



CHURN PREDICTION, RECOMMENDATION ENGINE AND CHATBOT DESIGNS IN VIDEO STREAMING SERVICES INDUSTRY

Senior Design Project I

Beyzanur Yıldız

Hamza Gözükara

Berkay Arat

2025

**MEF UNIVERSITY
FACULTY OF ENGINEERING**

DEPARTMENT OF COMPUTER ENGINEERING

**CHURN PREDICTION, RECOMMENDATION
ENGINE, AND CHATBOT DESIGNS IN VIDEO
STREAMING SERVICES INDUSTRY**

Senior Design Project I

Beyzanur Yıldız

Hamza Gözükara

Berkay Arat

Advisor: Asst. Prof. Tuna Çakar, Assoc. Prof. Şeniz Demir

2025

**MEF UNIVERSITY
FACULTY OF ENGINEERING**

DEPARTMENT OF COMPUTER ENGINEERING

Project Title : Churn Prediction, Recommendation Engine and Chatbot Designs in Video Streaming Industry

Students Name : Beyzanur Yıldız, Berkay Arat, Hamza Gözükara
Date : 10/01/2025

I hereby state that the design project prepared by Beyzanur Yıldız, Hamza Gözükara and Berkay Arat has been completed under my supervision. I accept this work as a “Senior Design Project”.

____/____/2025
Dr. Tuna Çakar, Assoc. Prof. Şeniz Demir

I hereby state that I have examined this senior design project by Beyzanur Yıldız, Hamza Gözükara and Berkay Arat. This work is acceptable as a “Senior Design Project”.

____/____/2025
Muhittin Gökmen

Head of the Department of
Computer Engineering

ACADEMIC HONESTY PLEDGE

In keeping with MEF University Student Code of Conduct, I pledge that this work is my own and that I have not received inappropriate assistance in its preparation. I further declare that all resources are explicitly cited.

<u>NAME</u>	<u>DATE</u>	<u>SIGNATURE</u>
Beyzanur Yıldız	10/01/2025	
Hamza Gözükara	10/01/2025	
Berkay Arat	10/01/2025	

ABSTRACT

CHURN PREDICTION, RECOMMENDATION ENGINE AND CHATBOT DESIGNS IN VIDEO STREAMING SERVICES INDUSTRY

Beyzanur Yıldız
Hamza Gözükara
Berkay Arat

MEF UNIVERSITY
Faculty of Engineering
Department of Computer Engineering

Advisor: Dr. Tuna Çakar, Assoc. Prof. Şeniz Demir

JANUARY, 2025

This project aims to enhance user engagement, satisfaction, and interaction in the video streaming services industry. It consists of three main components powered by machine learning and artificial intelligence solutions:

The first component of the project is a churn prediction model that utilizes machine learning to analyze user viewing behaviors and identify customers likely to discontinue the service. This prediction model enables businesses to take proactive measures to retain customers and reduce churn rates. The second component is a personalized content recommendation system that analyzes user preferences and viewing history to provide tailored content suggestions. This system is designed to improve user satisfaction and interaction. The final component is a chatbot developed to assist users in obtaining information about the available content or requesting personalized recommendations. This chatbot processes user inputs to provide relevant answers or suggestions, making it easier to navigate the content library. It is designed with Large Language Models (LLMs) to enable natural and intuitive interactions.

Together, these three components aim to improve customer engagement, satisfaction, and overall interaction with the streaming service.

Keywords: Machine Learning, Artificial Intelligence, Video Streaming Services, Churn Prediction, Personalized Content Recommendation, Chatbot, Large Language Models, User Satisfaction, Interaction, Customer Engagement

ÖZET

Video Servis Sağlayıcı Endüstrisinde Müşteri Kayıp Analizi, Öneri Sistemi ve Sohbet Robotu Tasarımı

Beyzanur Yıldız, Hamza Gözükara, Berkay Arat

MEF ÜNİVERSİTESİ
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Asst. Dr. Tuna Çakar, Assoc. Prof. Şeniz Demir

OCAK, 2025

Bu proje, video akış hizmetleri sektöründe kullanıcı bağlılığını, memnuniyetini ve etkileşimiğini artırmayı hedeflemektedir. Makine öğrenmesi ve yapay zeka tabanlı çözümler kullanılarak üç ana bileşenden oluşmaktadır:

Projenin ilk bileşeni, müşteri izleme davranışlarını analiz ederek hizmeti sonlandırmayı muhtemel müşterileri belirlemek için makine öğrenmesinden yararlanan bir müşteri kaybı tahmin modelidir. Bu tahmin modeli, işletmelerin müşterilerini elde tutmak ve kayıp oranlarını azaltmak için proaktif adımlar atmasına olanak sağlar. Projenin ikinci kısmı, kullanıcı tercihleri ve izleme geçmişini analiz ederek kişiselleştirilmiş içerik önerileri sunmayı hedefleyen bir kişiselleştirilmiş içerik öneri sistemidir. Bu sistem, kullanıcı memnuniyetini ve etkileşimiğini artırmayı amaçlamaktadır. Projenin son bileşeni ise kullanıcıların mevcut içerikler hakkında bilgi almasını veya kişiselleştirilmiş öneriler talep etmesini sağlayan bir chatbot geliştirmesidir. Bu chatbot, kullanıcı girdilerini işleyerek ilgili cevaplar veya öneriler sunmak üzere tasarlanmıştır ve içerik kütüphanesini daha kolay bir şekilde gezinmeye mümkün kılar. Bu bileşen, doğal ve sezgisel etkileşimleri mümkün kılmak için Büyük Dil Modelleri (LLM'ler) kullanmaktadır.

Bu üç bileşen bir araya gelerek, akış hizmetiyle olan müşteri bağlılığını, memnuniyetini ve genel etkileşimi artırmayı hedeflemektedir.

Anahtar Kelimeler: Makine Öğrenmesi, Yapay Zeka, Video Akış Hizmetleri, Müşteri Kaybı Tahmini, Kişiselleştirilmiş İçerik Önerisi, Chatbot, Büyük Dil Modelleri, Kullanıcı Memnuniyeti, Etkileşim, Müşteri Bağlılığı

TABLE OF CONTENTS

ABSTRACT	4
ÖZET	5
TABLE OF CONTENTS	6
LIST OF TABLES	9
LIST OF FIGURES	10
LIST OF ABBREVIATIONS	12
1. INTRODUCTION	14
1.1. Motivation	15
1.2. Broad Impact	15
1.2.1. Global Impact of the solution	19
1.2.2. Economic Impact of the Solution	19
1.2.3. Environmental Impact of the Solution	20
1.2.4. Societal Impacts of the Solution	20
1.2.5. Legal Issues Related to the Project	20
2. PROJECT DEFINITION AND PLANNING	21
2.1. Project Definition	22
2.2. Project Planning	23
2.2.1. Aim of the Project	24
2.2.2. Project Coverage	24
2.2.3. Use Cases	28
2.2.4. Success Criteria	28
2.2.5. Project Time and Resource Estimation	29
2.2.6. Solution Strategies and Applicable Methods	30
2.2.7. Risk Analysis	31
2.2.8. Tools Needed	32
3. THEORETICAL BACKGROUND	34
3.1. Literature Survey	34
3.1.1.1. What is Churn Rate?	34
3.1.1.2. What causes customers to churn?	34
3.1.1.3. Related Works	34
3.1.2. Recommendation Engine Literature Survey	35
3.1.2.1. Collaborative Filtering in Recommendations	35
3.1.2.2. Advances in Neural Network-Based Models	35
3.1.2.3. Hybrid Recommendation Models	36
3.1.3. Chatbot Literature Survey	36
3.1.3.1. Retrieval-Augmented Generation	36
3.1.3.2. Llama 3	37
3.2. Solution Methods	38

3.2.1. Churn Prediction Methods	38
3.2.1.1. Data Preparation	39
3.2.1.2. Exploratory Data Analysis	39
3.2.1.3. Model Selection and Training	39
3.2.1.4. Model Evaluation	40
3.2.2. Recommendation Engine Methods	41
3.2.2.1. Data Collection	41
3.2.2.2. Data Preprocessing	41
3.2.2.3. Model Selection	41
3.2.2.4. Implementation	42
3.2.2.5. Testing and Evaluation	43
3.2.2.6. Integration	43
3.2.3. Chatbot Methods	43
3.2.3.1. Web Scraping	44
3.2.3.2. Synthetic Data Generation	44
3.2.3.3. Fine Tuning Pre-trained Model	44
3.2.3.4. Retrieval-Augmented Generation	45
3.2.3.5. Final Evaluation	47
4. ANALYSIS AND MODELING	48
4.1. System Factors	48
4.1.1. System Factors for Churn Prediction System	48
4.1.2. System Factors for Recommendation Engine	48
4.1.3. System Factors for RAG Chatbot	49
4.2. How System Works	49
4.3. Modelling	50
4.3.1.1. Modelling of Churn Prediction Model	50
4.3.1.2. Modelling of the Recommendation Engine	51
4.3.1.3. Modelling of the Chatbot	52
4.3.2. UML (Unified Modeling Language) Diagrams	53
5. DESIGN, IMPLEMENTATION AND TESTING	57
5.1. Design	57
5.1.1. Design of the Churn Prediction Model	57
5.1.2. Design of the Recommendation Engine	57
5.1.3. Design of the Chatbot	58
5.2. Implementation	60
5.2.1 Implementation of the Churn Prediction Model	60
5.2.1.2. Implementation of the Recommendation Engine	61
5.2.1.3. Implementation of the Movie Recommendation Chatbot	62
5.3. Testing	63
5.3.1. Testing Phase of Churn Prediction Model	63
5.3.2 Testing Phase of Recommendation Engine	64
5.3.3 Testing Phase of the Chatbot	64

6. RESULTS	65
6.1. Results of Churn Prediction Model	65
6.2. Recommendation Engine Results	73
6.3. Movie Recommendation Chatbot Results	78
7. CONCLUSION	81
7.1. Life-Long Learning	81
7.2. Professional and Ethical Responsibilities of Engineers	82
7.3. Contemporary Issues	82
7.4. Team Work	83
APPENDIX A	83
REFERENCES	86

LIST OF TABLES

Table 1 *The Evolution of the Methodologies in Churn Prediction, Recommendation and Chatbot in Video Streaming Industry Systems*

Table 2 *Gantt-Chart for the Project Planning*

Table 3 *Performance of finetuned Llama 3 models on key benchmark evaluations*

LIST OF FIGURES

- Figure 1:** *UML Diagram of the Project*
- Figure 2:** *Overview of RAG structure*
- Figure 3:** *Illustration of the overall architecture and training of Llama 3*
- Figure 4:** *Confusion matrix and metric formulas.*
- Figure 5:** *Softmax Function and Layer Illustration*
- Figure 6:** *Cosine of the angle(θ) between two vectors(A and B)*
- Figure 7:** *Retrieval Process*
- Figure 8:** *Workflow of a RAG system*
- Figure 9:** *Content-Based Recommendation Engine UML Diagram*
- Figure 10:** *Deep Neural Network Recommendation Engine*
- Figure 11:** *Churn Prediction UML Diagram*
- Figure 12:** *ChatBot UML Diagram*
- Figure 13:** *Pinecone Document*
- Figure 14:** *Chatbot UI Design*
- Figure 15:** *Chatbot Graph Design*
- Figure 16:** *Distribution of churned and non-churned customers.*
- Figure 17:** *Histogram plot of churn distribution by ContentVariety*
- Figure 18:** *Pie plot of churn distribution by ContentVariety*
- Figure 19:** *Histogram plot of churn distribution by TotalLogins*
- Figure 20:** *Pie plot of churn distribution by TotalLogins*
- Figure 21:** *Histogram plot of churn distribution by AverageLoginsPerDay*
- Figure 22:** *Histogram plot of churn distribution by UniqueDevices*
- Figure 23:** *Pie plot of churn distribution by UniqueDevices*
- Figure 24:** *Histogram plot of churn distribution by AverageUsageDuration*
- Figure 25:** *Histogram plot of churn distribution by AverageContentCompleteRatio*
- Figure 26:** *Feature Importance Plot of RandomForest Model*
- Figure 27:** *Train-Test Accuracy Comparison of ML Models*
- Figure 28:** *Confusion Matrices Comparison of ML Models*
- Figure 27:** *Train-Test Accuracy Comparison of ML Models*
- Figure 28:** *Confusion Matrices Comparison of ML Models*

- Figure 29:** Classification Report of the Random Forest Model
- Figure 30:** Overview of Datasets Used for the Recommendation Engine.
- Figure 31:** Comparison of Models based on RS, Precision and RMSE
- Figure 32:** Test Accuracy for DNN-Softmax Recommendation Engine
- Figure 33:** Example output for User 17
- Figure 34:** The top content-based recommendations for the selected show ranked by similarity scores.
- Figure 35:** The top content-based recommendations for the selected show "Selvi Boylum Al Yazmalım" ranked by similarity scores.
- Figure 36:** The top recommendations for "Selvi Boylum Al Yazmalım" with and without AI support.
- Figure 37:** Retrieved documents for Example 1
- Figure 38:** Chatbot Responses for Example 1
- Figure 39:** Retrieved Documents for Example 2
- Figure 40:** Chatbot Response for Example 2
- Figure 41:** Response of the Query ‘Kara Şövalye filmini daha detaylı şekilde yorumlar mısın? Konu, oyuncu performansı, eleştiriler vs.’’

