

A

Mini Project

On

Predicting Rating Using Review Analysis

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

Yogitha Sree Annangi(197R1A0559)

D Veena Laxmi Gayatri (197R1A0511)

Yarlagadda Chandana Sri(197R1A0557)

Under guidance of

B. P. Deepak Kumar

(Assistant Professor)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGC Act. 1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE



This is to certify that the project entitled “**Data Rating Review Analysis and Predicting Ratings Using Reviews**” being submitted by **Yogitha Sree Annangi (197R1A0559), Deverakonda Veena Laxmi Gayatri (197R1A0511), Yarlagadda Chandana Sri (197R1A0557)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

B.P.Deepak Kumar
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on_____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **B.P Deepak Kumar**, Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel & J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

YOGITHA SREE ANNANGI (197R1A0559)
D VEENA LAXMI GAYATRI (197R1A0511)
YARLAGADDA CHANDANA SRI (197R1A0557)

ABSTRACT

Sentiment analysis is one of the fastest spreading research areas in computer science, making it challenging to keep track of all the activities in the area. We present a customer feedback reviews on product, where we utilize opinion mining, text mining and sentiments, which has affected the surrounded world by changing their opinion on a specific product. Data used in this study are online product reviews collected from <http://Amazon.com>. We performed a comparative sentiment analysis of retrieved reviews.

Amazon data with collection of review text, product details, rating and review are taken as input. In the first step data analysis is performed from data based and graphical representation of each product, category, count, comparison is shown (rating, label vs rating, label vs product category, rating vs product category, verified purchases after analysis part is done. Testing training and prediction of rating from given review is performed in this process text processing, feature vectorization, feature extraction from text, cross validation, splitting data and training classifier steps are performed using machine learning and then algorithm (**linear SVC algorithm**) is applied to the review text and classifier is generated.

In prediction step when review as given and input prediction of rating is performed using Trained classifier and accuracy, Precision, Recall, f1-score is performed.

CONTENTS

ABSTRACT	I
LIST OF FIGURES	IV
LIST OF TABLES	V
1. INTRODUCTION	
1.1. DOMAIN INTRODUCTION	2
1.2. PROBLEM DEFINITION	5
1.3. OBJECTIVES	5
1.4. SCOPE OF PROJECT	5
2. LITERATURE SURVEY	
2.1. EXISTING SYSTEM AND RELATED WORKS	7
3. REQUIREMENT ANALYSIS	
3.1. FUNCTIONAL REQUIREMENTS	17
3.2. NON-FUNCTIONAL REQUIREMENTS	17
3.2.1. PRODUCT	18
3.2.2. ORGANISATIONAL	18
3.2.3. USER	18
3.2.4. BASIC OPERATIONAL	18
3.3. HARDWARE REQUIREMENTS	19
3.4. SOFTWARE REQUIREMENTS	20
4. DESIGN	
4.1. OVERALL SYSTEM ARCHITECTURE	21
4.2. PROPOSED SYSTEM	21
4.3. DATA FLOW DIAGRAM	22
4.4. SEQUENCE DIAGRAM	24
4.5. USE CASE DIAGRAM	26
4.6. CLASS DIAGRAM	27

4.7.	PRODUCT ADOPTION RATE	28
5.	IMPLEMENTATION	
5.1.	DATASET	30
5.2.	CSV CONVERSION	31
5.3.	PRODUCT RATE ADOPTION	32
5.4.	CONCEPTS	33
5.4.1.	DATA MINING	33
5.4.2.	MACHINE LEARNING	37
5.4.3.	PYTHON	41
5.5.	ALGORITHM	42
6.	IMPLEMENTATION	
6.1.	UNIT YESTING	45
6.2.	INTEGRATION	46
6.3.	INTEGRATION TESTING	46
6.4.	VALIDATION TESTING	49
6.5.	OUTPUT TESTING	50
7.	RESULT	52
8.	CONCLUTION	56
9.	REFERENCES	57

LIST OF FIGURES

FIG NO.	FIG.NAME	PAGE NO.
1.1.	ANALYSIS	3
4.1	SYSTEM ARCHITECTURE	16
4.2	DATA FLOW DIAGRAM 1	18
4.3	DATA FLOW DIAGRAM 2	19
4.4	SEQUENCE DIAGRAM 1	20
4.5	SEQUENCE DIAGRAM 2	21
4.6	USE CASE DIAGRAM	22
4.7	CLASS DIAGRAM	23
5.1	RATING OF PRODUCTS	28
5.2	GRAPH OF RATING	28
5.3	MACHINE LEARNING	34
5.4	PYTHON LOGO	37
7.1	SNAPSHOT 1	20
7.2	SNAPSHOT 2	21
7.3	SNAPSHOT 3	20
7.4	SNAPSHOT 4	21
7.5	SNAPSHOT 5	20
7.6	SNAPSHOT 6	21

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
6.1	INTEGRATION TESTING	44
6.2	VALIDATING TESTING	44

1.INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1 DOMAIN INTRODUCTION

Market intelligence is an important area of study in business environment today to gather valuable insights into customers purchasing behavior. By analysis of huge volume of transactions, important knowledge about products preferred by customer, rate of usage of products by customer, customer spend analysis etc. can be gathered. Predicting customer's products adoption rate is very important for targeted marketing and marketing strategy development. This also helps in personalization of services to the customers.

The current works for product adoption prediction is in two categories

1. Adopter already consumed the product
2. New adopters not yet consumed the product

Though gathering information about product rate adoption based on current transactions is easy, predicting the future trend accurately is a challenging problem.

Nowadays, recommendation systems have improved to promote customer activity on electronic commerce websites. On Amazon.co.jp, some related products are displayed recommended when the user inspects a product of his/her interest. Many methods are proposed to recommend the products to users. Traditionally, the main method to recommend products is collaborative filtering. Study [Buettner R] concluded that aspect-based opinion mining and collaborative filtering can recommend the products effectively. We cannot obtain the goodness of fit of each product for users if we utilize only opinion mining technique. We also cannot obtain the details of customers' opinion of each product if we utilize only collaborative filtering. Therefore, it is necessary to combine the results of opinion mining and collaborative filtering to obtain better results of recommendation.

Predicting Rating Using Review Analysis

Sentiment analysis is a form of natural language processing for tracking the temperament of the public about a specific product or topic, or its text classification that classifies texts depend on the sentimental orientation (SO) of opinions they include. Sentiment analysis, which is also called opinion mining, take part in building a system to accumulate and test opinions about the product made in blog posts, comments, reviews or tweets. Opinion mining can be useful in several ways. i.e., in marketing it helps in judging the success of a new product launch, determine which versions of a product are popular and even differentiate which demographics like or dislike particular features. There are many challenges in opinion mining. The first is that people don't always phrase opinions in a same way. A second challenge is an opinion word that is considered to be positive in one state may be considered negative in another state. Most traditional text processing depends on the fact that small differences between two pieces of text don't change the meaning very much. In opinion mining, however, "the camera was great" is very different from "the camera was not great". People can be dissimilar in their statements. Farthest reviews will have both positive and negative comments, which is slightly manageable by analyzing sentences one at a time. However, in the more annular medium like twitter or blogs, the more likely people are to join different opinions in the same sentence which is easy for a human to understand, but more difficult for a computer to parse. Sometimes even other people have difficulty recognizing what someone thought based on a short piece of text because its shortage context.

Opinion mining and summarization process contain three main steps, first is Opinion Retrieval, Opinion Classification and Opinion Summarization.

Opinion Retrieval is the process of gathering review text from review websites. Different review websites involve reviews for products, movies, hotels and news. Information retrieval techniques like web crawler can be applied to accumulate the review text data from many sources and store them in database. This step includes retrieval of reviews, micro blogs, and comments of user.

Predicting Rating Using Review Analysis

Basic step in opinion mining is classification of review text. Given a review document $D = \{d_1, \dots, d_n\}$ and categories set $C = \{\text{positive}, \text{negative}\}$, sentiment classification is to classify each d_i in D , with a tag expressed in C . The method involves classifying review text into two forms namely positive and negative.

Summarization of opinion is a main part in opinion mining process. Summary of reviews provided should be established on lineaments or subtopics that are mentioned in reviews. Many works have been executed on condensation of product reviews. The opinion condensation process mainly includes the following two methods. Feature based condensation a type condensation involves returns of frequent terms (features) that are appearing in many reviews. Features present in review text can be identified using Latent Semantic Analysis (LSA) method. Figure below has the architecture of Opinion Mining which says how the input is being classified on various steps to summarize the reviews.

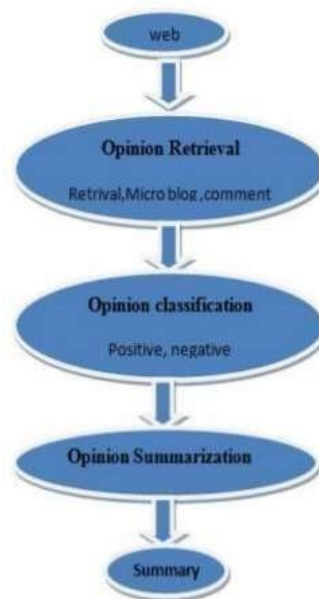


Fig.1.1 Analysis Method

1.2 PROBLEM DEFINITION

The customer interest on product changes over time especially with a highly competitive environment. Predicting the future adoption rate of product for a customer is very in accurate with the current statistics-based methods. This product deals with this problem of predicting the future adoption rate of product for customers.

