Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

Anefiyayeva Dayana

University course recommendation system

A thesis submitted for the degree of
Degree of Master of Science in Computer Science
(degree code: 7M06102)

Kaskelen, 2022

Suleyman Demirel University

Faculty of Engineering and Natural Sciences

Department of Computer Science

Dean of Faculty

Associate Professor, PhD Zhamanov A.

«30» may 2022

Topic of the thesis:

University course recommendation system

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science" SDU, 2020-2022

Head of Department Associate Professor, PhD Cemil Turan

Academic Supervisor PhD Talasbek Assem

Master's student Anefiyayeva Dayana

Kaskelen, 2022

Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University
Faculty of Engineering and Natural Sciences

University course recommendation system

A thesis submitted for the degree of
Degree of Master of Science in Computer Science
(degree code: 7M06102)

Author: **Anefiyayeva Dayana**

Supervisor: **PhD Assem Talasbek**

Dean of the faculty:
Associate professor, PhD Azamat Zhamanov

Kaskelen, 2022

Abstract

In universities with a credit system, students are given a choice of subjects - elective courses. There is a whole list of subjects of different profiles from which they must choose themselves. This decision will affect the student's future career, as he will choose not only the subject studied but also what he will do in the future. Understanding how difficult a choice needs to be made, and how difficult it is to choose a subject, it was decided to create a recommendation system for subjects. One of the most used applications of artificial intelligence is the recommendation engine. Since the advent of the Internet, recommender systems have become more and more common in everyday life. Recommender systems can be implemented using various machine learning methods, but this task attracts a large number of scientists from all over the world. Let's start with how to classify k-means while comparing two different approaches. The system was developed using machine learning technology and cross-platform application development. The algorithm was based on data that was collected from various public sources. The relevance of this topic led us to create an application, two types of recommender systems:

1) Entering the subject that the student liked, the system will give him a list of subjects with similar skills

2) Entering your interests and abilities, the system gives out an item that is most suitable for your preferences

Keywords: ml, k-means, courses, universities, recommendation system

Аңдатпа

Жоғары оқу орындары кредитік жүйе арқылы орындалады және олар студенттерге таңдау мен қосымша пәндерді таңдауды ұсынады. Бұл әртүрлі профильдегі пәндердің толық тізімін көрсете отырып, таңдау мүмкіндігін толығымен өздеріне қалдырады. Бұл шешім студенттің болашақ кәсібіне әсер етеді, өйткені ол тек оқылатын пәнді гана емес, жақын арада не істейтінін де таңдайды. Таңдаудың қаншалықты қиын екенін, пәнді таңдаудың қандай ауыр шешім екенін түсіне отырып, пәндерге ұсыныс беру жүйесін құру туралы шешім қабылдадым. Жасанды интеллекттің ең көп қолданылатын қосымшаларының бірі - ұсыныс қозғалтқышы. Ғаламтор пайда болғаннан бері рекомендациялық жүйелер күнделікті өмірде жиі кездеседі. Ұсынатын жүйелерді машиналық оқытудың әртүрлі әдістерін қолдану арқылы жүзеге асыруға болады, бірақ бұл тапсырма бүкіл әлемнің көптеген ғалымдарын әлі де қызықтырады. Екі түрлі тәсілді салыстыра отырып, k-means тәсілі арқылы қалай жіктеуге болатынынан бастаймыз. Жүйе машиналық оқыту технологиясын және кросс-платформалық қосымшаларда әзірлеуді пайдалана отырып жасалған. Алгоритм әртүрлі қоғамдық көздерден жиналған деректерге негізделген. Бұл тақырыптың өзектілігі бізге қосымшаны, екі түрдегі кеңес беру жүйесін құруға әкелді:
1) Студентке ұнаған пәнді енгізген кезде жүйе оған сипаттамлары ұқсас пәндердің тізімін береді
2) Сіздің қызығушылықтарыңыз бен қабілеттеріңізді қарай отырып, жүйе қалауыңыз бойынша ең қолайлы курстар ұсынады

Түйін сөздер:Жасанды интеллект, k-means тәсілі, курстар, университеттер, ұсыныс жүйесі.

Аннотация

В университетах с кредитной системой для студентов предоставляется выбор предметов, элективных курсов. Это целый список предметов разных профилей из которых они должны выбрать сами. Это решение повлияет на дальнейшую карьеру студента, так как он будет выбирать не только изучаемый предмет, но и то, чем он будет заниматься в ближайшем будущем. Понимая какой сложной выбор нужно сделать, как трудно выбрать предмет, принято решение создать рекомендательную систему по предметам. Одним из наиболее часто используемых приложений искусственного интеллекта является механизм рекомендаций. С момента появления Интернета рекомендательные системы становятся все более распространенными в повседневной жизни. Рекомендательные системы могут быть реализованы с использованием различных методов машинного обучения, но эта задача привлекает большое количество ученых со всего мира. Начнем с того, как классифицировать k-средних при этом сравним два разных подхода. Система разработана с использованием технологии машинного обучения и разработки кроссплатформенного приложение. Алгоритм был основан на данных, которые были собраны из различных открытых источников. Актуальность данной темы привела нас к созданию приложения, два вида рекомендательной системы:
1) Вводя предмет, который понравился студенту системы выдаст ему список предметов схожих по навыкам
2) Вводя свои интересы и способности, система выдает предмета максимально подходящих под предпочтения

Ключевые слова: машинное обучение, методология k-means, курсы, вузы, рекомендательная система.

Contents

Chapter 1

Introduction

1.1 Motivation

Universities with a credit system have compulsory and elective courses. The courses students must choose are named compulsory, and elective courses that students must take by students themselves. Compulsory subjects are primarily aimed at completing research. Electives, on the other hand, are aimed at a narrower discipline. This choice will affect your entire life as the course you complete will determine your future work. From this, we can conclude that this decision is one of the most important. Create an automated system based on this. The recommendations within the university wall are appropriate. It analyzes and emphasizes the main aspects that influence the successful delivery of the subject and recommends selected students based on this. Industry wants graduates to be as well-rounded and diversified as possible when their profiles are examined, therefore the number of electives given in qualitative courses has increased in recent years. When considering the business sector's attention to these issues, picking electives about urbanization and management, environmental management, and CSR (Corporate Social Responsibility) would be a smart option. Furthermore, there is nothing wrong with selecting electives that are often evaluated higher for a variety of reasons. Though your friends may scoff at you for padding your CV with easy-to-score and higher-grading electives, the fact is that you do not lose anything and benefit from the practice. Electives should not be chosen only for this purpose, as recruiters are typically savvy enough to recognize these characteristics, as they are primarily previous institute alumni. Furthermore, selecting high-quality electives is a fantastic way to keep your profile current and futuristic, since lesser-known courses with promise might make you valuable to niche businesses.

The goal of this work is:

- Determine the impact of personal interests on subject selection.

- Identifying the most accurate recommender algorithm to assist with subject selection
- Describe the construction (application) of an automated system that will assist with subject selection using a recommender system.

1.2 Aims and Objectives

The main goal of my research is to create a recommendation system for students to help them choose a subject based on the results obtained when they pass a psychological test. This recommender system is built using the latest machine learning algorithm technology.

- Availability. The application is free and any user of the Android platform, iOS or website can use it at any time.
- Credibility. The results given by our application are based on an algorithm that uses different vector transformations.
- Merged technology and education to make life easier for students.
- Save time. You don't have to sit down and make a list of pros and cons. You do not need to panic not to understand which subject to choose. Just enter into the system a subject that you have taken before or write your preferences, and the system will give you a list of subjects from which you can choose.

1.3 Recommendation System

Recommendation systems (RS) are widely used in many areas of modern life. They can also be used in education. Recommendation system are a type of machine learning that is used to rate or rank, in this work, students and subjects. A recommender system, in general, is a system that anticipates how a user would rate a certain item. After then, the user's predictions will be rated and returned to them.

Various well-known firms, such as Instagram, Google, Amazon, Spotify, Reddit, Netflix, and others, utilize them to keep and prolong users' time on their platforms, as well as for general advertising. For example, Netflix would suggest shows that are similar to the ones you frequently watch or enjoy so that you may continue to watch episodes and movies on their platform. Amazon utilizes recommendations to propose items from comparable businesses you follow, like,

or comment on, and they also promote the same accounts to your friends based on your data. .

Some recommender systems employ algorithmic and template-based techniques, while others use more modeling-oriented approaches such as collaborative filtering, content-based, link prediction, and so on. The complexity of each of these techniques varies, but complexity does not imply "excellent" performance. Frequently, the most basic concepts and implementations yield the best outcomes. Large firms like Google, for example, have promoted content on their platform using simple boilerplate implementations of recommendation engines.

Many algorithm developers struggle with determining what constitutes an effective recommender system. This definition of a "good" suggestion might assist you in determining the success of the recommender you've built. A number of approaches that measure coverage and accuracy can be used to evaluate the quality of a suggestion. Coverage indicates the proportion of items in the search space for which the system can produce suggestions, whereas accuracy is the proportion of right recommendations out of the total number of potential recommendations. The process for assessing a suggestion is completely determined by the data collection and the methodology employed to develop it.[14]

Thesis outline

The dissertation work includes an introduction, six chapters including a conclusion, a list of references and an abstract in three languages.

The first chapter is an Introduction. The significance and relevance of the dissertation subject are established in this chapter, as are the aims of the dissertation work and the tasks to be completed, as well as the work's scientific innovation and practical value.

In this second chapter, the relevance of the work is checked. Finding works that are related to the topic for the last 5 years. The discovery that they made, the method they used, what data were used in the work, what data the algorithms were tested on, what conclusion they made in this work, what they failed now and what they will add in the future are described.

The chapter 3 describes the methods, tools and architecture. We tell you what tools were used to create the application and for the recommender system. The method by which we build a recommender system is also described in more detail. Tells just different types of vectorizing. How the date was prepared for work, how the effectiveness of the method was evaluated, as well as what results were obtained.

In the conclusion, the work's findings are summarized, as well as the research's primary findings, difficulties with solutions, and future directions for scientific research.

Chapter 2

Architectural design

This UML diagram in Figure 1 depicts the user experience in the application created for the recommendation system. As soon as the application run, the user will be given a choice of whether he wants to start with the recommendation by course or by interest. As you can see, if he selects by interest, then he will only need to enter the text, which requires a minimum of 100 symbols for efficient prediction of recommendation courses. After that user presses the button to see the list of recommended courses. The program provides text to the model, which in turn returns similar subjects. He will choose any course he liked and preferred and can read more about that course: its released date, duration, and as well as the full description.

If the user prefers to find similar courses by providing an already studied or liked subject, he needs only write the id of the subject to filter from all of the courses. When the user finds a necessary course from the list, he selects the subject and the program sends back the results of related items. As mentioned above method he can get complete information about that course.

Users can repeat steps in the above paragraphs for infinite time because it is a matter to choose the right course to study. Moreover, they may run applications on platforms Android and IOS mobile phones, as well as on web-browser.

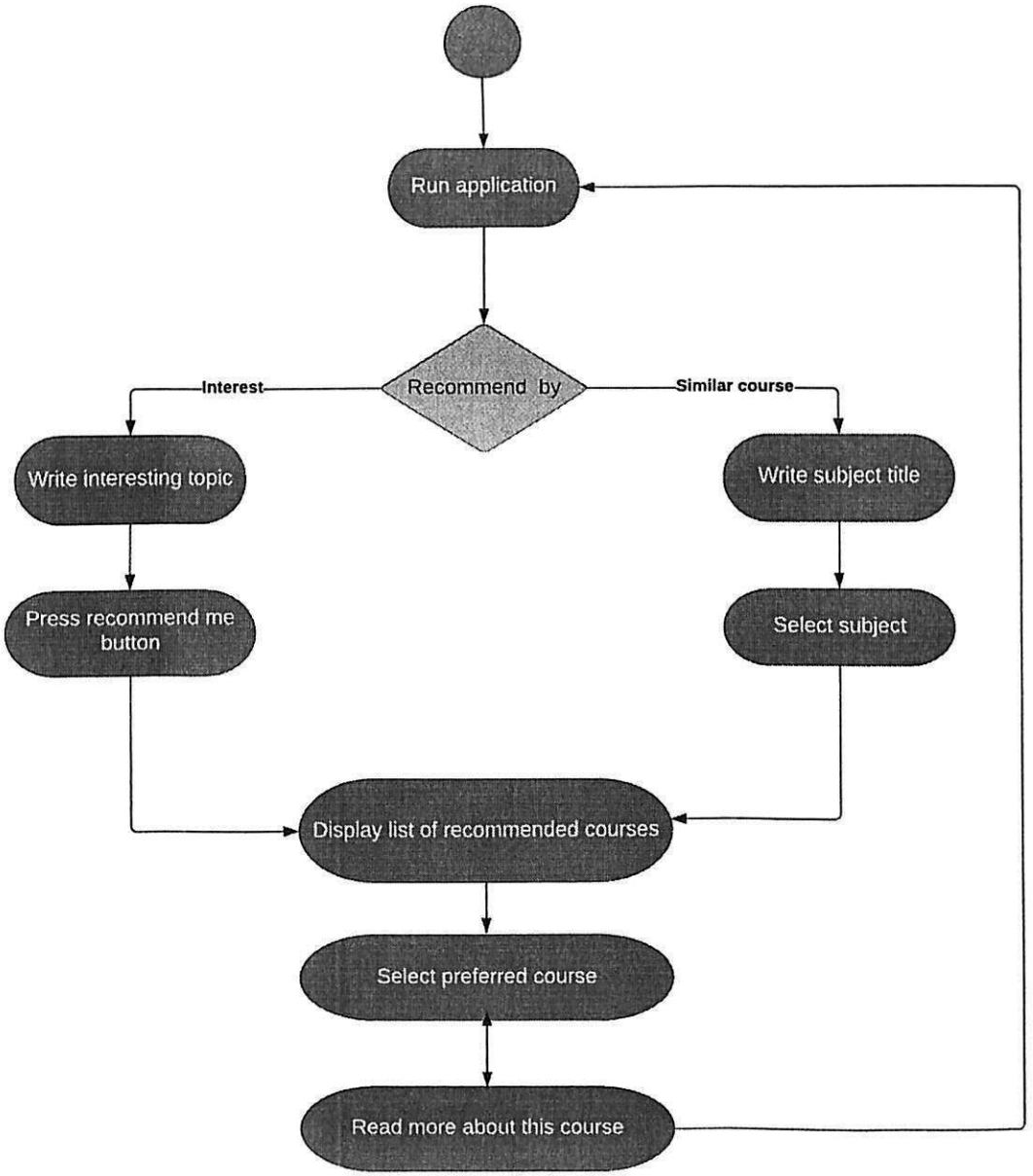


Figure 1. UML diagram for recommendation system application

Chapter 3

Literature Review

A recurrent neural network (RNN) is used to predict behavior while preserving previously observed behavior. In the case of recommender systems, that means a series of sentences without intuitive foresight is too narrow, putting the user in a filter bubble. This article solved this problem. Predict the outcome of the courses offered to students to choose from the ones that suit their interests. Create a set of models based on the course catalog (BOW) and another model based on the registration history (course2vec). Performance comparison of two models using validation datasets. Comparing the two models, user findings show a dramatic indifference to RNN policies. However, the machine learning of the course2vec model showed the best results for the task when validated offline. [1].

