ELECTIVE SUBJECT RECOMMENDATION SYSTEM

A PROJECT REPORT

Submitted By

ANIKET BHARTI
(2100290140027)
SUSHMITA SINGH
(2100290140157)

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Mr. PRASHANT AGRAWAL
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(FEBRUARY 2023)

CERTIFICATE

Certified that **Aniket Bharti (2100290140027), Sushmita Singh** (**2100290140140)** have carried out the project work having "**Elective Subject Recommendation System**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Aniket Bharti (2100290140027)
Sushmita Singh (2100290140140)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Mr. Prashant Agrawal
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

This project is mainly concerned with the choice made by students of the University to choose their Elective Subjects. There are many difficulties faced by students while selecting like what subject they need to choose, how they will perform in the chosen subject, and how similar that subject will be in reference to the other subject already studied by the student. In this paper, we tried to implement the recommendation system for elective subjects. The recommendation is made on the previous subject performance of the student, where we will choose only those subjects where the student performed well and suggest similar subjects to those subjects. For making recommendations we used the KNN working model with different algorithms for calculating distance like "hamming distance," "angular," "euclidean distance" and "manhattan distance" and made comparisons on them. For the dataset, we use two different versions of the dataset with 2 variations for each dataset. The first dataset divides all the subjects into different knowledge domains and the other dataset takes high school and intermediate marks as a basis of all the other subjects two variations are one-hot and percentage of presence for the feature of the subject. We observe recommendations over different sets of choice of subjects and compare them with each other. If two algorithms give the same recommendation for a set of subjects, we are sure that this is a good recommendation.

ACKNOWLEDMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, Dr. Prashant Agrawal for his guidance, help, and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor, and Head, of the Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other help. Without their support, the completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Aniket Bharti

Sushmita Singh

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

This project discusses the problem of subject selection and a recommendation system that may help students and faculty to find a suitable subject selection. In this scenario, students always look for those adepts at providing the proper subject selection, and they also think about whether the decision is correct. The recommendation system will focus on students' previous marks like high school marks, intermediate marks, and the previous semester's marks and a survey based on their liking of different knowledge domains of different subjects. The recommendation is made on the previous subject performance of the student, where we will choose only those subjects where the student performed well and suggest similar subjects to those subjects. For making recommendations we used the KNN working model with different algorithms for calculating distance like “hamming distance,” “angular,” “euclidean distance” and “manhattan distance” and made comparisons on them. For the dataset, we use two different versions of the dataset with 2 variations for each dataset. Based on all factors this project will recommend subjects that students can select to score good marks. In this paper, the proposed method will offer students to select the most suitable subject for opting for the elective subject and select the subject wisely rather than blindly.

1.2 OBJECTIVE

The objective of an elective subject’s recommendation system is to provide students with personalized recommendations for elective courses that suit their individual interests, skills, and academic goals. The system should be able to analyze a student’s academic background, preferences, and personal goals in order to generate a personalized list of elective courses that are best suited for their needs. The system should also provide students with information about the courses, such as course descriptions, faculty profiles, and requirements, in order to help them make informed decisions about their educational path.

The main objective of this paper is to recommend subjects that students can select to score good marks using KNN over different distance measurement schemes and two different versions of the dataset with 2 variations for each dataset.

1.3 AIM

Its aim is to collect the marks of students' datasets, apply hamming distance, angular, euclidean distance, and manhattan distance, and find scores of subjects based on the dataset prepared of different subjects and calculate their distance according to different knowledge domains.

1.4 SOFTWARE REQUIREMENT

Number	**Description**	**Type**
1	Operating System	Windows
2	Language	Python, HTML, CSS, JavaScript
3	Application	Postman, Jupyter Notebook
4	Browser	Google Chrome

1.5 HARDWARE REQUIREMENT

Number	**Description**	**Type**
1	Hardware	Processor Intel dual-core and above XP / Windows
2	Clock speed	3.0 GHz
3	RAM size	512 MB
4	Hard Disk capacity	400 GB

CHAPTER 2

LITERATURE REVIEW

2.1 OVERVIEW

Recommendation systems have become an integral part of modern-day e-commerce, social media, and content-based websites. These systems help users discover new products, services, or content that they may find useful based on their previous behavior or preferences. In this literature review, we will explore various approaches to building recommendation systems, such as collaborative filtering, content-based filtering, and hybrid approaches.

Collaborative filtering is a widely used technique in recommendation systems. It involves analyzing the behavior of multiple users and generating recommendations based on the similarities between their behaviors. This technique can be divided into two categories: user-based and item-based. User-based collaborative filtering recommends items based on the preferences of similar users, while item-based collaborative filtering recommends items based on the similarity between their attributes.

Content-based filtering, on the other hand, recommends items based on the similarity between their features and attributes. This technique uses machine learning algorithms to analyze user preferences and the features of items to make recommendations. For example, a content-based recommendation system for movies might recommend movies with similar genres or actors based on a user's viewing history.

Hybrid approaches combine the strengths of both collaborative and content-based filtering techniques to provide more accurate recommendations. These systems use a combination of collaborative filtering and content-based filtering to generate recommendations that are tailored to the user's preferences.

There are also various challenges in building recommendation systems, such as cold-start problems, scalability, and privacy concerns. The cold-start problem refers to the difficulty of making accurate recommendations for new users or new items that have limited or no user data. Scalability issues arise when dealing with large datasets, where traditional techniques may not be effective. Privacy concerns arise when user data is collected and analyzed to make recommendations, and users may not want their data to be shared or used for this purpose.

To address these challenges, researchers have proposed various solutions such as using deep learning techniques, incorporating context-awareness into the recommendation system, and using privacy-preserving techniques such as differential privacy.

In conclusion, recommendation systems have become an essential tool for providing personalized recommendations to users based on their preferences and behaviors. Collaborative filtering, content-based filtering, and hybrid approaches are the most widely used techniques for building these systems. However, there are various challenges that need to be addressed, such as cold-start problems, scalability, and privacy concerns. Further research is needed to address these challenges and improve the effectiveness and accuracy of recommendation systems.

2.2 LITERATURE SURVEY

YouTube is a very large video platform that users use for an equally large variety of reasons. YouTube uses a recommendation system to help each user pick videos that suits their requirements. However, filtering these videos is not an easy task, and the recommendation system may make mistakes. This paper discusses what problems the recommendation system has and how dire these problems are through a survey in order to understand the advantages and disadvantages of the current recommendation system. Out of 59 participants, 46 of them find the recommendation system is acceptable, good or excellent, but still points out several problems. Between the 24 creators participating in the survey, the recommendation system is still acceptable, but inclined much more towards being poor. We concluded that there is a lack of clarity on how the recommendation system works. This characteristic disliked mainly by creators because they are unable to direct their works to the prospective viewers.[1]

Matrix factorization (MF) is a prevailing technique in recommendation systems (RSs). Since MF needs to process a large amount of user data when generating recommendation results, privacy protection is increasingly being valued by users. Many existing privacy-preserving MF schemes only protect users’ rating values, but ignore the privacy preservation of item sets rated by users. To make up for this shortcoming, a strategy based on piecewise mechanism (PM) is specially designed to simultaneously protect the privacy of rating values and item sets rated by users. To utilize data effectively, an improved MF based on PM (IMFPM) is proposed by dividing item profiles into global and personal information. Furthermore, in the IMFPM, random projection technology is used to reduce the influence of privacy noise on the estimation error. Theoretical analysis and experiment results show that the IMFPM not only provides strong differential privacy protection for rating values and item sets rated by users, but also has high prediction quality. Thus, the IMFPM is a good candidate scheme with privacy preservation for distributed recommendation systems.[2]

The recommendation system is fundamental technology of the internet industry intended to solve the information overload problem in the big data era. Top-k recommendation is an important task in this field. It generally functions through the comparison of positive pairs and negative pairs based on Bayesian personalized ranking (BPR) loss. We find that the contrastive loss (CL) function used in contrastive learning is well-suited for top-k recommendation. However, there are two problems in the existing loss functions. First, all samples are treated the same, and hard samples are not considered. Second, all nonpositive samples are considered negative samples, which ignores the fact that they are unlabeled data containing items that users may like. Moreover, in our experiments, we find that when items are sorted by their similarities to the user, many negative items (or samples) appear before the positive items. We regard these negative items as hard samples and those at the top as potentially positive samples due to their high level of similarities with users. Therefore, we propose a ranking-based contrastive loss (RCL) function to exploit both hard samples and potentially positive samples. Experimental results demonstrate the effectiveness, broad applicability, and high training efficiency of the proposed RCL function.[3]

Universities have been focusing on increasing individualized training and providing appropriate education for students. The individual differences and learning needs of college students should be given enough attention. From the perspective of learning efficiency, we establish a clustering hierarchical progressive improvement model (CHPI), which is based on cluster analysis and context-dependent data envelopment analysis (DEA) methods. The CHPI clusters students' ontological features, employs the context-dependent DEA method to stratify students of different classes, and calculates measures, such as obstacles, to determine the reference path for individuals with inefficient learning processes. The learning strategies are determined according to the gap between the inefficient individual to be improved and the individuals on the reference path. By the study of college English courses as an example, it is found that the CHPI can accurately recommend targeted learning strategies to satisfy the individual needs of college students so that the learning of individuals with inefficient learning processes in a certain stage can be effectively improved. In addition, CHPI can provide specific, efficient suggestions to improve learning efficiency comparing to existing recommendation systems and has great potential in promoting the integration of education-related research and expert systems.[4]

