JSS MAHAVIDYAPEETHA

SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING (SJCE), Mysuru-570006

An Autonomous Institute Affiliated to Visvesvaraya Technological University, Belagavi

"CONNECTING SOCIAL MEDIA TO E-COMMERCE: COLD-START PRODUCT RECOMMENDATION USING MICRO BLOGGING INFORMATION"

Project submitted in partial fulfillment of curriculum prescribed for the award of the degree of

Bachelor of Engineering

in

Information Science and Engineering

By

Alapati Bharathkrishna (4JC13IS006)
Lohith S (4JC13IS036)
Rajiv P (4JC13IS046)
Siddhanth Janadri (4JC13IS055)

Under the guidance of
Vanishree Arun
Assistant professor

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

MAY 2017

J S S MAHAVIDYAPEETHA

SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING MYSORE-570006

An Autonomous Institute Affiliated to Visvesvaraya Technological University, Belagavi.

CERTIFICATE

This is to certify that the work entitled "**CONNECTING SOCIAL MEDIA TO E-COMMERCE: COLD-START PRODUCT RECOMMENDATION USING MICRO BLOGGING INFORMATION**" is a bonafied work carried out by **Alapati Bharathkrishna, Lohith S, Rajiv P, Siddhanth Janadri** in partial fulfillment of the award of the degree of Bachelor of Engineering in Information Science and Engineering of Visvesaraya Technological University, Belgaum during the year 2016 – 17. It is certified that all the corrections/suggestions indicated during CIE have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Guide	**Head of the Department**	**Principal**
Vanishree Arun, Assistant Professor, Dept of IS&E, SJCE, Mysore.	**S. K Padma,** Professor and Head, Department of IS&E, SJCE, Mysore.	**T.N. Nagabhushan** Principal, SJCE, Mysore.

Place: **Mysuru**
Date: **04 May 2017**

Examiners: 1. …….

2. ..……

3. .....…….

DECLARATION

We, **Alapati Bharathkrishna, Lohith S, Rajiv P, Siddhanth janadri,** students of final Year Bachelor of engineering with specialization in Information Science & Engineering of Sri Jayachamarajendra College of Engineering, Mysuru, here by declare that the dissertation entitled "**CONNECTING SOCIAL MEDIA TO E-COMMERCE: COLD-START PRODUCT RECOMMENDATION USING MICRO BLOGGING INFORMATION**" has been carried out by us under the guidance of **Vanishree Arun**, Assistant professor, SJCE, Mysuru and submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering by the Visveswaraya Technological University during the academic year 2013-2017. Further, the matter embodied in the dissertation has not been submitted previously by anybody for the award of any degree to any other university.

Alapati Bharathkrishna (4JC13IS006)
Lohith S (4JC13IS036)
Rajiv P (4JC13IS046)
Siddhanth janadri (4JC13IS055)

ACKNOWLEDGEMENT

We consider it a privilege to wholeheartedly express our gratitude and respect to everyone who guided and helped us in the successful completion of the project.

We wish to express our sincere gratitude to **Dr. T N Nagabhushan,** Principal, SJCE and **Dr. S. K Padma**, Head of the Department, IS & E, SJCE for the facilities, guidance and support provided throughout the project work.

It is our foremost duty to thank our guide, **Vanishree Arun**, Assistant Professor in the Information Science and Engineering Department, SJCE, Mysuru for her encouragement, effective guidance, and valuable suggestions right from the very beginning of this project work and up until its completion.

Finally, we are extremely thankful to our family and friends, who have helped us in our work and made this project a successful one.

ABSTRACT

In recent years, the boundaries between e-commerce and social networking have become increasingly blurred. Many e-commerce websites support the mechanism of social login where users can sign on the websites using their social network identities such as their Facebook or Twitter accounts. Users can also post their newly purchased products on microblogs with links to the e-commerce product web pages. In this paper, we propose a novel solution for cross-site coldstart product recommendation, which aims to recommend products from e-commerce websites to users at social networking sites in "cold-start" situations, a problem which has rarely been explored before. A major challenge is how to leverage knowledge extracted from social networking sites for cross-site cold-start product recommendation. We propose to use the linked users across social networking sites and e-commerce websites (users who have social networking accounts and have made purchases on e-commerce websites) as a bridge to map users' social networking features to another feature representation for product recommendation. In specific, we propose learning both users' and products' feature representations (called user embeddings and product embeddings, respectively) from data collected from e-commerce websites using recurrent neural networks and then apply a modified gradient boosting trees method to transform users' social networking features into user embeddings. We then develop a feature-based matrix factorization approach which can leverage the learnt user embeddings for cold-start product recommendation.

Experimental results on a large dataset constructed from the largest Chinese microblogging service SINA WEIBO and the largest Chinese B2C e-commerce website JINGDONG have shown the effectiveness of our proposed framework.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Most studies only focus on constructing solutions within certain e-commerce websites and mainly utilise users' historical transaction records. To the best of our knowledge, cross-site cold-start product recommendation has been rarely studied before. There has also been a large body of research work focusing specifically on the cold-start recommendation problem. In recent years, the boundaries between e-commerce and social networking have become increasingly blurred. Many e-commerce websites support the mechanism of social login where users can sign on the websites using their social network identities such as their Facebook or Twitter accounts. Users can also post their newly purchased products on microblogs with links to the e-commerce product web pages. In this paper, we propose a novel solution for cross-site coldstart product recommendation, which aims to recommend products from e-commerce websites to users at social networking sites in "cold-start" situations, a problem which has rarely been explored before. A major challenge is how to leverage knowledge extracted from social networking sites for cross-site cold-start product recommendation. We propose to use the linked users across social networking sites and e-commerce websites (users who have social networking accounts and have made purchases on e-commerce websites) as a bridge to map users' social networking features to another feature representation for product recommendation. In specific, we propose learning both users' and products' feature representations (called user embeddings and product embeddings, respectively) from data collected from e-commerce websites using recurrent neural networks and then apply a modified gradient boosting trees method to transform users' social networking features into user embeddings. We then develop a feature-based matrix factorization approach which can leverage the learnt user embeddings for cold-start product recommendation.

COLD START RECOMMENDATION: Personalized recommender systems take advantage of users past history to make predictions. The cold start problem concerns the personalized recommendations for users with no or few past history (new users). Providing recommendations to users with small past history becomes a difficult problem for CF models because their learning and predictive ability is limited. Multiple research have been conducted in this direction using hybrid models. These models use auxiliary information (multimodal information, side information, etc.) to overcome the cold start problem.

MICROBLOGGING : A type of blog that lets users publish short text updates. Bloggers can usually use a number of service for the updates including instant messaging, e-mail, or Twitter. The posts are called microposts, while the act of using these services to update your blog is called microblogging.

1.2 PROBLEM STATEMENT

Many e-commerce websites support the mechanism of social login where users can sign on the websites using their social network identities such as their Facebook or Twitter accounts. Users can also post their newly purchased products on microblogs with links to the e-commerce product web pages. In this paper, we propose a novel solution for cross-site cold-start product recommendation, which aims to recommend products from e-commerce websites to users at social networking sites in “cold-start” situations, a problem which has rarely been explored before. A major challenge is how to leverage knowledge extracted from social networking sites for crosssite cold-start product recommendation. We propose to use the linked users across social networking sites and e-commerce websites (users who have social networking accounts and have made purchases on e-commerce websites) as a bridge to map users’ social networking features to another feature representation for product recommendation.

1.3 AIM OF THE PROJECT

We study an interesting problem of recommending products from e-commerce websites to users at social networking sites who do not have historical purchase records, i.e., in "cold-start" situations. We called this problem cross-site cold-start product recommendation. In our problem setting here, only the users' social networking information is available and it is a challenging task to transform the social networking information into latent user features which can be effectively used for product recommendation. To address this challenge, we propose to use the linked users across social networking sites and e-commerce websites (users who have social networking accounts and have made purchases on e-commerce websites) as a bridge to map users' social networking features to latent features for product recommendation.

In specific, we propose learning both users' and products' feature representations (called user embeddings and product embeddings, respectively) from data collected from e-commerce websites using recurrent neural networks and then apply a modified gradient boosting trees method to transform users' social networking features into user embeddings. We then develop a feature-based matrix factorization approach which can leverage the learnt user embeddings for cold start product recommendation.

1.4 OBJECTIVES

The objective is to use the linked users across social networking sites and e-commerce websites (users who have social networking accounts and have made purchases on e-commerce websites) as a bridge to map users' social networking features to another feature representation for product recommendation. In specific, we propose learning both users' and products' feature representations (called user embeddings and product embeddings, respectively) from data collected from e-commerce websites using recurrent neural networks and then apply a modified gradient boosting trees method to transform users' social networking features into user

embeddings. We then develop a feature-based matrix factorization approach which can leverage the learnt user embeddings for cold-start product recommendation.

