**A**

**PROJECT REPORT**

**ON**

**PRODUCT RECOMMENDATION SYSTEM FOR E-COMMERCE USING COLLABORATIVE FILTERING AND TEXT CLUSTERING**

**Submitted in partial fulfillment of the requirements**

**For the award of Degree of**

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**Submitted By**

**V.CHAKRADHAR REDDY 245318737055**

**G.DEEPAK PAVAN 245318737013**

**M.A.WASEEM SHOAIB 245318737025**

**Under the guidance**

**Of**

**P.NAGESWARA RAO**

**Assistant Professor**

****

**Department of Information Technology**

**NEIL GOGTE INSTITUTE OF TECHNOLOGY**

****

**NEIL GOGTE INSTITUTE OF TECHNOLOGY**

A Unit of Keshav Memorial Technical Education (KMTES)

Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad

**2012-2016**

**CERTIFICATE**

*This is to certify that the project work entitled* “**PRODUCT RECOMMENDATION SYSTEM FOR E-COMMERCE USING COLLABORATIVE FILTERING AND TEXT CLUSTERING””** *is a bonafide work carried out by* **V.CHAKRADHAR REDDY(245318737055), G.DEEPAK PAVAN(245318737013), M.A. WASEEM SHOAIB(245318737025)** of IV year VIII semester**Bachelor of Engineering** *in* **INFORMATION TECHNOLOGY** *by Osmania University, Hyderabad during the academic year* **2018-2022** *is a record of bonafide work carried out by them*. *The results embodied in this report have not been submitted to any other University or Institution for the award of any degree*

**Internal Guide Head of Department**

Mr. P.NAGESWARA RAO Dr. K.MADHURI

**External Examiner**

**II**

****

**NEIL GOGTE INSTITUTE OF TECHNOLOGY**

A Unit of Keshav Memorial Technical Education (KMTES)

Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad

# **DECLARATION**

We hereby declare that the Major Project Report entitled, “**PRODUCT RECOMMENDATION SYSTEM FOR E-COMMERCE USING COLLABORATIVE FILTERING AND TEXT CLUSTERING”** submitted for the B.E degree is entirely our work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree.

**Date:**

**V.Chakradhar Reddy(245318737055)**

**G.Deepak Pavan(245318737013)**

**M.A.Waseem Shoaib(245318737025)**

**III**

**ACKNOWLEDGEMENT**

We are happy to express our deep sense of gratitude to the principal of the college **Dr. D Jaya Prakash, Professor**, Neil Gogte Institute of Technology, for having provided us with adequate facilities to pursue our project.

We would like to thank**,** **Dr.K. Madhuri**, **Head of the Department,** Information Technology, Neil Gogte Institute of Technology, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.

We would also like to thank my internal guide **Mr P.Nageshwar Rao, Assistant Professor** forhis Technical guidance & constant encouragement.

We sincerely thank our seniors and all the teaching and non-teaching staff of the Department of Computer Science & Engineering and Information Technology for their timely suggestions, healthy criticism and motivation during the course of this work.

Finally, we express our immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to our need at the right time for the development and success of this work.

**V.Chakradhar Reddy(245318737055)**

**G.Deepak Pavan(245318737013)**

**M.A.Waseem Shoaib(245318737025)**

**IV**

**ABSTRACT**

Recommender system is the most important technology in e-commerce .It is used to suggest valuable products for the customer and improve their business intelligence. Since they learn from the customer’s browsing habits to come up with recommendations, they tend to recommend products more from categories that the user has visited and purchased from before. Collaborative filtering is a technique which is used to suggest information from similar kinds of users. Scalability is the biggest challenge in collaborative filtering recommender system. When more number of users is increasing in the site the system should provide accurate recommendations for the super user. We use divisive hierarchical clustering approach to overcome this scalability issue when more number of users increases in terms of neighborhood size. Session based collaborative filtering tries to solve major drawbacks offered by these algorithms and obtains relevant recommendations for users.

The massive growth of internet in the past decade has lead to the birth of many E-Commerce websites and applications. More and more people prefer shopping online rather than going to retail stores. The main idea of online shopping is to ease the shopping experience by getting personalized recommendations of products. This is also what the E-Commerce websites are expected to do. The present recommendation system is ineffective because it doesn’t handle three main problems: Limited resource, cold start and data valid time. The recommendation system consists of user model, recommended model and recommendation algorithm. This paper includes the proposed model that focuses on the improvement to the recommendation algorithm by providing solutions to limited resource and cold start problem. The proposed system aims at better customer satisfaction.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table of Contents** | | | |
|  | | | Page No. |
| 1 |  | **Introduction**  DECLARATION  ACKNOWLEDGEMENT  ABSTRACT | III  IV  V |
|  | 1.1 | Problem statement | 7 |
|  | 1.2 | Motivation | 8 |
|  | 1.3  1.4 | Scope  Outline | 8  9 |
| 2 |  | **Literature Survey** |  |
|  | 2.1  2.2 | Existing System  Proposed System | 11  12 |
|  |  |  |  |
| 3 |  | **Software Requirements Specifications** |  |
|  | 3.1 | Overall Description | 13 |
|  | 3.2 | Operating Environment | 13 |
|  | 3.3 | Functional Requirements | 14 |
|  | 3.4 | Non-Functional Requirements | 15 |
| 4 |  | **Design Diagrams** | 17-21 |
|  |  |  |  |
| 5 |  | **Implementation** |  |
|  | 5.1 | Sample Code | 22-31 |
| 6 |  | **Testing** |  |
| 7 |  | **Screenshots** | 32-33 |
| 8 |  | **Conclusion & References** | 34-36 |

**CHAPTER-I**

* 1. INTRODUCTION:-

In the past decade internet usage has seen an exponential growth. This growth has given opportunity for other businesses which rely on it to grow and prosper too. One such avenue is E-commerce. E-commerce is growing rapidly and with increase in popularity of E-Commerce, Recommendation has become very important too. Recommendation means providing relevant suggestions to the user as per his/her interest and need. Recommendations to users can be given based on price, area of living, wish listed items, cart items, searched items and previous purchased items. Recommendation systems improve user experience, boost sales and enhance user's engagement. Our proposed recommendation system will recommend products both for new users and existing users. This recommendation system uses model based collaborative filtering and recommends products based on rating and previous purchase history of the old users. Also the new users will get recommendations of new products, trending products and products on sale. Existing users will get recommendations based on recently viewed products, complementary products, etc. Since we're setting up a new e-commerce website, initially there are no user ratings for different products, so in this case, recommendations are made based on the textual clustering analysis of product description. Model based Collaborative Filtering along with Textual Clustering will help us in improving accuracy and targeting all types of users. E Commerce is gaining popularity and the recommendation system with E Commerce is like an icing on the cake.

