# **RECOMMENDATION SYSTEM FOR COLLEGE ELECTIVES**

### **A PROJECT REPORT**

#### *Submitted by*

**SHREYAS RAORANE [Reg No: RA1811028010016]**

**SMITHA S [Reg No: RA1811028010020]**

#### *Under the Guidance of*

## **Dr. P. Supraja, Ph.D**.

(Associate Professor, Department of Networking and Communications)

### ***in partial fulfillment of the requirements* *for the degree of***

## BACHELOR OF TECHNOLOGY

## in

## COMPUTER SCIENCE ENGINEERING

## with specialization in Cloud Computing



## DEPARTMENT OF NETWORKING AND COMMUNICATIONS

## COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

## KATTANKULATHUR- 603 203

### **NOVEMBER 2021**



# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

## **BONAFIDE CERTIFICATE**

Certified that this B.Tech project report titled “**RECOMMENDATION SYSTEM FOR COLLEGE ELECTIVES”** is the bonafide work of **Mr. SHREYAS RAORANE and Ms. SMITHA S** who carried out the project work under my/our supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

**SIGNATURE SIGNATURE**

##### **DR. P. SUPRAJA DR. ANNAPURANI PANAIYAPPAN .K**

**SUPERVISOR**

|  |  |  |
| --- | --- | --- |
| **ASSOCIATE PROFESSOR**  Networking and Communications |  | **HEAD OF THE DEPARTMENT**  Networking and Communications |

### i

Department of Networking and Communications

##### **SRM Institute of Science & Technology**

##### **Own Work\* Declaration Form**

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

##### **Degree/ Course : B.Tech - Computer Science and Engineering**

**Student Name : Shreyas Raorane**

**Smitha S**

##### **Registration Number : RA1811028010016**

**RA1811028010020**

**Title of Work : RECOMMENDATION SYSTEM FOR COLLEGE ELECTIVES**

I / We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism\*\*, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

* Clearly references / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| If you are working in a group, please write your registration numbers and sign with the date for every student in your group. |

ii

**ACKNOWLEDGEMENT**

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department**, Dr K. Annapurani Panaiyappan**Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Panel Head, **Dr. P Supraja,**Associate professor, and program coordinators **Dr. M.Saravanan,**Department of Networking and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor**, Dr. M Suchitra,**Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. P.Supraja,** Associate professor, Department of Networking and Communication**,**SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under his/her/their mentorship. He/She/They provided me with the freedom and support to explore the research topics of my interest. Her/His/Their passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

**Shreyas Raorane [RA1811028010016]**

**Smitha S [RA1811028010020]**

iii

**ABSTRACT**

An elective Recommendation System has become an essential technology in the field of higher education in today’s world. The spectrum of subjects and technologies included in today’s education system has increased drastically. Over the span of a century, courses and subjects have penetrated their way in and out of an educational institute’s curriculum. This project is designed considering such wide variation in the list of subjects available to students. While choosing a particular elective in college, multiple parameters need to be considered. Thorough research might help students make this choice with ease, but not always are the details required for such an informed choice available.

The aim of this system is to help students make an informed decision about their electives through the use of machine learning algorithms and historical data. Although multiple course recommendation systems have been proposed, most of these cannot recommend courses based on historical data. The existing system provides a recommendation only based on the students marks and other accreditations. In this model we use collaborative filtering and FP Growth algorithm to put in place an effective structure that will provide recommendations based on the students interest and historical data.

Our system considers a marks threshold which filters out the subjects for the student in which the marks are lower than the given value. This enhances the system’s performance by ensuring that the subjects which have a higher average are being given preference, based on the previous years average marks. Furthermore, both item-item and user-item collaborative filtering is used to recommend the electives.

Similarly the Forward Pattern Growth algorithm only considers the subjects above threshold to generate the FP Tree and the association rules, using which the recommendations are made.

**TABLE OF CONTENTS**

CHAPTER NO. TITLE PAGE NO.

ABSTRACT 4

LIST OF TABLES 5

LIST OF FIGURES 8

LIST OF SYMBOLS, ABBREVIATIONS 9

1. INTRODUCTION. 10

1.1 COLLABORATIVE FILTERING 11

1.2 FP-GROWTH ALGORITHM 12

1.3 MOTIVATION 13

1.4 PURPOSE 13

1. LITERATURE SURVEY 14
2. SYSTEM ANALYSIS 17

3.1 PROBLEM STATEMENT 17

3.2 DATASET OVERVIEW 17

3.3 EXPLORATORY DATA ANALYSIS 18

3.4 MODEL BUILDING 20

3.4.1 COLLABORATIVE FILTERING 21

3.4.1.1 ITEM-ITEM COLLABORATIVE FILTERING 21

3.4.1.2 USER-ITEM COLLABORATIVE FILTERING 22

3.4.2 FP-GROWTH ALGORITHM 22

3.4.2.1 GENERATION OF FREQUENT ITEM SETS 22

3.4.2.2 GENERATION OF ASSOCIATION RULES 23

3.4.2.3 GENERATION OF FP TREE 23

1. SYSTEM DESIGN 24

4.1 BLOCK DIAGRAM 24

4.2 UML: USE CASE DIAGRAM 26

1. CODING AND TESTING 27

5.1 GENERAL 27

5.2 TEST CASE 27

5.3 TEST PLAN 28

5.4 TESTING 29

5.5 IMPORTANCE OF TESTING 30

5.6 LIBRARIES 31

5.6.1 GOOGLE.COLAB 31

5.6.2 NUMPY 31

5.6.3 PANDAS 31

5.6.4 SCIPY 31

5.6.5 SCIKIT-LEARN 31

5.7 SOFTWARE MODULE 32

5.7.1 COSINE SIMILARITY 32

5.7.2 ADJACENCY MATRIX GENERATION 32

5.7.3 SIMILARITY SCORE CALCULATION 33

5.7.4 ASSOCIATION RULES GENERATION 34

5.7.5 FREQUENT ITEMSET MINING 34

5.7.6 FP TREE 35

1. RESULT AND ANALYSIS 36

6.1 COLLABORATIVE FILTERING RESULT 37

6.2 FP GROWTH ALGORITHM RESULT 38

1. CONCLUSION AND FUTURE WORK 39
2. PAPER PUBLICATION STATUS 40

REFERENCES 41

APPENDIX 44

PLAGIARISM REPORT 49

**LIST OF FIGURES**

|  |  |
| --- | --- |
| FIGURE NO. | FIGURE NAME |
| Fig 1.1 | Collaborative Filtering |
| Fig 1.2 | FP Growth Algorithm |
| Fig 3.1 | Dataset |
| Fig 3.2 | Collaborative Filtering One-Hot encoding |
| Fig 3.3 | FP Growth Data Preparation |
| Fig 4.1 | Block diagram |
| Fig 4.2 | UML Use case diagram |
| Fig 5.1 | Cosine Similarity |
| Fig 5.2 | Adjacency Matrix Generation |
| Fig 5.3 | Similarity Score Calculation |
| Fig 5.4 | Association Rules |
| Fig 5.5 | Frequent Itemset |
| Fig 6.1 | Collaborative filtering Result |
| Fig 6.2 | FP Growth Algorithm Result |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| FP-Growth algorithm | Forward Pattern growth Algorithm |
| FP-Growth tree | Forward Pattern growth tree |
| RAD Model | Rapid Application Development |

**CHAPTER 1**

**INTRODUCTION**

Every day, we are bombarded with a load of information, many times in which we have to make choices that can affect our lives. When the number of these choices are excessively high, there is a need to filter, categorize and deliver relevant information in order to omit the problem of information overload. The human mind is capable of doing such complex activities only after thorough training and long durations of practice.[1][2] Even when done so, the efficiency of the filtration is subjected to multiple facets thereby reducing the accuracy of the result. This major reason has given rise to the use of recommendation systems that provision the execution of such complex processes through machine learning and deep learning algorithms.[3] Recommendation systems solve the problem by scanning through a large set of information to provide users with personalized content based on parameters that affect the accuracy of the result.

Education systems these days have a variety of courses and subjects to choose from. Though such wide variation is advantageous in multiple ways, studies suggest that students derail from academics after choosing the wrong course. This is primarily due to the unavailability of information and inaccurate research results that misguides the student. [4]Therefore using a recommendation system for such in-depth decisions is a smart solution for avoiding problems.

The aim of our system is to help students make an informed decision about their college electives through the use of machine learning algorithms and historical data. In our model, we use collaborative filtering and FP growth algorithm to give a recommendation for the electives to be chosen in college. Our system considers a marks threshold which filters out the subjects for the student in which the marks are lower than the given value. [5][6]This enhances the system’s performance by ensuring that the subjects that have a higher average are being given preference. The courses are compared based on the similarity of the choice inputted to the list of electives that are available for the given semester.

**1.1 COLLABORATIVE FILTERING**

Collaborative filtering is a method used by recommendation systems to make self-stimulated predictions based on preferences collected and the similarity of information from many users. It is one of the most common techniques used to build an intelligent recommendation system.[7][8] It works on huge data sets with a lot of preferences. From this huge dataset, the system narrows to a smaller set of users with similar tastes. Broadly, collaborative filtering can be divided into 2 parts.

* **Item-Item Collaborative Filtering**

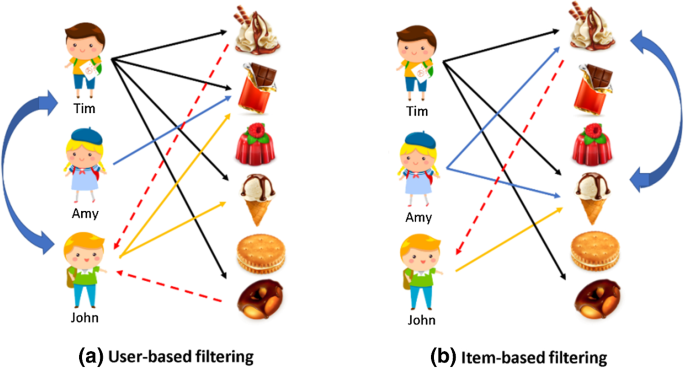
Item-Item Collaborative Filtering is a type of collaborative filtering method that looks for similar items based on the items that the users (or student) have already interacted with. In our use case, the courses have higher similarity values when they are often chosen together by other students or students from previous years.

* **User-Item Collaborative Filtering**

User-Item collaborative Filtering technique works on the principle that similar people have similar taste. An user-item matrix is made and based on that the similarity score is calculated for each item.

To explain this in lay-man terms, it’s primarily assumed that if person A has the same preference as person B in a given issue/situation, it’s likely that he has the same preference in another random scenario too.

