

> use RetailDB
switched to db RetailDB
> db.usersdata.find()
{ "_id" : ObjectId("58a3d996ca390d5c96c613f7"), "User-ID" : 1, "Location" : "nyc, new york, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613f8"), "User-ID" : 2, "Location" : "stockton, california, usa", "Age" : 18 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613f9"), "User-ID" : 3, "Location" : "moscow, yukon territory, russia", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fa"), "User-ID" : 4, "Location" : "porto, v.n.gaia, portugal", "Age" : 17 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fb"), "User-ID" : 5, "Location" : "farnborough, hants, united kingdom", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fc"), "User-ID" : 6, "Location" : "santa monica, california, usa", "Age" : 61 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fd"), "User-ID" : 9, "Location" : "germantown, tennessee, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fe"), "User-ID" : 11, "Location" : "melbourne, victoria, australia", "Age" : 14 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613ff"), "User-ID" : 12, "Location" : "fort bragg, california, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61400"), "User-ID" : 13, "Location" : "barcelona, barcelona, spain", "Age" : 26 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61401"), "User-ID" : 14, "Location" : "mediapolis, iowa, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61402"), "User-ID" : 7, "Location" : "washington, dc, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61403"), "User-ID" : 10, "Location" : "albacete, wisconsin, spain", "Age" : 26 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61404"), "User-ID" : 8, "Location" : "timmins, ontario, canada", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61405"), "User-ID" : 18, "Location" : "rio de janeiro, rio de janeiro, brazil", "Age" : 25 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61406"), "User-ID" : 19, "Location" : "weston, ,,", "Age" : 14 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61407"), "User-ID" : 20, "Location" : "langhorne, pennsylvania, usa", "Age" : 19 