Personalized medical recommendation system based on Sentiment Analysis and Hybrid Matrix Factorization

A project report submitted for the partial fulfillment of the Bachelor of Technology Degree in Computer science & Engineering under Maulana Abul Kalam Azad University of Technology

BY

Abhishek Dikshit

(ROLL NO : 10400214048, REGISTRATION NO : 141040110209)

&

Kumar Avishkar

(ROLL NO :10400214048, REGISTRATION NO:141040110254)

Under the Guidance of:

Prof. Dr. Avijit Bose
Department of Information Technology
For the Academic Year 2014-2018

IEM
श्रद्धावान लभते ज्ञानम्
INSTITUTE OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Institute of Engineering & Management
Y-12, Salt Lake, Sector-V, Kolkata-700091

Affiliated To:

MAKAUT WB
Utech
Maulana Abul Kalam Azad University of Technology, West Bengal
formerly known as West Bengal University of Technology
In Pursuit of Knowledge and Excellence

Maulana Abul Kalam Azad University of Technology
BF-142, Salt Lake, Sector I, Kolkata-700064

ACKNOWLEDGEMENT

We should like to take this opportunity to extend our gratitude to the following revered persons without whose immense support, completion of this project wouldn’t have been possible.

We are sincerely grateful to our advisor and mentor **Avijit Bose** of the **Information Tecchnology** IEM Kolkata, for his constant support , significant insights and for generating in us a profound interest for this subject that kept us motivated during the entire duration of this project.

We would also like to express our sincere gratitude to **Prof. Dr. Satyajit Chakrabarti** (Director**,**IEM)**, Prof. Dr. Amlan Kusum Nayak (**Principal,IEM) and **Prof. Dr Mohuya Chakraborty**, HOD of **Information Technology** and other faculties of Institute of Engineering & Management, for their assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Abhishek Dikshit
Reg. No:141040110209
Dept. of Information Technology
Institute of Engineering & Management, Kolkata

Kumar Avishkar
Reg .No:10400214048
Dept. of Information Technology
Institute of Engineering & Management, Kolkata

श्रद्धावान लभते ज्ञानम

INSTITUTE
OF ENGINEERING & MANAGEMENT
Salt Lake Electronics Complex, Kolkata - 700 091, WB, INDIA

Phone : (033) 2357-2969/2059/2995
(033) 2357-8189/8908/5389
Fax : 91-33-23578302
E-mail : director@iemcal.com
Website : www.iemcal.com

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the project report entitled "**Personalized medical recommendation system based on Sentiment Analysis and Hybrid Matrix Factorization** ", submitted by

1.Abhishek Dikshit

(*Registration No.* 141040110209 ***of 2014-2015 Roll no.*** 10400214004

2. Kumar Avishkar

(*Registration No.* 141040110254 ***of 2014-15 Roll no.*** 10400214048

Students of **INSTITUTE OF ENGINEERING & MANAGEMENT,** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology in Information Technology,** is a bona fide work carried out under the super vision and guidance of **Prof. Avijit Bose** during the final year of the academic session of 2014-2018.The content of this report has not been submitted to any other University or Institute for the award of any other degree.

It is further certified that work is entirely original and its performance has been found to be quite satisfactory.

Prof . **Avijit Bose**
Project Guide
Dept. of **Information Technology**
Institute of Engineering & Management

Prof. Dr. **Mohuya Chakraborty**
H.O.D
Dept. of **Information Technology**
Institute of Engineering &Management

Prof. Dr.**A.K.Nayak**
Principal
Institute of Engineering & Management
Sector-V, Salt Lake Electronics Complex, Kolkata-700091

ABSTRACT

Nowadays, crowd-sourced review websites provide decision support for various aspects of daily life including shopping, local services, healthcare, etc. However, one of the most important challenges for existing healthcare review websites is the lack of personalized and professionalized guidelines for users to choose medical services. In this paper, we develop a novel healthcare recommendation system called Health Recommendation System, which is based on hybrid matrix factorization methods. Health Recommendation System differs from previous work in the following aspects: (1) emotional offset of user reviews can be unveiled by sentiment analysis and be utilized to revise original user ratings; (2) user preference and doctor feature are extracted by Latent Dirichlet Allocation and incorporated into conventional matrix factorization. We compare Health Recommendation System with previous healthcare recommendation methods using real datasets. The experimental results show that Health Recommendation System provides a higher prediction rating and increases the accuracy of healthcare recommendation significantly.

TABLE OF CONTENTS

TOPIC **PAGE NUMBER**

Chapter 1

Introduction

Healthcare Recommendation System: The following project aims at developing a next generation recommendation platform which is more than a general recommendation system. Its main fundamental is hybrid matrix factorization which includes sentiment analysis of user reviews for revising the original rating given by user and topic modelling for user preference and doctor selection. It will overcome the challenges that people face while choosing medical services using health related medical websites such as Vitals, Health grades, RateMDs etc. This innovative process of medical consultancy exhibits high efficiency compared to traditional onsite doctor selection. However some challenges exist to enable personalized and accurate medical services.

- **Business Domain:** Healthcare
- **Technical Domain:** Web-App, ML(Sentiment Analysis, Topic Modelling, Matrix Factorization)
- **Software Requirement:**
  - **Front End:** HTML(using EJS), CSS, JS.
  - **Back End:** Python, MongoDB ,NodeJs ,Express Js.
- **Operating System:** Windows, Linux, Mac.

1.3 GLOSSARY

Web-App	Web application
ML	Machine Learning
HTML	Hypertext Mark-up Language
CSS	Cascading Style Sheet
JS	Java Script
Mac	Macintosh

Table 1: Glossary

1.1 Motivation

We plan to develop a system which can have important practical application such as recommending the right doctor to the user based on his requirements that can be anything based on his location, specialty requirement. Data would be presented based on the star based as well the result based on the sentiment analysis of the textual reviews. This would lead to a more personalized

1.2 Objective

This project aims at solving the problem of selection of doctors online without having sufficient knowledge and specification about doctors. Online search results may contain too many doctors to meet diverse needs. It will provide user with professionalized and personalized doctor recommendation through mining user emotion and preference from user rating and reviews about doctors. Machine learning is also implemented for better filtering and classification in further advancement. It also gives the doctor a platform to analyse their performance.

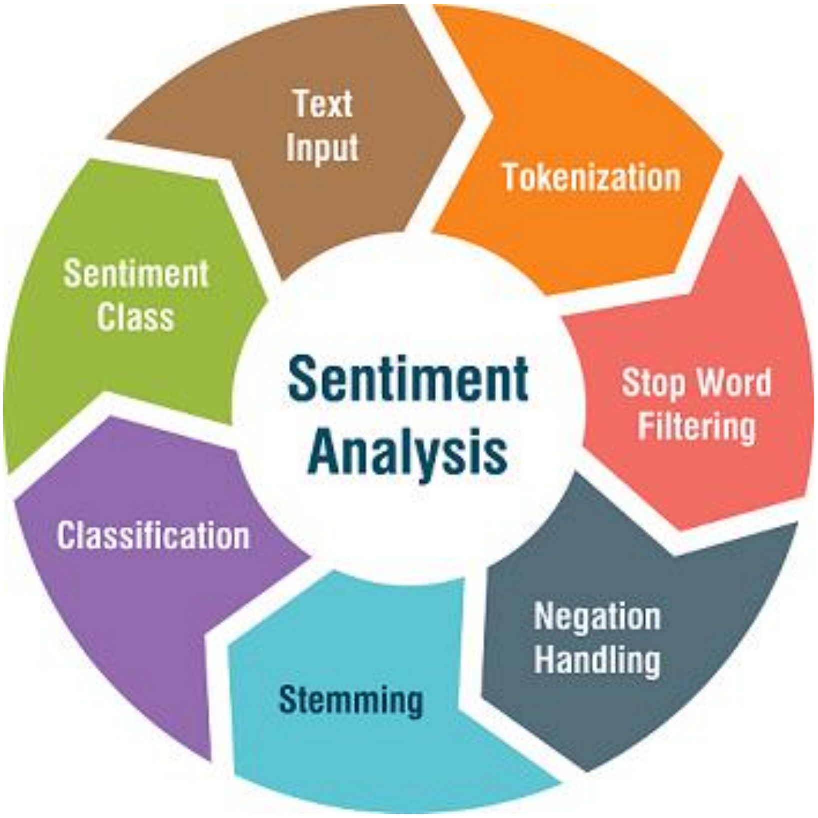
Chapter 2

Background

Sentiment Analysis

Sentiment analysis (sometimes known as **opinion mining** or **emotion AI**) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

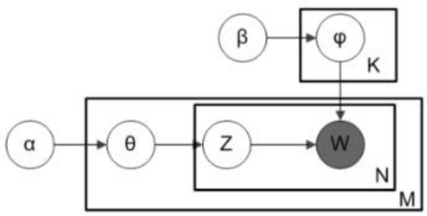
Generally speaking, emotional recognition is complex through language and facial expression whereas the development of psychology and linguistics simplifies the process of text sentiment analysis. At present, quite a few works try to integrate emotional actors into personalized recommendation [6]. In [7], Poirier et al. proposed a collaborative filtering recommendation according to sentiment analysis of user reviews instead of rating. In [8], Ko et al. proposed a hybrid recommendation algorithm based on content and collaborative filtering, in which the vector of item features and user emotion is calculated from item descriptions and user reviews.



