IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Product Recommendation Using Emerging Technology

[1]Dr. Sai Madhavi, [2]Palthuru Hirematam Aishwarya, [3]Neha Math, [4]Niveditha G

[1] Professor, Guide (CSE), RYMEC ,[2] Student (CSE), RYMEC, [3] Student (CSE), RYMEC, [4] Student (CSE), RYMEC,

Abstract— With the development of the internet and intelligent computing technology, e-commerce is increasingly being used. This recommender system aims to propose the right products to the customers using the best ratings and customer reviews. When a user visits the site and selects a product the site shows him/her the ratings and reviews for that product. And based on their previous views of products and best ratings the system will recommend the products to the customer. Product recommender systems attempt to predict products in which a user might be interested. We aim to fulfill the customer's needs and expectations. Thus the product recommendations are expected to provide recommendations by the user's interest with the help of ratings and reviews on the products.

Index Terms— ***Recommender Systems, E-Commerce, Ratings, Reviews.***

I. INTRODUCTION

Recommender systems are programs that attempt to predict the right product to the customers based on their interests and some given information in their profile. Most existing product recommender systems use the major approaches that include collaborative filtering methods, content-based, hybrid filtering methods which is a combination of both approaches and so on. This paper shows the implementation of simple naive bayes algorithm to recommend the right product to the customers.

II. AREA AND PROBLEM STATEMENT

A. Area:

Machine Learning.

B. Problem:

Change of decision is a problem that occur when the user come across advertisement or offers for other products.

C. Solution:

When a user is in search of offers on products, our recommender system recommends them with quality products on high ratings and reviews given by user. Based on this our recommender system recommends the products to the user.

III. BENEFITS TO THE CUSTOMER/ADMIN

Customer Satisfaction:

Customer has to look at reviews and ratings, in order to find right products, will find better options for good products.

This system will identify and analyses information of product reviews and ratings. According to that, the system will display both positive and negative rating of the products. Which is more useful for the users to get information of that product.

IV. LITERATURE SURVEY

The literature survey helped in understanding the use of different algorithms in recommending systems for various applications. From the literature survey we were able to find the impact of algorithms and try to identify the most suitable one for different recommending systems. The literature survey is as shown in following table. Table shows the Literature survey that concentrates on obtaining the comparisons among different recommendation systems about their implementations, methodologies, advantages, and disadvantages.

PAPER	METHOD	ADVANTAGES	DISADVANTAGES
E-commerce recommendation Opportunity model: Right product; correct time. **AUTHORS:** Y.Zang and J. Wang	Wang Based Promotion Zhang Push.	• Problem solving that suggests products based on time to fix. • Effective estimation	• The decision is made it off without market preference.
Prediction of retail sales and item suggestions using store-level consumer demographics. **AUTHORS:** M. Giering	Giering Inborn-baseline Methodologies.	• High performance. And precision.	• With low usability, response time is improved.
Recommendations from Amazon.com: collaborative filtering Item-to-item. **AUTHORS:** G. Linden, B. Smith, and J. York	Collaborative on Filtering.	• Personalized client shopping experiences. • Elevated optimization/scalability.	• Elevated cost of service. • Poor quality.
The new demographics and polarization of the market. **AUTHORS:** V. A. Zeithaml	Fragmentation.	• Descriptive Analysis and nonparametric on all variables was used by various populations of shoppers.	• Needed to have more parameters considered.
Demographic-based microblog recommendation framework for products. **AUTHORS:** W. X. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li	Microblogs.	• Detect users from their microblogs to buy intentions. • Viability in matching the personal data of users gathered from the media.	• Elevated cost of service.
Recommender list focused on feedback of consumer goods. **AUTHORS:** Silvana Aciar, Debbie Zhang, John Debenham, Simeon Simoff.	Conduct of customer exchange.	• Improving the recommendation's efficacy using experience of consumers.	• Elevated cost of service.
A case study focused on Purchase Data in a Recommender Framework. **AUTHORS:** Bruno Pradel, Savaneary Sean, Julien Delporte, Nicolas Usunier.	Filtering Collaboratively.	• Shortest optimization based on best output rules. • Well-performing algorithms.	• High information Sparity problem.
We suggest: A product review-based recommendation framework. **AUTHORS: Malony Alphonso, Vedita Velingker.**	Mining Belief, Excavation of knowledge.	• Enhancing the recommendation's efficacy. • Decision making.	• Elevated cost of service.