This article introduces some of the methods used in the course recommendation system. With the help of this system, students can know the approximate results in advance, which helps them choose to study. There is also a table and comparative analysis of the effectiveness of the educational data for each method. Then the best method was selected. And the final version of the recommended system was created. It can already be used in practice. [2].

The course recommendation system is designed to allow students to choose from a variety of subjects. However, the student's choice cannot depend solely on his or her interests. Teachers, peers, and others influence them. Existing rating matrices in the form of interpersonal relationships, rating text, and "user elements" form a multimodal data structure from multiple sources. Therefore, to make recommendations based on these different qualities, need a way to systematically combine the data. Therefore, this article proposes a hybrid recommended model that combines structured network functionality with the use of graph neural networks and interactive student actions with factorization of tensors. First, a structured graphical learning evaluation network is developed using student evaluations to characterize students, courses, commentary texts, grades, and interpersonal relationships. By examining the student's relationship structure, use a neural network with a random walk to create a vectorized

representation of the student. Finally, use the Bayesian probability tensor decomposition to examine and predict the scores of students in subjects that are not attending and recognize these personalized traits as the third dimension of the evaluation tensor. Due to the small prediction error and improved recommendation accuracy, the proposed method outperforms other current matrix factorization models and neural networks (RTTF, xSVD++, DSE).[3].

Pakistani universities offer credit programs of all kinds to applicants. Pakistan Virtual University-Developed with the latest information technology. This university offers a choice of a large number of professions and specialties. Each program has subjects that the student must complete. The chosen course that suits the competencies and interests greatly affects the final grade (CGPA) of the student. In this work, a system for recommending subjects in the Virtual University was created. The system has been tested on 470 courses that are accessible, including simulated data from 2,600 people. As a result, it has been proved that grades are influenced by the average score of the student in the courses already studied and the average score in similar courses. The implemented system and its accuracy were evaluated using the mean absolute error for 100 observations. The MAE was in the acceptable range[4].

This work describes a hybrid recommendation system that uses Content-Driven Filtering (CBF) and Collaborative Filtering (CF) to propose the most relevant courses for students based on several factors linked access to both student and course data. The genetic algorithm (GA) was created to find the best RS configuration, which takes into account the most significant criteria and the remaining factors. The pilot project employed real data from the University of Cordoba's computer science program, comprising data collected from students over three academic years and based on 2,500 records from 95 individuals and 63 courses. The examination of the most acceptable criteria for course suggestions is demonstrated by experimental findings. The experimental findings show the relevance of employing a hybrid model that includes both student and course information to increase the dependability of suggestions, as well as better performance compared to earlier models [5].

In several disciplines, recommendation systems are commonly employed. These systems function by proposing tailored lists of articles to users based on their interests, therefore assisting users in navigating the overabundance of information available to them. When beginning a new level of study, users such as students find selecting the appropriate courses to be a challenging undertaking. Choosing the incorrect courses can have a negative impact on students' academic performance as well as their future professional prospects. This article will look at how the recommendation system may be used to assist students to choose courses that are a good fit for their skills and interests. The findings of this study imply that a combined counseling approach/system may be the most effective strategy for assisting students in selecting the appropriate

courses for their future jobs.[8]

Students in higher education experience difficulties while selecting optional courses for their studies. Advisors are employed by the majority of higher education institutions to assist with this responsibility. Recommender systems have their roots in commerce and are now employed in a variety of fields, including education. Recommender systems are a viable substitute for human advisers. The purpose of this study is to look at the breadth of recommender systems that help students choose optional courses. To do so, a systematic literature review (SLR) was done on the recommender systems corpus for picking optional courses published between 2010 and 2019. Only 24 papers out of 16 981 were found that discuss recommender systems for selecting optional courses and were included in the final analysis. Several recommender systems techniques and data mining algorithms are utilized in these articles to accomplish the goal of recommending optional courses. Current research on the usage of recommender systems for selecting optional courses is lacking, according to this study. To improve the performance of recommender systems for optional courses, more studies in numerous undiscovered areas might be investigated. This research adds to the corpus of knowledge on recommender systems, particularly those used to help students choose optional courses in higher education.[9]

Choosing a few optional courses from a wide range of options has proven to be a challenging undertaking for many students throughout the years. Guidance and counselors, sometimes known as level advisers, are often employed at many higher education institutions to aid students in making the best course choices. In actuality, these counselors and advisers are frequently overburdened with too many pupils to serve, and they frequently do not have enough time to devote to them. Most of the time, the academic strength of the student based on previous results is not taken into account while making new optional choices. Advanced data analysis techniques are used by recommender systems to assist users in finding items of interest by generating a projected likeliness score or a list of top suggested items for a specific active user. As a result, a collaborative filtering-based recommender system was built in this study to dynamically recommend optional courses to undergraduate students based on their previous grades in relevant courses. The k-nearest Neighbour method was used in this technique to uncover hidden linkages between relevant courses passed by students in the past and currently accessible elective courses. The recommendation model was built and tested using a dataset of real-life student results. The new model outperformed previous findings in the literature. The established approach will not only assist students in better their academic performance, but it will also aid level advisers and school counselors minimize their workload. [10]

We propose a unique recommendation system for course selection in the specialty of information management at Chinese universities, as an essential

component of smart education. To construct this system, we first collect the course enrollment data set for a specific group of students. In our approach, the sparse linear method (SLIM) is used to create top-N course suggestions that are acceptable for the students. Meanwhile, the aL0 regularization term is used as an optimization approach in the existing recommendation system, which is based on the observation of course items. To assess the performance of our method, comparative experiments between state-of-the-art approaches and our approach are carried out. The results of experiments on a variety of themes and with a large number of courses reveal that our suggested technique beats state-of-the-art methods in terms of accuracy and efficiency.[11]

One of the most severe issues we confront today is the failure of university students to complete their studies. According to data from the Ministry of National Education Republic of Indonesia's Centre for Education Statistics Research and Development, the proportion of students who graduated on time between 2001 and 2011 was just 51.97 percent. Furthermore, incidences of students dropping out at the start of the semester are extremely common. One of the reasons for the study's failure was the substantial selection faults made when applying to university. Using the Association Rule approach, this study proposes a subject selection recommendation system based on profile data and student interest.The results of the rules of the connection will then be matched with prospective students via dynamic surveys, so new students may expect more valid subject suggestions that fit their profile and interests. The system developed as a result of this study makes use of student data recorded in Dian Nuswantoro University's academic system. This paradigm, on the other hand, maybe used by any university with an academic information system. At the end of the day, this approach should be employed to reduce failures due to student study major election blunders.[12]

A customized hybrid course recommendation system (PHCRS) takes into account student interest, talents, and progression career in this study. To design a PHCRS that meets the needs of individual students, we applied the five widely algorithms that were employed: popularity-based techniques, content-based, item-based collaborative, user-based collaborative, also score-based methods. By gathering course syllabus and class labels (for example, skills offered, different levels). Following that, trained five recommendation models using course names as well as students' previous course selections and grades. To conducted trials with students at the Department of Electrical and Computer Engineering offers 1794 courses to 925 students and uses the receiver operating characteristic curve (ROC) and normalized discounted cumulative gain (NDCG) as metrics to evaluate system's correctness. The findings indicated suggested system obtain ROC accuracies of 80% and NDCG accuracies of 90%. We asked 46 people to try out system and fill out survey. Total, the recommended courses piqued the attention of a large number of participants 60 to 70%, and the content-based

filtering course suggestion lists matched 67.40 percent of students' real course preferences. The courses at the top of the suggestion lists piqued the students' attention, also across the five recommendation systems, more participants were self-motivated than those who were motivated by extrinsic information. These findings showed that a recommended course suggestion system can help students choose courses that they are most interested in. [13]

Many colleges and universities have used flexible curriculum systems, as well as increasing the a wide range of courses are provided, in order to provide reliable information to students about the courses in which they may display better ability and interest, word-of-mouth no longer suffices. There is currently a need to guarantee that students make the greatest use of available data in order to make better course selection decisions. There are three types of recommendation systems: collaborative filtering, association-rule based, and content based.

There are three types of recommendation systems: collaborative filtering, association-rule based, and content based. The proposed recommendation system aims to promote courses based on a number of parameters, including course relevance to students' departments, quantitatively measurable aptitude in the form of course grades, and a quality-control mechanism based on filtering courses with low teacher ratings. To account for these variables, a modified two-tiered user-based pattern matching recommendation system is developed. For forecasting future grades, using a user-based collaborative suggestion system might be a good option. In a user-based collaborative recommendation system, such as an e-commerce site, product suggestions are based on a user's affinity for a user group. One user, for example, buys a phone, a screen protector, and a phone case, while another buys a phone and a screen guard. A collaborative filtering system would detect the users' relationship and suggest a phone cover for the second user.

In the case of user-based collaborative filtering, user clusters will be identified quantitatively based on their ratings/scores, and prediction values will be presented to new users based on their affinity to the cluster. Students getting grades will be placed in comparable cluster patterns in our course recommendation system by using an Artificial Immune System (AIS) clustering technique to compute the affinities between distinct students in a training data pool. The AIS has been shown to be an excellent tool for clustering data. The application of the AIS approach will improve the quality of the solution presented here. The anticipated values are then calculated using the affinity values of new students in clusters as weights.[15]

With recent developments in contrastive learning techniques, the gap between supervised and unsupervised pretraining has shrunk. These provisional approaches are often used online and rely on a high number of explicit pairwise

feature comparisons, which can be time consuming. We present SwAV, an online approach that uses contrastive techniques without the need to conduct pairwise comparisons, in this study. Instead of comparing features directly as in contrastive learning, our technique clusters the data while maintaining consistency across cluster assignments obtained for distinct augmentations (or views) of the same picture. Simply put, we employ a flipped prediction process, in which the code of one view is predicted from the representation of another. With recent developments in contrastive learning techniques, the gap between supervised and unsupervised pretraining has shrunk. These contrastive algorithms are often employed online and rely on a large number of explicit pairwise feature comparisons, which might take a long time. In this paper, we introduce SwAV, an online method that employs contrastive techniques without the necessity for pairwise comparisons. Rather of simply comparing features like contrastive learning does, our method clusters the data while retaining consistency among cluster assignments achieved for different augmentations (or views) of the same picture. Simply put, we use a flipped prediction technique, which predicts the code of one view from the representation of another.[16]

This article has developed a web-based course recommendation system. When students are having problems deciding which courses to take, they might utilize a course recommendation system to help them. This research develops a recommendation system based on our previously described Prediction Methodology. The prediction approach incorporates both Data Mining and Graph Theory methodologies, which will be presented in this paper to give readers an understanding of how we may create appropriate course recommendations for students. The results in this research were obtained by applying our prediction mechanism as well as the questionnaire to two classes (classes 2001 and 2002) over the course of two school years. The questionnaire is meant to determine whether or not our prediction approach is effective.[17]

This study proposes a user-based Course Recommendation System. The recommender system is built as an online website/application that may provide a tailored list of courses for a user to take. In this domain of user-based recommendation, modern variants of conventional recommender systems, such as collaborative filtering, are seen to be effective. The recommendation method is separated into three parts based on the collaborative filtering principle: representation of student data, generation of neighbor user (student), and generation of course suggestions. During the neighbor creation process, the Cosine algorithm is used to calculate the likenesses between each user. The course suggestions are then created based on the evaluations of other students as well as his own personal ratings.[18]

The fundamental issue for today's students is that they demand customized access to data and figures depending on their choices and needs. To overcome this problem, a recommender system is used to analyze data automatically

based on the user's preferences, and the best option is offered in a variety of options The recommender system is used to customise data by recognizing a similar user or identifying specific issues of importance to the user. RS prioritized information, related it to products, and offered users relevant suggestions based on their preferences. A recommender system is an intelligent system that extracts a customized set of data from a large volume of data generated constantly. The information is mostly filtered by recommender systems based on user ratings or opinions on specific items (collaborative filtering) or previous users' choices (content-based filtering). The option made by a comparable individual in the past is recommended through content-based filtering. Clustering is a machine learning approach for grouping data into a finite number of groups based on data similarity. These techniques are based on unsupervised learning and are mostly used to categorize unlabeled historical data. Members of the categorised categories are similar to one another but different from members of other groupings.[19]

The major focus of this research is to determine the impact of career exploration. The module is a computer-assisted software for making career decisions based on grades and preferences. The study's target audience was 10+2 graduates eager to pursue careers in sectors like as engineering, medicine, business, and the arts, among others, while students who had already made up their minds were used to assess the system's trustworthiness. Random sampling approaches were used to choose a sample of this accessible population. One of the most important variables in a student's academic achievement is the availability of career counseling services. The main aspect of student utilities is that they provide students with the finest plan for their future that suits their attitude and attributes. Students choose specific fields of study based on projected career possibilities, hobbies, and expected future advancements at the time of course completion. If a student is uninterested in the course or if a career does not precisely fit the student's abilities, problems develop. Learner counseling should contain advice on professional paths, addressing inter-personal relationships, learning strategy traits, as well as attitude and aptitude. This service is often given by counsellors or advisers with extensive expertise in the organization. However, with an increasing number of students and options, as well as the amount of work placed on these advisers who are unable to handle the situation, the faculty of higher secondary education institutions lacks appropriate expertise and experience in courses and programs other than education. Due to their busy schedules, they also do not have time to counsel their students.[20]

Consider all the methods that were used in the works that are described above, as well as to conduct a comparative analysis and show the results of the work. The method proposed consists of several stages: (1) after reading all the resources, identify what technique is used (2) what was dataset used (3) what

classification method was used (4) what the result works. To identify methods that will help me in further work, I brought out all the algorithms used, as well as the results of the work in Table 1.