1.5 SCOPE

Our proposed framework is indeed effective in addressing the cross-site cold-start product recommendation problem. We believe that our study will have profound impact on both research and industry communities. We formulate a novel problem of recommending products from an ecommerce website to social networking users in “cold-start” situations. To the best of our knowledge, it has been rarely studied before. We propose to apply the recurrent neural networks for learning correlated feature representations for both users and products from data collected from an e-commerce website. We propose a modified gradient boosting trees method to transform users’ microblogging attributes to latent feature representation which can be easily incorporated for product recommendation. We propose and instantiate a feature-based matrix factorization approach by incorporating user and product features for cold-start product recommendation.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

Plenty of research has been undertaken in the domain of cold start product recommendation system. Our work is mainly related to three lines of research:

Recommender systems: In recent years, the matrix fatortization approach has received much research interests. With the increasing volume of Web data, many studies focus on incorporating auxiliary information into the matrix factorization approach. Two typical frameworks of such studies are the SVDFeature and Factorization machine.

Cross-domain recommendation: One of the key techniques for cross-domain recommendation is Transfer Learning and the idea is to learn transferable knowledge from the source domain, and further apply it in a target domain.

Social network mining: We follow the early commercial mining studies on social networking websites. Our work is built upon these studies, especially in the areas of cross-domain and coldstart recommendation.

1) Opportunity model for E-commerce recommendation: Right product; right time

AUTHORS: J. Wang and Y. Zhang

Most of existing e-commerce recommender systems aim to recommend the right product to a user, based on whether the user is likely to purchase or like a product. On the other hand, the effectiveness of recommendations also depends on the time of the recommendation. Let us take a user who just purchased a laptop as an example. She may purchase a replacement battery in 2 years (assuming that the laptop's original battery often fails to work around that time)

and purchase a new laptop in another 2 years. In this case, it is not a good idea to recommend a new laptop or a replacement battery right after the user purchased the new laptop. It could hurt the user's satisfaction of the recommender system if she receives a potentially right product recommendation at the wrong time. We argue that a system should not only recommend the most relevant item, but also recommend at the right time.

This paper studies the new problem: how to recommend the right product at the right time? We adapt the proportional hazards modeling approach in survival analysis to the recommendation research field and propose a new opportunity model to explicitly incorporate time in an ecommerce recommender system. The new model estimates the joint probability of a user making a follow-up purchase of a particular product at a particular time. This joint purchase probability can be leveraged by recommender systems in various scenarios, including the zero-query pullbased recommendation scenario (e.g. recommendation on an e-commerce web site) and a proactive push-based promotion scenario (e.g. email or text message based marketing). We evaluate the opportunity modeling approach with multiple metrics. Experimental results on a data collected by a real-world e-commerce website(shop.com) show that it can predict a user's follow-up purchase behavior at a particular time with descent accuracy. In addition, the opportunity model significantly improves the conversion rate in pull-based systems and the user satisfaction/utility in push-based systems.

2) Retail sales prediction and item recommendations using customer demographics at store level

AUTHORS: M. Giering

This paper outlines a retail sales prediction and product recommendation system that was implemented for a chain of retail stores. The relative importance of consumer demographic characteristics for accurately modeling the sales of each customer type are derived and implemented in the model. Data consisted of daily sales information for 600 products at the store level, broken out over a set of non-overlapping customer types. A recommender system was built based on a fast online thin Singular Value Decomposition. It is shown that modeling data at a finer level of detail by clustering across customer types and demographics yields

improved performance compared to a single aggregate model built for the entire dataset. Details of the system implementation are described and practical issues that arise in such real-world applications are discussed. Preliminary results from test stores over a one-year period indicate that the system resulted in significantly increased sales and improved efficiencies. A brief overview of how the primary methods discussed here were extended to a much larger data set is given to confirm and illustrate the scalability of this approach.

3) Amazon.com recommendations: Item-to-item collaborative filtering

AUTHORS: G. Linden, B. Smith, and J. York

Recommendation algorithms are best known for their use on e-commerce Web sites, where they use input about a customer's interests to generate a list of recommended items. Many applications use only the items that customers purchase and explicitly rate to represent their interests, but they can also use other attributes, including items viewed, demographic data, subject interests, and favorite artists. At Amazon.com, we use recommendation algorithms to personalize the online store for each customer. The store radically changes based on customer interests, showing programming titles to a software engineer and baby toys to a new mother. There are three common approaches to solving the recommendation problem: traditional collaborative filtering, cluster models, and search-based methods. Here, we compare these methods with our algorithm, which we call item-to-item collaborative filtering. Unlike traditional collaborative filtering, our algorithm's online computation scales independently of the number of customers and number of items in the product catalog. Our algorithm produces recommendations in real-time, scales to massive data sets, and generates high quality recommendations.

4) The new demographics and market fragmentation

AUTHORS: V. A. Zeithaml

The underlying premise of this article is that changing demographics will lead to a splintering of the mass markets for grocery products and supermarkets. A field study investigated the relationships between five demographic factors-sex, female working status, age, income, and marital status-and a wide range of variables associated with preparation for and execution of supermarket shopping. Results indicate that the demographic groups differ in significant ways from the traditional supermarket shopper. Discussion centers on the ways that changing demographics and family roles may affect retailers and manufacturers of grocery products.

5) We know what you want to buy: A demographic-based system for product recommendation on microblogs

AUTHORS: W. X. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li

Product recommender systems are often deployed by e-commerce websites to improve user experience and increase sales. However, recommendation is limited by the product information hosted in those e-commerce sites and is only triggered when users are performing e-commerce activities. In this paper, we develop a novel product recommender system called METIS, a MErchanT Intelligence recommender System, which detects users' purchase intents from their microblogs in near real-time and makes product recommendation based on matching the users' demographic information extracted from their public profiles with product demographics learned from microblogs and online reviews. METIS distinguishes itself from traditional product recommender systems in the following aspects: 1) METIS was developed based on a microblogging service platform. As such, it is not limited by the information available in any specific e-commerce website. In addition, METIS is able to track users' purchase intents in near real-time and make recommendations accordingly. 2) In METIS, product recommendation is framed as a learning to rank problem. Users' characteristics extracted from

their public profiles in microblogs and products' demographics learned from both online product reviews and microblogs are fed into learning to rank algorithms for product recommendation. We have evaluated our system in a large dataset crawled from Sina Weibo. The experimental results have verified the feasibility and effectiveness of our system. We have also made a demo version of our system publicly available and have implemented a live system which allows registered users to receive recommendations in real time.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 INTRODUCTION

Software requirements specification is a description of a software system to be devel- oped. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later re- design. It should also provide a realistic basis for estimating product costs, risks, and schedules.

3.1.1 FUNCTIONAL REQUIREMENTS

In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. As defined in requirements engineering, functional re- quirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

3.1.2 NON FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional re- quirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 RETRIEVING INPUT

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

3.3 NON FUNCTIONAL REQUIREMENTS

3.3.1 PERFORMANCE

Microblog will provide promptly results of the data using the various software pack- ages available to it. The output should display the latest results at all times, and if it lags behind, the user should be notified. The application should be capable of operating in the background should the user wish to utilize other applications.

3.3.2 RELIABILITY

The software will meet all of the functional requirements without any unexpected behavior. At no time should the output display incorrect or outdated information without alerting the user to potential errors. In this instance error message will be shown.

3.3.3 AVAILABILITY

The software will be available at all times on the users device desktop or laptop, as long as the device is in proper working order. The functionality of the software will depend on any external services such as internet access that are required.

3.3.4 SECURITY

The software should never disclose any personal information of amazon users, and should collect no personal information from its own users. The programs will be performed on a password protected laptop and desktop to ensure maximum security.

3.3.5 MAINTAINABILITY

The software should be written clearly and concisely. The code will be well documented. Particular care will be taken to design the software modularly to ensure that maintenance is easy.

3.4 HARDWARE AND SOFTWARE

3.4.1 HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

3.4.2 SOFTWARE REQUIREMENTS:

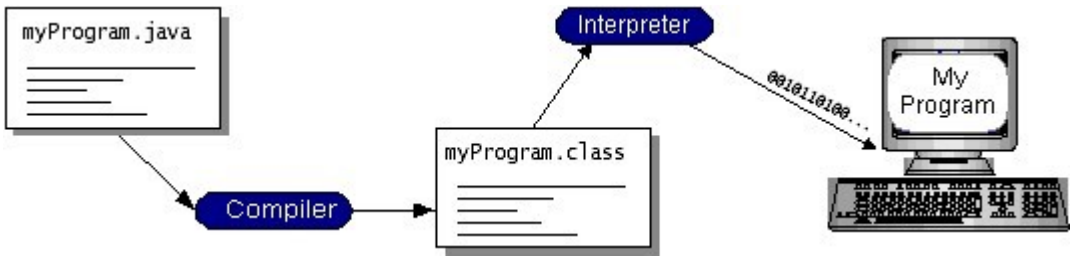
- Operating system : - Windows XP/7.
- Coding Language : JAVA/J2EE
- Data Base : MYSQL

CHAPTER 4

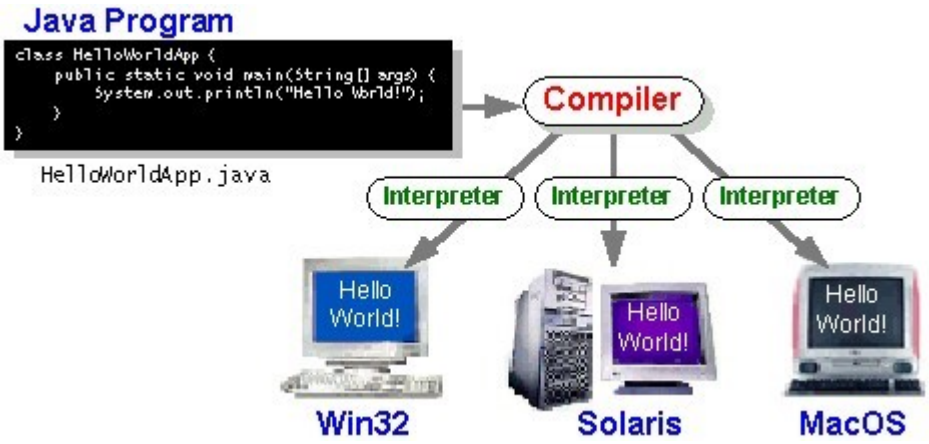
4.1 SOFTWARE ENVIRONMENT

4.1.1 JAVA TECHNOLOGY:

Java technology is both a programming language and a platform. With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



The Java byte codes run as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



4.1.2 THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. 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.

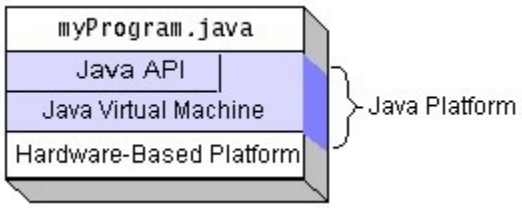
The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the Java VM. 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 of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-intime byte code compilers can bring performance close to that of native code without threatening portability.