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
SVD	Singular Value Decomposition
SVDpp	Singular Value Decomposition++
IBCF	Item-Based Collaborative Filtering
Hybrid	Hybrid Recommendation System
HybridNN	Hybrid Neural Network
NMF	Non-negative Matrix Factorization
KNN	K-nearest Neighbors
LLMs	Large Language Models
LLMRG	Large Language Model Reasoning Graphs
UES	User Engagement Scale
CF	Collaborative Filtering
CBF	Content-Based Filtering
GC	Google Collaborative
GCP	Google Cloud Platform
GDPR	General Data Protection Regulation
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
BLEU	Bilingual Evaluation Understudy
RAG	Retrieval-Augmented Generation

1. INTRODUCTION

In the current video streaming environment, platforms are facing considerable challenges in maintaining customer loyalty and offering personalized content in the face of increasing competition and a significant expansion in the range of available options. This project addresses these issues by integrating advanced machine learning and artificial intelligence techniques into a unified system comprising churn prediction, recommendation engines, and chatbot designs. As Chen et al. (2020) have observed, the use of personalized AI-driven systems is crucial for improving customer satisfaction and developing long-term loyalty to digital services [1]. By identifying users who are at risk of leaving, providing customized content recommendations through recommendation engines, and facilitating smooth user interactions through chatbots powered by large language models (LLMs), this system aims to reshape the user experience.

This project has the potential to effect transformative change across a range of domains. From an economic standpoint, the system is designed to improve revenue generation by reducing customer loss, increasing subscription renewals, and optimizing operational costs. From a societal perspective, the system promotes inclusivity and accessibility by simplifying the process of discovering content and fostering thoughtful consumption habits. From a health perspective, the system mitigates the risks associated with excessive screen time by aligning recommendations with user preferences and encouraging balanced viewing habits. In regard to the legal aspects of the system, the developers have ensured compliance with the relevant data protection regulations, such as the General Data Protection Regulation (GDPR), and incorporated advanced encryption techniques to safeguard user privacy and build trust. Moreover, the incorporation of hybrid recommendation methodologies, as exemplified by Mehrabani et al. (2020), addresses issues such as sparse data and the cold-start problem, thereby achieving high-quality personalization [1].

On a global scale, the system represents a new standard for innovation in video streaming services, combining predictive analytics, user-centric design, and advanced AI capabilities. By optimizing resource allocation and reducing redundancy in data usage, it also supports environmental sustainability, minimizing the energy footprint of streaming operations. This

project aims not only to advance the user experience but also to contribute to the shaping of a more efficient, ethical, and sustainable future for the video streaming industry.

1.1. Motivation

The motivation behind our project comes from that we recognize and address the major issues that the video streaming sector is now dealing with, such as high churn rates and the requirement for tailored content distribution. A complete solution that greatly increases user pleasure and commitment can be offered by combining recommendation engines, chatbot designs, and churn prediction.

The system's features include recognizing users who might be considering canceling their subscription, making content recommendations based on user preferences to reduce the effects of information overload, and making navigating easier with chatbots that are easy to use. Through reducing churn rates and boosting subscription renewals, these features aim to increase profitability while encouraging inclusion and intentional content consumption. This project is a big step in the direction of making streaming more effective and user-focused.

1.2. Broad Impact

The implementation of our combined chatbot, recommendation engine, and churn prediction system has the potential to completely transform the video streaming services sector, with far-reaching implications in the fields of security, health, and the economy. By detecting customers who are prone to churn and providing proactive engagement techniques, the technology improves client retention in an economical manner. At the same time, the customized recommendation engine expands platform utilization and improves user satisfaction, which leads to more subscription renewals and more income. In order to increase operational efficiency, the system also optimizes resource allocation and streamlines content administration.

Societally, by providing personalized content recommendations, this technology encourages a more user-centered streaming experience and reduces decision fatigue. By making large information libraries easy to navigate and accessible to a wide range of people, it promotes inclusion. The system has the ability to impact cultural trends and enhance users' viewing

habits by promoting the deliberate intake of significant content, which will strengthen their bond with their entertainment experiences.

From a health perspective, the personalized recommendation system minimizes excessive screen time and exposure to irrelevant or unsuitable content by aligning suggestions with user preferences and habits. The chatbot also offers customers an easy-to-use interface for interacting with the site, offering helpful reminders to encourage pauses and encourage balanced viewing habits. When combined, these characteristics support enhanced mental health and a more health-conscious method of consuming content.

The system places a high priority on user privacy and data protection in terms of security. Strict data governance guidelines and cutting-edge encryption technology are used to protect sensitive client data and guarantee adherence to international data privacy laws like the GDPR. Long-term platform loyalty is encouraged and user trust is reinforced by this dedication to safe and moral data usage.

On a broader scale, this project sets a new benchmark for innovation in the video streaming industry, influencing global strategies for customer engagement and retention. It advances the industry toward a future where streaming services are not only more personalized and efficient but also align with ethical, health-conscious, and secure practices, driving sustainable growth and user satisfaction worldwide.

Table 1. *The Evolution of the Methodologies in Churn Prediction, Recommendation and Chatbot in Video Streaming Industry Systems*

Year	Methodology	Description	Related Benefits	Example(s)
2000	Rule-Based Churn Prediction	This method uses predefined business rules and thresholds to identify users who are likely to experience churn	Provides quick and simple insights into customer behavior.	Early telecom churn detection systems

2005	Collaborative Filtering for Recommendations	Analyzes user behavior to recommend content based on similar user profiles.	Enhances personalization by leveraging user similarities.	Movie or music streaming platforms.
2010	Decision Tree-Based Churn Prediction	Applies machine learning models like decision trees to predict churn probabilities.	Offers interpretable predictions for proactive retention efforts.	Customer segmentation in streaming services.
2012	Content-Based Filtering for Recommendations	Suggests content based on individual user preferences and previously consumed content.	Targets recommendations accurately based on user interests.	Personalized video playlists.
2015	Hybrid Models for Recommendations	Combines collaborative and content-based filtering to improve recommendation quality.	Addresses cold-start issues and enhances overall accuracy.	Personalized dashboards in streaming apps.
2017	NLP-Based Chatbots	Leverages natural language processing (NLP) to enable text-based interactions with users.	Improves user engagement and makes content discovery intuitive.	Customer service chatbots for platforms.
2018	Deep Learning for Churn Prediction	Employs neural networks to detect complex churn	Improves prediction accuracy by identifying	Retention strategies in OTT platforms.

		patterns from user behavior data.	non-linear relationships.	
2019	Large-Scale Recommendation Systems	Utilizes matrix factorization and scalable deep learning techniques for recommendations.	Handles vast datasets efficiently, ensuring real-time updates.	Netflix's recommendation engine.
2020	Conversational AI for Chatbots	Integrates advanced AI models for context-aware and interactive dialogues with users.	Provides dynamic and personalized user interactions.	Voice-controlled assistants in streaming.
2022	Predictive Maintenance via Churn Models	Incorporates churn predictions into system maintenance to optimize content and resource allocation.	Reduces operational costs and ensures optimal user experience.	Intelligent resource allocation systems.
2023	Personalized Engagement Bots	Enhances chatbots with AI-driven recommendations and real-time content updates.	Boosts engagement by tailoring responses and suggestions to individual user needs.	Dynamic video recommendations through chat.
2024	Personalized Engagement Bots	Utilizes LLMs for seamless, human-like conversations and real-time	Enhances user satisfaction with nuanced,	AI chatbots in streaming platforms.

		personalized recommendations.	context-aware content discovery.	
--	--	-------------------------------	----------------------------------	--

1.2.1. Global Impact of the solution

The project could create a global standard for innovation in video streaming services through showing off the ways in which AI-powered systems can improve user engagement, content customisation, and retention. The system may affect globally customer engagement strategies across sectors by using chatbots, recommendation engines, and improved churn prediction. Its scalability supports a more inclusive digital environment by enabling adaption to a range of regional preferences. Furthermore, it increases global cooperation by demonstrating how machine learning and massive language models can improve user experiences.

The initiative minimizes the carbon footprint of large-scale streaming operations by improving resource allocation and delivering content while consuming less energy. This research can be a step toward green technology advancement as these sustainable practices complement global efforts to slow climate change.

1.2.2. Economic Impact of the Solution

The proposed system is expected to provide substantial economic benefits for video streaming platforms, including a reduction in customer loss and an increase in subscription renewals. The implementation of personalized recommendations has been demonstrated to improve customer satisfaction and increase platform usage.

Furthermore, the ability to predict customer churn allows companies to take action in a timely manner to retain customers who are at risk of leaving. The achievement of these objectives will result in increased revenue, reduced acquisition costs, and an improvement in return on investment. Another significant economic advantage is operational efficiency. The solution reduces overhead costs and allows for the effective management of extensive content libraries through the automation of user interactions with chatbots and the optimization of resource allocation.

Consequently, businesses are able to enhance their financial management, directing their resources towards user interaction and content creation strategies that maximize profitability.

1.2.3. Environmental Impact of the Solution

The system's promotion of sustainability is achieved through the minimization of redundant streaming procedures and the optimization of data utilization. The energy consumption associated with user queries and extensive streaming is reduced by customized recommendations and effective content delivery. The system's capacity to predict user attrition also assists platforms in more efficient resource management, which reduces waste in server usage and content creation.

Moreover, the approach indirectly reduces excessive streaming and the environmental costs associated with it by inspiring users to interact with meaningful and pertinent information. These methods assist the video streaming sector in reducing its carbon footprint and advancing broader environmental sustainability objectives.

1.2.4. Societal Impacts of the Solution

The project's objective is to improve the accessibility and inclusivity of video streaming platforms, thereby facilitating navigation and consumption of personalized content for a diverse demographic of viewers. The project advances a user-centered approach to streaming by reducing choice fatigue and simplifying content discovery. By providing a seamless user experience, the chatbot component facilitates the acquisition of digital literacy skills among individuals with limited technological expertise.

Furthermore, the personalized recommendation engine, which matches recommendations to user preferences, has the effect of promoting an intentional approach to content consumption. It is possible that an increase in user pleasure and cultural engagement may result from more intentional viewing habits. The project's emphasis on personalization and diversity may eventually influence how society views the consumption of digital information.

1.2.5. Legal Issues Related to the Project

The implementation of this project requires strict observation of data protection regulations, such as the General Data Protection Regulation (GDPR), in order to guarantee the privacy and security of users. The solution minimizes the risk of data breaches and safeguards vital

customer data through the implementation of advanced encryption techniques and best practices in data management.

Machine learning libraries and models must be properly licensed because the chatbot and recommendation engine components depend on intellectual property frameworks. The project's ethical AI practices, including the minimization of algorithmic bias and the safeguarding of user confidentiality, contribute to the avoidance of human rights violations.

2. PROJECT DEFINITION AND PLANNING

2.1. Project Definition

The objective of this project is to develop an intelligent system for the video streaming services industry, combining churn prediction, personalized recommendation engines, and chatbot designs that will improve user retention, satisfaction, and commitment. The system is hoping to provide personalized content recommendations, proactive churn interventions, and smooth user interactions through the use of innovative machine learning approaches, such as large language models (LLMs) and hybrid recommendation methods.

The project focuses on accuracy, scalability, and user data security when designing, implementing, and deploying this integrated system. The recommendation engine provides customized recommendations based on user preferences and watching history, while the churn prediction component detects users who are at risk based on behavioral patterns. Users will find it easy to navigate material because of the chatbot's natural language-based, intuitive interactions.

The system architecture will make use of reliable servers which allow GPUs to perform efficient model training, Python programming, and machine learning frameworks like PyTorch and TensorFlow. For vector database management, the project includes cloud services like Pinecone and Google Colab. In addition to improving user happiness and experience, this system is made to be scalable and operationally efficient, guaranteeing that it can be customized to meet the continually evolving needs of the video streaming market.

2.2. Project Planning

Table 2. Gantt-Chart for the Project Planning

Task	Responsible Person	Weeks												
		1	2	3	4	5	6	7	8	9	10	11	12	13
Literature Review on Churn Prediction, Recommendation Engine and ChatBot	Beyzanur Yıldız Hamza Gözükara Berkay Arat													
Dataset Collection and Preprocessing	Beyzanur Yıldız Hamza Gözükara Berkay Arat													
Algorithm Selection for Recommendation	Beyzanur Yıldız													
Learning necessary NLP concepts(Fine-tune, RAG)	Berkay Arat													
Feature Engineering for Churn Prediction	Hamza Gözükara													
System Modeling and Analysis	Beyzanur Yıldız Hamza Gözükara Berkay Arat													
System Integration and Testing	Beyzanur Yıldız Hamza Gözükara Berkay Arat													
System Integration, Testing and Debugging	Beyzanur Yıldız Hamza Gözükara Berkay Arat													
Report Writing	Beyzanur Yıldız Hamza Gözükara Berkay Arat													
Preparing Presentation	Beyzanur Yıldız Hamza Gözükara Berkay Arat													

2.2.1. Aim of the Project

This project has three main objectives. Firstly, we aim to develop and implement a TV content recommendation system via semantics and user's favorites, likes, dislikes and watch history data. In the semantics-based part of the system, vector embeddings and vector database implementations to catch similarities between TV contents and its story, subjects etc and recommend the results to the user. Secondly, we aim to develop and deploy a customer churn prediction model for Dsmart-GO using several machine learning algorithms, a model that will predict users at risk of discontinuing the service via the customer's using behaviors of the system and relevant data. And lastly, we aim to develop a chatbot for Dsmart-GO's user interactions using NLP techniques and LLMs. This chatbot will interact with the user and provide relevant information about the contents from the system's database to the user, answer questions and help the user if needed.