Biodegradable starch films are promising as primary food packaging, and the k-Nearest Neighbor (KNN) algorithm enables selection and classification according to pre-established parameters. Here, the KNN algorithm and principal component analysis prove to be useful tools for sorting and selecting biodegradable starch packaging. Twelve biodegradable films were produced using starch from different botanical sources by the casting method. The KNN analysis evaluated data on thickness, water vapor permeability, tensile strength, elongation, water activity, transparency, and opacity, to obtain an information bank with 36 samples. Biodegradable films are visually homogeneous, transparent, without deformation, and easy to handle. The formulation (Cassava 5%) was classified as the best film, with WVP 1.21 × 10^{-10} (g. m^{-1}. s^{-1}. Pa^{-1}), TS 2.34 (MPa), thickness 0.193 (mm), Aw 0.408, transparency 0.55 and opacity 0.63. The KNN algorithm and principal component analysis are advanced tools for classifying and selecting biodegradable starch films.[5]

Among the most popular collaborative filtering algorithms are methods based on the nearest neighbors (KNN). In their basic operation, KNN methods consider a fixed number of neighbors to make recommendations. However, it is not easy to choose an appropriate number of neighbors. Thus, it is generally fixed by calibration to avoid inappropriate values which would negatively affect the accuracy of the recommendations.

In the literature, some authors have addressed the problem of dynamically finding an appropriate number of neighbors. But they use additional parameters which limit their proposals because these parameters also require calibration. In this paper, we propose a parameter-free KNN method for rating prediction. It is able to dynamically select an appropriate number of neighbors to use. The experiments that we did on four publicly available datasets demonstrate the efficiency of our proposal. It rivals those of the state of the art in their best configurations.[6]

Business-to-business (B2B) social media efforts have largely focused on creating brand engagement through online content. We propose to analyse company social media texts (tweets) according to its two main dimensions, content and linguistic style, and to evaluate these in comparison to the overall content and style of the company's community of Twitter followers. We combine 15 million tweets originating from 254,884 followers of ten company profiles and link these to 10,589 B2B company tweets. Using advanced text analytics, we show that content similarity has positive effects on all engagement metrics, while linguistic style similarity mainly affects likes. Readability acts as a moderator for these effects. We also find a negative interaction effect between the similarity metrics, such that style similarity is most useful if content similarity is low. This research is the first to integrate content and linguistic style similarity and contributes to the brand engagement literature by providing practical message composition guidelines, informed by the social media community.[7]

Clarifying the mechanisms governing volumetric soil water content (VSWC) dynamics in soil profiles is essential, as it can help to elucidate soil water transport processes and improve the prediction accuracy of soil hydrological processes. Using Spearman's rank correlation and wavelet coherence analysis methods, similarity in soil profile VSWC dynamics and factors governing VSWC soil profile dynamics in upslopes and downslopes under three vegetation types (evergreen forest [EG], secondary deciduous forest mixed with shrubs [SDFS], and deforested pasture [DP]) at different time scales (hourly, daily, weekly, and monthly) and in different seasons were analyzed. The results revealed significant similarity in the VSWC of different soil depths ($P < 0.01$), with the similarity decreasing in accordance with the increment in soil depth. Greater VSWC similarity was found in EG than SDFS and DP sites and in upslope than downslope areas at both forest sites. The average significant coherence area (SCA) of VSWC similarity among surface and deep soil layers varied with the time scale, which was in the order of monthly (58.6%) > weekly (42.8%) > daily (21.8%). The effects of soil properties (e.g., texture, saturated hydraulic conductivity), rainfall, and potential evapotranspiration (ET_p) on VSWC similarity were related to the time scale and season in which VSWC monitoring took place. Soil properties had apparent effects on VSWC similarity at longer time scales (i.e., monthly), with a high SCA. In contrast, the effects of rainfall and ET_p on VSWC similarity were concentrated at weekly and daily scales, with a relatively low SCA. Rainfall and ET_p dominated VSWC dynamics in the summer and fall, respectively. These results imply the use of measured VSWC at one soil depth to predict the VSWC at other soil depths was a reliable method. While the influence of time scale effects and seasonal variations on prediction accuracy of VSWC should be considered.[8]

Similarity caching systems have recently attracted the attention of the scientific community, as they can be profitably used in many application contexts, like multimedia retrieval, advertising, object recognition, recommender systems and online content-match applications. In such systems, a user request for an object, which is not in the cache, can be (partially) satisfied by a similar stored object, at the cost of a loss of user utility. In this paper we make a first step into the novel area of similarity caching networks, where requests can be forwarded along a path of caches to get the best efficiency–accuracy tradeoff. The offline problem of content placement can be easily shown to be NP-hard, while different polynomial algorithms can be devised to approach the optimal solution in discrete cases. As the content space grows large, we propose a continuous problem formulation whose solution exhibits a simple structure in a class of tree topologies. We verify our findings using synthetic and realistic request traces.[9]

Formal Concept Analysis (FCA) is revealing interesting in supporting difficult activities that are becoming fundamental in the development of the Semantic Web. Assessing *concept similarity* is one of such activities since it allows the identification of different concepts that are semantically close. In this paper, a method for measuring the similarity of FCA concepts is presented, which is a refinement of a previous proposal of the author. The refinement consists in determining the similarity of concept descriptors (attributes) by using the *information content* approach, rather than relying on human domain expertise. The information content approach which has been adopted allows a higher correlation with human judgement than other proposals for evaluating concept similarity in a taxonomy defined in the literature.[10]

Interdependence among financial return series primarily originate from correlation between underlying assets. However, correlation fully describes interdependence only if the financial system behaves linearly and if an assumption of multivariate normal distribution additionally holds true. At the same time, with intrinsic z score normalization, correlation ignores means (expected return) and variances (risk) when calibrating the interdependence. Such oversight raises the significant question of whether security return networks can be realistically modelled and interpreted by market correlations. This paper proposes the Euclidean (dis)similarity metric which enables incorporation of risk and return along with the primary correlation component. We apply this metric to explain the collective behavior of the MSCI world market and compare the results with other correlation networks. Findings show that realized volatility accounts for 71% of the observed topology whereas correlation explains only 29% of market structure. No evidence was found supporting the importance of expected return. Power law exponents and degree distributions reveal that the centrality of hub nodes are considerably higher in the Euclidean as opposed to correlation networks. Accordingly, the importance and influence of central countries (like US and Japan hubs) in the spreading of high volatility is considerably higher than what correlation networks report.[11]

Increasing sugar levels in the body which exceeds the normal limit is a metabolic disease commonly called diabetes mellitus. Long-term diabetes mellitus is one of the causes of other diseases such as liver, heart and other body organs. Early diagnosis of diabetes mellitus in a person is very important to know earlier. Early diagnosis is made to prevent other diseases to reduce the occurrence of complications in the body.

The use of existing cases can be compared to new cases to diagnose whether the patient has diabetes.[12]

In this work an unsupervised fuzzy learning method for the identification of nonlinear dynamical systems is designed. Accordingly, the learning process is featured by an incremental fuzzy clustering algorithm involving, in addition to the usual Euclidian distance, a new angular deviation. It turns out that: (i) the domain associated to each local model is better located compared to methods based on only Euclidian distance; (ii) the concentration phenomenon, observed when using standard metric classification, is highly reduced. These futures are confirmed by simulation.[13]

Mutation point extraction in acoustic emission (AE) signals is always a complex and challenging task since the mutation is often hidden in AE signals and with a short duration. This paper proposes a novel method based on cosine similarity (CS) to detect change points in AE signals. Disregarding the specific value of AE signals, the proposed method extracts the similarity features from the adjacent waveforms. Compared with traditional AE analysis and state-of-art methods, the proposed method performs better for extracting mutation points in an AE monitoring laser scanning experiment. A combination of a de-negative step and a linear normalization step is applied in the preprocessing procedure to efficiently eliminate the oscillation and vibration in CS calculation. Key parameters (window length and sampling frequency) are demonstrated to affect the mutation points extracting accuracy. The proposed

CS method provides an alternative for mutation extraction in AE signals and can be used in other practical applications.[14]

Currently, coal mining faces the uncertainty of the risk of coal and gas outbursts and inaccurate prediction results. Owing to this, an artificial immune algorithm (AIA) was developed for coal and gas outburst prediction based on the Hamming distance (HD) calculation method of antibody and antigen affinity called the Hamming distance artificial intelligence algorithm (HDAIA). The correlation matrix of coal and gas outburst indicators was constructed using the interpolation function in the algorithm. The HD algorithm was used to obtain the affinity between the antibody and antigen, and the minimum HD was screened to obtain the prediction result. The collected dynamic data of the drilling cuttings gas desorption index K_1 and the drilling cuttings weight S during the excavation process of the 11192-working face of a coal mine in Guizhou Province, China, were used as prediction indices. The results indicate that the prediction result of the HDAIA for the risk of coal and gas outbursts is consistent with the actual risk of outbursts, and it has a good prediction of the risk of coal and gas outbursts. The HDAIA can be used as a novel method for predicting the risk of coal and gas outbursts.[15]

CHAPTER 3

FEASIBILITY STUDY

After studying and analyzing all the existing and requires functionalities of the system, the next task is to do the feasibility study for the project. The feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on future upcoming requirements.