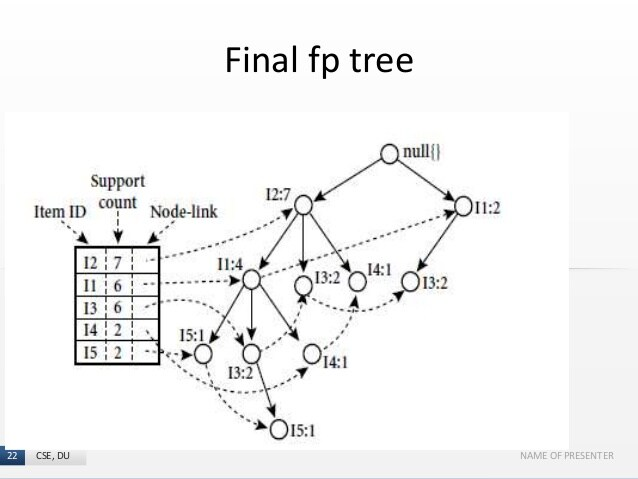
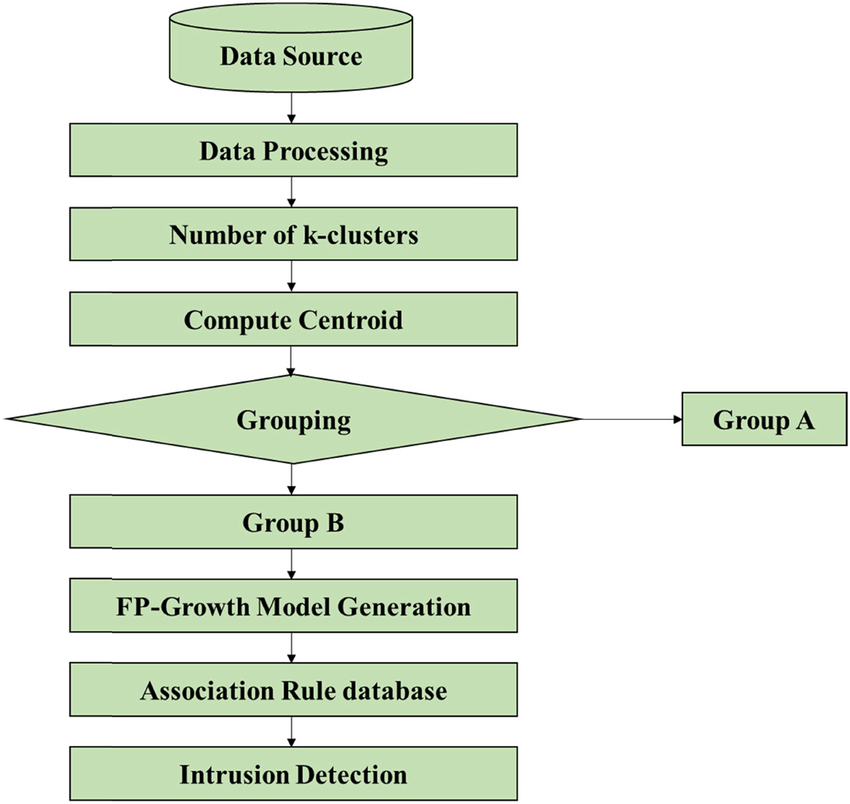
This technique works by searching large volumes of entities/people and finding a smaller collection of users with a palate similar to a particular user/entity. It considers the items they like and integrates them to create a ranked list of recommendations. The fig 1.1 demonstrates collaborative filtering.



**Fig 1.1 Collaborative Filtering**

**1.2 FP-GROWTH ALGORITHM**

FP- Growth Algorithm is an improvised form of the Apriori Algorithm that is commonly used for frequent pattern mining.[9] The basic idea behind this algorithm is to find frequent patterns/itemsets or associations from data sets.[10] The Apriori Algorithm basically traverses the data set back and forth to check the concurrence of the products. In order to be faster, the FP-Growth algorithm organizes the dataset into a tree data structure, which quickens the process of traversing data.



**Fig 1.2 FP Growth Algorithm**

The root of the FP- Growth tree holds the null node. Each node holds an item, while the relationship between the nodes is the itemsets. The order is maintained while forming the tree. The FP-tree is concise and is used to directly generate large itemsets. FP-tree uses a recursive divide-and-conquer technique to traverse the frequent itemsets. Fig 1.2 is a flowchart representing FP Growth algorithm.

**1.3 MOTIVATION**

**“We make a living by what we get, we make a life by what we give.” – Winston Churchill**

Quite a few students are confused about the electives they should opt for for the college curriculum, and a lot of them turn to their seniors for guidance. But when information isn't available there too, they make choices that they might regret later. The aim of this project is to make this process simpler and help students make an informed choice by making use of historical data to recommend students electives based on what their seniors have chosen over the years.

**1.4 PURPOSE**

The purpose of this project is to help students make the right choice about their elective from a large list of subjects. Choosing electives in college is a rigorous process and takes a lot of resources and time to make an apt choice. If provided with an alternative to ease this process, students can save up on a lot of time. This has been the sole motivation and purpose of this project. Once the students get the top 5 recommendations from our system, they can explore them based on their personal interests.

**CHAPTER 2**

**LITERATURE SURVEY**

This Project on Recommendation system for college electives is designed based on the following literature reviews listed below, that have been studied and analysed in depth. We will be discussing them in depth with their methodology and outcomes.

**Movie Recommendation System**

This paper uses the information filtering approach to predict the movie preference of a user. It makes the use of information provided by the user and is based on a collaborative filtering approach.MovieREC recommends a movie using the K-means algorithm. It calculates the cumulative weight of different attributes.A combination of collaborative and Content-based filtering of data is used to provide recommendations to users.[11] It uses similarity of users, genres and preference of movies as parameters to give the recommendations. The system is based on the belief that similar users will rate movies in a similar fashion and gives the correct preference.The system was developed in PHP using Dreamviewer 6.0 and Apache Server 2.0. Results were not validated because of the absence of a large Dataset.Machine Learning algorithms were not implemented. [12]The results were based on historical data and were not tailored for every user.The algorithm utilizes the preferences of previous users across different genres and the association of the genre to movies to calculate the movie score for that particular user.A new score is computed for each movie. This system works with a low margin of error.

[13]MovReC is a personalized movie recommendation for children to provide suitable entertaining opportunities along with education for children. For every new movie, the "children appropriate score" is calculated using: 1. Backpropagation value, 2. MF Rating Prediction, 3. LDA Topic Analysis, 4. Sentiment Analysis, 5. genre similarity, 5. Like/Dislike Counts. The ratings are predicted using matrix factorization. The sentiments are approximated using the user's reviews.It is one of the first personalised recommendation systems.It applied the appropriate ranking analysis and topic analysis.This is followed by sentiment analysis and like/dislike counts. Finally genre analysis is used to match the type of movie.

**Collaborative Filtering**

This paper discusses the use of a user-user collaborative filtering algorithm to suggest related items to the current or other users. [14]It also discusses algorithms like Check URL Algorithm and Subspace Clustering algorithm for writing the research paper. The paper uses data from other researchers’ browsing patterns along with collaborative filtering to find patterns. This method avoids issues such as mis-interpretation of content. The proposed solution is fast, scalable and produces high quality recommendations[15].A more sophisticated ranking scheme could be developed to improve perceived quality and usefulness of recommendation. This paper discusses three recommendation systems based on rating estimation. 1. Collaborative Filtering System, 2. Content Based System, 3. Hybrid System. It dwells into the details of each system to analyse its working. It also states the merits and demerits of various approaches used in the system as well. Each method has its own drawbacks. The collaborative filtering memory based approach has a cold start problem. i.e always sufficient user information is needed.[16] New items cause problems in both collaborative and content based approaches. Content based approaches have problems such as Limited Content Analysis and Over-Specialization. Hybrid Models have issues with Integration and efficient calculation.

[17] This paper presents a Recommender System which uses collaborative Filtering. The algorithm generates a list of items similar to buyers preferences. It incorporates item-item collaborative filtering along with Jaccard Similarity for the recommendations. JS is the ratio of common users divided by the sum of users who have rated the two books in question individually.[18] Book recommendation system uses Root Mean Square Error(RMSE) statistical accuracy metrics for evaluation.The smaller the value of RMSE, the better the recommendation system. The recommendation system faces limitations due to the amount of data that needs to be handled. The model also does not consider absolute rating and hence might give the wrong recommendation.The trustworthiness of the algorithm is the third and final issue.

**FP-Growth Algorithm:**

The Forward Pattern growth algorithm generates a FP Tree and uses the prefix tree representation of the transactional database.[19] The advantage of this system is the considerable amount of memory it saves. It looks for frequent datasets above a certain threshold. Two passes are carried out over the dataset to build a complex but informative structure called a FP-Tree. Next, the Forward Pattern Tree is traversed to extract the frequently occuring itemsets.[20] This paper uses Frequent Pattern Growth Algorithm for mining medical data to generate patterns and rules. The aim is to find the association between several attributes of patient data. Data preprocessing consists of Data discretization and transformation . The input data entered by the user in the model at the beginning determines the corresponding result. Next step is to select the appropriate method for the mining task. Finally the result is generated and new rules and patterns are extracted from the data.

The results provide guidance to the doctors as well as more understanding about the relation of a doctor and a patient. [21]The presented model can be adapted by any type of organisation, marketing strategies, sales forecast, insurance companies etc. [22] This method uses the operation of computing a projection of an FP-tree in its two variants of the core . The tree is pruned by removing items that have become infrequent. First the items are sorted in descending or a random order.Next the Initial abstract tree is generated. This paper described the implementation of the FP growth algorithm. Apriori and Eclat algorithms are clearly outperformed by FP growth even in highly optimised versions.

[23] This paper proposes the parallelization of the FP growth algorithm on distributed machines. This ensures that each machine is given an independent group of mining tasks. FP growth algorithm works on two scans. In the first scan, the items are sorted according to frequency in a descending order and in the second sac, a FP tree is generated. This tree is constructed recursively. It has been demonstrated that PFP can achieve virtually linear speedup. This is a promising technology for query recommendation search engines. [24] [25]The FP growth algorithm has certain advantages such as it takes only two passes and due to the overlapping paths, the data is compressed. It is not a necessity to perform Candidate generation. Next step is to select the appropriate method for the mining task. Finally the result is generated and new rules and patterns are extracted from the data.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 Problem Statement**

Given a data set of users with their previous ratings and input on average marks for a set of courses/electives and the current users semester and interested courses/electives, the system can recommend a set of electives to the user.