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61408"), "User-ID" : 21, "Location" : "ferrol / spain, alabama, spain", "Age" : 46 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61409"), "User-ID" : 22, "Location" : "erfurt, thuringen, germany", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c6140a"), "User-ID" : 23, "Location" : "philadelphia, pennsylvania, usa", "Age" : "NULL" }
Type "it" for more
> exit
bye
    
```

Figure 3.4 MongoDB Documents

IV. Testing

Testing is defined as a fault detection technique that creates failure or erroneous states. Before releasing software to the customer, this testing phase is done to detect the failures in the system. The testing phase is done by the developer. The process of testing is done at different levels. Each level in testing has its own methodology. All the methodologies are used while conducting software testing. The software testing is done after the implementation phase of software life cycle

4.1 Testing our proposed system

4.1.1 Checking the connection with Backend

We test our system in the following scenarios.

1. What happens when the hotspot is not available and the user launches the mobile application?

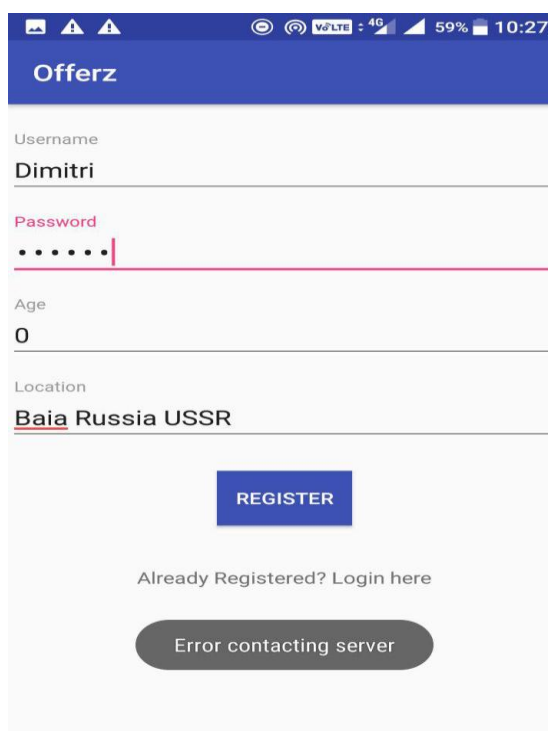


Fig. 4.1 Error Connecting with Backend

4.1.2 Validating all Input fields

2. Is the app validating User's Age

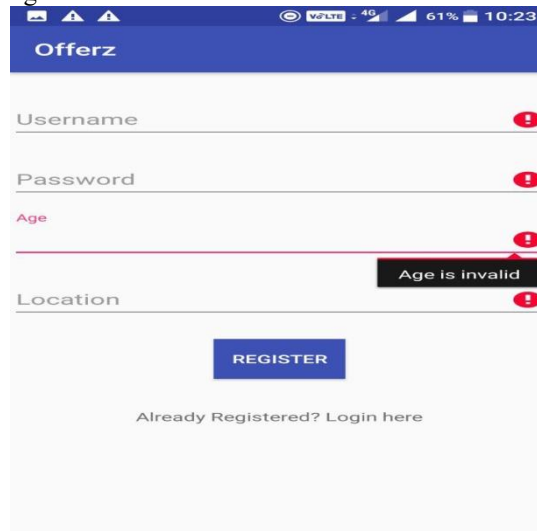


Fig. 4.2 Invalid Age Error

3. Validating the length of username

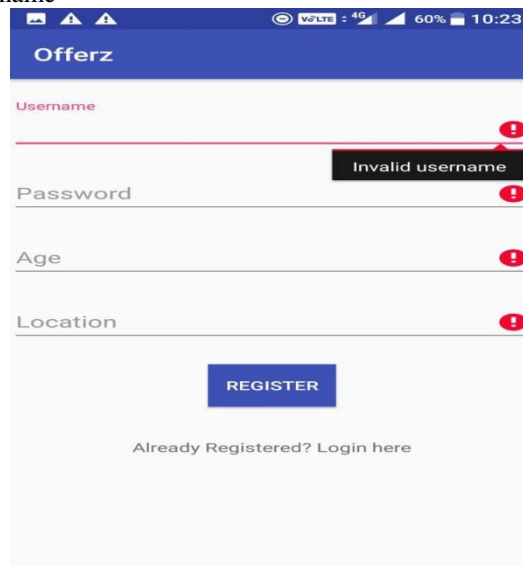


Fig 4.3 Validating username

4. Validating User location

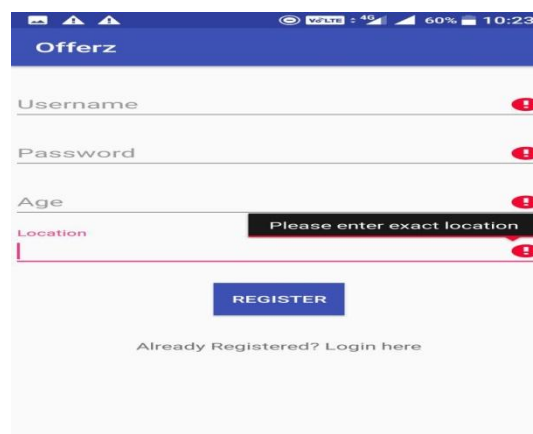


Fig 4.4 Validating user location

5. Validating password length

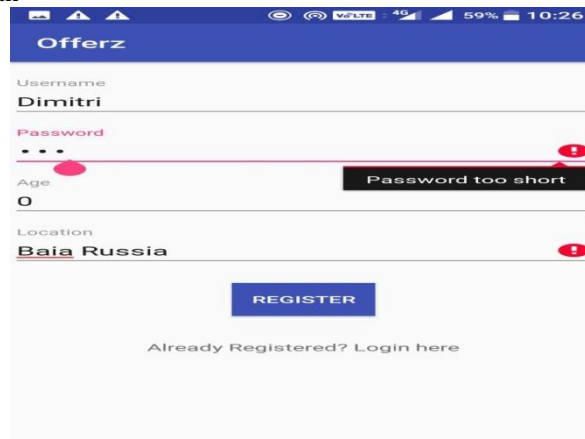


Fig 4.5 Validating password length

V. Data Set

Book Rating Data Set is used. Our Dataset consists of 111,662 users and 14,467 books and a total of 1,149,366 ratings mapped among them. It is collected by Cai-Nicolas Ziegler, University of Freiburg

URL: <http://www2.informatik.uni-freiburg.de/~cziegler/BX/>

The BX-Book Ratings Dataset consists of 3 tables

- BX_BOOKS
- BX_USERS
- BX_BOOK RATINGS

BX_Books

- Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset.