1.3 OBJECTIVES

The following are the objectives of this project.

- Process large volumes of customer transactions.
- Predict the future adoption rate of product.
- Measure the accuracy of the proposed solution.

1.4 SCOPE OF PROJECT

The scope of the product is in building machine learning models to capture the customer behavior and use these models to predict the product adoption rate based on customer price and product competition information.

2.LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

1.1 RELATED WORKS

The current works are detailed below

Recommender Systems: Recommender systems infer each user's preferences to products that she has not rated before, and then recommend those products that have the largest predicted ratings. All these recommendation systems saliently assumed that users would adopt the products once (e.g., movies and travel attractions), thus they focused on predicting the preferences of users to products that have not been adopted yet.

Product Adoption Prediction in Social Networks: With the proliferation of online social networks, a hot research topic is how to leverage the social network for better product adoption prediction performance. A distinct characteristic of the social network is the existence of the social influence, which usually presents in two forms: the global crowd influence shows the herding effect among the population level while the local social neighbors influence argues that users are more likely to be influenced by the social neighbors' decisions than others.

In [1] author proposed a personality-based product recommendation system which used social media data to predict user's personality. Machine learning approaches are applied to predict user personality traits based on social networking features. Product recommendation system uses the relationships between the personality-based consumer preferences and the product characteristics to rank the products. The scale used for personality trait in this work did not have higher correlation with product characteristics as it could not model user interest and temporal changes in interests.

Authors in [2] proposed a deep learning model for image recommendation. Latent features are learnt from user image using Deep Learning network and similarity of these features with new images are measured using Euclidean distance. The new images are then ranked based on distance and top N ranked images are recommended to the users. Though this concept applies only for images, user of deep learning for learning latent features is a salient feature which can be used for product recommendations.

A memory-based technique for group recommendation system is proposed in [3]. Support vector-based regression model is used to compute the similarity between the items. This work used Pearson Universal Kernel (PUK) function to model the similarities between the items. Use of Support vector regression is able to solve the data sparsity problem. The method works only for single dimension ratings and kernel function needs to be adapted for multi-dimensional ratings.

A two-stage cascaded recommendation system using decision tree and collaborative filtering is proposed in [4] for people recommendation in online dating services. At first stage collaborative filtering is applied and the recommendations from it are re-ranked using a decision tree critic. Due to this two-stage cascaded recommendation, the success rate of match making improved. An important take away from this work, applying post re ranking procedures to collaborative filtering help to achieve better personalization.

In [5] propose a highly scalable k-means clustering based recommendation algorithm. A new centroid selection algorithm exploiting underlying data correlation structures provides better accuracy than random centroid selection.

Author in [6] proposed a new neural complementary recommender system called ENCORE which use the complementary itemrelationships and user preferences. A neural network model is built to learn the complex (non-linear) relationships between items for flexible and scalable complementary product recommendations.

A mixture model approach for post purchase complementary product recommendation is proposed in [7]. The mixture model is trained to learn latent prediction contexts, which are determined by user and item profiles, and then make open rate predictions accordingly. Expectation Maximization (EM) algorithm is used to optimize the parameters of mixture model. A major problem in this method is that it could model the temporal features of user behavior.

Authors in [8] proposed a complementary recommendation system that learns visual cues in an unsupervised manner to calculate the co-occurrence distribution of items. A salient feature in this solution is that a conditional generative model is trained to produce multiple novel samples of complementary items (in the feature space) for a given query item.

Authors in [9] proposed a deep learning solution using Siamese Convolutional Neural Network architecture to learn style compatibility from the products. The deep learning model is able to find the related products based on style compatibility and recommend those related products to the users. The solution is trained on word model and could be extended for more sophisticated sentence models to be useful in real world environment.

Authors in [10] extended the item based collaborative filtering to work in the framework of neural word embedding. Item embedding is generated in latent space and using it the item-to-item relationship is inferred. Skip gram with negative sampling is the word embedding method used in this work. A salient feature in this solution is that item relationships can be learnt from unstructured product descriptions.

Authors in [11] proposed a new user similarity model for collaborative recommendation which solves the cold start problems. The solution is able to increase the recommendation performance when only fewer ratings are available using the local context information of ratings and global preference of user behavior.

A hybrid recommendation system combining content based, collaborative filtering and data mining techniques is proposed in [12] to solve the efficiency problems in recommendation for huge size of transactions. The customers are clustered and association rule mining is done for customers in same cluster to provide a more assertive and personalized recommendations.

Authors in [13] extended the collaborative filtering recommendation for the case of implicit feedbacks. The implicit user observations are transformed into two paired magnitudes: preferences and confidence levels. For each user-item pair, this work derives from the input data an estimate to whether the user would like or dislike the item (“preference”) and couple this estimate with a confidence level. This preference-confidence partition has no parallel in the widely studied explicit-feedback datasets, yet serves a key role in analyzing implicit feedback. Latent factor algorithm is designed that directly addresses the preference-confidence paradigm. Unlike explicit datasets, here the model should take all user-item preferences as an input, including those which are not related to any input observation (thus hinting to a zero preference). This is crucial, as the given observations are inherently biased towards a positive preference, and thus do not reflect well the user profile. However, taking all user-item values as an input to the model raises serious scalability issues –the number of all those pairs tends to significantly exceed the input size since a typical user would provide feedback only on a small fraction of the available items.

Predicting Rating Using Review Analysis

Authors in [14] proposed a personalized ranking algorithm based on both implicit and explicit user feedback. The proposed MERRSVD++ algorithm optimizes the well-known evaluation metric Expected Reciprocal Rank (ERR) and is based on the newest xCLiMF model and SVD++ algorithm.

Author in [15] propose a new matrix factorization model named PSVD, which allows us to capture user's different preferences over different items flexibly in rating prediction. Specially, authors use a pair of preferences to represent the whole preference of user over items. Then the dual preferences are considered simultaneously in building the latent feature vector of user. Moreover, PSVD model allows users to adjust their own feature vector when selecting different products.

A unified one class collaborative filtering approach is proposed in to simultaneously optimize both rating and rank of recommended items. The proposed solution integrated Collaborative less is more filtering (CLMF) and probabilistic matrix factorization (PMF) approaches by sharing common latent features of users and items in CLMF and PMF.

A collaborative recommendation algorithm with importance to tags is proposed in [16]. Type and frequency of use of the label reflect user preferences and preferences, in order to establish a new user preferences model for better mining and use implicit user feedback data will affect the degree of the label on the user to quantify, to establish a new method for similarity computation.

A learning-to-rank recommender system was proposed in [17] which use implicit feedback signals from multiple channels. The solution was on focused on Factorization Machines (FMs) with Bayesian Personalized Ranking (BPR), a pairwise learning-to-rank method that allows to experiment with different forms of exploitation.

In the literature, much of the active research has been devoted to the product adoption prediction problem. Specifically, these works usually classified users into two categories: the adopters that already consumed this product and the non-adopters that have not consumed it till now. In other words, these methods described users' product adoption states with a binary buy-or-not representation. Then some learning algorithms are proposed to model the future adoption possibilities of those non-adopters. E.g., the popular recommender systems deal with the task of predicting users' preferences to the products that they have not consumed before. In contrast to these products that are usually consumed only once (e.g., books and movies), there are plenty of products users may use frequently after buying them, such as smart devices. Fig. 1 shows an illustrating example of users' preferences to two different smart devices over time.

The traditional buy-or-not binary-valued adoption representation only captures the fact that both users have consumed the two smart devices in the past. Actually, in a specific competitive market (e.g., mobile devices), it is nature for a user to switch among different products over time after she consumes these products (e.g., iPhone, Samsung, and Windows). Compared to the traditional static buy-or-not adoption representation, the merchants care more about users' loyalty and commitment to the products over time after users consume the products. To better capture users' loyalty to the frequently used products after purchase over time, we argue, the measure of adoption rate, i.e., the usage rate and regularity that consumers use a product at a particular time, is more appropriate to describe users' preference changes to different products. As each user's adoption rate over time could be summarized into an adoption series, by capturing each user's adoption rate series, the two observations in Fig.1 can be easily obtained.

After introducing the product adoption rate measure, the problem we study in this paper is how to predict the future product adoption rate of each user in a competitive market. Unfortunately, none of the existing models (e.g., models for recommender system, the time-series forecasting models) could be directly applied to this problem due to the following challenges. First, a user's decision-making process is very complex as many heterogeneous sources around her may contribute to the final decision, e.g., the users' own profiles, and the social network structure. How to design a flexible prediction model that can leverage many heterogeneous data sources in a unified framework remains pretty much open. Second, based on the heterogeneous data sources around users, the adoption decision process varies from person to person. For example, some users may weight more on social neighbors' opinions while others are unlikely to change their decisions. Thus, from a limited adoption rate series of each user, how to explore users' unique preferences becomes another challenge. Last but not least, in a competitive market, the fierce competition among different products is a significant factor to track the transitions of users' adoption rates over time. In fact, in the marketing domain, product competition is well recognized as a focal part that influences a company's market performance. How to mine the competitive relationships among products in a competitive market to improve the product adoption rate prediction results? In summary, the data heterogeneity, the unique user preference and product competition compose three main challenges of the problem we study.

