

Recommending E-commerce products Using Transaction Data

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

KAVURI SAI KRISHNA (Reg. No. 37110340)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **KAVURI SAI KRISHNA** (Reg. No. **37110340**) who carried out the project entitled "**Recommending E-commerce products Using Transaction Data**" under my supervision from **August 2020** to **March 2021**.

Internal Guide
Dr..Meera Gandhi, M.E.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KAVURI SAI KRISHNA** hereby declare that the Project Report entitled "**Recommending E-commerce products Using Transaction Data**" is done by me under the guidance of **Mrs.Meera Gandhi, M.E.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs.Meera Gandhi, M.E.,Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The number of “hits” has been widely regarded as the lifeblood of many web systems, e.g., e-commerce systems, advertising systems and multimedia consumption systems. However, users would not hit an item if they cannot see it, or they are not interested in the item. Recommender system plays a critical role of discovering interesting items from near-infinite inventory and exhibiting them to potential users. Yet, two issues are crippling the recommender systems. One is “how to handle new users”, and the other is “how to surprise users”. The former is well-known as cold-start recommendation. In this paper, we show that the latter can be investigated as long-tail recommendation. We also exploit the benefits of jointly challenging both cold-start and long-tail recommendation, and propose a novel approach which can simultaneously handle both of them in a unified objective. For the cold-start problem, we learn from side information, e.g., user attributes, user social relationships, etc. Then, we transfer the learned knowledge to new users. For the long-tail recommendation, we decompose the overall interesting items into two parts: a low-rank part for short-head items and a sparse part for long-tail items. The two parts are independently revealed in the training stage, and transferred into the final recommendation for new users. Furthermore, we effectively formulate the two problems into a unified objective and present an iterative optimization algorithm. A fast extension of the method is proposed to reduce the complexity, and extensive theoretical analysis are provided to proof the bounds of our approach. At last, experiments of social recommendation on various real-world datasets.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 Synopsis	1
2.	SYSTEM ANALYSIS	2
	2.1 EXISTING SYSTEM	2
	2.2 PROPOSED SYSTEM	2
3.	REQUIREMENT SPECIFICATIONS	3
	3.1 INTRODUCTION	3
	3.2 HARDWARE AND SOFTWARE SPECIFICATION	3
	3.3 TECHNOLOGIES USED	3
4.	4. PROJECT PURPOSE AND SCOPE	12
	4.1 PROJECT PURPOSE	12
	4.2 PROJECT SCOPE	12
	4.3 PRODUCT PRESEPCTEIVE	12
	4.4 SYSYTEM FEAUTERS	13
	4.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	13
	4.6 OTHER NON-FUNCTIONAL REQUIREMENTS	14
5.	5. ARCHITECTURE DIAGRAM	15
6.	6.SYSTEM DESIGN	17
	6.1 MODULES	18
7.	7. CODING AND TESTING	19
	7.1 CODING	19
	7.2 CODING STANDARDS	19
	7.3 TEST PROCEDURE	20
	7.4 TEST DATA OUTPUT	21
	7.5 TESTING TECHNIQUES/TESTING STRATEGIES	23

8.	RESULTS AND DISCUSSION	26
9.	CONCLUSION AND FUTURE WORK	29
	REFERENCES	31
	APPENDIX	
	A. PAPER ACCEPTANCE MAIL	32
	B. PLAGIARISM REPORT	33
	C. JOURNAL PAPER	37
	D. SOURCE CODE	42

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
3.1	JVM DIAGRAM	6
3.2	COMPILER DIAGRM	6
3.3	PLATFORM DIAGRAM	7
3.4	ANDROID ARCHITECTURE	10
5.1	ARCHITECTURE DIAGRAM	15
8.1	LOGIN PAGE	26
8.2	PRODUCT PAGE	27
8.3	OFFER PAGE	27
8.4	DATABASE PAGE	28
8.5	USER DETAILS PAGE	29

CHAPTER

1 INTRODUCTION

1. INTRODUCTION

The main aim of this project is to achieve the cold start and long tail problems and recommend the products using social data.

1.1 Synopsis:

The number of “hits” has been widely regarded as the lifeblood of many web systems, e.g., e-commerce systems, advertising systems and multimedia consumption systems. However, users would not hit an item if they cannot see it, or they are not interested in the item. Recommender system plays a critical role of discovering interesting items from near-infinite inventory and exhibiting them to potential users. Yet, two issues are crippling the recommender systems. One is “how to handle new users”, and the other is “how to surprise users”. The former is well-known as cold-start recommendation. In this paper, we show that the latter can be investigated as long-tail recommendation. We also exploit the benefits of jointly challenging both cold-start and long-tail recommendation, and propose a novel approach which can simultaneously handle both of them in a unified objective. For the cold-start problem, we learn from side information, e.g., user attributes, user social relationships, etc. Then, we transfer the learned knowledge to new users. For the long-tail recommendation, we decompose the overall interesting items into two parts: a low-rank part for short-head items and a sparse part for long-tail items. The two parts are independently revealed in the training stage, and transferred into the final recommendation for new users. Furthermore, we effectively formulate the two problems into a unified objective and present an iterative optimization algorithm. A fast extension of the method is proposed to reduce the complexity, and extensive theoretical analysis are provided to proof the bounds of our approach. At last, experiments of social recommendation on various real-world datasets, e.g., images, blogs, videos and musics, verify the superiority of our approach compared with the state-of-the-art work.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Recommender system plays a critical role of discovering interesting items from near-infinite inventory and exhibiting them to potential users. Yet, two issues are crippling the recommender systems. One is “how to handle new users”, and the other is “how to surprise users”. The former is well-known as cold-start recommendation. In this paper, we show that the latter can be investigated as long-tail recommendation. Cold-start problem has been regarded as a quite challenging problem in the community and no direct information available. In existing system have revealed that recommending popular items do not bring much benefits to either users or content providers.

2.2 PROPOSED SYSTEM

Based on the problems due to cold start and long tail, products in the ecommerce retailers getting sold out so soon . And some kind of products remains stagnant for a long duration. Thus to overcome these problems we are going to track the selling of products as well as recommending the products to the customers of same kind based on the previous purchases by using clustering and classification we achieve this First user register the details like name ,password,address and registers then login with valid credentials then the large amount of transactional data collected from customer for segmentation and classification and these data are preprocessed and cluster and classify the data cold product classified based on sale and long tail is classified based product that getting sold frequently after that cold products that are available in the ecommerce site will be recommended to the user to make retailer some profit for the product and finally long tail problem solved by recommending relevant item and product that are not sold for long period gets attractive offer based on the user request.We use hadoop environment to do manipulation with the products dataset.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

3.1 INTRODUCTION

The success of any kind of social epidemic is heavily dependent on the involvement of people with a particular and rare set of social gifts,” writes Malcolm Gladwell in his bestseller *The Tipping Point* [1]. He further named these people as connectors, mavens and salesmen. The connectors know people, the mavens accumulate information of commodity, and the salesman persuade people to buy things. What’s particularly intriguing is that in online retail stores, recommender system [2], [3], [4], [5], [6] plays the three roles at the same time. Specifically, a recommender system discovers the interesting items from a near-infinite inventory and exhibits them to potential users. In other words, a recommender system must know people to find out potential users, must accumulate information to pick up interesting items, and must persuade the potential users to buy the recommended items. As a result, the success of on-line retailing is heavily dependent the recommender system. Actually, Amazon.com has received a 35% of their revenue from recommendations, and 75% of what people watch in Netflix is as a result of recommendations¹.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

HARDWARE REQUIREMENTS

Hard Disk	:	250GB and Above
RAM	:	4GB and Above
Processor	:	I3 and Above(64 bit)
Wifi Router	:	1No.

SOFTWARE REQUIREMENTS

Windows 7
JDK 1.7
J2EE
Tomcat 6.0
MySQL
Hadoop

3.3 TECHNOLOGIES USED

JAVA

Android

JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be

used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the `argv []` of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out**, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things : 1.Programming language

2.Platform

THE JAVA PROGRAMMING LANGUAGE

Java is a high-level programming language that is all of the following:

Simple

Object-oriented

Distributed

Interpreted

Robust

Secure

Architecture-neutral

Portable

High-performance

Multithreaded

Dynamic

The code can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called Java byte codes – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:

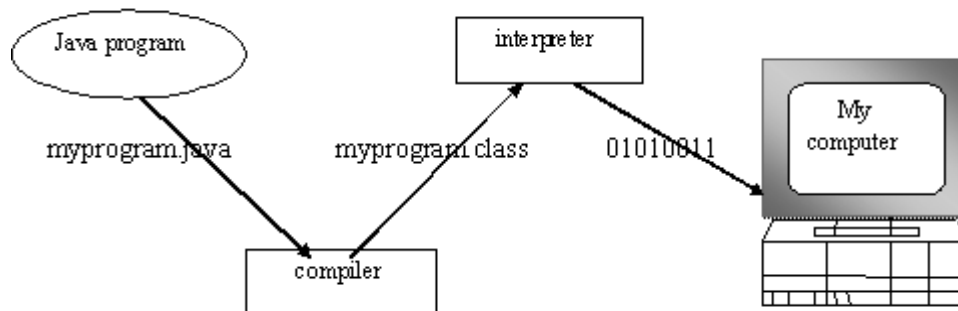
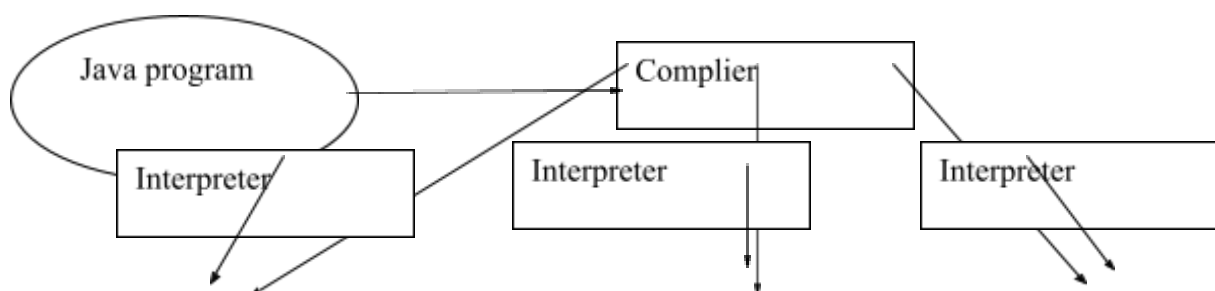


Fig.3.1:JVM DIAGRAM

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (JVM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can be run on Windows NT, Solaris and Macintosh



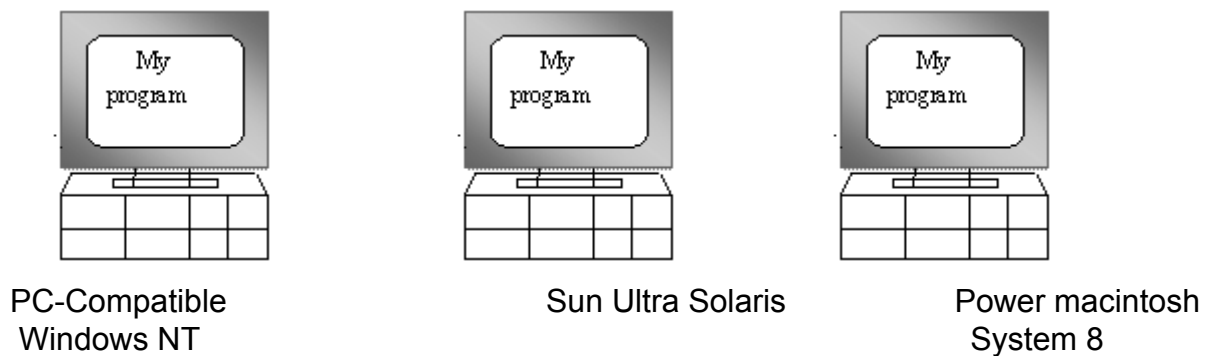


FIG:3.2:COMPILER DIAGRAM

THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components :

- 1.The Java Virtual Machine (JVM)
- 2.The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (packages) of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

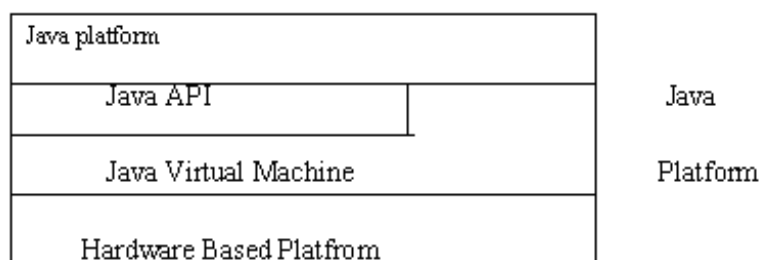


Fig.3.3:JAVA PLATFORM

As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Environment

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

Product features

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications

- servlet reloading

- basic HTTP functionality Tomcat 4.x

- implements the Servlet 2.3 and JSP 1.2 specifications

- servlet container redesigned as Catalina

- JSP engine redesigned as Jasper

- Coyote connector

- Java Management Extensions (JMX), JSP and Struts-based administration

Tomcat 5.x

implements the Servlet 2.4 and JSP 2.0 specifications

reduced garbage collection, improved performance and scalability

native Windows and Unix wrappers for platform integration

faster JSP parsing

History

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

Android Introduction:

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The operating system has developed a lot in last 15 years starting from black and white phones to recent smart phones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.

The android is a powerful operating system and it supports large number of applications in Smartphones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it. The android has got millions of apps available that can help you managing your life one or

other way and it is available low cost in market at that reasons android is very popular.

The android development supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 and latest updated version is jelly bean.

The android is a operating system and is a stack of software components which is divided into five sections and four main layers that is

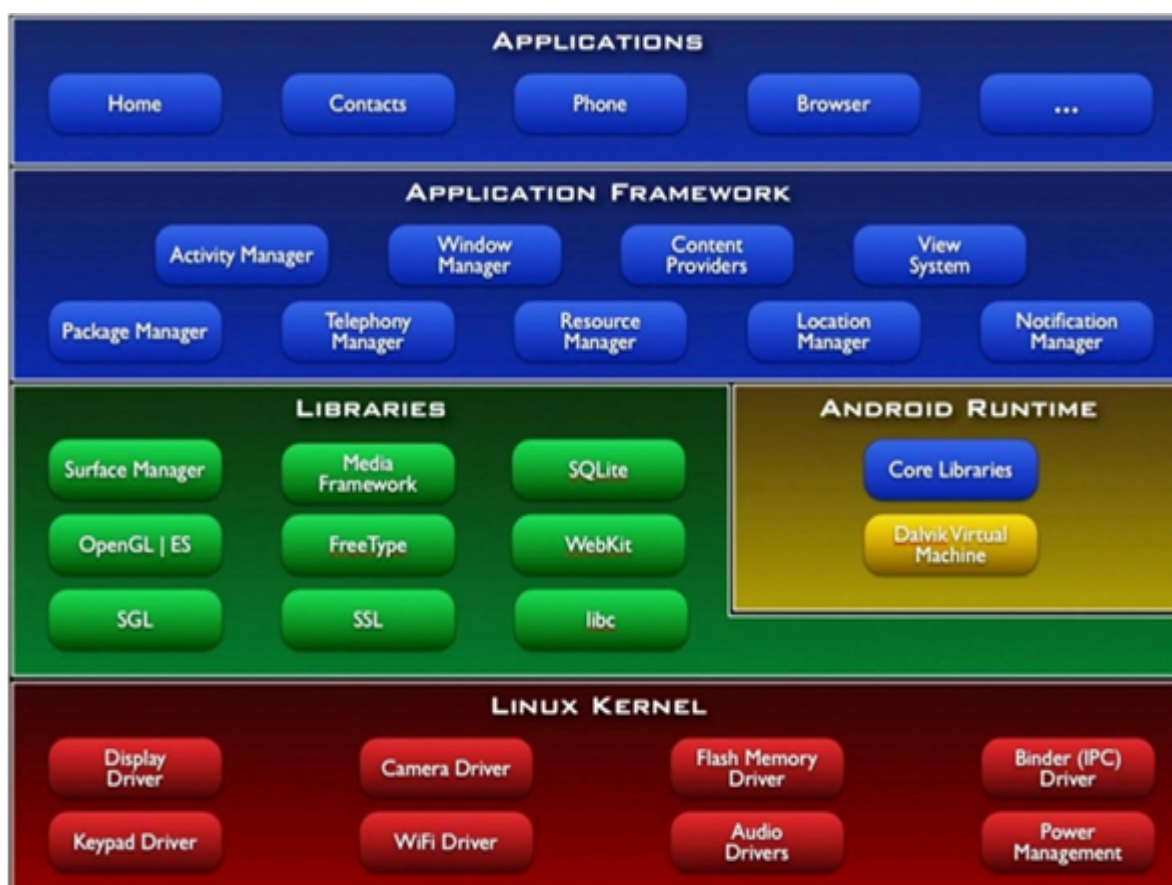


Fig:3.4:Android Architecture

Linux kernel:

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management, device management like camera, keypad, display etc the kernel handles all the things. The Linux is really good at networking and it is not

necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hardware devices on the system.

Libraries:

On top of a Linux kernel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and video. The SQLite is a database which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

Android Runtime:

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run its own process. The Dalvik VM executes the files in the .dex format.

Application framework:

The application framework layer provides many higher level services to applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

Applications:

You will find all the android applications at the top layer and you will write your application and install on this layer. Examples of such applications are contacts, books, browsers, services etc. Each application performs a different role in the overall applications.

Advantages:

Android is Linux based open source operating system, it can be developed by anyone

Easy access to the android apps

You can replace the battery and mass storage, disk drive and UDB option

Its supports all Google services

The operating system is able to inform you of a new SMS and Emails or latest updates.

It supports Multitasking

Android phone can also function as a router to share internet

Its free to customize

Can install a modified ROM

Its supports 2D and 3D graphics

CHAPTER 4

Project Purpose and Scope

4.1 Project Purpose

The main aim of this project is to achieve the cold start and long tail problems and recommend the products using social data.