- Moreover, some content-based information is given (``Book-Title``, ``Book-Author``, ``Year-Of-Publication``, ``Publisher``), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided.

Book Data is loaded into MongoDB Instance. A screenshot is provided below.

```

usersdata
> db.books.find()
{ "_id" : ObjectId("58b381c1828caf69abf3d9b2"), "ISBN" : "2005018", "Book-Title" : "Clara Callan", "Book-Author" : "Richard Bruce Wright", "Year-Of-Publication" : 2001, "Publisher" : "HarperFlamingo Canada", "Image-URL-S" : "http://images.amazon.com/images/P/0002005018.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9b3"), "ISBN" : "393045218", "Book-Title" : "The Mummies of Urunchi", "Book-Author" : "E. J. W. Barber", "Year-Of-Publication" : 1999, "Publisher" : "W. W. Norton & Company", "Image-URL-S" : "http://images.amazon.com/images/P/0393045218.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9b4"), "ISBN" : "399135782", "Book-Title" : "The Kitchen God's Wife", "Book-Author" : "Amy Tan", "Year-Of-Publication" : 1991, "Publisher" : "Putnam Pub Group", "Image-URL-S" : "http://images.amazon.com/images/P/0399135782.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9b5"), "ISBN" : "425176428", "Book-Title" : "What If?: The World's Foremost Military Historians Imagine What Might Have Been", "Book-Author" : "Robert Cowley", "Year-Of-Publication" : 2000, "Publisher" : "Berkley Publishing Group", "Image-URL-S" : "http://images.amazon.com/images/P/0425176428.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9b6"), "ISBN" : "671870432", "Book-Title" : "PLEADING GUILTY", "Book-Author" : "Scott Turow", "Year-Of-Publication" : 1993, "Publisher" : "Audioworks", "Image-URL-S" : "http://images.amazon.com/images/P/0671870432.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9b7"), "ISBN" : "679425608", "Book-Title" : "Under the Black Flag: The Romance and the Reality of Life Among the Pirates", "Book-Author" : "David Cordingly", "Year-Of-Publication" : 1996, "Publisher" : "Random House", "Image-URL-S" : "http://images.amazon.com/images/P/0679425608.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9b8"), "ISBN" : "074322678X", "Book-Title" : "Where You'll Find Me: And Other Stories", "Book-Author" : "Ann Beattie", "Year-Of-Publication" : 2002, "Publisher" : "Scribner", "Image-URL-S" : "http://images.amazon.com/images/P/074322678X.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9b9"), "ISBN" : "771074670", "Book-Title" : "Nights Below Station Street", "Book-Author" : "David Adams Richards", "Year-Of-Publication" : 1988, "Publisher" : "Enlben Editions", "Image-URL-S" : "http://images.amazon.com/images/P/0771074670.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9ba"), "ISBN" : "195153448", "Book-Title" : "Classical Mythology", "Book-Author" : "Mark P. O. Morford", "Year-Of-Publication" : 2002, "Publisher" : "Oxford University Press", "Image-URL-S" : "http://images.amazon.com/images/P/0195153448.