To address the challenges of data heterogeneity and user uniqueness, we provided a preliminary study on the product adoption rate prediction problem from a multi-factor view. Specifically, we first introduced a flexible factor-based decision function to capture users' product adoption rate changes over time, where various factors from heterogeneous data sources that may influence users' decisions can be leveraged. Using this factor-based decision function, we then provided a Generalized Adoption Model (GAM) and a Bayesian Personalized Adoption Model (BPAM) to learn the parameters of the decision function with both generalized and personalized assumptions of users' preferences. In this paper, we further extend our previous work and study the product adoption rate prediction problem with multiple products in a competitive market. A naive method is to divide the multi-product adoption rate prediction problem into a set of independent single product prediction problems, then our previous proposed GAM and BPAM models could be applied directly.

This independent assumption among products enjoys the advantage of simplicity, however, it fails to consider the competition among products in real-world adoption decisions. In a competitive market, products turn to compete with each other to attract the attention of users. Take the competition among smart devices as an example, as shown in Fig. 1, since John turns to adopt product A more frequently at t_{p1} , the adoption rate of B decreases at that time. Therefore, we argue, in order to predict users' adoption rate more accurately, it is essential to take the competition effect among products into consideration. Specifically, given a competitive market with multiple products compete with each other, we study how to incorporate product competition into the proposed GAM and BPAM models, and jointly learn user preference and product competition in a unified framework. The extended models are termed as GAM with Competition (GAM-C) and BPAM with Competition (BPAM-C) respectively. We argue that the joint modeling of users' preferences and product competition is significant as users interact with multiple products at the same time, and product competition is considered as an indispensable part for users to transit commitment to different products.

In summary, by extending the problem definition from predicting the adoption rate of a particular product to multiple products in a competitive market, we further address the technical challenge of how to model product competition as a factor in the decision function, and how to jointly learn the parameters of users' preferences and product competition in our proposed models of GAM-C and BPAM-C. Finally, we conduct extensive experiments on two markets: a smartphone device market and a internet access technology market. The experimental results on these two real-world datasets show the effectiveness of our proposed model.

3.REQUIREMENT ANALYSIS

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENT

The functional requirements are below:

- Load user rating information about products
- Load user profile data
- Load user social network data about relationships
- Train a Factor based adoption rate function based on three inputs of user rating

3.2 NON - FUNCTIONAL REQUIREMENT

Nonfunctional requirements are the requirements, which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy.

Nonfunctional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as: -

1. Product Requirement
2. Organizational Requirements
3. User Requirement
4. Basic Operational Requirements

3.2.1 PRODUCT REQUIREMENTS

- Scalability: Project should work for any number of users and products Debug
- Ability: Any errs in training the adoption rate function must be displayed in log
- Modularity: The project design should be modular so that any new model for adoption rate prediction can be added.

3.2.2 ORGANIZATIONAL REQUIREMENT

- **Process Standards:** IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world.
- **Design Methods:** Object oriented model designing is done. All design elements are represented in UML and coding done as per UML specification.

3.2.3 USER REQUIREMENT

- Various options or display of information in various forms like charts, pie charts etc. must be available in the system.
- The user must be able to view the adoption rate, target marketing and competitive visualization

3.2.4 BASIC OPERATIONAL REQUIREMENT

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points: -

- **Mission profile or scenario:** The mission of the project is to predict the adoption rate of product by the customers
- **Performance and related parameters:** The performance of the proposed solution is measured in terms of Mean Square Error.

- **Utilization environments:** The system will be very useful in all e-commerce and business service provider

3.3 HARDWARE REQUIREMENTS

We need 1 machine with following minimal configuration

Processor	:	2.2GHz
RAM	:	4GB
Hard Disk	:	50GB

3.4 SOFTWARE REQUIREMENTS

The following are the software requirements

OS	:	Windows
Platform	:	Python3.6
Language	:	Python
Development tool	:	IDLE python.

4.DESIGN

CHAPTER 4 DESIGN

4.1 OVERALL SYSTEM ARCHITECTURE

The modules of the system and the relationship between the modules are documented in the architecture diagram below.

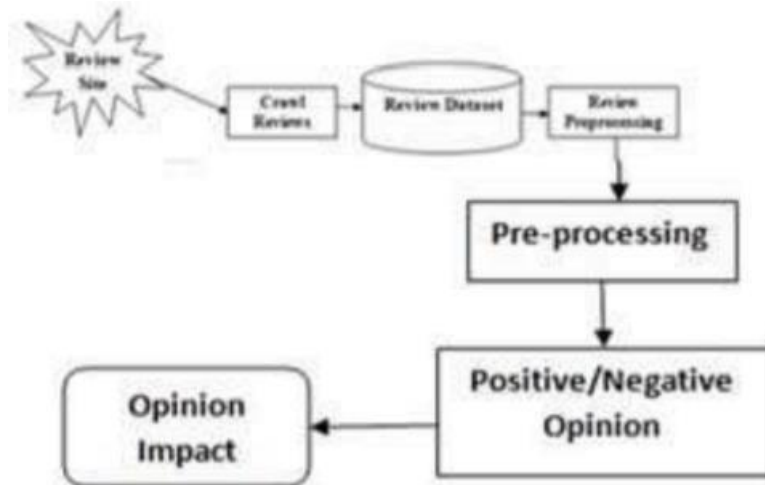


Fig.4.1 System Architecture

4.2 PROPOSED SYSTEM

Machine learning techniques are used for preprocessing, vectorizing, classifier and features are extracted from dataset and prediction is performed for given review and accuracy of model is calculated.

Advantages:

- Various data analysis factors are covered for product analysis.
- Automatic rating calculation is done using machine learning classifier.

4.3 DATA FLOW DIAGRAM

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

Level 0 Data flow diagram

The top-level process and input, output for the top level process is given below

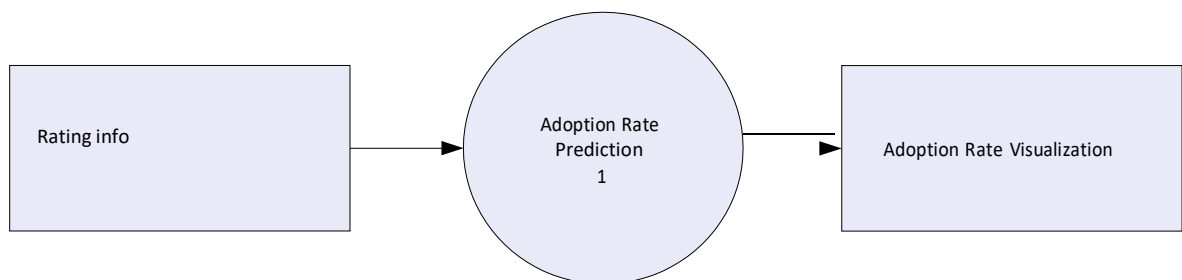


Fig.4.2 Data Flow Diagram Level 0

Adoption rate prediction takes heterogeneous inputs and predicts the rating of the product.

Level 1 Data flow diagram

The adoption rate prediction process is split to sub process as given below.