Table 1. Analyze of similar works

No	Research	Goals and objectives	Strategy / Approach	Performance	Limitation and Future Work
1	Zachary A. Pardos and Weijie Jiang 2020[1]	Design serendipity for university recommendation system	Course2vec, multifactor course2vec and bag-of-words model	Unexpectedness, novelty, variety, and the major measure of serendipity were all greater in the diversity-based algorithms than in the and non-diversity algorithms.	L: Students have to rely on the semantics of the course description. FW: Additional information, such as latent semantics, should be added to course descriptions.
2	H. Thanh-Nhan, N. Thai-Nghe, H. Nguyen 2016[2]	Building the course recommendation systems.	KNN each for student and course; MF and BMF with RMSE measure	Because the BMF method has the least root mean square error, it is employed in course recommendation systems.	FW: To make better use of relational data, take a multi-relational strategy.
3	Y.Zhu, H.Lu, P.Qiu, K.Shi, J.Chambua, Z.Ni 2020[3]	Network-based offline course recommendation with graph learning and tensor factorization for heterogeneous teaching assessment	SCD, DSE, RTTF, TENTF	When compared to PMF techniques (RTTF, DSE) and SVD-based methods, the proposed model obtains the lowest MAE and RMSE.	Simplified computation method and lower the amount of time spent training and forecasting

Table 1. Analyze of similar works

4	A.Akhtar 2020[4]	Recommender system for Virtual University in course selection process	Collaborative Filtering, KNN, VU-CRS algorithm	The difference between the expected and real values stayed within 10%. For 100 observations, the MAE came out to be 5.12, which is a reasonable mistake.	More tailored findings can be obtained using a psychological test-based method.
5	A. Esteban, A. Zafra, C. Romero 2020[5]	A system with genetic optimization was developed using a hybrid multi-criteria recommendation system.	Proposed hybrid RS CBF with clustering user-based & item-based CF MCSeCF	Using a hybrid approach with multiple criteria are significantly better than the rest models (RMSE values)	Add limitations to course recommendations to filter them; expand with more degrees
6	N. Lynn, A.Emanuel 2021[6]	Explore the use of RS to assist students in selecting courses that matches to their skills and interests.	CB, CF, knowledge-based, hybrid	Hybrid RS combines other techniques advantages and was decided to be the best method	
7	M.Maphosa, W.Doorsam, B.Paul, 2020[7]	Assist to choose elective courses	Matrix factorization, Review protocol, CF	Colloborative filtering is most popular one with 45%	
8	A.O.Ogunde, E.Ajibade, 2019[8]	propose RS with dynamical advice to students for optimum academic performance	KNN to determine relation between past and available course	Overall accuracy 95.65% exceeds related works results	In the future work needs to be considered more subjects with a generic system
9	J.Lina, H.Pu, Y.Li, J. Lian, 2017[9]	Improve state-of-art-methods both accuracy and performance	sparse linear method (SLM) calculating sparseness aggregation strategy	Results of metrics as hit rate and average reciprocal Hit-Rank demonstrated that technique outperforms the related works	

Table 2. Results of similar works

Study	Technique	Dataset	Classifier	Obtained Results	
				Metrics	
Zachary A. Pardos and Weijie Jiang 2020[1]	Diversity based algorithms (BOW, Equivalency) vs Non-diversity algorithms (Equivalency, RNN)	student course enrollments at UC Berkeley from Fall 2008 through Fall 2017	BOW (div)	unexpectedness	3.550
				serendipity	3.227
				novelty	3.896
			Analogy (div)	diversity	4.286
			Equivalency (non-div)	successfulness	3.619
				commonality	4.500
H. Thanh-Nhan, N. Thai-Nghe, H. Nguyen 2016[2]	KNN, MF, BMF	Grading system at Can Tho University from 1994 to 2004 in ICT	UserKNN	RMSE	0.998
			ItemKNN	RMSE	0.862
			MF	RMSE	0.862
			BMF	RMSE	0.831
Yifan Zhu, Hao Lu, Ping Qiu, Kaize Shi, James Chambua, Zhendong Ni 2020[3]	xSVD++, DSE, TENTF	Beijing Institute of Technology, BIT-UASET evaluation system	xSVD++	MAE	8.4237
				RMSE	11.9673
			DSE	MAE	7.4385
				RMSE	10.7338
			TENTF (Proposed)	MAE	6.9133
				RMSE	9.5913
Aleem Akhtar 2020[4]	Collaborative Filtering, select neighbours, predict score	Virtual University 2600 students and 470 courses data during 4 years	VU-CRS	MAE	5.12
A. Esteban, A. Zafra, C. Romero 2020[5]	Recommendation System, Collaborative Filtering, Content-based Filtering	Academic survey of University Cordoba, CS, 95 students and 2500 ratings of 63 courses from 2016 to 2018	Proposed hybrid RS	RMSE	0.971
				nDCG	0.682
			CBF with clustering	RMSE	1.224
				nDCG	0.234
			User-based & item-based CF	RMSE	1.166
				nDCG	0.549
			MCSeCF	RMSE	1.595
				nDCG	0.112

Chapter 4

Project's components

4.1 Machine Learning and Data Science

Data Science is a set of specific disciplines from different areas responsible for analyzing data and finding optimal solutions based on it. Previously, only mathematical statistics were engaged in this, then machine learning and artificial intelligence began to be used, which added optimization and computer science as methods of data analysis to mathematical statistics. Data science is important because it combines tools, methods, and technologies to extract meaning from data. Modern organizations are overloaded with data; there are many devices that can automatically collect and store information.

This part describes which platform used to create ML models and which language used for part of machine learning.

4.1.1 Anaconda/Jupyter

Anaconda is the platform most famous for data science, which fits the needs of ML, integrated with several data sources and tools. It is an open-source Python and R languages distributive for ML, data science, analytics, scientific computing, etc. which simplifies management and deployment. Anaconda is a data analysis tool that is available for free. Depending on the operating system (Windows, Linux, or MacOS), installation is feasible. The Anaconda interface is shown in Figure 2. To run, many scientific packs require particular versions of other packs. Data scientists frequently utilize various versions of many software and isolate these versions using distinct environments. Navigator allows you to interact with packages and environments without having to input conda commands into a terminal window. You may use Navigator to identify the packages you need, install them in your environment, execute them, and update them.

Anaconda Enterprise is the commercial version of Anaconda. This allows businesses to create enterprise-grade, scalable, and secure apps. However, you may install python and then optionally install programs using pip to execute Data Science operations. Anaconda is an option that installs all of the required packages all at once. Users will find it more convenient as a result of this. Both strategies accomplish the same goal. Developers are free to select any of them based on their preferences. Anaconda is widely used in the data science field since it solves many common difficulties early on in the development process as well as later on. Anaconda, in general, makes data science and machine learning activities easier.

Conda is a package and environment management system that runs on different operating systems, like Windows, macOS, and Linux. Conda helps packages and their dependencies install, run and update promptly. Conda easily constructs, keeps, loads, and switches between environments on the platform. Initially is intended for Python language, but with updates, any language can be packed and distributed. It is supported in all Anaconda versions [11]. Conda is a free and open source programming framework and environment manager. Conda installs, executes, and updates packages and their dependencies in a matter of seconds. On your local system, Conda makes it simple to build, save, load, and switch between environments. It was designed to package and deliver Python applications, although it may package and distribute software written in any language.

Conda is a package manager that aids in the discovery and installation of packages. You don't need to switch to a separate environment manager if you need a package that requires a different version of Python because conda is also an environment manager. You can build up a fully new environment to run this alternative version of Python while still running your regular version of Python in your regular environment with just a few keystrokes. All versions of Anaconda and Miniconda come with the conda package and environment manager. Anaconda Enterprise, which allows local administration of enterprise packages and environments for Python, R, Node.js, Java, and other application stacks, is also a component of Conda. Conda is also available on the community channel conda-forge. You can also acquire conda from PyPI, although that method isn't as up to date.

JupyterLab is a flexible, integrable, and easily extensible environment that supports simultaneous work with several Jupyter notebooks, text files, datasets, terminals, and other components, and can organize documents in the workspace in a convenient manner using tabs and separators. JupyterLab supports the visualization and editing of many data formats: CSV, JSON, IPYNB, etc. It also provides an adjustable interface with the ability to move blocks and layouts in a convenient order, with a focus on the interactivity of the calculations [12].

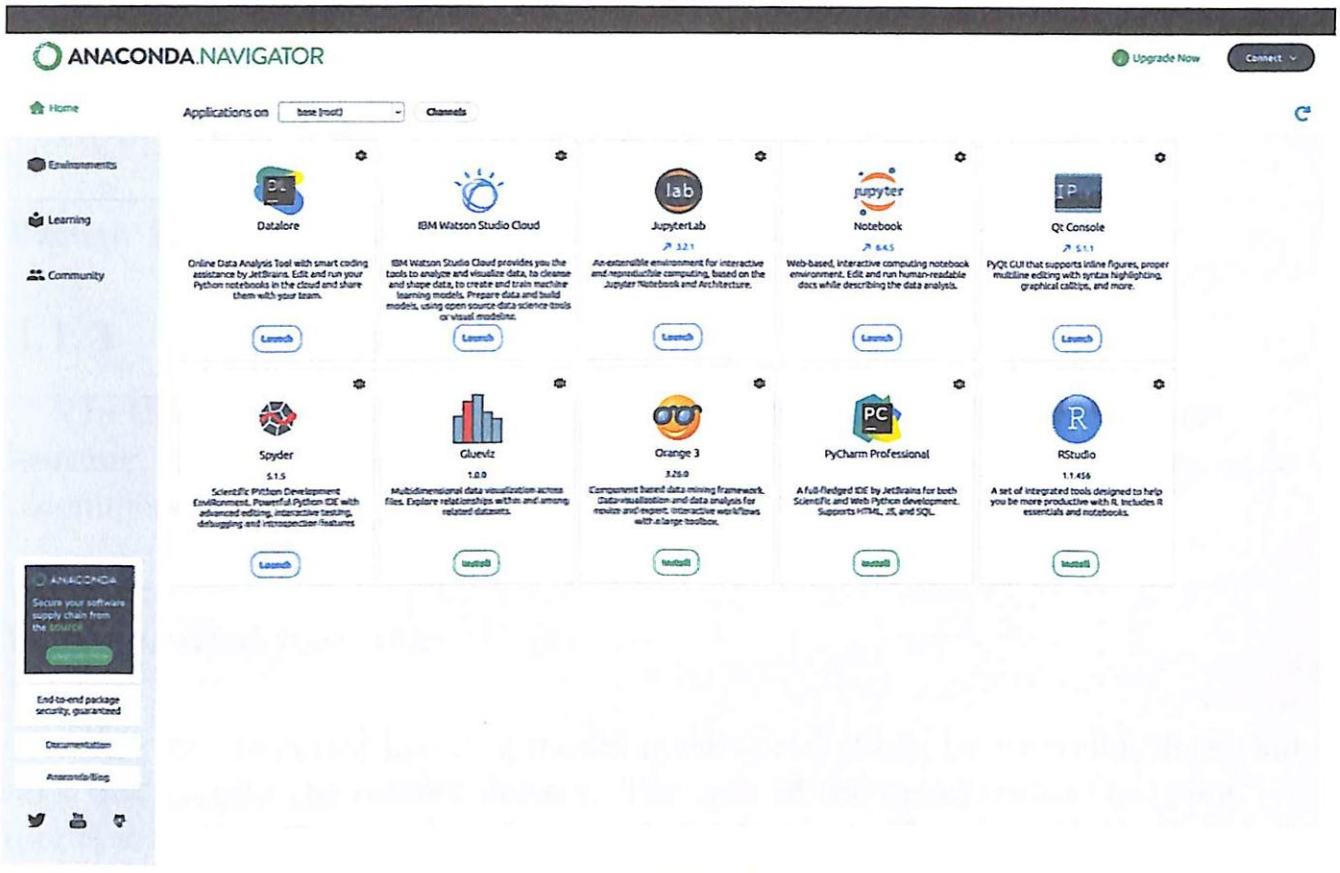


Figure 2. Anaconda Navigator interface

4.1.2 Python

Python is a high-level programming language. One of the advantages is that can be easy to pick up both for first-time beginners and advanced programmers with other languages. Python is a multi-paradigm, object-oriented, structural, generic, functional, and meta-programming are fully supported. Basic support for aspect-oriented programming is provided through metaprogramming. Many other techniques, including contract and logic programming, can be implemented using extensions. Language licenses are written in a such way, that everything is open and free, everyone can use it, develop and spread it among other developers for commercial use. Owner company already presents thousands of open libraries which were developed not only by themselves but also by community participants. This is a great opportunity for developers with an infinity of possibilities. [13]

Python is a widely used high-level, general-purpose, dynamic programming language that is interpreted. Its design philosophy prioritizes readability, and its syntax enables programmers to express concepts in fewer lines of code than in languages like C++ or Java. The language has constructs that make it possible to write clear programs on a small and big scale.

Figure 2. Python Logo

4.1.3 Algorithms

In this work, used the recommendation system algorithms of machine learning. These algorithms learn from input data and then use this learning to recommend courses.