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9bb"), "ISBN" : "080652121X", "Book-Title" : "Hitler's Secret Bankers: The Myth of Swiss Neutrality During the Holocaust", "Book-Author" : "Adan Lebor", "Year-Of-Publication" : 2000, "Publisher" : "Citadel Press", "Image-URL-S" : "http://images.amazon.com/images/P/080652121X.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9bc"), "ISBN" : "887841740", "Book-Title" : "The Middle Stories", "Book-Author" : "Sheila Heti", "Year-Of-Publication" : 2004, "Publisher" : "House of Anansi Press", "Image-URL-S" : "http://images.amazon.com/images/P/0887841740.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9bd"), "ISBN" : "1552041778", "Book-Title" : "Jane Doe", "Book-Author" : "R. J. Kaiser", "Year-Of-Publication" : 1999, "Publisher" : "Mira Books", "Image-URL-S" : "http://images.amazon.com/images/P/1552041778.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9be"), "ISBN" : "1558740218", "Book-Title" : "A Second Chicken Soup for the Woman's Soul (Chicken Soup for the Soul Series)", "Book-Author" : "Jack Canfield", "Year-Of-Publication" : 1998, "Publisher" : "Health Communications", "Image-URL-S" : "http://images.amazon.com/images/P/1558740218.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9bf"), "ISBN" : "1567407701", "Book-Title" : "The Witchfinder (Amos Walker Mystery Series)", "Book-Author" : "Loren D. Estleman", "Year-Of-Publication" : 1998, "Publisher" : "Brilliance Audio - Trade", "Image-URL-S" : "http://images.amazon.com/images/P/1567407701.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9c0"), "ISBN" : "06973129", "Book-Title" : "Decision in Normandy", "Book-Author" : "Carlo D'Este", "Year-Of-Publication" : 1991, "Publisher" : "HarperPerennial", "Image-URL-S" : "http://images.amazon.com/images/P/0696973129.01.THUMBZZZ.jpg" }
{ "_id" : ObjectId("58b381c1828caf69abf3d9c1"), "ISBN" : "374157065", "Book-Title" : "Flu: The Story of the Great Influenza Pandemic of 1918 and
    
```

Fig 5.1 BX-Books Collection

BX_USERS

- Contains the user's data. Note that user IDs (``User-ID``) have been anonymized and map to integers.
- Demographic data is provided (``Location``, ``Age``) if available. Otherwise, these fields contain `NULL` values.


```
> use RetailDB
switched to db RetailDB
> db.usersdata.find()
{ "_id" : ObjectId("58a3d996ca390d5c96c613f7"), "User-ID" : 1, "Location" : "nyc, new york, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613f8"), "User-ID" : 2, "Location" : "stockton, california, usa", "Age" : 18 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613f9"), "User-ID" : 3, "Location" : "moscow, yukon territory, russia", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fa"), "User-ID" : 4, "Location" : "porto, v.n.gala, portugal", "Age" : 17 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fb"), "User-ID" : 5, "Location" : "farnborough, hants, united kingdom", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fc"), "User-ID" : 6, "Location" : "santa monica, california, usa", "Age" : 61 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fd"), "User-ID" : 9, "Location" : "germantown, tennessee, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c613fe"), "User-ID" : 11, "Location" : "melbourne, victoria, australia", "Age" : 14 }