**3.2 Dataset Overview**

Dataset: [University Dataset | Kaggle](https://www.kaggle.com/ananta/student-performance-dataset)

The dataset comprises details that include Information about its departments, Professors, Student Counselling, Courses offered, course students selected to get admission and student performance in different examinations of an imaginary institution.

Feature Engineering:

The dataset is an amalgamation of categorical and numerical values. We will be using a threshold value to convert the numerical value of marks to a categorical variable. Further processes are carried out after concatenation and differentiation of data based on semester and paper respectively.

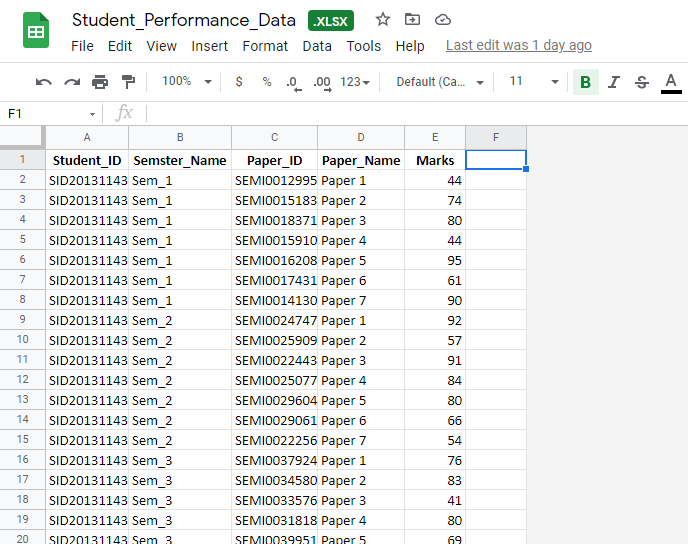
We will be focusing on just one sheet from the Database, namely “Student\_Performance\_Data.xlsx”. It contains 5 columns:

1. Student\_ID: Unique ID used to identify the student
2. Semester\_Name: Name of the semester
3. Paper\_ID: Unique ID to recognise the paper
4. Paper\_Name: Name of the paper
5. Marks: Marks in that particular paper

The Dataset contains 209612 entries for papers and marks split across seven semesters for each student. We only consider one semester at a time while recommending the courses.

**3.3 Exploratory Data Analysis**

This section was added so that we can evaluate the importance of each column in the Dataset.



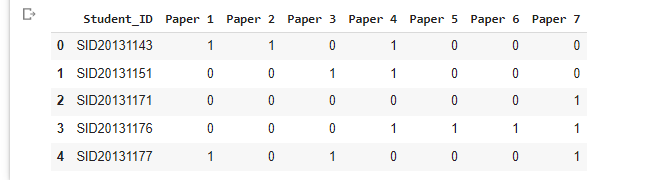
**Fig 3.1 Dataset**

Steps:

1. The Dataset is split based on the semester and stored in a dictionary. Only the data with the semester chosen by the user is considered for further calculations. Figure 3.1 shows sample dataset
2. A marks threshold is applied which changes the numerical data of Marks to categorical data. The threshold value is given by the user.
3. Only the Columns, ‘Student\_ID’ and ‘Paper\_Name’ are chosen for further analysis. Moreover, only the rows which have marked value as True are considered.

**Collaborative Filtering**

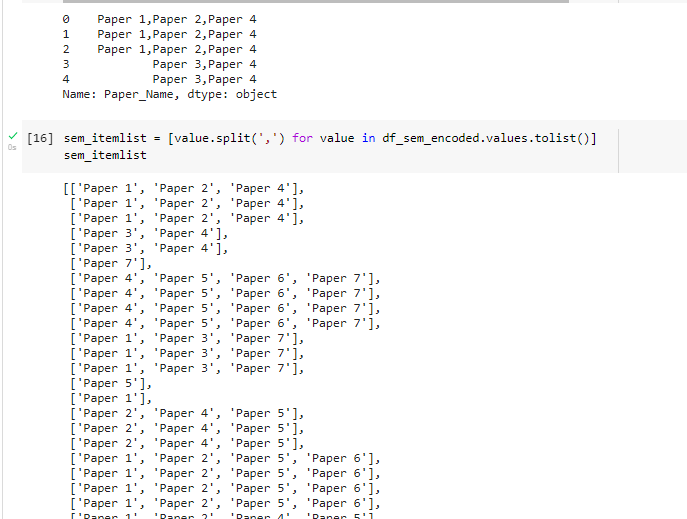
1. The data is one-hot encoded for calculation purposes.



**Fig 3.2 Collaborative Filtering One-Hot encoding**

**FP Growth**

4. The data is then grouped based on Student\_ID converted to a list of lists to be passed into the ‘fpgrowth’ module from ‘fpgrowth\_py’ library.. Fig 3.2 shows one-hot encoding



**Fig 3.3 FP Growth Data Preparation**

**3.4 Model Building:**

A model can be defined as a representation of a process from a particular perspective. The main components of any model are:

1. Specification: Our system is designed to help students choose college electives based on historical data.
2. Design and Implementation: Model was designed to use collaborative filtering and FP growth algorithm. It is implemented in the google colab environment.
3. Validation: The model and the results were validated.
4. Evolution: Changes were made to the system in order for it to cater to the consumer needs more efficiently.

There are several types of software process models. These are tailored according to every project. Some of the most used ones are:

1. Waterfall Model: Declustering of activities into linear sequential phases where in each phase is dependent on the previous one.
2. V Model: Process bent upwards after coding phase, so as to inculcate testing.
3. Incremental Model: Model is designed, implemented and tested incrementally.
4. Iterative Model: Starts with simple features and progressively gains more complexity.
5. RAD Model: This model facilitates incremental prototyping approach to software development. This allows the end users to produce better feedback when examining a live system.
6. Spiral Model: This is a risk driven model. The project is delivered in loops while supporting and handling risks.
7. Agile Model: Supports a way of thinking that enables businesses to innovate, respond to changing demand while mitigating risk.

Source: [13] [What is a Software Process Model? (visual-paradigm.com)](https://www.visual-paradigm.com/guide/software-development-process/what-is-a-software-process-model/)

We have used an incremental build model. Our project was split into two major portions and each section was worked on separately in an incremental fashion. Detailed Elaboration of the modules used has been given in section 5.7 of this document.

**3.4.1 Collaborative Filtering**

This method filters data by using the interactions and data previously collected by the system from other interactions and users. Collaborative Filtering in other use cases normally is content based or people based. To enhance our accuracy we are using a hybrid approach which combines both content based (item-item) collaborative filtering and people based (user-item) collaborative filtering.

**3.4.1.1 Item-Item Collaborative Filtering**

In item-item collaborative filtering technique, we calculate the similarity between the items. It is content based and is based on the assumption that if a person likes some product A, s/he is also likely to like B based on how frequently they're bought together.

We use cosine similarity to find the similarity between items and then store it in a matrix. Hence we get the similarity between the items and complete a module of the project.

**3.4.1.2 User-Item Collaborative Filtering**

In user-item collaborative filtering we calculate how similar people or in this case the combination of the subjects are. It works on the basis that if 2 people A and B like product P1 and person A likes product P2, it is also likely that Person B will like product P2.

The values with the most score from the item-item matrix when considering the partially known solution (accepted from the user) are taken and a new Matrix called Neighbourhood matrix is formed. This contains the top 4 most similar values to any given paper in the known solution.

Another Matrix is extracted which tells us the similarity between the elements in the partial solution provided by the user. The result of this matrix is used to determine which subjects are closest to all of the other subjects.

Finally, this data is corresponded with the neighborhood Matrix to calculate a score for each paper of interest. This score is the similarity score calculated using dot product and normalisation. The papers with the highest 2 values are given as outputs as the recommendation.

**3.4.2 FP Growth Algorithm**

FP Growths stands for Forward Pattern Growth Algorithm. It forms a Tree which then it uses to give recommendations. It is an improvement on the existing Apriori Algorithm which is used to Generate the association rules. We inculcate minimum support and minimum confidence as inputs as well.

**Support = (Number of times an item or itemset appears in the database) / (Number of baskets in the database)**

Confidence: is the percentage of transactions , containing both the values in question.

**3.4.2.1 Generation of Frequent Item sets**

In FP Growth Algorithm, the first step is the generation of frequent item sets. Once the preparation of Data is completed, we put the items in a list of lists. We go through the list and save which items occur more frequently. The frequency table is made.

**3.4.2.2 Generation of Association Rules**

Based on the frequency table generated above, the association rules are generated. The items occurring more frequently together are more likely to be considered as a rule.

Association rules help us map the solutions so as to generate the FP Tree.

**3.4.2.3 Generation of FP Tree**

As the final step of the FP Growth Algorithm, a Forward Pattern tree is generated. It combined the frequency table and the association rules. This is generated on the second pass of the item-list

Once we have the FP Tree, it can be transversed ahead to get the potential recommendation values based on the already provided partial input. The values which do not match the minimum support and/or the confidence values are deleted from the FP Tree.