Unsupervised learning

The unsupervised learning model makes predictions by receiving data that does not contain the correct answer. The goal of the unsupervised learning model is to identify meaningful patterns in the data. That is, the model has no clue as to how to classify each piece of data, but instead, you have to derive your own rules. The commonly used unsupervised learning model uses a technique called clustering. The model finds data points that represent natural grouping. [18]

Unsupervised learning, also called as unsupervised machine learning, uses machine learning algorithms to evaluate and cluster unlabeled data.These algorithms find hidden patterns or data groupings without the need for human input. It's ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and picture and pattern recognition because of its ability to detect similarities and contrasts in data. Two common approaches for lowering the number of features in a model through the dimensionality reduction process are principal component analysis and singular value decomposition. Unsupervised learning employs neural networks, k-means clustering, probabilistic clustering techniques, and other algorithms. [19]

Cluster is a type of unsupervised learning strategy that widely used, investigated on the field of vision software for computers. There hasn't been much work done to modify it for end-to-end visual feature training on huge datasets. DeepCluster is a cluster algorithm that learn both parameters of a neural net are discussed, as well as the cluster assignments of the produced features. DeepCluster employs a typical clustering technique, k-means, to iteratively group the features and uses the following assignments as supervision to update the network's weights. On huge datasets like ImageNet and YFCC100M, we use DeepCluster to train convolutional neural networks unsupervised. On all typical benchmarks, the generated model exceeds the

present state of the art by a wide margin.[20]

K-means

K-means is an unsupervised learning algorithm that divides data into clusters, grouping unlabeled inputs into the most similar ones. K-means is a distance-based technique, where distances are calculated to designate a point to a cluster, where each cluster is associated with a centroid (Figure 3). Its main purpose is to reduce the distance between points and the relative center. Clustering is different from classification because categories are not user-defined. For example, an unsupervised model can group meteorological datasets based on temperature and reveal seasonal segmentations. You can then name these clusters based on your understanding of the dataset.[18] Clustering can solve many data science problems and be applied in the fields of recommendation systems, segmentation of the market, grouping search results, analyzing social networks, and so on proposing centroid, distribution, connection, and density models.

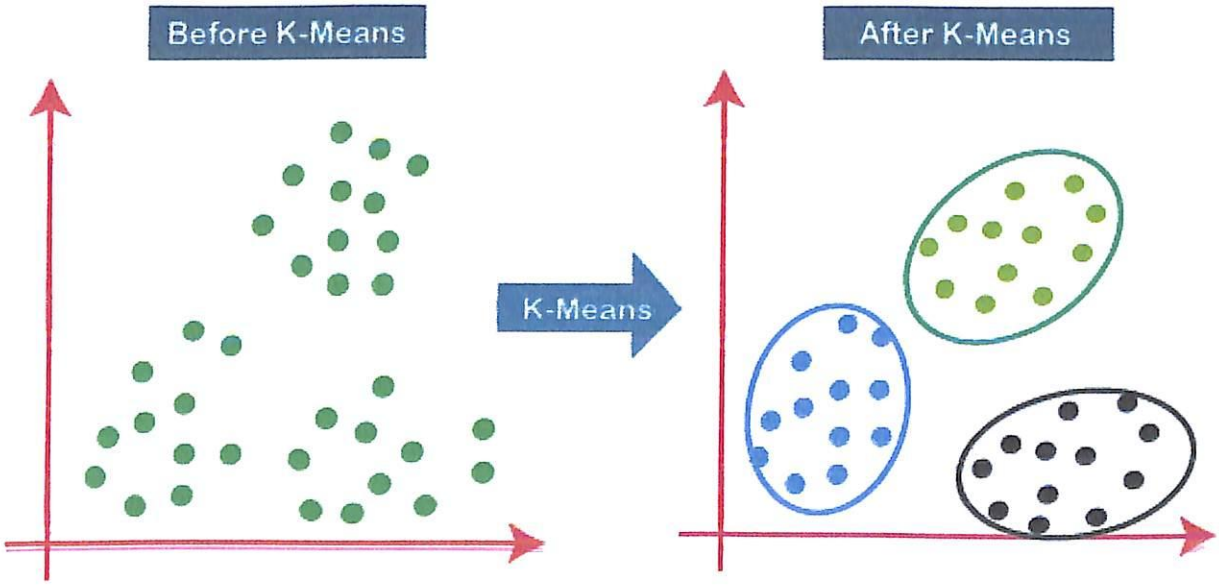


Figure 3. K-Means technique explanation on graph [15]

Partitional clustering is the most basic type of clustering, and it seeks to split a given data set into disjoint subsets (clusters) in order to maximize specific clustering parameters. The cluster error criteria, which computes each point's squared distance from the appropriate cluster center and then adds these distances for all points in the data set, is the most often employed. The k-means algorithm is a prominent clustering approach that reduces clustering error. The k-means method, on the other hand, is a local search technique with the well-known problem that its success is highly dependent on the basic starting circumstances. Several more solutions based on stochastic global optimization

methods have been developed to address this issue (e.g. simulated annealing, genetic algorithms). However, it should be emphasized that these strategies have not received widespread popularity, and in many practical applications, the k-means method with multiple restarts is employed instead.[]The global k-means clustering algorithm Aristidis Likasa; , Nikos Vlassisb, JakobJ. Verbeekb aDepartment of Computer Science, University of Ioannina, 45110 Ioannina, Greece bComputer Science Institute, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands Received 23 March 2001; accepted 4 March 2002

The Elbow (also known as Knee) approach is a useful visualization tool for determining the optimal number of clusters, at a moment where the errors are considerably reduced. This method can be visualized by drawing a curve between the sum of square errors (SSE) and the number of clusters (Figure 4), there will be a sluggish change in the value of SSE at that point that can be considered as the final number of clusters.

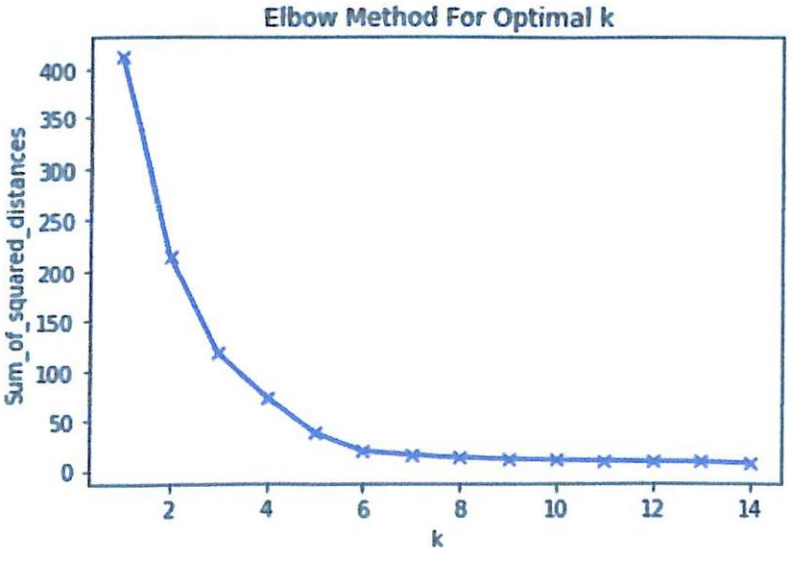


Figure 4. Elbow approach example

LDA (Latent Dirichlet Allocation)

LDA is a well-known modeling method to extract topics from a given corpus. The phrase latent refers to something that exists but is not fully formed. LDA can be characterized as a generative probabilistic model for a set of discrete textual data. One application of LDA in machine learning - specifically, topic discovery, a subproblem in natural language processing - is discovering topics in a set of corpora and then automatically classifying any individual document within the set in terms of how "relevant" it is to each of the discovered topics. A topic is a collection of keywords (individual words or phrases) that collectively express a single concept.

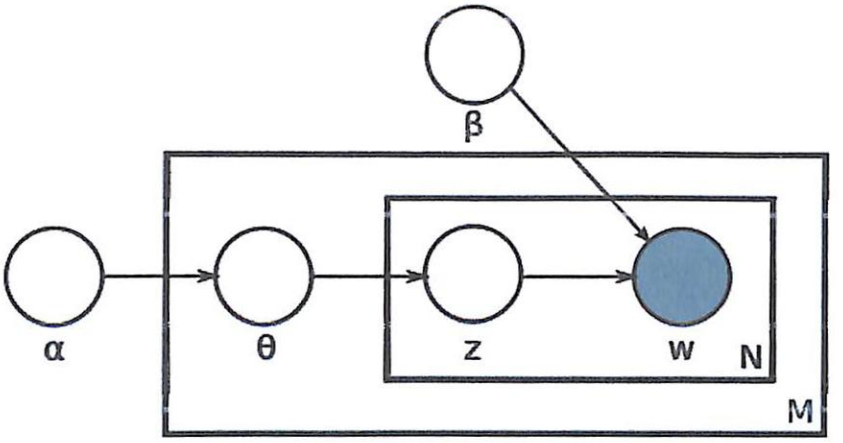


Figure 5. LDA model representation

Plate notation, which is often used to express PGMs, may be utilized to describe the dependency among the many variables in a succinct manner. The boxes are "plates" that symbolize replicates, or entities that echo themselves. The exterior plate shows documents, while the inner plate illustrates the document's repeated word state; each state is related to a topic and term choice (Figure 5). The following are the variable names:

- M indicates count of documents
- N is a size of terms in that document
- α indicates subject allocation of Dirichlet prior parameter per document.
- β indicates word allocation of Dirichlet prior parameter per document.
- θ is the subject allocation for document
- z is the subject of specific term in document
- w is exact term

4.2 Application

4.2.1 Android studio

Having considered all the strengths, it was decided to write our application on a platform called Android Studio covering all users. Team IntelliJ IDEA by a well-known corporation released Android Studio IDE (Integrated Development Environment), which originally was specially constructed for the Android platform to develop apps, but now Flutter is also available. The main reason is that both techniques were released and continue to be developed by Google. Since Android Studio has experienced a large number of updates, additions, and revisions, and now it is an enormous forceful code editor, it represents the best sandbox along with developer tools. To develop applications AS offers lots of features that improve efficiencies, such as a rapid and feature-rich emulator, an

environment that can be developed not only for android devices, easy changes with GIT, code manipulation, and modification of resources during runtime wi a hot restart. Last year, ide improved dramatically, including Jetpack Compos and Flutter frameworks. As we know Google has a bunch of technologies and methodologies in a Google Cloud Platform and Firebase, and of course, all of these are integrated to make it easy to usage and be user friendly.

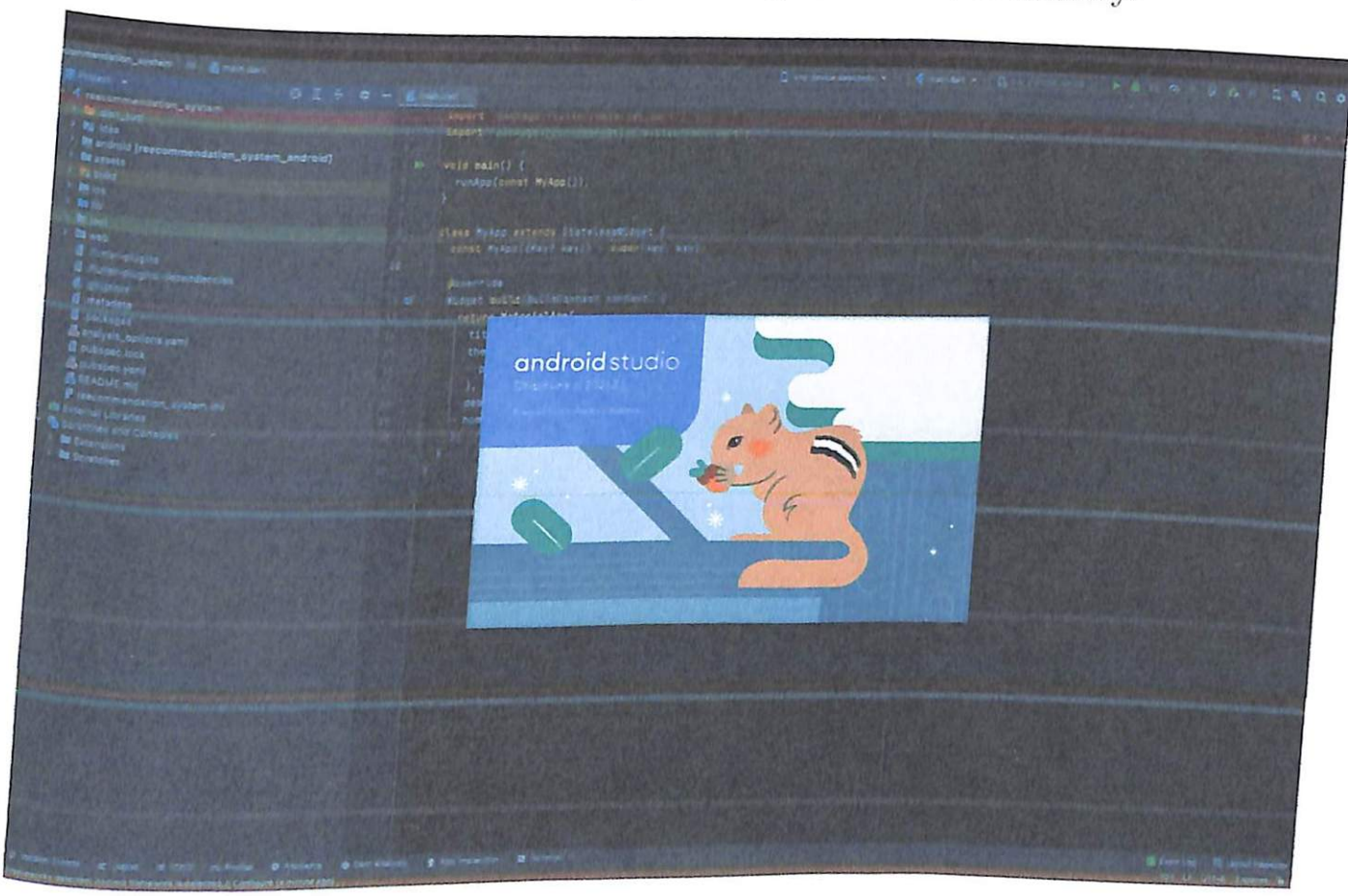


Figure 6. Android Studio interface

After creation of new project on Android Studio, it displays project files by default in the Android Project View, which can be changed by selection Project level view. These structures is divided into modules to allow quickly access the important source files for your project[16]. Figure 6 illustrates interface of Android Studio, this IDE was chosen to run Flutter app. In the picture structure of project also shown:

- pubspec.yaml is file where all libraries and environment settings is listed
- assets folder contains images, data and other raw files
- android, ios, web folders contains native settings of each component
- lib folder consist of main flutter files written in dart language for all components
- and other project build gen files

4.2.2 Flutter

Recent years crossplatform working frameworks have been attracting, giving solution to write one code to different platforms. Google announced their own product Flutter to developers, and this framework can do many things in one moment. To develop one product used to hire at least 3 different native code developers: android, ios, and front-end. But using Flutter all of this problems will be solved by one developer, also beautiful and adaptive, speed-animated UI. Nowadays, framework is going forward giving possibility write additionally on linux, macos, windows desktop platform(Windows desktop support is now available on the stable channel; macOS and Linux support is still in beta, although a beta snapshot is accessible on the stable channel). Flutter makes it easier to get started with app development. This minimizes the cost and complexity of developing cross-platform apps by speeding up application development. Flutter acts as a blank canvas for creating a high-quality user experience, and as a base uses Dart programming language.