Those recommender systems are able to give out some recommendations for customer, but, from the point of view of customers, there are still some problems which are needed to be solved.

From the point of view of existing E-Commerce recommendation system, some of the problems are:

Limited resource: Personal recommendations for products with limited quantity should happen with priority, so that consumers can buy with ease.

Data valid time: The records which are recorded to be older than three months must not be recommended because such records may become outdated and may not be accurate.

Cold start: The visitor cold start means that a recommendation system meets a new visitor for the first time that is there is no user history and the system doesn’t know the personal preferences of the visitor. Getting to know visitors is crucial in creating a great shopping experience for them.

1.2 motivation

We’ll look at short-term goals and interests in the context of a user accessing an online product catalog The user will reveal his/her constraints and preferences in the form of a query (or sequence of queries) Products in the catalog will have descriptions these will tend to be structured descriptions The recommender system will be primarily content-based: matches the query against the product descriptions reactive: makes recommendations on request.

1.3 scope

Recommender systems are widely used in several different domains for the recommendation of articles, music, movies, and even people. Portals such as Amazon and Submarino use recommender systems to suggest products to their customers. Meanwhile, social networks such as LinkedIn and Facebook use them to suggest new contacts.

To accomplish that, the most used techniques employed in recommender systems are The collaborative filtering and content-based systems. The collaborative filtering does not take into account the type of items, nor their attributes. It takes exclusively into account the expressed opinion about the other items in order to make recommendations. Meanwhile, content-based filtering uses the knowledge it has of the items and their attributes to make recommendations.

Evaluate the benefits to recommender systems originated from the knowledge provided by the social network. This database has missing values that are treated as non-evaluated items. Recommender systems try to predict the user's evaluation of an item that has not yet been evaluated. Based on the concept that people who are more closely connected have more influence on each other's opinions (Mendes 2008), this article will try to predict these missing values based on relations, so as to generate more relevant results.

Evaluate the applicability and imputation efficiency of the missing evaluations with refeeds using the identified degrees of separation among participants. Verify, in contrast to the traditional approach, if there are scenarios in which this application is more useful.

1.4 outline

Model-based collaborative filtering works best when a user has bought and rated many products, but this is not possible for new users or new products. In this case, we get irrelevant and erroneous recommendations. To overcome this we have used textual clustering of the description of product. Using both model-based collaborative systems and textual clustering simultaneously we obtain higher accuracy in terms of meaningful recommendation and higher optimization in terms of computing time.

The powerful recommendation system will recommend products both for new users and existing users. This recommendation system will use collaborative filtering and will recommend products based on rating and previous purchase history of the user. Also, the new users will get recommendations of new products, trending products and products on sale. Existing users will get recommendations based on recently viewed products, complementary products, etc. Since we're setting up a new e-commerce website, initially we won't have user ratings on the different products, so in this case, we will be using textual clustering of the product description. Model based CollaborativeFiltering and Textual Clustering together will make the system more efficient. Recommendation System will increase the sale and will be helpful to both the buyers and the sellers.

These are the techniques used generally for recommendation systems but have used:

1. Collaborative Filtering
2. Textual Clustering

Collaborative Filtering targets the user’s needs as it is related to the needs, purchase history, wish list or seen items by the users. It takes into account all these things and recommends the highest correlated product or a list of products to users to reduce time and increase efficiency. It also depends on the similar user’s interest who is interested in similar items or purchased similar products.

Textual Clustering is a clustering technique to cluster text-related documents. It targets the new users as our system will be unable to get any information about the user at the very beginning but our recommendation system should show some products even then, so the user will get the recommendations related to the search they have done.

**CHAPTER-2**

LITERATURE SURVEY

Over the years, recommender systems have been studied widely and are divided into different categories according to the approach being used. The categories are collaborative filtering (CF), content based and context based. Collaborative filtering (CF) uses the numerical reviews given by the user and is mainly based upon the historical data of the user available to the system . The historical data available helps to build the user profile and the data available about the item is used to make the item profile. Both the user profile and the item profile are used to make a recommendation system. The Netflix Competition has given much popularity to collaborative filtering. Collaborative filtering is considered the most basic and the easiest method to find recommendations and make predictions regarding the sales of a product. It does have some disadvantages which has led to the development of new methods and techniques. Content Based Recommender System Content based systems focus on the features of the products and aim at creating a user profile depending on the previous reviews and also a profile of the item in accordance with the features it provides and the reviews it has received .It is observed that reviews usually contain product feature and user opinion in pairs . It is observed that users’ reviews contain a feature of the product followed by his/her opinion about the product. Content based recommendation systems help overcome sparsity problem that is faced in collaborative filtering based recommendation system. Context Based Recommender System Extending the user/item convention to the circumstances of the user to incorporate the contextual information is what is achieved in context-based recommender systems . This helps to abandon the cumbersome process of making the user fill a huge number of personal details.

2.1 EXISTING SYSTEM

Recommender systems are used quite extensively these days. As a result of this, this topic has caught the attention of many researchers over the past couple of years. The first recommender systems dates back to 1998. This focused on using implicit feedback in place of explicit ratings given by individual users. The 2003 industry report on amazon.com recommendations by Greg Linden, Brent Smith and Jeremy York from Amazon.com focuses on the algorithms the shopping website employed in order to recommend products to their customers. This covers collaborative filtering and clustering models. However, this gives rise to a number of irregularities and inaccurate results as it assumes that the customers purchase sparse items or rate only a few selected categories. In a real world scenario this is not true. Though this model is scalable, the task of clustering and categorizing the user base is computationally difficult when the user base is constantly changing and expanding. This also explains Item to Item collaborative filtering. Here the items from the user’s shopping cart are used for recommendations. We focused here mainly on the collaborative filtering algorithm. “Recommender Systems Based on Consumer Product Reviews”. The reviews are given by the user to generate a rating for the product. It also rates the quality of the review and tries to extract information from the review to find how likely someone else is going to buy the product. It uses opinion quality metrics to get these results. We then proceeded to look at other approaches such as hybrid methods which used both collaborative filtering, user based and item based filtering to give product recommendations. This faced the drawbacks similar to collaborative filtering and thus we looked for other better approaches towards collaborative filtering.