V. IMPLEMENTATION

A. Methodology

The approach to build a recommending system considers emerging technology and studies done on the requirements specifications. Usage of python language is done as it has good libraries, packages and supporting tools such as Numpy, pandas, keras, sklearn, matplotlib, to handle the operations with suitable algorithms at the back-end to intersect with database. PythGWT (Google Web Toolkit) is used to maintain front-end applications.

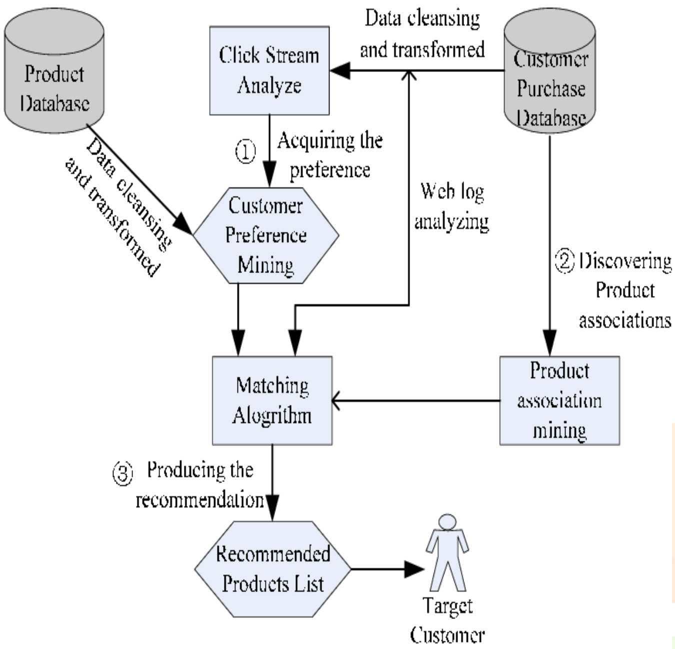

Fig 1: Overview of recommeder process of the system

B. Existing System

With the advent of emerging technologies and rapid growth of internet, the world is moving towards-world where most of the things are digitalized and available on the mouse click. Most of the commercial transactions are performed on internet with the help of on-line shopping. The huge amount of data puts an extra overload to the user in performing online task. Product recommendation techniques are being used widely to reduce this extra overload and recommend the scrutinized product to the customers. For a single item, there are many brands and models available. The opportunity for the customer to select from a large number of products increases the burden of information processing before he decides which products meet his needs. If the customer is not sure about product of his choice, he may face the problem of information overload. He may come across a situation, where he may be unable to decide which product to buy.

Dis-Advantages: The opportunity for the customer to select from a large number of products increases the burden of information processing before he decides which products meet his needs.

C. Proposed System:

This paper makes five contributions to the understanding of recommender systems in E-commerce. First, we provide a set of recommender system examples that span the range of different applications of recommender systems in E-commerce. Second, we analyze the way in which each of the examples uses the recommender system to enhance revenue on the site. Third, we describe a mapping from applications of recommender systems to a taxonomy of ways of implementing the applications. Fourth, we examine the effort required from users to find recommendations. Fifth, we describe a set of suggestions for new recommender system application based on parts of our taxonomy that have not been explored by the existing applications.

Advantages: The products can be recommended based on the top overall sellers on the site, based on the demographics of the customer, or based on an analysis of the past buying behavior of the customer as a prediction for future buying behavior. Enabling individual personalization for each customer.

D. UML Diagram

The objective for UML is to turn into a typical language for making models of article arranged PC programming. In its present structure UML is involved two significant segments: a Meta-model and a documentation. Later on, some type of strategy or procedure may likewise be added to; or related with, UML. The UML speaks to an assortment of best designing practices that have demonstrated fruitful in the displaying of huge and complex frameworks. It is a significant piece of creating objects arranged programming an the product advancement process. The UML utilizes for the most part graphical documents to communicate the plan of programming ventures.

Use Case Diagram:

An utilization case chart in the Unified Modeling Language (UML) is a kind of conduct graph characterized by and made from a Use-case examination. Its motivation is to introduce a graphical outline of the usefulness gave by a framework as far as on-screen characters, their objectives, and conditions between those utilization cases.