2.2.2. Project Coverage

The objective of this project is to develop and implement a TV content recommendation system, a customer churn prediction model, and an interactive chatbot for the Dsmart-GO platform. The goal is to enhance user engagement, increase retention, and improve the overall customer experience by using several machine learning and natural language processing (NLP) techniques. In summary, it can be said that the project is organized into three key parts: content recommendation, churn prediction, and chatbot development.

System Design and Architecture

The first phase of the project will focus on designing and implementing a system capable of delivering personalized recommendations and semantics based recommendations, predicting customer churn, and supporting real-time chatbot interactions. This includes several parts such as the development of a detailed software architecture and an optimized database schema that can store and handle large volumes of user interaction data, TV content metadata, and other system-related information. It is also important that the performance of vector embeddings and the vector database needed for content similarity analysis. For scalability and easier, faster queries we used Pinecone API service for vector database and its query implementations.

Algorithm Development

In the content recommendation system development, machine learning techniques such as SVD, IBCF etc. are used to develop a model to predict and recommend a customer's next TV content that he/she might enjoy based on user behavior, including favorites, likes, dislikes, and watch history etc. In the semantics based part, vector embeddings will be implemented to analyze the semantic relationships between TV content (e.g., story, genre, subject matter). These embeddings will allow the system to identify similarities between content and recommend TV shows and movies that match the user's previously watched contents.

For the churn prediction model, various machine learning algorithms (e.g., decision trees, random forests, logistic regression, and neural networks) will be created and tested, then the best one will be implemented to predict if a user is discontinuing the service or not. By analyzing user behavior data, system usage patterns, and other relevant customer's system using metrics, the model will help identify customers at churn risk and provide insights into improving customer retention.

Lastly, an interactive chatbot will be developed using natural language processing (NLP) techniques to enable interactions between users and the system. The chatbot will be designed to answer user queries, suggest relevant content, provide detailed information, and assist users in navigating the Dsmart-GO platform. To be able to achieve this, at first stage, we planned to fine-tune Llama models and therefore to generate synthetic but meaningful question-answer pairs for the training dataset using high-end generative AI models via detailed TV content information scrapped from famous content review sites. After some research, we realized that it would be better to use retrieval augmented generation(RAG) methods using existing LLMs (e.g ChatGPT) for our chatbot. RAG helps the language model to have up-to-date movie metadata even if it wasn't trained with newly released movie metadata and reviews. Finally, we decided to use RAG methods for our chatbot at first stage and if needed fine-tuning is also going to be considered. There are two main parts for the methods of RAG: retrieval and generation. These methods will be explained in detail in the later part of this report.

2.2.2.1. Data Management and Processing

Data management is a crucial aspect of the project, especially given the variety of data types involved (e.g., user behavior, content metadata, interaction logs). The system will require

secure and efficient databases to store and manage user data, TV content metadata, recommendation logs, and churn prediction inputs. Data privacy issues will not be a problem from the development side, since we will have user data with privacy masking and also we will use the company servers to be able to be in their IT security scope, this and all else will ensure compliance with all relevant regulations.

2.2.2.2. Testing and Quality Assurance

The system will be tested to ensure that all components are functioning correctly and efficiently. Testing will cover multiple areas, including unit testing for individual functions, integration testing for ensuring compatibility between subsystems, system testing for overall functionality etc. For both of our three main parts, testing will be the main concern before the deployment, we will be sure that the recommendation system gives relevant content, we will be sure that the churn model performs satisfactorily (although not final, most likely over %80 accuracy) and there is no under/over fitting situations, and also we will be sure that the chatbot gives relevant responses to the customers via testing.

2.2.2.3. Deployment and Integration

The deployment and integration phase will focus on streaming the system to a live environment. This includes deploying the recommendation system, churn prediction model, and chatbot on the Dsmart-GO platform. The deployment will cover merging these systems into one system or UI, and delivering it to the company for company testing. System will work through several APIs and it will be easy for them to integrate it to their systems.

2.2.2.4. Training and Support

Following deployment, training sessions are planned to be given to the administrators in order to ensure they are familiar with the system's capabilities. This will include training on how our TV content recommendation and interactive chatbot system works and how to implement it to any system. Ongoing technical support will be available to detect or help any issues that will be a problem after the deployment, including bug fixes, performance enhancements, and system updates. Also there will be updates in a stable interval of time to the content database of the vector database, recommendation system and chatbot's training data in order to satisfy the system's needs for newly added TV contents. Via these updates,

chatbot will be able to train itself continuously and the recommendation system will be always up to date.

2.2.2.5. Performance Monitoring and Optimization

After the deployment, the system needed to be monitored to see its performance, particularly in terms of recommendation accuracy, churn prediction effectiveness, and chatbot's responses accuracy. User feedback will be very important, especially for chatbot's accuracy, and it will be gathered to guide iterative improvement sessions. It will all ensure that the system will remain efficient, accurate and listens to the user. Optimizations will be done in order to improve the success of the content recommendation system, churn predictions, and increase the chatbot's ability to handle a variety of user interactions.

2.2.3. Use Cases

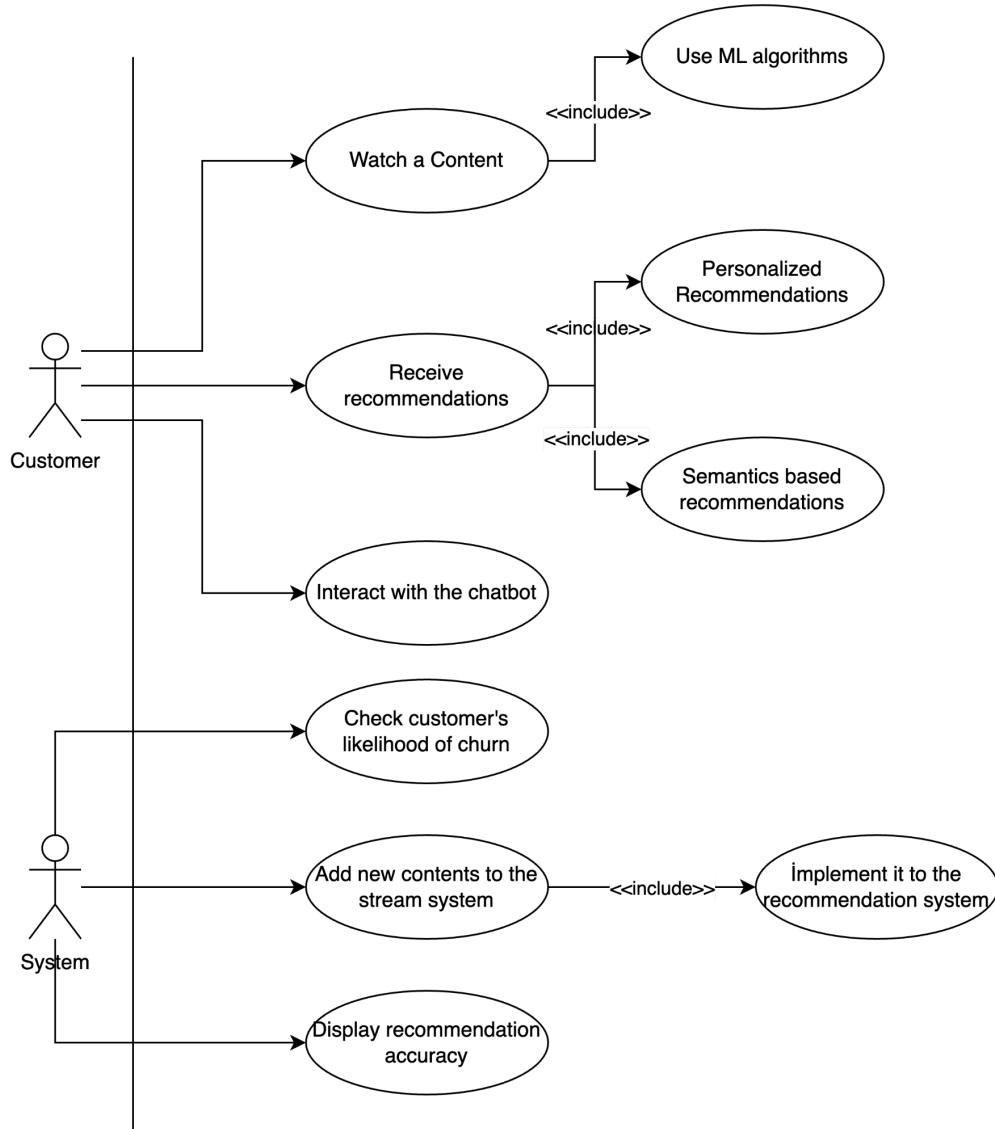


Figure 1: UML Diagram of the Project

2.2.4. Success Criteria

The general success criteria of our three AI products is maximizing user experience on discovering new movies and TV series based on their watching history and current mood, and increasing the user's interaction time with the product. To achieve that, it is aimed to develop a churn prediction model which aims to reach a good accuracy on predicting when customers are likely to discontinue using the product by analyzing user satisfaction based on their interactions with the D-Smart GO product, a recommendation engine that suggests relevant

movies/tv series based on the liked and favourite content, and finally a chatbot that users can interact with to get meaningful recommendations with cinematic explanations based on their mood and preferences.

The churn prediction success is determined by machine learning models' ability to predict users that are likely to leave. The aim of these models is to identify users that are not satisfied with the product and increase the user satisfaction. To evaluate the model performance, classification metrics such as accuracy, precision, recall and F1-score will be used.

The recommendation engine's success is determined by precise and diverse movie recommendations that are predicted from liked, disliked and favorite contents. Both offline and online evaluation metrics can be used to determine the accuracy of the recommendations. For the offline evaluation, precision, recall and F1-score are important evaluation criteria and are used in our models. For the online evaluation, click-through rate, which measures how often users interact with recommendations, and view rate, which gives the percentage of recommendations that result in the user starting to watch the content, can be considered valid evaluation metrics. These metrics are also planned to be used as evaluation metrics.

The movie recommendation chatbot's success is determined by ability to recommend related movies and TV series based on the user prompt while also ensuring natural, human-like conversations. The chatbot should support the movie recommendations by explaining why it recommends that movie based on the movie summary and reviews obtained from Beyazperde website. Those metrics also directly affect the user satisfaction ratings and increased session lengths. For the evaluations of the models, both automated methods such as BLEU and ROUGE and human evaluation via surveys will be used.

2.2.5. Project Time and Resource Estimation

The literature review part will take one week. This part is very important to start the project with necessary information and background about the project and what is done previously. In the next week, we will analyze the data we have been provided and plan the strategies of how to use and implement the data into our models or systems. In the following two months (weeks 2-10) we are going to develop the TV content recommendation model powered by

both customer behavior and content semantics. Then, we will begin to develop the churn prediction model; it will take approximately 1.5 months (weeks 11-17). After that, we will begin the longest, most time-consuming part of the project, which is developing an interactive chatbot. We plan to complete it approximately 3.5 months (weeks 18-30). Lastly, we will cover all the progress so far and document it inside of a final report in 2 weeks (weeks 31-32). The effort of each group member on the total project should be equal. But even though we all participate in the three main parts of the project, each one of us will be head of one main part and be the director of its progress and documentation. The project cost will be estimated via computation power and resources. Since we will train a LLM model, we will have to need a high-end GPU. For processing, analyzing data and developing an ML model for recommendation engine and churn prediction, we will need high CPU power and some RAM capacity. And also we might have to need a paid plan for API services such as Pinecone API or ChatGPT if the free plan will not be enough for our vector database and chatbot system. The total cost estimate also has a risk buffer, likely around 10-15% of the total estimated cost, to accommodate unforeseen expenses. This buffer accounts for potential risks. The risks that any project might have and should be considered such as extended development time due to unexpected technical challenges. Or it can be increased API costs if additional computational or generative resources are needed, or increasing GPU service costs if additional training is needed.

2.2.6. Solution Strategies and Applicable Methods

In the content semantics based recommendation, we used vector embeddings and vector database systems. We vectorized all the text-based contents about a TV content via HuggingFace sentence-transformers model in 384 dimensions and inserted it into a cloud based vector database provided by Pinecone API. This database service has a fast query service which provides us faster recommendations. Also, in collaborative filtering, we used several user metrics such as liked and disliked contents, watchlists, watched contents ,watching durations and text feedback data to train models using machine learning algorithms such as SVD, IBCF etc. By these applications, we will be able to recommend new content to the user both based on content semantics and user's personal likings.

In the churn recommendation part, we will use the user usage data of Dsmart-GO system we have provided to firstly understand the data and segmentate the users with unsupervised learning techniques. By that we will deeply understand the dataset we have and we will have insights about our possible preprocessing processes. After we complete the preprocessing phase via extracting new features such as subscription duration, different devices accessed, average watch time etc. we will be able to catch patterns with customer churn and make our future model more predictive and powerful. Then lastly, we will train several classification models with our processed dataset such as Random Forests, Linear Regression, Decision Tree Classification etc. We will select the best performing model via best hyperparameters then deploy the model. Our estimated success rate will be at least %80-85 for primitive models.