1. Operational Feasibility
2. Technical Feasibility
3. Economic Feasibility
4. Legal Feasibility

3.1 OPERATIONAL FEASIBILITY

The purpose of this study is to determine the operational feasibility of an Elective Subjects Recommendation System. This system has been proposed as a tool to assist students in selecting the most suitable elective subjects in their curriculum. It is expected that the system will provide students with a list of recommended subjects based on their academic performance and other criteria.

The objective of this study is to evaluate the operational feasibility of the proposed system by analyzing the following aspects:

1. System Requirements: The system must be able to collect, process and analyze the data necessary to generate recommendations for students.
2. User Requirements: The system must be intuitive and easy to use for students.
3. Cost Analysis: The cost of developing, implementing and maintaining the system must be taken into consideration.
4. Technical Requirements: The system must be able to handle the necessary data processing and analysis tasks.

In order to assess the operational feasibility of the proposed system, the following steps will be taken:

3.1.1 System Requirements Analysis

The first step is to identify the system requirements to ensure that the proposed system can perform the tasks necessary to generate recommendations for students.

3.1.2 User Requirements Analysis

The second step is to analyze the user requirements to ensure that the system is intuitive and easy to use for students.

3.1.3 Cost Analysis

The third step is to analyze the cost of developing, implementing and maintaining the system.

3.1.4 Technical Requirements Analysis

The fourth step is to analyze the technical requirements to ensure that the system can handle the necessary data processing and analysis tasks.

3.1.5 Final Assessment

The final step is to assess the overall operational feasibility of the proposed system. Based on the analysis of the system requirements, user requirements, cost analysis, and technical requirements, it is concluded that the proposed system is operationally feasible. The system should be able to provide students with the necessary information to make informed decisions about their elective subjects.

3.2 TECHNICAL FEASIBILITY

The technical requirements for an Elective Subjects Recommendation System will depend on the complexity of the system and the technologies used. The system will likely need to be able to access data from multiple sources, as well as store and process this data in an efficient manner. Additionally, the system should be able to make recommendations based on the data. This will require the use of algorithms and machine learning techniques.

3.3 ECONOMICAL FEASIBILITY

The cost of developing an Elective Subjects Recommendation System will depend on the complexity of the system, the size of the development team, and the duration of the project. Depending on these factors, the cost can range from a few hundred dollars to tens of thousands of dollars. It is important to consider the cost of the system against its potential value to the organization.

3.4 LEGAL FEASIBILITY

The legal implications of developing an Elective Subjects Recommendation System should be considered before the project is started. Depending on the type of data the system will access and process, there may be requirements for data security and privacy. Additionally, there may be regulations regarding the use of algorithms and machine learning techniques. It is important to ensure that the system is compliant with all applicable laws.

Overall, the development of an Elective Subjects Recommendation System is a complex undertaking. It requires careful consideration of the cost, time, personnel, technical, and legal requirements. With the right resources and a well-defined plan, the project can be completed successfully and provide value to the organization.

CHAPTER 4

DESIGN

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

4.1 DESIGNING AIM

4.1.1 Gave Rational Decisions

For making recommendations we used the KNN working model with different algorithms for calculating distance like “hamming distance,” “angular,” “euclidean distance” and “manhattan distance” and made comparisons on them. Hence, similar subjects based on domain knowledge come closed and recommendations can be generated.

4.1.2 Initial Advice for further research

The existing system for elective subjects’ recommendation is either by the interest of the student or the advice from teachers and friends, which can be biased by the personal understanding and interest of the suggesting party.

4.1.3 Based on students' previous performance

For making the recommendation for the elective subjects our project uses previous performance on the curriculum subjects and only those subjects are selected where students performed well.

4.2 MODEL USED: INCREMENTAL MODEL

An incremental model is a software development approach where the product is developed in small increments or portions. Each increment adds new features or functionality to the product, and the development team works on these increments in a cyclical manner.

The incremental model is characterized by its iterative and incremental approach, which involves dividing the entire software development process into smaller, more manageable portions. Each iteration involves a series of steps, including planning, requirements gathering, design, implementation, testing, and deployment. The development team works on each increment, and once it is completed, it is tested and delivered to the end-user for feedback.

One of the main advantages of the incremental model is that it allows for greater flexibility and adaptability in the development process. Changes can be made more easily, and feedback from users can be incorporated into subsequent increments. Additionally, the incremental model allows for early delivery of a working product, which can provide benefits to both the development team and the end-user.

Overall, the incremental model is a popular approach in software development, particularly for projects where requirements may be unclear or subject to change. It provides a structured and iterative approach to development, allowing for greater flexibility and adaptability throughout the process.

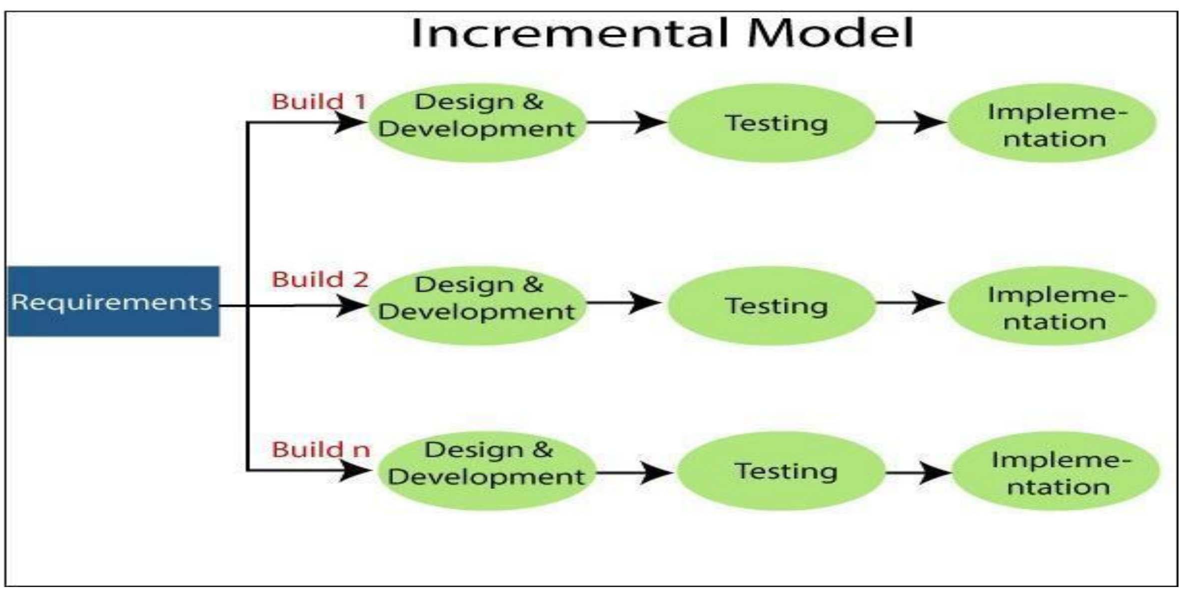

Figure 4.2 Incremental Model.

4.3 GANTT CHART

A Gantt chart is a popular project management tool that is used to schedule and track project activities. It displays the timeline of a project in a visual way, making it easier for project managers and team members to see the progress of the project and ensure that it is on track.

For the development of an elective recommendation system using the iterative model, the Gantt chart would include the various activities required for the project, including the time spent on requirement gathering. Assuming a total development time of 12 weeks, with the first 3 weeks dedicated to requirements gathering, next 4 weeks on designing, next 5 weeks on coding and finally last three week on testing and maintenance, the Gantt chart came up like this:

Figure 4.3 Gantt chart.

4.4 CONTROL FLOW DIAGRAM

Students will enter marks of their past subjects along with subject names/IDs. Marks will be pre-processed to the required formatting so that compared with stored subject embeddings. Unique embedding for the student will be created and compared with all the other embeddings already stored. The similarity will be calculated one by one for all the embeddings stored in the database. Subjects with a higher value will be recommended.

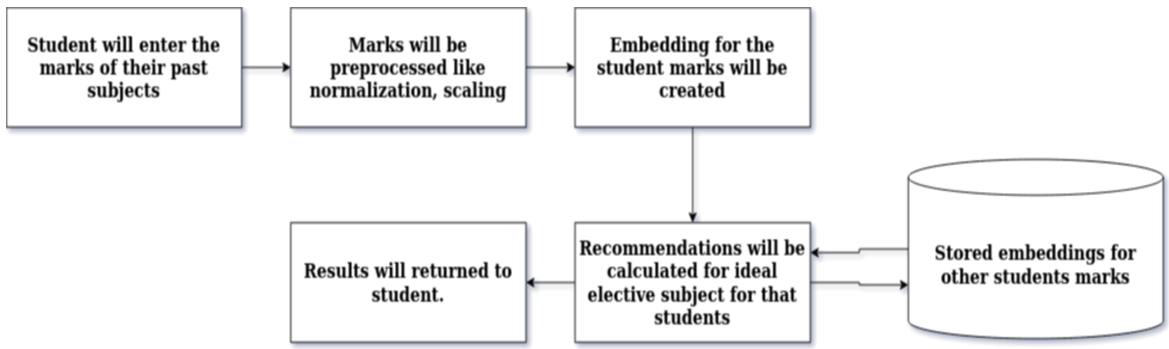

Figure 4.4 Recommendation system control flow diagram.