4.2 Project Scope

Based on the problems due to cold start and long tail, products in the ecommerce retailers getting sold out so soon . And some kind of products remains stagnant for a long duration. Thus to overcome these problems we are going to track the selling of products as well as recommending the products to the customers of same kind based on the previous purchases by using clustering and classification we achieve this First user register the details like name ,password,address and registers then login with valid credentials then the large amount of transactional data collected from customer for segmentation and classification and these data are preprocessed and cluster and classify the data cold

product classified based on sale and long tail is classified based product that getting sold frequently after that cold products that are available in the ecommerce site will be recommended to the user to make retailer some profit for the product and finally long tail problem solved by recommending relevant item and product that are not sold for long period gets attractive offer based on the user request. We use hadoop environment to do manipulation with the products dataset.

4.3 Product Perspective

The number of “hits” has been widely regarded as the lifeblood of many web systems, e.g., e-commerce systems, advertising systems and multimedia consumption systems. However, users would not hit an item if they cannot see it, or they are not interested in the item. Recommender system plays a critical role of discovering interesting items from near-infinite inventory and exhibiting them to potential users. Yet, two issues are crippling the recommender systems. One is “how to handle new users”, and the other is “how to surprise users”. The former is well-known as cold-start recommendation. In this paper, we show that the latter can be investigated as long-tail recommendation. We also exploit the benefits of jointly challenging both cold-start and long-tail recommendation, and propose a novel approach which can simultaneously handle both of them in a unified objective. For the cold-start problem, we learn from side information, e.g., user attributes, user social relationships, etc. Then, we transfer the learned knowledge to new users. For the long-tail recommendation, we decompose the overall interesting items into two parts: a low-rank part for short-head items and a sparse part for long-tail items. The two parts are independently revealed in the training stage, and transferred into the final recommendation for new users. Furthermore, we effectively formulate the two problems into a unified objective and present an iterative optimization algorithm. A fast extension of the method is proposed to reduce the complexity, and extensive theoretical analysis are provided to proof the bounds of our approach. At last, experiments of social recommendation on various real-world datasets, e.g., images, blogs, videos and musics, verify the superiority of our approach compared with the state-of-the-art work.

4.4 System Features

To the best of our knowledge, this is the first work in the community which challenges both cold-start and long-tail recommendation in a unified objective. An iterative optimization algorithm is further proposed to solve it. Since scalability is a main challenge in

recommender systems, we propose a fast extension of the method to reduce the computational complexity. Extensive theoretical analysis, including estimation bound and generalization bound, are presented. Experimental results on four real-world datasets, e.g., images, blogs, videos and musics, verify the superiority of our approach compared with the state-of-the-art work. It is worth noting that our recommendation is beyond accuracy, we also keep an eye on the diversity and serendipity of the recommendations.

4.5 Design and Implementation Constraints

Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions

Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

4.6 Other Nonfunctional Requirements

Performance Requirements

The application at this side controls and communicates with the following three main general components.

embedded browser in charge of the navigation and accessing to the web service;

Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

Safety Requirements

The software may be safety-critical. If so, there are issues associated with its integrity level

The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.

If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.

There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.

If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.

Systems with different requirements for safety levels must be separated.

CHAPTER 5

ARCHITECTURE DIAGRAM

5.1 Architecture Diagram:

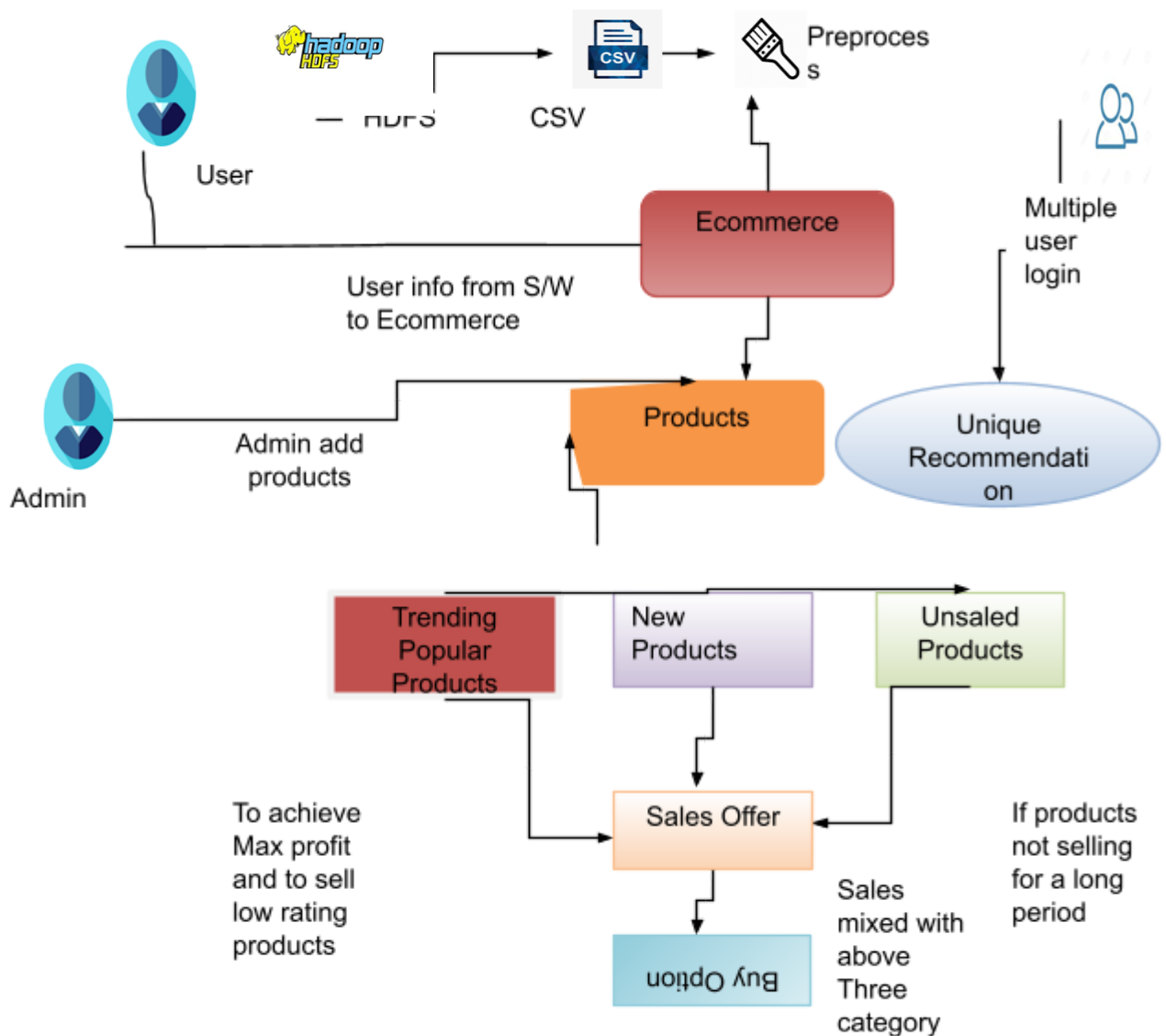


Fig: 5.1

Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

Usecase Overview

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

Rounded rectangles represent activities.

Diamonds represent decisions.

Bars represent the start or end of concurrent activities.

A black circle represents the start of the workflow.

An encircled circle represents the end of the workflow.

Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes, objects.

Data Flow Diagram:

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

Level 0:

Level 1:

Level 2:

Level 3:

5.7 Class Diagram

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

CHAPTER 6

SYSTEM DESIGN

Initially user need to create their own account and register their basic detail in the server. Here whereas the Database server will maintain the user personal detail and all the transaction detail which are processed by the user. By the way user need to register into the bank application. A new user account number will be generated where user can deposit initial amount. A huge amount of transactional data has to be collected for customer segmentation and products classification. From the collected dataset the long tail products and cold products will be clustered and classified based on the sales data. Here the most popular products which are getting sold in the online markets is classified which comes under long tail group. Whereas some products which are popular once in trend but was not sold for recent days , these criteria are considered and data preprocess will be done. For a group of customers who came into ecommerce application their previous transactions will be obviously null. For this new customers our recommendation system will provide some valuable products recommendation to attain margin for the retailers. In retailers perspective some cold products will be pushed to recommender platform for the new customers. Thus we can solve the cold start problem by recommending the users some products from cold product category. In-order to solve the long tail problem from the categorized group of

products, most selling products has to be collected and some product descriptions will be compared and based on the matching patterns these relevant products will be recommended to the customer. In-order to push the unsold products for a long period based on the customer's request some attractive offers should be promoted. So we implement a product combo offer logic to make the sale possible . Thus long tail and cold start problem has been handled.

6.1 Modules

- User Authentication
- Data Preprocessing
- Cold Product Recommendation
- Combo Products Recommendation

Module explanation

User Authentication

Initially user need to create their own account and register their basic detail in the server. Here whereas the Database server will maintain the user personal detail and all the transaction detail which are processed by the user. By the way user need to register into the bank application. A new user account number will be generated where user can deposit initial amount

Data preprocessing

A huge amount of transactional data has to be collected for customer segmentation and products classification. From the collected dataset the long tail products and cold products will be clustered and classified based on the sales data. Here the most popular products which are getting sold in the online markets is classified which comes under long tail group. Whereas some products which are popular once in trend but was not sold for recent days , these criteria are considered and data preprocess will be done.