And lastly, in the chatbot development part, we first planned to fine-tune the Llama 3.2 model for our use case. We were going to detect and change the best parameters for our training process and planned to train with a synthetic dataset. We scraped the TV contents that our system does have from several famous TV content review websites and used them to generate question-answer dataset via high-end generative AI assistants ChatGPT and Google generative AI gemini. Since we changed our chatbot design to use RAG methods, the dataset structure has been changed to cover all the information about each of the movies in the Dsmart-GO platform rather than possible question-answer pairs. The movie metadata documents are transformed into vector embeddings and stored in a vector database. These vector embeddings store the semantic information of the documents numerically in order to be processed while preserving the meaning of documents.

In the RAG system, vector similarity between the documents stored as vector embeddings and user query is calculated. Then most relevant movie documents are retrieved from the database. The relevant documents and user query are combined in a single prompt and sent to the large language model. Large language model generates a response based on the provided information.

2.2.7. Risk Analysis

The implementation of an integrated system for chatbot designs, recommendation engines, and churn prediction is associated with a number of potential risk factors that could impact

the project's overall outcome. As the user base continues to grow, potential issues with scalability may result in slower system performance. Such issues can be minimised through the utilisation of cloud-based solutions that are capable of scaling automatically, coupled with the implementation of a regular assessment of the system's performance, allowing for the effective management of growing loads.

Additionally, issues of algorithmic bias and accuracy must be considered, as the implementation of unfair or inaccurate suggestions has the potential to result in a corresponding reduction in user satisfaction. The reduction of bias and the increase of accuracy can be achieved through the implementation of monitoring procedures, the utilisation of frequent updates, and the incorporation of a diverse range of datasets. To prevent any potential issues from arising during the implementation process, any risks associated with the integration of system components, such as incompatibilities, will be addressed through comprehensive testing and documentation.

It is possible that users may be unable to recognize the system's value or may find it challenging to use, which could result in difficulties with adoption. Pilot testing will prioritize user training, intuitive design, and feedback collection to optimize the user experience and facilitate the adoption of the system. The potential for data breaches represents a significant risk to the security and privacy of user information, as such incidents could erode user confidence in the system. Access controls will be strictly regulated, and data will be encrypted using strong algorithms. Furthermore, the system will comply with all relevant data protection legislation, including the General Data Protection Regulation (GDPR).

The objective is to present the video streaming industry with a dependable, effective, and user-friendly solution by proactively addressing the concerns that have been identified.

2.2.8. Tools Needed

Servers: We are going to need some servers that provide us with a high-end CPU with strong computational power and some RAM space that is at least over 32 GB. Our data-analyzing and processing tasks require a strong CPU and some relatively high RAM space, and our chatbot model training/fine-tuning tasks require an high-end GPU.

Python: We will use the most commonly used programming language for machine learning tasks, which is Python. It will make our tasks easier with its built-in and community-built libraries. It has very easy-to-understand syntax for any implementation and has very large community support for any place we might face an error or exception.

Libraries: We are going to use the scikit-learn library for most of our machine learning model development parts. The scikit-learn library provides several tools and algorithms already implemented and ready to run with the option to change any parameters, providing us scalability. We are going to use libraries like pandas, numpy, etc., to analyze and process our data to prepare it for training and fine-tuning. These libraries provide us flexibility in the use and manipulation of data.

API Services: Pinecone API service for our vector embeddings and vector database write/read/query operations are essential. Our embedded vectors are stored in Pinecone's safe servers, and we are able to make quick queries with the API service. For the primitive version of the system, the unpaid service will be enough, but for the deployment, the paid service and commercial license (if needed) should be provided. In addition, the ChatGPT API service is necessary for the RAG chatbot, as we use the ChatGPT-4o mini as our model. The ChatGPT API has also been used to generate synthetic question-answer pairs based on the scraped data from popular movie review platforms.

3. THEORETICAL BACKGROUND

3.1. Literature Survey

3.1.1. Churn Prediction Literature Survey

3.1.1.1. What is Churn Rate?

The churn rate is a critical indicator of customer retention and the overall health of a business. It reflects a company's ability to maintain its customer base and can highlight potential challenges such as inadequate customer service, substandard product quality, or intense market competition. Calculated as a percentage, the churn rate is determined using the formula:

$$\left(\frac{\text{Number of Customers Lost During Period}}{\text{Number of Customers at Start of Period}} \right) \times 100$$

This metric provides organizations with a quantifiable measure of customer attrition, enabling the tracking of trends over time. In essence, customer churn signifies the loss of clients, while the churn rate quantifies the extent of this loss during a specified period. [2]

3.1.1.2. What causes customers to churn?

Customer churn can result from a variety of factors, including bad customer service, not enough product quality, and increasing competitive pressures. Ineffective or delayed customer support, products failing to meet expectations, and superior alternatives offered by competitors are significant contributors to customer attrition. Pricing issues, such as excessive costs or aggressive discounting by competitors, also play a critical role in influencing churn. Furthermore, external factors such as economic fluctuations and evolving consumer preferences can exacerbate customer loss. To mitigate churn, organizations must implement holistic strategies that prioritize customer satisfaction, sustain competitive offerings, refine pricing models, promote customer engagement, and adapt to dynamic market conditions. [2]

3.1.1.3. Related Works

Xiaojun Wu and Sufang Meng developed a churn prediction model for an e-commerce company. The dataset they have been provided was an imbalance dataset which is possibly the same in our case, where churn samples typically far exceed non-churn samples. This situation was causing classification models to be biased on the majority class. In order to address this

issue, they created an improved SMOTE technique to balance the dataset by synthesizing a controlled number of positive and negative samples. They concluded that the improved SMOTE technique of theirs is a valuable method for all e-commerce churn prediction research. [3]

Ishrat Jahan and Dr. Tahsina Farah Sanam also worked on a churn prediction model on the e-commerce sector. They followed similar solution techniques to ours. The work they have done has five key aspects: exploratory data analysis (EDA), data preprocessing, model tuning, comparison of different models, and generating insights and recommendations. Unlike general approaches, they have applied recursive feature elimination techniques after selecting their best model with the best hyperparameters. Lastly, they have suggested that future work could be able to explore more advanced machine learning algorithms and techniques in order to enhance the system's performance further. [4]

3.1.2. Recommendation Engine Literature Survey

Recommendation engines are important for improving the user experience in digital platforms by providing personalized suggestions. This section reviews the key research and development techniques that go into creating successful recommendation systems.

3.1.2.1. Collaborative Filtering in Recommendations

In recommendation engines, collaborative filtering is one of the most popular methods. To predict user preferences, it makes use of past user-item interaction data. Collaborative filtering techniques that are item-based or user-based use similarities between users or items to generate predictions.

Item-based collaborative filtering was first presented in a foundational research by Sarwar et al., which demonstrated how well it scales for big datasets. By examining item co-occurrence patterns, this method optimizes system performance in systems with a large number of users and items [5].

3.1.2.2. Advances in Neural Network-Based Models

The integration of neural networks into recommendation systems has improved the ability to capture non-linear and complex user-item interactions. Deep learning frameworks such as

Neural Collaborative Filtering (NCF) replace traditional matrix factorization models with multi-layer perceptrons to model non-linear relationships.

He et al. presented NCF, which applies deep learning to collaborative filtering and demonstrates superior performance over traditional methods. Their approach emphasizes learning user-item interaction patterns through non-linear transformation layers, resulting in more accurate recommendations [6].

3.1.2.3. Hybrid Recommendation Models

Hybrid models combine collaborative and content-based filtering techniques to address the limitations of each. These models improve recommendation accuracy and overcome challenges such as data sparsity and cold-start problems.

Burke proposed a comprehensive taxonomy of hybrid recommendation systems, highlighting the benefits of combining multiple methods. Hybrid systems address the shortcomings of single-method approaches by integrating different data sources and algorithms, resulting in more reliable and accurate recommendations [7].

3.1.3. Chatbot Literature Survey

3.1.3.1. Retrieval-Augmented Generation

Retrieval-augmented generation was firstly introduced by the researchers at Facebook AI. The researchers published a paper called “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. They suggest a new approach that combines parametric and non-parametric memory for language generation. Parametric memory corresponds to a pre-trained seq2seq model and non-parametric memory corresponds to retrieval based knowledge from external sources which is a dense vector index of Wikipedia in their case[8].

The non-parametric memory is accessed using a pre-trained retriever (Dense Passage Retriever, DPR) which retrieves relevant documents based on input. Then the seq2seq model generates an output based on both input and the retrieved relevant documents. The more detailed structure of their RAG approach is visualized in Figure 2.

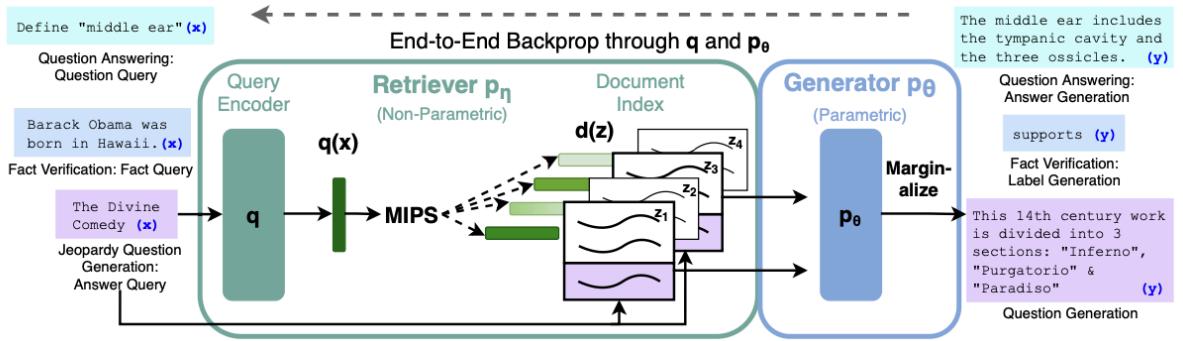


Figure 2: Overview of RAG structure[8]

3.1.3.2. Llama 3

The paper “The Llama 3 Herd of Models” presents Meta’s next-generation family of foundation models called Llama 3. Llama 3 is a collection of models that supports multilinguality, coding, reasoning, and tool usage. The largest model in the collection contains a dense transformer with 405 billion parameters and a context window of up to 120K tokens[9]. Llama 3 is publicly available including pre-trained and post-trained versions of the 405B parameter language model [9]. For the purpose of this project, we mostly focused on the language part of the model.

Figure 3 shows the model architecture of Llama 3. The model development is divided into 2 main stages:

- **Language Model Pre-training:** A large multilingual text corpus is tokenized and used to pre-train a language model for next-token prediction. At this stage, the model gains a deep understanding of language and general knowledge. The Llama 3 is trained using a 405-billion parameter model on 15.6 trillion tokens with an 8,000-token context window, later extended to 128,000 tokens in a continued pre-training stage[9].
- **Language Model Post-training:** The pre-trained model is further fine-tuned using supervised instruction tuning and Direct Preference Optimization (DPO), aligned with human feedback. This stage adds capabilities such as tool use, reasoning, and coding, while also incorporating safety measures[9].

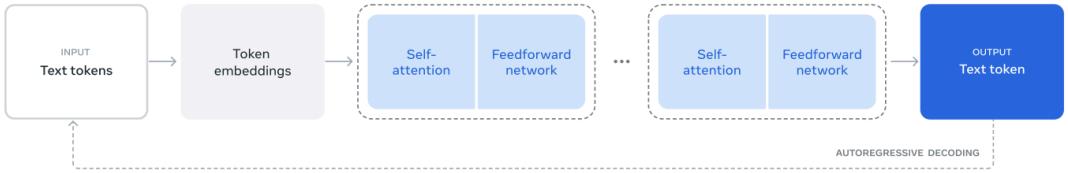


Figure 3: Illustration of the overall architecture and training of Llama 3 [9].

Benchmarking

Llama 3 is a collection of three multilingual language models with 8B, 70B, and 405B parameters. The performance of Llama 3 has been evaluated using plenty of benchmark sets. These datasets contain a wide range of language understanding tasks. As it can be seen in Table 3, it is shown that Llama 3 has almost equal performance compared to leading language models such as GPT-4 [9].

Table 3: Performance of finetuned Llama 3 models on key benchmark evaluations [9].