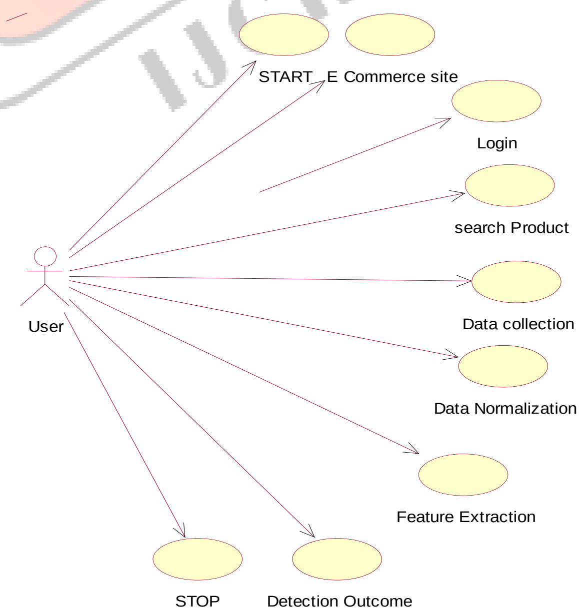

Fig 3: Use Case Diagram

Class Diagram

In programming building, a class graph in the Unified Modeling Language (UML) is a kind of static structure outline that portrays the structure of a framework by indicating the frameworks classes, their properties, tasks, and the connections among the classes. It clarifies which class contains data.

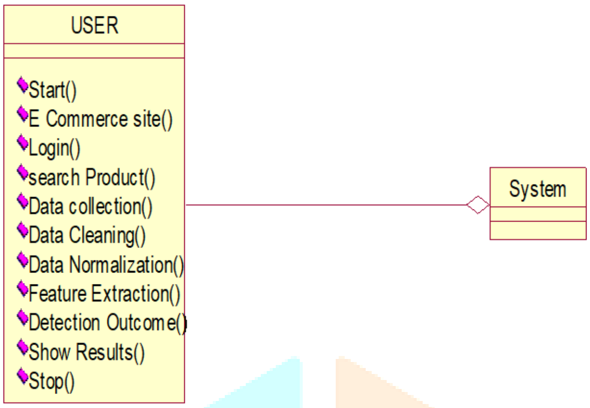

Fig 4: Class Diagram

Sequence Diagram

An arrangement outline in Unified Modeling Language (UML) is a sort of connection graph that shows how procedures work with each other and in what request. It is a build of a Message Sequence Chart. Arrangement outlines are now and then called occasion graphs, occasion situations, and timing charts.

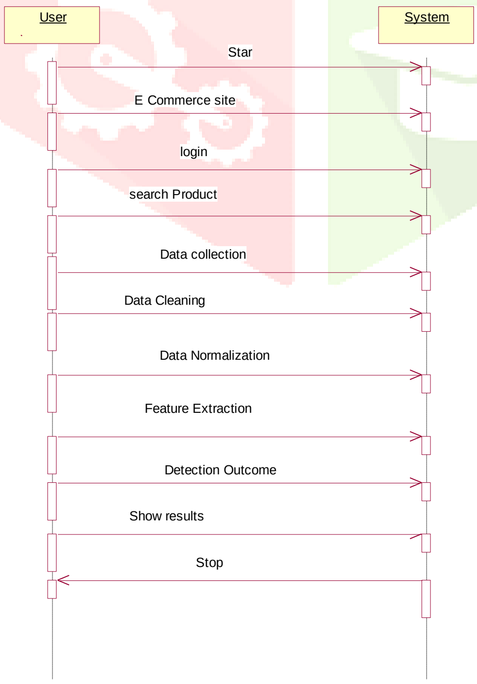

Fig 4: Sequence Diagram

E. Software Requirements

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- Python ide 3.7 version (or)
- Anaconda 3.7 (or)
- Jupiter (or)
- Google colab

F. Hardware Requirements

Minimum hardware requirements are very dependent on the particular software being developed by a given Python/ Canopy/ VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- Operating system: windows, linux
- Processor: minimum intel i3
- RAM: minimum 4GB
- Harddisk: minimum 250GB

G. Algorithm

- Naive Bayes learns from supervised learning algorithm it is simple but also effective in performing complex solutions.

  p(A|B) = p(B|A) * p(A) / p(B)

  Assume A - Category
  B - Test data
  p(A|B) - Category given the Test data

  Here,
  Ignoring p(B) in the denominator (Since it remains same for every category)
- Collaborative Filtering builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.
- Sentiment analysis is a machine learning tool that analyzes texts for polarity, from positive to negative. By training machine learning tools , machines automatically learn how to detect sentiment without human input.