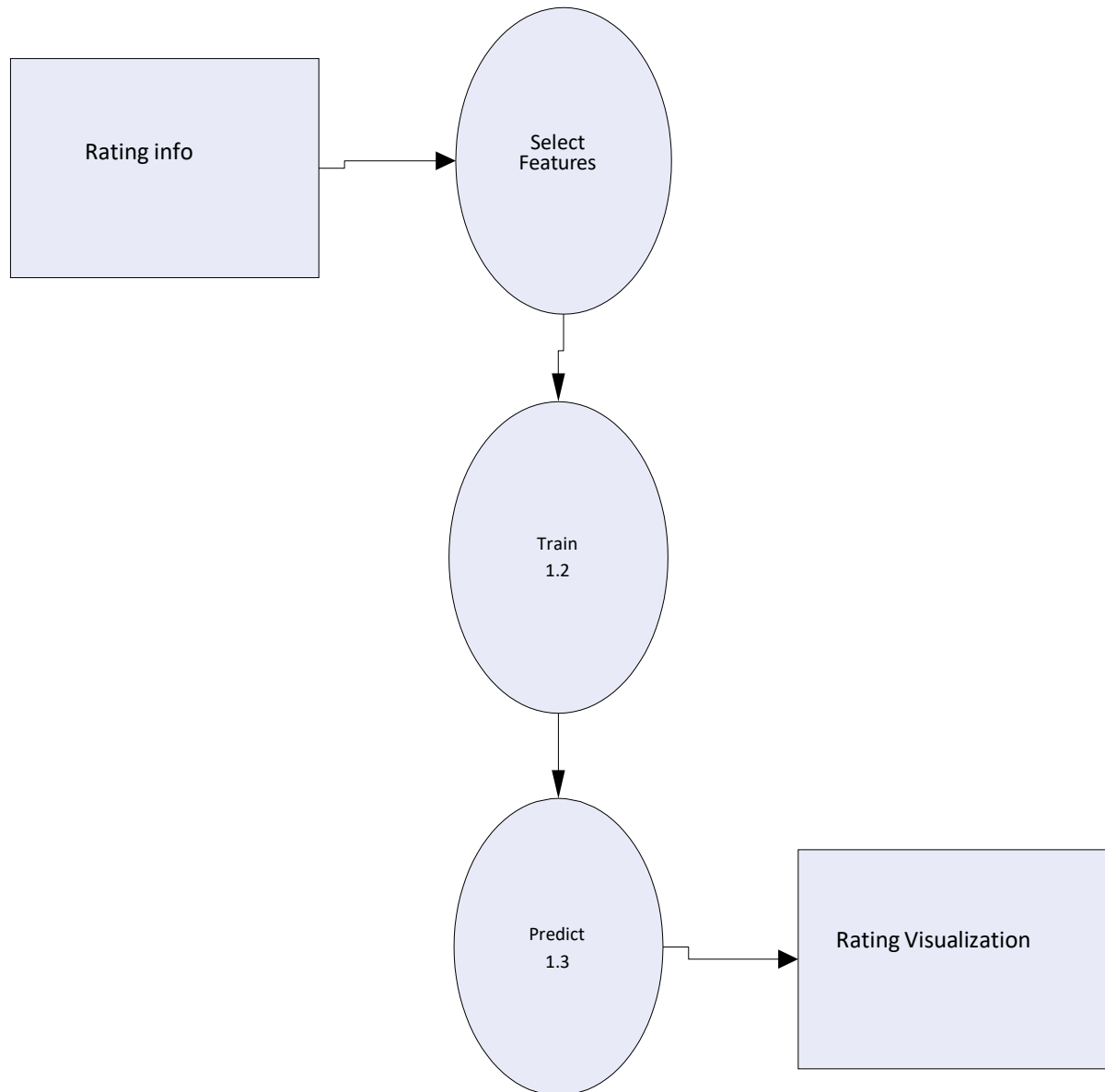


Fig.4.3 Data Flow Diagram Level 1

4.4 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 diagram for training the model

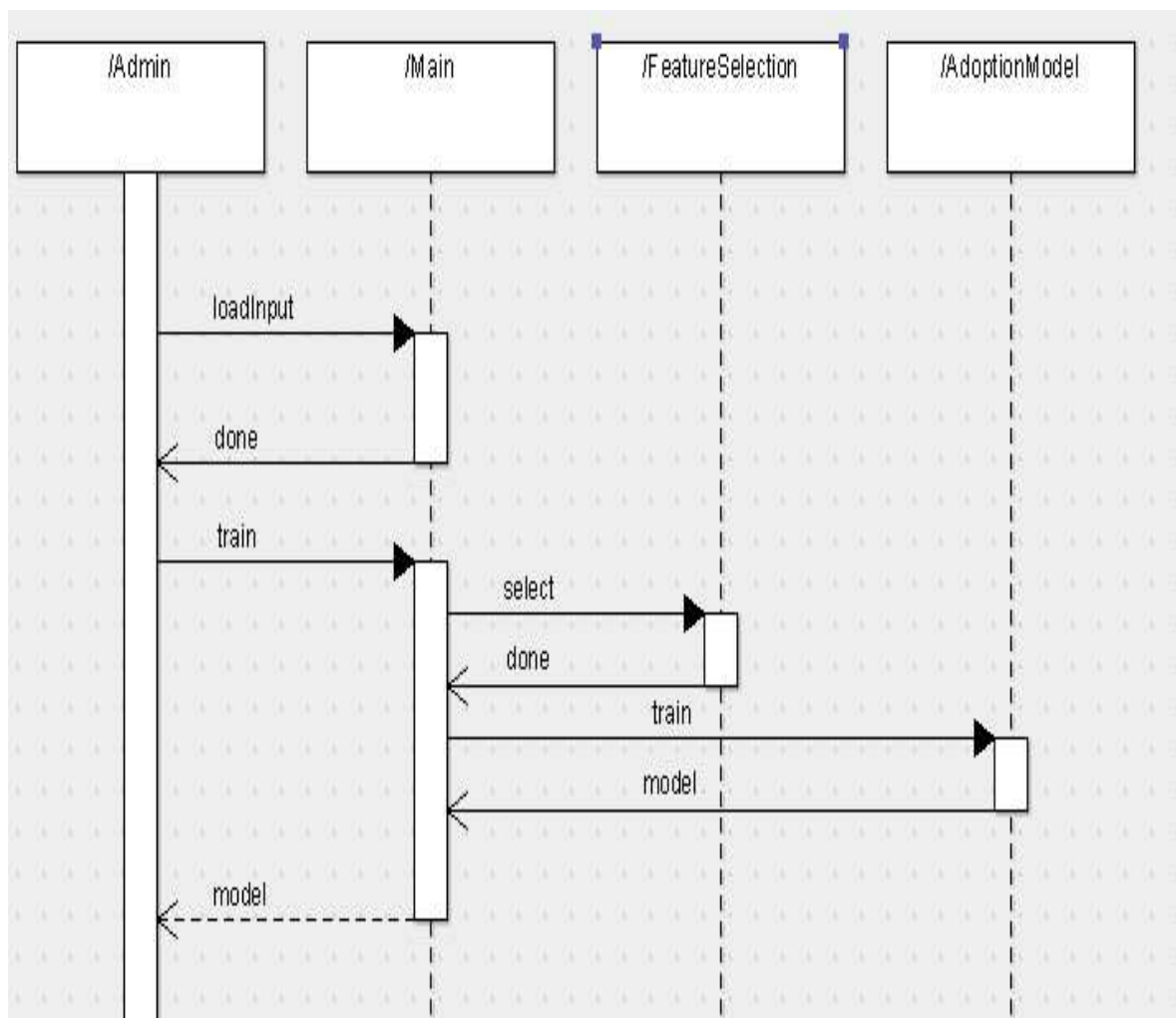


Fig.4.4 Sequence Diagram 1

The interaction of class for the case of prediction is given below

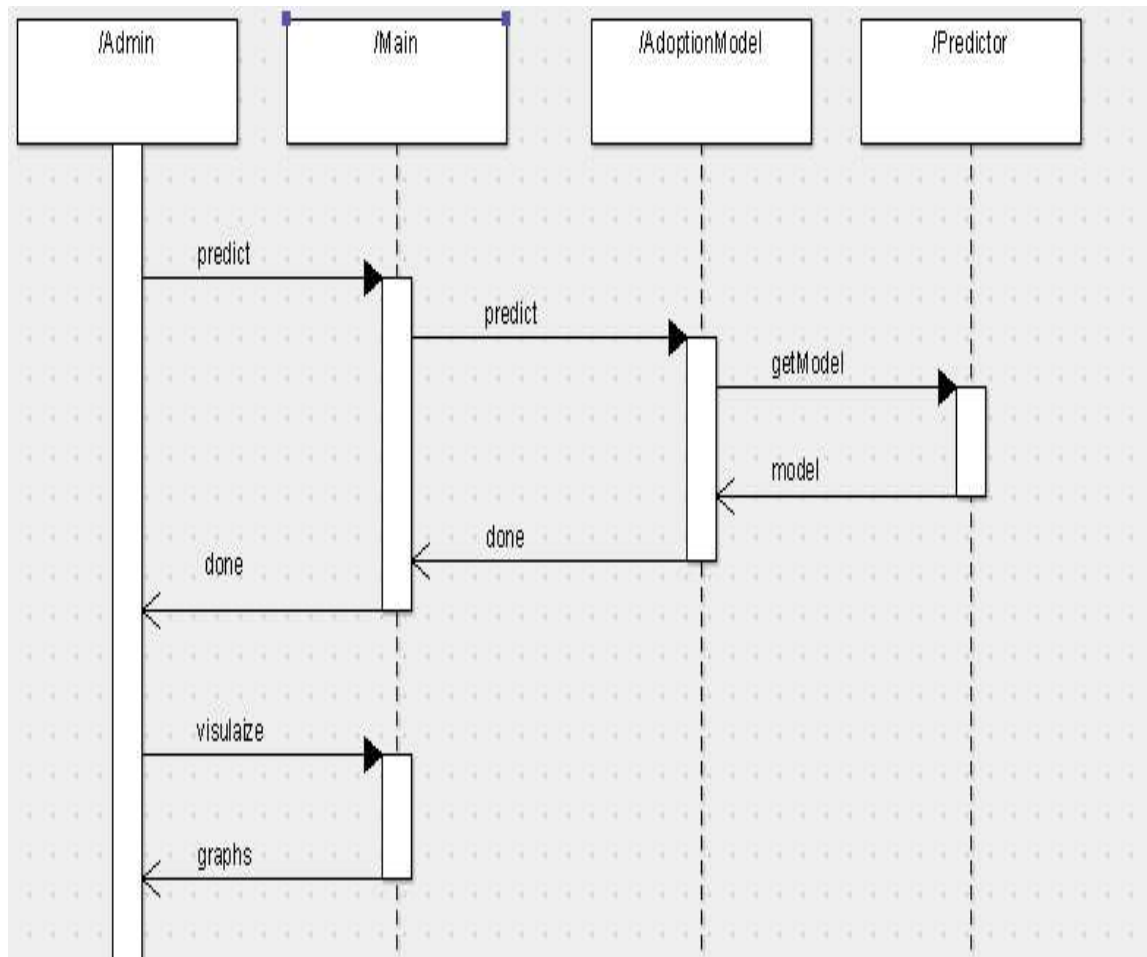


Fig.4.5.Sequence Diagram 2

4.5 USE CASE DIAGRAM

The use case diagram of the system is given below.