Category	Benchmark	Llama 3 8B	Gemma 2.9B	Mistral 7B	Llama 3 70B	Mistral 8x22B	GPT 3.5 Turbo	Llama 3 405B	Nemtron 4.340B	GPT-4 ₄₀₉₆	GPT-4 ₄₀	Claude 3.5 Sonnet
General	MMLU (o-shot)	69.4	72.3	61.1	83.6	76.9	70.7	87.3	82.6	85.1	89.1	89.9
	MMLU (b-shot, CoT)	73.0	72.3 [△]	60.5	86.0	79.9	69.8	88.6	78.7 [□]	85.4	88.7	88.3
	MMLU-Pro (5-shot, CoT)	48.3	—	36.9	66.4	56.3	49.2	73.3	62.7	64.8	74.0	77.0
	IEval	80.4	73.6	57.6	87.5	72.7	69.9	88.6	85.1	84.3	85.6	88.0
Code	HumanEval (o-shot)	72.6	54.3	40.2	80.5	75.6	68.0	89.0	73.2	86.6	90.2	92.0
	MBPP EvalPlus (o-shot)	72.8	71.7	49.5	86.0	78.6	82.0	88.6	72.8	83.6	87.8	90.5
Math	GSM8K (8-shot, CoT)	84.5	76.7	53.2	95.1	88.2	81.6	96.8	92.3 [△]	94.2	96.1	96.4 [△]
	MATH (o-shot, CoT)	51.9	44.3	13.0	68.0	54.1	43.1	73.8	41.1	64.5	76.6	71.1
Reasoning	ARC Challenge (o-shot)	83.4	87.6	74.2	94.8	88.7	83.7	96.9	94.6	96.4	96.7	96.7
	GPQA (o-shot, CoT)	32.8	—	28.8	46.7	33.3	30.8	51.1	—	41.4	53.6	59.4
Tool use	BFCL	76.1	—	60.4	84.8	—	85.9	88.5	86.5	88.3	80.5	90.2
	Nexus	38.5	30.0	24.7	56.7	48.5	37.2	58.7	—	50.3	56.1	45.7
Long context	ZeroSCROLLS/QuALITY	81.0	—	—	90.5	—	—	95.2	—	95.2	90.5	90.5
	InfiniteBench/En.MC	65.1	—	—	78.2	—	—	83.4	—	72.1	82.5	—
	NIH/Multi-needle	98.8	—	—	97.5	—	—	98.1	—	100.0	100.0	90.8
Multilingual	MGSM (o-shot, CoT)	68.9	53.2	29.9	86.9	71.1	51.4	91.6	—	85.9	90.5	91.6

3.2. Solution Methods

3.2.1. Churn Prediction Methods

There are four main parts in the churn prediction system. These are “data preparation,” “exploratory data analysis”, “model selection and training,” and “model evaluation.” In the following part, these methods are explained in detail.

3.2.1.1. Data Preparation

Data preparation is a crucial step for developing a churn prediction model. This process includes data cleaning and transformation. In the data cleaning process, missing data are handled using imputation methods such as replacing missing numerical values with the mean or median and categorical variables with the most frequent values. Outliers are also identified using Z-scores. Those data cleaning steps will be used in our churn prediction model as well. Feature engineering is also another important step. This step involves scaling and normalization, which ensures that the numerical features in the data are on the same scale. Categorical variables are also converted into numerical formats like one-hot encoding or label encoding, which helps to transform the data in order to be suitable for machine learning models. All these data preprocessing methods will be used in our churn prediction model development process.

3.2.1.2. Exploratory Data Analysis

The exploratory data analysis part involves exploring and uncovering patterns, trends, and relationships in the data that might be helpful to detect churn users. This part mainly consists of statistical summary of the data, data visualization, and correlation analysis. For the visualization part, plotting the churn rate across different customer segments, such as age groups or regions, helps reveal key patterns in the data. In addition, histograms and box plots are used for identifying anomalies in the dataset. For the statistical analysis part, heatmaps are used to analyze the correlation between variables. For example, the correlation heatmap can reveal important relationships between variables such as churn and customer satisfaction scores. Those methods will be applied in our churn prediction models.

3.2.1.3. Model Selection and Training

Choosing the right model for the churn prediction is a crucial step in the model development process. At the first step, logistic regression and decision trees are good starting points. They are simple and cost-efficient compared to more complex models like random forests and neural networks. Based on the accuracy of the model on the training and test data, if there is a bias in the basic models mentioned above, more complex models like shallow and deep neural networks can be used for the prediction. This procedure is what we plan to apply in our churn prediction model development process.

The first step of model training is to initialize some hyperparameters based on the model type. For the decision tree, “max depth”, “min samples split,” and “max features” are considered hyperparameters. For the logistic regression, “regularization penalty,” “regularization strength,” “solver,” and “max iteration” are considered widely used hyperparameters. For the neural networks, “number of layers,” “number of units per layer,” “dropout rate,” “learning rate,” “optimizer,” “batch size,” “number of epochs”, and “regularization parameters” are some important parameters used to train the model. These hyperparameters listed for each model type will be used in the hyperparameter tuning stage of our churn prediction model training. The next step is evaluating the model performance on both training and test data based on some metrics that will be explained in detail in the “Model Evaluation” part. After evaluating the model, hyperparameters are changed to improve prediction accuracy. This tuning and evaluation process continues iteratively until reaching the optimal model.

3.2.1.4. Model Evaluation

Evaluation of the churn prediction model is measured by the metrics “accuracy,” “precision,” “recall,” and “F1-score.” Before evaluation, the dataset is split into training, validation and test data. After that, a confusion matrix that shows true positives, true negatives, false positives, and false negatives is created. Using a confusion matrix, key evaluation metrics “accuracy,” “precision,” “recall,” and “F1-score” are calculated using the formulas shown in Figure 4.

	Function Name	Formula
	Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
	Precision	$\frac{TP}{TP + FP}$
	Recall	$\frac{TP}{TP + FN}$
	F1 score	$\frac{2.Precision.Recall}{Precision + Recall}$

Figure 4: Confusion matrix and metric formulas. [10]

Accuracy represents the percentage of total correct predictions (churners and non-churners) out of all predictions. Precision measures how many of the predicted churners were actual

churners. Recall measures how many of the actual churners were correctly identified. The F1 score balances precision and recall, providing a single metric to assess model performance.

3.2.2. Recommendation Engine Methods

The six main stages of the recommendation system are data collection, data preprocessing, model selection, implementation, testing and evaluation, and integration. The following section provides a detailed explanation of each of these stages.

3.2.2.1. Data Collection

The first step in building a recommendation engine is to collect data. In this phase, the company is going to collect the data of the users while they are interacting with their system, this data contains their viewing history, preferences, ratings, and implicit feedback like watch time and click events, etc. The data points in question come from user profiles and activity logs on the site. How this data is aggregated and structured affects the accuracy and the ability to customize recommendations. The company always provides the dataset files, but we are responsible for organizing the data according to our needs.

3.2.2.2. Data Preprocessing

Once the data has been gathered, it is processed in advance of being used by machine learning algorithms. In this phase, any missing data is addressed. This may occur, for instance, through the substitution of default placeholders for null entries or the utilization of average values to supplement missing ratings. In order to ensure the uniformity of ranges, the features are subjected to scaling or normalization. Furthermore, encoding methods, such as label encoding and one-hot encoding, are employed to transform categorical data into numerical representations. The data is rendered fit for model training through the application of these preprocessing techniques, whereby it is made clean, organized, and prepared for training.

3.2.2.3. Model Selection

Model selection involves identifying suitable algorithms for the recommendation system. Collaborative filtering, content-based filtering, and hybrid methods are considered for this project. Collaborative filtering utilizes user-item interaction matrices, while content-based filtering focuses on item attributes. Hybrid approaches, such as combining collaborative and content-based methods, aim to improve recommendation accuracy and handle limitations like

the cold-start problem. More advanced methods, like neural network-based hybrid models, are explored for a deeper understanding of user-item relationships.

The DNN-Softmax recommendation engine combines a deep neural network (DNN) with a softmax layer to generate probabilistic outputs for item recommendations. Input features, such as user preferences and item characteristics, are transformed through hidden layers, capturing complex interactions. The final softmax layer computes probabilities, as shown in Figure 5, with the formula:

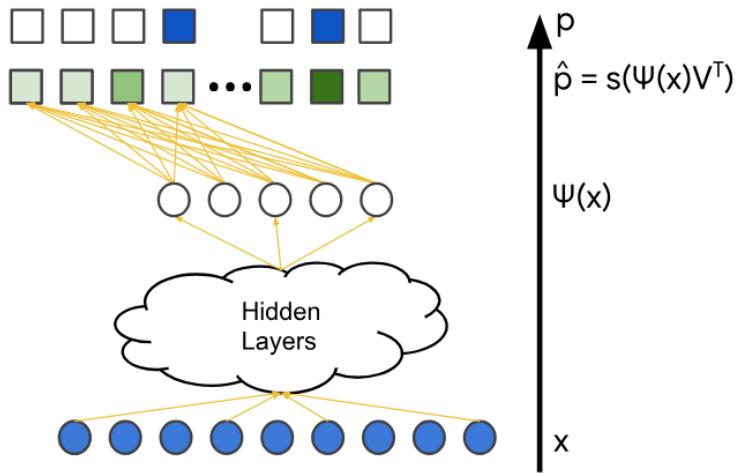


Figure 5: Softmax Function and Layer Illustration [14].

In order to optimize performance, it was necessary to adjust the hyperparameters. In order to guarantee successful learning, ReLU activation was employed in conjunction with adjustments to the learning rate, batch size, and L2 regularization. Cross-entropy loss and evaluation measures, including Precision@K, Recall@K, and nDCG, were employed to assess and rank the relevance of the suggestions. The selection of hyperparameters was guided by the metrics of validation, following the division of the dataset into training, validation, and test sets.

3.2.2.4. Implementation

Python and machine learning libraries such as TensorFlow or PyTorch are applied during the implementation phase to create and optimize the selected algorithms. The prepared data is employed for the development and training of models such as Singular Value Decomposition (SVD), Item-Based Collaborative Filtering (IBCF), and hybrid neural network models. In

order to achieve optimal results, the hyperparameters, including matrix dimensions, regularization strengths, and learning rates, are adjusted. To advance the quality of metadata-based suggestions, this phase incorporates sophisticated elements such as natural language processing (NLP).

3.2.2.5. Testing and Evaluation

Following implementation, a number of metrics, including precision, recall, and Root Mean squared error (RMSE), are calculated and evaluated to ascertain the model's performance. The accuracy of the system's user preference prediction is evaluated through the aforementioned metrics. To allow for a comprehensive evaluation, the data set is partitioned into three subsets: a training set, a validation set, and a testing set. To improve the models and increase their accuracy and relevance, iterative testing is conducted. Prior to deployment, this procedure confirms that the system meets the requisite quality criteria.

3.2.2.6. Integration

The recommendation engine will be integrated with other system elements, such as the chatbot and churn prediction, in the final stage of the process. The integration facilitates seamless communication between diverse project components, thereby providing a uniform user experience. To guarantee optimal functionality of the recommendation engine within the broader application framework, comprehensive testing is conducted from inception to completion.

3.2.3. Chatbot Methods

There are four main methods that were planned to be used in our first chatbot design. These are “web scraping”, “synthetic data generation”, “fine tuning pre-trained model”, “model evaluation”. However, as explained in the previous parts, our design strategy has changed, there are also RAG methods used in our chatbot implementation which are basically retrieval and generation. In addition, because there is also a possibility for us to fine-tune a pretrained model to locally run a large language model, fine tuning methods have been also researched and can be used in the future. In the following parts, all those methods are explained in detail.

3.2.3.1. Web Scraping

First step is making an HTTP request to the website that will be scraped from to obtain the HTML form. Next step is parsing the HTML file in order to traverse and manipulate the HTML structure easily. The desired data is extracted from the specific parts of the parsed HTML. Finally, the extracted data is stored in an appropriate format such as CSV, JSON etc. Beautiful Soup which is a python package for parsing HTML was used to implement web scraping. Those steps have been implemented to obtain the movie metadata and reviews from the “Beyazperde” website.

3.2.3.2. Synthetic Data Generation

Synthetic data generation method was used to obtain meaningful, natural, human-like request-response pairs dataset. The content of the request-response pairs is movie recommendation based on the movie summary and human-written reviews obtained from a Beyazperde website. To automate the data generation process, ChatGPT and Gemini API have been used. These LLMs were given with movie summary and reviews dataset and were prompted properly to generate meaningful recommendations supported by the movie metadata.

3.2.3.3. Fine Tuning Pre-trained Model

Data Preprocessing

First stage of the fine tuning process is data preprocessing. Data is cleaned by handling special characters and missing values. Next step is Tokenization. Tokenization splits the text into smaller units, such as words or subwords that the model can process. Those processes are applied to our synthetic datasets that consist of movie recommendation request-response pairs.

Training the Model

Fine tuning is the process of training a pre-trained model by adjusting model weights based on the new data. Firstly, hyperparameters such as “learning rate”, “batch size”, “optimizer”, “number of epochs”, “regularization parameter”, “dropout rate” are tuned. Then, performance of the model on both training and validations dataset is monitored to prevent overfitting. There are also different training strategies such as full model fine tuning, parameter efficient fine tuning(Low-Rank Adaption(LoRA), adapter layers, prefix tuning). These training

strategies will be evaluated to decide which one to use in our movie recommendation chatbot training process.

Model Evaluation

After monitoring the performance of the model via training metrics, and loss curves on training and test data, hyperparameters are tuned based on the results. Then, the model is trained with these new hyperparameters and monitored. This iterative process continues until the loss function converges to global minima, which means the model has high accuracy.

3.2.3.4. Retrieval-Augmented Generation

Retrieval-augmented generation(RAG) is a method for enhancing the quality and relevance of generated text of LLMs by providing additional information. It consists of two main components: retrieval and generation. In the retrieval process, relevant documents from an external knowledge base are retrieved. In the generation process, the large language models take these relevant documents as input and generate a response, using the information provided by the retrieved documents.

- **Knowledge Base**

The knowledge base is a very crucial component of a RAG system. Knowledge base provides a short-term memory to a large language model and enables it to generate answers based on the provided information. It consists of structured or unstructured documents. In our design, structured documents are used. Documents are structured in a way that there is a list of metadata(name, genre, summary etc.) and reviews for each movie in our platform. These documents have been created combining the information from both Dsmart-GO platform and Beyazperde website.