2.2 PROPOSED SYSTEM

Here we present several different e-commerce recommender system approaches and their advantages or disadvantages. We see recommendation systems utilizing data mining techniques, clustering analysis, Dynamic Table based approach which takes into account the customer’s life cycle for his recommendations, mining user-contributed photos, and using visual and UGC for user interest mining, Fuzzy logic, portal. One of the main conclusions is that the mentioned systems aim to enhance conversion rate by providing more personalized recommendations.

It is stated that in the last year research regarding lowering computational complexity and increasing accuracy has drawn much attention. Researchers have demonstrated approaches that outperform state-of-the-art approaches in accuracy and scalability and that lowering complexity improves the performance of RS. Findings and drawbacks are being presented and conclude with proposing some areas that need further research.

**CHAPTER-3**

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 OVERALL DESCRIPTION

This SRS is an overview of the whole project scenario. This document is to present a detailed description of thecourse management system. It will explain thepurpose and features of the system, the interfaces of the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both stakeholders and developers of the system.

3.2 Operating Environment

**Software Requirements**

Front End : Streamlit

Database : XML File

Tools : Jupiter notebook

**Hardware Requirements**

Processor : Intel Pentium® Dual Core Processor (Min)

Speed : 2.9 GHz (Min)

RAM : 4 GB (Min)

Hard Disk : 256 GB (Min)

3.3 Functional Requirements

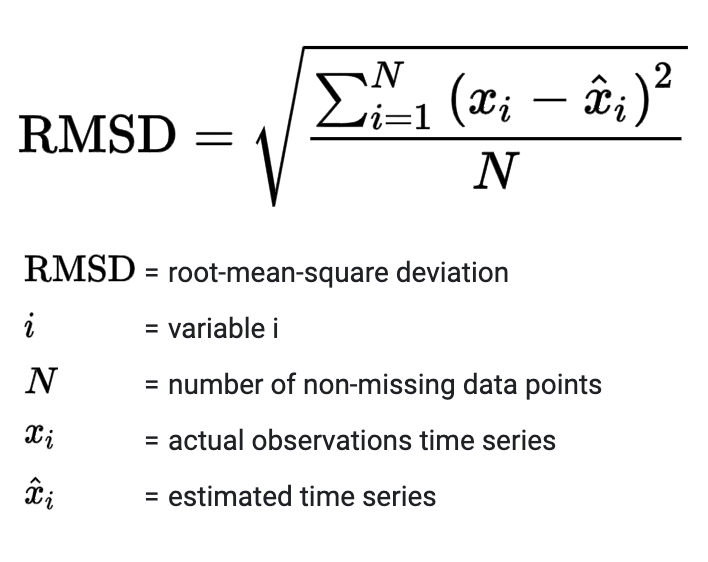
* The Application should provide the relevant recommendations based on User Activity.
* The Recommendation should be more accurate than the current existing systems.

3.4 NON-FUNCTIONAL Requirements

* Data security and scalability.
* Extensibility for future conditions.
* There should have no problem for handling larger data files.

# Evaluation Metrics

**Root Mean Square Error** (**RMSE**) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; **RMSE** is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.



* + 1. **Performance Requirements**

The performance requirements refer to static numerical requirements placed on the interaction between the users and the software.

Response Time

Average response time shall be less than 2 sec.

Recovery Time

In case of system failure, redundant system shall resume operations within 30 secs. Average repair time shall be less than 1 hr.

Start-Up/Shutdown Time

The system shall be operational within 1 minute of starting up.

Capacity

The system accommodates 4000 Concurrent Users.

Utilization of Resources

The system shall store in the database no more than one million transactions. If the database grows over this limit, old transaction shall be backed up and deleted from the operational database.

Security Requirements

The system will be made secure by assigning all users with separate registered IDs i.e. each user will be responsible for his/her assigned id.

* + 1. **Software Quality Attributes**

Reliability:

The system shall be reliable i.e. in case the server crashes, a backup server will be there to work which will be maintained continuously.

Security:

The system will be made secure by assigning all users with separate registered IDs i.e. each user will be responsible for his/her assigned ID.

Maintainability:

The CMD shall be designed in such a way that it will be very easy to maintain it in future. CMS is a web based system and will much depend on the web server and network. However, the web application will be designed using an object oriented approach and proper database modeling along with extensive documentation which will make it easy to troubleshoot and maintain in future.

Usability:

The interfaces of the system will be user friendly enough that every user will be able use it easily.