Cold Product Recommendation

For a group of customers who came into ecommerce application their previous transactions will be obviously null. For this new customers our recommendation system will provide some valuable products recommendation to attain margin for the retailers. In retailers perspective some cold products will be pushed to recommender platform for the new customers. Thus we can solve the cold start problem by

recommending the users some products from cold product category.

Combo Products Recommendation

In-order to solve the long tail problem from the categorized group of products, most selling products has to be collected and some product descriptions will be compared and based on the matching patterns these relevant products will be recommended to the customer. In-order to push the unsold products for a long period based on the customer's request some attractive offers should be promoted. So we implement a product combo offer logic to make the sale possible . Thus long tail and cold start problem has been handled.

CHAPTER 7

CODING AND TESTING

7.1 CODING

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

- Naming conventions

- Value conventions

Script and comment procedure

Message box format

Exception and error handling

NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.

- Proper validation of values in the field.

- Proper documentation of flag values.

SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to

the user. They are as follows:

X – User has performed illegal operation.

! – Information to the user.

7.3 TEST PROCEDURE

SYSTEM TESTING:

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

Static analysis is used to investigate the structural properties of the Source code.

Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.4 TEST DATA AND OUTPUT

UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

Performance Test

Stress Test

Structure Test

PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.

- Execute all loops at their boundaries and within their operational bounds.

- Exercise internal data structures to assure their validity.

- Checking attributes for their correctness.

- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and

tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

7.5 TESTING TECHNIQUES / TESTING STRATEGIES

TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the

specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

Testing begins at the module level and works “outward” toward the integration of the entire computer based system.

Different testing techniques are appropriate at different points in time.

The developer of the software and an independent test group conducts testing.

Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- * The function or performance characteristics confirm to specifications and are accepted.
- * A validation from specification is uncovered and a deficiency created.

USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system.

Input screen design.

Output screen design.

CHAPTER 8

RESULTS AND DISCUSSIONS

A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system.

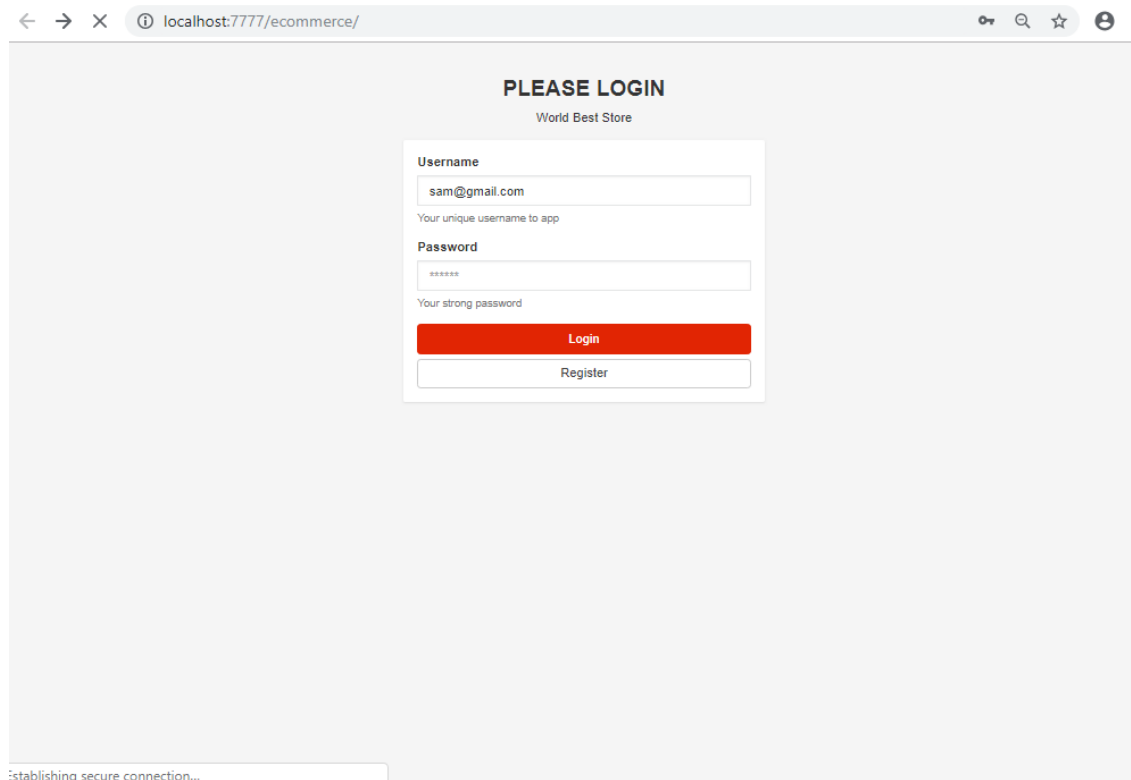


FIG 8.1: Login Page:

After the accessing the account it displays the product either in combo or in offer and users specifies their choice on their basis When the user enter the product name it shows with combo offers on the same price and in stock products along with them. Now, After completing the this process the next step is payment which the way can user own the product it displays with the options like to enter the bank username and bank password.

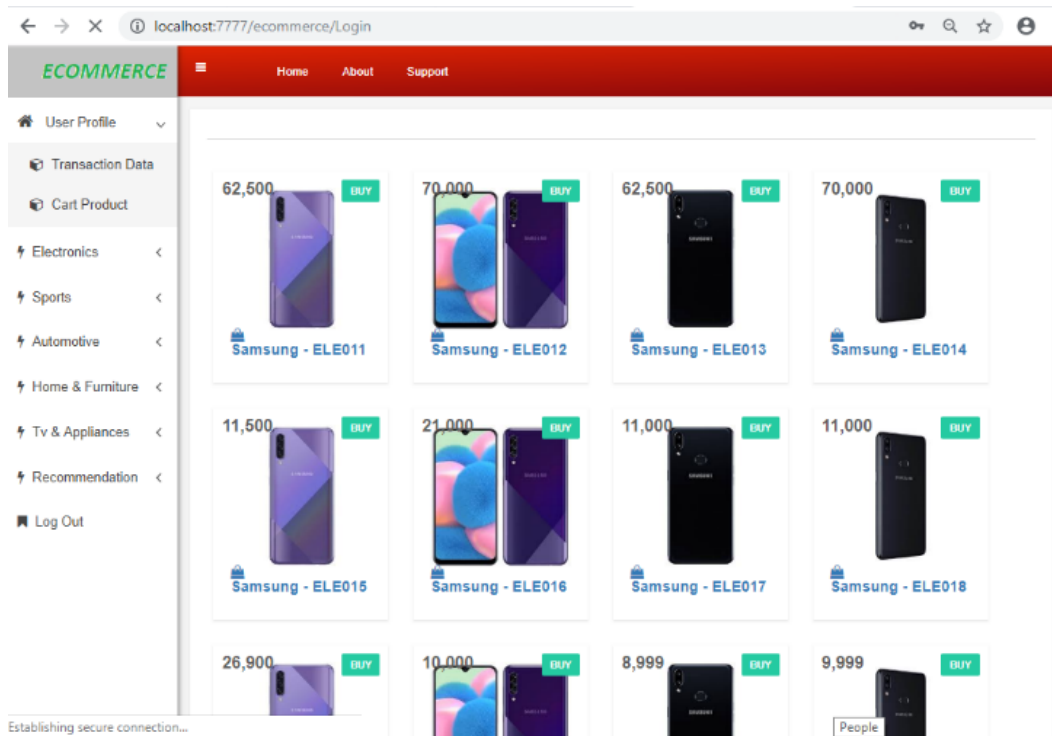


FIG 8.2: Product Page

After generating the products it displays like one plus one offer like below image. It added or it shows the one plus one offer while it displaying the product can access or it maintain by the server.