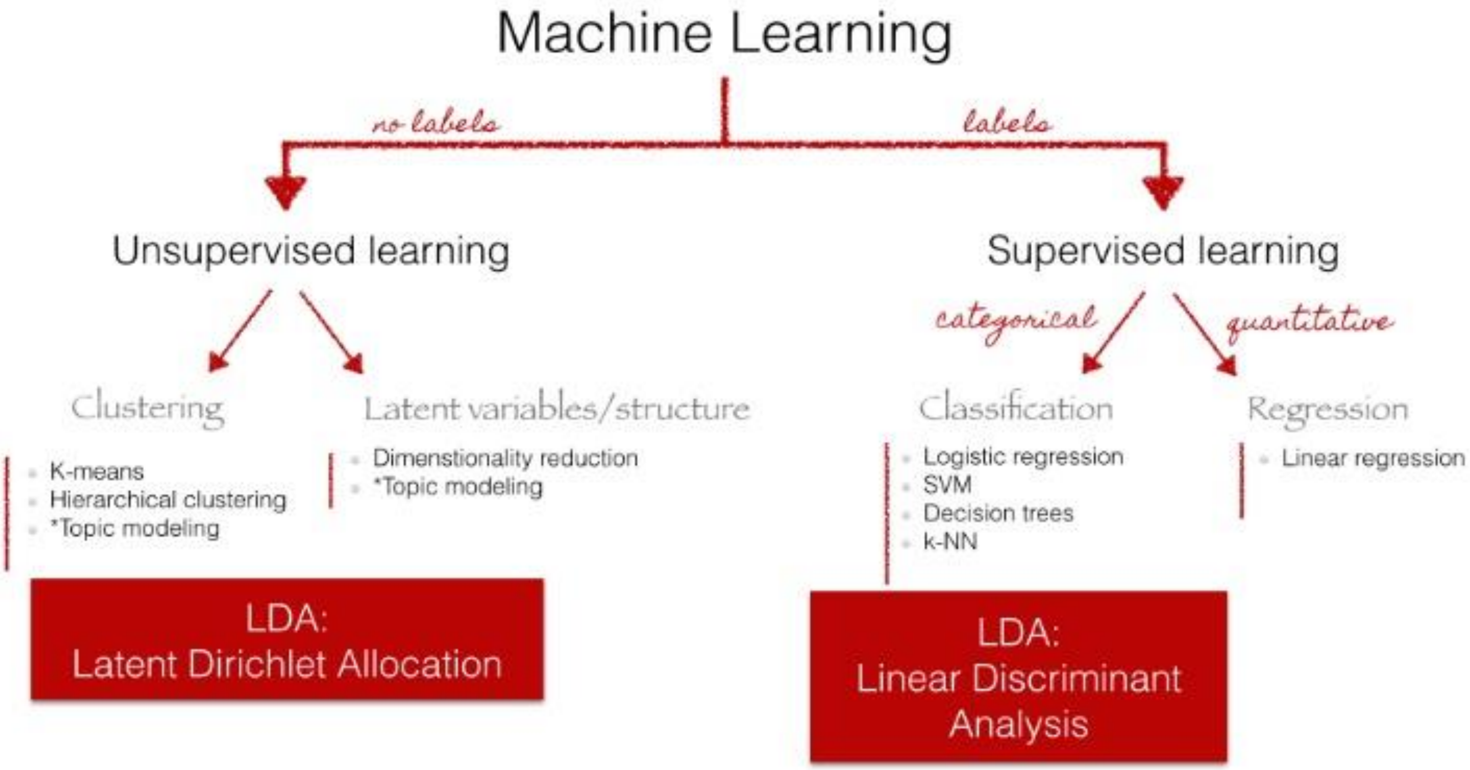
Topic Modelling

One such technique in the field of text mining is **Topic Modelling**. As the name suggests, it is a process to automatically identify topics present in a text object and to derive hidden patterns exhibited by a text corpus. It is an unsupervised approach used for finding and observing the bunch of words (called “topics”) in large clusters of texts. Topics can be defined as “a repeating pattern of co-occurring terms in a corpus”. A good topic model should result in – “health”, “doctor”, “patient”, “hospital” for a topic – Healthcare, and “farm”, “crops”, “wheat” for a topic – “Farming”. Currently, the Latent Dirichlet Allocation (LDA) topic model is widely adopted for document topic extraction. Through LDA, topics and their probability distribution can be calculated for analyzing document similarity, which is essential for document classification and personalized recommendation. Currently, there are many proposals that try to incorporate topic model into matrix factorization.

LDA Plate notation



- K is the number of topics
- N is the number of words in the document
- M is the number of documents to analyse
- α is the Dirichlet-prior concentration parameter of the per-document topic distribution
- β is the same parameter of the per-topic word distribution
- $\phi(k)$ is the word distribution for topic k
- $\theta(i)$ is the topic distribution for document i
- $z(i,j)$ is the topic assignment for $w(i,j)$
- $w(i,j)$ is the j-th word in the i-th document
- ϕ and θ are Dirichlet distributions, z and w are multinomials.



Machine Learning

no labels

labels

Unsupervised learning

Clustering

- K-means
- Hierarchical clustering
- *Topic modeling

Latent variables/structure

- Dimenstionality reduction
- *Topic modeling

LDA:
Latent Dirichlet Allocation

Supervised learning

categorical

quantitative

Classification

- Logistic regression
- SVM
- Decision trees
- k-NN

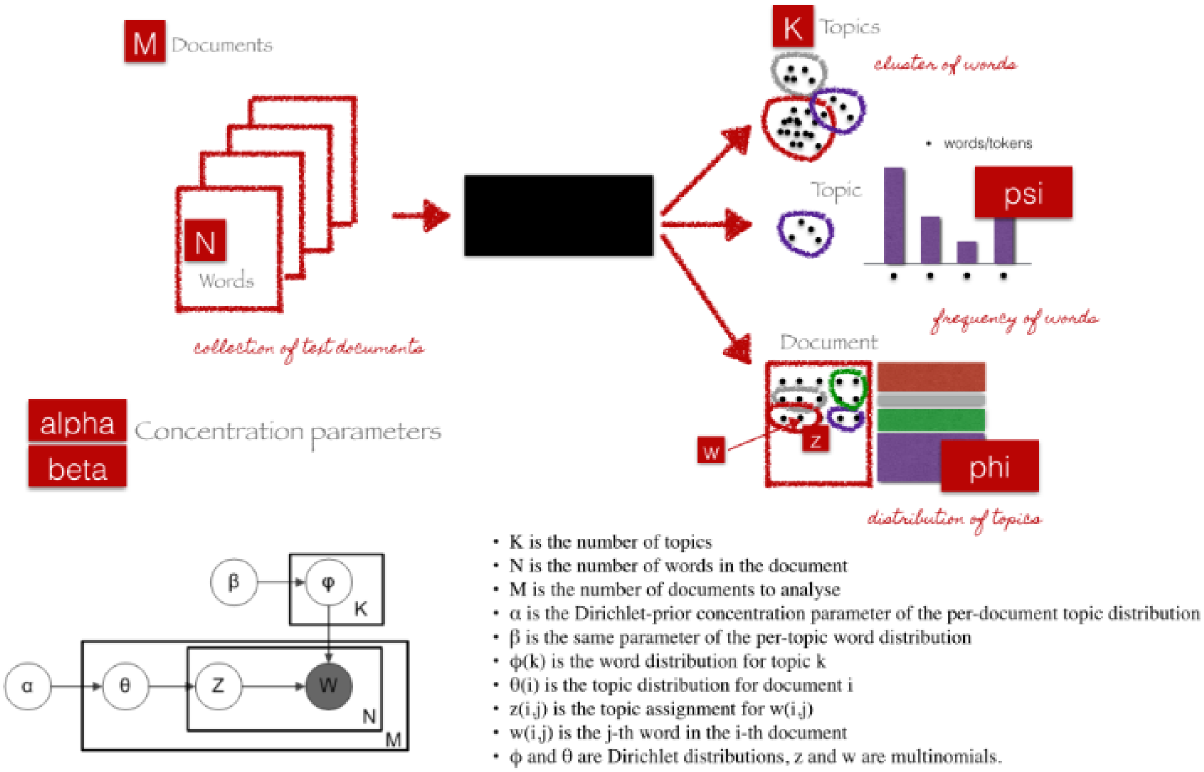
LDA:
Linear Discriminant Analysis

Regression

- Linear regression

Parameters of LDA

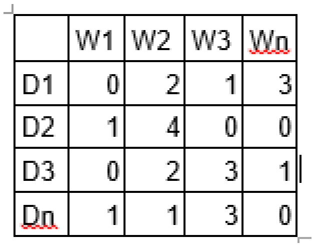
Alpha and **Beta** Hyper parameters – alpha represents document-topic density and Beta represents topic-word density. Higher the value of alpha, documents are composed of more topics and lower the value of alpha, documents contain fewer topics. On the other hand, higher the beta, topics are composed of a large number of words in the corpus, and with the lower value of beta, they are composed of few words.