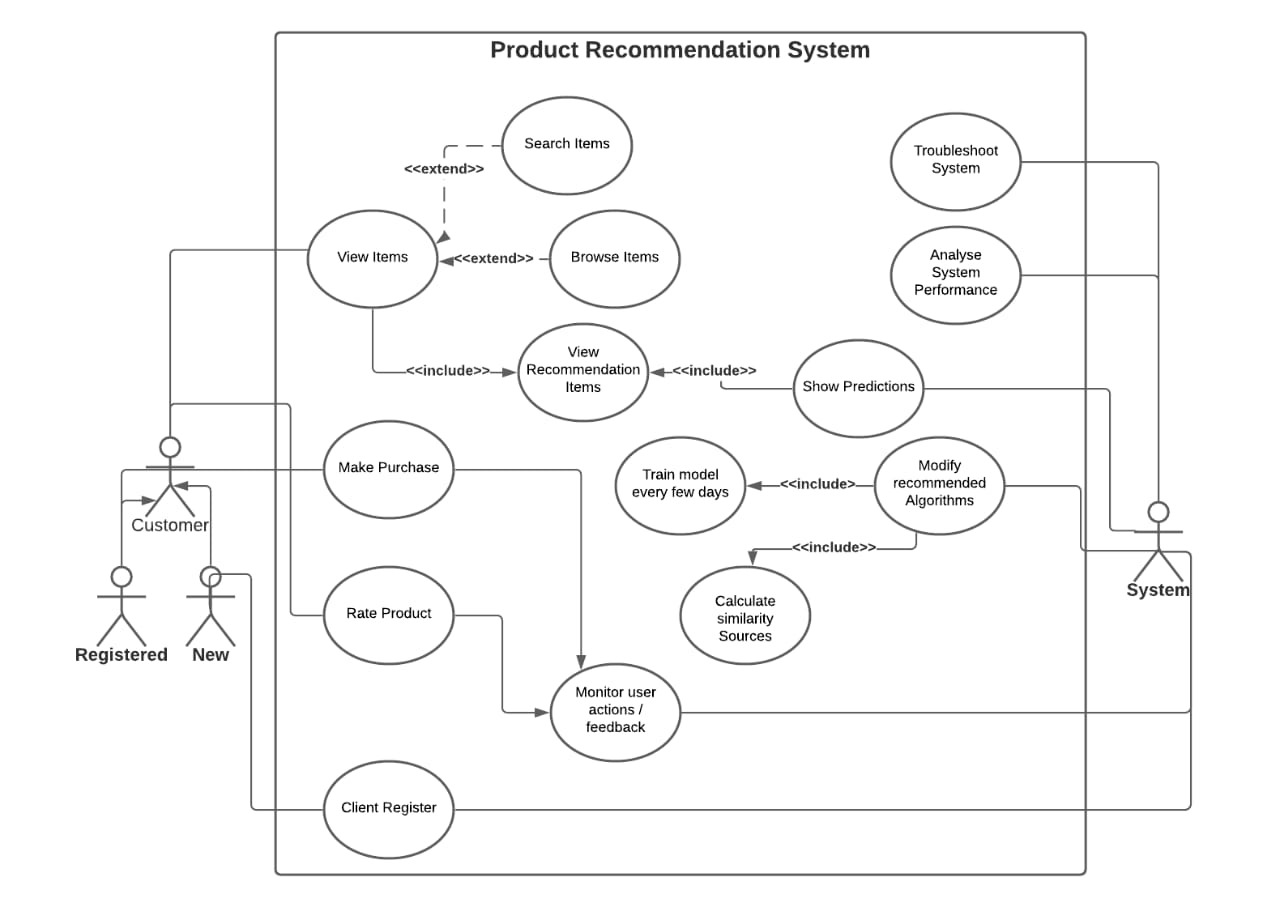
Scalability:

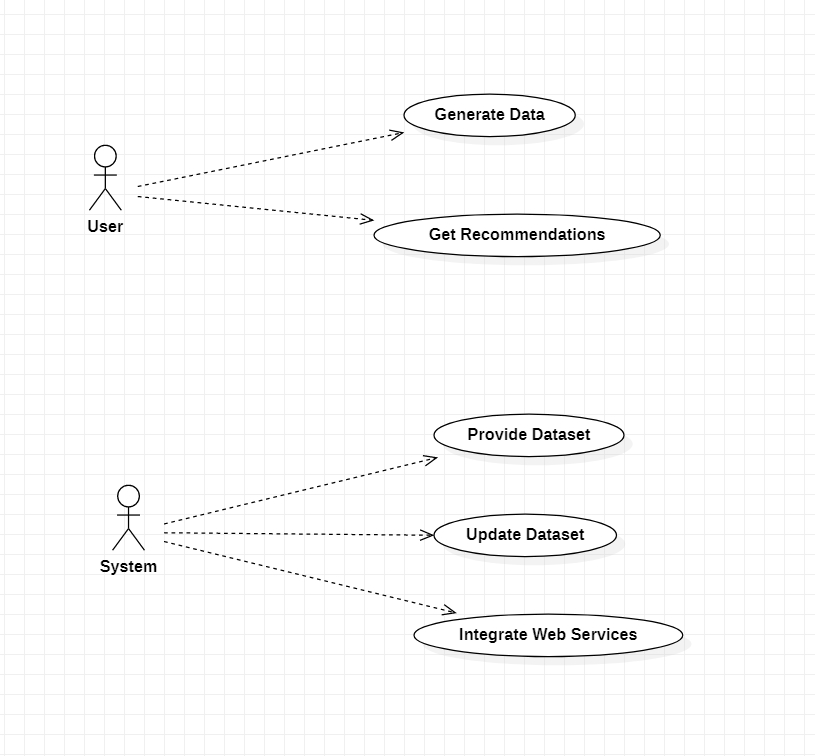
The system will be designed in such a way that it will be extendable. If more disciplines and departments are going to be added in the system, then it would easily be done i.e. if other departments want to implement the same system for them, then it will be very easy to implement.