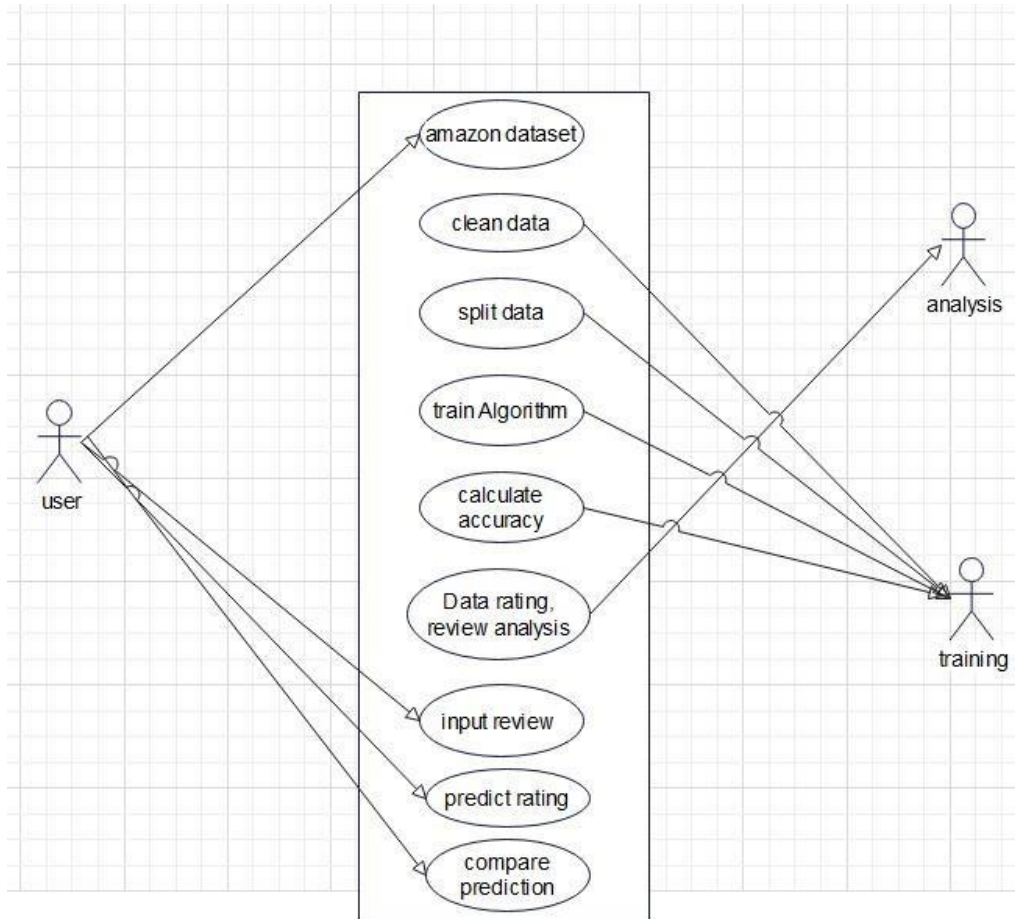


Fig.4.6.Use Case Diagram

Admin can do the following.

1. Load dataset
2. train
3. predict
4. visualize adoption rate

4.6 CLASS DIAGRAM

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, and the relationships between the classes.

The class diagram is shown below.

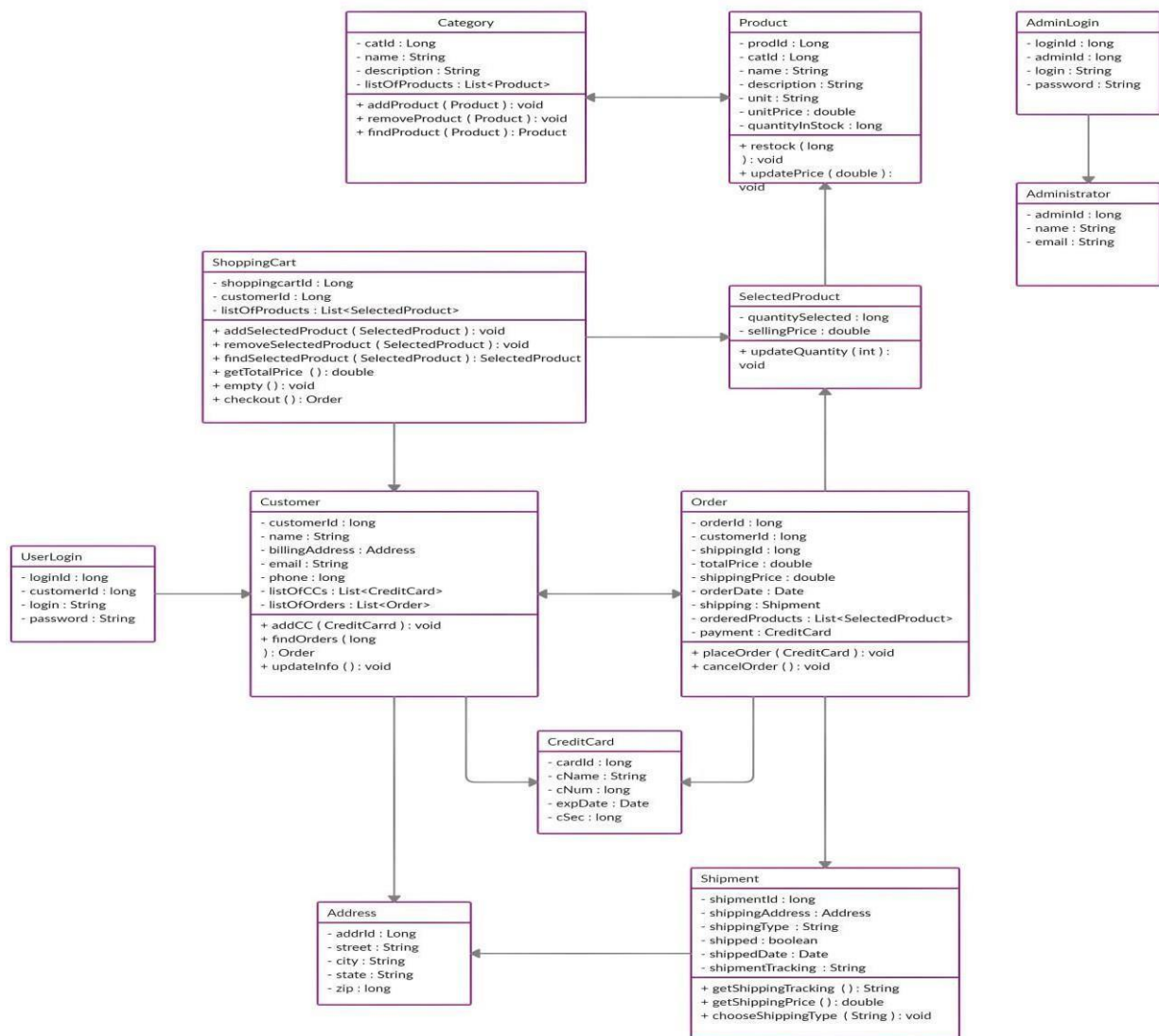


Fig.4.7 Class Diagram

The class diagram has the following classes

Main: This is the user interface class which provides the front end for accessing all the basic functionalities of the system

Factor Selection: This class selects the factors from the input which will influence the product rate adoption.

Adoption Model: This class builds various models about user product adoption behavior and fuse them to give the unified model.

Predictor: This class uses the trained model to predict the adoption rate of products for customers

4.7 PRODUCT Rating MEASURE

Let us consider a competitive market with a set of N users and a set of M products. Users from a social network G , with U denotes the same set of users in the market and the edge, set A represents the relationship between users. E.g., if user u follows user v , then $2A$. The product set B is application dependent that contains all the products that compete with each other in the competitive market, which can be obtained by the merchants or the domain experts. E.g., in a smart device market, the products include iPhone, Windows, Android and so on. Here, to track users' preferences and loyalty to products in a competitive market, we introduce a product adoption rate notion to measure the frequency and regularity of users' preferences to products at each time.

5.IMPLEMENTAION

CHAPTER 5 IMPLEMENTATION

The pseudo code of the modules, their inputs and outputs is detailed in this chapter.

5.1 DATASET

Amazon consumer reviews dataset is used for evaluation. We select reviews in category of electronics.

The dataset is collected from the link:

< <http://jmcauley.ucsd.edu/data/amazon/> >

The reviews are in JSON format and the review is of form:

```
{ "reviewerID": "A2SUAM1J3GNN3B", "asin": "0000013714", "reviewerName": "J. McDonald", "helpful": [2, 3], "reviewText": "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!", "overall": 5.0, "summary": "Heavenly Highway Hymns", "unixReviewTime": 1252800000, "reviewTime": "09 13, 2009" }
```

where

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. 2/3
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

5.2 CSV CONVERSION

The dataset in JSON format is converted to CSV file using the following python

code:

```
import numpy as np

import scipy as sp

import pandas as pd

import json

import gzip

def parse(path):

    g = gzip.open(path, 'rb')

    for l in g:

        yield json.loads(l)

def getDF(path):

    i = 0

    df = {}

    for d in parse(path):

        df[i] = d

        i += 1

    return pd.DataFrame.from_dict(df, orient='index')

df = getDF('electronics_5.json.gz')

df.to_csv('reviews.csv', sep=',', index=False)
```


5.3 PRODUCT RATING PREDICTION

The reviews are split to a window of 1 month and the mean rating of the electronic products in each of window of 1 month is calculated to find the product rate adoption.

Group the reviews of the user in window of 1 month and extract terms from the reviews. Calculate the preference of terms over a window of time to give the important factors influenced the purchase of the product.