Flutter framework realising each week a lot of technologies, so there will be no problem with OOP, classes, loops, conditions and etc. Even if developer needs a library, but it doesn't support Flutter yet, it can be easily solved by using native appeal to Android/Kotlin and IOS/Swift relation. Also it support ffi to run much quicker using C/C++ methods.

Flutter consists of the following elements:

- For mobile devices, a highly efficient 2D rendering engine with good text support.
- Modern framework in the React style
- For unit and integration testing, a robust set of widgets that implement
- Material Design and iOS style API.
- Connecting to the system via interop and API plugins, as well as third-party SDKs
- Test support for desktop platforms (Windows, Linux, and Apple)
- Developer Tools for testing using debug points, and profiling application on memory and network
- CI/CD tools for automatizing of testing and compiling

Android NDK was used to compile the C and C++ engines code. Dart code is pre-compiled (AOT) into native, ARM, and x86 libraries (both the SDK and yours). These libraries are part of the Android 'runner' project, and they're all

included in the.apk file. The Flutter library is loaded when the program starts up. Any rendering, input, or event processing, for example, is outsourced to Flutter and application code that has been built. Many gaming engines function in a similar way.[13]

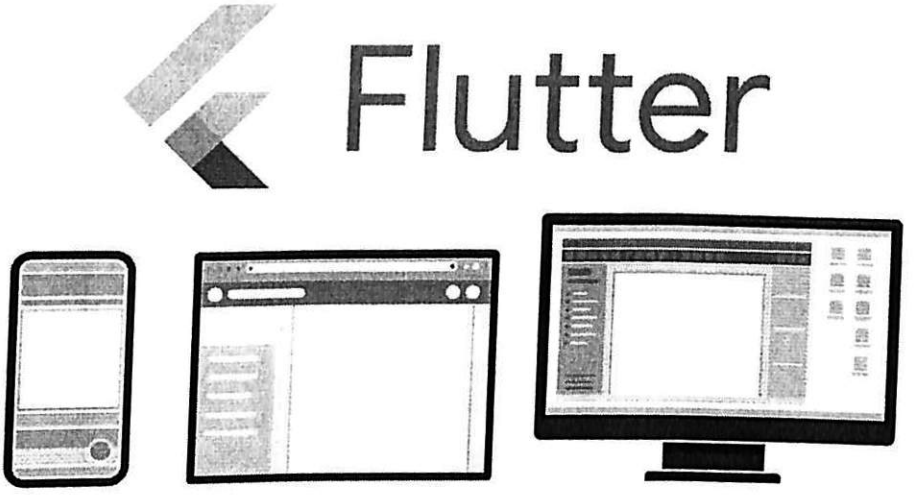


Figure 7. Flutter Logo

4.2.3 Dart

In last decade Google started to invent a new language which supports cross platform, also works much faster and high level programming language. They introduced Dart language which was written on C based language to process quickly, and it is client based solution. Languages are characterized by their technical shell, which consists of design decisions that form the language's abilities and power. Dart is constricted for a technological shell that is especially suited to client-side programming, focusing both development (hot reload with statefulness in less than a second) and high-quality production across a wide range of compilation targets (web, mobile, and desktop).

Flutter's backbone is likewise Dart. Dart not only offers the language and runtime for Flutter apps, but it also helps developers with formatting, parsing, and testing their code.

Dart is type safe, and it employs static type checking to verify that a variable's value always matches the static type of the variable. Sound printing is a term used to describe this process. Type annotations are optional owing to type inference, although types are necessary. Dart's typing system is very versatile, enabling you to utilize a dynamic type in conjunction with run-time checks, which is ideal for experimentation or work that requires a lot of movement. Dart comes with a large number of core libraries that cover a wide range of programming tasks:

- Every Dart application comes with built-in types, collections, and other essential features (dart:core)
- Queues, linked lists, hash maps, and binary trees are more advanced collection types (dart: collection)
- Converting between multiple data formats, such as JSON and UTF-8, with encoders and decoders (dart:convert)
- Random number generation and math constants and functions (dart:math)
- Non-web apps can use file, socket, HTTP, and other I/O methods (dart:io)
- Future and Stream classes provide asynchronous programming support (dart:async)
- Fixed-size data (e.g. 8-byte unsigned integers) and SIMD numeric types (dart:typed data) are effectively handled by lists.
- External function interfaces represent a C-style interface for dealing with other programs (dart:ffi)

Figure 8. Dart Logo

4.2.4 Flask

Flask is a compact and lightweight framework with handy instruments and possibilities that constructing networked applications simple. Even beginner developers feel free and might simply build website using only Python language. Framework is also expansible, and it doesn't tie any particular directory structures or boilerplate code to get up and running. Flask utilizes the Jinja templating engine to produce HTML pages dynamically using Python concepts like variables, loops, and lists. These templates will be used as part of this project. Other applications written using Flask is more understandable and clear than alternative framework Django. Moreover Flask provides functions as URL routing handling and site rendering. It also referred to mini framework since doesn't work with database and authentication. Instead, Flask extensions, which are Python packages, provide this capability. Extensions work in tandem

with Flask and appear to be a component of the framework. Flask, for example, does not provide a page templating engine, although the Jinja templating engine is included in the default Flask installation. We commonly refer to these settings as being part of Flask for convenience. Flask is very straightforward to learn for a newbie because there isn't much boilerplate code to construct and operate a small application.

In this project, Flask added as server part of recommendation system, which will provide API to client side apps. Recommendation function added to the class as network routing using POST method. It accepts two parameters in JSON type sent in requests body part: course_id and interest. When queried request with the first parameter , then recommendation function was triggered to predict by course id, and a list of courses was sent in the response. The same situation with the second parameter, but the only difference is now recommendation by interest of user is generated. If both parameters are missing or empty, then the response wrapped in an error will be returned. After that, code was committed and pushed successfully to cloud based web-hosting server Heroku. (See Appendix A, Figure A-3)

Figure 9. Flask Logo

Chapter 5

Implementation

5.1 Data

The data was taken from an open source as a basis collected by Pluralsight last 20 years [7]. In Table 3, the top 5 rows displayed are sorted in descending order by the date the course was released. It consists of 13384 rows and 7 columns about the course, with following features:

- CourseId - unique identifier for every course
- Course Title - name of the course
- Duration In Seconds - contuation of course in seconds per week
- Released Date - course first announced date
- Description - detailed explained information about the course
- Assessment Status - course position for evaluation, values as Live or None
- Is Course Retired - information if the course is out-of-date, valued as yes or no

Table 3. Courses data first 5 rows

CourseId	CourseTitle	DurationIn Seconds	ReleaseDate	Description	AssessmentStatus	IsCourseRetired
fostering-effective-team-collaboration-communication	Fostering Effective Team Collaboration and Communication	9320	2021-12-10	This course is for any manager who is interested in building a communication plan and leading a self-managing team. In this course, Fostering Effective Team Collaboration and Communication, you will learn scenarios and techniques to overcome ...	None	no
sap-mm-fundamentals-beginners	SAP MM Fundamentals for Beginners	3496	2021-12-10	Learn the basics of the SAP Materials Management (MM). The course consists of basic overview topics in material master, master data, procurement, and inventory management. With graphics and through demos in a...	None	no
windows-server-2022-intro-exam-az-800-cert	Introduction to Exam AZ-800: Administering Windows Server 2022 Hybrid Core Infrastructure	2375	2021-12-10	The Windows Server Hybrid Administrator Associate certification validates your systems administration skills with Windows Server in both on-premises and Microsoft Azure cloud environments. In this course, Introduction to Exam AZ-800: Administering Windows Server 2022 Hybrid Core Infrastr,..	None	no
dot-net-6-bcl-big-picture	.NET 6 BCL: Big Picture	1510	2021-12-10	You can create different types of applications and libraries with .NET 6. In this course, .NET 6 BCL: Big Picture, you'll learn how the BCL lets you reuse capabilities across application types. First, you'll explore how the .NET 6 BCL is differe...	None	no
getting-started-google-kubernetes-engine-6	Getting Started with Google Kubernetes Engine	10212	2021-12-09	In this course, each module aims to build on your ability to interact with GKE, and includes hands-on labs for you to experience functionalities first-hand. In the first module, you'll be introduced to a range of Google Cloud...	None	yes

5.1.1 Data analyze

Before starting to work with the data, it was decided to investigate them. This is necessary to get rid of noisy data and select columns that will help with evaluation and that will participate in building an efficient model. I need to understand when courses started to realize and how many courses have been released annually. Figure 10 shows that the first courses were implemented in 2003 with the amount of 13 subjects and every year the number of courses increased. In 2013, the number of announced courses increased to 1000, and until 2018 it was changed negligibly. The maximum number of courses reached a peak in 2019-2020, assumed when the whole world switched to online learning. The pie chart in Figure 11 illustrates the ratio of actual courses to the retired courses with just over a half, which means it is necessary to drop a significant fraction of the courses from the list.

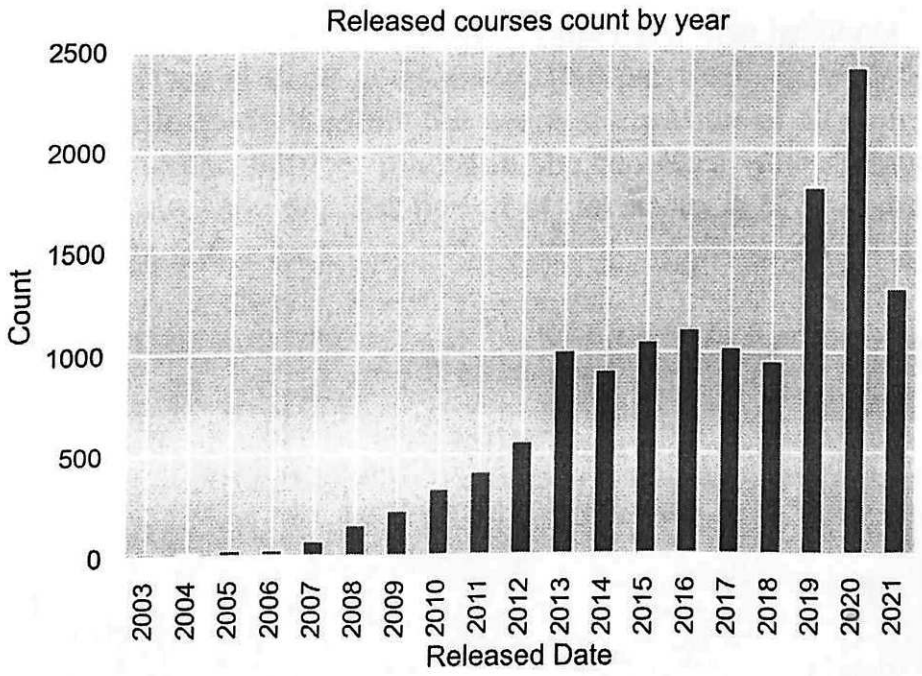


Figure 10. Released courses quantity by year

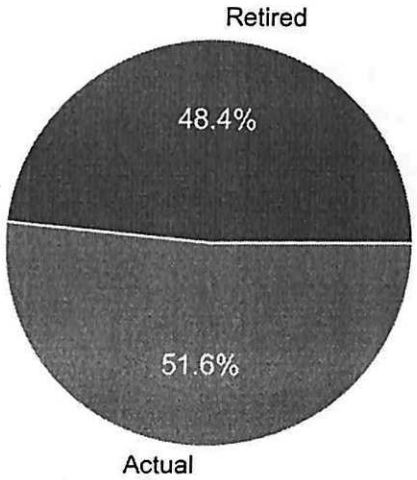


Figure 11. Comparison of actual and retired courses

The importance of the duration of the course can also influence the choice of the course, how often student can devote time per week. The histogram in figure 12 presents information about the average duration of all courses which is 7170 seconds, the median number placed in the center of values is equal to 6328 seconds and the mode - the popular period of durations is 5751 seconds or equivalently an hour and a half.

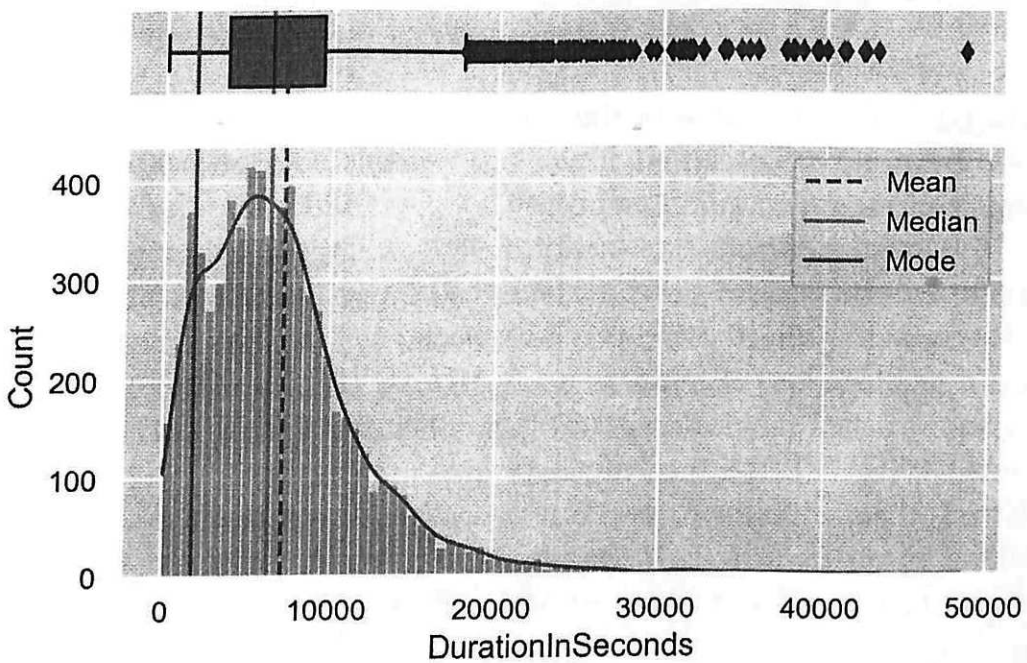


Figure 12. Statistical data for the duration of courses

5.1.2 Data preprocessing

A. Data Cleaning. It increases the quality of data and makes it easier to get useful conclusions from data. Cleaning and arranging raw data to make it acceptable for creating and training machine learning models are referred to it. The following steps describe the way of normalizing data cleaned from noisy and missing data:

- Drop retired courses as explained on part 3.3
- Combine course id, title and description into one text
- Convert all letters to lower case
- Replace 'll with empty space
- Remove all numbers
- Remove all punctuations and characters
- Remove all null values
- Remove stopwords - common used phrases
- Lemmatization - remove the endings from the words and return the basic form of the word, known as the lemma.