4.5 Entity Relationship Diagram

This ER Diagram represents the model of the Workplace Coaction System Entity. The Entity Relationship Diagram shows all visual instruments of the Database table and the relation between the student and the system. All of it has Structured data and entities may have some attributes.

Elective Subject Recommendation System: -

1. Student: Previous marks, Fav_Subjects.
2. Marks: High School, Intermediate, Previous Sem.
3. Dataset: Knowledge domains, Subjects, Score.
4. Electives: Subjects, Score.
5. Recommendation: Subjects, Score.

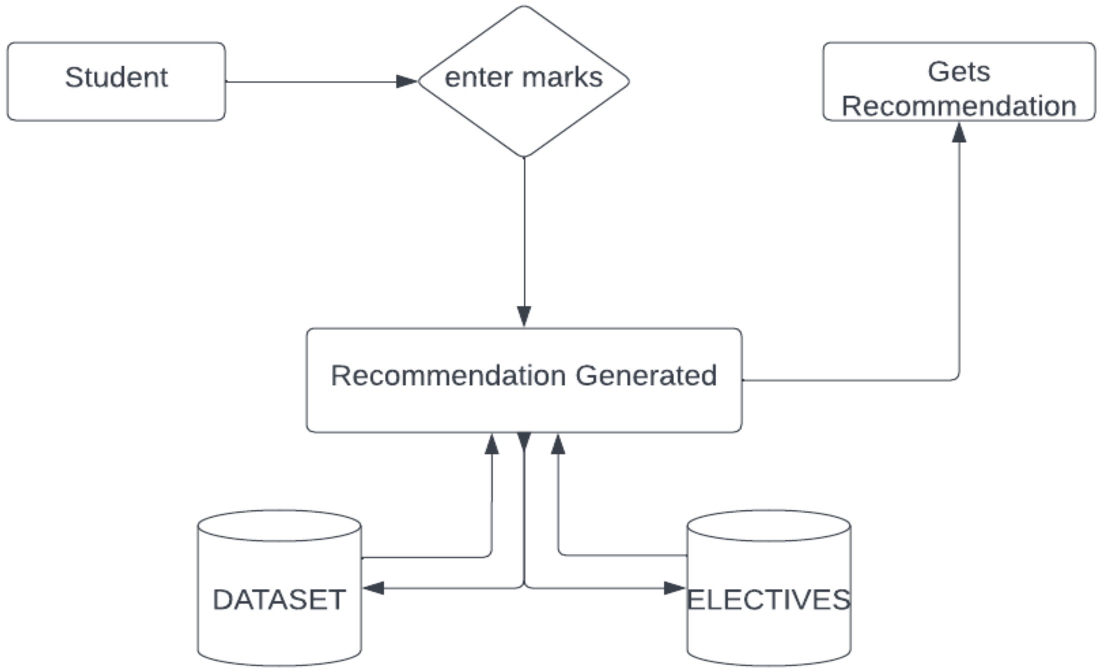

Figure 4.5 Entity Relationship diagram.

4.6 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between actors (such as a student) and a system (such as a grading system). It depicts the various use cases, or functions, of the system from the perspective of its users, and how these functions are related to one another.

In this case, the use case diagram would look something like this:

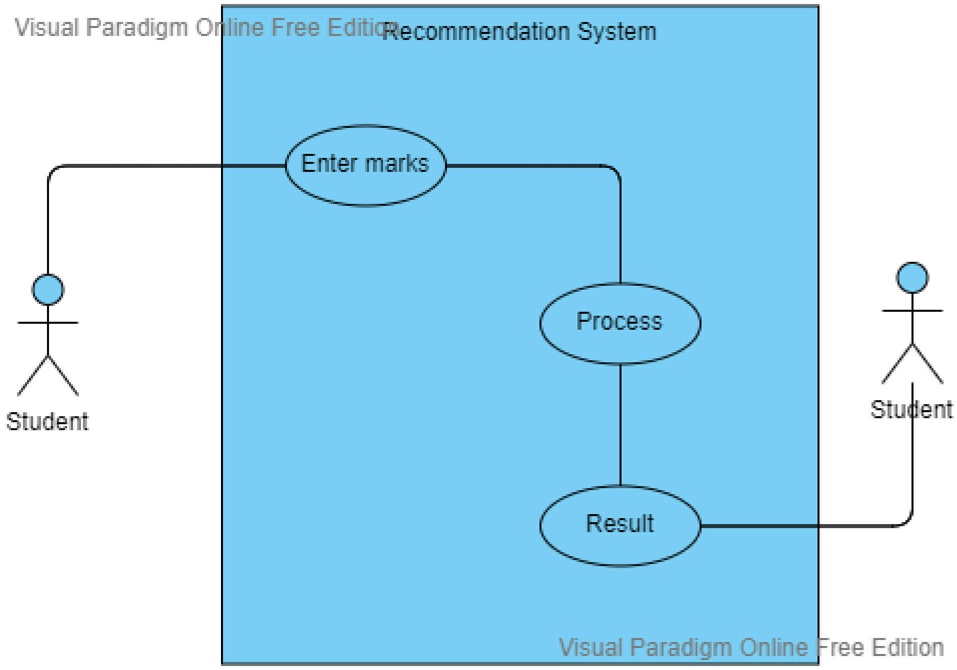

Figure 4.6 Use Case Diagram.

As you can see, there are three use cases depicted in this diagram:

1. "Enter marks" - This use case allows the student to input their grades into the system.
2. "Process" - This use case describes the processing of the student's grades, which could include calculations of averages or determination of pass/fail status.
3. "Recommendation" - This use case provides recommendations to the student based on their performance, such as suggesting areas for improvement or recommending courses to take in the future.

The use case diagram shows that the student is the primary actor in this system, meaning they are the one interacting with it directly. The "enter marks" and "recommendation" use cases are connected to the student actor with arrows, indicating that these functions are initiated by the student. The "process" use case is located in between "enter marks" and "recommendation," indicating that it is the bridge between the two functions and is responsible for performing the necessary calculations and analysis to generate the recommendations.

4.7 SUBJECT DISTRIBUTION DIAGRAM

We present our recent work on creating a subject distribution plot over a knowledge domain and applying the t-SNE algorithm to reduce the dimensionality of the plot to two dimensions.

Our subject distribution plot provides an overview of how various subjects are distributed across a particular knowledge domain. This helped us to identify areas of potential overlap or interdisciplinary collaboration and gain a better understanding of how different areas of study are related to one another.

To further enhance the interpretability of our plot, we utilized the t-SNE algorithm, which is a machine learning technique used for dimensionality reduction. By reducing the dimensionality of the plot to two dimensions, it became much easier to visualize and interpret the relationships between the subjects within the knowledge domain.

Overall, we are thrilled with the results of our work. The subject distribution plot, combined with the dimensionality reduction provided by t-SNE, has produced a visually appealing and informative representation of the distribution of subjects within the knowledge domain. We believe that this work has the potential to benefit many researchers and educators in the field.

In conclusion, our work on creating a subject distribution plot and applying the t-SNE algorithm for dimensionality reduction has provided valuable insights into the distribution of subjects within a knowledge domain. We hope that this work will inspire further research and collaboration in this area.

Figure 4.7 Subject Distribution Diagram.

4.8 SUBJECT CROSS SIMILARITY DIAGRAM

Our subject distribution plot provides an overview of how various subjects are distributed across a particular knowledge domain. This helped us to identify areas of potential overlap or interdisciplinary collaboration and gain a better understanding of how different areas of study are related to one another.

To further enhance the interpretability of our plot, we utilized the t-SNE algorithm, which is a machine learning technique used for dimensionality reduction. By reducing the dimensionality of the plot to two dimensions, it became much easier to visualize and interpret the relationships between the subjects within the knowledge domain.

Overall, we are thrilled with the results of our work. The subject distribution plot, combined with the dimensionality reduction provided by t-SNE, has produced a visually appealing and informative representation of the distribution of subjects within the knowledge domain. We believe that this work has the potential to benefit many researchers and educators in the field.