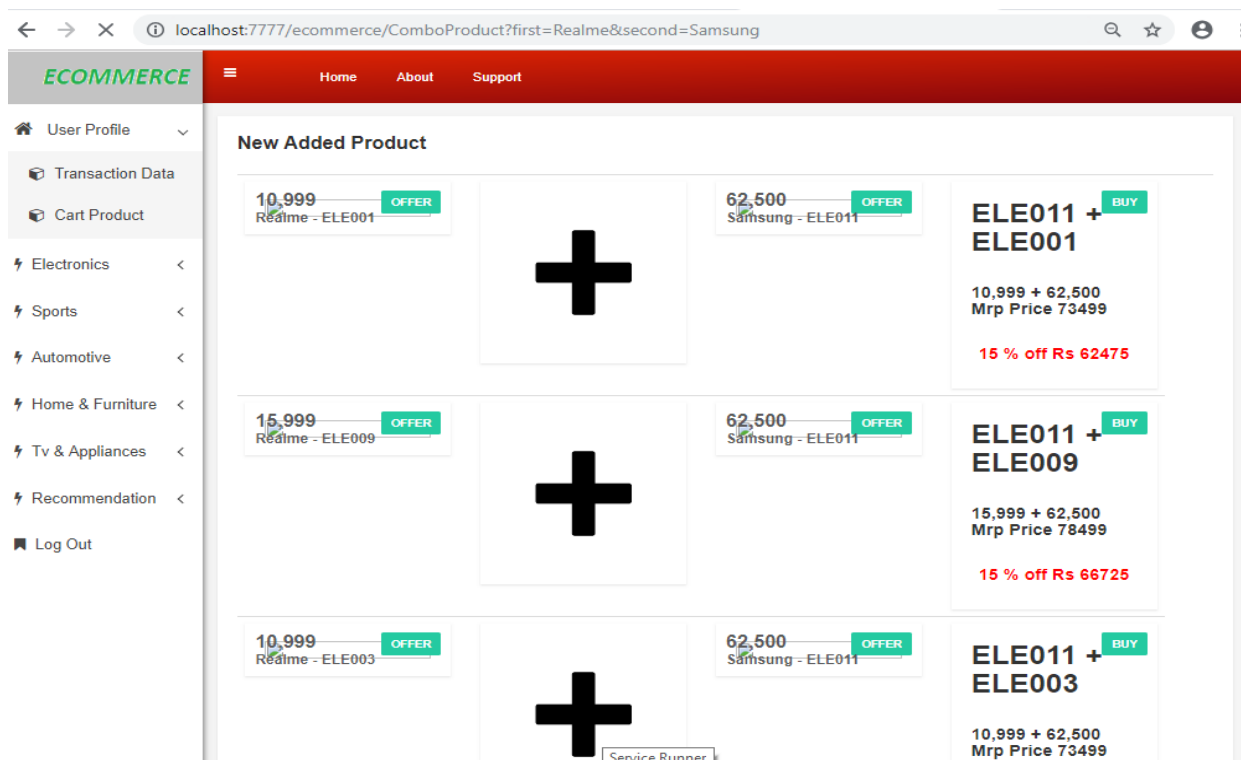


FIG 8.3: Offer Page:

Now these all information recorded or stored in database and user address also stored in database. Here the database is MySQL.

Here the database can add the user id, password, email, mobile number also

The figure consists of three screenshots of the MySQL-Front application interface, showing different database tables.

MySQL-Front - localhost - bankadmindb.userinfo (Data Browser)

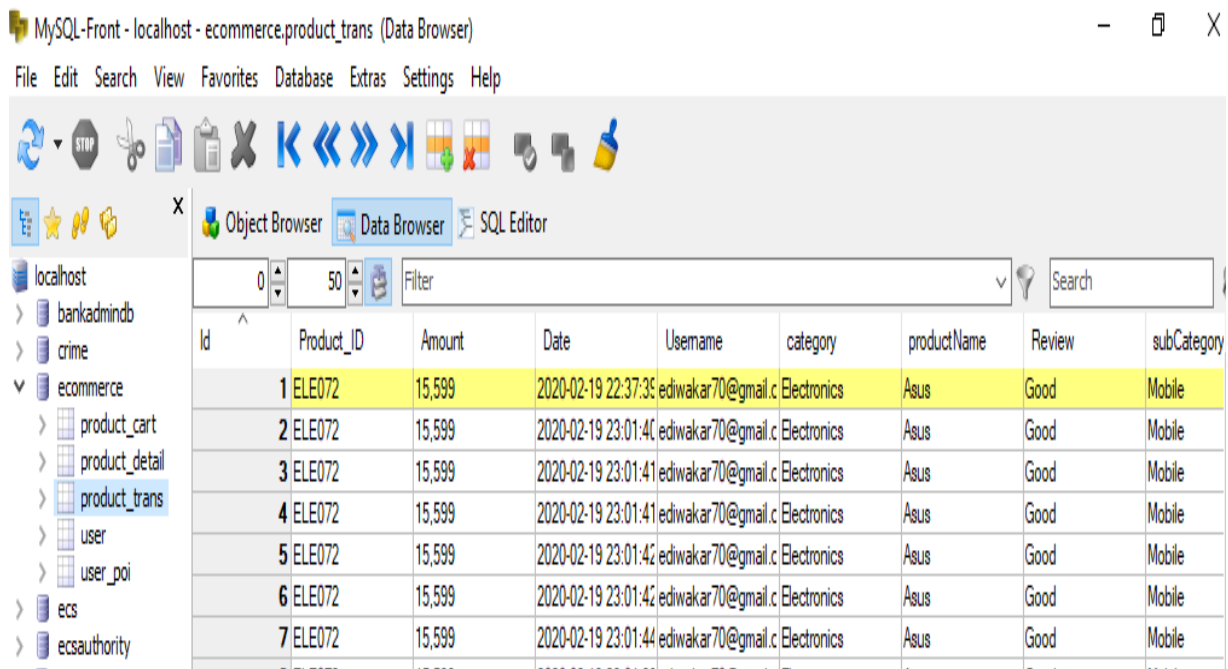
accountno	pinpass	amount	opass	transferamount
06797274	3399	86753	#1XA+42	null
14543675	0438	384334	SC1oiK1z	929
30998432	2906	981258	IDvsEEHc	null
72483896	1411	10000	null	null
75372272	7915	527798	null	null
81450999	2035	9689672	null	null
96707515	5455	571544	Bmfwm7	null
85339945	9192	19775483	LRCMa8/7	null

MySQL-Front - localhost - ecommerce (Object Browser)

Name	Type	Records	Size	Last Update	Extras	Comment
product_cart	InnoDB	~8	16 KB	???		
product_detail	InnoDB	~517	112 KB	???		
product_trans	InnoDB	~23	16 KB	???		
user	InnoDB	~9	16 KB	???		
user_poi	MyISAM	46	3,709 B	2/29/2020 10:...		

FIG :8.4 Database Pages

The database can also include the user owned products also which the both combo and offered. It shows with complete detail in below image.



The screenshot shows the MySQL-Front interface with the 'product_trans' table selected in the 'ecommerce' database. The table contains 7 rows of data, all with 'ELE072' as the Product_ID and 'Asus' as the product name. The first row is highlighted in yellow.

Id	Product_ID	Amount	Date	Username	category	productName	Review	subCategory
1	ELE072	15,599	2020-02-19 22:37:35	ediwakar70@gmail.c	Electronics	Asus	Good	Mobile
2	ELE072	15,599	2020-02-19 23:01:40	ediwakar70@gmail.c	Electronics	Asus	Good	Mobile
3	ELE072	15,599	2020-02-19 23:01:41	ediwakar70@gmail.c	Electronics	Asus	Good	Mobile
4	ELE072	15,599	2020-02-19 23:01:41	ediwakar70@gmail.c	Electronics	Asus	Good	Mobile
5	ELE072	15,599	2020-02-19 23:01:42	ediwakar70@gmail.c	Electronics	Asus	Good	Mobile
6	ELE072	15,599	2020-02-19 23:01:42	ediwakar70@gmail.c	Electronics	Asus	Good	Mobile
7	ELE072	15,599	2020-02-19 23:01:44	ediwakar70@gmail.c	Electronics	Asus	Good	Mobile

FIG:8.5 User Details Page

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

Both cold-start recommendation and long-tail recommendation are challenging problems in the community. To the best of our knowledge, this work is the first one which challenges both coldstart recommendation and long-tail recommendation in a unified optimization problem. Extensive experiments on four real-world datasets verify the effectiveness of the proposed method. This paper shows that one can use side information to warm-up the recommender system when there is no available historical records. We also find that considering long-tail items in the process of coldstart can be beneficial. In fact, our ideas of independently handling the short-head items and long-tail items can also be used in regular recommendations (relative to cold-start recommendation). It is one of the work we will study in the future.

FUTURE ENHANCEMENT

On the basis of the traditional classification regression algorithm

The EN-LDA model uses the idea of the theme model to refine the mapping of the LDA algorithm, mapping the breadth of the theme

The long tail data diversity, and tries to solve the long tail problems. From the results of the experiment

It can be seen that the LDA model shows a very strong performance in the long tail evaluation criterion.

The indicators have verified the model for long tail data hypersensitivity and high utilization.

Reflecting that recommendation and long tail phenomenon are not two irreconcilable

REFERENCES

- [1] Z. Cheng and J. Shen, “Just-for-me: An adaptive personalization system for location-aware social music recommendation,” in ICMR, 2014, p. 185.
- [2] C. Zhiyong and S. Jialie, “On effective location-aware music recommendation,” ACM TIST, vol. 34, no. 2, pp. 13:1–13:32, 2016.
- [3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in WWW. ACM, 2001, pp. 285–295.
- [4] G. Linden, B. Smith, and J. York, “Amazon. com recommendations: Item-to-item collaborative filtering,” IEEE Internet computing, vol. 7, no. 1, pp. 76–80, 2003.
- [5] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.
- [6] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: an open architecture for collaborative filtering of netnews,” in CSCW. ACM, 1994, pp. 175–186.
- [7] Y. Hu, X. Yi, and L. S. Davis, “Collaborative fashion recommendation: a functional tensor factorization approach,” in ACM MM. ACM, 2015, pp. 129–138.
- [8] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, 2009.
- [9] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in ACM RecSys. ACM, 2010, pp. 135–142.
- [10] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. SchmidtThieme, “Multi-relational matrix factorization using bayesian personalized ranking for social network data,” in ACM WSDM. ACM, 2012, pp. 173–182.
- [11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” in WWW. WWW Steering Committee, 2017, pp. 173–182.
- [12] L. Shi, W. X. Zhao, and Y.-D. Shen, “Local representative-based matrix factorization for cold-start recommendation,” ACM TOIS, vol. 36, no. 2, p. 22, 2017.
- [13] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, “Addressing cold-start problem in recommendation systems,” in ACM IMCOM. ACM, 2008, pp. 208–211.
- [14] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in ACM SIGIR. ACM, 2002, pp. 253–260.
- [15] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. SchmidtThieme, “Learning attribute-to-feature mappings for cold-start recommendations,” in ICDM. IEEE, 2010, pp. 176–185.
- [16] Z.-K. Zhang, C. Liu, Y.-C. Zhang, and T. Zhou, “Solving the cold-start problem in recommender systems with social tags,” EPL, vol. 92, no. 2, p. 28002, 2010.