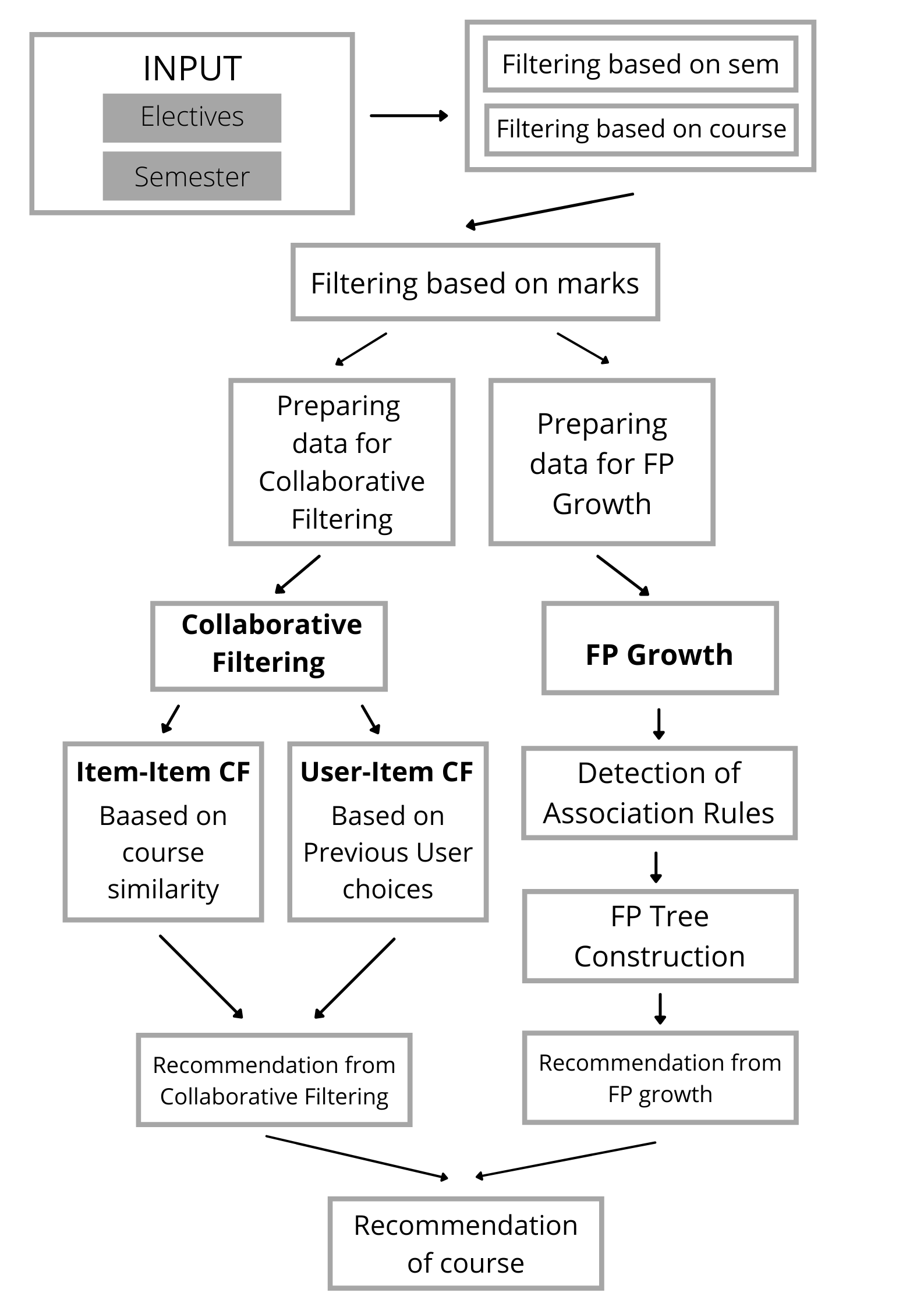
We print out the most frequently occurring item sets from the table and give the user an option to compare them with the results in collaborative filtering to make an informed decision.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 BLOCK DIAGRAM**

The fig 4.1 represents the block diagram for collaborative filtering

****

**Fig 4.1 : Block diagram**

The block diagram mentioned above explains in brief the execution of the process involved in providing a recommendation of an elective. Choices of electives interested in and semester are taken as input. The system then filters the data based on semester and course opted. The next level of filtering happens based on marks. Subjects with an average lower than the given range are dropped off. Now the data is prepared in 2 sets, one for collaborative filtering method and the other for FP-Growth algorithm.

In the collaborative filtering technique, two methods of filtering are used;

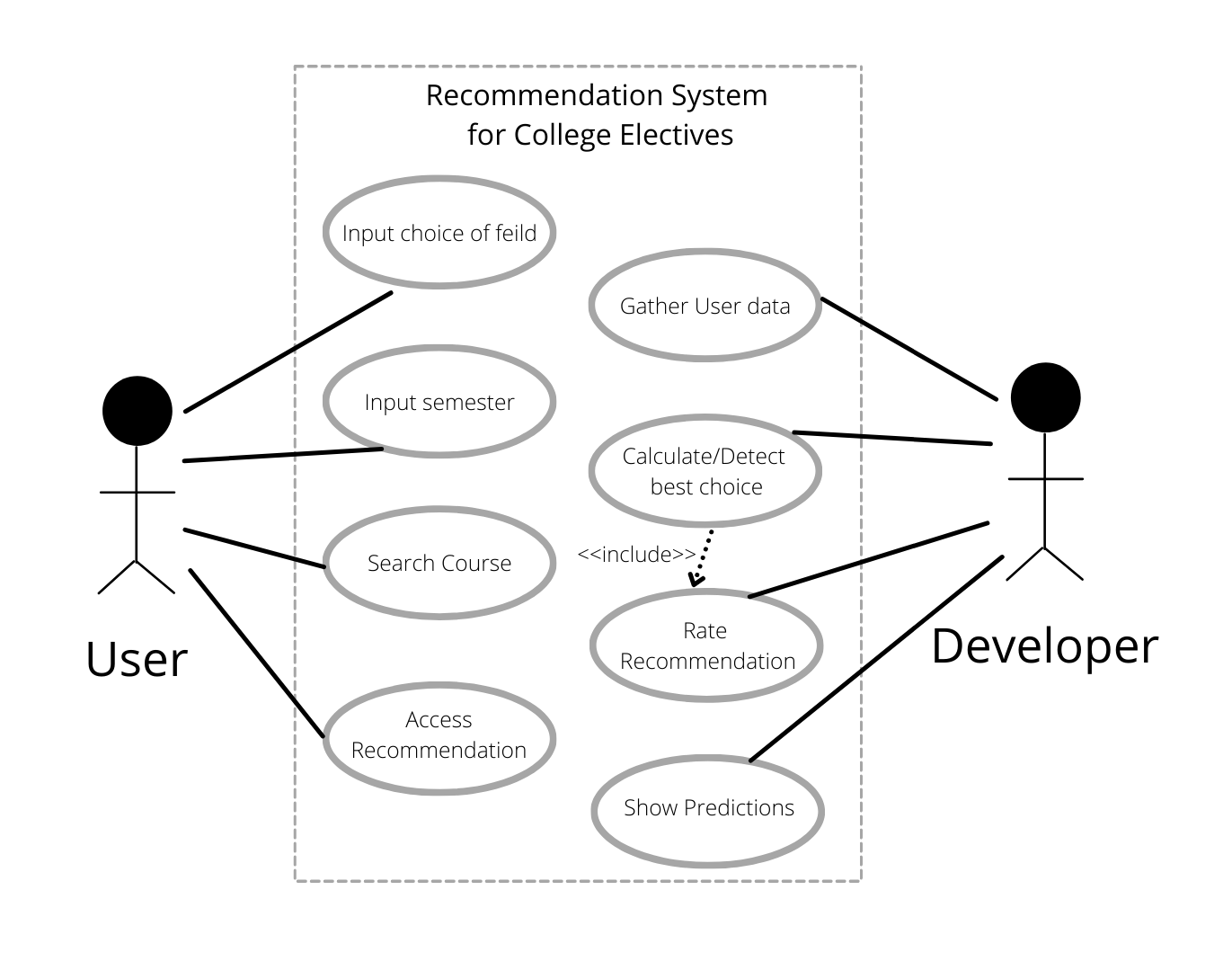
1. Item-item and
2. User-item

In item-item filtering, the data gets segregated based on the similarity of the course while in user-item filtering, the segregation happens depending on the similarity of the previous user's choices.

The FP-Growth algorithm performs the segregation based on the formation and detection of the association rules. The construction of the FP tree facilitates the traversal of data and overcomes a lot of setbacks of the previous models.

Recommendations from both these algorithms are considered while giving out the result.

**4.2 UML: USE CASE DIAGRAM**

****

**Fig 4.2: UML Use case diagram**

The above diagram explains in brief the relation between the use cases of the proposed system. The input, consisting of course and semester, are directly mapped to the user. The user will also be able to search through the courses and access the recommendations.

On the developers end, gathering and segregation of data tops the list of use cases. Calculating and detecting the best choice is also an essential function. While evaluating the best choices, the developer needs to consider the rating from the previous user. Displaying the predictions ends the process from the developers end. Fig 4.2 represents the UML use case diagram for the recommendation system.

**CHAPTER 5**

**CODING AND TESTING**

**5.1 GENERAL**

In simple terms, testing is finding out if something works. It is the process of evaluating and verifying that a software product or application does what it is supposed to do. The aim is to find a mistake which could have harmed the reputation of the product if released. A good test is defined by its ability to spot errors. If the tests are written before the program it is called as test driven deployment.

The final aim of program testing is to make sure that the product has no defects and the outputs are the

desired outputs.

**5.2 TEST CASE**

Test cases can be in the form of variables or conditions which are used to predict the real results according to the needs of the customers.

By definition, in software engineering, a test case is a set of specific inputs, execution criteria, testing methodology, and derived results that define a single test to be executed to in order to achieve the required target, such as to Source: [14] [Test case - Wikipedia](https://en.wikipedia.org/wiki/Test_case)

Our test cases here are considered to be some predefined values, which we know occur together often in the dataset.

There are several characteristic that define a good test case:

1. Potential to find Defects: A test case must be designed in a way that it points out the defects in the code. In the use case mentioned here, based on the recommendation, we can verify the output.
2. Atomic: Test case must have a single specific goal targeting a particular area.
3. Accurate: The given result of the test case must be known and accurate.
4. Traceable: The result of the test case must be traceable back to the requirements of the test case
5. Comprehensible: All test cases must be understandable by others.
6. Complete: The test case must be complete in itself, specific inputs and conditions if necessary must be specified.
7. Repeatable: Test case must be repeatable so that it can be used any number of times.
8. Unique: Test case must be unique.

**5.3 TEST PLAN**

A test plan is the formal way of testing any software/product in a project . It provides a detailed description of the scope, approach, resources and schedule of activities that the project comprises.. It identifies features, testing tasks, degree of independence of the tester, environment, design techniques, criteria for entry and exit, risk and contingency plans. They are prepared by test engineers prior to testing.

There are 8 steps to write a Test Plan:

1. Analyze the product: Analyze the system and jot down important points and criteria. Proper documentation is essential.
2. Design the Test Strategy: Testing to be done based on known commonly occurring courses with a higher average. This can be divided into sub-steps such as:
   1. Scope Definition
   2. Testing Type
   3. Documentation of Risks
   4. Creating test logistics
3. Define the Test Objectives: To verify the code and the formulae used.
4. Resource Planning: Coding done on Google Colab, Colab resources are used for user convenience.
5. Plan Test Environment: Colab Environment with provided Disk Space and RAM.
6. Schedule and Estimation: Estimated time: 5 min
7. Determine Test Deliverables: Output determined and Present.

**5.4 TESTING**

Testing is the procedure of running a program in order to spot errors within it. This helps us to rectify the errors and hence better the final product.

There are several principles of testing:

1. Tests must meet the customer requirements: Our system must be able to help students choose their electives and give an unbiased opinion.
2. Software testing, if possible, should be performed by a third party: Our system was tested by our peers based on the small dataset to verify the results.
3. Exhaustive testing is not possible: Huge datasets and a higher number of people are needed.
4. All tests conducted were planned beforehand
5. Testing follows Pareto rule (80/20 rule)that explains that 80% of the errors come from 20% of the program components: This was observed while generating the FP Tree for certain support and confidence values, the problem was solved by defining the values and providing the user with the ability to alter these values.
6. Testing starts from small parts and extends to larger parts.