H. Main Code

- Naive Bayes Implementation:

```
def naive_bayes(test):
    results={}
    for i in dataset.keys():
    # Category Probability
    # Number of items in category/total no. of items
    cat_prob=
float(no_of_items[i])/sum(no_of_items.values())
    # p(test data | category)
    test_prob1=test_prob(test,i)

    results[i]=test_prob1*cat_prob
```

```
    return results

    """print ('Enter the sentence')
      text=input()
      result=naive_bayes(text)

      if result['1'] > result['-1']:
         print ('positive')
      else:
      print ('negative')"""
```

- Sentimental Analysis:

To analyze weather the given reviews are positive/ negative. This works against available dataset to obtain the feature set that holds simplified form of the reviews. Later, which is passed on to another function that calculates the weighted probability of a word for a particular category of items.

```
import re
import csv
import os.path
BASE = os.path.dirname(os.path.abspath(__file__))

fh=open(os.path.join(BASE,"dataset.csv"))

# The delimiter in the csv file is '+' instead of comma. This
#was done to compromise with the commas in the sentence
in #the sentence of the dataset used.

reader = csv.reader(fh, delimiter='+')

# It is the dictionary that has the data : {
#label(positive/negative) : { word : count of number of
#occurences of the word } }

dataset={}

# It is the dictionary that keeps the count of records that are
#labeled a label l for each label l
#That is, { label l : No. of records that are labeled l }

no_of_items={}

# This is the dictionary that contains the count of the
occurences of word under each label
# That is, { word : { label l : count of the occurence of word
#with label l } }

feature_set={}

# For each sentence in dataset
for row in reader:
    # Initialize the label in the dictionary if not present
#already

no_of_items.setdefault(row[1],0)
# Increase the count of occurence of label by 1 for every
#occurence
no_of_items[row[1]]+=1

# Initialize the dictionary for a label if not present
#dataset.setdefault(row[1],{})
# Split the sentence with respect to non-characters, and
#donot split if apostophe is present

split_data=re.split('[^a-zA-Z\']',row[0])

# For every word in split data
for i in split_data:
   # Removing stop words to a small extent by ignoring
   #words with length less than 3
   if len(i) > 2:
      # Initialize the word count in dataset
      dataset[row[1]].setdefault(i.lower(),0)
            # Increase the word count on its occurence
      #with label row[1]

      dataset[row[1]][i.lower()]+=1

      # Initalze a dictionary for a newly found word in
      #feature set

      feature_set.setdefault(i.lower(),{})

      # If the label was found for the word, for the first time,
      #initialize corresponding          #count value for
      #word as key

      feature_set[i.lower()].setdefault(row[1],0)

      # Increment the count for the word in that label

      feature_set[i.lower()][row[1]]+=1



    ##To get weighted probability

import re from .sensitive_data import
dataset,feature_set,no_of_items

# To calculate the basic probability of a word for a category

def calc_prob(word,category):

   if word not in feature_set or word not in #dataset[category]:

      return 0

   return
      float(dataset[category][word])/no_of_items[category]


# Weighted probability of a word for a category
def weighted_prob(word,category):
         # basic probability of a word - calculated by
         #calc_prob
   basic_prob=calc_prob(word,category)

         # total_no_of_appearances - in all the #categories
   if word in feature_set:
            tot=sum(feature_set[word].values())
   else:
       tot=0
```

```
        # Weighted probability is given by the #formula
        # (weight*assumedprobability +
        #total_no_of_appearances*basic_probability)#/(tot
        al_no_of_appearances+weight)
    # weight by default is taken as 1.0
    # assumed probability is 0.5 here

        weight_prob=((1.0*0.5)+(tot*basic_prob))/(1.0+tot
)
    return weight_prob


# To get probability of the test data for the given category
def test_prob(test,category):
    # Split the test data
    split_data=re.split('[^a-zA-Z][\'][ ]',test)

    data=[]
    for i in split_data:
        if ' ' in i:
            i=i.split(' ')
            for j in i:
                if j not in data:
                    data.append(j.lower())
        elif len(i) > 2 and i not in data:
            data.append(i.lower())

    p=1
    for i in data:
        p*=weighted_prob(i,category)
    return p
```

CONCLUSION

In this paper, we discussed Recommender systems are a key way to automate mass customization for E-commerce sites. They will become increasingly important in the future, as modern business are increasingly focused on the long-term value of customers to the business. E-commerce sites will be working hard to maximize the value of the customer to their site, providing exactly the pricing and service they judge will create the most valuable relationship with the customer. Since customer retention will be very important to the sites, this relationship will often be to the benefit of the customer as well as the site- but not always. Important ethical challenges will arise in balancing the value of recommendations to the site and to the customer.