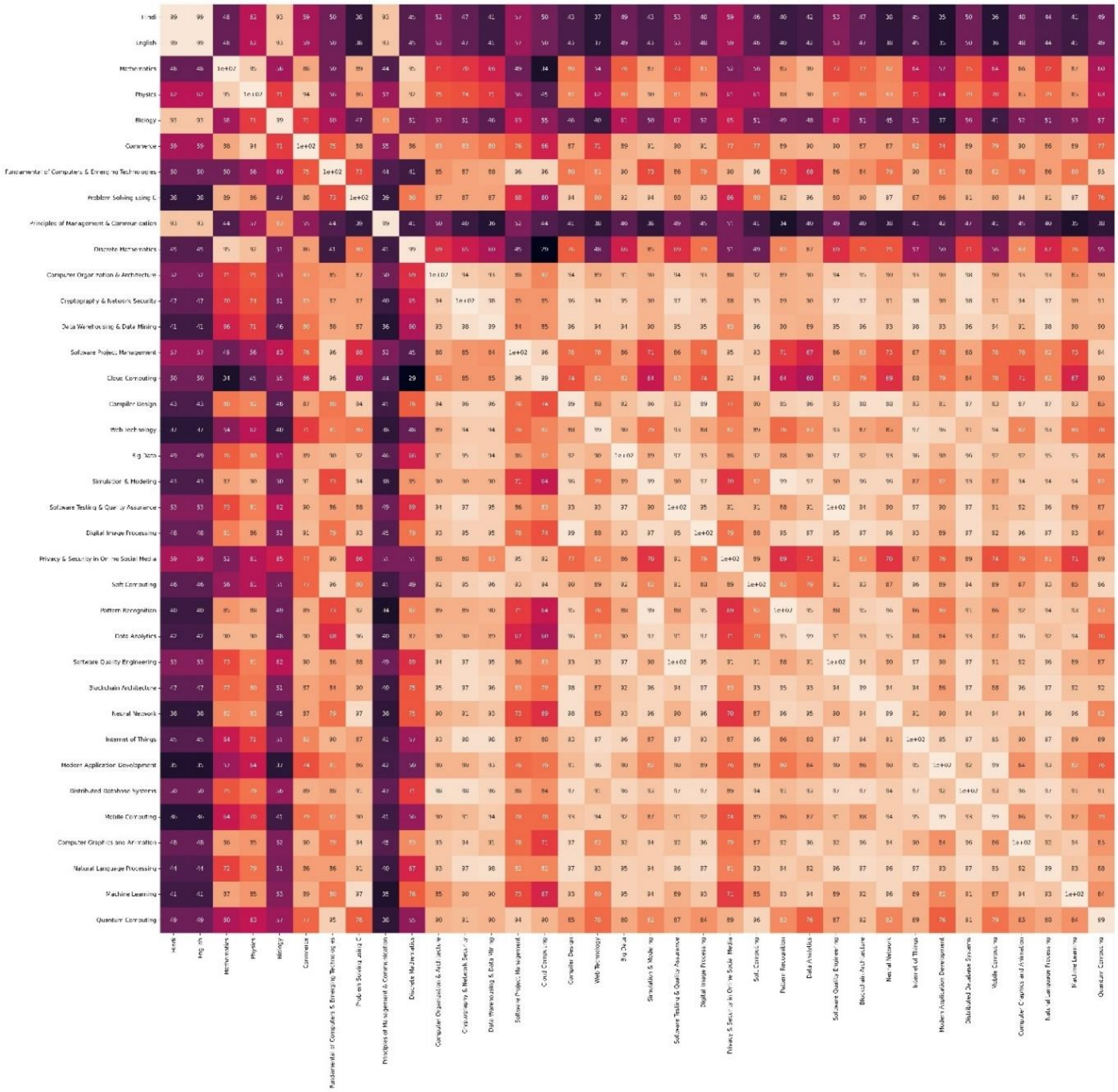

Figure 4.8 Subject Cross Similarity Diagram.

CHAPTER 5

CODING

5.1 FRONT-END

We have developed a frontend interface that enables users to input student information, including marks obtained in high school, intermediate, and graduation.

The frontend interface is designed to be user-friendly and intuitive, with clear and easy-to-use input fields for each type of marks. Users can enter the marks for each subject or course, and the system will automatically calculate the total marks and percentage.

The frontend interface also includes features to help users verify and validate the information they enter. For example, the system may prompt users to confirm their input if the marks entered exceed the maximum possible marks for a particular subject or course.

Overall, our frontend interface is an essential component of the system, enabling users to input and verify student information quickly and accurately. By providing a user-friendly and intuitive interface, we aim to ensure that the system is accessible to a wide range of users, including educators, researchers, and students.

5.1.1 App.js

```
import LinearStepper from "./LinearStepper";
import { Container, Paper, Box } from "@material-
ui/core";
import "./App.css";

function App() {
  return (
    <>
    <div id="back">
      <div>
        <br/>
        <br/>
        <br/>
        <br/>
      </div>
      <Container component={Box} p={4}>
        <Paper component={Box} p={3}>
          <LinearStepper />
```

```
        </Paper>
      </Container>
    </div>
    </>
  );
}

export default App;
```

5.1.2 LinearStepper.css

```css
.box {
    direction: rtl;
    display: flex;
    justify-content: flex-end;
  }
.text{
    font-size: small;
  }
```