Types of testing includes:

1. Unit Testing: In this, the smallest unit of the software design is the point of focus. For our use case, these consist of cosine similarity calculations, neighbourhood matrix generation, similarity score, Generation of association rules, Generation of freq item list.
2. Integration Testing: Includes putting the units together and building a program structure dictated by design. This includes putting cosine similarity calculations, neighbourhood matrix generation, similarity score under collaborative filtering and Generation of association rules, Generation of frequent item list under FP Growth Algorithm.
3. Regression Testing: This is done every time a new module is added to the project.
4. Smoke Testing: Allows to check whether the software is ready for further testing and all current issues have been resolved.
5. Alpha testing: Testing was done by the developers
6. Beta testing: Testing was done with the help of a limited number of peers.
7. System testing: We focus on only the input and output. Generally this is done to make sure that the software is compatible with different operating systems. We have eliminated this problem by moving the functionality to Google Colab.
8. Stress testing: Unfavorable conditions were given to the system to see how it performs. Colab Limits the hardware capabilities.
9. Performance testing: This testing methodology tests the software’s run-time performance. Speed and effectiveness of the software were tested. Most of the time is spent in Uploading the file, user-item calculations in collaborative filtering and generation of FP Tree in FP Growth Algorithm.
10. Object-Oriented testing: Combination of various testing methods to determine the validity of each testing methodology.

Source:[16] [Types of Software Testing - GeeksforGeeks](https://www.geeksforgeeks.org/types-software-testing/)

**5.5 Importance of Testing**

Testing is considered to be a crucial phase of the SDLC (software development life cycle). It helps identify bugs and errors at an early stage and helps avoid them.

Some of the advantages of testing are:

1. Reduction of cost: The cost of fixing a bug is higher at later stages of the SDLC.
2. Improves Quality: As the bugs and errors are fixed at an early stage, there are no patches to be put later. This increases the efficiency and hence the quality of the code.
3. Necessity: As no software can be error free, testing is a necessity.
4. Environment: Testing is done in a production environment. In our use case, we have used google colab and hence have avoided this issue altogether.

**5.6 Libraries**

**5.6.1 Google.colab**

Colab is a Google Research product. It permits its users to write and execute python code and is well adapted for concepts such as ML, data analysis etc. As Colab is hosted by Jupyter notebook services, it does not require any setup. It provides free access to computing resources including GPUs

**5.6.2 Numpy**

Noted as a fundamental library for scientific computing in Python, Numpy provides a multi-dimensional array object, multiple derived entities, and a large spectrum of computational methods to enable fast mathematical, logical and analytical operations.

Other operations that numpy facilitates includes shape manipulation, selecting I/O, basic linear algebra, basic statistical operations, sorting, discrete Fourier transforms and random simulation

**5.6.3 Pandas**

A python library majorly used while working on datasets. It comprises functions that allow data to be analyzed, cleaned, explored, and manipulated easily. Pandas makes data more readable and relevant.

**5.6.4 Scipy**

An open-source library in python, scipy is used for solving scientific, mathematical and technical problems. Using a wide spectrum of high-level commands, Scipy enables its users to manipulate data and visualize the data. This library is built on the Python and NumPy extension.

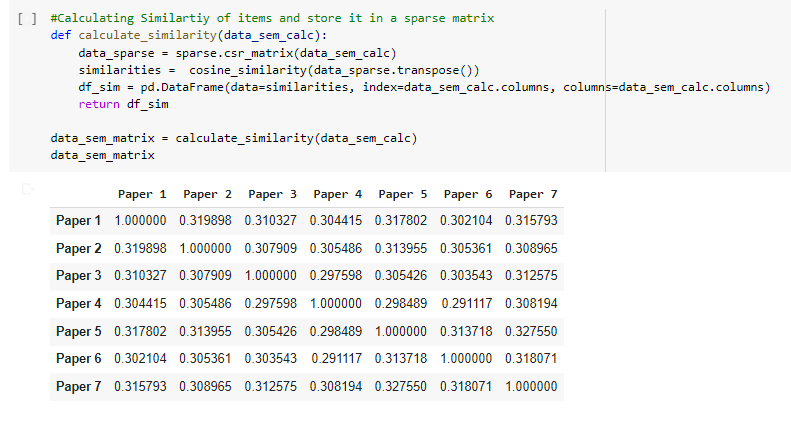
**5.6.5 Scikit-learn**

Scikit-learn (Sklearn) is a robust library in python for machine learning. It provides a set of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library is built upon NumPy, SciPy and Matplotlib.

**5.7 Software Modules**

**5.7.1 Cosine Similarity**

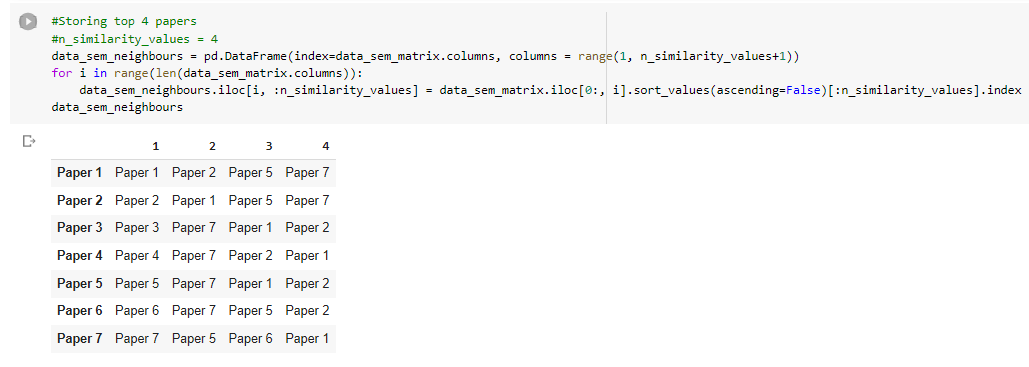
A metric used to measure the similarity of 2 entities regardless of their size. Mathematically it is the measure of the cosine of the angle swiped in between two vectors. In this project, we use this metric to measure the similarity in the filtering techniques.Fig 5.1 shows the matrix once cosine similarity is calculated for each of the values. The cosine similarity value for each course with itself will always be 1.

****

**Fig 5.1: Cosine Similarity**

**5.7.2 Adjacency Matrix Generation**

It is a representation of a finite graph. Each entity of the matrix indicates whether a pair of finite sets of vertices are adjacent in the graph. Fig 5.2 shows the Adjacency or the neighbourhood matrix generated for top 4 values.

****

**Fig 5.2: Adjacency Matrix Generation**

**5.7.3 Similarity Score Calculation**

Once we have the neighborhood Matrix and the Matrix for the similarities between the papers in the known solution. A dot product is done to get a score corresponding to each potential recommendation. This list of scores is called the similarity score. Fig 5.3 shows the similarity score calculation with respect to the courses already selected.

****

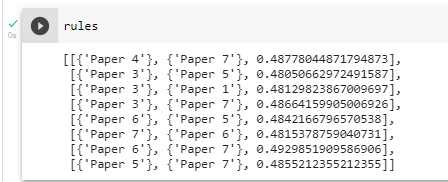
**Fig 5.3: Similarity Score Calculation**

**5.7.4 Association Rule Generation**

Association Rules explains the relationship between two or more attributes of a dataset. It is mainly in the form of- If antecedent than consequent. It is a two step process comprising of

1. Finding frequent Itemsets
2. Generation of strong association rules from frequent itemsets

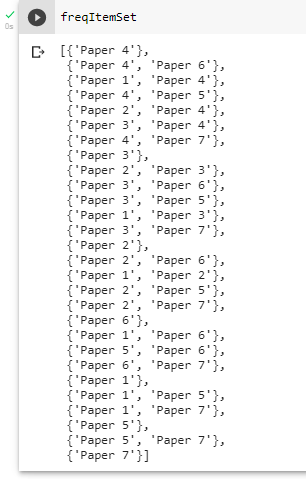
Fig 5.4 Shows the association rules generated after the first pass of the dataset.

****

**Fig 5.4: Association Rules**

**5.7.5 Frequent Itemset Mining**

Frequent itemset mining is simply the process of finding items that will go together. It does so by combining various methods such as Association Rules, Correlation of items, finding frequent patterns or even casual structures within a set of items or objects in transaction databases, relational databases, and other information repositories. The aim of this method is to find regularities in a given behavior. This way we can classify or club the items as one itemset. Fig 5.5 shows the frequent item sets generated.

****

**Fig 5.5: Frequent Itemset**

**5.7.6 FP Tree**

A FP-tree is a data structure that represents the data set in the form of a tree. Each traversal is read and mapped onto a path in the FP-tree. This is done until all branches have been read. The compactness of the tree is ensured by grouping common subsets.

**CHAPTER 6**

**RESULT AND ANALYSIS**

Result and analysis is important to verify that the model is performing exactly how it is supposed to. Analysis of the result is important to verify the accuracy of the system.

For both the models we started with data containing only the marks for the papers from all semesters. We worked on the data to be inputted into the collaborative filtering and the FP Growth Algorithm.

Item-item collaborative filtering was used which was followed by user-item collaborative filtering. Finally based on the score the Recommendation was calculated.

The Result of Collaborative filtering was a direct recommendation of the top two subjects the student should choose based on their current choice.

For the FP Growth Algorithm, we started with the first pass which gives us the recurring item sets and the association rules to be used. In the second pass we generate the FP Tree. The minimum support and the minimum confidence values were accepted by the user. Hence, FP Growth Algorithm gives us the most frequently occurring solutions and leaves the user with the freedom to choose the elective based on that information.

Finally it is left up to the user to choose the elective.

**5.6.1 Collaborative filtering Result:**

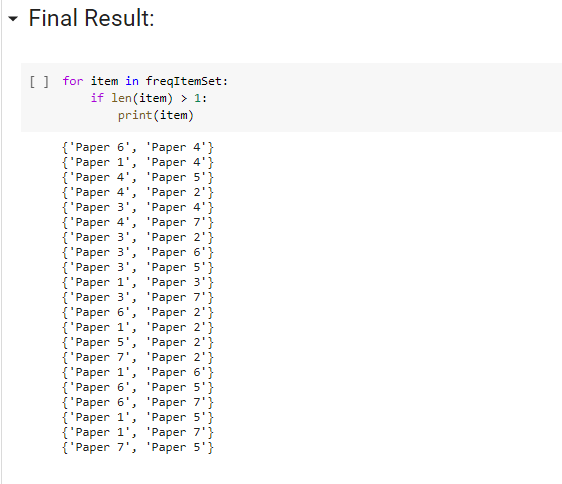
Fig 6.1 shows the result of the Collaborative Filtering Algorithm. The algorithm gives the top course recommendations based on the courses already selected.