After these steps amount of rows was halved and now it is equal to 6908 rows. Further, data became cleaner and might be passed to the next step

B. Vectorizing - transform text to vector numbers to make the calculation easy. Machine learning models accept numbers and matrices instead of texts or other types of data, since they all use mathematical computational operations. In this research two approaches being used to achive it: Bag of words (BOW) and TF-IDF. Bag-of-words, as explained by written, is representing a bag where collected words that indicate text for various calculations, feature represented as frequency dictionary - values appeared count in content. TF-IDF (term frequency - inverse document frequency) is a metric which rates importance of words in corpus, document and used to analyze textual information. After the texts are converted into numbers, data can be passed to the model and make a recommendation system.

5.2 Model

Now it's time to build models using preprocessed data, and first of all, I use the K-means technique (as was written in section 3.2.3, this model divides data

into clusters). To run K-means, the necessary class was imported from sklearn library, and it requires paramcters such as cluster size, iteration size, and initial centroid number. To begin with, the model has been fitted with data and run with default values - 8 clusters, 300 iterations, and 10 initial centroid seeds respectively. [21] Although my task was to find an optimal number of feature parameters to construct an efficient recommendation system algorithm. To achieve this goal, I iterated the model on different clusters from 1 to 30, the result was measured by the SSE metric. Experiment with K-Means model held with data which was vectorized with different technologies: TF-IDF, BOW.

Consequently, another approach, called LDA (Latent Dirichlet Allocation) used to run data, class was imported from gensim library. LDA model also was trained using technologies like TF-IDF and BOW, additionally Trunkated SVD (Singluar Value Decomposition) matrix. LDAMulitcore class takes as a paramater this technologies as well as number of topics, parallel works size and pass count.

5.3 Evaluation

Trained K-means and LDA models and a bunch of experiments need a quality estimation. Like the previous point evaluation will be started on k-means model. The line on Figure 13 below illustrates sum-of-squared errors change over cluster size. With regards to the count of clusters, SSE falling dramaticaly to about 800 unit. This illustration is also called Elbow method, where SSE is decreasing slightly is a valuable point to choose final cluster size. Moreover, the Table 4 compares different cluster parameters and according to the results of the error, we can say that 30 clusters show better result. Hence, top keywords for each cluster were displayed to understand that is truly result and Table 5 shows 7 best terms of first 5 clusters.

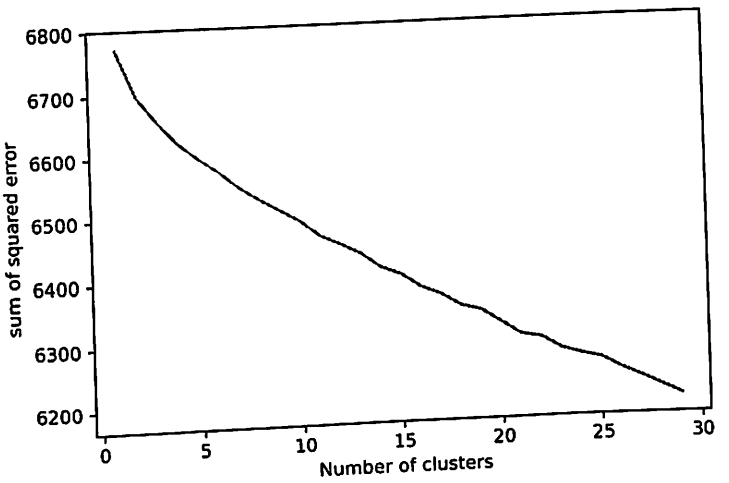


Figure 13. SSE ratio to cluster size (Elbow method)

Table 4. Model comparison

Parameters		
Cluster size	8	30
Iteration count	100	500
Centroid initialization	8	15
SSE	7583.8	7015.9

Table 5. Top of the list of clusters results

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
modeling	network	azure	data	test
tools	router	cloud	hadoop	studio
rendering	protocol	storage	science	unit
components	docker	virtual	big	automated
effects	access	aws	course	framework
animation	ccnp	services	analysis	penetration
techniques	comptia	database	visualisation	verification

Apart from that, LDA model also was trained with different corpuses and model outputs coefficents of each term by the related topics (Figure 14).

```
Results predicted from BOW model.

Score: 0.5170942544937134
Topic: 0.034*"data" + 0.020*"play" + 0.016*"creat" + 0.013*"work" + 0.012*"fundament"

Score: 0.46329399943351746
Topic: 0.062*"data" + 0.023*"design" + 0.022*"build" + 0.015*"youll" + 0.013*"visual"

Results predicted from TF-IDF model.

Score: 0.664462685585022
Topic: 0.019*"azur" + 0.010*"youll" + 0.009*"microsoft" + 0.008*"databas" + 0.008*"data"

Score: 0.2984499931335449
Topic: 0.010*"azur" + 0.009*"youll" + 0.009*"data" + 0.008*"secur" + 0.007*"network"

Score: 0.023114796727895737
Topic: 0.012*"azur" + 0.007*"secur" + 0.007*"youll" + 0.007*"oracl" + 0.007*"manag"
```

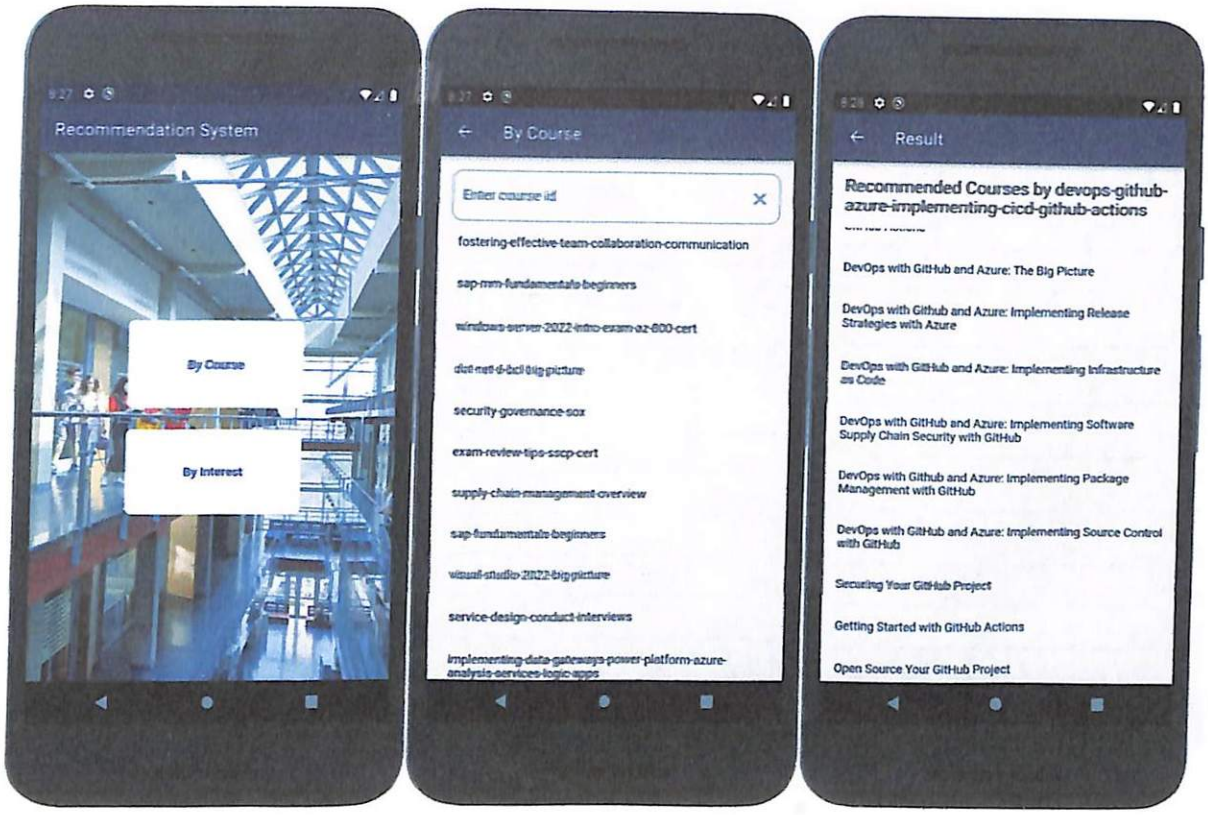
Figure 14. Coefficients score for BOW and TFIDF approaches

5.4 Prediction Visualization

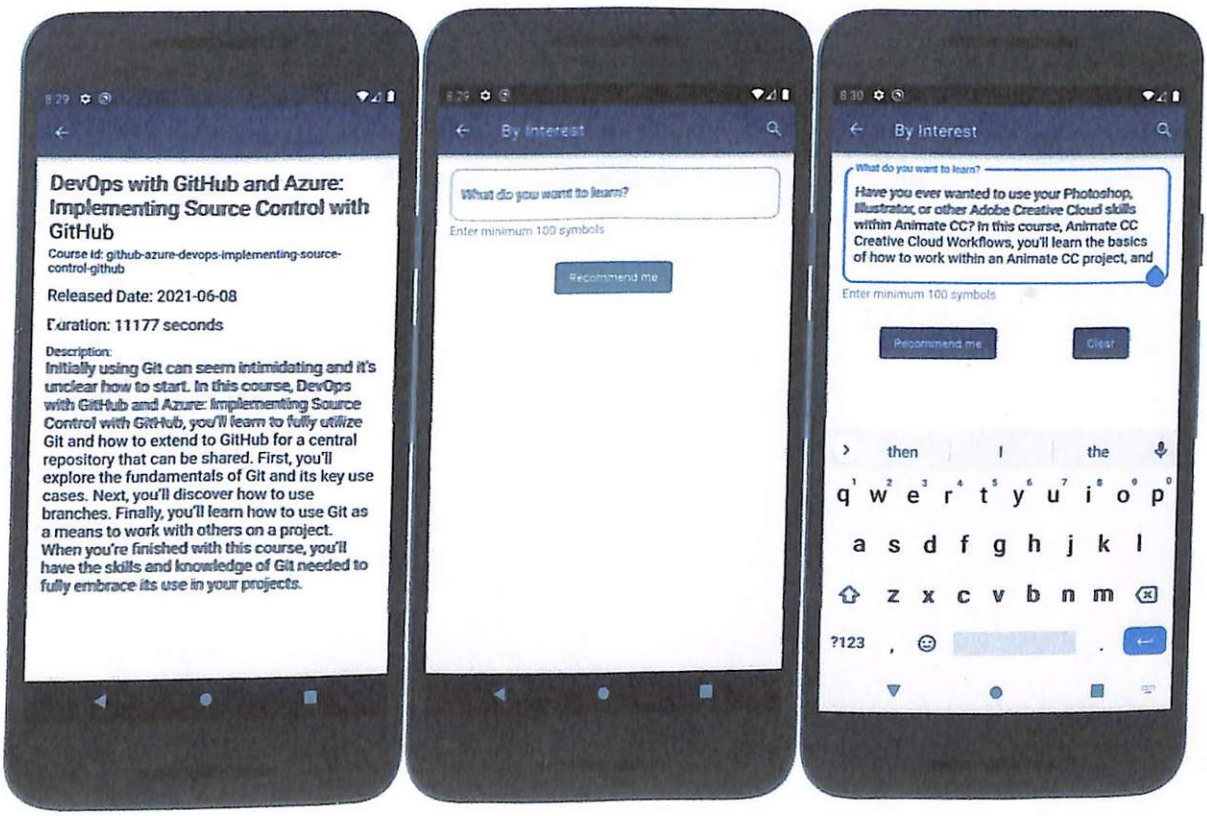
In addition, decided to make an application on different platforms such as Android, IOS, and website-browser where other people can write their interests and get recommendation courses by this system. Also, the application will

recommend subjects by course id, if a student wants to get similar subjects. To accomplish this idea model is necessary, but the model was written using Python language on Anaconda/Jupyter distributive and the application will be written in other languages. So the model will be saved using the pickle serialization tool for future usage, also each course will be checked by recommended cluster for comparing incoming applications. The recommendation will be provided to an application by API written also in Python, but with the help of Flask framework which gives easy backend query handle. To visualize prediction chose the Flutter framework as it allows to write one code and build one UI using Dart language for every platform listed above (See part Tools). Summarizing the above, user actions will be as follows (Figure 15 - Figure 23):

1. The application gives choice to recommend by interest or similar course
2. If the user chooses a similar course, a list of courses will be displayed with a feature to filter by searching id of course.
3. When the user selects the course he wants, API will obtain results by recommendation model and displays a list of 10 items
4. By selecting any course full description will be shown to the user
5. If the user wants a recommendation by interest, he needs to write text with at least 100 symbols, and request an API, hence steps 3 and 4 can be repeated