- K is the number of topics
- N is the number of words in the document
- M is the number of documents to analyse
- α is the Dirichlet-prior concentration parameter of the per-document topic distribution
- β is the same parameter of the per-topic word distribution
- φ(k) is the word distribution for topic k
- θ(i) is the topic distribution for document i
- z(i,j) is the topic assignment for w(i,j)
- w(i,j) is the j-th word in the i-th document
- φ and θ are Dirichlet distributions, z and w are multinomials.

Matrix Factorization

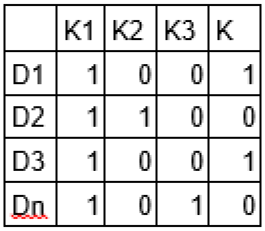
In **matrix factorization** technique vector space, any corpus (collection of documents) can be represented as a document-term matrix. The following matrix shows a corpus of N documents D1, D2, D3 … Dn and vocabulary size of M words W1,W2 .. Wn. The value of i,j cell gives the frequency count of word Wj in Document Di.



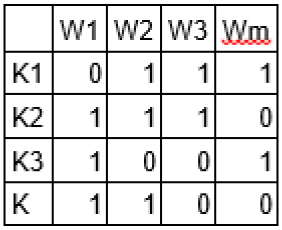
	W1	W2	W3	Wn
D1	0	2	1	3
D2	1	4	0	0
D3	0	2	3	1
Dn	1	1	3	0

Figure 1 DTM

This Document-Term Matrix into two lower dimensional matrices – M1 and M2. M1 is a document-topics matrix and M2 is a topic – terms matrix with dimensions (N, K) and (K, M) respectively, where N is the number of documents, K is the number of topics and M is the vocabulary size.



	K1	K2	K3	K
D1	1	0	0	1
D2	1	1	0	0
D3	1	0	0	1
Dn	1	0	1	0



	W1	W2	W3	Wm
K1	0	1	1	1
K2	1	1	1	0
K3	1	0	0	1
K	1	1	0	0

Figure 2 LDM-1

Figure 3 LDM-2

Chapter 3

Proposed Strategy

3.1 Overview

With the rapid development of mobile networks such as the fifth generation (5G) system [1,2], a significant amount of professional knowledge from various sectors is available to Internet users at anywhere and can be accessed anytime to provide assistance decision[3,4]. For example, we can choose different movies according to the ratings on IMDB,1 while the selection of restaurants, hotels and stores can be referred to other users'reviews on Yelp.2 Similarly, the way that people choose medical service is changing with health related reviews websites, such as Vitals,3 Healthgrades4 and RateMDs,5 etc. Through these websites, detailed information about doctors can be obtained for choosing doctor with an online appointment. This innovative process of medical consultation exhibits high efficiency compared to traditional onsite doctor selection [5]. However, several challenges exist to enable personalized and accurate medical services:

• **Personalized and professionalized demand**:

It is very common nowadays for patients to search medical service by disease, but the search results may contain too many doctors to meet diverse needs [6]. Furthermore, user experience is always the most important for the system design [7]. For example, some users would much rather find a doctor nearby, while others prefer prescription to injection as treatment. Unfortunately, at present such personalization and professionalization demand cannot be satisfied intelligently according to user and doctorFeature.

• **Emotional offset:**

Like other sectors, it is a great issue for the medical crowd-sourced reviews that rating accuracy is often interfered by users emotion [9]. For example, a doctor's rating is 4 and a review about him is presented in Table 1. It can be concluded that this doctor is not so welcomed and such rating is possibly an encouragement, which has directly influenced the objectivity and accuracy of doctors estimation. To address these challenges, this article proposes a personalized and professionalized doctor recommendation system named iDoctor, which can conduct comprehensive analysis on healthcare crowd-sourced reviews and perform text sentiment analysis, topic model, matrix factorization and other methods. Specifically, this article makes the following contributions: (1) We propose a topic model based approach to discover user preference distribution and doctor feature distribution, which are incorporated into the matrix factorization model to provide more accurate and personalized medical recommendation. (2) We propose an emotion-aware approach to identify emotional offset in user reviews via sentiment analysis, which is incorporated into the matrix factorization model to provide more objective recommendation. The remainder of this article is organized as follows. Section 2 presents related works of matrix factorization, text sentiment analysis, and topic model. In Section 3, we introduce iDoctor architecture, theories foundation, and technical details. Section 4 analyzes the performance of recommendation provided by iDoctor, and compares it with other recommendations. Finally ,we conclude this article in Section 5.

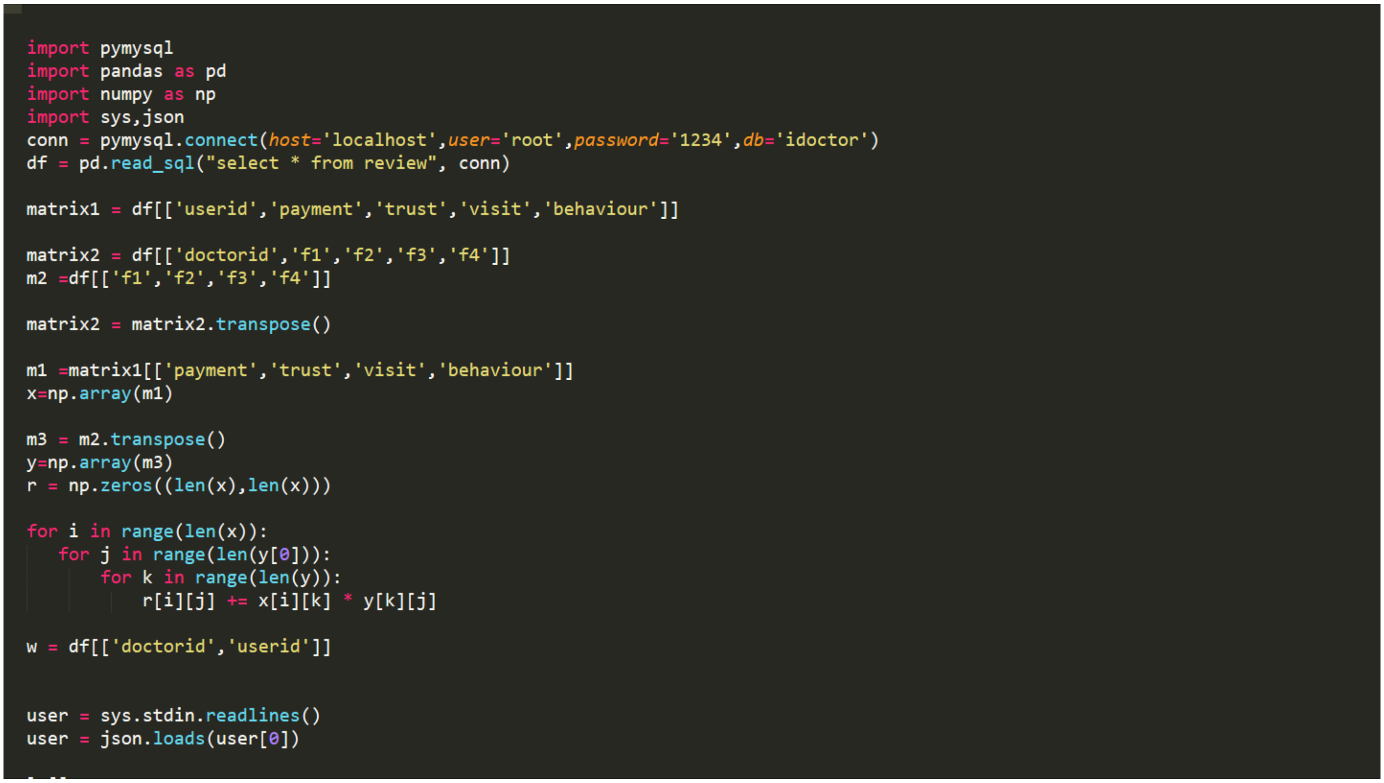
Implementation:

1. Matrix factorization

Nowadays, the recommendation based on matrix factorization proposed by Koren et al. in [10], has achieved acceptable result for rating prediction. Through this model, users and items are mapped to a low dimensional latent factor space which is the explanation to users ratings, and a user-item rating matrix is regarded as the product of user and item as presented in Eq. (1).

$$Rm\times n = Pm\times k * Qk\times n \quad (1)$$

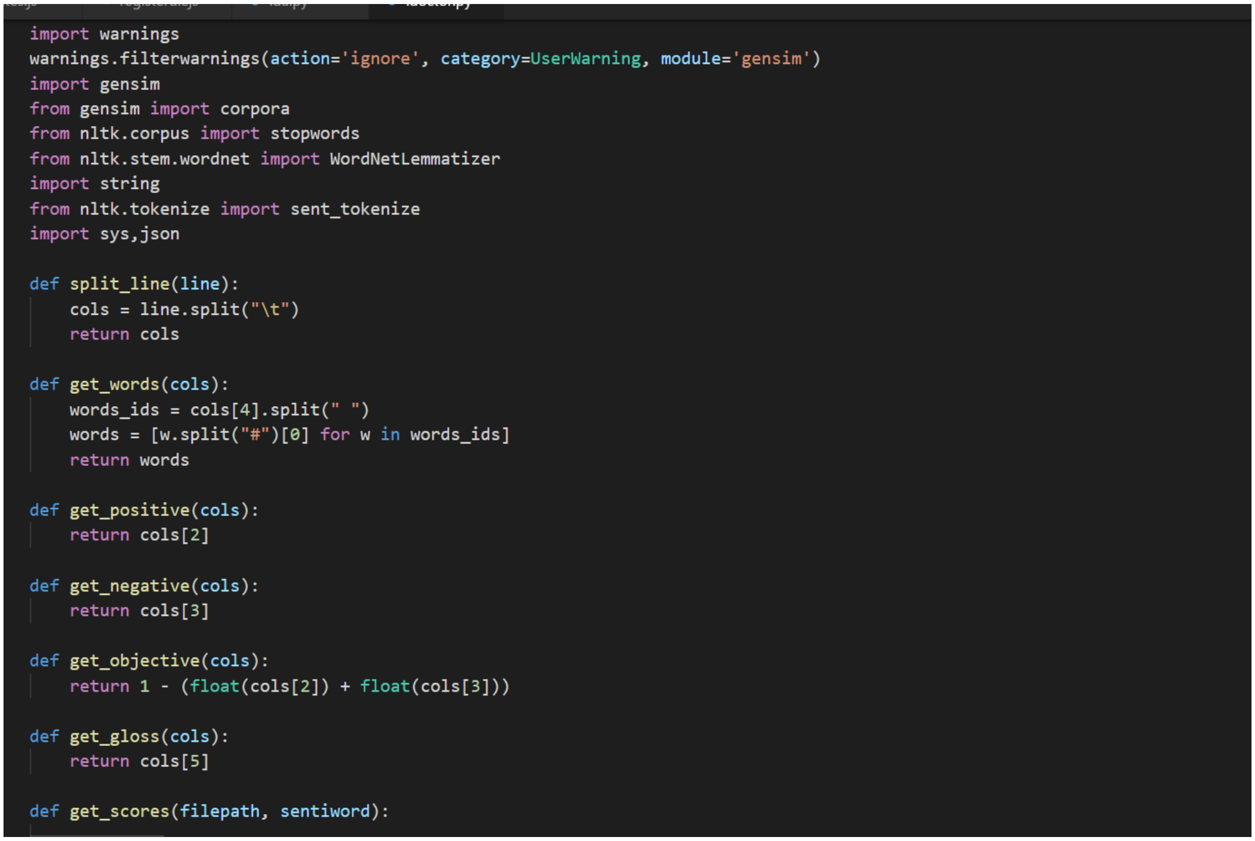
in which k represents the number of selected latent factors, P and Q represent the weights of each user and item for each characteristic in latent factor space, which are the result of rating matrix R factorization and used for rating prediction. Usually, Stochastic Gradient Descent (SGD) [11] is used for calculating P and Q.Remarkably, some latent factors can be ignored, so k < m, n. For example, category, director and actor attract more attention than duration and language, which can be ignored in matrix factorization. Because of the good performance of matrix factorization, many researchers try to extend this work. In [12], Jamali et al. proposed a matrix factorization with trust propagation for recommendation in social networks. In [13], Baltrunas et al. proposed matrix factorization based approach for context aware recommendation.



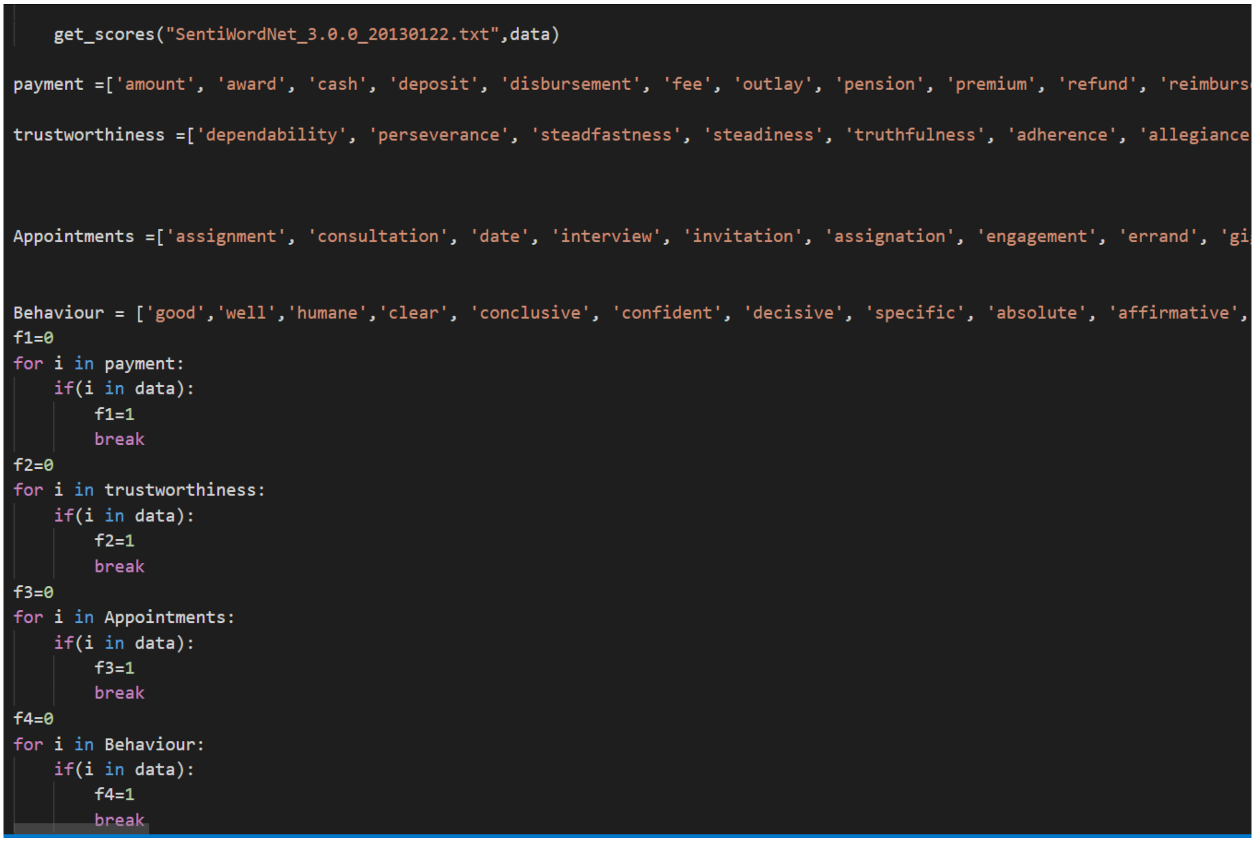
```
import pymysql
import pandas as pd
import numpy as np
import sys,json
conn = pymysql.connect(host='localhost',user='root',password='1234',db='idoctor')
df = pd.read_sql("select * from review", conn)

matrix1 = df[['userid','payment','trust','visit','behaviour']]

matrix2 = df[['doctorid','f1','f2','f3','f4']]
m2 =df[['f1','f2','f3','f4']]

matrix2 = matrix2.transpose()

m1 =matrix1[['payment','trust','visit','behaviour']]
x=np.array(m1)

m3 = m2.transpose()
y=np.array(m3)
r = np.zeros((len(x),len(x)))

for i in range(len(x)):
    for j in range(len(y[0])):
        for k in range(len(y)):
            r[i][j] += x[i][k] * y[k][j]

w = df[['doctorid','userid']]


user = sys.stdin.readlines()
user = json.loads(user[0])
```

2.2. Text sentiment analysis

User emotion plays an important role in market analysis, opinion mining and human–computer interaction, so more and more attention has been attracted to emotion recognition [14]. In general, it is complex for emotion recognition through language and facial expression, whereas the development of psychology and linguistics simplifies the process of text sentiment analysis. The text document contains not only topics but also user's emotional features that users expressions always correlate well with the emotion at that time. At present, quite a few works try to integrate emotional actors into personalized recommendation [15]. In [16], Poirier et al. proposed a collaborative filtering recommendation according to sentiment analysis of user reviews instead of rating. In [17], Ko et al. proposed a hybrid recommendation algorithm based on content and collaborative filtering, in which the vector of item features and user emotion is calculated from item descriptions and user reviews.