****

**Fig 6.1: Collaborative filtering Result**

**5.6.2 FP Growth Algorithm Result:**

Fig 6.2 shows the result of the FP growth Algorithm when ‘Paper 3’ and ‘Paper 4’ from the sample data were given as the input. The algorithm generates frequently occurring itemsets as recommendations to the user.

****

**Fig 6.2: FP Growth Algorithm Result**

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

Our software provides the students with an option to take guidance from their seniors and peers in an indirect way by looking at the data. We hope our system helps students and helps alleviate the confusion students have while choosing the ever increasing number of college electives.

The system does the calculation and gives out a set of its best recommendations so that the student can have more choices to choose his/her electives. The number of total papers and the number of electives the student is interested in has also been taken as a direct input hence making the software more robust and flexible. We have given an option to students for whom the marks matter as well by accepting a user threshold value for the subject even to be considered.

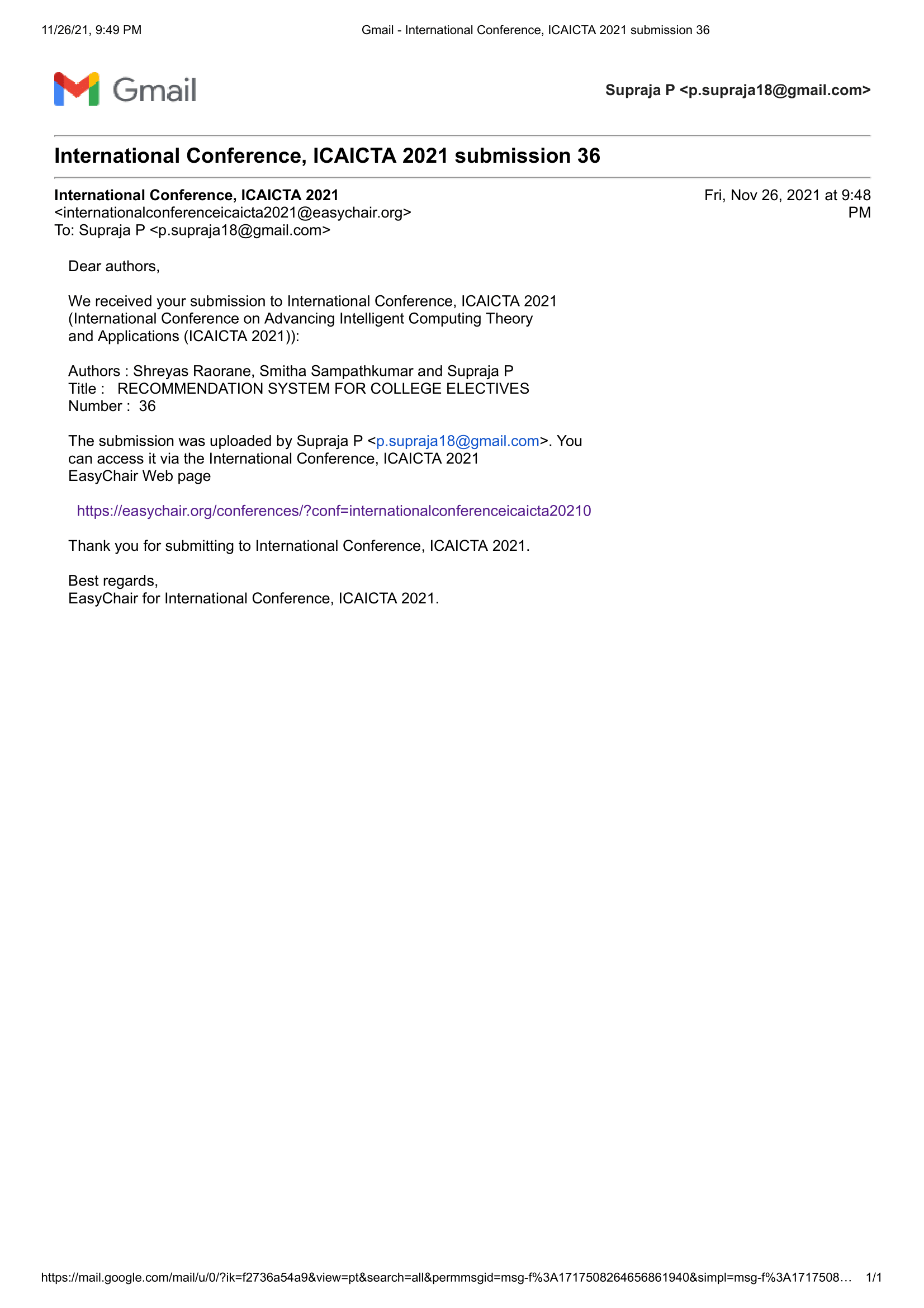
The model will help students make decisions practically as well as passionately while keeping a perfect balance between them.

The future scopes include testing the model with actual data from the colleges and also hosting it on a website. If done so, it'll be a much more friendly user experience. Our future work also includes keeping the numerical data as such and calculation of the score based on the marks rather than changing it to categorical data.

This system is robust enough to be expanded into other domains such as business management and even stocks.

**PAPER PUBLICATION STATUS**

Paper status remains communicated.



**REFERENCES**

[1] Christian Borgelt,Recursion Pruning for the Apriori Algorithm, Department of Knowledge Processing and Language Engineering School of Computer Science, Otto-von-Guericke-University of Magdeburg.

[2] Xiaoyu Tang, Yue Xu, Shlomo Geva, Refining User and Item Profiles based on Multidimensional Data for Top-N Item Recommendation, iiWAS '14: Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services, December 2014.

[3]Xiaofeng Li and Dong Li, An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy, *Mobile Information Systems, may 2019*

[4] Torres, R. McNee, S.M. Abel, M. Konstan, J.A. and Riedl, J. 2004. “Enhancing Digital Libraries with TechLens”, Proceedings of JCDL’04, pp. 228-236.

[5] Pennock, D.M. Horvitz, E. Lawrence, S. and Giles, L.C.2000. “Collaborative Filtering by Personality Diagnosis:A Hybrid Memory- and Model-Based Approach”, inProceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (San Francisco).

[6] F. Bonchi and B. Goethals. FP-Bonsai: the Art of Growing and Pruning Small FP-trees. Proc. 8th

Pacific-Asia Conference on Knowledge Discovery andData Mining (PAKDD’04, Sydney, Australia),

155–160. Springer-Verlag, Heidelberg, Germany 2004

[7] Zhang, Z., & Li, L. (2010, December). A research paper recommender system based on spreading activation model.In Information Science and Engineering (ICISE), 2010 2nd International Conference on (pp. 928-931). IEEE.

[8] Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). Paper recommender systems: a literature survey. International Journal on Digital Libraries, 17(4), 305-338.

[9] F. Bodon. A Fast Apriori Implementation. Proc. Ist IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL). CEUR Workshop Proceedings 90, Aachen, Germany 2003. hxp://[www.ceur-ws.org/Vol-90/](http://www.ceur-ws.org/Vol-90/)

[10] C. Borgelt. Efficient Implementations of Apriori and Eclat. Proc. Ist IEEE ICDM Workshop on Fre quent Item Set Mining Implementations (FIMI 2003. Melbourne, FL). CEUR Workshop Proceedings 90, Aachen, Germany 2003.

[11] Manoj Kumar, D.K. Yadav, Ankur Singh, Vijay Kr. Gupta, A MOVIE RECOMMENDER SYSTEM: MOVREC, International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 2015,

[12] Kartik Narendra Jain, Praveen Kumar, Tanupriya Choudhary, Movie Recommendation System Hybrid Information Filtering System, Intelligent Computing and Information and Communication, 2018,

[13] Yiu-Kai Ng, MovReC: A Personalized Movie Recommendation System for Children Based on Online Movie Features, iiWAS 2016 Proceedings,

[14] Megha Jain, ALGORITHM FOR RESEARCH PAPER RECOMMENDATION SYSTEM, International Journal of Information Technology and Knowledge Management, July-December 2012, Volume 5, No. 2, pp. 443-445

[15] Bhumika Bhatt, Prof. Permal J Patel, Prof. Hetal Gaudani, A Review Paper on Machine Learning Based Recommendation System, 2014, IJEDR

[16] M Viswa Murali, Vishnu T G, Nancy Victor, A Collaborative Filtering based Recommender

System for Suggesting New Trends in Any Domain of Research, 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)

[17] Avi Rana, K. Deeba, Online Book Recommendation System using Collaborative Filtering (With Jaccard Similarity), 2019, Journal of Physics: Conference Series

[18] Rajpurkar, S., Bhatt, D., Malhotra, P., Rajpurkar, M. S. S., Bhatt, M.D. R., & Malhotra, M. P. (2015). Book recommendation system. International Journal for Innovative Research In Science And

Technology, ISSN: 314–316, 1(11, 1).

[19] Amanvir Kaur, Dr. Gagandeep Jagdev, Analyzing working on FP-Growth Algorithm for Frequent Pattern Mining, International Journal of Computer Science and Network Security (IJCSNS), 2017

[20] Abdullah Saad Almalaise Alghandi, Efficient Implementation of FP Growth Algorithm - Data Mining on Medical Data, International Journal of Computer Science and Network Security (IJCSNS), 2011

[21] Christian Borgelt, An Implementation of FP growth Algorithm, Article - Research gate, 2010

[22] Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, Edward Y, PFP: Parallel FP-growth for Query Recommendation, ACM Conference on Recommender systems, 2008

[23] Chen, YC., Hui, L. & Thaipisutikul, T. A collaborative filtering recommendation system with dynamic time decay. J Supercomput 77, 244–262 (2021).