4.2 ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers

has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

4.3 JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

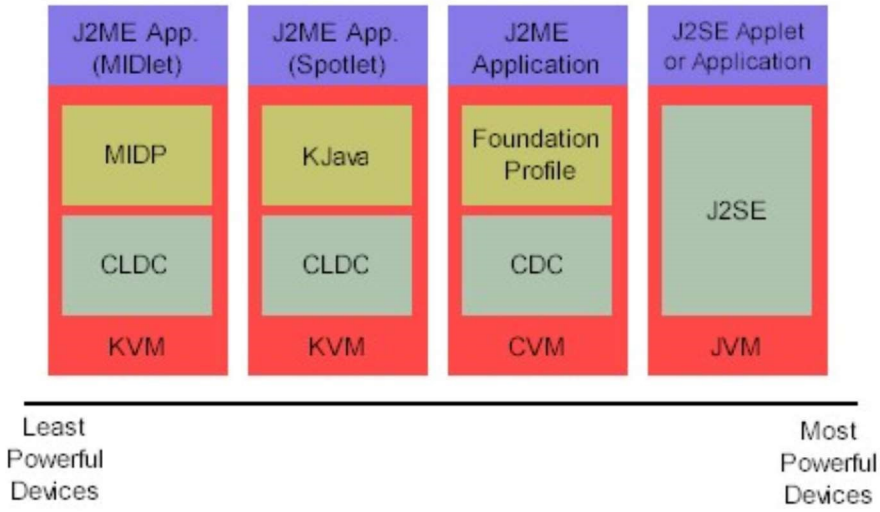
JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC.

4.4 J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

4.4.1 GENERAL J2ME ARCHITECTURE :

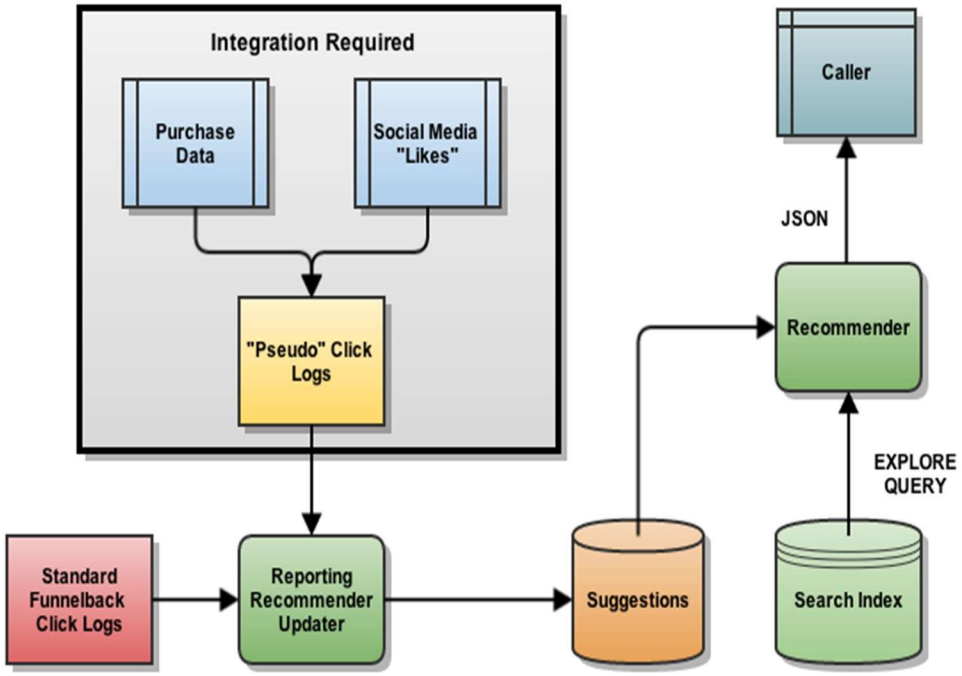


J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

CHAPTER 5

METHODOLOGY

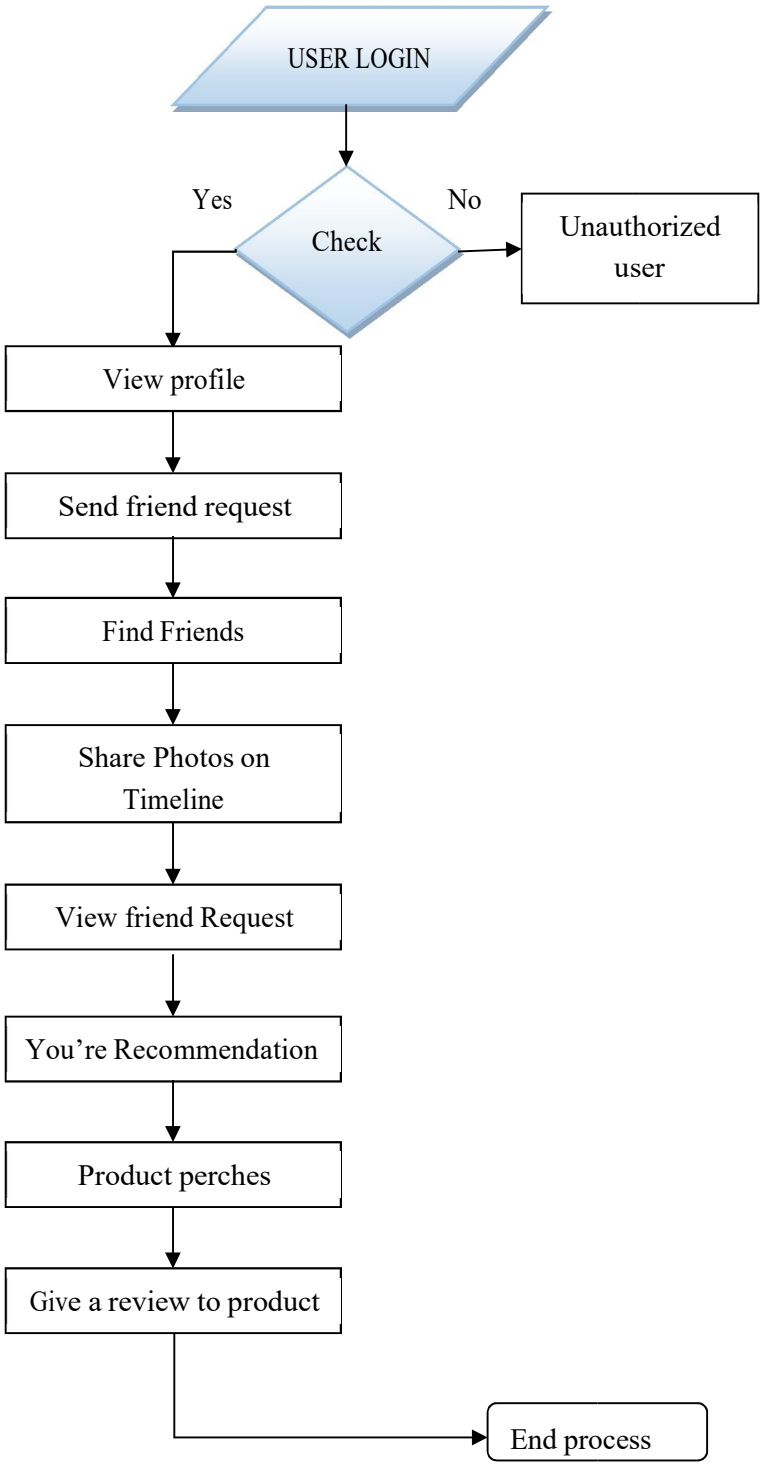
5.1 SYSTEM ARCHITECTURE:



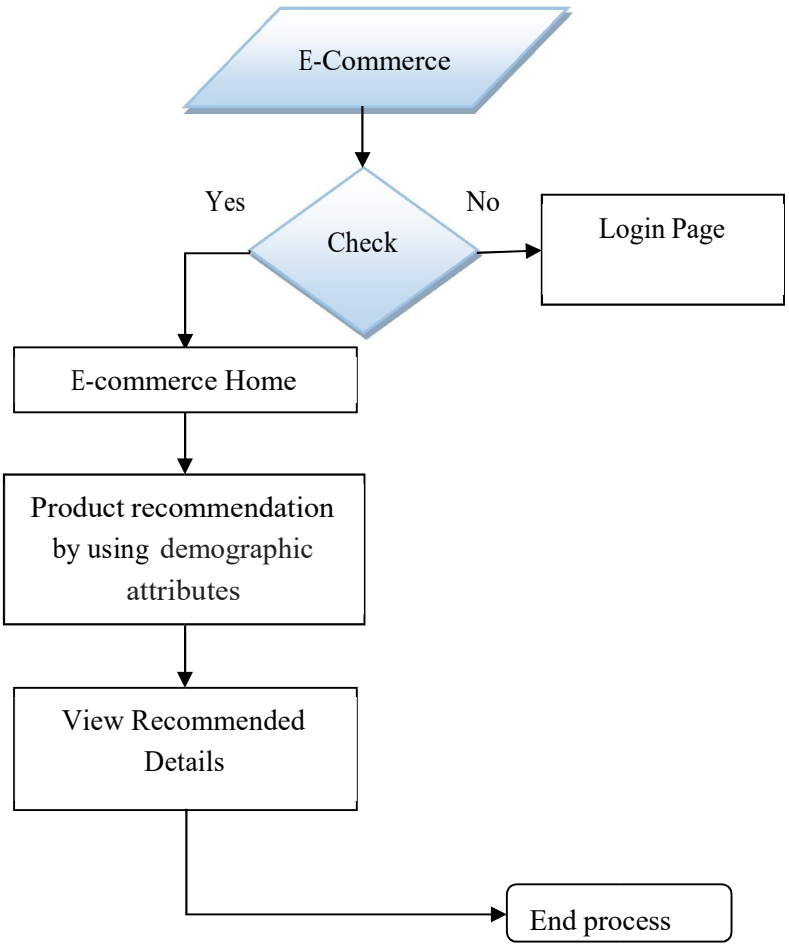
5.2 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



Data Flow Diagram of user Login



Data Flow Diagram of E-Commerce

5.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other nonsoftware systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

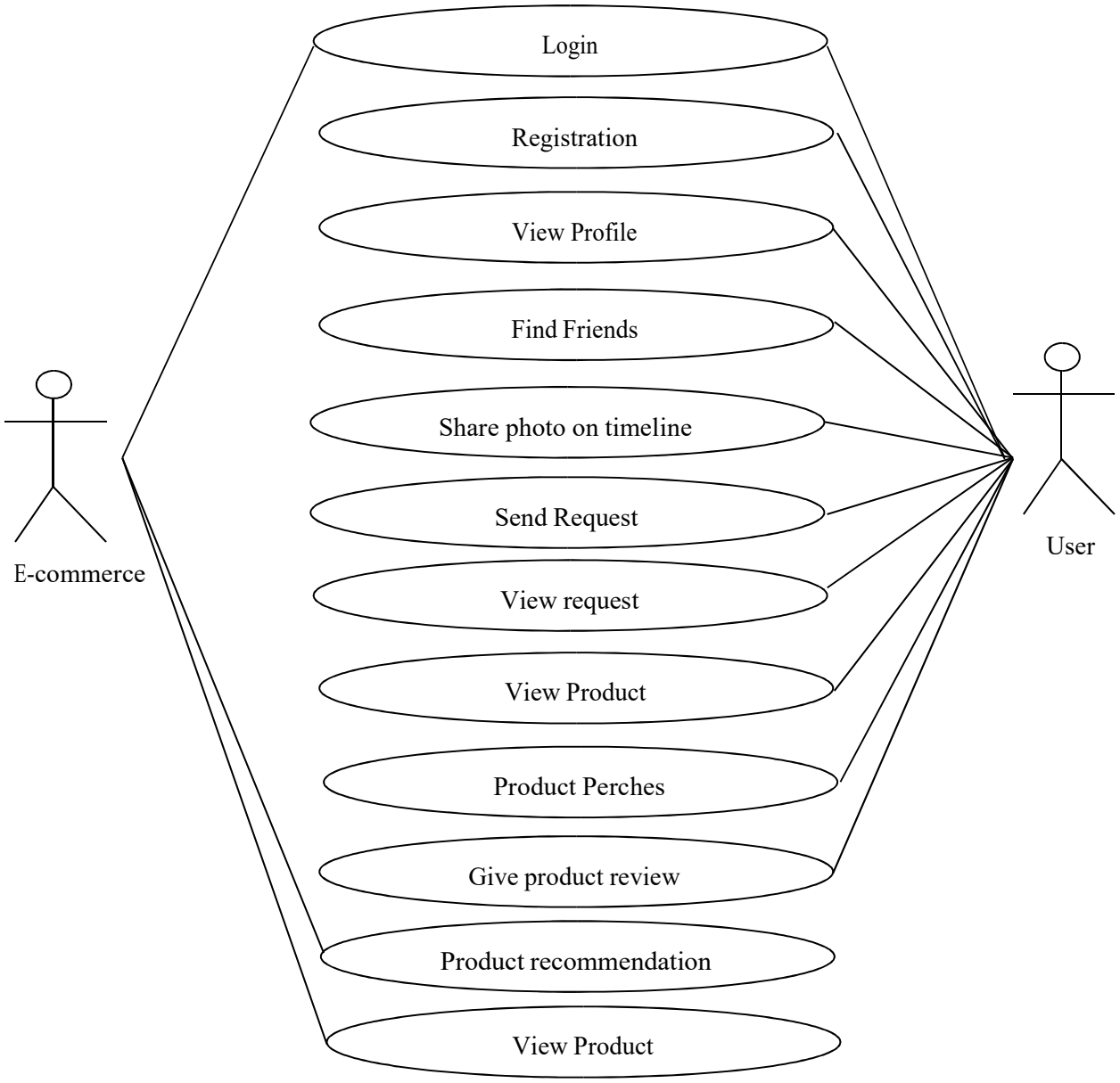
The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

5.4 GOALS:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.

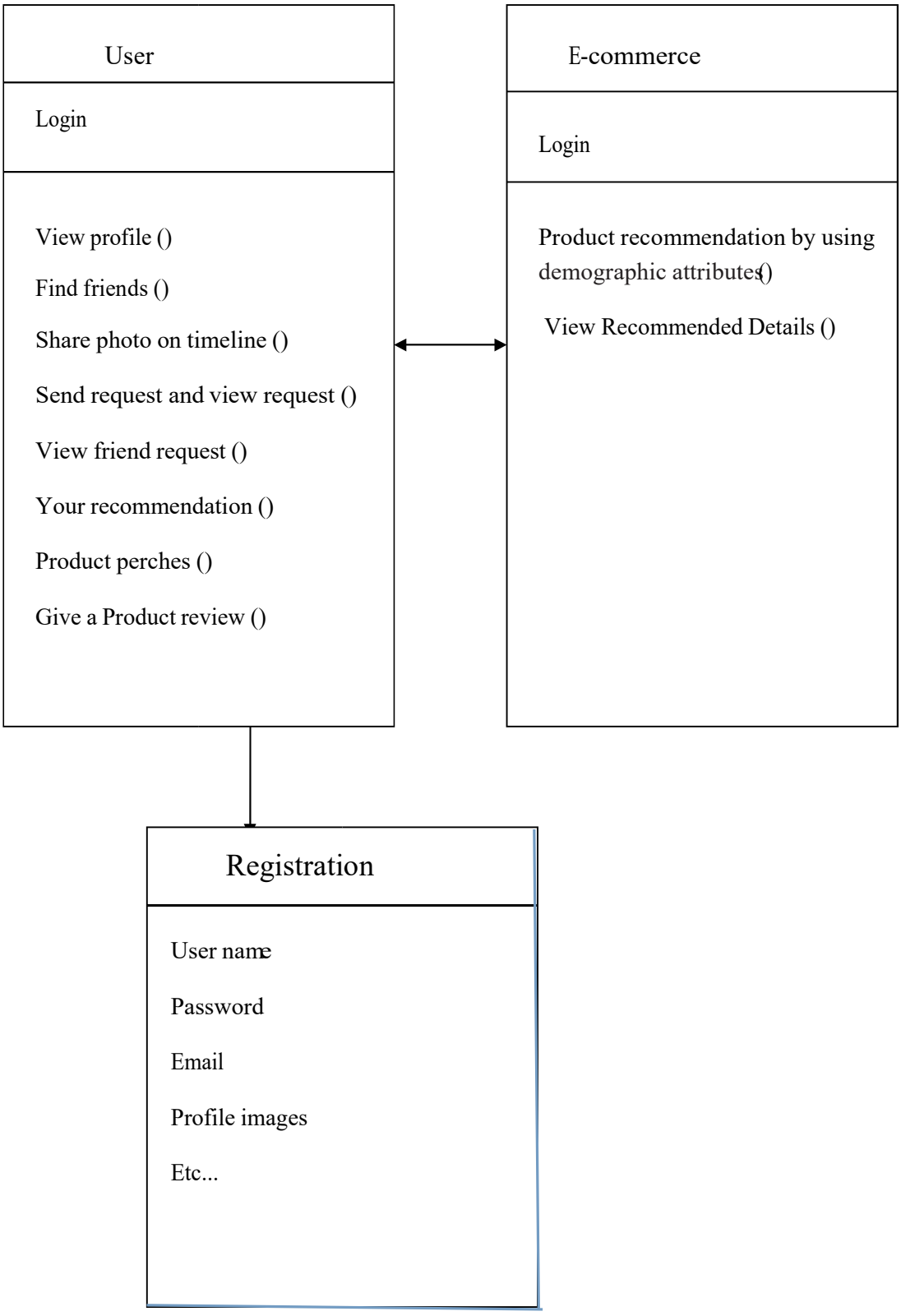
5.5 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