The distribution of rating for a product over a period of time is calculated and displayed

```
clf_benchmark = LogisticRegression(random_state=22).fit(X_train, y_train)
example_review_set = pd.DataFrame(y_test[191:197].index).join(data, on=0)
print (pandas_df_to_markdown_table(example_review_set[['Rating', 'Reviews']]))
#reviewsq = map(clean_text, data['Reviews'][0:10])
#inputreview = vectorizer.fit_transform(reviewsq)

#print(clf_benchmark.predict(inputreview))
#print( classification_report(y_test, clf_benchmark.predict(X_test), digits=4))
```

Rating	Reviews
1.0	Got my iphone and the home button was broke
5.0	I loved it and the speed it came in
5.0	Cool
5.0	Works as expected!
1.0	This was a great phone it's time but my 5s was defective. I bought a SE and couldn't be happier with it.
5.0	Good

Fig.5.1 Rating Of Products

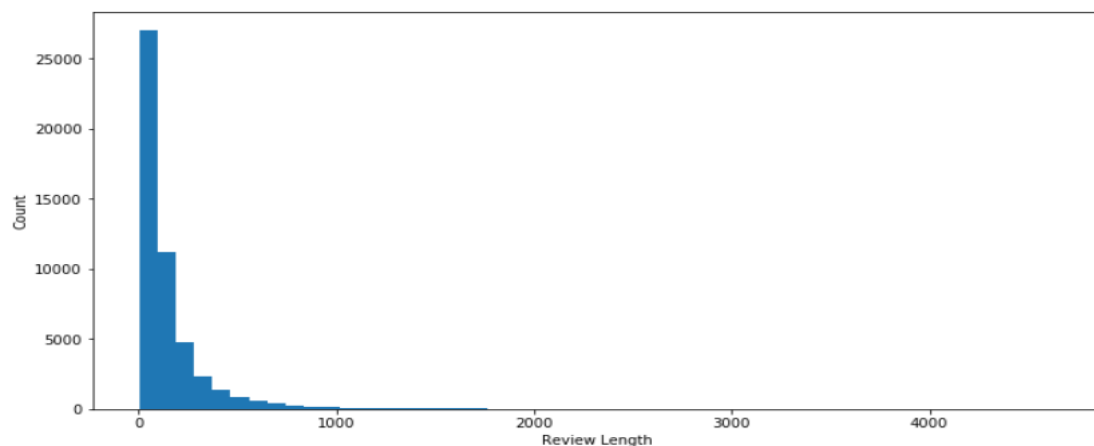


Fig.5.2 Graph Of Rating

5.4 CONCEPTS

5.4.1 DATA MINING

Data Mining is defined as extracting information from huge sets of data. In other words, we can say that data mining is the procedure of mining knowledge from data. The information or knowledge extracted so can be used for any of the following applications:

- Market Analysis
- Fraud Detection
- Customer Retention
- Production Control
- Science Exploration

Data Mining Applications

- Market Analysis and Management
- Corporate Analysis & Risk Management
- Fraud Detection

MARKET ANALYSIS AND MANAGEMENT

Listed below are the various fields of market where data mining is used:

- Customer Profiling - Data mining helps determine what kind of people buy what kind of products.
- Identifying Customer Requirements - Data mining helps in identifying the best products for different customers. It uses prediction to find the factors that may attract new customers
- Cross Market Analysis - Data mining performs Association/correlations.
- Target Marketing - Data mining helps to find clusters of model customers.

Predicting Rating Using Review Analysis

- Determining Customer purchasing pattern - Data mining helps in determining customer purchasing pattern.
- Providing Summary Information - Data mining provides us various multidimensional summary reports.

Corporate Analysis and Risk Management

Data mining is used in the following fields of the Corporate Sector:

- Finance Planning and Asset Evaluation - It involves cash flow analysis and prediction, contingent claim analysis to evaluate assets
- Resource Planning - It involves summarizing and comparing the resources and spending.
- Competition - It involves monitoring competitors and market directions.

DESCRIPTIVE FUNCTION

The descriptive function deals with the general properties of data in the database.

Class/Concept Description

Class/Concept refers to the data to be associated with the classes or concepts. For example, in a company, the classes of items for sales include computer and printers, and concepts of customers include big spenders and budget spenders. Such descriptions of a class or a concept are called class/concept descriptions. These descriptions can be derived by the following two ways:

Mining of Frequent Patterns

Frequent patterns are those patterns that occur frequently in transactional data. Here is the list of kind of frequent patterns: Frequent Item Set - It refers to a set of items that frequently appear together, for example, milk and bread. Frequent Subsequence- A sequence of patterns.

Mining of Association

Associations are used in retail sales to identify patterns that are frequently purchased together. This process refers to the process of uncovering the relationship among data and determining association rules. For example, a retailer generates an association rule that shows that 70% of time milk is sold with bread and only 30% of times biscuits are sold with bread.

Mining of Correlations

It is a kind of additional analysis performed to uncover interesting statistical correlations between associated-attribute-value pairs or between two item sets to analyze that if they have positive, negative or no effect on each other.

Mining of Clusters

Cluster refers to a group of similar kind of objects. Cluster analysis refers to forming group of objects that are very similar to each other but are highly different from the objects in other clusters.

CLASSIFICATION AND PREDICTION

Classification is the process of finding a model that describes the data classes or concepts. The purpose is to be able to use this model to predict the class of objects whose class label is unknown. This derived model is based on the analysis of sets of training data.

- The derived model can be presented in the following forms:
- Classification (IF-THEN) Rules
- Decision Trees
- Mathematical Formulae
- Neural Networks The list of functions involved in these processes are as follows:
- Classification - It predicts the class of objects whose class label is unknown. Its

Predicting Rating Using Review Analysis

concepts. The Derived Model is based on the analysis set of training data i.e. the data object whose class label is well known.

- Prediction - It is used to predict missing or unavailable numerical data values rather than class labels. Regression Analysis is generally used for prediction. Prediction can also be used for identification of distribution trends based on available data.
- Outlier Analysis - Outliers may be defined as the data objects that do not comply with the general behavior or model of the data available.
- Evolution Analysis - Evolution analysis refers to the description and model regularities or trends for objects whose behavior changes over time.

Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels, such as “safe” or “risky” for the loan application data; “yes” or “no” for the marketing data; or treatment A,” “treatment B,” or “treatment C” for the medical data. These categories can be represented by discrete values, where the ordering among values has no meaning

Data classification is a two-step process, consisting of a learning step (where a classification model is constructed) and a classification step (where the model is used to predict class labels for given data).

Learning Step (Model construction): describing a set of predetermined classes

- Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
- The set of tuples used for model construction is training set
- The model is represented as classification rules, decision trees, or mathematical formulae.

Classification Step (Model usage):

for classifying future or unknown objects Estimate accuracy of the model

- The known label of test sample is compared with the classified result from the model
- Accuracy rate is the percentage of test set samples that are correctly classified by the model
- Test set is independent of training set, otherwise over-fitting will occur
- If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

5.4.2 MACHINE LEARNING

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .

Applications Of Machine Learning Algorithms

- Vision processing
- Language processing
- Forecasting things like stock market trends, weather
- Pattern recognition
- Games
- Data mining
- Expert systems
- Robotics

Steps Involved in Machine Learning

- Defining a Problem
- Preparing Data
- Evaluating Algorithms
- Improving Results
- Presenting Results

Types Of Machine Learning

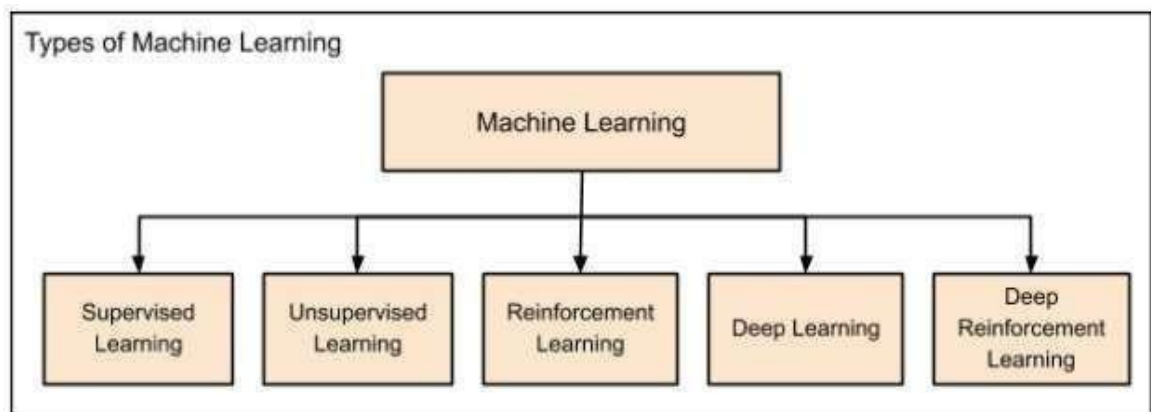


Fig.5.3 Use Case Diagram

Machine Learning is broadly categorized under the following headings. Machine learning evolved from left to right as shown in the above diagram.

- Initially, researchers started out with Supervised Learning. This is the case of housing price prediction discussed earlier.
- This was followed by unsupervised learning, where the machine is made to learn on its own without any supervision.
- Scientists discovered further that it may be a good idea to reward the machine when it does the job the expected way and there came the Reinforcement Learning.