{ "_id" : ObjectId("58a3d996ca390d5c96c613ff"), "User-ID" : 12, "Location" : "fort bragg, california, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61400"), "User-ID" : 13, "Location" : "barcelona, barcelona, spain", "Age" : 26 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61401"), "User-ID" : 14, "Location" : "mediapolis, iowa, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61402"), "User-ID" : 7, "Location" : "washington, dc, usa", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61403"), "User-ID" : 10, "Location" : "albacete, wisconsin, spain", "Age" : 26 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61404"), "User-ID" : 8, "Location" : "timmins, ontario, canada", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c61405"), "User-ID" : 18, "Location" : "rio de janeiro, rio de janeiro, brazil", "Age" : 25 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61406"), "User-ID" : 19, "Location" : "weston, ", "Age" : 14 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61407"), "User-ID" : 20, "Location" : "langhorne, pennsylvania, usa", "Age" : 19 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61408"), "User-ID" : 21, "Location" : "ferrol / spain, alabama, spain", "Age" : 46 }
{ "_id" : ObjectId("58a3d996ca390d5c96c61409"), "User-ID" : 22, "Location" : "erfurt, thuringen, germany", "Age" : "NULL" }
{ "_id" : ObjectId("58a3d996ca390d5c96c6140a"), "User-ID" : 23, "Location" : "philadelphia, pennsylvania, usa", "Age" : "NULL" }
Type "it" for more
> exit
bye
```

Fig. 5.2 BX_Users Collection

BX_BOOK RATINGS

- Contains the book rating information. Ratings (*Book-Rating*) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

1	User-ID	ISBN	Book-Rating
2	276725	034545104X	0
3	276726	155061224	5
4	276727	446520802	0
5	276729	052165615X	3
6	276729	521795028	6
7	276733	2080674722	0
8	276736	3257224281	8
9	276737	600570967	6
10	276744	038550120X	7
11	276745	342310538	10
12	276746	425115801	0
13	276746	449006522	0
14	276746	553561618	0
15	276746	055356451X	0

Fig. 5.3 BX_Book Ratings Table

VI. Results

6.1 Clustering

Clustering Users based on location and age and plotting them on a scatter plot.

A total of 10 clusters were formed. A scatter plot was plotted with users on X-axis and Clusters on Y-axis

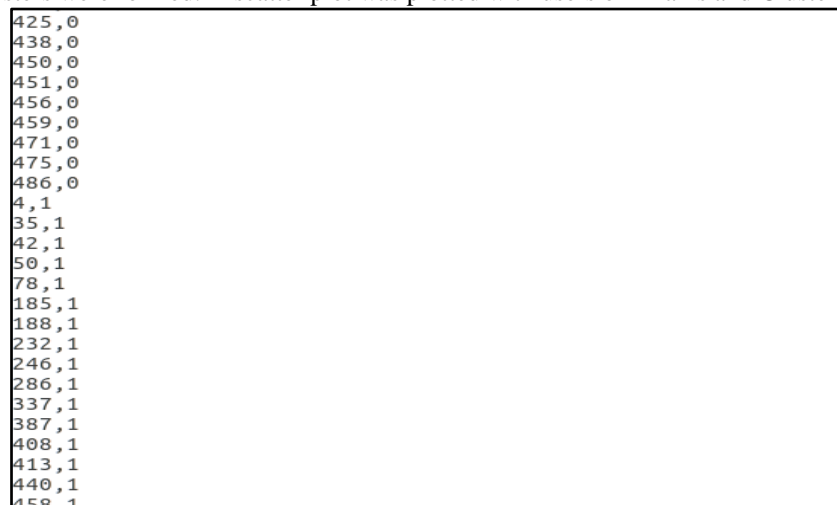


Fig. 6.1 Users and their Clusters

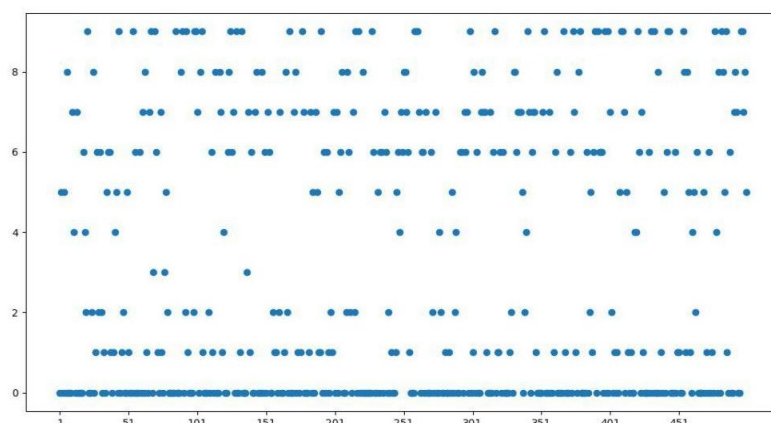


Fig. 6.2 Clustered User's Scatter Plot

6.2 Finding Popular Books and generating Recommendations Specific to a user

Users are Merged with their Clusters.