Step 1 Step 2 Step 3



Step 4 Step 5 Step 5

Figure 15. Recommendation application steps on Android

Figure 16. Recommendation application on Web. Step 1

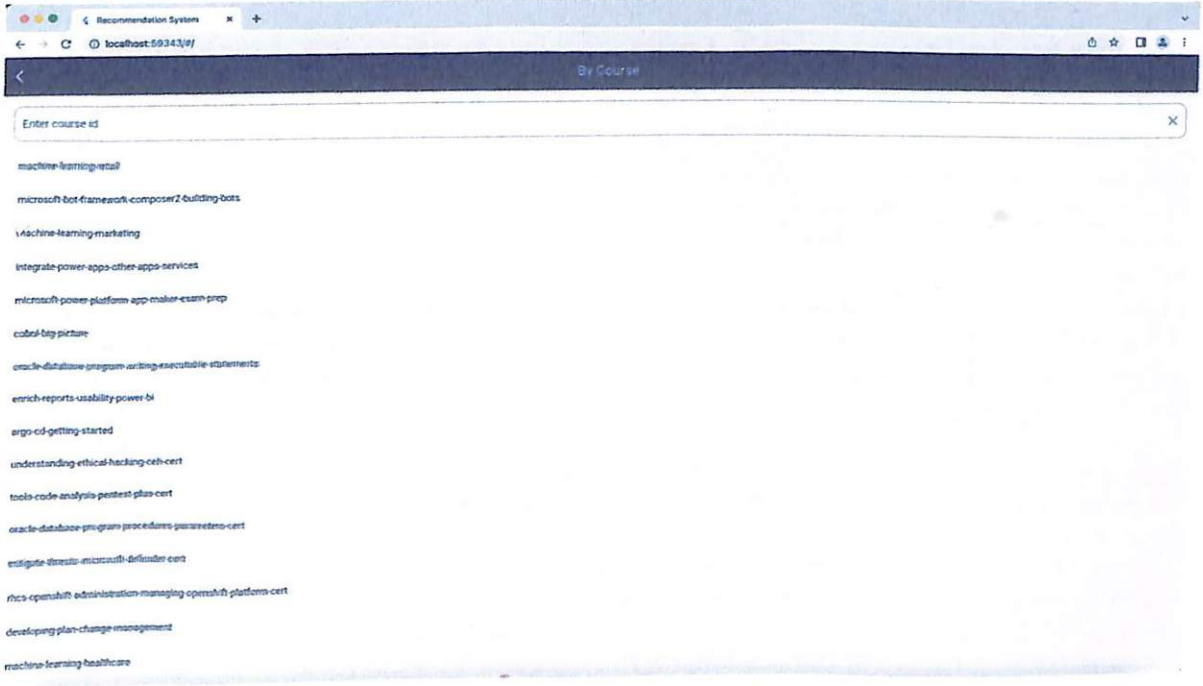


Figure 17. Recommendation application on Web. Step 2

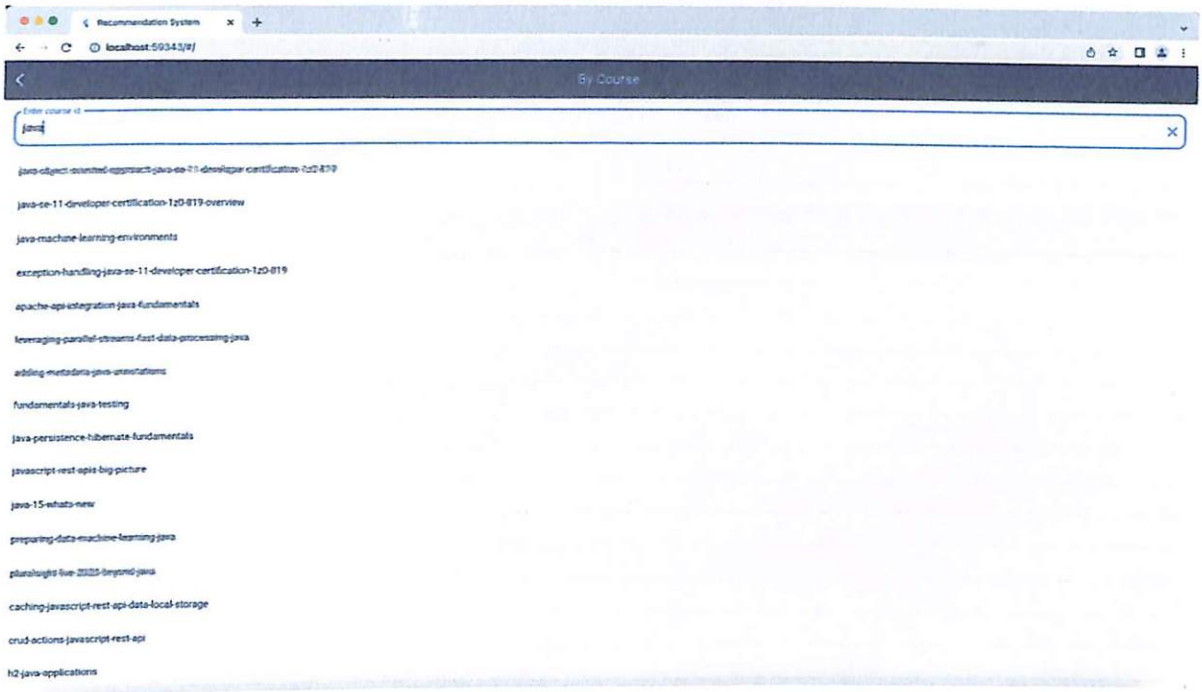


Figure 18. Recommendation application on Web. Step 2

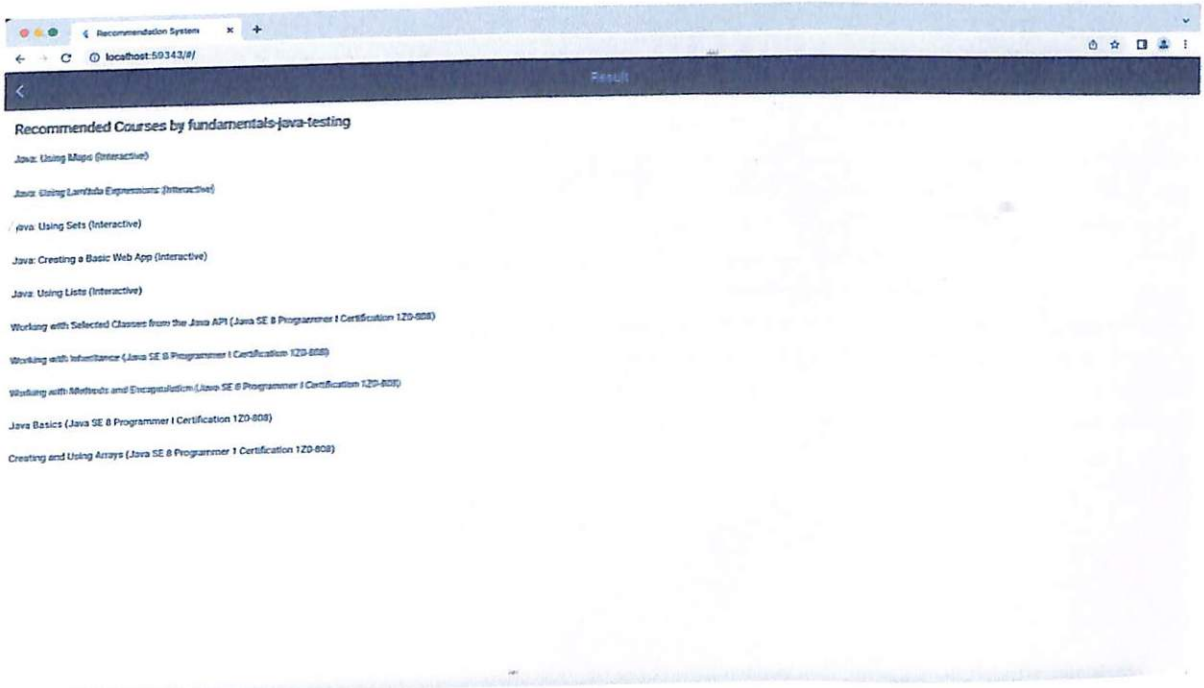


Figure 19.Recommendation application on Web. Step 3

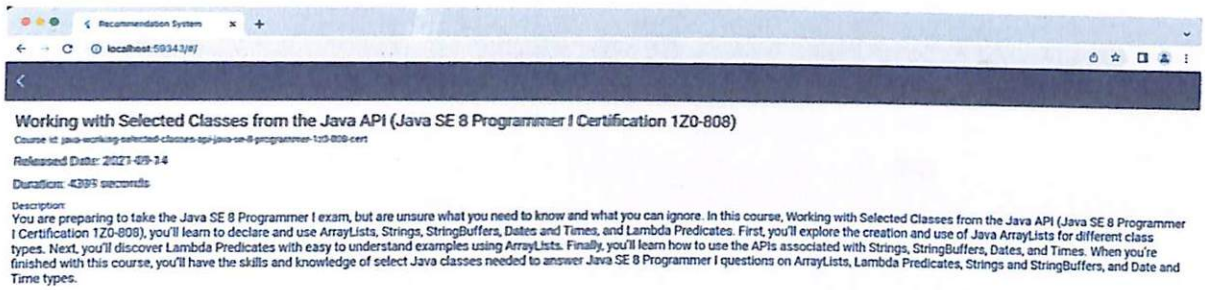


Figure 20. Recommendation application on Web. Step 4

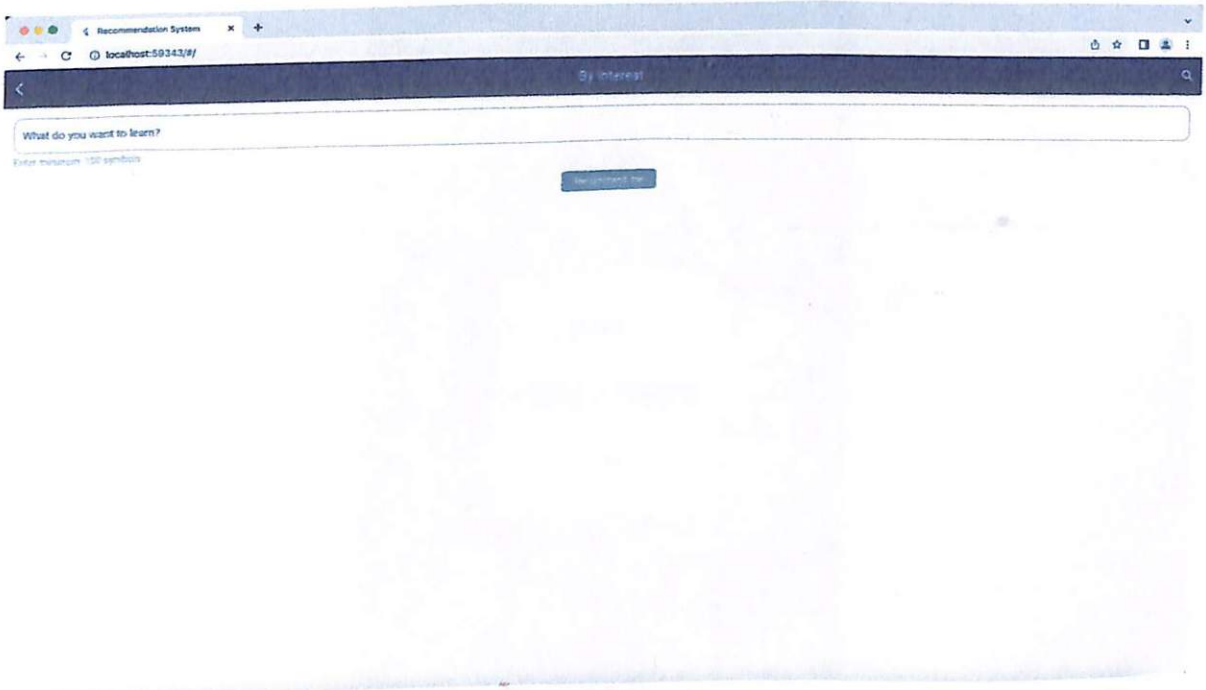


Figure 21. Recommendation application on Web. Step 5

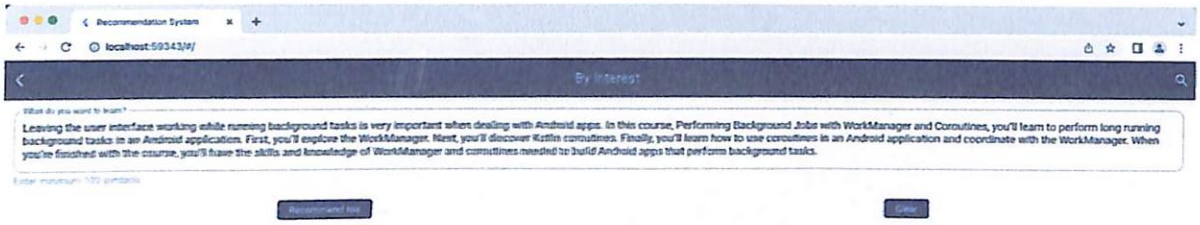


Figure 22. Recommendation application on Web. Step 5

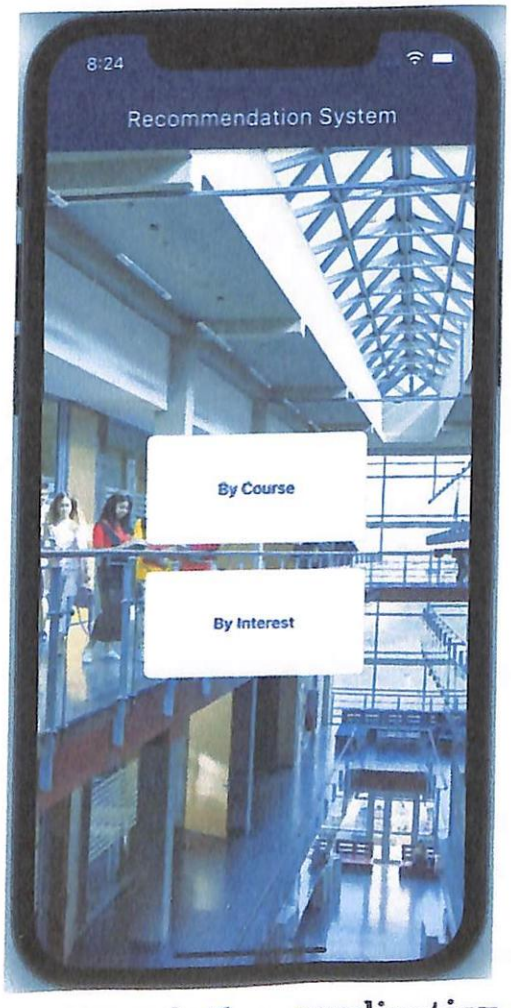


Figure 23. Recommendation application steps on IOS

Chapter 6

Conclusion

This chapter summarizes the whole research with its matter findings related to aims and motivation. The goal of this work was to make a recommendation system that offers courses to students based on previous subjects or their interests. An intuitive, user-friendly, and adaptive application was designed and built to achieve the purpose, it was constructed on multiple platforms (IOS, Android operating system mobile phones, and Web-based browser) so that the student could use it on any device.