Predicting Rating Using Review Analysis

- Very soon, the data that is available these days has become so humongous that the conventional techniques developed so far failed to analyze the big data and provide us the predictions.
- Thus, came the deep learning where the human brain is simulated in the Artificial Neural Networks (ANN) created in our binary computers.
- The machine now learns on its own using the high computing power and huge memory resources that are available today.
- It is now observed that Deep Learning has solved many of the previously unsolvable problems.
- The technique is now further advanced by giving incentives to Deep Learning networks as awards and there finally comes Deep Reinforcement Learning.

Advantages Of Machine Learning

1. Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus software's; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model.

Predicting Rating Using Review Analysis

As the amount of data, you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

Disadvantages Of Machine Learning

- **Data acquisition**

In the process of machine learning, a large amount of data is used in the process of training and learning. So, this use of data should be of good quality, unbiased. During the process of machine learning with help of software development services, there are also moments when we need to wait. In that period of time new data is being generated and can be used for further process.

- **Time and resources**

During the procedure of machine learning process, the algorithms that help to manage all the functions to manage the data and use of certain data in the process of rectification if any errors this all requires time. And also trusted and reliable resources for the functioning of this system.

- **Interpretation**

When the algorithms help in all these processes and give a resulting output. This given output must be checked for any errors and the correction operation should be followed to get the desired accuracy. And during the selection of this algorithm, we must select that algorithm which you require for the purpose.

- **High error susceptibility**

In the process of machine learning, the high amount of data is used and on the other hand, many algorithms are used and tested. Hence there is a huge change to experience many errors. Because while you are training your dataset at that particular many algorithms are used if there is any mistake.

5.4.3 PYTHON

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

Applications

- web development (server-side)
- software development
- mathematics
- system scripting.



Fig 5.4 Python Logo

Characteristics Of Python

- **Interpreted Language:** Python is processed at runtime by Python Interpreter
- **Easy to read:** Python source-code is clearly defined and visible to the eyes.
- **Portable:** Python codes can be run on a wide variety of hardware platforms having the same interface.
- **Extendable:** Users can add low level-modules to Python interpreter.
- **Scalable:** Python provides an improved structure for supporting large programs than shell-scripts.
- **Object-Oriented Language:** It supports object-oriented features and techniques of programming.

Predicting Rating Using Review Analysis

- Interactive Programming Language: Users can interact with the python interpreter directly for writing programs.
- Easy language: Python is easy to learn language especially for beginners.
- Straightforward Syntax: The formation of python syntax is simple and straightforward which makes it popular.

Advantages Of Python

- Platform independent
- Easy to Read, Learn and Write
- Minimum programming knowledge is required
- Allows complex tasks to be performed in relatively few steps
- Allows simple creation and editing in a variety of text editors
- Allows the addition of dynamic and interactive activities to web pages
- Edit and running of code is fast
- Free and Open-Source

DISADVANTAGES OF PYTHON

- Usually run quite slowly
- Limited access to low level and speed optimization code
- Limited commands to run detailed operations on graphics

5.5 ALGORITHM

Steps to implement linear SVC ALGORITHM

Algorithms

Step 1 - Import the Libraries

We will start by importing the necessary libraries required to implement the SVC Algorithm in Python. We will import the numpy libraries for scientific calculation.

Predicting Rating Using Review Analysis

Step 2 - Fetch the Data

We will fetch the data from csv file using '**pandas_datareader**'. We store this in a data frame 'df'

Step 3 - Split the Dataset

We will split the dataset into **training dataset** and **test dataset**. We will use 70% of our data to train and the rest 30% to test. To do this, we will create a split parameter which will divide the dataframe in a 70-30 ratio.

Instantiate SVM Model

After splitting the dataset into training and test dataset, we will instantiate SVM classifier. Fit the train data by using '**fit**' function. Then, we will calculate the train and test accuracy by using '**accuaracy_score**' function.

Prediction:

Based on given input values data is passed as array to trained model and prediction of RATING AND REVIEWS is displayed in Jupyter notebook.

6. TESTING

CHAPTER 6

TESTING

Framework testing is really a progression of various tests whose basic role is to completely practice the PC based framework. Albeit every test has an alternate reason, all work to confirm that all the framework components have been appropriately incorporated and perform dispensed capacities. The testing procedure is really completed to ensure that the item precisely does likewise what should do. Testing is the last check and acceptance action inside the association itself.

In the testing stage taking after objectives are attempted to accomplish: -

- To attest the nature of the undertaking.
- To find and wipe out any leftover blunders from past stages.
- To accept the product as an answer for the first issue.
- To give operational dependability of the framework.

6.1 UNIT TESTING

Here every module that contains the general framework is tried exclusively. Unit testing centers confirmation endeavors even in the littlest unit of programming configuration in every module. This is otherwise called "Module Testing". The accompanying unit testing table demonstrates the capacities that were tried at the season of programming. The primary section records every one of the capacities which were tried and the second segment gives the portrayal of the tests done.

6.2 INTEGRATION

After successful completion of unit testing or module testing, singular capacities are incorporated into classes. Again, joining of various classes assumes into position lastly mix of front-end with back-end happens.

Integration of capacities into classes

Toward the begin of coding stage just the capacities required in various parts of the system are created. Each of the capacities is coded and tried autonomously. After check of rightness of the distinctive capacities, they are coordinated into their separate classes.

Integration of various classes

Here the distinctive classes are tried autonomously for their usefulness. After confirmation of accuracy of yields in the wake of testing every class, they are incorporated together and tried once more.

Integration of front-end with back-end

The front-end of the undertaking is created in Java Swing environment. The client interface is intended to encourage the client to enter different summons to the framework and perspective the framework's typical and flawed conduct and its yields. The back-end code is then coordinated with the GUI and tried.

6.3 INTEGRATION TESTING

Information can be lost crosswise over interface. One module can adverse affect another. Sub capacities when joined, ought not lessen the sought real capacity. Mix testing is a precise procedure for developing the system structure. It addresses the issues connected with the double issues of confirmation and project development.

TOP-DOWN INTEGRATION

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward, beginning with the main program module. Modules that subordinate to the main program module are incorporated into the structure in either a depth first or breadth first manner

BOTTOM-UP INTEGRATION

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from bottom to up, processing required for modules subordinate to a given level is always available. Therefore, in this case the need for stubs is eliminated.

The following integration testing table shows the functions that were combined into different classes and the class as a whole tested for its functionality.

Predicting Rating Using Review Analysis

Table 6.1 Integration Testing

Classes integrated	Functions integrated in each class	Tests done	Remarks
Class: Main	load Input train Model predict	Class tested to check whether able to to load reviews, train the model from the reviews and able to predict the adoption rate	Success
Class: Features election	select Features loadInput	Class tested to check whether able to load the input and select features from the input for building the model	Success
Class: Predictor	getAdoptionRate	Class tested to check whether able to return the predicted adoption model	Success
Class:Adoption Model	Train	Class tested to check whether adoption model is trained	Success

6.4 VALIDATION TESTING

At the culmination of integration testing, software is completed and assembled as a package. Interfacing errors are uncovered and corrected. Validation testing can be defined in many ways. Here the testing validates the software function in a manner that is reasonably expected by the customer.

Table 6.2 Validation Testing

Functionality to be tested	Input	Tests done	Remarks
Working of Front-End	User interaction with help of a mouse and keyboard	Appropriate forms open when buttons are clicked	Success
Working of loading reviews	User add the review excel sheet in the project folder	System loads the reviews	Success
Working of adoptionmodel predcition	User press the button to predict the reviews	Adoption model is predicted in time scale and for each user for 5 products	Success

6.5 OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Therefore, the output testing involves first of all asking the users about the format required by them and then to test the output generated or displayed by the system under consideration.

The output format is considered in 2 ways

- On screen
- Printed format

Printed format is used for validating the sites crawled, the relevance score of sites, the ranking score of sites.

7.INTERPRETATION OF RESULT

CHAPTER 7 INTERPRETATION OF RESULT

The following snapshots define the results or outputs that we will get after step-by-step execution of all the modules of the system.

```
data.head(10)
```