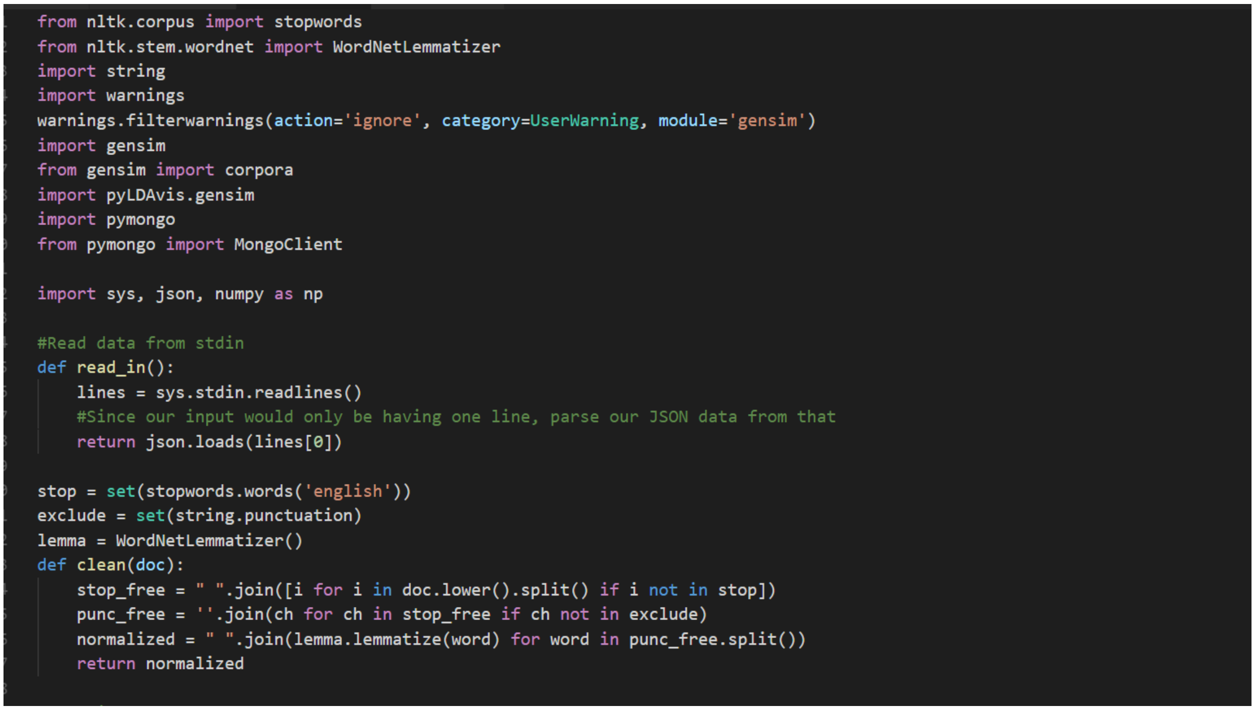
```
import warnings
warnings.filterwarnings(action='ignore', category=UserWarning, module='gensim')
import gensim
from gensim import corpora
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import string
from nltk.tokenize import sent_tokenize
import sys,json

def split_line(line):
    cols = line.split("\t")
    return cols

def get_words(cols):
    words_ids = cols[4].split(" ")
    words = [w.split("#")[0] for w in words_ids]
    return words

def get_positive(cols):
    return cols[2]

def get_negative(cols):
    return cols[3]

def get_objective(cols):
    return 1 - (float(cols[2]) + float(cols[3]))

def get_gloss(cols):
    return cols[5]

def get_scores(filepath, sentiword):
```



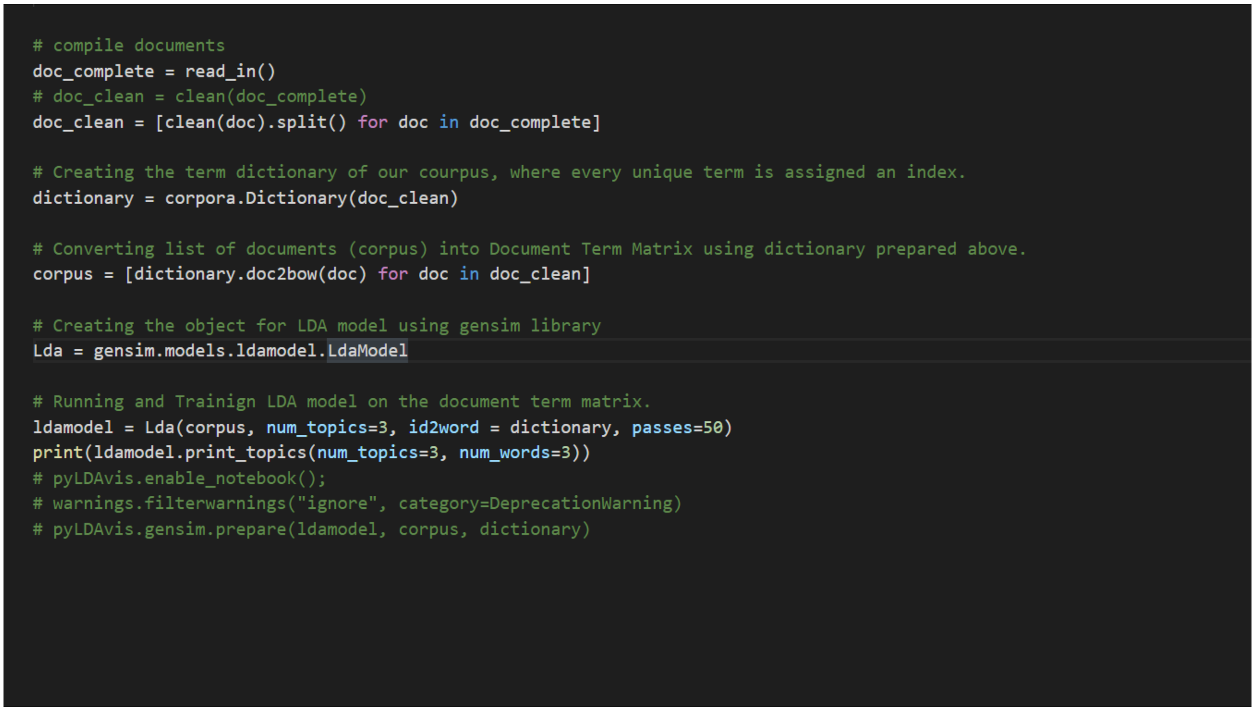
```
    get_scores("SentiWordNet_3.0.0_20130122.txt",data)

payment =['amount', 'award', 'cash', 'deposit', 'disbursement', 'fee', 'outlay', 'pension', 'premium', 'refund', 'reimburs

trustworthiness =['dependability', 'perseverance', 'steadfastness', 'steadiness', 'truthfulness', 'adherence', 'allegiance



Appointments =['assignment', 'consultation', 'date', 'interview', 'invitation', 'assignation', 'engagement', 'errand', 'gi


Behaviour = ['good','well','humane','clear', 'conclusive', 'confident', 'decisive', 'specific', 'absolute', 'affirmative',
f1=0
for i in payment:
    if(i in data):
        f1=1
        break
f2=0
for i in trustworthiness:
    if(i in data):
        f2=1
        break
f3=0
for i in Appointments:
    if(i in data):
        f3=1
        break
f4=0
for i in Behaviour:
    if(i in data):
        f4=1
        break
```