- **Retrieval**

The retrieval component is responsible for searching for the most relevant documents for a given user query. Each document is stored as vector embeddings which is a meaningful numeric representation of the documents. User query is also embedded into a vector. The vector similarity between the user query vector and document vectors is calculated using cosine similarity(Figure 6). Then, the most relevant documents are retrieved. The visualization of the embedding and retrieval processes are shown in Figure 7.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Figure 6: Cosine of the angle(θ) between two vectors(A and B)[11]

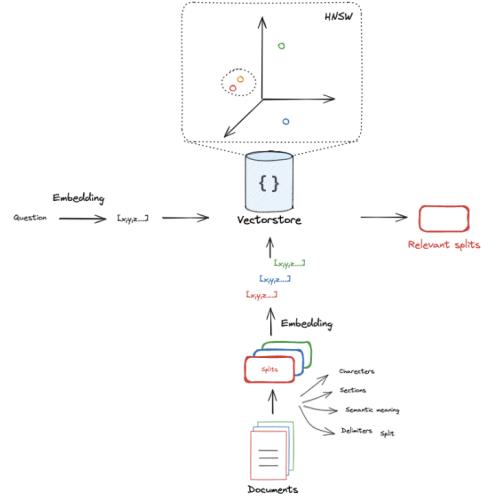


Figure 7: Retrieval Process[12]

- **Generation**

The generation component is responsible for generating responses based on the retrieved documents which are related to the user query. The retrieved content and the user query are concatenated into a single input for the model. Then the model generates the output based on the given information.

The visualization of the workflow of a RAG system is shown in Figure 8. It contains both the retrieval and generation parts.

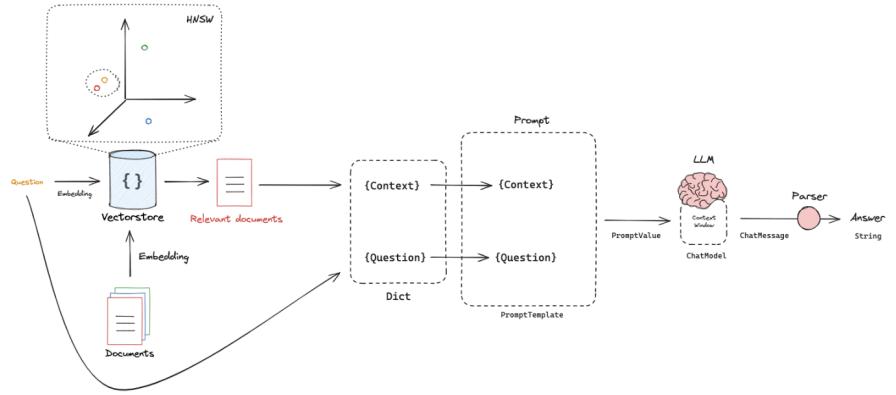


Figure 8: Workflow of a RAG system [13].

3.2.3.5. Final Evaluation

It is crucial to evaluate the fine-tuned model performance in real world-tasks. This is done by both automated metrics and human evaluation. For the automated evaluation, standard metrics such as BLEU, ROUGE, and Perplexity were planned to be used in order to determine the model's linguistic quality, fluency, and relevance. However, those metrics are not perfect identifiers for the model performance. Therefore, human evaluation is also needed. For that purpose, a user feedback mechanism will be used where real users interact with the chatbot and answer some questions such as "Was this answer useful?" "Did you like the recommendation?" etc. will be used. The RAG chatbot is not yet available to real-world users. Therefore, the user feedback mechanism is going to be applied later.

4. ANALYSIS AND MODELING

4.1. System Factors

4.1.1. System Factors for Churn Prediction System

In the churn prediction phase of the project, the most important factor that affects the performance of the model is the quality of the dataset. The performance of regression models and segmentation processes is crucial since it will determine the understanding of the relations of the features in the dataset, and this will determine our performance on the feature engineering phase. The phase that we will extract new features is very important since the performance of the classification models we are going to use depends on the quality of the dataset after it. Also we need to consider the computing power limitations. Even if we are capable of extracting the best possible features, our system's RAM and CPU power will not be able to handle it in a waitable amount of time, then we will have to work a smaller portion of the dataset in order to proceed. This situation is likely to affect the model performance due to ignoring large amounts of data.

4.1.2. System Factors for Recommendation Engine

In the personalized recommendation part, the reliability of the data used, the methods chosen, and the system's ability to handle user preferences are just a few of the variables that affect the performance of the recommendation engine. Obtaining high-quality data—like user ratings and watching preferences—and choosing the right data analysis methods are important. The system's ability to identify comparable items and position information based on user activity or genres can have a consequence on how effective it is. The system's capacity to handle huge data quantities is another important aspect. If the system doesn't have enough storage or processing power, it might have to use smaller data sets, which could lower the quality of the suggestions. In the similarity-based recommendation phase, the selection of vector distance algorithms is likely to affect performance. Other factors, such as selecting the relevant content and then shaping the text to be vectorized, adjusting queries effectively, and giving weight to the specifying factors like genres, etc., are some factors that will highly affect the performance.

4.1.3. System Factors for RAG Chatbot

Dataset Quality: Document quality plays a very crucial role in RAG systems. They should contain necessary information needed as much as possible. Documents must be clean and well-organized to avoid irrelevant information. In addition, size of the documents are also important because there is a context window limit for each large language model.

Quality of Retrieval System: The performance of the RAG system mostly depends on how well the retrieval system retrieves relevant documents based on the input query. The documents are stored as vector embeddings. These vector embeddings should contain as much semantic information as possible to enhance the retrieval process.

Capability of Large Language Model: RAG systems are designed for a specific domain(in our case movie recommendation). Therefore, the model should have the ability to generate context-specific responses. If needed the model should be fine-tuned with domain specific dataset. Since, we use ChatGPT 4o-mini as our model and it is very comfortable with generating movie-related responses. We haven't fine-tuned the model at this stage.

4.2. How System Works

Our system's recommendation engine uses machine learning to provide consumers with customized recommendations. It makes use of information about the products themselves as well as data about what customers view and their preferences. This data is converted into vectors by the system and saved in a vector database. The system recommends related items from the database when a user sees something. The engine uses generative AI to organize the results if AI support is enabled, making the recommendations even more precise and beneficial. Users receive the greatest and most relevant recommendations in such a way.

Similarity-based recommendation engine works with vectorization techniques. Firstly, we prepare the TV content data that has been gathered for us by the provider company. Later, we gather all the relevant data about a TV content in one text then vectorize it and store it in a vector database. So when a customer is watching some content, since the system already has that content's vector in the vector database, the system queries the similar results from the database then recommends the closest ones to the user. Also if the AI support is enabled, the

system queries more results than needed and sorts them with the help of the generative AI, lastly the system recommends requested numbers of recommendations from the top of the sorted query result.

The churn prediction model tries to predict if the customer is going to leave the service or not. And it is capable of doing that via considering the customer's behaviors. The datasets have been gathered for us by the provider company about the customer behaviors that are going to be prepared and preprocessed in order to train the model that is going to make these predictions. These dataset includes user list data, system usage data, login data, watchlist data and like/dislike data. After applying some feature extraction techniques and preparing the data, one of the best performing classification models will be selected to deploy. Best performing model will be used to predict the future churned customers. The system will train itself in a specific period of time with the new data that comes.

Movie Recommendation Chatbot is designed to use RAG methods. We use a vector database to store movie information(name, genre, cast, summary etc.) and movie reviews scraped from Beyazperde movie content website. Our chatbot uses this vector database to retrieve relevant documents based on the user query. To explain the workflow in detail, firstly, the user query is embedded into a vector. The similarity check between the query and the documents is handled to retrieve the most relevant documents. Then, the combination of the query and the relevant documents are sent to the ChatGPT 4o-mini model using OpenAI API. Finally, the response of the model is shown to the user. Chat history is also stored in the local memory. The previous user and chatbot messages are also given to the model in the upcoming prompts. In this way, users can experience a natural conversation rather than a question answering system.

4.3. Modelling

4.3.1.1. Modelling of Churn Prediction Model

The churn prediction system model was designed to use user usage data to predict customer loyalty and chance of service cancellation. The system uses multiple data sources and processes the data to engineer meaningful features, and trains machine learning models to predict the churn status of users. The model follows a structured approach:

Data Preparation: Relevant data, such as login logs, watch history, and subscription details, were extracted, cleaned, and merged to create a comprehensive dataset.

Feature Engineering: Custom metrics, including such as TotalUsagePeriod, AverageContentCompleteRatio, and UniqueDeviceTypes, were generated to capture customer behavior.

Machine Learning Pipeline: The final dataset was used to train multiple machine learning models (Random Forests, SVM, etc.), which were evaluated based on accuracy, precision, recall, and F1 scores.

Prediction and Evaluation: The best-performing model was selected for predicting churn and fine-tuned to achieve optimal results.

4.3.1.2. Modelling of the Recommendation Engine

According to user behavior and content information, the recommendation engine was created to offer customized recommendations. Several important steps were included in the modeling process:

Data Preparation: User interaction data, such as likes, and ratings, was provided to us, and then we cleaned it. Categorical features were encoded, and numerical features were scaled to ensure uniformity across the dataset.

Feature Engineering: Simple features like movie likes, favorite genres, and user preferences were created to improve the accuracy of the recommendations.

Algorithm Selection: Both collaborative filtering and content-based filtering approaches were explored. Matrix factorization techniques like Singular Value Decomposition (SVD) were used for collaborative filtering, while content-based filtering utilized metadata such as genre, cast, and descriptions to make suggestions.

DNN-Softmax Recommendation Engine: A deep neural network (DNN) with a softmax layer was implemented to handle complex user-item interactions. Input features included user preferences and item characteristics. The network's hidden layers captured non-linear relationships, while the softmax layer generated probabilistic outputs for recommendations. Hyperparameters, such as the number of layers, learning rate, and regularization strength, were tuned to optimize performance.

Evaluation: The model was evaluated using metrics such as precision, recall, and Root Mean Squared Error (RMSE). User interaction data was split into training, validation, and test sets to ensure robust evaluation and prevent overfitting.

4.3.1.3. Modelling of the Chatbot

There are three main components in our movie recommendation chatbot design: vector database, retrieval, generation. Here is the detailed explanation for each of them:

Vector Database: Documents that contain movie information are stored as vector embeddings in a vector database. For the vector database, we preferred to use “Pinecone” because it is free and user-friendly.

Retrieval: This component is responsible for retrieving the most relevant documents by checking the vector similarity between the documents and the user query. To retrieve the documents, Pinecone API is used. These documents provide a short-term memory to the language model. By using this memory, the model generates conceptual information which the model normally can't correctly provide in the answer.

Generation: Combination of last user query, retrieved documents and the chat history is sent to ChatGPT 4o-mini using OpenAI API. Then, the response of the model is shown to the user via a user-friendly user interface.

4.3.2. UML (Unified Modeling Language) Diagrams

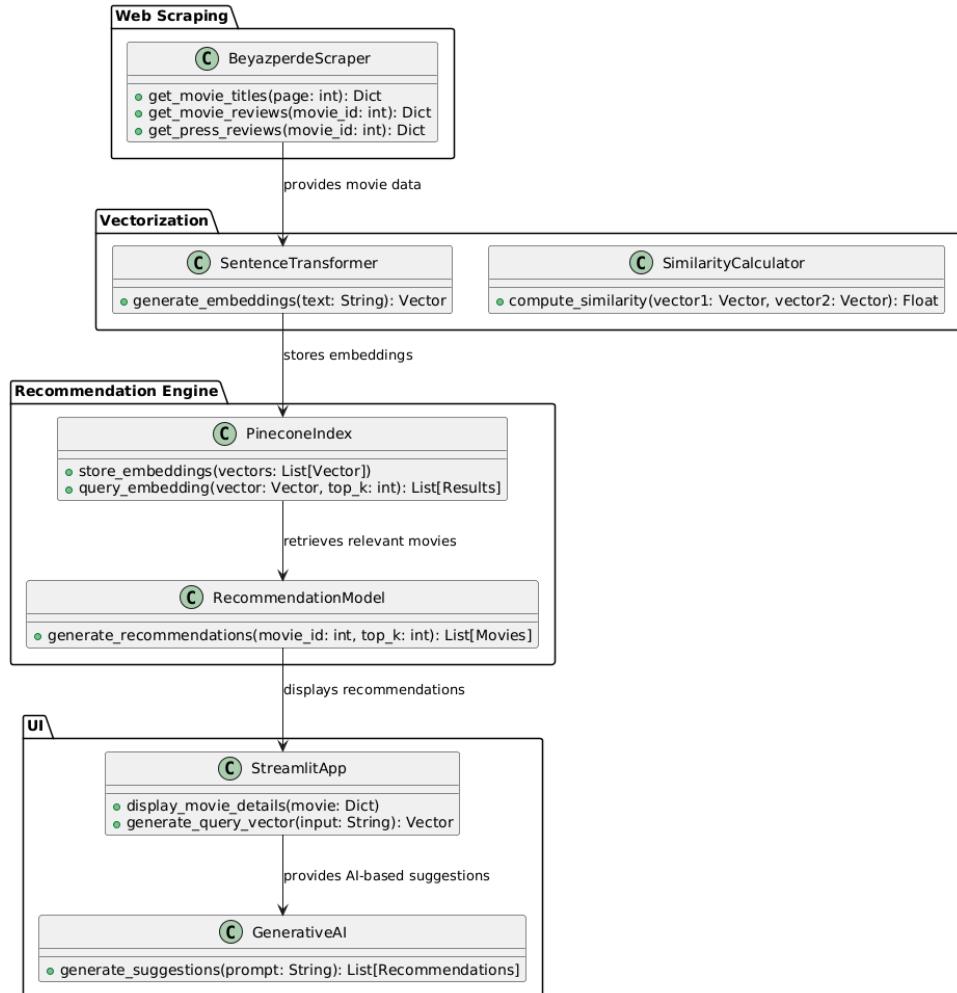


Figure 9: Content-Based Recommendation Engine UML Diagram

In Figure 9, the UML's content-based recommendation engine is initiated by three fundamental processes: web scraping, vectorization for embeddings, and a recommendation engine to rate movies. The integration of artificial intelligence (AI) with the user interface (UI) aims to facilitate the accessibility and utilization of the system's concepts.