**CHAPTER-4**

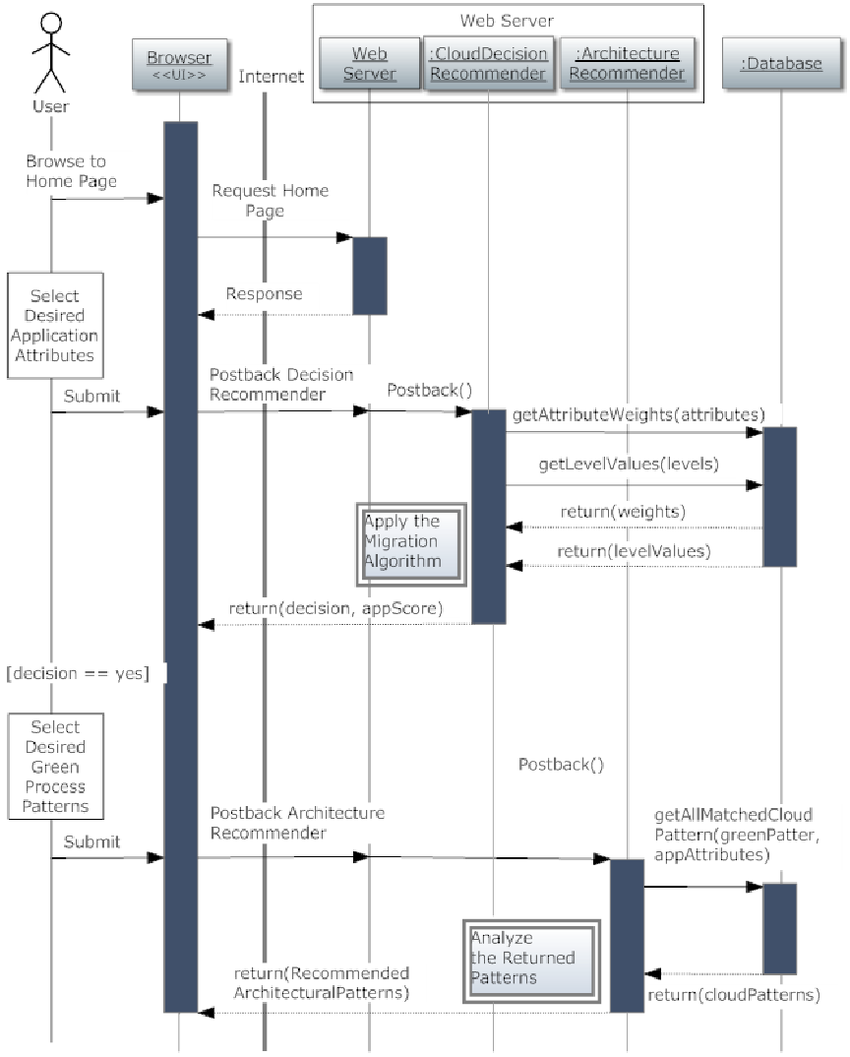
SYSTEM DESIGN

Use case Diagram:

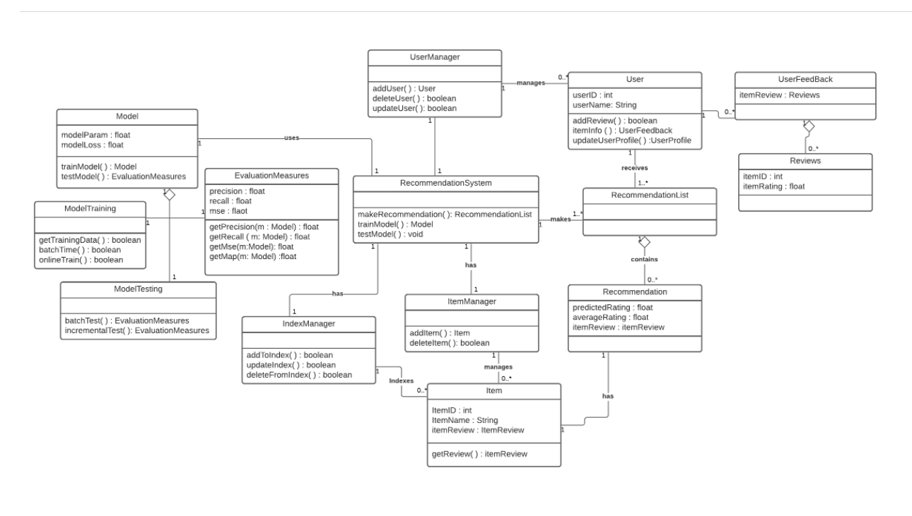


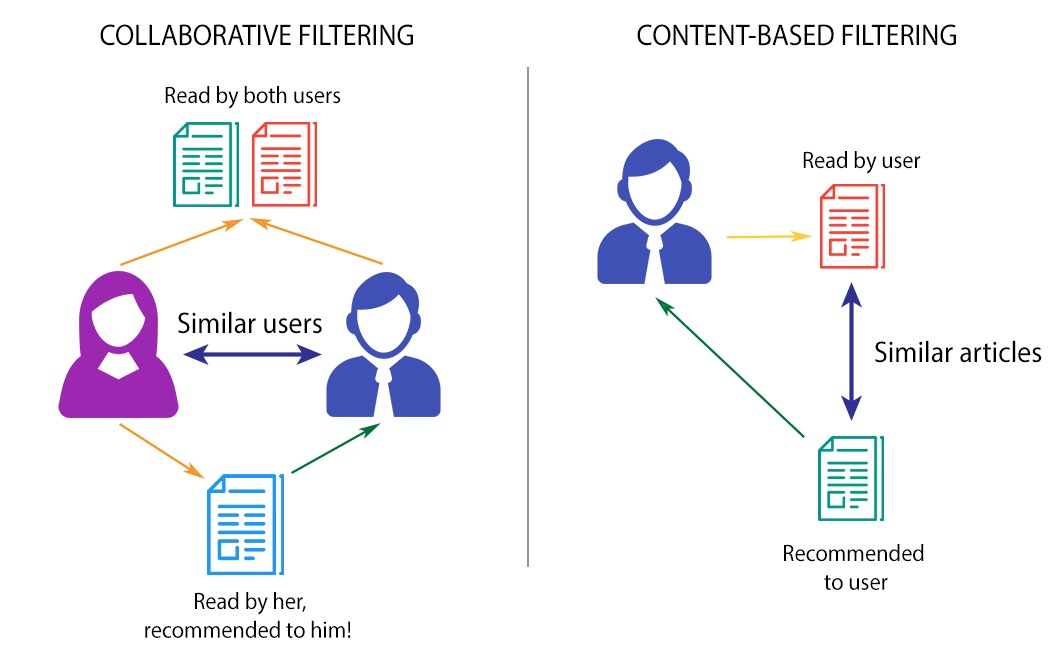
****

Sequence Diagram:



Class Diagram:





**CHAPTER-5**

Sample code

import streamlit as st

from multiapp import MultiApp

from apps import colabfilter, textcluster, topproducts # import your app modules here

from streamlit\_lottie import st\_lottie

import requests

app = MultiApp()

st.set\_page\_config(page\_title="Recommendation System",layout="wide")

st.title("Recommendation System")

def load\_lottie(url):

r=requests.get(url)

if r.status\_code!=200:

return None

return r.json()

search = load\_lottie("https://assets4.lottiefiles.com/packages/lf20\_bd97kkxh.json")

filter = load\_lottie("https://assets7.lottiefiles.com/packages/lf20\_lqge6px5.json")

topProduct = load\_lottie("https://assets7.lottiefiles.com/packages/lf20\_ao823ilv.json")

# Add all your application here

app.add\_app("Top Products", topproducts.app)

app.add\_app("Textual Clustring", textcluster .app)

app.add\_app("Colabrative Filtering", colabfilter.app)

# The main app

app.run()

import streamlit as st

class MultiApp:

def \_init\_(self):

self.apps = []

def add\_app(self, title, func):

self.apps.append({

"title": title,

"function": func

})

def run(self):