2.3. Topic model

In the topic model, a document is regarded as the mixture and combination of multiple topics, and each word in the document is generated by such a procedure that a topic is selected with certain probability from document-topic distribution and then this word is selected with certain probability from topic-word distribution.Currently, the Latent Dirichlet Allocation (LDA) topic model is widely adopted for document topic extraction. Through LDA, topics and their probability distribution can be calculated for analyzing document similarity, which is essential for document classification and personalized recommendation. Currently, there are many proposals that try to incorporate topic model into matrix factorization. In [18], Agarwal et al.represented item with some words, which are mapped to a multi-topic distribution, and provide recommendation through regression forecasting. In [19], McAuley et al. used topic model to extract item features from user reviews, integrate them with matrix factorization, and verify that the accuracy of this proposal is higher than rating matrix.



```python
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import string
import warnings
warnings.filterwarnings(action='ignore', category=UserWarning, module='gensim')
import gensim
from gensim import corpora
import pyLDAvis.gensim
import pymongo
from pymongo import MongoClient

import sys, json, numpy as np

#Read data from stdin
def read_in():
    lines = sys.stdin.readlines()
    #Since our input would only be having one line, parse our JSON data from that
    return json.loads(lines[0])

stop = set(stopwords.words('english'))
exclude = set(string.punctuation)
lemma = WordNetLemmatizer()
def clean(doc):
    stop_free = " ".join([i for i in doc.lower().split() if i not in stop])
    punc_free = ''.join(ch for ch in stop_free if ch not in exclude)
    normalized = " ".join(lemma.lemmatize(word) for word in punc_free.split())
    return normalized
```



```python
# compile documents
doc_complete = read_in()
# doc_clean = clean(doc_complete)
doc_clean = [clean(doc).split() for doc in doc_complete]

# Creating the term dictionary of our courpus, where every unique term is assigned an index.
dictionary = corpora.Dictionary(doc_clean)

# Converting list of documents (corpus) into Document Term Matrix using dictionary prepared above.
corpus = [dictionary.doc2bow(doc) for doc in doc_clean]

# Creating the object for LDA model using gensim library
Lda = gensim.models.ldamodel.LdaModel

# Running and Trainign LDA model on the document term matrix.
ldamodel = Lda(corpus, num_topics=3, id2word = dictionary, passes=50)
print(ldamodel.print_topics(num_topics=3, num_words=3))
# pyLDAvis.enable_notebook();
# warnings.filterwarnings("ignore", category=DeprecationWarning)
# pyLDAvis.gensim.prepare(ldamodel, corpus, dictionary)
```

Medical recommendation based on hybrid matrix Factorization

In this article, we propose iDoctor to provide user with professionalized and personalized doctor recommendation through mining user emotion and preference from user rating and reviews about doctors. Specifically, it includes the following modules, and the architecture is illustrated in Fig. 1:

- Sentiment analysis module, which can calculate user emotional offset from user reviews text.
- Topic modeling module, which is used to extract the distribution of user preferences and doctor features.
- Hybrid matrix factorization module, which is integrated with two feature distributions extracted by LDA for rating prediction.

Considering the emotional offset in user reviews about doctor, sentiment analysis is necessary to calculate the offset for revising the original rating. Specifically, we calculate the emotional offset through non-supervisory learning method based on sentiment lexicon in the following steps:

1. Each user review text is preprocessed, including word segmentation and morphological normalization.
2. After removing stop word and punctuation, we have a collection of words which may involve emotional offset.+
3. With sentiment lexicon SentiWordNet 3.0 [20], the emotional offset of each review can be calculated.

4. Furthermore, considering the influence of negative words, it is necessary to split each review in punctuation. In the separated Sub sentence, if the number of negatives is odd, the polarity (positive ‘‘+’’ or negative ‘‘−’’) of this sub sentence should be reversed. Obviously, the overall emotional offset is the sum of all the sub sentence.

Assume that there are i words in sub sentences sub, we can calculate the sentiment value with Eqs. (2) and (3).

$Pol(sub) =-1$ if the number of negative words in sub is odd

1 if the number of negative words in sub is even

(2)

$Sen(sub) = SentiWordNet(wi) \times Pol(sub)$(3)

Wherein, Pol(sub) represents the polarity of sub; SentiWordNet (wi) represents the sentiment value of word i calculated according to SentiWordNet 3.0.

However, there is significant difference among user review lengths, and the emotional offset of longer review is most likely higher than shorter one. In order to balance the influences that the different length of review exert over overall variance,normalization is essential to calculate the overall emotional offset of user review.

Assume that there are j subsentences in review re including n sentiment words, we can calculate the emotional offset with Eq. (4).

Offset(re) = subj∈r Sen(subj)n. (4)

Obviously, the emotional offset of each user review falls in the range of [−1, 1], which is used to revise the original rating.

3.2. Topic modeling for user preference

Generally, user reviews contain detailed personalized evaluation about doctors, such as their feelings towards medical environment, doctor's ability and attitude. On the one hand, every user preference can be discovered from these details, which is the basis for personalized recommendation. On the other hand, Table 2 describes that doctor features can be summarized from different user's reviews, such as specialty, fee range and prescribing habits, which also may be the basis of doctor selection. Therefore, LDA model is adopted by iDoctor to extract the topics of user latent preference and doctor features can be extracted from user review comments on doctors, which are involved in matrix factorization for providing more accurate and personalized recommendation.

3.3. Hybrid matrix factorization for personalized and professionalized doctor recommendation

Based on the emotional offset calculated in Section 3.1, the original rating is expected to be revised with Eq. (5).

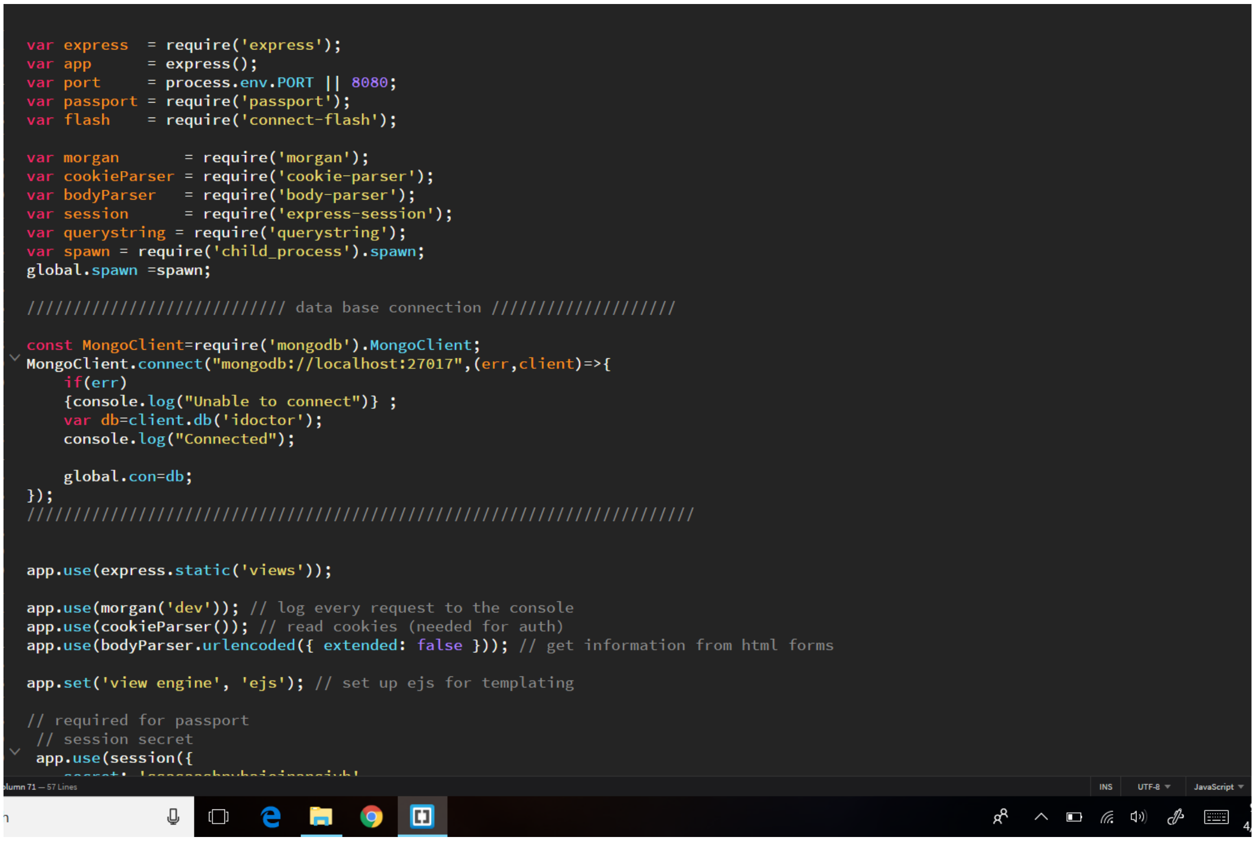
RSij = ρRij + (1 − ρ)Sij (5)

in which RSij represents the revised rating on doctor j by user i, Rij represent the original rating, Sui represents the emotional offset and rho is used for adjusting the weight between original rating and emotional offset. With Eq. (5), user emotional factor is involved in the matric factorization through the revised user–doctor rating matrix.