5.6 CLASS DIAGRAM:

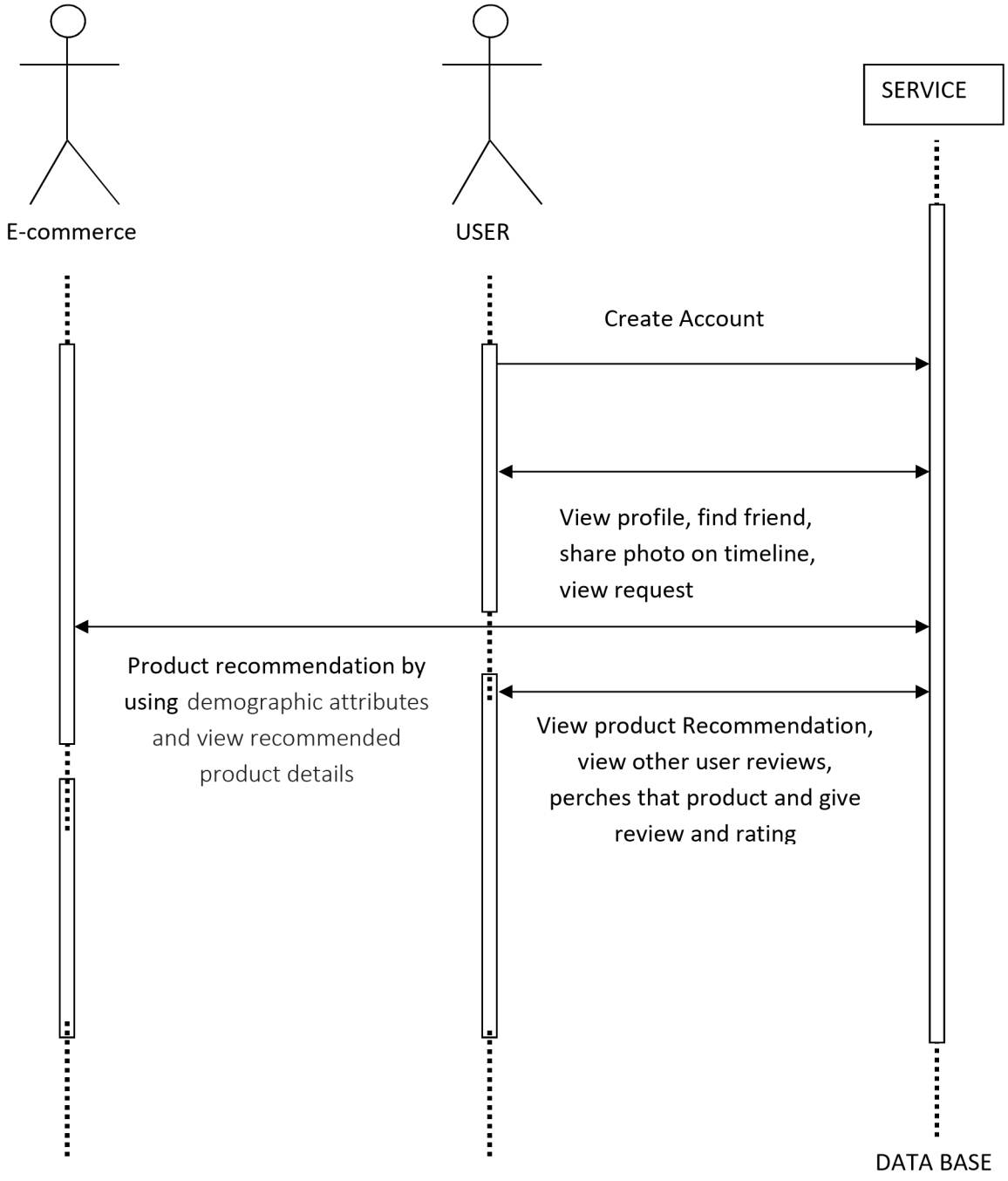
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



Class Diagram of User and E-Commerce

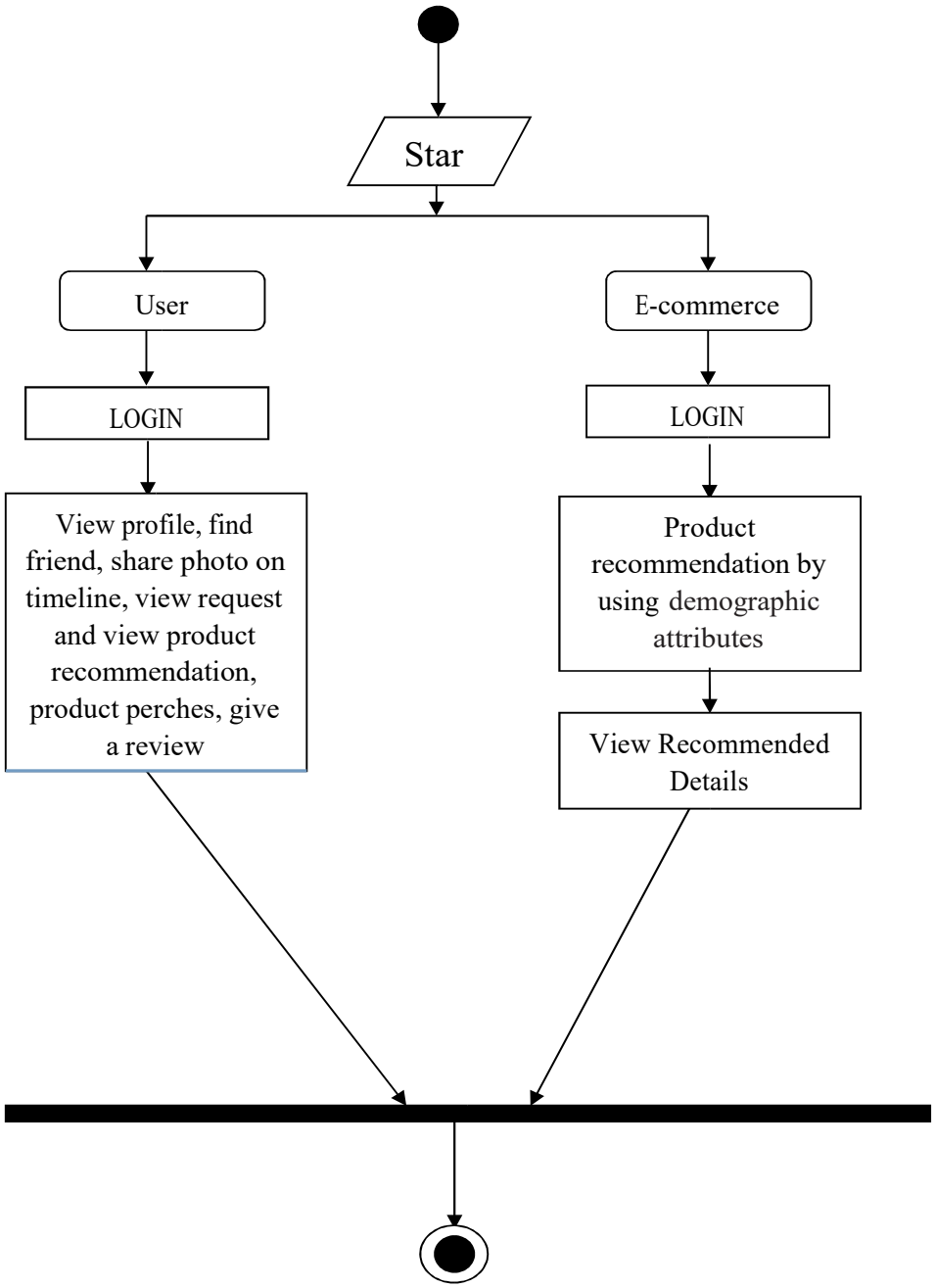
5.7 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



5.8 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Activity Diagram of User and E-Commerce

CHAPTER 6

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

6.1.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 INTEGRATION TESTING :

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 FUNCTIONAL TESTING:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2 SYSTEM TEST :

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.3 WHITE BOX TESTING :

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.4 BLACK BOX TESTING :

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.5 UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.6 INTEGRATION TESTING :

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.7 ACCEPTANCE TESTING :

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.8 TEST CASES :