APPENDIX

A. Paper Acceptance Letter

B. PLAGIARISM REPORT



Don't Worry! This report is 100% safe & secure. It's not available publicly and it's not accessible by search engines (Google, Yahoo, Bing, etc)

Sentence

The main objective of this project is discovering interesting items from near-infinite inventory and exhibiting them to potential users. On this case we solving or introducing the two issues of crippling the recommender systems. one is the way to deal with new clients and the other is the means by which to astonish clients. This project undergoes on Hadoop and java platform which having both hardware and software environment to execute the program. This java platform manages the java API and JVM (java virtual machine) where, the java API handles the large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets and the Java Virtual Machine (JVM) is a virtual machine that allows a computer to run Java programs, as well as programs written in other languages that are also compiled to. This paper proposing detailed work based on the problems due to cold start and long tail, products in the ecommerce retailers getting sold out so soon as well as recommending the products to the customers of same kind based on the previous purchases by using clustering. So that, the long tail problem solved by recommending relevant item and product that are not sold for long period gets attractive offer based on the user attention.. Keywords: Long tail, java API, JVM, Hadoop, Crippling ————— 2 .

INTRODUCTION: The success of any kind of social epidemic is heavily dependent on the involvement of people with a particular and rare set of social gifts. Specifically, a recommender system discovers the interesting items from a near-infinite inventory and exhibits them to potential users. In other words, a recommender system must know people to find out potential users, must accumulate information to pick up interesting items, and must persuade the potential users to buy the recommended items. As a result, the success of on-line retailing is heavily dependent on the recommender system. Generally, Online goods has received a 35% of their revenue from recommendations, and 75% of what people watch in Netflix is as a result of recommendations. Since there is no direct information available for cold-start problem, we learn from side information. Specifically, we leverage the social data of users friend groups, special interest groups and page likes as side information in our experiments. At first, mapping functions which map side information to user-item relationship are learned on the training dataset. For the long tail recommendation, we break down the interesting elements in general into two parts: a low range part for short head elements and a sparse part for long tail components. The parts are independently disclosed at the training stage and carried over to the final recommendation for new users.. Given a system of near-infinite inventory, popular items are only a small fraction of the total items. These items, however, have positive relationship with a majority of users. This project undergoes on Hadoop and java platform which having both hardware and software environment to execute the program. This java platform manages the java API and JVM (java virtual machine) where, the java API handles the large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets and the JVM is described by a specification that formally describes what is required in a JVM implementation With a specification, the interoperability of Java programs in different. 3. LITERATURE STUDY 3.1: Relational Learning via Latent Social Dimensions: In this paper Social media like blogs, Facebook, Flickr etc. present data in a network format and not in a classic IID distribution. To remedy the interdependence between data instances, relational learning was proposed and collective inference based on network connectivity was used for prediction.. To address the interdependence between data instances, relational learning has been proposed and collective inference based on network connectivity is adopted for prediction; However, an actor can change due to various factors, such as Connecting former students, colleagues who live in the same city or share similar activities. However, social media connections are often multi-dimensional. An actor can connect to another actor for a number of reasons, such as: E.g. B. alumni, colleagues who live in the same city, share similar interests, etc. Collective conclusions usually do not distinguish these associations.. In this work we propose to extract latent social dimensions based on network information and then utilize them as characteristics for discriminative learning. 3.2: Multi-Relational Matrix Factorization using Bayesian Personalized Ranking for Social Network Data A key element of the social networks on A key element of internet social networks like Facebook and Flickr is that they encourage users to make connections between themselves, other users, and objects. An important task was this. The literature dealing with this data attempts to use social charts to use social graphs to predict user behavior (e.g. Join the a group of personal activities). Specifically, we examine the cold start problem where users only participate in some relationships that we call social relationships, but not the relationship that makes the predictions that we call goal relationships. 3.3: Item-based Collaborative Filtering Recommendation Algorithms: Recommender systems apply knowledge discovery techniques to the recommendation systems apply knowledge discovery techniques to the problem of providing personalized recommendations for information, products, or services during live. In this article, we analyze three different algorithms for generating recommendations.. The huge increase in the information available and the number of visitors to websites in recent years has posed a number of key challenges for recommender systems. These are:

creating high quality recommendations, running many recommendations per second for millions of users and items, and achieving high coverage given the data chariy. With traditional collaborative filter systems, the workload increases with the number of participants. 3.4: Neural Collaborative Filtering: In recent years, deep neural networks have been immensely successful in speech recognition, computer vision, and natural language. However, the exploration of deep neural networks in recommender systems has received relatively less scrutiny. In this work, we strive to develop neural network-based techniques to address the key problem in recommendation-collaborative filtering-on the basis of implicit feedback. Although some recent works have used deep learning as a recommendation, they have used it mainly to model additional information, such as: B. Text descriptions of elements and acoustic characteristics of music. When it comes to model the key factor in collaborative filtering — the interaction between user and item features, they still

long tail problems. From the results of the experiment • It can be seen that the LDA model shows a very strong performance in the long tail evaluation criterion. • The indicators have verified the model for long tail data hypersensitivity and high utilization. • Reflecting that recommendation and long tail phenomenon are not two irreconcilable opposition application scenarios.

Report Title:	project report
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/229218b61ff1b763da501bbf137d4c78b562b1614581881
Report Generated Date:	01 March, 2021
Total Words:	2160
Total Characters:	13828
Keywords/Total Words Ratio:	80.59%
Excluded URL:	No
Unique:	94%
Matched:	6%

Sentence wise detail:

Keywords Density		
One Word	2 Words	3 Words
tion 8.98%	long tail 1.36%	neural collaborative filtering 0.42%
user 3.39%	recommender system 1.02%	latent social dimensions 0.34%
work 2.2%	collaborative filtering 0.85%	recommendation long tail 0.25%
recommendation 2.03%	cold start 0.59%	long tail recommendation 0.25%
tail 1.86%	social network 0.51%	long tail problem 0.25%

Plagiarism Report

By check-plagiarism.com

C. PUBLICATION PAPER

Recommending E-commerce products Using Transaction Data

Kavuri Sai Krishna¹, Katta Sumanth², Dr.G.Meera Gandhi³

¹ B.E. Computer science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

² B.E. Computer science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

³ Dr. G. Meera Gandhi, Assistant Professor, Department of Computer Science and Engineering, SIST, Chennai, India.

1. ABSTRACT:

The main objective of this project is discovering interesting items from near-infinite inventory and exhibiting them to potential users. On this case we solving or introducing the two issues of crippling the recommender systems. one is the way to deal with new clients and the other is the means by which to astonish clients. This project undergoes on Hadoop and java platform which having both hardware and software environment to execute the program. This java platform manages the java API and JVM (java virtual machine) where, the java API handles the large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets and the Java Virtual Machine (JVM) is a virtual machine that allows a computer to run Java programs, as well as programs written in other languages that are also compiled to. This paper proposing detailed work based on the problems due to cold start and long tail, products in the ecommerce retailers getting sold out so soon as well as recommending the products to the customers of same kind based on the previous purchases by using clustering. So that, the long tail problem solved by recommending relevant item and product that are not sold for long period gets attractive offer based on the user attention..

Keywords: Long tail, Java API, JVM, Hadoop, Crippling

2. INTRODUCTION:

The success of any kind of social epidemic is heavily dependent on the involvement of people with a particular and rare set of social gifts. Specifically, a recommender system discovers the interesting items from a near-infinite inventory and exhibits them to potential users. In other words, a recommender system must know people to find out potential users, must accumulate information to pick up interesting items, and must persuade the potential users to buy the recommended items. As a result, the success of on-line retailing is heavily dependent on the recommender system. Generally, Online goods has received a 35% of their revenue from recommendations, and 75% of what people watch in Netflix is as a result of recommendations. Since there is no direct information available for cold-start problem, we learn from side information. Specifically, we leverage the social data of users friend groups, special interest groups and page likes as side information in our experiments. At first, mapping functions which map side information to user-item relationship are learned on the training dataset. For the long tail recommendation, we break down the interesting elements in general into two parts: a low range part for short head elements and a sparse part for long tail components. The parts are independently disclosed at the training stage and carried over to the final recommendation for new users.. Given a system of near-infinite inventory, popular items are only a small fraction of the total items. These items, however, have positive relationship with a majority of users. This project undergoes on Hadoop and java platform which having both hardware and software environment to execute the program. This java platform manages the java API and JVM (java virtual machine) where, the java API handles the large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets and the JVM is described by a specification that formally describes what is required in a JVM implementation. With a specification, the interoperability of Java programs in different.