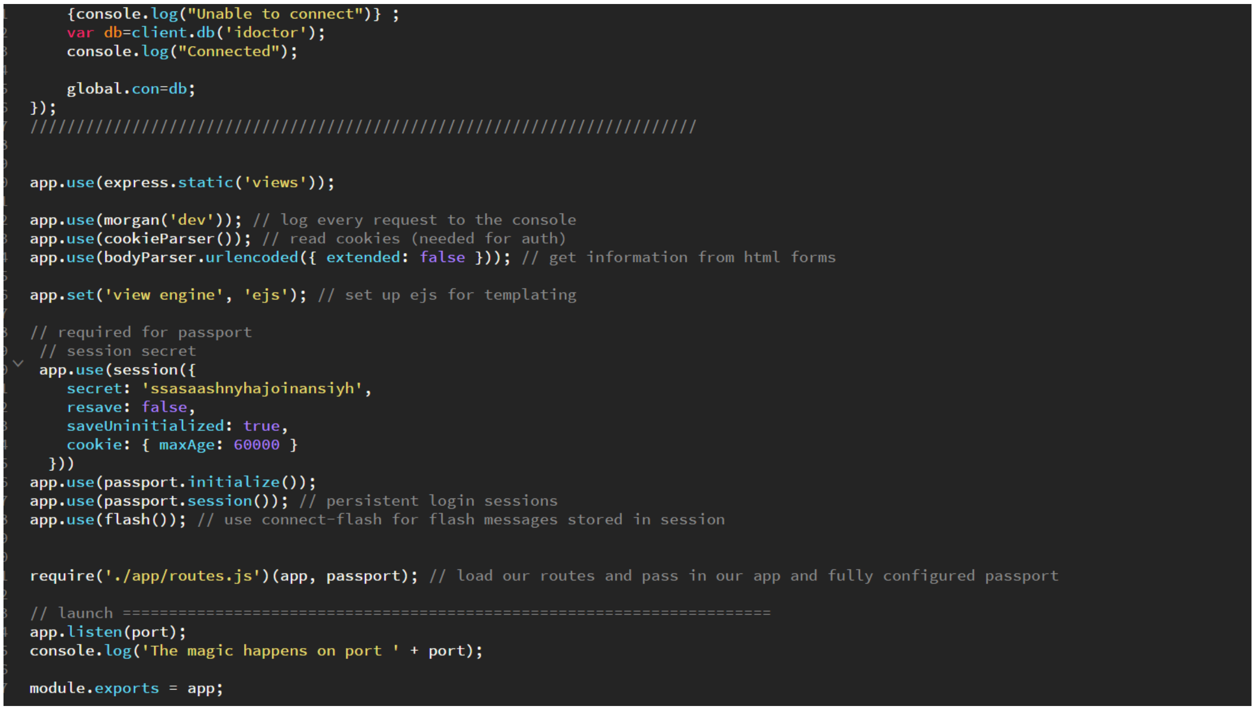
In Section 3.2, the preference topic distribution of user i,namely, $Ui = (x1, x2, . . . , xk)$, and the features topic distribution of doctor j, namely, $Ij = (y1, y2, . . . , yk)$, are calculated through LDA. In the above expressions, k represents the number of latent topics. In the conventional matrix factorization, which is named Basic Matrix Factorization (BMF) in this article, the factorized matrixes cannot represent personalization. Hence, we propose Hybrid Matrix Factorization (HMF) involving user preference and doctor feature for more personalized recommendation, and loss function is defined as presented in Eq. (6),

in which RS represents the rating matrix revised with emotional offset, P represents user latent factor matrix, Q represents doctor latent factor matrix, α is used for adjusting the weight of user preference distribution, while β is a parameter used for adjusting the weight of doctor feature distribution, and λ is used for regulating the weight of regularizing filter.

In order to achieve the best performance for HMF, Eq. (6) is expected to be minimized. After factorization, lower dimension matrixes of P, Q and A represent user, doctor, and the latent-topic mapping matrix of latent-factor-rating matrix respectively. Finally, rating is available to be predicted through matrix multiply $P \times Q$, and latent factor matrices P and Q are calculated by SGD, which includes the partial derivatives presented.

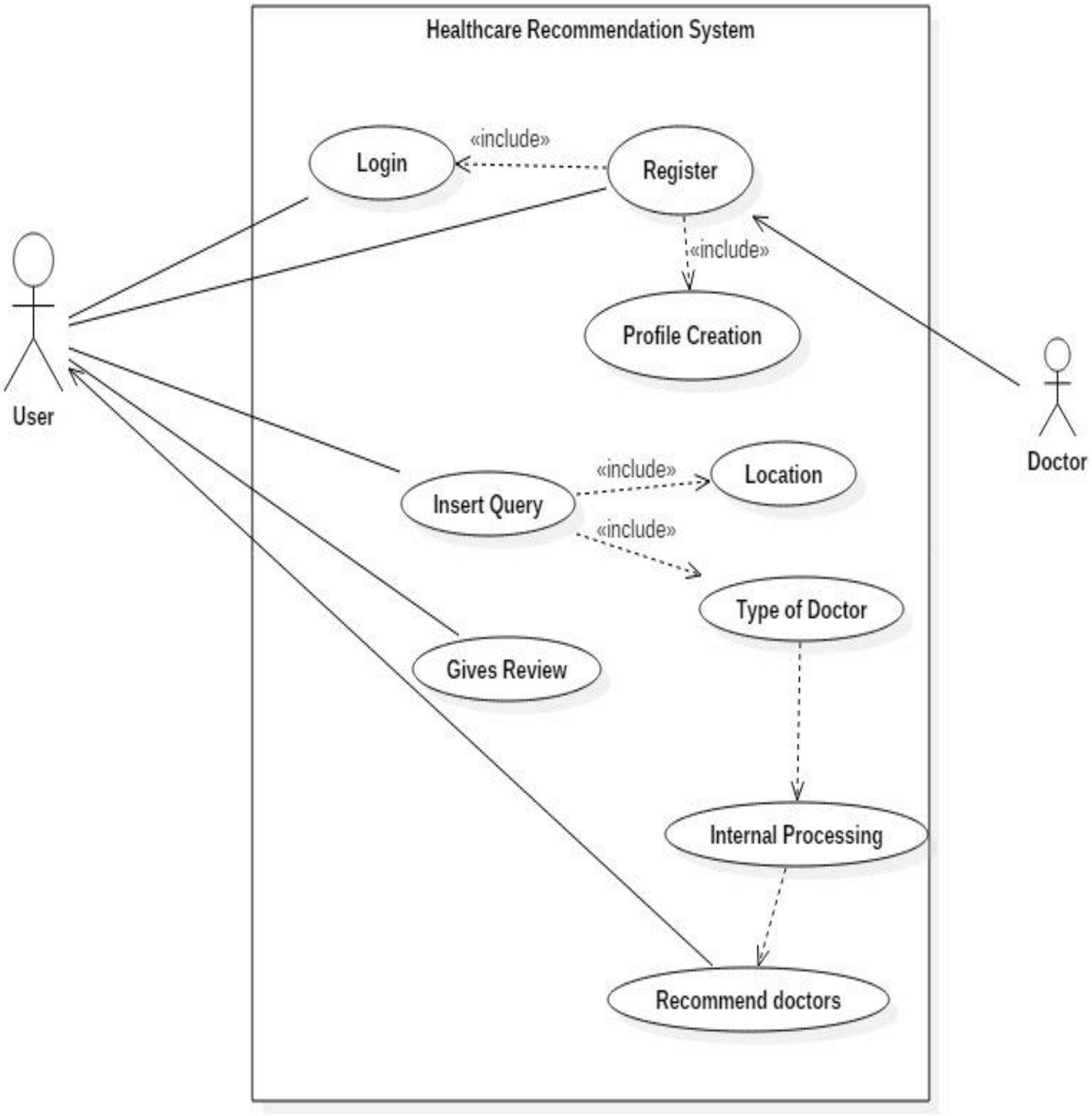


```javascript
var express  = require('express');
var app      = express();
var port     = process.env.PORT || 8080;
var passport = require('passport');
var flash    = require('connect-flash');

var morgan       = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser   = require('body-parser');
var session      = require('express-session');
var querystring = require('querystring');
var spawn = require('child_process').spawn;
global.spawn =spawn;

//////////////////////////// data base connection ////////////////////

const MongoClient=require('mongodb').MongoClient;
MongoClient.connect("mongodb://localhost:27017",(err,client)=>{
    if(err)
    {console.log("Unable to connect")} ;
    var db=client.db('idoctor');
    console.log("Connected");

    global.con=db;
});
///////////////////////////////////////////////////////////////////////


app.use(express.static('views'));

app.use(morgan('dev')); // log every request to the console
app.use(cookieParser()); // read cookies (needed for auth)
app.use(bodyParser.urlencoded({ extended: false })); // get information from html forms

app.set('view engine', 'ejs'); // set up ejs for templating

// required for passport
 // session secret
 app.use(session({
    secret: 'ssasaashnyhajoinansiyh'
```



```javascript
    {console.log("Unable to connect")} ;
    var db=client.db('idoctor');
    console.log("Connected");

    global.con=db;
});
///////////////////////////////////////////////////////////////////////


app.use(express.static('views'));

app.use(morgan('dev')); // log every request to the console
app.use(cookieParser()); // read cookies (needed for auth)
app.use(bodyParser.urlencoded({ extended: false })); // get information from html forms

app.set('view engine', 'ejs'); // set up ejs for templating

// required for passport
 // session secret
 app.use(session({
    secret: 'ssasaashnyhajoinansiyh',
    resave: false,
    saveUninitialized: true,
    cookie: { maxAge: 60000 }
  }))
app.use(passport.initialize());
app.use(passport.session()); // persistent login sessions
app.use(flash()); // use connect-flash for flash messages stored in session


require('./app/routes.js')(app, passport); // load our routes and pass in our app and fully configured passport

// launch ======================================================================
app.listen(port);
console.log('The magic happens on port ' + port);

module.exports = app;
```