	Rating	Reviews
0	5.0	I feel so LUCKY to have found this used (phone...
1	4.0	nice phone, nice up grade from my pantach revu...
2	5.0	Very pleased
3	4.0	It works good but it goes slow sometimes but i...
4	4.0	Great phone to replace my lost phone. The only...
5	1.0	I already had a phone with problems... I know ...
6	2.0	The charging port was loose. I got that solder...
7	2.0	Phone looks good but wouldn't stay charged, ha...
8	5.0	I originally was using the Samsung S2 Galaxy f...
9	3.0	It's battery life is great. It's very responsi...

```
[65]: # Sometimes people leave ratings without reviews. We are going to ignore empty reviews.
data = data[data['Reviews'].isnull()==False]

# Get the ratings column.
ratings = data['Rating']

# Get the reviews column, and clean the reviews.
reviews = map(clean_text, data['Reviews'])
print(data['Reviews'][0:10])

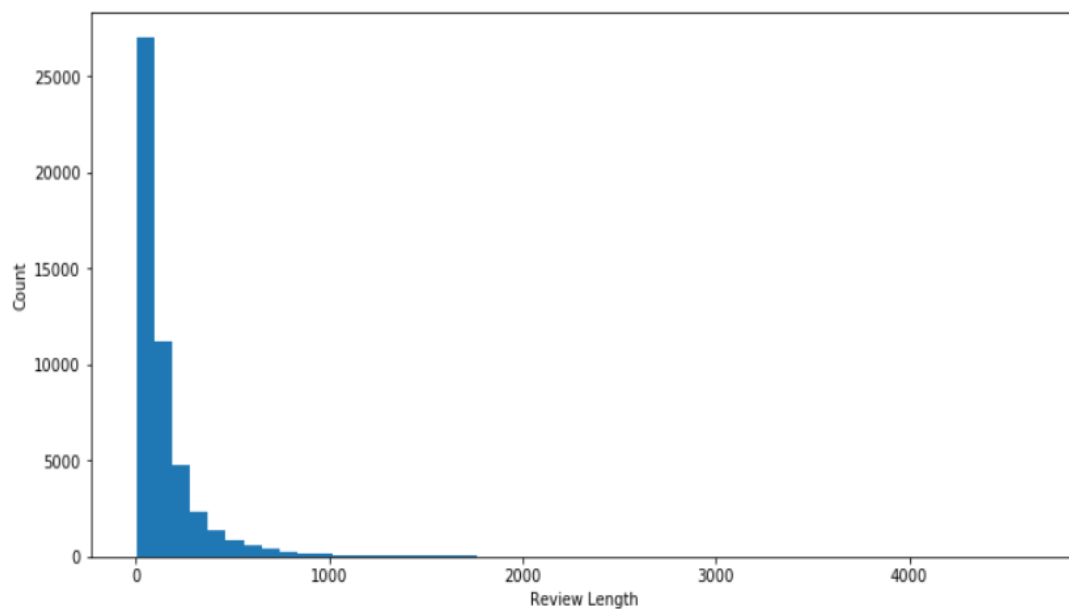
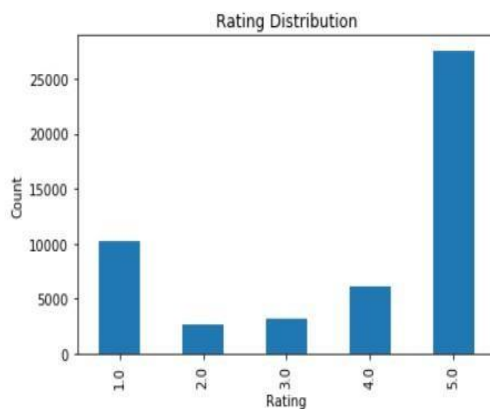
0    I feel so LUCKY to have found this used (phone...
1    nice phone, nice up grade from my pantach revu...
2                                     Very pleased
3    It works good but it goes slow sometimes but i...
4    Great phone to replace my lost phone. The only...
5    I already had a phone with problems... I know ...
6    The charging port was loose. I got that solder...
7    Phone looks good but wouldn't stay charged, ha...
8    I originally was using the Samsung S2 Galaxy f...
9    It's battery life is great. It's very responsi...
```

Explore Data

```
# https://github.com/ipython/ipython/issues/9242
warnings.filterwarnings('ignore', category=DeprecationWarning, message='.*use @default decorator instead.*')

# Create and print a rating distribution graph.
rating_distribution_plt = data.groupby(['Rating']).count().plot(kind='bar', legend=None, title="Rating Distribution")
rating_distribution_plt.set_xlabel("Rating")
rating_distribution_plt.set_ylabel("Count")
```

Text(0, 0.5, 'Count')



Predicting Rating Using Review Analysis

```
clf_benchmark = LogisticRegression(random_state=22).fit(X_train, y_train)
example_review_set = pd.DataFrame(y_test[191:197].index).join(data, on=0)
print (pandas_df_to_markdown_table(example_review_set[['Rating', 'Reviews']]))
#reviewsq = map(clean_text, data['Reviews'][0:10])
#inputereview = vectorizer.fit_transform(reviewsq)

#print(clf_benchmark.predict(inputereview))
#print( classification_report(y_test, clf_benchmark.predict(X_test), digits=4))
```

```
|Rating|Reviews|
|---|---|
|1.0|Got my iphone and the home button was broke|
|5.0|I loved it and the speed it came in|
|5.0|Cool|
|5.0|Works as expected!|
|1.0|This was a great phone it's time but my 5s was defective. I bought a SE and couldn't be happier with it.|
|5.0|Good|
```

8.CONCLUSION

CHAPTER 8

CONCLUSION

In this paper, we studied the problem of tracking and predicting users' adoption rates of products in a competitive market. We first introduced a flexible factor-based decision function to capture the change of users' product adoption rate over time, where various factors from heterogeneous data sources can be generally leveraged. Using this factor-based decision function, we developed the GAM and BPAM models by assuming the generalized and personalized user preference respectively. Furthermore, we presented how to leverage product competition effect into the GAM and BPAM models, and designed the GAM-C and BPAM-C models by simultaneously learning product competition and users' preferences with both generalized and personalized assumptions. Finally, the experimental results on two real-world datasets clearly validated the effectiveness and efficiency of our proposed models.

9. REFERENCES

CHAPTER 9

REFERENCES

- [1] Buettner, R. (2016). Predicting user behavior in electronic markets based on personality-mining in large online social networks: a personality-based product recommender framework. *Electronic Markets*, 1–19. doi:10 . 1007 / s12525-016-0228-z.
- [2] Geng, X., Zhang, H., Bian, J., & Chua, T.-S. (2016). Learning image and user features for recommendation in social networks. (Vol. 11-18-December-2015, pp. 4274–4282). doi:10.1109/ICCV.2015.486.
- [3] Ghazarian, S. & Nematbakhsh, M. (2015). Enhancing memory-based collaborative filtering for group recommender systems. *Expert Systems with Applications*, 42(7), 3801–3812. doi:10.1016/j.eswa.2014.11.042.
- [4] Krzywicki, A., Wobcke, W., Kim, Y., Cai, X., Bain, M., Mahidadia, A., & Compton, P. (2015). Collaborative filtering for people-to-people recommendation in online dating: data analysis and user trial. *International Journal of Human Computer Studies*, 76, 50–66. doi:10.1016/j.ijhcs.2014.12.003.
- [5] Zahra, S., Ghazanfar, M., Khalid, A., Azam, M., Naeem, U., & Prugel-Bennett, A. (2015). Novel centroid selection approaches for kmeans-clustering based recommender systems. *Information Sciences*, 320, 156–189. doi:10 . 1016/j.ins.2015.03.062. Y. Zhang, H. Lu, W. Niu, J. Caverlee, "Quality-aware neural complementary item recommendation", *Proceedings of the 12th ACM Conference on Recommender Systems ser. RecSys '18*, pp. 77-85, 2018, [online] Available: <http://doi.acm.org/10.1145/3240323.3240368>.
- [6] H. Zhao, L. Si, X. Li, Q. Zhang, "Recommending complementary products in e-commerce push notifications with a mixture model approach", *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval ser. SIGIR '17*, pp. 909-912, 2017, [online] Available: <http://doi.acm.org/10.1145/3077136.3080676>.
- [7] C. P. Huynh, A. Ciptadi, A. Tyagi, A. Agrawal, L. Leal-Taixé, S. Roth, "Craft: Complementary recommendation by adversarial feature transform" in

Publishing, pp. 54-66, 2019.

- [8] K. Zhao, X. Hu, J. Bu, C. Wang, "Deep style match for complementary recommendation", AAAI Workshops, 2017, [online] Available: <https://aaai.org/ocs/index.php/WS/AAAIW17/paper/view/15069>.
- [9] O. Barkan, N. Koenigstein, "Item2vec: Neural item embedding for collaborative filtering", 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6, Sep. 2016.
- [10] H. Liu, Z. Hu, A. Mian, H. Tian, X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering", Knowledge-Based Systems, vol. 56, pp. 156-166, 2014.
- [11] FátimaRodrigues, "Product Recommendation based on Shared Customer's Behaviour",Procedia Computer Science Volume 100, 2016, Elsevier 2016.
- [12] Cui, H. & Zhu, M.. (2014). Collaboration filtering recommendation optimization with user implicit feedback. Journal of Computational Information Systems. 10. 5855-5862. 10.12733/jcis10758.
- [13] Gai Li and Qiang Chen, "Exploiting Explicit and Implicit Feedback for Personalized Ranking," Mathematical Problems in Engineering, vol. 2016, Article ID 2535329, 11 pages, 2016.
- [14] Yuan Li, and Kedian Mu, "Matrix Factorization Model with Dual Preferences for Rating Prediction," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), pp. 364–372
- [15] Zhiqiang Zhang, Gai Li, Liyang Wang, Qiang Chen, and Jincai Pan, "One-class collaborative filtering based on rating prediction and ranking prediction," Knowledge-Based Systems, vol. 124, pp. 46–54, 2017
- [16] Yuehua Dong, and Xuelel Liang, "Personalized Recommendation Algorithm Based on Commodity Label," Computational Intelligence and Intelligent Systems, vol. 873, pp. 499–512, 2018.
- [17] Babak Loni, Roberto Pagano, Martha Larson, and Alan Hanjalic, "Top-N recommendation with multi-channel positive feedback using factorization machines," ACM Transactions on Information Systems, vol. 37, no. 2, 2019.

- [18] Wilson, Theresa & Wiebe, Janyce & Hoffmann, Paul. (2009). Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. Computational Linguistics. 35. 399-433. 10.1162/coli.08-012-R1-06-90.

GITHUB LINK

<https://github.com/VeenaGayatriDeverakonda/miniproject>