3. LITERATURE STUDY

3.1: Relational Learning via Latent Social Dimensions:

In this paper Social media like blogs, Facebook, Flickr etc. present data in a network format and not in a classic IID distribution. To remedy the interdependence between data instances, relational learning was proposed and collective inference based on network connectivity was used for prediction.. To address the interdependence between data instances, relational learning has been proposed and collective inference based on network connectivity is adopted for prediction; However, an actor can change due to various factors, such as Connecting former students, colleagues who live in the same city or share similar activities. However, social media connections are often multi-dimensional. An actor can connect to another actor for a number of reasons, such as: E.g. B. alumni, colleagues who live in the same city, share similar interests, etc. Collective conclusions usually do not distinguish these associations.. In this work we propose to extract latent social dimensions based on network information and then utilize them as characteristics for discriminative learning.

3.2: Multi-Relational Matrix Factorization using Bayesian Personalized Ranking for Social Network Data

A key element of the social networks on A key element of internet social networks like Facebook and Flickr is that they encourage users to make connections between themselves, other users, and objects. An important task was this. The literature dealing with this data attempts to use social charts to use social graphs to predict user behavior (e.g. Join the a group of personal activities). Specifically, we examine the cold start problem where users only participate in some relationships that we call social relationships, but not the relationship that makes the predictions that we call goal relationships.

3.3: Item-based Collaborative Filtering Recommendation Algorithms:

Recommender systems apply knowledge discovery techniques to the recommendation systems apply knowledge discovery techniques to the problem of providing personalized recommendations for information, products, or services during live. In this article, we analyze three

different algorithms for generating recommendations.. The huge increase in the information available and the number of visitors to websites in recent years has posed a number of key challenges for **recommender** systems. These are: creating high quality recommendations, running many recommendations per second for millions of users and items, and achieving high coverage **given** the data chary. **With traditional** collaborative filter systems, the workload increases with the number of participants.

3.4: Neural Collaborative Filtering:

In recent years, deep neural networks have been immensely successful in speech recognition, computer vision, and natural language. However, the exploration of deep neural networks in recommender systems has received relatively less scrutiny. In this work, we strive to develop **neural network-based** techniques to address the key problem in recommendation-collaborative filtering-on the basis of implicit feedback. Although some recent **works have** used deep learning as a recommendation, they have used it **mainly to** model additional information, such as: B. **Text** descriptions of elements and acoustic characteristics of music. When it comes to model the key factor in collaborative filtering — the interaction between user and item features, they still resorted to matrix factorization

3.5: A Neural Collaborative Filtering Model with Interaction-based Neighborhood:

Recently, deep neural networks have been widely applied to recommender systems. A representative work consists in using deep learning to model complex user-element. Similar to traditional latent factor models by factoring user-element interactions, **however**, they are **usually** ineffective at capturing localized information such as **The neighborhood** are important in recommending systems to **supplement** the user element interaction information. Based on this consideration, we propose a novel, Based on this consideration, we propose a novel Neighborhood-based Neural Collaborative Filtering (NNCF).

4. INFERENCES FROM LITERATURE SURVEY

4.1: Relational Learning via Latent Social Dimensions:

The network is highly dynamic and might consists of multiple entities. To capture different affiliations among actors in a network, we propose to extract latent social dimensions via modularity maximization.

The diversity of connections does not necessarily indicate that two connected users would share certain class label. This kind of information is rare or not refined enough for to be informative.

4.2: Multi-Relational Matrix Factorization using Bayesian Personalized Ranking for Social Network Data

The Collaborative filtering is the most popular approach to building recommender systems. The methods do not consider the propagation of confident. The cold start user is not connected to the social network, then social network based approaches have no additional information to improve the quality of recommendation for that user.

4.3: : Item-based Collaborative Filtering

Recommendation Algorithms:

on items, the unobserved entries can be just missing data and there is a natural scarcity of negative feedback. The recommendation problem with implicit feedback is formulated as the problem of estimating the scores of unobserved entries in Y , which are used for ranking the items. Model-based approaches assume that data can be generated

4.4: Neural Collaborative Filtering:

The performance is poor. The interplay of the music popularity, music content, and users' contextual music preferences is complex and cannot be effectively represented by using a simple gathering. The key objective of location-aware music recommender system is to satisfy users' music information needs based on user's location with minimum user efforts on providing feedbacks. Apparently, accurate users' preferences acquisition is essential to the performance of a music recommender system.

4.5: A Neural Collaborative Filtering Model with Interaction-based Neighborhood:

MF is that it adopts a simple linear factorization, which may not be sufficient to model the complex user-item. The method may not work well when the degree of a vertex is either too large or too small. A novel neural network model NNCF, which jointly characterized both user-item interactions and neighborhood information for recommendation.

5. PROPOSED SYSTEM

Based on the problems due to cold start and long tail, products in the ecommerce retailers getting sold out so soon. And some kind of products remains stagnant for a long duration. Thus to overcome these problems we are going to track the selling of products as well as recommending the products to the customers of same kind based on the previous purchases by using clustering and Finally long tail problem solved by recommending relevant item and product that are not sold for long period gets attractive offer based on the user attention. We use hadoop environment to do manipulation with the products dataset.

6. RESULTS AND DISCUSSIONS

The main process running by the below architecture diagram.

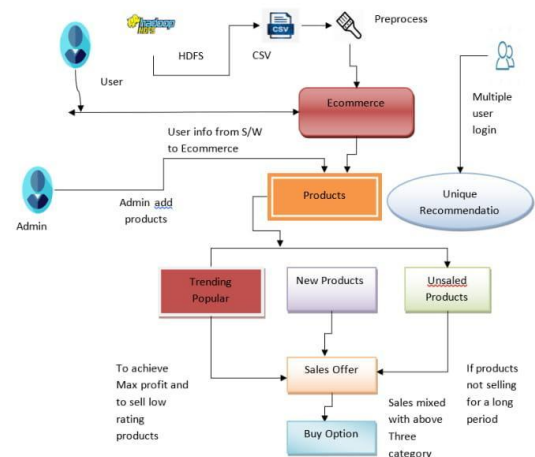


Fig:6.1 architecture diagram

This is the login page which the users can access their account.

Fig:6.2 Login page

After the accessing the account it displays the product either in combo or in offer and users specifies their choice on their basis.

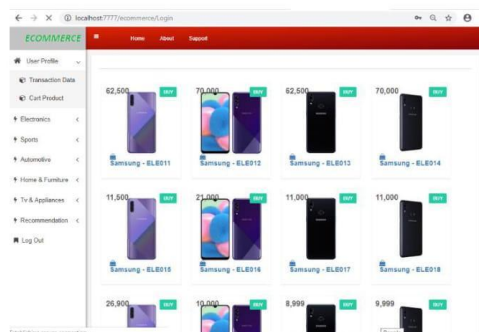


Fig:6.3 Product pannel

When the user enter the product name it shows with combo offers on the same price and in stock products along with them.

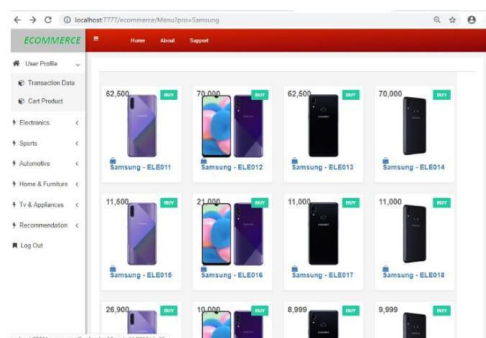


Fig:6.4 Offer Page

Now, After completing the this process the next step is payment which the way can user own the product it displays with the options like to enter the bank username and bank password.

Fig:6.5 Payment Page

Now the admin can handle the this pages and insert the products to combo offer and recommendation and also set the generating combo products .Below the figure shows the diagram of admin page.

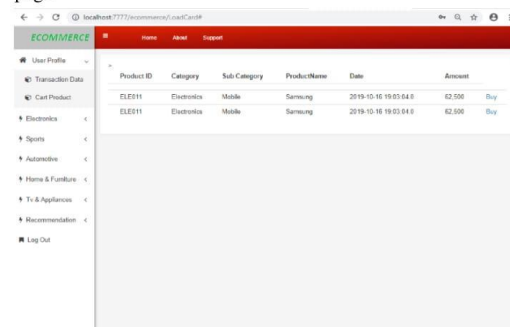


Fig:6.6 Admin Page

After generating the products it displays like one plus one offer like below image.

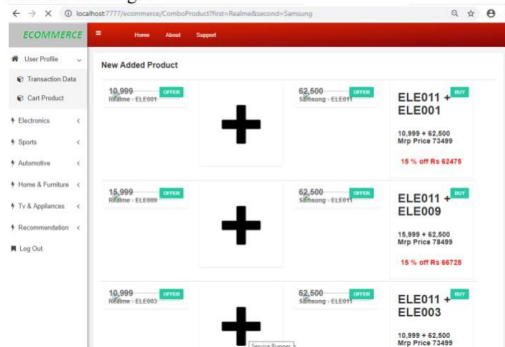


Fig:6.7 Offer page

Now these all information recorded or stored in database and user address also stored in database. Here the database is MySQL.