Use-Case Diagram



ACTIVITY FLOW DIAGRAM

Module 1

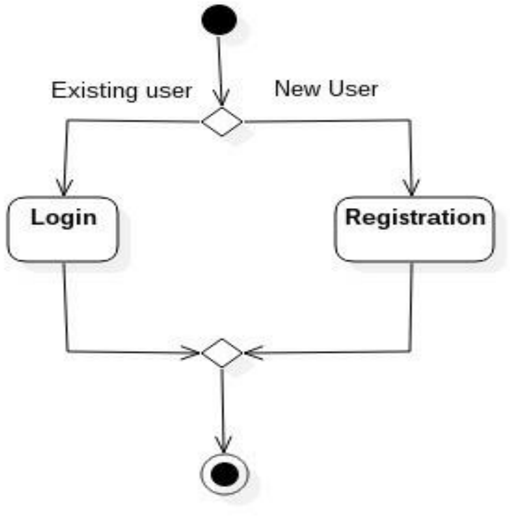


Figure 10: Activity Diagram (Module 1)

Module 2

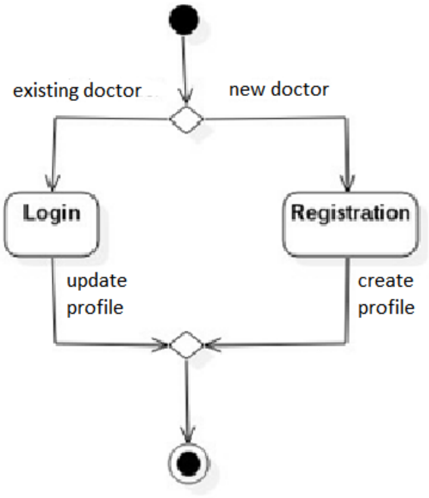


Figure11Activity Diagram (Module 2)

Module 3

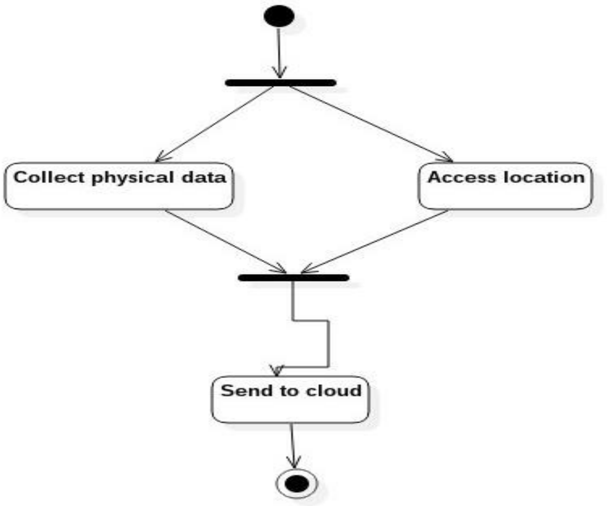


Figure 12 Activity Diagram (Module 3)

Module 4

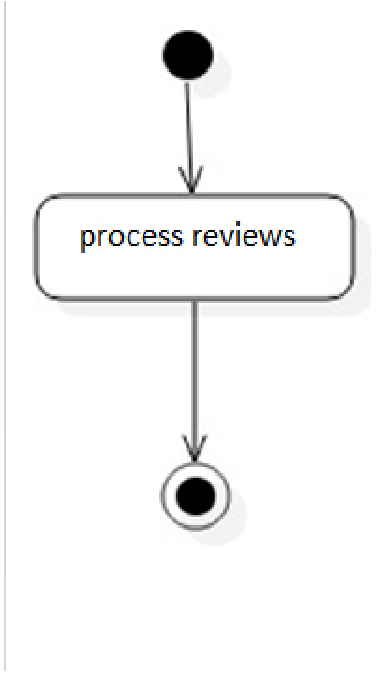


Figure 13: Activity Diagram (Module 4)

Module 5

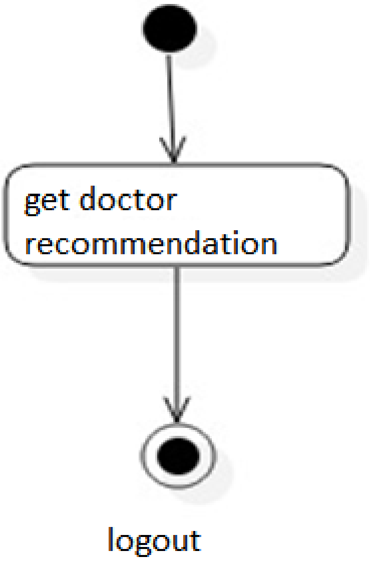


Figure 14 Activity Diagram (Module 5)

Chapter 4

Future Works

Further after completion of the above mentioned project we will try to include live location of the users which will help find doctors closer to the patient's requirement. Along with these we will be including a chat box which will help in better communication between users and doctors. Further, an android application will be developed for these so most of the population can have easy access to the application.

Chapter 5

Conclusion:

The main challenge of finding doctors online will be solved with this healthcare recommendation system which is based on hybrid matrix factorization. It will recommend doctors in far better way than other sites using sentiment analysis of user's reviews and along with that it will take care of user preferences and doctors feature. Through this project we believe that most of the population will be benefitted.

Further after completion of the above mentioned project we will try to include live location of the users which will help find doctors closer to the patient's requirement. Along with these we will be including a chat box which will help in better communication between users and doctors. Further, an android application will be developed for these so most of the population can have easy access to the application.

Chapter 6

References:

[1] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer (8) (2009) 30–37.

[2] R. Gemulla, E. Nijkamp, P.J. Haas, Y. Sismanis, Large-scale matrix factorization with distributed stochastic gradient descent, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 69–77.

[3] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, 2010, pp. 135–142.

[4] L. Baltrunas, B. Ludwig, F. Ricci, Matrix factorization techniques for context aware recommendation, in: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, 2011, pp. 301–304.

[5] M.S. Hossain, G. Muhammad, B. Song, M.M. Hassan, A. Alelaiwi, A. Alamri, Audio-visual emotion-aware cloud gaming framework 25 (12).

[6] Y. Moshfeghi, J.M. Jose, Role of emotional features in collaborative recommendation, in: Advances in Information Retrieval, Springer, 2011, pp. 738–742.

[7] D. Poirier, F. Fessant, I. Tellier, Reducing the cold-start problem in content recommendation through opinion classification, in: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Vol. 1, IEEE, 2010, pp. 204–207.

[8] M. Ko, H.W. Kim, M. Yi, J. Song, Y. Liu, Moviecommenter: Aspect-based collaborative filtering by utilizing user comments, in: 2011 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), IEEE, 2011, pp. 362–371.

[9] D. Agarwal, B.-C. Chen, flda: matrix factorization through latent dirichlet allocation, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, 2010, pp. 91–100.

[10] J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 165–172.

[11] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in: LREC, Vol. 10, 2010, pp. 2200–2204