5.1.3 LinearStepper.js

```javascript
import React, { useState } from "react";
import {
  Typography,
  TextField,
  Button,
  Stepper,
  Step,
  StepLabel,
} from "@material-ui/core";
import { makeStyles } from "@material-ui/core/styles";
import {
  useForm,
  Controller,
  FormProvider,
  useFormContext,
} from "react-hook-form";
import "./LinearStepper";
import {Card,
  CardBlock,
```

```
  CardFooter,
  CardTitle, } from "react-bootstrap-card";


const useStyles = makeStyles((theme) => ({
  button: {
    marginRight: theme.spacing(1),
  },
}));

function getSteps() {
  return [
    "Enter your high school marks",
    "Enter your intermediate marks",
    "Enter your first semester marks",
    "Enter your first semester marks",
  ];
}
const HighSchoolForm = () => {
  const { control, formState:{ errors }, } =
useFormContext();
  console.log(errors);
  return (
    <>
      <Controller
        control={control}
        name="maths"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="maths"
            label="Mathematics"
            variant="outlined"
            placeholder="Enter maths marks"
            fullWidth
            margin="normal"
            {...field}
            error={Boolean(errors?.maths)}
            helperText={errors.maths?.message}
          />
        )}
      />

      <Controller
```

```
        control={control}
        name="science"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="science"
            label="Science"
            variant="outlined"
            placeholder="Enter science marks"
            fullWidth
            margin="normal"
            {...field}
            error={Boolean(errors?.science)}
            helperText={errors.science?.message}
          />
        )}
      />

      <Controller
        control={control}
        name="eng"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="eng"
            label="English"
            variant="outlined"
            placeholder="Enter english marks"
            fullWidth
            margin="normal"
            {...field}
            error={Boolean(errors?.eng)}
            helperText={errors.eng?.message}
          />
        )}
      />

      <Controller
        control={control}
        name="hin"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="hin"
```

```
            label="Hindi"
            variant="outlined"
            placeholder="Enter hindi marks"
            fullWidth
            margin="normal"
            {...field}
            error={Boolean(errors?.hin)}
            helperText={errors.hin?.message}
          />
        )}
      />

      <Controller
        control={control}
        name="com"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="com"
            label="Computer"
            variant="outlined"
            placeholder="Enter computer marks"
            fullWidth
            margin="normal"
            {...field}
            error={Boolean(errors?.com)}
            helperText={errors.com?.message}
          />
        )}
      />

    </>
  );
};
const InterForm = () => {
  const { control } = useFormContext();
  return (
    <>
      <Controller
        control={control}
        name="math"
        render={({ field }) => (
          <TextField
            id="math"
```

```
          label="Mathematics"
          variant="outlined"
          placeholder="Enter maths marks"
          fullWidth
          margin="normal"
          {...field}
        />
      )}
    />

    <Controller
      control={control}
      name="phy"
      render={({ field }) => (
        <TextField
          id="phy"
          label="Physics"
          variant="outlined"
          placeholder="Enter physics marks"
          fullWidth
          margin="normal"
          {...field}
        />
      )}
    />
    <Controller
      control={control}
      name="en"
      render={({ field }) => (
        <TextField
          id="en"
          label="English"
          variant="outlined"
          placeholder="Enter english marks"
          fullWidth
          margin="normal"
          {...field}
        />
      )}
    />

    <Controller
      control={control}
      name="hind"
```

```
        render={({ field }) => (
          <TextField
            id="hind"
            label="Hindi"
            variant="outlined"
            placeholder="Enter hindi marks"
            fullWidth
            margin="normal"
            {...field}
          />
        )}
      />

      <Controller
        control={control}
        name="comp"
        render={({ field }) => (
          <TextField
            id="comp"
            label="Computer"
            variant="outlined"
            placeholder="Enter computer marks"
            fullWidth
            margin="normal"
            {...field}
          />
        )}
      />
    </>
  );
};
const FirstSem = () => {
  const { control } = useFormContext();
  return (
    <>
      <Controller
        control={control}
        name="fcet"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="fcet"
            label="Fundamental of Computers & Emerging
Technologies"
```

```
            variant="outlined"
            placeholder="Enter fcet marks"
            fullWidth
            margin="normal"
            {...field}
          />
        )}
      />
      <Controller
        control={control}
        name="c"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="c"
            label="Problem Solving using C"
            variant="outlined"
            placeholder="Enter C marks"
            fullWidth
            margin="normal"
            {...field}
          />
        )}
      />
      <Controller
        control={control}
        name="pmc"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="pmc"
            label="Principles of Management &
Communication "
            variant="outlined"
            placeholder="Enter Principles of Management
& Communication marks"
            fullWidth
            margin="normal"
            {...field}
          />
        )}
      />

      <Controller
```

```
        control={control}
        name="dm"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="dm"
            label="Discrete Mathematics "
            variant="outlined"
            placeholder="Enter DM marks"
            fullWidth
            margin="normal"
            {...field}
          />
        )}
      />

      <Controller
        control={control}
        name="coa"
        rules={{ required: "this field is required." }}
        render={({ field }) => (
          <TextField
            id="coa"
            label="Computer Organization & Architecture
"
            variant="outlined"
            placeholder="Enter COA marks"
            fullWidth
            margin="normal"
            {...field}
          />
        )}
      />
    </>
  );
};
const SecondSem = () => {
  const { control } = useFormContext();
  return (
    <>
      <Controller
        control={control}
        name="aut"
        render={({ field }) => (
```

```
            <TextField
              id="aut"
              label="Theory of Automata & Formal
Languages"
              variant="outlined"
              placeholder="Enter Automata marks"
              fullWidth
              margin="normal"
              {...field}
            />
          )}
        />
        <Controller
          control={control}
          name="oop"
          render={({ field }) => (
            <TextField
              id="oop"
              label="Object Oriented Programming"
              variant="outlined"
              placeholder="Enter OOPs marks"
              fullWidth
              margin="normal"
              {...field}
            />
          )}
        />
        <Controller
          control={control}
          name="os"
          render={({ field }) => (
            <TextField
              id="os"
              label="Operating Systems "
              variant="outlined"
              placeholder="Enter Operating Systems marks"
              fullWidth
              margin="normal"
              {...field}
            />
          )}
        />

        <Controller
```

```
          control={control}
          name="dbms"
          render={({ field }) => (
            <TextField
              id="dbms"
              label="Database Management Systems"
              variant="outlined"
              placeholder="Enter DBMS marks"
              fullWidth
              margin="normal"
              {...field}
            />
          )}
        />

        <Controller
          control={control}
          name="ds"
          render={({ field }) => (
            <TextField
              id="ds"
              label="Data Structures & Analysis of
Algorithms"
              variant="outlined"
              placeholder="Enter DSA marks"
              fullWidth
              margin="normal"
              {...field}
            />
          )}
        />

        <Controller
          control={control}
          name="cyb"
          render={({ field }) => (
            <TextField
              id="cyb"
              label="Cyber Security"
              variant="outlined"
              placeholder="Enter Cyber Security marks"
              fullWidth
              margin="normal"
              {...field}
```

```
          />
        )}
      />
    </>
  );
};

function getStepContent(step) {
  switch (step) {
    case 0:
      return <HighSchoolForm />;
    case 1:
      return <InterForm />;
    case 2:
      return <FirstSem />;
    case 3:
      return <SecondSem />;
    default:
      return "unknown step";
  }
}

const LinaerStepper = () => {
  const classes = useStyles();
  const methods = useForm({
    defaultValues: {
      maths: "",
      science: "",
      eng: "",
      hin: "",
      com: "",
      math: "",
      phy: "",
      en: "",
      hind: "",
      comp: "",
      fcet: "",
      c: "",
      pmc: "",
      dm: "",
      coa: "",
      aut: "",
      oop: "",
      os: "",
```

```
      dbms: "",
      ds: "",
      cyb: "",

    },
  });
  const [activeStep, setActiveStep] = useState(0);
  const [skippedSteps, setSkippedSteps] = useState([]);
  const steps = getSteps();

  const isStepOptional = (step) => {
    return step === 1 || step === 3;
  };

  const isStepSkipped = (step) => {
    return skippedSteps.includes(step);
  };

  const handleNext = (data) => {
    console.log(data);
    if (activeStep === steps.length - 1) {

fetch("https://jsonplaceholder.typicode.com/comments")
        .then((data) => data.json())
        .then((res) => {
          console.log(res);
          setActiveStep(activeStep + 1);
        });
    } else {
      setActiveStep(activeStep + 1);
      setSkippedSteps(
        skippedSteps.filter((skipItem) => skipItem !==
activeStep)
      );
    }
  };

  const handleBack = () => {
    setActiveStep(activeStep - 1);
  };

  const handleSkip = () => {
    if (!isStepSkipped(activeStep)) {
      setSkippedSteps([...skippedSteps, activeStep]);
```

```
    }
    setActiveStep(activeStep + 1);
  };

  // const onSubmit = (data) => {
  //   console.log(data);
  // };
  return (
    <div>
      <Stepper alternativeLabel activeStep={activeStep}>
        {steps.map((step, index) => {
          const labelProps = {};
          const stepProps = {};
          if (isStepOptional(index)) {
            labelProps.optional = (
              <Typography
                variant="caption"
                align="center"
                style={{ display: "block" }}
              >
                optional
              </Typography>
            );
          }
          if (isStepSkipped(index)) {
            stepProps.completed = false;
          }
          return (
            <Step {...stepProps} key={index}>
              <StepLabel
{...labelProps}>{step}</StepLabel>
            </Step>
          );
        })}
      </Stepper>

      {activeStep === steps.length ? (
        <Typography variant="h3" align="center">
          Recommendation Generated
          <div class="box">
            <div ></div>
            <div >
            <Card>
              <CardBlock>
```

```
                <CardTitle>
                  Subject:
                </CardTitle>
                  SCORE:
              </CardBlock>
            </Card>
            </div>
            <p class="text" style={{fontSize: "20px",
}}>Note: Less score means highly recommended</p>
          </div>
        </Typography>
      ) : (
        <>
          <FormProvider {...methods}>
            <form
onSubmit={methods.handleSubmit(handleNext)}>
              {getStepContent(activeStep)}

              <Button
                className={classes.button}
                disabled={activeStep === 0}
                onClick={handleBack}
              >
                back
              </Button>
              {isStepOptional(activeStep) && (
                <Button
                  className={classes.button}
                  variant="contained"
                  color="primary"
                  onClick={handleSkip}
                >
                  skip
                </Button>
              )}
              <Button
                className={classes.button}
                variant="contained"
                color="primary"
                // onClick={handleNext}
                type="submit"
              >
                {activeStep === steps.length - 1 ?
"Finish" : "Next"}
```

```
                </Button>
              </form>
            </FormProvider>
          </>
        )}
      </div>
    );
};

export default LinaerStepper;
```

5.1.3 Api.js

```
var axios = require('axios');

var data = JSON.stringify({
  "subjects": [
    "Principles of Management & Communication"
  ],
  "k": 2
});

var getSimilarSubject = (data) => {

    var config = {
        method: 'post',
        url: 'http://127.0.0.1:5000/api/v1/similar-
subject',
        headers: {
          'Content-Type': 'application/json'
        },
        data : data
    };

    axios(config)
    .then(function (response) {
        console.log(JSON.stringify(response.data));
    })
    .catch(function (error) {
        console.log(error);
    });
}
```

```
getSimilarSubject(data);
```

5.1.4 index.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
ReactDOM.render(<App />,
document.getElementById("root"));
```

5.1.5 package.json

```
{
  "name": "stepper",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@emotion/react": "^11.10.5",
    "@emotion/styled": "^11.10.5",
    "@material-ui/core": "^4.12.4",
    "@material-ui/icons": "^4.11.3",
    "@mui/icons-material": "^5.10.9",
    "@mui/material": "^5.10.13",
    "@testing-library/jest-dom": "^5.11.4",
    "@testing-library/react": "^11.1.0",
    "@testing-library/user-event": "^12.1.10",
    "bootstrap-3-card": "^0.2.0",
    "npm-check": "^6.0.1",
    "react": "^17.0.2",
    "react-bootstrap": "^2.6.0",
    "react-bootstrap-card": "^0.2.1",
    "react-dom": "^17.0.2",
    "react-hook-form": "^7.39.2",
    "react-router-dom": "^6.4.3",
    "react-scripts": "4.0.3",
    "web-vitals": "^1.0.1"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
```

```
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

5.2 BACKEND

We have developed a backend using Flask, a popular Python web framework. The backend is responsible for accepting inputs from the user and generating recommendations based on the inputs provided.

The backend accepts a list of subjects, a value for k (the number of recommendations to generate), and a list of representing electives. Each elective in the list represents a set of five subjects from which the student can choose. For example, if the student is interested in pursuing a career in computer science, they may select an elective representing courses in programming, data structures, algorithms, databases, and computer networks.

The backend uses a recommendation algorithm to generate a list of recommended electives based on the user's inputs. The algorithm takes into account the user's selected subjects, the number of electives available, and the difficulty level of each elective. The resulting list of recommendations is returned to the user via the frontend interface.