Based on all of the above, we are confident that creating a university recommender system is significant yet nowadays. This task will help students find a course that suits their abilities by simply writing down the notes. Research work analyzed many similar articles to have a complete understanding and study their problems and limitations. The investigation method is to determine a more accurate cluster and output the corresponding rate, also improving the previous method to get a more detailed clustering model. K-means methodology was applied with multiple parameters and experiments to divide all courses into corresponding clusters. Dataset went through multiple processes to be norimalized, after that TF-IDF and Bag-of-words approaches are used to vectorize textual data into numerics. Data went through multiple cluster sizes and also experimented by adding the TruncatedSVD model. LDA. Then it passed to LDA technology and the K-means method to learn to cluster from data. These models showed almost similar results, and k-means presented slightly better outcome with 30 clusters and sum-of-squared errors metric. Elbow method helped to choose optimal number of cluster size, and subsequently were chosen to solve further recommendation issues on application.

This model works well when you need to predict a course category with a fairly complete description of the subject and a description of the student's key skills, as the sum of connectivity and root-mean-squared depends on it. Courses with a fewer frequency not being able to correspondent satisfying comparison. Hence in future work, there is an idea to improve by gathering key phrases of each sphere. In addition, I would like to analyze the dataset from the University

of Suleyman Demirel and add a script of recommendation to their portal and ensure the recommender system works well on the level of each university.

Chapter 7

References

[1] Jennifer Golbeck, Cristina Robles, and Karen Turner. 2011 Predicting Personality with social media. In CHI'11 extended abstracts on human factors in computing systems. ACM, 253–262.

[2] Keirsey and Bates 1978; Kroeger and Thuesen 1992, 1988; Myers 1987; Silver and Hanson 1980

[3] Goldberg, L.R. (1992). The development of markers for the Big Five factor structure. Psychological Assessment, 4, 26-42. Digman, R.M. (1990). Personality structure: Emergence of the five-factor model. Annual Review of Psychology, 41, 417-440.

[4] Boyle, GJ. (1989). Re-examination of the major personality-type factors in the Cattell, Comrey, and Eysenck scales: Were the factor solutions by Noller et al. optimal? Personality and Individual Differences, 10, 1289-1299.

[5] Costa, P.T., McCrae, R.R. (1992). NEO Personality Inventory. Psychological Assessment, 4, 5- 13.

[6] Paul D. Tieger and Barbara Barron-Tieger. Do What You Are: Discover the Perfect Career for You Through the Secrets of Personality Type

[7] Conda (2019) Available at: https://conda.io/en/latest/s Accessed [31.01.2020]

[8] Project Jupyter, Anaconda Inc. (2020). Available at:https://jupyter.org/ Accessed [31.01.2020]

[11] Nigam V. Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning (2018). Available at: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90 Accessed [17.02.2020]

[12] Official Developer Android site (2020). Available at: https://developer.android.com/studio/intro Accessed [01.02.2020]

[13] The Java Language Environment, Oracle Technology Network , Sun

Microsystems, Inc (2019). Available at:
https://www.oracle.com/technetwork/java/intro-141325.html Accessed [01.02.2020]
[14] Mcmilligan R. Is Java Losing Its Mojo? (2013). Available at: https://www.wired.com/2013/01/java-no-longer-a-favorite/ Accessed [01.02.2020]
[15] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A, Irving G. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (November 9, 2015). Available at: http://download.tensorflow.org/paper/whitepaper2015.pdf Accessed [02.03.2020]
[16] NERIS Analytics Limited. Personality Types (2020). Available at: https://www.16personalities.com/personality-types Accessed [06.04.2020]
[17] P. Chang, C.Lin 2 and M.Chen, "A Hybrid Course Recommendation System by Integrating Collaborative Filtering and Artificial Immune Systems",2020
[18] M.Caron, I.Misra, J.Mairal, P.Goyal, P.Bojanowski, A.Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", 2020
[19] Ko-Kang Chu, Maiga Chang, Yen-Teh Hsia, Chung-Yuan Christian Univ., Taiwan,"Designing a Course Recommendation System on Web based on the Students' Course Selection Records", 2015
[20] A. Revathi, D. Kalyani, Somula Ramasubbareddy K. Govinda, "Critical Review on Course Recommendation System with Various Similaritie", 2020
[21] Developing an Intelligent Recommendation System for Course Selection by Students for Graduate Courses Grewal DS1 * and Kaur K2
[22] Dart overview from Dart Dev https://dart.dev/overview, 2022
[23] Pyton logotype from Python Org
https://code.visualstudio.com/docs/python/
[24] P.Bojanowski, A.Joulin, M.Douze, "Deep Clustering for Unsupervised Learning of Visual Features Mathilde Caron", Proceedings of the European Conference on Computer Vision (ECCV), 2018
[25] N.Vlassisb, JakobJ, "The global k-means clustering algorithm Aristidis Likasa",aDepartment of Computer Science, University of Ioannina, 2002

Appendix A

Apendex A

```
def preprocess_text():
    course_text = course_text.replace("'ll", " ")
    course_text = course_text.replace("-", " ")
    course_text = re.sub(r'[^A-Za-z ]+', '', course_text)
    course_text = course_text.lower()
    course_text = lemmatize_text(course_text)
    course_text = ' '.join(map(str, course_text))
    return course_text
```

Figure A.1 Preprocess text

```
def run_model(X):
    initial_centroid = 8
    cluster_size = 100
    iteration_size = 500
    init_method = 'k-means++'
    model = KMeans(n_clusters=cluster_size, init=init_method, max_iter=iteration_size, n_init=init
    model.fit(X)
    return model

for cluster_number in range(1, cluster_size):
    model = run_model(X)
    labels = kmeans.labels_
    sse[cluster_number] = model.inertia_

def plot_kmeans_result(sse, cluster_size):
    plt.figure(dpi=300)
    plt.plot(cluster_size, sse)
    plt.xlabel("Cluster size")
    plt.ylabel("Sum of squared error")

def get_cluster_top_terms():
    centroid_points = model.centroids.args()
    for index in range(cluster_size):
        print(centroid_points[index])
        keywords = []
        for point in centroid_points[index, :]:
            keywords.append('%s' % features[point])
        return keywords

def save_model(model):
    pickle.dump(model, open('frozen_model.sav', 'wb'))
```

```
def get_predicted_cluster_custom_text():
    Y = vectorizer.transform([test_text])
    predicted_cluster = model.predict(Y)
    return predicted_cluster
```

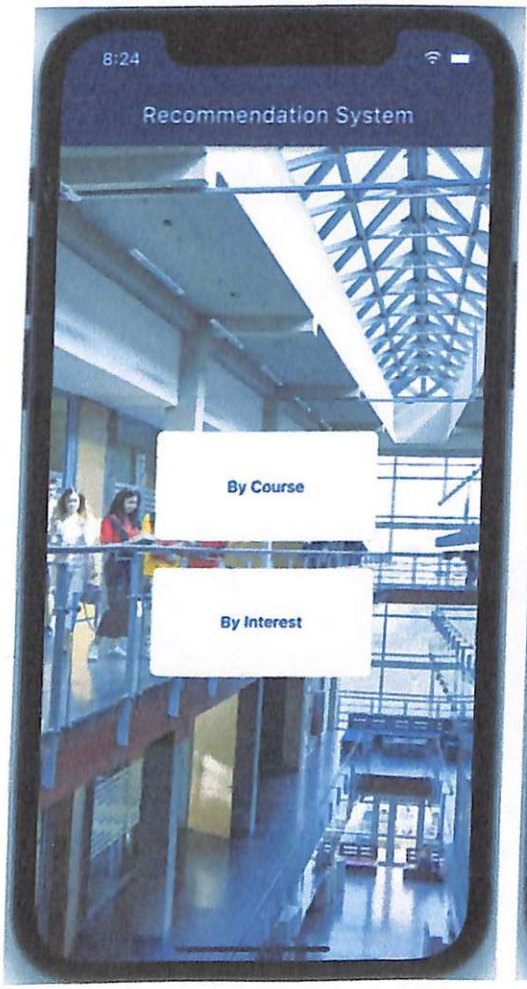
Figure A.2 Building a K-Means Model

```
import json
from flask import Flask, request

import utils

app = Flask(__name__)


@app.route('/recommend', methods=['POST'])
def recommend():
    event = json.loads(request.data)
    course_id = event.get('course_id')
    interest_text = event.get('interest')
    if course_id is not None:
        courses = utils.recommend_by_course_id(course_id)
        return '[' + ', '.join(map(str, courses)) + ']'
    elif interest_text is not None:
        courses = utils.recommend_by_interest(interest_text)
        return '[' + ', '.join(map(str, courses)) + ']'
    else:
        return "{'error': 'Choose recommendation type'}"


@app.after_request
def after_request(response):
    header = response.headers
    header['Access-Control-Allow-Origin'] = '*'
    return response


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

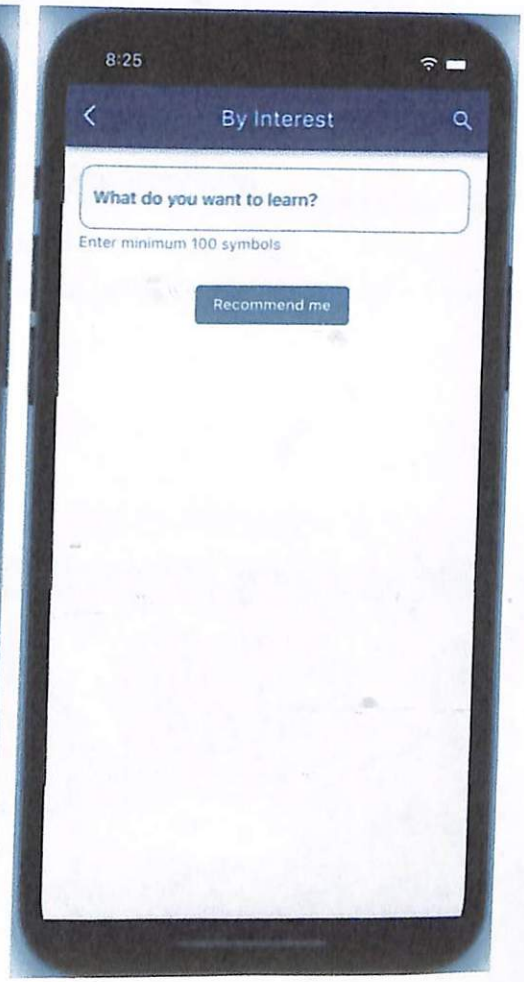
Figure A.3 Flask API to recommend from frozen model

Appendix B

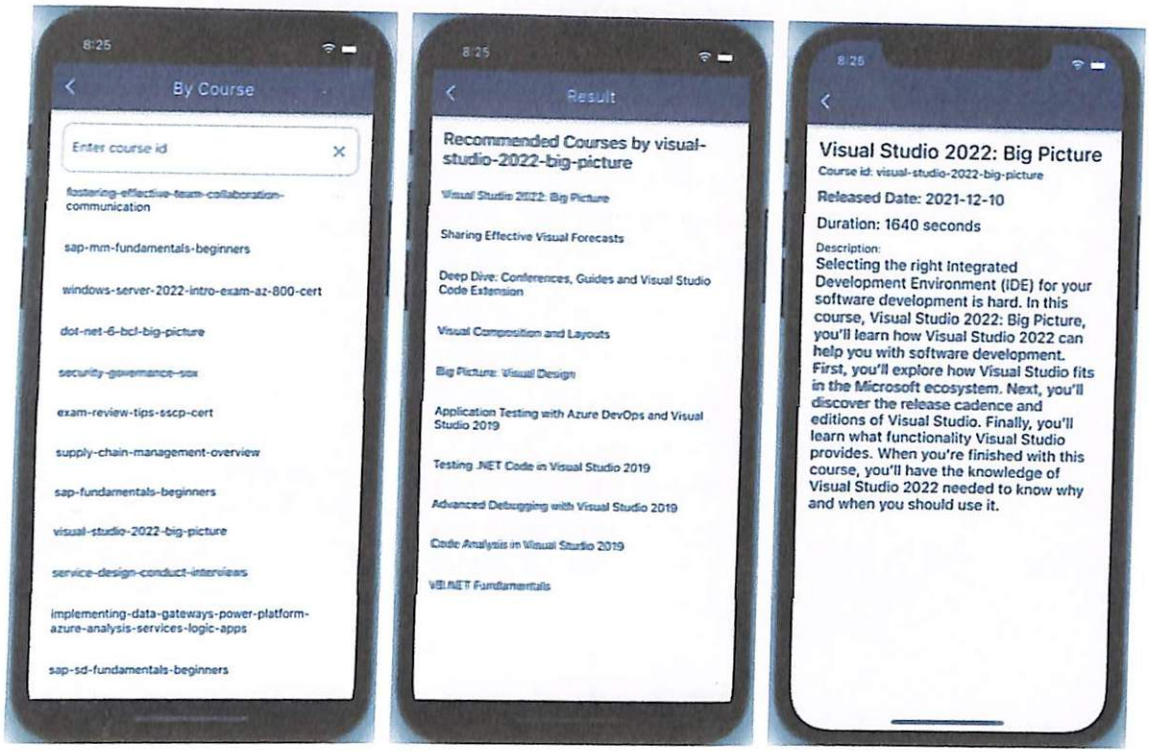
Apendex B



Step 1



Step 2



Step 3 Step 4 Step 5

Figure B1. Recommendation application steps (ios)

Figure B2. Recommendation application steps (web)