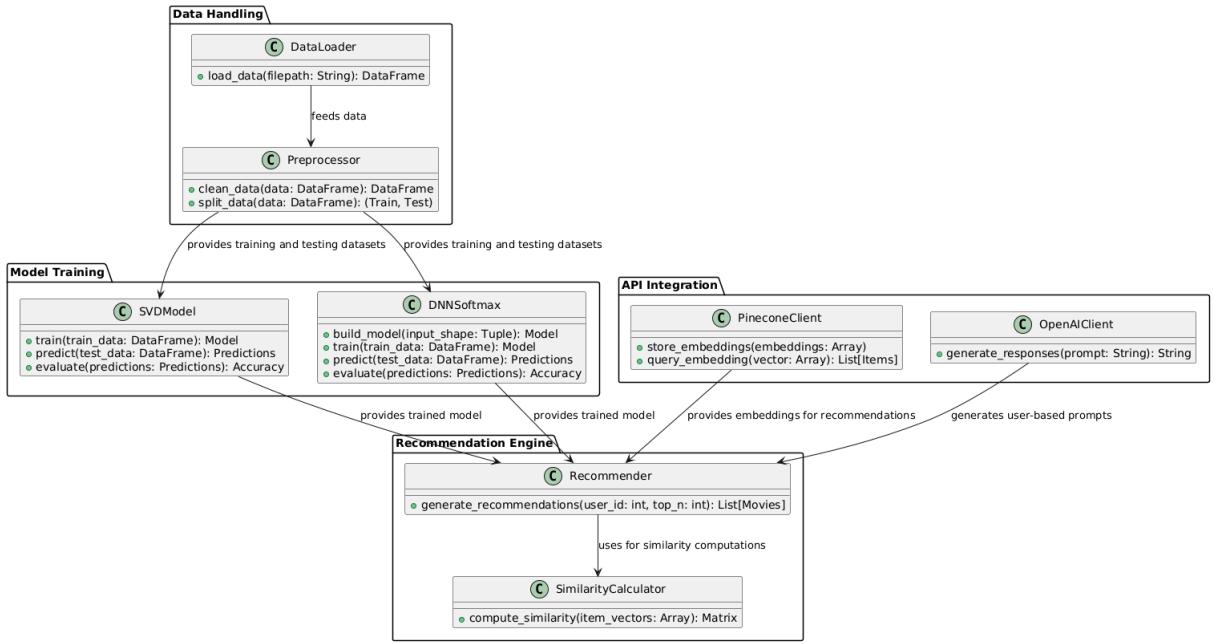


Figure 10: Deep Neural Network Recommendation Engine

The Figure 10 shows UML illustrates a recommendation system that utilizes DNN-Softmax and SVD models. Once the data has undergone a thorough cleansing, division, and feeding into the models, the models utilize API integrations for prompts and embeddings, as well as similarity calculations, to provide recommendations.

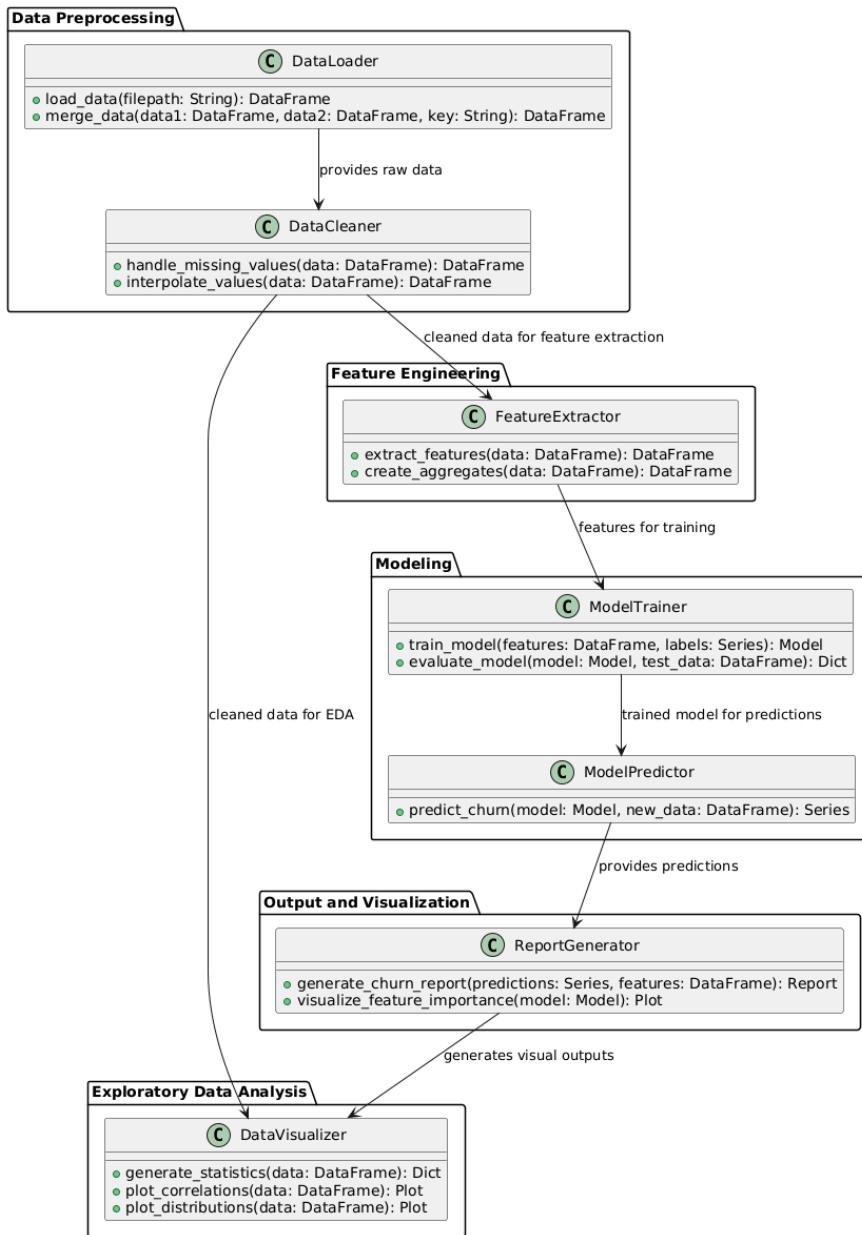


Figure 11: Churn Prediction UML Diagram

In Figure 11 the UML, a churn prediction workflow is illustrated, ranging from data loading and cleaning to feature extraction, modeling, and producing churn reports with visual insights. The UML employs sequential connections between components to provide a clear illustration of data flow and dependencies.

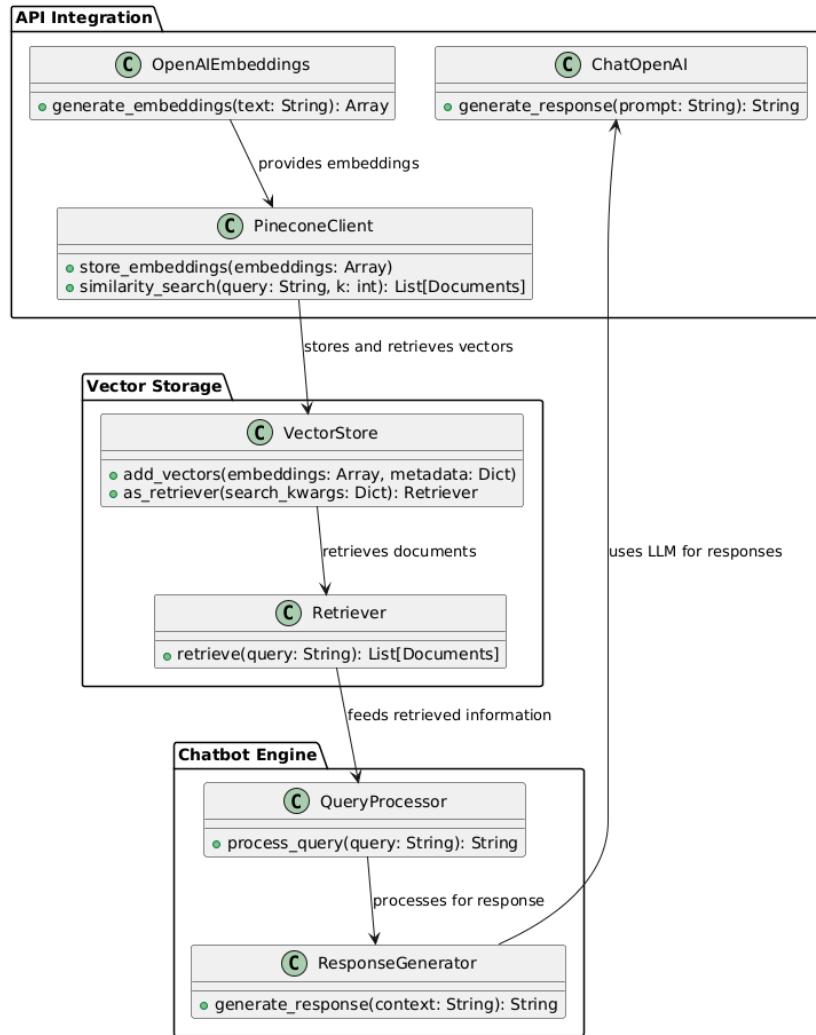


Figure 12: ChatBot UML Diagram

The Figure 12 represents UML diagram illustrates a chatbot system. It comprises a chatbot engine, which processes questions and provides answers; vector storage, which stores and recovers documents; and API integration, which facilitates embeddings and LLM replies. The system's functionality is derived from the sequential interaction of these components.

5. DESIGN, IMPLEMENTATION AND TESTING

5.1. Design

5.1.1. Design of the Churn Prediction Model

The design of the churn prediction model included three key steps. The hard work is to prepare the dataset according to our goal. The final dataset is needed to have relevant information about the users patterns related to the loyalty to the service.

Firstly, the dataset was to contain information about the usage behaviour of the customer inside of the specific periods determined by the customer's package start and end dates. By that, the model would understand the customer's usage patterns inside of a package and will predict if the customer will purchase a new packet after the current one expires or not.

Accordingly, the prepared dataset is visualized with different types of plots in order to understand the features' relevance to the target label or to the other features. The approach known as exploratory data analysis will help us to become more familiar with the data, and the data will be more descriptive. Also, it will help us to build a stronger dataset in case the modeling phase does not work as expected.

Finally, the final dataset will be fitted to the best-performing machine learning model. After, the accuracy reports of several machine learning algorithms are compared, and the best one is used after hyperparameter fine-tuning.

5.1.2. Design of the Recommendation Engine

The design of the recommendation engine included several key steps to ensure accurate and personalized content delivery. The main focus was on creating a solid dataset that includes important information about both people and things, including past interactions, item attributes, and user preferences. In order to handle various recommendation techniques, the dataset was created with features including user ratings, genres, tags, and other information.

First of all, the system combined collaborative and content-based filtering methods with other strategies including item-based and user-based collaborative filtering (UBCF and IBCF).

While collaborative filtering techniques used user and item interactions to find patterns, the content-based model used item attributes to suggest comparable things based on user preferences. To improve the recommendation accuracy, a deep learning strategy involving a DNN with a softmax output layer was also used.

In order to build a solid foundation for modeling, exploratory data analysis was conducted to visualize and understand the relationships between features and identify patterns in user behavior. In order to further improve the dataset, this stage directed feature engineering and offered insights about feature significance.

In the end, hyperparameter optimization was used to maximize the most successful approach after the models' performances were evaluated and compared. The recommendation engine was able to maintain its scalability, flexibility, and ability to provide extremely relevant suggestions because of its architecture.

5.1.3. Design of the Chatbot

Our movie recommendation chatbot has four main parts: vector store, retrieval, generation and user interface. The system is designed to provide accurate and contextually meaningful movie recommendations by using the relevant documents based on the user query.

Vector Store: Pinecone vector database is used to store movie information as vector embeddings. This provides us to check the similarity between the movie documents and the user query. An example documentation from our Pinecone database is shown in Figure 13. The values correspond to the vector embedding of that document.

ID	VALUES
71089244-c...	-0.0134543963, 0.0545412451, -0.00631775754, 0.000956352393, -0.025728080... □

METADATA
Cast: "Heath Ledger & Michael Caine & Christian Bale"
Directors: "Christopher Nolan"
Genre: "gerilim aksiyon-macera"
Name: "Kara Şövalye"
Summary: "Joker'in ortaya çıkmasıyla birlikte Gotham sokaklarında tekrar kaos hakim olur. Batman, Teğmen Gordon ve
text: "\nİsim: Kara Şövalye\nTür: gerilim aksiyon-macera\nKonu: Joker'in ortaya çıkmasıyla birlikte Gotham sokaklarında

Figure 13: Pinecone Document

Retrieval: For the retrieval part, there is a condition check if the document retrieval is necessary or not. This condition is based on the user query. If it is necessary, it runs the retrieval code which will be explained later in detail.