A cluster has two types of users

- Those who gave ratings to books
- Those who didn't give any ratings

A threshold of minimum rating, 6 out of 10 is placed to get popular books.

A list of best rated books specific to individual users are generated. Randomly 5 best rated books are picked and stored in the database. If the number of best rated books for a given cluster are less than 5, then all the recommendations specific to the individual users are picked and stored in the database.

```

Recommendations for user: 433
0671621009
8429725792
8483008505
847592543X
0312261594
Recommendations for user: 90
0446677450
0679429220
8475960936
039575514X
8478098267
Recommendations for user: 431
0440223571
0679429220
8429725792
8474103347
055356451X
yogitha clusters $ _
    
```

Fig. 6.3 Recommendations generated for individual user

6.3 Storing Top Recommendations of each cluster in MongoDB

```

1 {
2   "_id": 0,
3   "recommendations": [
4     {
5       "book_title": "Airframe",
6       "isbn": "0345402871",
7       "author": "Micheal Aelcrichton",
8       "image_url": "http://images.amazon.com/images/P/0345402871.01.THUMBZZZ.jpg"
9     },
10    {
11      "book_title": "To kill Mockingbird",
12      "isbn": "0446310786",
13      "author": "Harper Lee",
14      "image_url": "http://images.amazon.com/images/P/0446310786.01.THUMBZZZ.jpg"
15    },
16    {
17      "book_title": "Seabiscuit: An American Legend",
18      "isbn": "0449005615",
19      "author": "Laura Hillenbrand",
20      "image_url": "http://images.amazon.com/images/P/0449005615.01.THUMBZZZ.jpg"
21    },
22    {
23      "book_title": "All the KingsMen",
24      "isbn": "0156047614",
25      "author": "Robert Penn Warren",
26      "image_url": "http://images.amazon.com/images/P/0156047614.01.THUMBZZZ.jpg"
27    }
28  ]
29 }
    
```

Fig. 6.4 Storing Top Recommendations for each cluster in MongoDB

6.4 User Interface of Mobile Application and Displaying Recommendations on user's phone

User Interface of the mobile application allows the user to

- Register
 - Login
- After logging in, the user can request for recommendations.
- Mobile Application on launch - Turning on WiFi and connecting to CHIP's hotspot (if available) automatically.

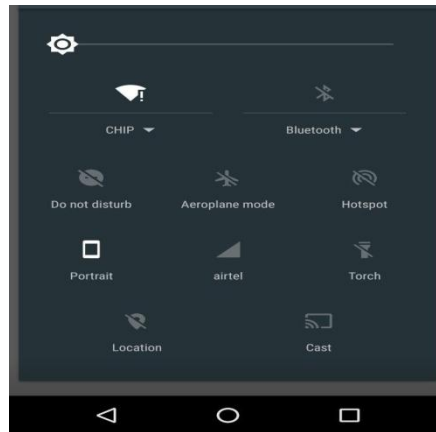


Fig. 6.5 Connecting to Hotspot

- Registration Screen

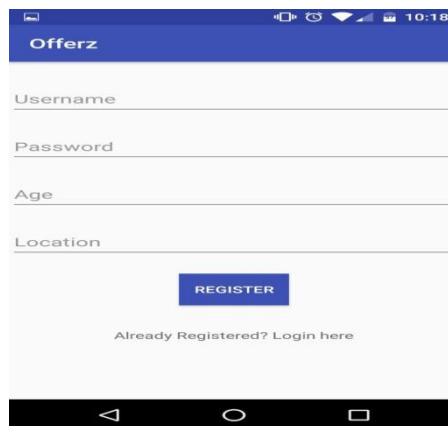


Fig. 6.6 Registration Screen

- Registering a New User

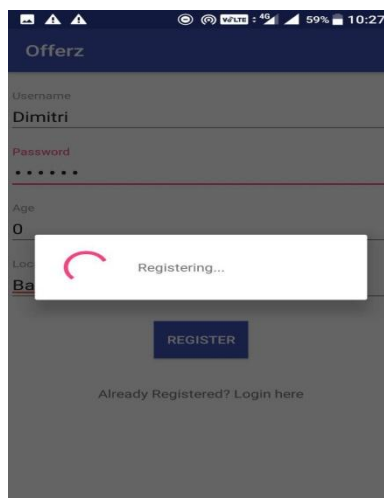


Fig. 6.7 Registering New User

- After connection to the hotspot is established, retrieving the recommendations..