5.2.1 app.py

```
from flask import Flask, request, jsonify
from flask_cors import CORS
from annoy import AnnoyIndex
import pickle
import os
from collections import defaultdict
from helper import load_elective


app = Flask(__name__)

# Enable CORS for API endpoints
cors = CORS(app, resources={r'/api/*': {'origins':
'*'}})

# Variables
f = 9
K_MAX = 5
ANNOY_ONEHOT = AnnoyIndex(f, 'hamming')
ANNOY_PERCENTAGE = AnnoyIndex(f, 'angular')
ANNOY = None
SUB_TO_IDX = {}
IDX_TO_SUB = {}


@app.route("/api/v1/subject-list", methods=['GET'])
def subject_list():
    '''
    Returns the list of all the subjects in database
    '''
    subjects = SUB_TO_IDX.keys()
    return jsonify({'subjects': list(subjects)})


@app.route("/api/v1/similar-subject", methods=['POST'])
def similar_subject():
    args = parse_args(request.get_json())
    if 'error' in args.keys():
        return jsonify(args)
    recommendation = recommendation_electivewise(args)
    return jsonify(recommendation)
```

```
def parse_args(data):
    subjects = data['subjects']
    # loading index for subject
    try:
        ids = [SUB_TO_IDX[subject] for subject in
subjects]
        k = data['k'] if 'k' in data.keys() else 3
        electives = data['electives'] if 'k' in
data.keys() else [1]
        type = data['type'] if 'type' in data.keys()
else 2
    except KeyError:
        return {"error": "Sorry given subject is not in
database"}
    return {'ids': ids, 'k': min(k, K_MAX), 'electives':
electives, 'type': type}


def recommendation_electivewise(args):
    ids, electives, k = args['ids'], args['electives'],
args['k']
    res = []
    for elective in electives:
        d = {
            "elective_{}".format(elective):
electivedwise_generation(elective, ids, k, args['type'])
        }
        res.append(d)
    return res


def electivedwise_generation(elective_id, ids, k,
type=2):
    global ANNOY
    if type==1:
        ANNOY = ANNOY_ONEHOT
    elif type==2:
        ANNOY = ANNOY_PERCENTAGE
    subs = load_elective(elective_id)
    dists = []
    for idx in ids:
        dists += [(s, ANNOY.get_distance(idx,
SUB_TO_IDX[s])) for s in subs]
```

```
        dists = refactor_dists(sorted(dists, key=lambda
x: x[1]))[:k]
    score_sum = sum([d[1] for d in dists])
    return [{ "name": d[0], "score":
round((d[1]/score_sum), 2) } for d in dists]


def load_variables():
    '''
        This function loads all the variable required to
make recommendations for the given
        subject through API
    '''
    global SUB_TO_IDX, IDX_TO_SUB
    # Common Path to all variables
    for ann, t in zip([ANNOY_ONEHOT, ANNOY_PERCENTAGE],
['ONE_HOT', 'PERCENTAGE']):
        mid_path = [os.curdir, 'model', 'KNB_MODEL', t]

        print("LOADING ANNOY...")
        ann.load(os.path.join('', *(mid_path +
['tree.ann'])))

    print("LOADING SUBJECT-TO-INDEX MAP...")
    with open(os.path.join('', *(mid_path +
['subject2idx.pkl'])), 'rb') as f:
        SUB_TO_IDX = pickle.load(f)

    print("LOADING INDEX-TO-SUBJECT MAP...")
    with open(os.path.join('', *(mid_path +
['idx2subject.pkl'])), 'rb') as f:
        IDX_TO_SUB = pickle.load(f)

def refactor_dists(dists):
    d = defaultdict(lambda: 1e10)
    for dist in dists:
        name, score = dist
        d[name] = min(d[name], score)
    return [(name, value) for name, value in d.items()]


load_variables()
```

```
if __name__ == '__main__':
    app.run()
```

5.2.2 extractor.py

```
from annoy import AnnoyIndex
import pandas as pd
import numpy as np
import pickle
import os


def make_np(df):
  df = df.drop(df.columns[0], axis=1)
  data = np.array(df).astype(float)
  return data


def load_to_tree(data, f, path):
  t = AnnoyIndex(f, 'angular')
  for i in range(len(data)):
    t.add_item(i, data[i])
  t.build(10)
  t.save(path)
  print("SAVED TREE")


def save_dicts(df, paths):
  subject2idx = { v:i for i, v in
enumerate(list(df[df.columns[0]]))}
  idx2subject = { i:v for i, v in
enumerate(list(df[df.columns[0]]))}

  print("SAVING SUBJECT TO INDEX...")
  with open(paths[0], "wb") as f:
    pickle.dump(subject2idx, f)

  print("SAVING INDEX TO SUBJECT...")
  with open(paths[1], "wb") as f:
    pickle.dump(idx2subject, f)


def extract_everything(df_path, tree_path, s2i_path,
```

```
i2s_path):
  df = pd.read_csv(df_path)
  data = make_np(df)
  load_to_tree(data, len(data[0]), tree_path)
  save_dicts(df, [s2i_path, i2s_path])


if __name__ == '__main__':
  extract_everything(
    os.path.join(os.curdir, 'dataset',
'KNB_PERCENTAGE.csv'),
    os.path.join(os.curdir, 'model', 'KNB_MODEL',
'PERCENTAGE', 'tree.ann'),
    os.path.join(os.curdir, 'model', 'KNB_MODEL',
'PERCENTAGE', 'subject2idx.pkl'),
    os.path.join(os.curdir, 'model', 'KNB_MODEL',
'PERCENTAGE', 'idx2subject.pkl'),
  )
```

5.2.3 helper.py

```
import pandas as pd
import os

def load_elective(choice):
    subs = pd.read_csv(os.path.join(os.curdir,
'dataset', 'elective_subjects.csv'))
    return list(subs['elective_{}'.format(choice)])
```

CHAPTER 6

TESTING

Test cases are important to ensure the reliability, functionality, and accuracy of the recommendation system for university elective courses. Here are some potential test cases:

6.1 INPUT VALIDATION

Test cases should be designed to check whether the system can handle invalid or unexpected inputs, such as incorrect data formats or empty fields.

6.2 PERFORMANCE TESTING

Test cases should be designed to evaluate the performance of the system under different loads and user traffic.

6.3 ACCURACY TESTING

Test cases should be designed to evaluate the accuracy of the recommendation system. This can be done by comparing the system's recommendations to a manually curated list of recommended courses.

6.4 EDGE CASE TESTING

Test cases should be designed to evaluate the system's ability to handle edge cases, such as students with unique academic histories or course requirements.

6.5 INTEGRATION TESTING

Test cases should be designed to evaluate the system's ability to integrate with other systems, such as learning management systems or student databases.

Overall, a comprehensive testing plan should be developed to ensure the reliability, functionality, and accuracy of the recommendation system. By conducting thorough testing, we can identify and address potential issues before they impact the user experience.

6.6 UNIT TEST

We mainly took context of unit testing and here are the following testcases we have written to test our backend:

6.6.1 For /api/v1/subject-list:

```
def test_subject_list():
    with app.test_client() as client:
        response = client.get('/api/v1/subject-list')
        data = response.get_json()
        assert response.status_code == 200
        assert 'subjects' in data
        assert isinstance(data['subjects'], list)
```

6.6.2 For /api/v1/similar-subject:

```
def test_similar_subject():
    with app.test_client() as client:
        # Test with valid input
        input_data = {
            'subjects': ['Math', 'Physics',
'Chemistry'],
            'electives': [1, 2],
            'k': 2,
            'type': 2
        }
        response = client.post('/api/v1/similar-
subject', json=input_data)
        data = response.get_json()
        assert response.status_code == 200
        assert isinstance(data, list)
        assert len(data) ==
len(input_data['electives'])
        for res in data:
            assert isinstance(res, dict)
```

```
            assert
'elective_{}'.format(input_data['electives'][data.index
(res)]) in res
            assert
isinstance(res['elective_{}'.format(input_data['electiv
es'][data.index(res)])], list)
            assert
len(res['elective_{}'.format(input_data['electives'][da
ta.index(res)])]) == input_data['k']

        # Test with invalid input
        input_data = {
            'subjects': ['Math', 'Physics',
'Chemistry'],
            'electives': [1, 2],
            'k': 10, # higher than the max limit
            'type': 2
        }
        response = client.post('/api/v1/similar-
subject', json=input_data)
        data = response.get_json()
        assert response.status_code == 200
        assert 'error' in data
        assert isinstance(data['error'], str)
```

6.6.3 Invalid Subject

```
def test_similar_subject_invalid_subject():
    data = {
        "subjects": ["Invalid Subject"],
        "k": 3,
        "electives": [1],
        "type": 2
    }
    response = client.post("/api/v1/similar-subject",
json=data)
```

```
    assert response.status_code == 200
    assert response.json() == {"error": "Sorry given
subject is not in database"}
```

6.6.4 Invalid Elective

```
def test_similar_subject_invalid_elective():
    data = {
        "subjects": ["Subject 1", "Subject 2"],
        "k": 3,
        "electives": [10],
        "type": 2
    }
    response = client.post("/api/v1/similar-subject",
json=data)
    assert response.status_code == 200
    assert response.json() == {"error": "Sorry given
elective is not in database"}
```

6.6.5 Missing Elective

```
def test_similar_subject_missing_elective():
    data = {
        "subjects": ["Subject 1", "Subject 2"],
        "k": 3,
        "type": 2
    }
    response = client.post("/api/v1/similar-subject",
json=data)
    assert response.status_code == 200
    assert len(response.json()) == 1
    assert "elective_1" in response.json()[0].keys()
```

CHAPTER 7

FUTURE SCOPE

The recommendation system for university elective courses has immense potential for future expansion and development. Here are some potential future scopes for this project:

7.1 INTEGRATION WITH ADDITIONAL DATA SOURCES

Currently, the system uses data from past performance on curriculum subjects to recommend elective courses. In the future, the system could be expanded to incorporate additional data sources, such as student surveys or trends in the job market.