[24] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithms for Fast Discovery of Association

Rules. Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD’97), 283–296. AAAI Press,

Menlo Park, CA, USA 1997

[25] A. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. Fast Discovery of Association Rules. In: [7], 307–328

[26][https://www.](https://www.researchgate.net/publication/351361530/figure/fig1/AS:1024913271955456@1621369774111/Flowchart-of-FP-growth-algorithm.ppm)researchgate[.net/publication/351361530/figure/fig1/AS:1024913271955456@1621369774111/Flowchart-of-FP-growth-algorithm.ppm](https://www.researchgate.net/publication/351361530/figure/fig1/AS:1024913271955456@1621369774111/Flowchart-of-FP-growth-algorithm.ppm)

[26] [What is a Software Process Model? (visual-paradigm.com)](https://www.visual-paradigm.com/guide/software-development-process/what-is-a-software-process-model/)

[27] [Test case - Wikipedia](https://en.wikipedia.org/wiki/Test_case)

[28] [TEST PLAN: What is, How to Create (with Example) (guru99.com)](https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html#:~:text=%20How%20to%20write%20a%20Test%20Plan%20,procedure%20or%20test%20judgment%20can%20be...%20More%20)

[29] [Types of Software Testing - GeeksforGeeks](https://www.geeksforgeeks.org/types-software-testing/)

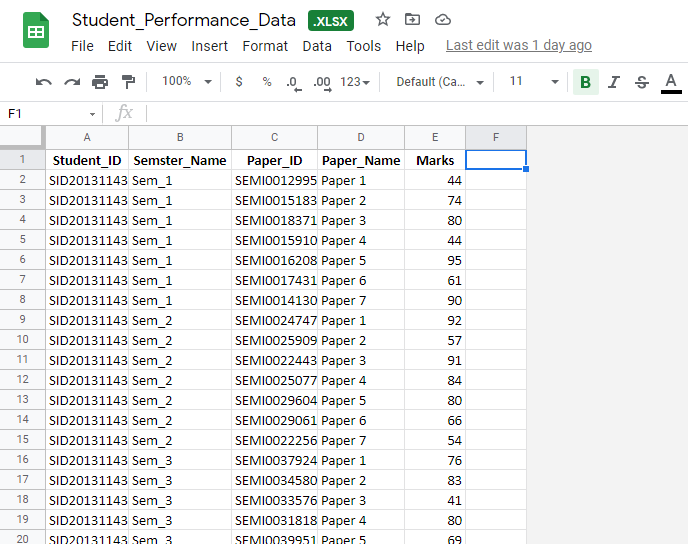
**APPENDIX**

**Dataset:**

Name: Student\_Performance\_Data.xlsx

Link: <https://docs.google.com/spreadsheets/d/1orEAz06p-fMU86s3QMJ86Xwo1EK-ppFT/edit?usp=sharing&ouid=104216667892731766164&rtpof=true&sd=true>

Image:

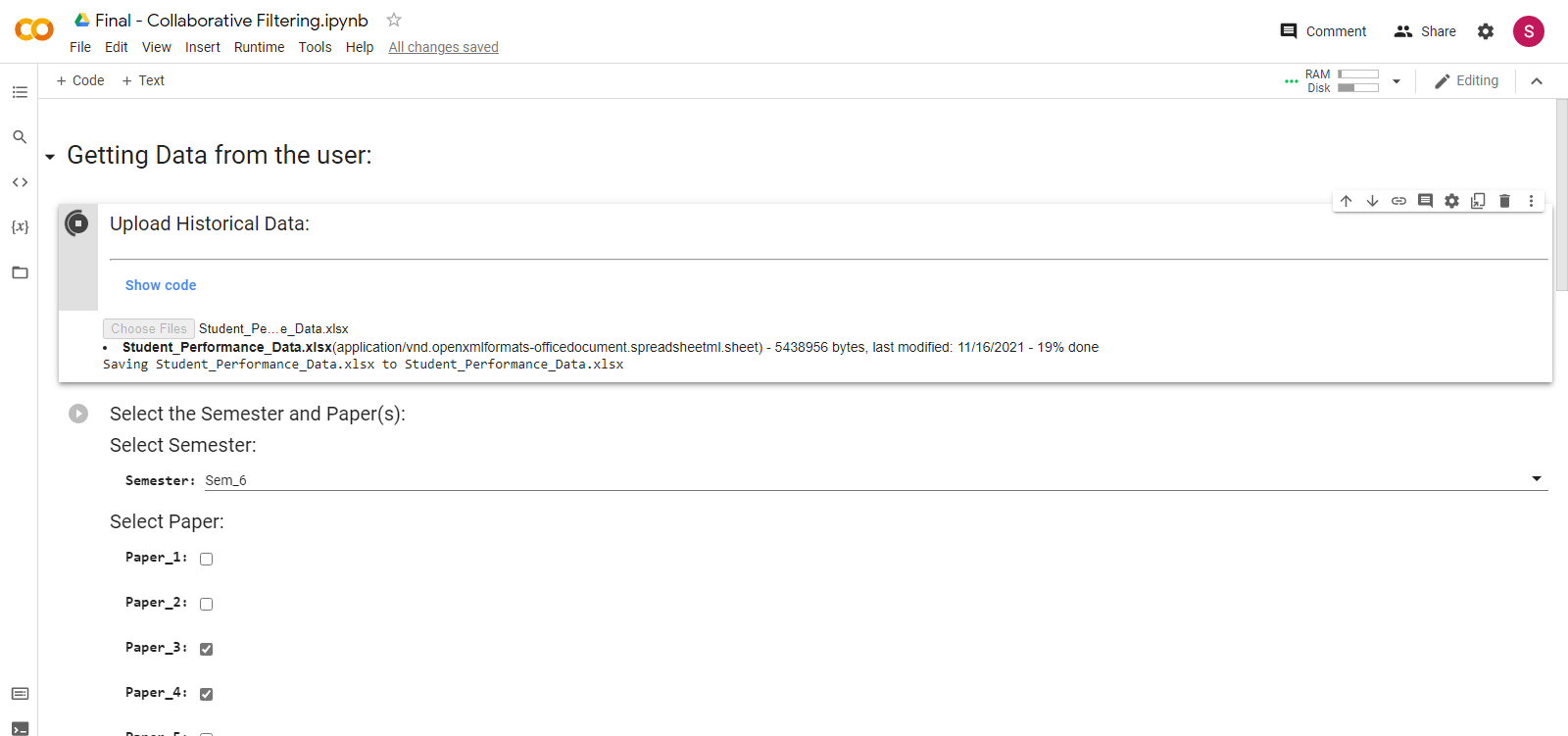


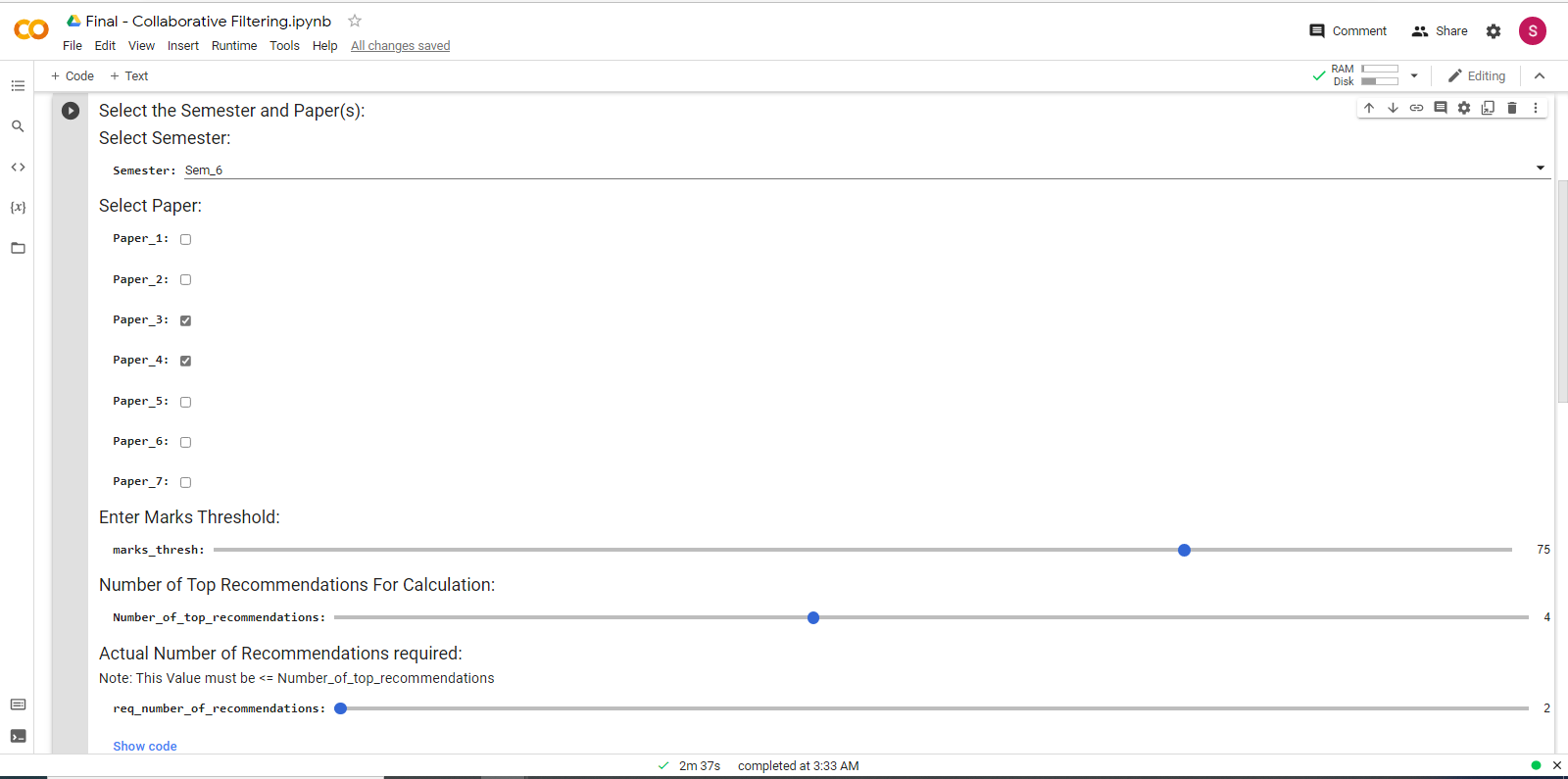
**Collaborative Filtering:**

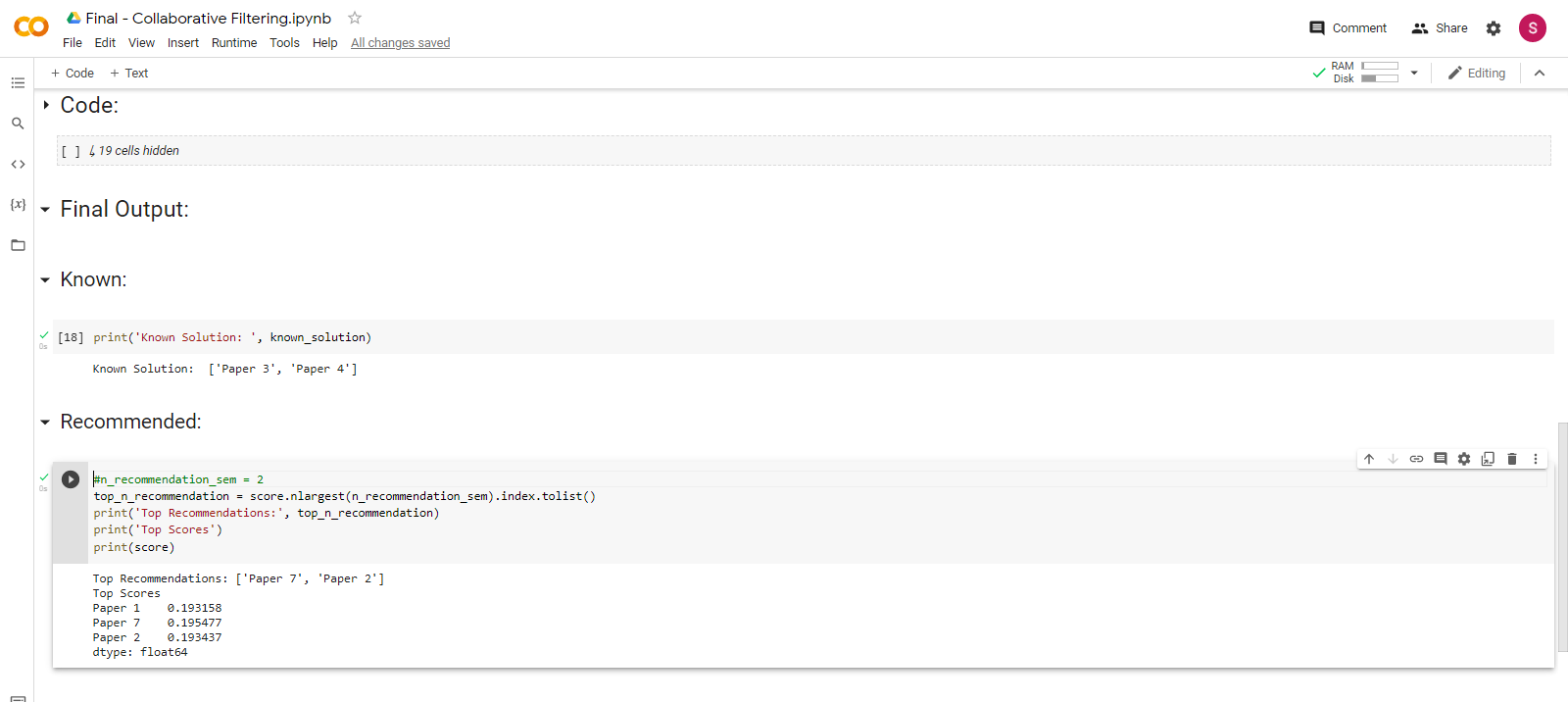
Name: Final - Collaborative Filtering.ipynb

Link: <https://colab.research.google.com/drive/1wLql6Vy_0g2LGPLRPOl3uCce2vvm5jlY?usp=sharing>

Images:







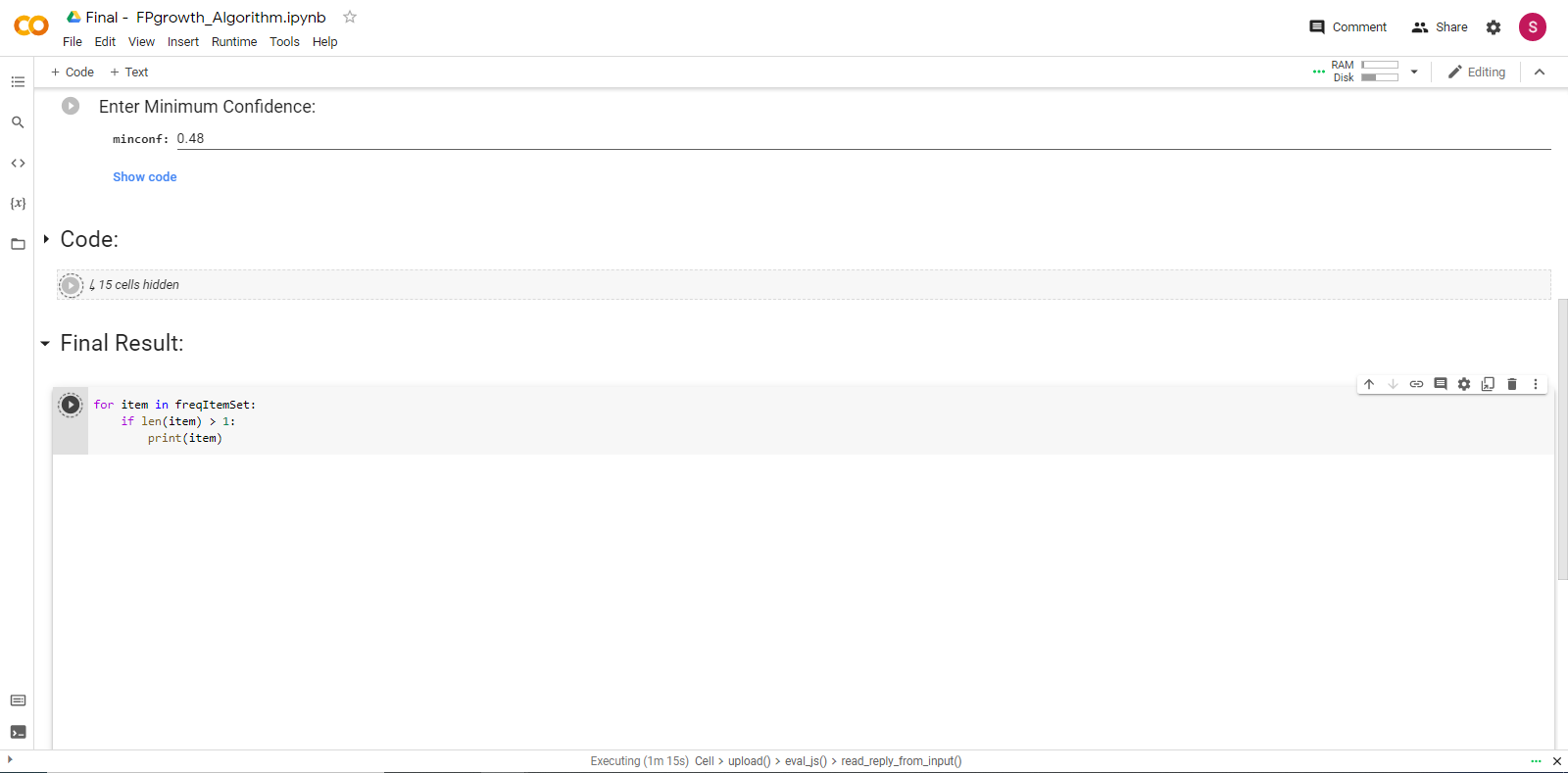
**FP Growth Algorithm:**

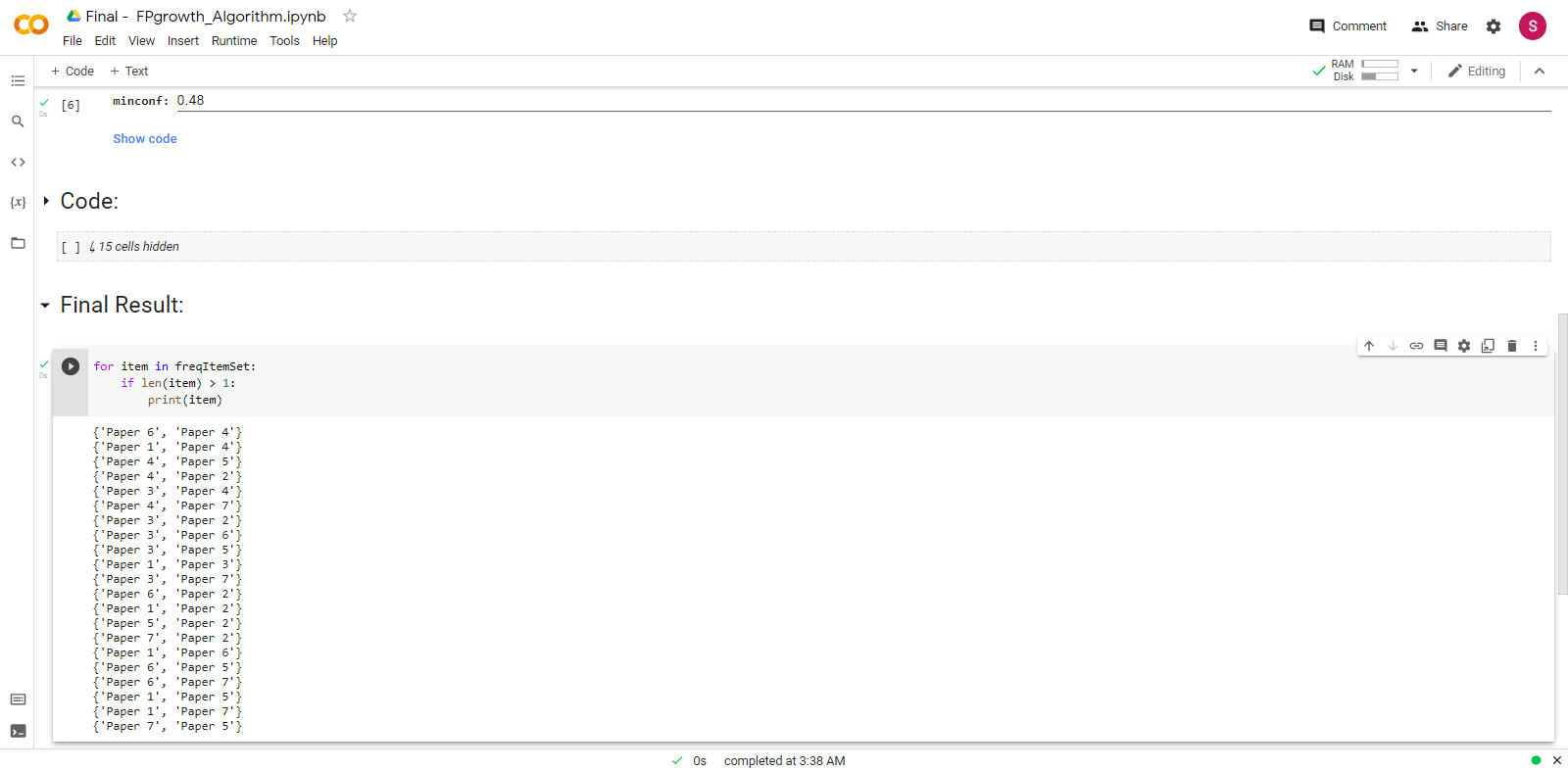
Name: Final - Collaborative Filtering.ipynb

Link:<https://colab.research.google.com/drive/19MevHnhmR-lWYiFFHgXMoDI8uXyJ3XLP?usp=sharing>

Images:







**PLAGIARISM REPORT**