# app = st.sidebar.radio(

app = st.selectbox(

'Navigation',

self.apps,

format\_func=lambda app: app['title'])

app['function']()

import streamlit as st

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from streamlit\_lottie import st\_lottie

import requests

def load\_lottie(url):

r=requests.get(url)

if r.status\_code!=200:

return None

return r.json()

# %matplotlib inline

plt.style.use("ggplot")

import sklearn

from sklearn.decomposition import TruncatedSVD

from sklearn.feature\_extraction.text import TfidfVectorizer, CountVectorizer

from sklearn.neighbors import NearestNeighbors

from sklearn.cluster import KMeans

from sklearn.metrics import adjusted\_rand\_score

def app():

amazon\_ratings = pd.read\_csv('/Users/sainathyalala/Documents/projects/ratings\_Beauty.csv')

amazon\_ratings = amazon\_ratings.dropna()

amazon\_ratings1 = amazon\_ratings.head(10000)

ratings\_utility\_matrix = amazon\_ratings1.pivot\_table(values='Rating', index='UserId', columns='ProductId', fill\_value=0)

# ratings\_utility\_matrix.head()

# ratings\_utility\_matrix.shape

X = ratings\_utility\_matrix.T

# X.head()

# X.shape

X1 = X

SVD = TruncatedSVD(n\_components=10)

decomposed\_matrix = SVD.fit\_transform(X)

# # decomposed\_matrix.shape

correlation\_matrix = np.corrcoef(decomposed\_matrix)

# correlation\_matrix.shape

product\_names = list(X.index)

# score = metrics.accuracy\_score(y\_test,k\_means.predict(X\_test))

# product\_name=pickle.load(open('/Users/sainathyalala/Documents/projects/colab\_filtering','rb'))

def SimilarUsers(i):

product\_ID = product\_names.index(i)

# product\_ID

correlation\_product\_ID = correlation\_matrix[product\_ID]

# correlation\_product\_ID.shape

Recommend = list(X.index[correlation\_product\_ID > 0.90])

# Removes the item already bought by the customer

Recommend.remove(i)

# st.write(Recommend[0:9])

for i in Recommend[0:9]:

st.write(i)

st.subheader("Collaborative Filtering")

st.write(""" Recommend items to users based on purchase history

and similarity of ratings provided by other users who bought

items to that of a particular customer. """)

filter = load\_lottie("https://assets7.lottiefiles.com/packages/lf20\_lqge6px5.json")

with st.container():

left\_column, right\_column =st.columns(2)

with left\_column:

product\_names = list(X.index)

i = st.text\_input(" ")

if i in product\_names:

SimilarUsers(i)

else:

SimilarUsers("6117036094")

with right\_column:

st\_lottie(filter,height="500",key="filter")

st.empty()

import streamlit as st

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from streamlit\_lottie import st\_lottie

import requests

def load\_lottie(url):

r=requests.get(url)

if r.status\_code!=200:

return None

return r.json()

# %matplotlib inline

plt.style.use("ggplot")

import sklearn

from sklearn.decomposition import TruncatedSVD

from sklearn.feature\_extraction.text import TfidfVectorizer, CountVectorizer

from sklearn.neighbors import NearestNeighbors

from sklearn.cluster import KMeans

from sklearn.metrics import adjusted\_rand\_score

from sklearn import metrics

def app():

product\_descriptions = pd.read\_csv('/Users/sainathyalala/Documents/projects/product\_descriptions.csv')

product\_descriptions = product\_descriptions.dropna()

product\_descriptions1 = product\_descriptions.head(500)

vectorizer = TfidfVectorizer(stop\_words='english')

X1 = vectorizer.fit\_transform(product\_descriptions1["product\_description"])

X=X1

kmeans = KMeans(n\_clusters = 10, init = 'k-means++')

y\_kmeans = kmeans.fit\_predict(X)

true\_k = 10

model = KMeans(n\_clusters=true\_k, init='k-means++', max\_iter=100, n\_init=1)

model.fit(X1)

# print("Top terms per cluster:")

order\_centroids = model.cluster\_centers\_.argsort()[:, ::-1]

terms = vectorizer.get\_feature\_names()

def print\_cluster(i):

for ind in order\_centroids[i, :10]:

st.write(' %s' % terms[ind])

def show\_recommendations(product):

Y = vectorizer.transform([product])

prediction = model.predict(Y)

#print(prediction)

print\_cluster(prediction[0])

st.subheader("Textual Clustering")

st.write(""" For a business without any user-item purchase history,

a search engine based recommendation system can be designed

for users. The product recommendations can be based on textual

clustering analysis given in product description. """)

search = load\_lottie("https://assets4.lottiefiles.com/packages/lf20\_bd97kkxh.json")

with st.container():

left\_column,right\_column = st.columns(2);

with left\_column:

name=st.text\_input("")

show\_recommendations(name)

with right\_column:

st\_lottie(search,height=500,key="online shoping",)

st.empty()

import streamlit as st

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import pickle

from streamlit\_lottie import st\_lottie

import requests

def load\_lottie(url):

r=requests.get(url)

if r.status\_code!=200:

return None

return r.json()

# %matplotlib inline

plt.style.use("ggplot")

import sklearn

from sklearn.decomposition import TruncatedSVD

def app():

amazon\_ratings = pd.read\_csv('/Users/sainathyalala/Documents/projects/ratings\_Beauty.csv')

amazon\_ratings = amazon\_ratings.dropna()

popular\_products = pd.DataFrame(amazon\_ratings.groupby('ProductId')['Rating'].count())

most\_popular = popular\_products.sort\_values('Rating', ascending=False)

popular\_products = pd.DataFrame(most\_popular.head(10))

topProduct = load\_lottie("https://assets7.lottiefiles.com/packages/lf20\_ao823ilv.json")

with st.container():

st.subheader("Top Products ")

st.write("""Popularity based are a great strategy to target the new customers with the most

popular products sold on a business's website and is very useful to

cold start a recommendation engine.""")

left\_column,right\_column = st.columns(2)

with left\_column:

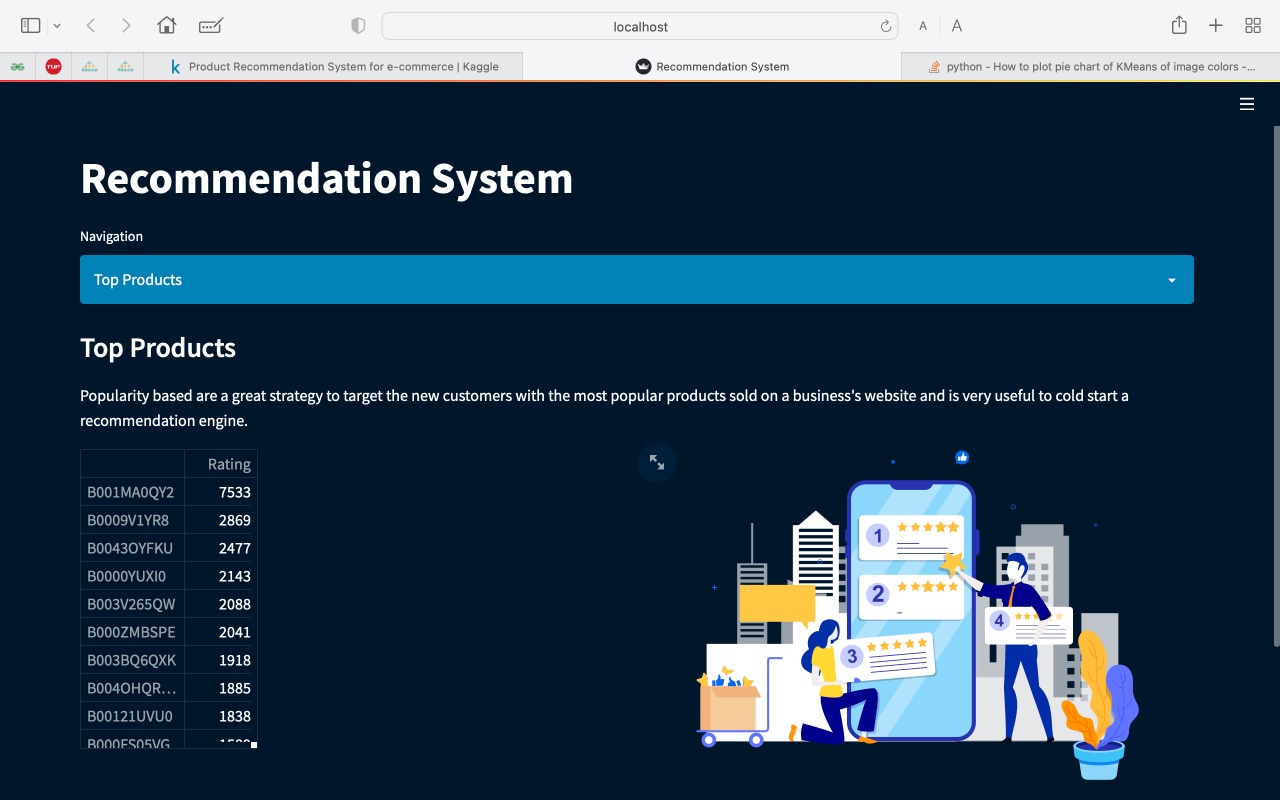
st.write(popular\_products)

with right\_column:

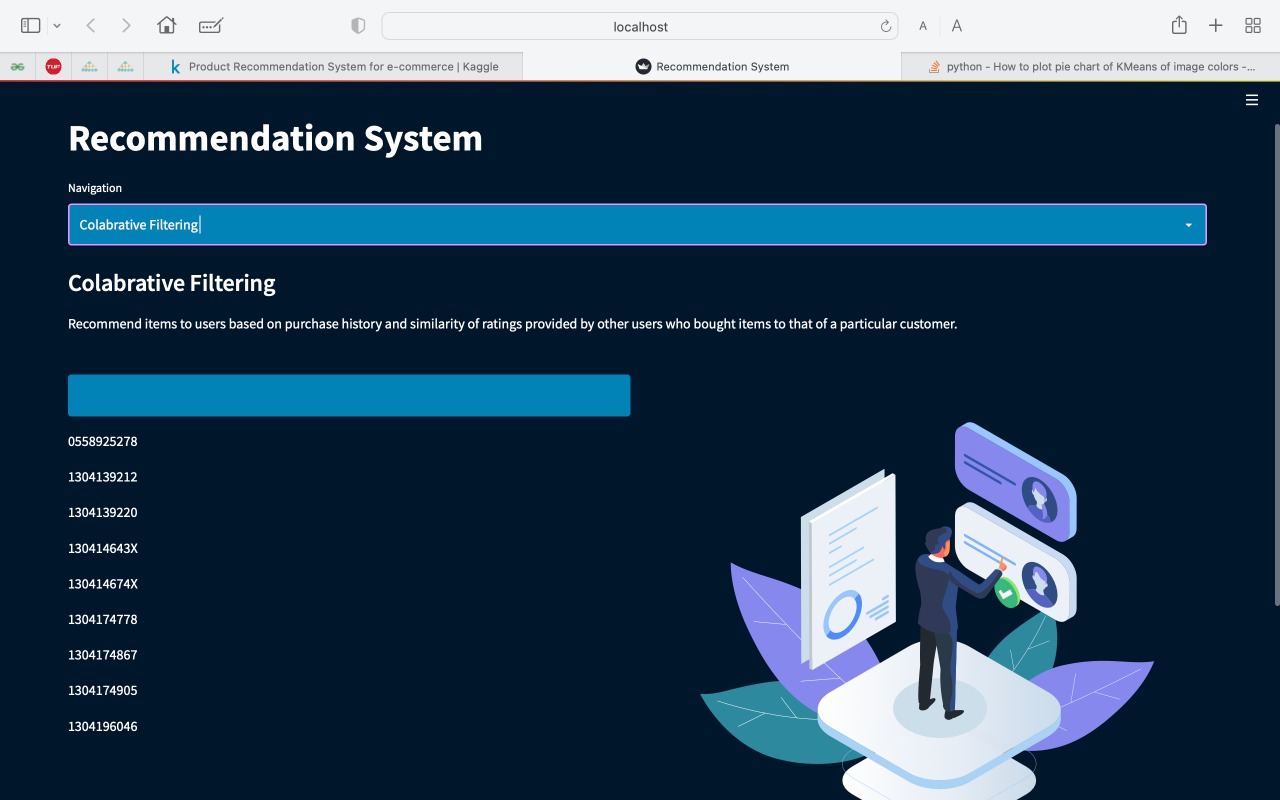
st\_lottie(topProduct,height=350,key="topProduct")