Test case No	Description	Input	Expected Result	Actual Result	Test case Status
TC01	To verify that authorized user has	Enter valid user Email ID and	Log in successful	Log in successful	Pass

	entered valid user Email ID and password	password			
TC02	To verify that unauthorized user has entered user Email ID and password	Enter invalid user Email ID and password	Log in Unsuccessful	Log in Unsuccessful	Pass
TC03	To verify that the user has registered by entering valid details	Enter valid user details	Registered Successfully	Registered Successfully	Pass
TC04	To verify that the user has registered by entering invalid details	Enter invalid user details	Not Registered Successfully	Not Registered Successfully	Pass
TC05	To verify that the user send friend request Is working	Enter send request link	Waiting for Approve	Waiting for Approve	Pass
TC06	To verify that the user Accept friend request Is working	Click approve button	Approved message	Now they can Recommend book or movie	Pass

TC07	To verify that the user block friend request Is working	Click block button	Blocked message	Now they cannot Recommend book or movie	Pass
TC08	To verify that the user Edit profile button is working	Click edit profile button	Showing already existing details of user	Showing already existing details of user	Pass
TC09	To verify that authorized Admin has entered valid user Id and password	Enter valid User Id and password	Log in successful	Log in successful	Pass
TC10	To verify that unauthorized Admin has entered valid user Id and password	Enter invalid user Email Id and password	Log in Unsuccessful	Log in Unsuccessful	Pass
TC11	To verify that logout	Click on logout	Log out successful	Log out successful	Pass
TC12	To verify submit button	Click on submit	Upload image to Database	Upload image to Database	Pass
TC13	To verify select image button	Click on select image	Select an image from folder	Select an image from folder	Pass

TC14	To verify reset button	Click on reset	Clear all the field	Clear all the field	Pass
TC15	To verify exit button	Click exit button	Close the respective module	Close the respective module	Pass

6.9 PERFORMANCE ANALYSIS:

Methods to Compare:

We consider the following methods for performance comparison:

- **Popularity (Pop):** products are ranked by their historical sale volumes.
- **Popularity with Semantic Similarity (Pop++):** the ranking score is a combination of two scores:

  (1) the popularity score S1;

  (2) the cosine similarity S2 between product description and user text information, including profile, tweets and tags.
- **Embedding Similarities (ES):** Similarity scores between a user embedding and a list of product embedding's vp are used to rank products.
- **MF with user attributes (MFUA):** User attributes areincorporated into the basic matrix factorisation algorithmfor product rating prediction. fairness, we also use the pairwise loss function to train the model.
- **FM without User Interactions (FMUI):** Rendle applied the Factorization Machines for "follow" recommendation in KDDCup 2012. It has beenfound that similar performance was obtained with or without the interactions of user features. FM without user feature interactions is equivalent to SVDFeature. We reimplement this method in the SVDFeature framework with our extracted microblogging features.
- **ColdE:** Our proposed approach which uses the fitted user embedding features and product embedding features .

- **ColdDþE:** Our proposed approach which uses the microblogging features, the product embedding features and the fitted user embedding features. Especially, we only use demographic attributes here, since they have been shown important to product recommendation.
- **Coldþþ:** Since the user and product embeddings canbe learned for all the users and products respectively in the e-commerce website, we can train ColdE withal the users in U, not limited to the linked users UL. This variant is called Coldenhanced.

6.10 EVALUATION METRICS FOR PRODUCT RECOMMENDATION

Five widely used metrics are used for the evaluation of product recommendation results, including Precision@k, Recall@k, the Mean Average Precision (MAP), the Mean Reciprocal Rank (MRR) and the area under the ROC Curve (AUC).

Experimental Results on Ddense

We first evaluate the performance of product recommendation on Ddense, where d percent linked users are used as the training data, and the remaining percent linked users as the test data. To examine the performance with varying amount of training data, we set d to 80, 50, 20 and 10, which correspond to the #training/#test Split Ratios (SR) of 4:1,
1:1, 1:4 and 1:9 respectively.

The results of different methods for overall product recommendation are presented in below Table.

Performance Comparisons of Different Methods on Cold-Start Product Recommendation on Dsparse

Methods	MAP	MRR	R@10	AUC
Pop	0.175	0.125	0.120	0.684
Pop_{++}	0.175	0.175	0.120	0.684
MFUA	0.251	0.337	0.419	0.718
FMUI	0.252	0.337	0.421	0.720
$Cold_E$	**0.275***	**0.363***	**0.458***	**0.757***

$*$ *indicates that* $Cold_E$ *is significantly better than the best baseline at the level of 0.01.*

Running Time and Memory Costs for Our Approach on $\mathcal{D}_{dense}$ with the Split $\frac{\#train}{\#test}$ Ratio of 1:1

Phases	#users	Time (sec.)	Space (MB)
Training	7,927	563 (MART) 304 ($Cold_E$)	4.67 (MART) 15.72 ($Cold_E$)
Test	7,926	13.8 (MART) 5.1 ($Cold_E$)	4.67 (MART) 15.72 ($Cold_E$)

CHAPTER 7

IMPLEMENTATION

In this section, we present detailed description of the implementation, experimental procedures carried out and discussion on the results obtained.

7.1 MODULES:

- OSN System Construction Module
- Microblogging Feature Selection
- Learning Product Embeddings
- Cold-Start Product Recommendation

7.2 MODULES DESCSRIPTION:

7.2.1 OSN System Construction Module

- In the first module, we develop the Online Social Networking (OSN) system module. We build up the system with the feature of Online Social Networking. Where, this module is used for new user registrations and after registrations the users can login with their authentication.
- Where after the existing users can send messages to privately and publicly, options are built. Users can also share post with others. The user can able to search the other user profiles and public posts. In this module users can also accept and send friend requests.
- With all the basic feature of Online Social Networking System modules is build up in the initial module, to prove and evaluate our system features.
- Given an e-commerce website, with a set of its users, a set of products and purchase record matrix, each entry of which is a binary value indicating whether has purchased product. Each user is associated with a set of purchased products with the purchase timestamps.

Furthermore, a small subset of users can be linked to their microblogging accounts (or other social network accounts).

7.2.2 MICROBLOGGING FEATURE SELECTION

- In this module, we develop the Microblogging Feature Selection. Prepare a list of potentially useful microblogging attributes and construct the microblogging feature vector for each linked user. Generate distributed feature representations using the information from all the users on the ecommerce website through deep learning. Learn the mapping function, which transforms the microblogging attribute information au to the distributed feature representations in the second step. It utilises the feature representation pairs of all the linked users as training data.
- Ademographic profile (often shortened as “a demographic”) of a user such as sex, age and education can be used by ecommerce companies to provide better personalised services. We extract users’ demographic attributes from their public profiles. Demographic attributes have been shown to be very important in marketing, especially in product adoption for consumers

7.2.3 LEARNING PRODUCT EMBEDDINGS

- In the previous module, we develop the feature selection, but it is not straightforward to establish connections between users and products. Intuitively, users and products should be represented in the same feature space so that a user is closer to the products that he/she has purchased compared to those he/she has not. Inspired by the recently proposed methods in learning word embeddings, we propose to learn user embeddings or distributed representation of user in a similar way.
- Given a set of symbol sequences, a fixed-length vector representation for each symbol can be learned in a latent space by exploiting the context information among symbols, in which “similar” symbols will be mapped to nearby positions. If we treat each product ID as a word token, and convert the historical purchase records of a user into a timestamped sequence, we can then use the same methods to learn product embeddings. Unlike matrix factorization, the order of historical purchases from a user can be naturally captured.

7.2.4 COLD-START PRODUCT RECOMMENDATION

- We used a local host based e-commerce dataset, which contains some user transaction records. Each transaction record consists of a user ID, a product ID and the purchase timestamp. We first group transaction records by user IDs and then obtain a list of purchased products for each user.
- For our methods, an important component is the embedding models, which can be set to two simple architectures, namely CBOW and Skip-gram.

7.3 MATHEMATICAL LOGIC

Input:-

Let S is the Whole System Consist of S= {I, P, O}

I = Input. I = {U, Q, D} U = User U = {u1,u2….un}

Q = Query Entered by user Q = {q1, q2, q3…qn}

D = Dataset

P = Process

Step1: Admin will upload the product in E-commerce site.

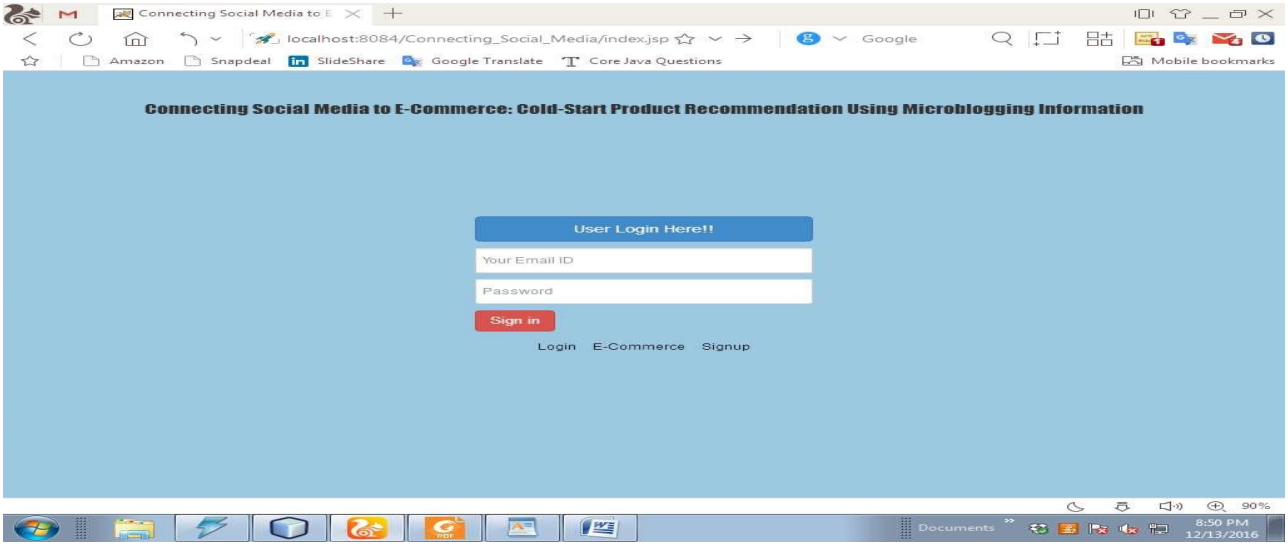
Step2: That uploaded product will be seen on Social sites where user can view, share and give comments on that product. User can send and receive friend request.