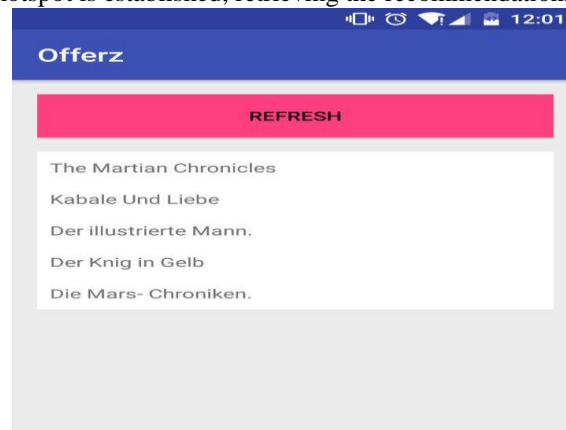


Figure 6.8 Displaying Offers

VII. Conclusion

7.1 Conclusion

Main purpose of the proposed system is to provide business advantage to retail industry. By moving the recommender technology to offline stores, a plethora of possibilities would be opened for entrepreneurs and retail stores. By serving their customers with personalised offers, they increase their customer loyalty and repeat customer transactions. This equips them to effectively combat the major online retailers like Flipkart, SnapDeal, Amazon etc. Offline stores are an integral part of any economy and this work will help them to survive the battle with online retailers.

7.2 Future Work

In future we plan to extend this system and incorporate content-based recommendation system. It would also be interesting to cluster books based on their rating, content and reviews by the user. That way, not only the books read by similar users will be recommended to the target user, but also the books similar to a target book will also be recommended to him.

References

- [1]. Shani G, Gunawardana, A Evaluating Recommendation Systems. In: Recommender Systems Handbook, pp. 257–297, 2011.
- [2]. Wojtek Kowalczyk, Recommender systems for e-shops, faculty of Sciences, Vrije Universiteit, Amsterdam, 2011.
- [3]. Harpreet Kaur Virk, Er. Maninder Singh, Er. Amritpal Singh, Analysis and Design of Hybrid Online Movie Recommender System, International Journal of Innovations in Engineering and Technology, 5(2), 2015
- [4]. Jyoti Gupta, Jayant Gadge, Performance Analysis of Recommendation System Based on Collaborative Filtering and Demographics, International Conference on Communication, Information & Computing Technology (ICCICT), Jan. 16-17, 2015 Mumbai, India.
- [5]. Manoj Kumar, D.K. Yadav, Ankur Singh, Vijay Kr. Gupta, A Movie Recommender System: MOVREC, International Journal of Computer Applications (0975 – 8887), 124(3), August 2015.
- [6]. Zhexue Huang, Clustering Large Data sets with mixed Numeric and Categorical Values, CSIRO Mathematical and Information Sciences, GPO Box 664 Canberra ACT 2601, Australia