Here the database can add the user id, password, email, mobile number also

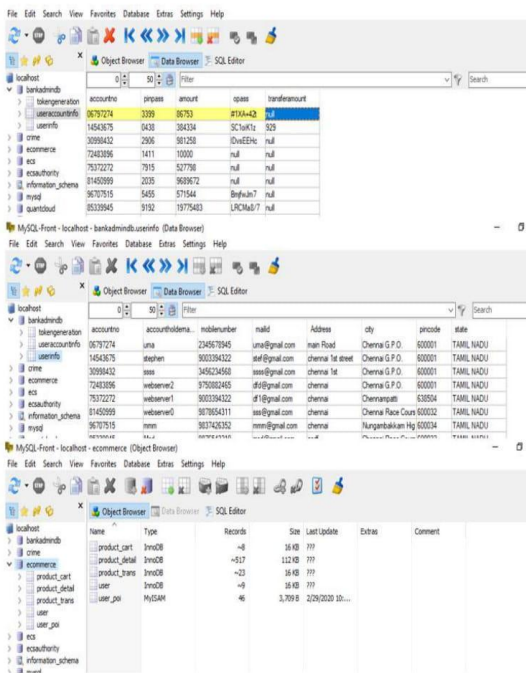


Fig:6.8 Database page

The database can also include the user owned products also which the both combo and offered. It shows with complete detail in below image.

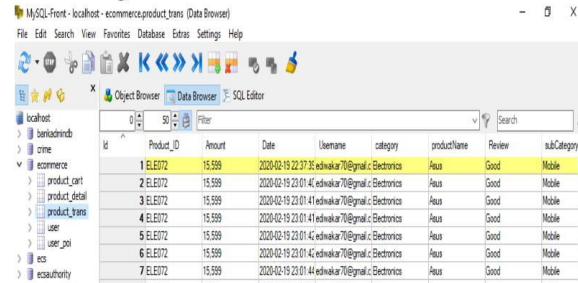


Fig:6.9 User details page

7. CONCLUSION

Both cold-start recommendation and long-tail recommendation are challenging problems in the community. To the best of our knowledge, this work is the first one which challenges both coldstart recommendation and long-tail recommendation in a unified optimization problem. Extensive experiments on four real-world datasets verify the effectiveness of the proposed method. This paper shows that one can use side information to warm-up the recommender system when there is no available historical recorders. We also find that considering long-tail items in the process of coldstart can be beneficial. In fact, our ideas of independently handling the short-head items and long-tail items can also be used in regular recommendations (relative to cold-start recommendation). It is one of the work we will study in the future.

8. FUTURE ENHANCEMENT

- On the basis of the traditional classification regression algorithm
- The EN-LDA model uses the idea of the theme model to refine the mapping of the LDA algorithm, mapping the breadth of the theme
- The long tail data diversity, and tries to solve the long tail problems. From the results of the experiment
- It can be seen that the LDA model shows a very strong performance in the long tail evaluation criterion.
- The indicators have verified the model for long tail data hypersensitivity and high utilization.
- Reflecting that recommendation and long tail phenomenon are not two irreconcilable opposition application scenarios.

9. REFERENCES:

- [1]Malcolm Gladwell es autor **de cinco bestsellers del New York Times**: The Tipping Point, **Blink**, Outliers, **What the Dog Saw** y David and Goliath,2000.
- [2] F. Ricci, L. Rokeach (1877-1957), the fourth **Rebbe of Belzer**; Elazar Rokeach (c. 1176-1238), Talmudist and **Kabbalist**
- [3] Ramunas Adomavicius is **on Facebook Connect with Facebook to communicate with Ramunas Adomavicius and others who may** time domain spectroscopy systems **that are** activated by femtosecond lasers **in** the infrared wavelength range are **being investigated .,**" IEEE TKDE, vol. 17, no. 6, 6pp. 734–749, 2005.
- [4] Corpus ID: 5176749 Accuracy isn't always good: How **precision metrics** have hurt recommender systems @inproceedings {McNee2006AccurateIN, title = {Accuracy isn't always good: How **precision metrics** have hurt **systems recommendation**}, author = {Sean M.McNee and J. Riedl and J.Konstan}, year = {2006}}
- [5] Z. Cheng and J. Shen, "Just-for-me: An adaptive personalization system for location-aware social music recommendation," in ICMR, 2014, p. 185.
- [6] C. Zhiyong and S. Jialie, "On effective location-aware music recommendation," ACM TIST, vol. 34, no. 2, pp. 13:1–13:32, 2016.
- [7]**Element-based collaborative filtering recommendation algorithms** Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl f sarw ar, k arypis, k onstan, riedl g GroupLens Research Group / Army HPC Research Center @ edu Department of Science **Computing** and Engineering University of Minnesota, Minneapolis, MN 55455 ABSTRACT Recommendation systems .285– 295.
- [8] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Recommendation systems apply knowledge discovery techniques to the problem of providing personalized recommendations for information, products, or services during a live interaction" IEEE Internet computing, 2003.
- [9] **At 27, at Juio**, 1999 · D. Ld berg, D. Nicho ls, B. Ok y D. Terry. Use of collaborative filtering to weave a tapestry of **information**. ACM **Communications**, 35 (12): 61–70, 1992 CrossRef Google Scholar [HGBSO98] Ralf Herbirch, Thore Graepel, Peter Bollmann-Sdorra, and Klaus Obermayer. Learning preference relations**hips** for information retrieval. 1992.
- [10] Resnick, N.Iacovou, M.Suchak, P.Bergstrom and J.Riedl, "**Grouplens**: An Open Architecture for Collaborative Filtering of **Netnews**", ACM Conference on **Computational Collaborative Work**.

A. SOURCE CODE

JAVA

```
package com.ecommerce.controller;

import java.io.File;
import java.io.IOException;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileItemFactory;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

import com.ecommerce.model.Product;
import com.ecommerce.service.AdminImpl;
import com.ecommerce.util.EcommerceConstant;
import com.ecommerce.util.PreProcessing;

/**
 * Servlet implementation class ProductController
 */
@WebServlet("/Product")
public class ProductController extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

```

/**
 * @see HttpServlet#HttpServlet()
 */
public ProductController() {
    super();
    // TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse
 *     response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    System.out.println("Admin Start the Product Preprocessing");
    System.out.println("Pre Processing for the Un Structured Data");
    boolean unStructuredData =
PreProcessing.processUnstructuredData();
    System.out.println("Pre Processing For Un-Structured Completed
and Ser File Has Been Generated");
    try {
        Thread.sleep(3000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    System.out.println("Pre Processing for the Structured Data");
    boolean structureData = PreProcessing.processStructuredData();
    System.out.println("Pre Processing For Structured Data was
Completed and Corresponding Ser Generated");

    System.out.println("-----Cold Data-----");
    try {

```

```

        boolean ColdData = PreProcessing.analyzeCold();
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    System.out.println("Cold Product Identify for Each Product
Category ");

    System.out.println("-----Long Tail Data-----");
    try {
        boolean longTailData = PreProcessing.analyzeLongTail();
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    System.out.println("Long Tail Product is Identify for Each Product
Category");

    request.setAttribute("MSG",
EcommerceConstant.PRODUCT_PROCESSING);
    RequestDispatcher dispatcher =
request.getRequestDispatcher("addproduct.jsp");
    dispatcher.forward(request, response);
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
 * response)
 */
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    // TODO Auto-generated method stub
    Product products = new Product();

```

```

        String productid = request.getParameter("productid");
        String category = request.getParameter("category");
        String subcategory = request.getParameter("subcategory");
        String productname = request.getParameter("productname");
String description = request.getParameter("description");
        String spec = request.getParameter("spec");
        String mrp = request.getParameter("mrp");
        String offer = request.getParameter("offer");


        products.setProductId(productid);
        products.setCategory(category);
        products.setSubCategory(subcategory);
        products.setProductName(productname);
        products.setDescription(description);
        products.setSpec(spec);
        products.setMrp(mrp);
        products.setDiscount(offer);
        products.setRating("4.5");
        products.setBuyers("0");
        AdminImpl.getInstance().addProduct(products);


        request.setAttribute("MSG",
EcommerceConstant.PRODUCT_ADDED);
        RequestDispatcher dispatcher =
request.getRequestDispatcher("addproduct.jsp");
        dispatcher.forward(request, response);

    }

}

```


publication paper report

by Plagiarism Checker

Submission date: 03-Apr-2021 01:00AM (UTC-0400)

Submission ID: 1507679226

File name: publication_paper_1_2_-converted.pdf (648.22K)

Word count: 2480

Character count: 13892

Recommending E-commerce products Using Transaction Data

Kavuri Sai Krishna¹, Katta Sumanth², Mrs. Meera Gandhi³

¹ B.E. Computer science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

² B.E. Computer science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

³ Dr. G. Meera Gandhi, Assistant Professor, Department of Computer Science and Engineering, SIST, Chennai, India.