Step3: All the reviews should be seen in E-commerce site when user login to E- commerce site.
Output: User will get recommendation regarding of that product on ecommerce website.

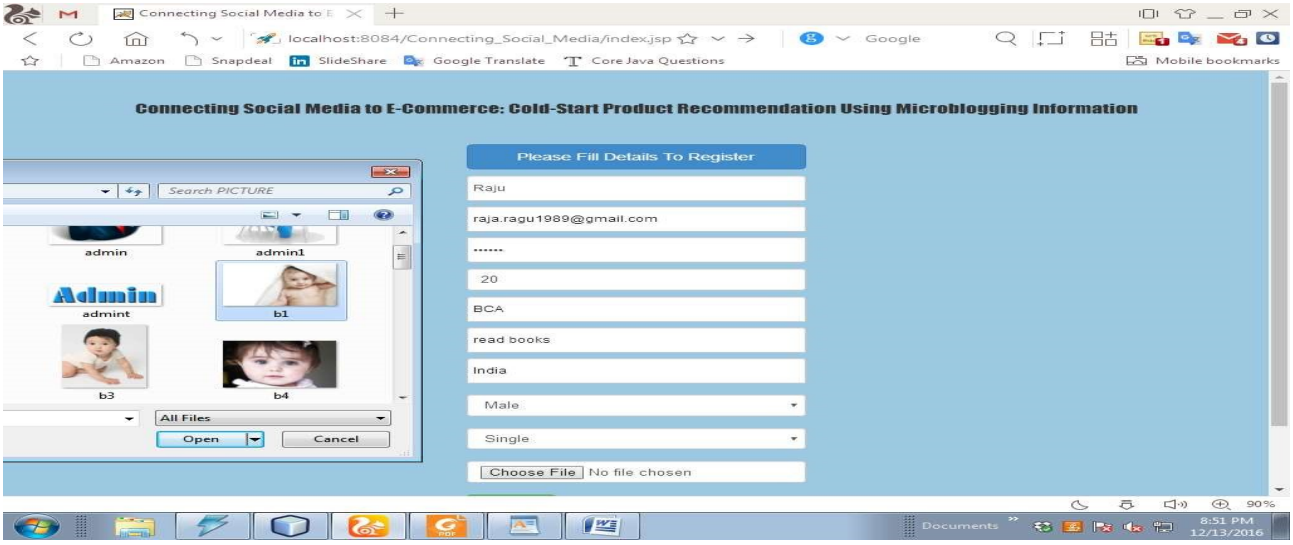
CHAPTER 8

RESULTS

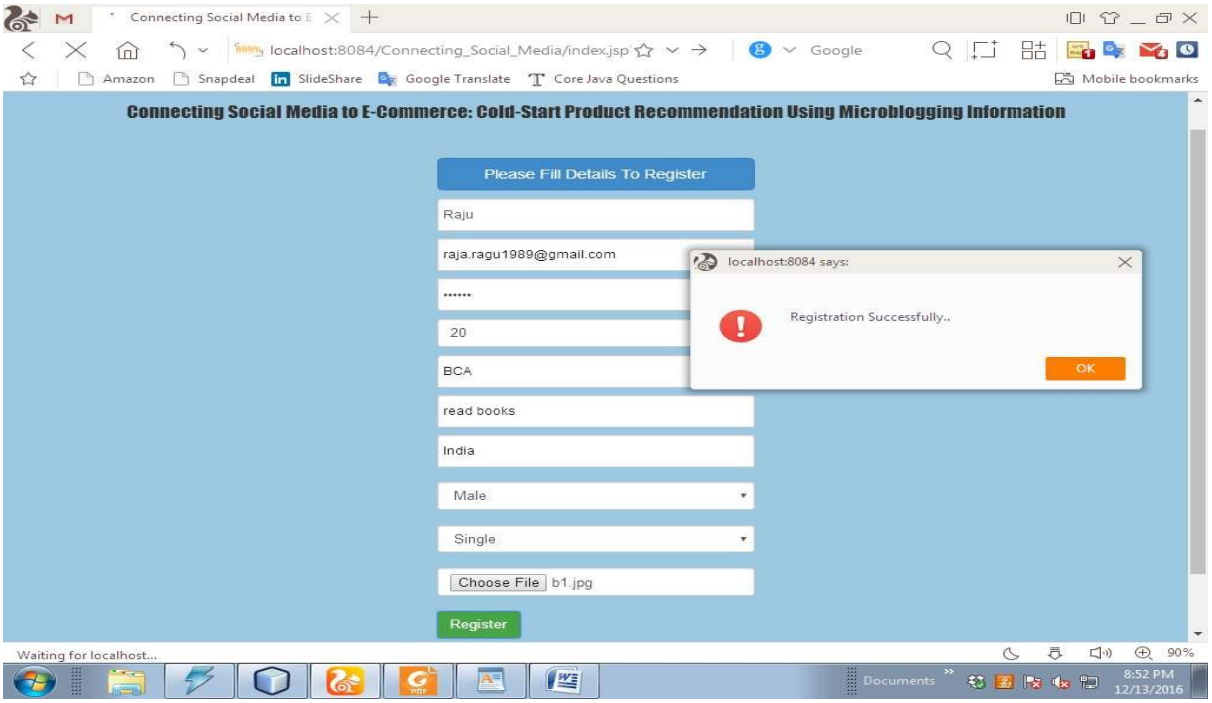
SCREENSHOTS



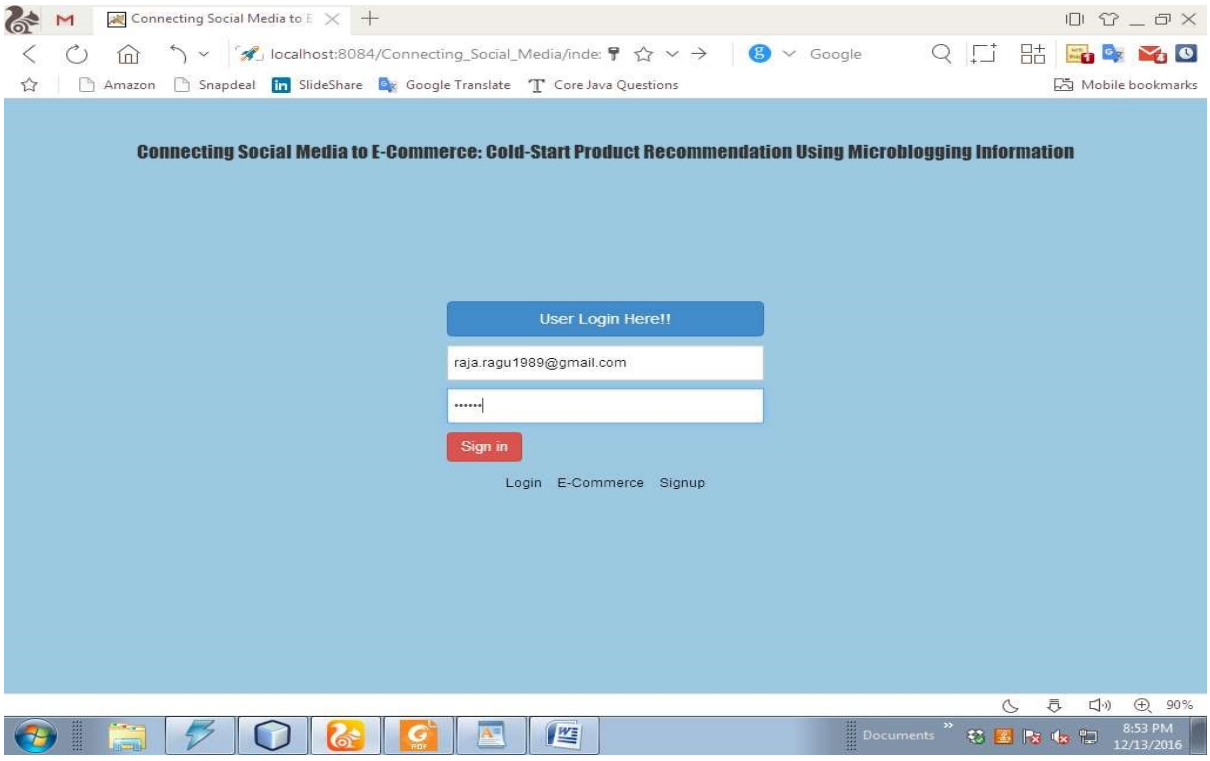
Login Page

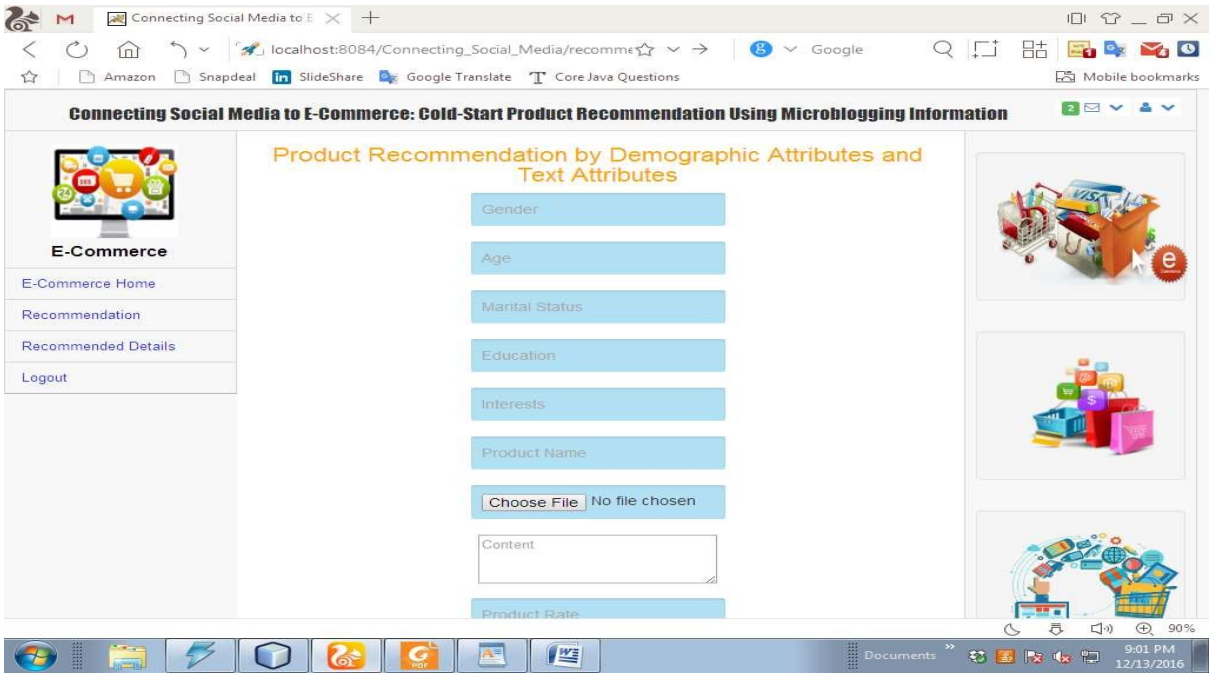


Registration Page

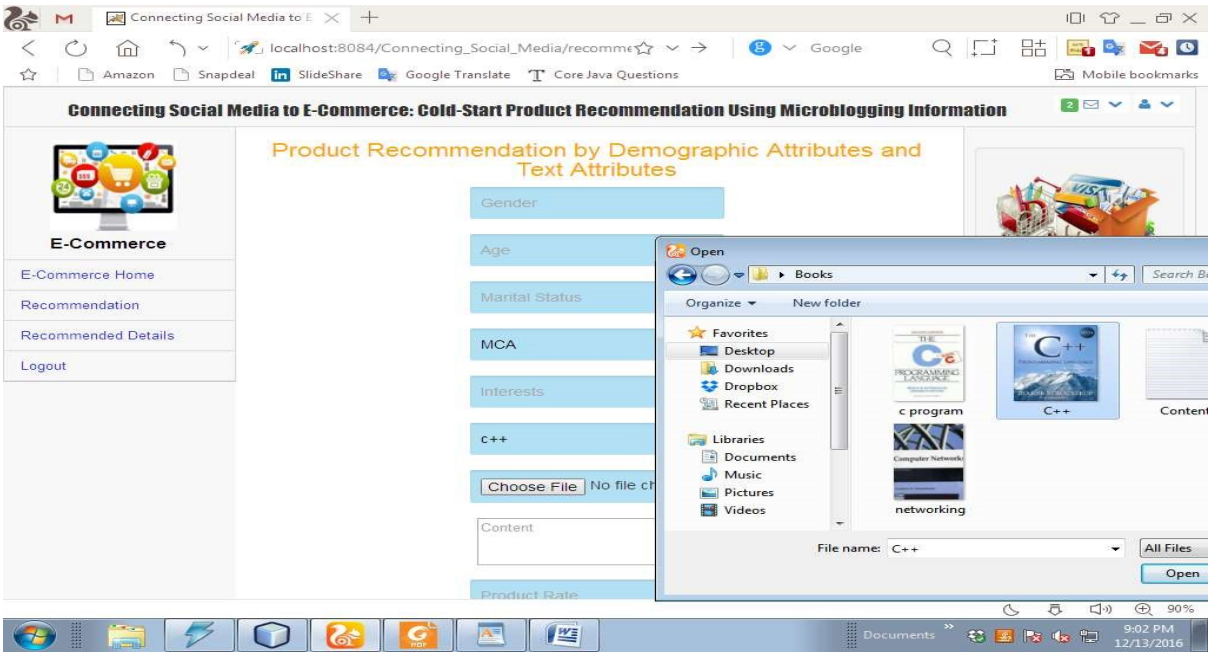


After Successful Registration





E-Commerce Home Page



Recommendation Details Page

CHAPTER 9

9.1 CONCLUSION AND FUTURE SCOPE

In this paper, we have studied a novel problem, cross-site cold-start product recommendation, i.e., recommending products from e-commerce websites to microblogging users without historical purchase records. Our main idea is that on the e-commerce websites, users and products can be represented in the same latent feature space through feature learning with the recurrent neural networks. Using a set of linked users across both e-commerce websites and social networking sites as a bridge, we can learn feature mapping functions using a modified gradient boosting trees method, which maps users' attributes extracted from social networking sites onto feature representations learned from e-commerce websites. The mapped user features can be effectively incorporated into a feature-based matrix factorization approach for cold-start product recommendation. We have constructed a large dataset from WEIBO and JINGDONG. The results show that our proposed framework is indeed effective in addressing the cross-site cold-start product recommendation problem. We believe that our study will have profound impact on both research and industry communities. Currently, only a simple neutral network architecture has been employed for user and product embeddings learning. In the future, more advanced deep learning models such as Convolutional Neural Networks13 can be explored for feature learning. We will also consider improving the current feature mapping method through ideas in transferring learning.

9.2 FUTURE SCOPE

Currently, only a simple neutral network architecture has been employed for user and product embeddings learning. In the future, more advanced deep learning models such as Convolutional Neural Networks13 can be explored for feature learning. We will also consider improving the current feature mapping method through ideas in transferring learning.

REFERENCES