By this we conclude that our recommender system recommend the right products to the customer based on highest ratings and reviews.

REFERENCES

[1] J. Bennett, S. Lanning, and N. Netflix. The Netflix price. In In KDD Cup and Workshop, KDD '07, 2007.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering pages 43-52. Morgan Kaufmann, 1998.

[3] Y. Ding and X. Li. Time weight collaborative filtering. In Proc. Of CIKM '05, pages 485-492, New York, NY, USA, 2005. ACM.

[4] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In Proc. Of CIKM '01 conference, pages 247-254, New York, NY, USA, 2001.

[5] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proc. Of ACM KDD '08 conference, pages 426-434, New York, NY, USA, 2008. ACM.

[6] E. Kim, W. Kim, and Y. Lee (2000). "Purchase propensity prediction of EC customer by combining multiple classifier based on GA", In Proceedings of International Conference on Electronic Commerce 2000, pages 274-280.

[7] J. B. Schafer, J. A. Konstan and J, Riedl (2001). "E-commerce recommendation applications", Data Mining and Knowledge Discovery, volume 5, issue 1-2, pages 115-153.

[8] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes and M. Sartin (1999). "Combining content-based and collaborative filters in an online newspaper". In Proceedings of ACM SIGIR "99 Workshop on Recommender Systems, Berkeley, CA.

[9] B. Sarwar, G. Karypis, J. Konstan and J. Riedl (2000). "Analysis of recommendation algorithms for e-commerce", In Proceedings of ACM E-Commerce 2000 Conference, pages 158- 167.

[10] G. Xu (2008). "Web Mining Techniques for Recommendation and Personalization", PhD Thesis submitted to The School of Computer Science & Mathematics, Faculty of Health, Engineering &Science, Victoria University, Australia.

[11] B. Mobasher, R. Cooley, and J. Srivastava (2000). "Automatic personalization based on web usage mining", Communications of the ACM, volume 43, issue 8, pages 142-151.

[12] Aciar, S., Zhang, D., Simoff, S., & Debenham, J. (2006, December). Recommender system based on consumer product reviews. In 2006 IEEE/ WIC/ ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06) pages 719-723. IEEE.

[13] Pradel, B., Sean, S., Delporte, J., Guerif, S., Rouveirol, C., Usunieer, N., & Dufau-Jeol, F. (2011, August). A case study in a recommender system based on purchase data. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining pages 377-385.

[14] Velingker, V., & Alphonso, M. We Recommend: Recommender System based on Product Reviews.

[15] [9]. Applications:

i). Source:
https://tech.flipkart.com/e-commerce-recommendations-using-machine-learning-5002526e531a?gi=bb3c3284c735#:~:text=We%20use%20Point%2Dwise%20Logistic,to%20rank%20the%20Recommended%20products

ii). Source:
https://www.computer.org/publications/tech-news/trends/amazon-all-the-research-you-need-about-its-algorithm-and-innovation#:~:text=Instead%2C%20Amazon%20devised%20an%20algorithm,item%2Dbased%20collaborative%20filtering.%E2%80%9D

iii). Source:
https://amp.theatlantic.com/amp/article/575212/

iv). Source:
https://analyticsindiamag.com/the-ai-behind-instagram-explore/#:~:text=The%20Instagram%20Explore%20is%20a,artificial%20intelligence%20and%20machine%20learning.

v). Source:
https://www.slideshare.net/planetcassandra/e-bay-nyc

vi). Source:
https://www.codecademy.com/articles/how-netflix-recommendation-works-data-science#:~:text=Instead%2C%20Netflix%20uses%20the%20personalized,data%20to%20make%20informed%20suggestions

vii). Source:
http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/#:~:text=The%20Amazon%20Recommendations%20Secret%20to%20Selling%20More%20Online,-Rejoiner&text=%E2%80%9CJudging%20by%20Amazon's%20success%2C%20the,the%20same%20time%20last%20year