7.2 PERSONALIZATION

Currently, the system recommends courses based on a student's past performance and interests. In the future, the system could be expanded to provide personalized recommendations based on a student's individual learning style, personality, and career goals.

7.3 REAL-TIME UPDATES

The system currently uses historical data to generate recommendations. In the future, the system could be updated in real-time, providing students with up-to-date recommendations based on their current academic progress.

7.4 INTEGRATION WITH LEARNING MANAGEMENT SYSTEMS

The system could be integrated with learning management systems used by universities, allowing students to access their recommendations and course information in one centralized location.

Overall, the future scope of this project is vast, with many opportunities for expansion and development. By incorporating new data sources, personalization, real-time updates, and integration with learning management systems, we can continue to improve the accuracy and relevance of the system, providing students with the best possible recommendations for their elective courses.

CHAPTER 8

LIMITATIONS

Like any other project, the recommendation system for university elective courses has its limitations. Here are some of the limitations of this project:

8.1 LIMITED DATA AVAILABILITY

The accuracy of the recommendation system is highly dependent on the data available. If there is limited data available for a particular subject or student, the recommendations may not be as accurate.

8.2 LIMITED SCOPE

The recommendation system is limited to elective courses offered by the university. It does not take into account courses offered by other universities or online learning platforms.

8.3 LACK OF STUDENT INPUT

While the recommendation system considers past performance and interests, it does not consider the student's personal preferences or experiences. Students may have unique learning styles, which are not accounted for in the system.

8.4 LACK OF CONSIDERATION FOR EXTERNAL FACTORS

The recommendation system does not consider external factors that may impact a student's ability to take a particular course, such as scheduling conflicts or prerequisites.

Overall, while the recommendation system is a valuable tool for students, it has limitations. By recognizing these limitations and working to improve the accuracy and relevance of the system, we can continue to provide students with the best possible recommendations for their elective courses.

CHAPTER 9

CONCLUSION

In general, an elective recommendation system can be a useful tool for helping students choose courses that align with their interests and goals. These systems can use a variety of factors, such as student academic history, test scores, and interests, to make personalized recommendations. However, it is important to note that these recommendations should be considered alongside other factors, such as the availability of courses and the advice of teachers and counselors. Ultimately, the decision about which elective courses to take should be made by the student, with input from the relevant parties.

The Elective Subjects Recommendation System provides an efficient and effective way for students to search for and select suitable elective subjects that best meet their needs. It helps them to save time and effort in searching for elective subjects, and by providing personalized recommendations, it can help students to find the best-suited elective subjects quickly and easily. The system can also help teachers to better understand the needs of their students and to provide appropriate guidance. Overall, the system provides an efficient and effective solution to an important problem in the educational sector.

To conclude this project, we came to the result that it is a highly rational model for getting suggestions for the elective subjects. Please notice we are concluding this project as suggestion model for elective subject because whole purpose was to getting suggestions from system, not concrete decision.

For rationality it uses students' previous subjects' marks and uses machine learning algorithms and mathematical models to map them to future elective subjects by distributing subjects to knowledge domain percentage, we make it easy to compare subjects with each other.

Basically, this is a Suggestions Expert system which is a subpart of Knowledge Expert System, here we will always take output of the model as initial point to check against with Knowledge Experts which are professors in our case.

CHAPTER 10

BIBLIOGRAPHY

[1]. Erdvin, Farhan Muhammad Ardiansyah, Hendika Solim, Alexander A.S. Gunawan,
Level of user satisfaction with the current you tube recommendation system,
Procedia Computer Science,
Volume 216,
2023,
Pages 442-452,
ISSN 1877-0509,
https://doi.org/10.1016/j.procs.2022.12.156.

[2]. Yong Wang, Mingxing Gao, Xun Ran, Jun Ma, Leo Yu Zhang,
An improved matrix factorization with local differential privacy based on piecewise mechanism for recommendation systems,
Expert Systems with Applications,
Volume 216,
2023,
119457,
ISSN 0957-4174,
https://doi.org/10.1016/j.eswa.2022.119457

[3]. Hao Tang, Guoshuai Zhao, Yujiao He, Yuxia Wu, Xueming Qian,
Ranking-based contrastive loss for recommendation systems,
Knowledge-Based Systems,
Volume 261,
2023,
110180,
ISSN 0950-7051,
https://doi.org/10.1016/j.knosys.2022.110180.

[4]. Lu-Tao Zhao, Dai-Song Wang, Feng-Yun Liang, Jian Chen,
A recommendation system for effective learning strategies: An integrated approach using context-dependent DEA,
Expert Systems with Applications,
Volume 211,
2023,
118535,

ISSN 0957-4174,
https://doi.org/10.1016/j.eswa.2022.118535.

[5]. Maurício Madson dos Santos Freitas, Jhonatas Rodrigues Barbosa, Emilly Marry dos Santos Martins, Luiza Helena da Silva Martins, Fabricio de Souza Farias, Lúcia de Fátima Henriques Lourenço, Natácia da Silva e Silva,
KNN algorithm and multivariate analysis to select and classify starch films,
Food Packaging and Shelf Life,
Volume 34,
2022,
100976,
ISSN 2214-2894,
https://doi.org/10.1016/j.fpsl.2022.100976.

[6] Medjeu Fopa, Modou Gueye, Samba Ndiaye, Hubert Naacke,
A parameter-free KNN for rating prediction,
Data & Knowledge Engineering,
Volume 142,
2022,
102095,
ISSN 0169-023X,
https://doi.org/10.1016/j.datak.2022.102095.

[7] Matthijs Meire, Kristof Coussement, Arno De Caigny, Steven Hoornaert,
Does it pay off to communicate like your online community? Evaluating the effect of content and linguistic style similarity on B2B brand engagement,
Industrial Marketing Management,
Volume 106,
2022,
Pages 292-307,
ISSN 0019-8501,
https://doi.org/10.1016/j.indmarman.2022.09.006.

[8] Muxing Liu, Qiuyue Wang, Jun Yi, Hailin Zhang, Ji Liu, Wei Hu,
Influence of vegetation type and topographic position on volumetric soil water content dynamics and similarity among surface and deep soil layers,
International Soil and Water Conservation Research,
Volume 11, Issue 1,
2023,
Pages 183-196,
ISSN 2095-6339,

https://doi.org/10.1016/j.iswcr.2022.07.002.

[9] Michele Garetto, Emilio Leonardi, Giovanni Neglia,
Content placement in networks of similarity caches,
Computer Networks,
Volume 201,
2021,
108570,
ISSN 1389-1286,
https://doi.org/10.1016/j.comnet.2021.108570.

[10] Anna Formica,
Concept similarity in Formal Concept Analysis: An information content approach,
Knowledge-Based Systems,
Volume 21, Issue 1,
2008,
Pages 80-87,
ISSN 0950-7051,
https://doi.org/10.1016/j.knosys.2007.02.001.

[11] Hamidreza Esmalifalak,
Euclidean (dis)similarity in financial network analysis,
Global Finance Journal,
Volume 53,
2022,
100616,
ISSN 1044-0283,
https://doi.org/10.1016/j.gfj.2021.100616.

[12] Reza Zubaedah, Fransiskus Xaverius, Hasanudin Jayawardana, Serli Hatul Hidayat,
Comparing euclidean distance and nearest neighbor algorithm in an expert system for diagnosis of diabetes mellitus,
Enfermería Clínica,
Volume 30, Supplement 2,
2020,
Pages 374-377,
ISSN 1130-8621,
https://doi.org/10.1016/j.enfcli.2019.07.121.

[13] Abdelhadi RADOUANE, Fouad GIRI, Abdessamad NAITALI, Fatima Zahra Chaoui, Similarity improvement using angular deviation in multimodel nonlinear system identification,
IFAC Proceedings Volumes,
Volume 46, Issue 11,
2013,
Pages 605-610,
ISSN 1474-6670,
ISBN 9783902823373,
https://doi.org/10.3182/20130703-3-FR-4038.00064.

[14] Weinan Liu, Youmin Rong, Guojun Zhang, Yu Huang,
A novel method for extracting mutation points of acoustic emission signals based on cosine similarity,
Mechanical Systems and Signal Processing,
Volume 184,
2023,
109724,
ISSN 0888-3270,
https://doi.org/10.1016/j.ymssp.2022.109724.

[15] Peng Ji, Shiliang Shi,
Hazard prediction of coal and gas outburst based on the Hamming distance artificial intelligence algorithm (HDAIA),
Journal of Safety Science and Resilience,
2023,
,
ISSN 2666-4496,
https://doi.org/10.1016/j.jnlssr.2022.12.001.