1] F. Cheng, C. Liu, J. Jiang, W. Lu, W. Li, G. Liu, W. Zhou, J. Huang, and Y. Tang. Prediction of drug-target interactions and drug repositioning via network-based inference.PLoS Computational Biology, 8:e1002503, 2012.

2] E. Constantinides. Influencing the online consumer's behavior: the web experience.Internet research, 14:111–126, 2004.

3] J. L. Herlocker, J. A. Konstan, and J. Riedl.Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM conference on Computer supported cooperative work, pages 241–250. ACM,2000.

4] C. Jayawardhena, L. T. Wright, and C. Dennis. Consumers online: intentions, orientations and segmentation. International Journal of Retail &Distribution Management, 35:515–526, 2007.

5] A. Karatzoglou. Collaborative temporal order modeling. In Proceedings of the _fth ACM conferenceon Recommender systems, pages 313–316, 2011.

6] I. Konstas, V. Stathopoulos, and J. Jose. On social networks and collaborative recommendation.InProceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 195–202. ACM, 2009.

7] A. Liaw and M. Wiener.Classification and regression by randomforest.R news, 2:18–22, 2002.

8] C.-H. Park and Y.-G. Kim. Identifying key factors affecting consumer purchase behavior in an online shopping context.International Journal of Retail & Distribution Management, 31:16–29, 2003.

9] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM

Conference on Computer Supported Cooperative Work, pages 175–186. ACM, 1994. 10] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl.Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th InternationalConference on World Wide Web, pages 285–295. ACM, 2001.

11] J. B. Schafer, J. A. Konstan, and J. Riedl.E-commerce recommendation applications. In Applications of Data Mining to Electronic Commerce, pages 115–153. Springer, 2001.

12] E. Shen, H. Lieberman, and F. Lam. What am i gonna wear?: scenario-oriented recommendation. In Proceedings of the 12th international conference onIntelligent user interfaces, pages 365–368. ACM, 2007. 13] K. H. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme.Tag-aware recommender systemsby fusion of collaborative filtering algorithms. In Proceedings of the 2008 ACM symposium on Appliedcomputing, pages 1995–1999. ACM, 2008.

14] R. Verheijden. Predicting purchasing behavior throughout the clickstream.Master's thesis, Eindhoven University of Technology, May 2011.

15] F. Wu and B. A. Huberman.Novelty and collective attention.Proceedings of the National Academy ofSciences, USA, 104:17599–17601, 2007.