**CHAPTER-7**

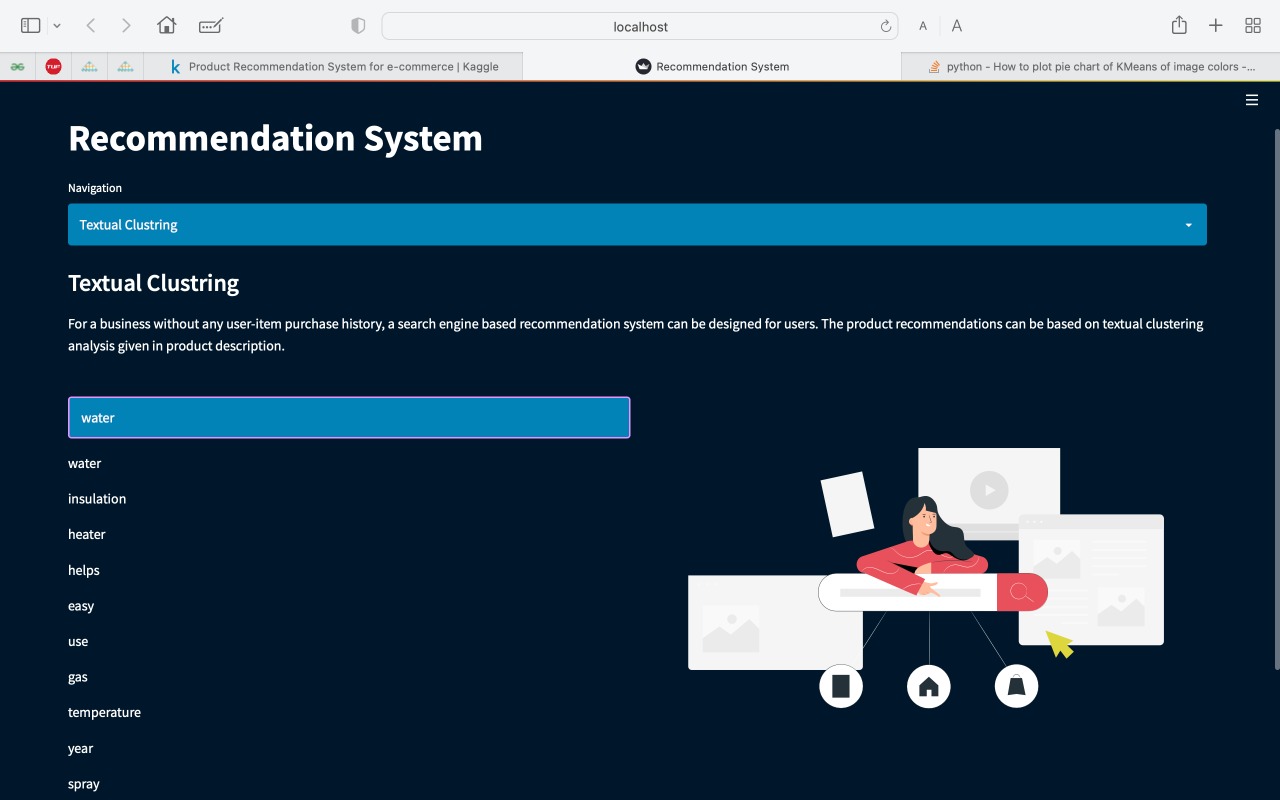
SCREENSHOTS



Recommendations based on popularity



Recommendation based on collabrative filtering



Recommendation based on Textual Clustering

**CHAPTER-8**

CONCLUSION & REFERENCES

Conclusion and future work:

Recommender structures are proving to be a useful device for addressing a part of the records overload phenomenon from the internet. Its evolution has followed the evolution of the internet. The primary technology of recommender system used conventional web sites to gather information from the following sources: (a) content material-primarily based records (b) demographic statistics, and (c) memory-primarily based information. Latest research shows the use of Sentimental Analysis in developing of more accurate recommender system. These types of methods are commonly used in e-commerce business. In this paper we have classify various approaches of recommender system that are based on Sentiemental analysis. Future research will deal with advancing the prevailingtechniques and algorithms to enhance the nice of recommenderstructures predictions and hints.

E-commerce is not only a trend nowadays but also a necessityin the current times which is expanding day by day i.e. the number and variety of products increases on a daily basis as new customers and sellers join the network to get some profit and follow the trend. Not only customers are attracted due to the large variety, sellers also anticipate profit or growth in business and launch their products as well, as a result, the same product is sold by different sellers which increases the competition among different sellers. When there is a competition and you can’t do anything like reducing the price etc so how to attract the customers to buy their product? So, the idea behind doing this is to make an equally likely platform for all the sellers and customers too. So, for this we can do multiple things and this is the future work which will make an E Commerce much better.

REFERENCES:

* Article by Becky Thames, Dated: August 25, 2020. <http://beckythames.com/motivation-for-e-commerce/>
* Article by Josh Reyes, Dated: Mar 31, 2018.

[https://medium.com/ecommerce-marketing-blog/7-types-of-product- recommendation-engines-your-e-commerce-store-needs-468627157d0](https://medium.com/ecommerce-marketing-blog/7-types-of-product-%20recommendation-engines-your-e-commerce-store-needs-468627157d0)

* Harsh Khatter, Anil K Ahlawat, “An intelligent personalize web blog searching technique using fuzzy-based feedback recurrent neural network”. Soft Comput 24, 9321–9333 (2020).
* Zhijun Zhang, Gongwen Xu, Pengfei Zhang, "Research on E- Commerce Platform-Based Personalized Recommendation Algorithm", Applied Computational Intelligence and Soft Computing, vol. 2016, pp. 1 – 6.
* Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R.Kashef, “Recommendation Systems: Algorithms, Challenges,Metrics, and Business Opportunities,” Applied Sciences, vol. 10, no. 21, p. 7748, Nov. 2020.
* R. M. Bell, Y. Koren and C. Volinsky, "The BellKor  solution to the Netflix grand prize," March 2012. [http://www2.research.att.com/~volinsky/netflix/Progress](http://www2.research.att.com/~volinsky/netflix/Progress%20)