Generation: If relevant documents are retrieved from the Pinecone vector database, query documents and the chat history are combined and sent to ChatGPT 4o-mini model by using ChatGPT API.

User Interface: For the UI design, we used Streamlit to provide a user-friendly experience. Figure 14 shows our Chatbot UI design.



Figure 14: Chatbot UI Design

5.2. Implementation

5.2.1 Implementation of the Churn Prediction Model

The datasets that are going to be used on developing a predictive model was containing:

- User Log-In Data (login dates and device information)
- User Watch Data (content watch information such as time, duration, watched content)
- User Package Data (user' subscription information)
- User Liked Content Data (not used in churn model)
- User Favorite Content Data (not used in churn model)
- Content Information Data (content duration, genre, directors etc.)

One of the main issues for getting the final dataset for model training was to decide how to determine if a customer is churned or not. The process of determining churn was keeping track of if a customer has purchased a new package after the current one of his/hers expired, if yes, the customer is labeled as churned, otherwise not. Additionally, even if a customer has a purchase after the current package expires, the customer's hold on for the next purchase matters, if it's more than 60 days, this customer is also labelled as churned. This labeling process is done on the package dataset.

Then, the watch data for each customer is partitioned according to the customer's package's start and end dates. By that, we get the user's usage behavior according to each package period and its churn information. After the feature engineering phase for this part, several new features are added such as NumbersOfProfiles, AppVariety, ContentVariety, AverageUsageDuration, TotalUsagePeriod, AverageContentCompleteRatio. After these metrics are calculated, the dataset is merged into the package dataset where the churn is determined.

Accordingly, the login data for each customer is partitioned according to the customer's package's start and end dates just as we did to the watch data with the same hope of capturing behavior patterns for each customer. The new features added after feature engineering are TotalLogins, AverageLoginsPerDay, UniqueDevices, UniqueDeviceTypes. At the end, this data is merged to the previously merged dataset to get the final data. But there was a unmatching entries for login data for some customers which causes Nan values. This situation

is handled by both filling them with the column's mean values and predicting them linearly, and the filling with the mean approach worked slightly better in terms of model accuracy.

Finally, the prepared dataset is used to train the models using the following machine learning algorithms:

- Random Forests
- SVM (Support Vector Machines)
- Logistic Regression
- K-Nearest Neighbors
- Decision Trees

The train-test accuracy plots, classification reports, confusion matrices and feature importance plot (if it exists) are provided for these models, and the best one is selected for hyperparameter fine-tuning. Lastly, the hyperparameter fine-tuning process is done on the best model.

5.2.1.2. Implementation of the Recommendation Engine

The implementation of the recommendation engine involved multiple steps, integrating various techniques to provide reliable performance. The datasets used included user interaction data such as ratings, tags, and timestamps (from datasets like ratings.csv and tags.csv) and content metadata like genres, descriptions, and production details (from datasets like movies.csv and ubcfv2.csv). Additional datasets such as ibcf.csv and dsmart_favorites_report.csv were also used to provide richer context. This data was processed and improved through feature engineering, utilizing scripts for data collection (e.g., web scraping from Beyazperde) and secure storage mechanisms.

Both content-based and user-based collaborative filtering strategies were used for the recommendation models. To handle complicated interactions and increase accuracy, a deep learning model comprising a DNN and a softmax output layer was also created. Each model was trained and evaluated against test datasets for accuracy and precision in predictions. In order to create a solid and organized dataset for modeling, exploratory data analysis was carried out to better understand the correlations between attributes.

The system was deployed using a Streamlit-based user interface. To improve outcomes, the user interface included embeddings, similarity metrics (cosine, dot product, and Euclidean), and optional Google Generative AI support.

As a stand-alone recommendation system, the DNN with a softmax output layer was put into practice to manage complicated relationships and provide precise predictions. A separate Streamlit-based user interface was used to install the system, and to improve results, it included embeddings, similarity metrics (cosine, dot product, and Euclidean), and optional Google Generative AI support. Two adaptable and scalable technologies that can each provide accurate and individualized suggestions based on user preferences are the outcome of these efforts.

5.2.1.3. Implementation of the Movie Recommendation Chatbot

The implementation of the chatbot is all about combining the necessary RAG components(retrieval, generation) and vector database into a single working system. We preferred to use the Langchain and LangGraph frameworks.

These frameworks provide a general structure for building various kinds of chatbot. It is easy to customize which model to use and which vector database service to use. There is a standardized component structure. LangGraph provides a very useful way to orchestrate different components like retrieval generation, other tools and agents considering conditional flow. Figure 15 shows our chatbot implementation as a graph that is implemented using LangGraph.

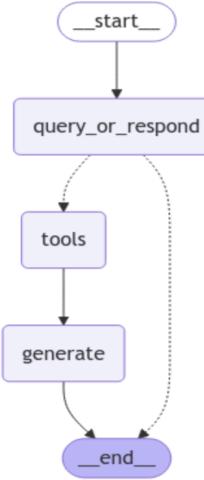


Figure 15: Chatbot Graph Design

“query_or_respond” Node: This node decides whether the system should retrieve relevant documents from our vector database or it should directly respond to the user query. To implement this, a retrieval function is implemented that checks the vector similarity and retrieves most relevant documents from the vector database. This function is stored in the “tools” node. When the system decides to retrieve documents, it goes to the “tools” node.

“tools” Node: Tools node is responsible for storing all the tools. In our design, there is only one tool which is retriever.

“generate” Node: Generate node is responsible for combining the system prompt, retrieved documents, user query and chat history into a single prompt template and sending this template to our large language model(ChatGPT 4o-mini).

5.3. Testing

5.3.1. Testing Phase of Churn Prediction Model

The testing phase on developing a churn model was always the consideration for every step or iteration of the process. In the preprocessing phase, we manually checked if the data is merged as expected or how many values were mismatched, then handled accordingly. In the

feature engineering part, we again manually checked if the features are calculated accurately as we designed throughout different samples. And in the model phase part, different models are tested and will be tested with different hyperparameters and will be validated via charts like confusion matrix, train-test accuracy plots, etc.

5.3.2 Testing Phase of Recommendation Engine

The testing phase of the recommendation engine focused on validating its functionality as part of the overall project. Data merging was carefully reviewed throughout the pretreatment phase to make sure user-item interactions and content metadata were appropriately matched and that any differences were fixed. For confirmation that they had been created correctly, feature engineering outputs, including embeddings and similarity metrics, were examined. During the testing phase, measures such as accuracy and recall were used to evaluate the models, which included content-based, collaborative filtering (both item-based and user-based), and the DNN with a softmax output layer.

5.3.3 Testing Phase of the Chatbot

The testing phase of the movie recommendation chatbot was mostly handled with human feedback. Several scenarios has been tried using different Turkish queries like “Aksiyon filmi izlemek istiyorum. Ne önerirsin?”, “Christopher Nolan'in yönettiği filmlerden önerir misin?” etc. Responses of the chatbot have been compared with the information provided in the documents retrieved. As a result, the chatbot responses are contextually meaningful and contain lots of information from the retrieved documents.

6. RESULTS

6.1. Results of Churn Prediction Model

After the model training phase, our dataset was containing the following features:

1. **UserId**: A unique identifier for each customer in the dataset.
2. **NumbersOfProfiles**: The total number of profiles created under a single user account.
3. **AppVariety**: The number of different apps or platforms the user accessed the service from.
4. **ContentVariety**: The number of different contents consumed by the user.
5. **AverageUsageDuration**: The average duration of time the user spends on the platform per session.
6. **TotalUsagePeriod**: The total time span the user has been actively using the service.
7. **AverageContentCompleteRatio**: The average proportion of content completed by the user.
8. **Churn**: A binary indicator where 1 means the user has stopped using the service and 0 means they are still active. (Target Label)
9. **TotalLogins**: The total number of times the user has logged into the platform.
10. **AverageLoginsPerDay**: The average number of logins per day during the user's active period.
11. **UniqueDevices**: The total number of unique devices used to access the service.
12. **UniqueDeviceTypes**: The total number of unique types of devices (e.g., smartphone, tablet, desktop) used by the user.

The distribution of the churn information on the dataset was balanced. The percentage of churned and non-churned customers can be seen in Figure 16.

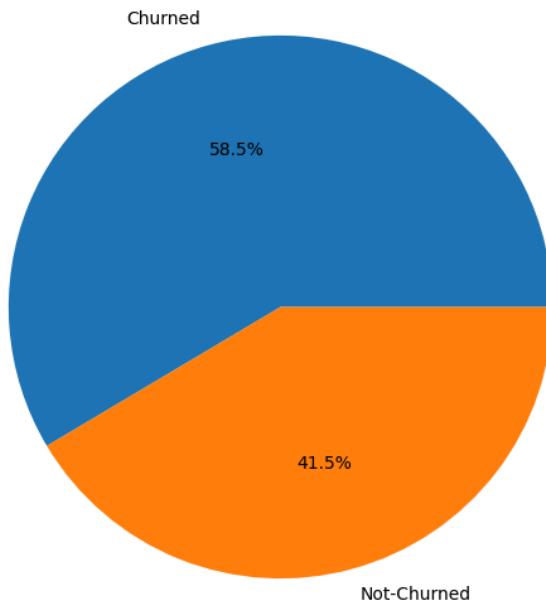


Figure 16: Distribution of churned and non-churned customers.

There were observable relationships between some of the features of the dataset and the churn information. Such as in the content variety, which indicates the number of different contents consumed by the customer, while this increases, the customer is less likely to churn. The correlation can be seen in Figure 17 and Figure 18.

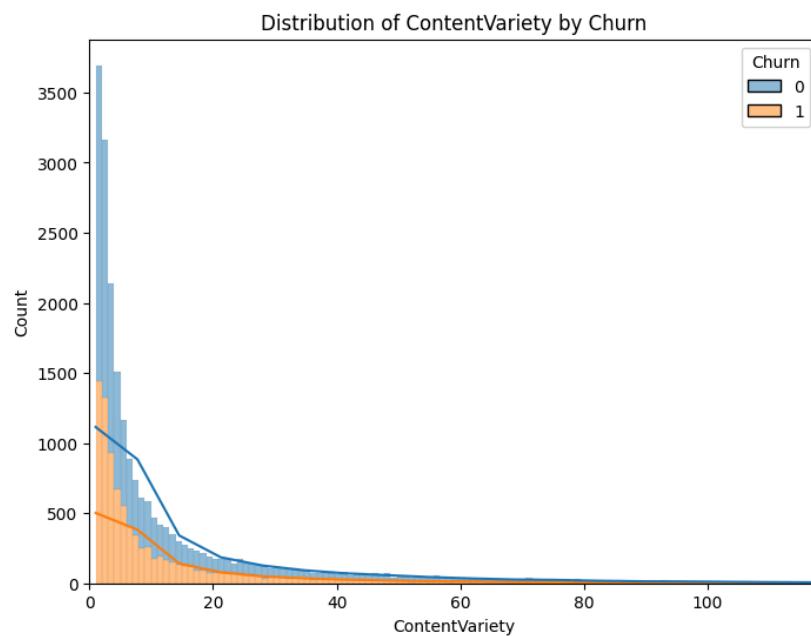


Figure 17: Histogram plot of churn distribution by ContentVariety

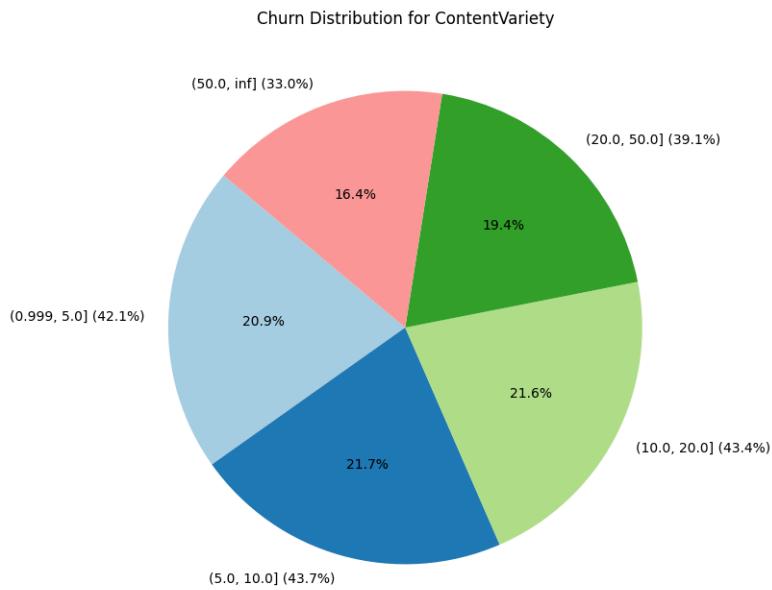


Figure 18: Pie plot of churn distribution by ContentVariety

Also the number of logins to the system was a clear indicator of customer's possible loyalty. More logins likely reflects the customer's satisfaction with the system. As it is visible on the figures 19, 20 and 21 below, if the customer logs in to the system more frequently, he/she is less likely to churn.

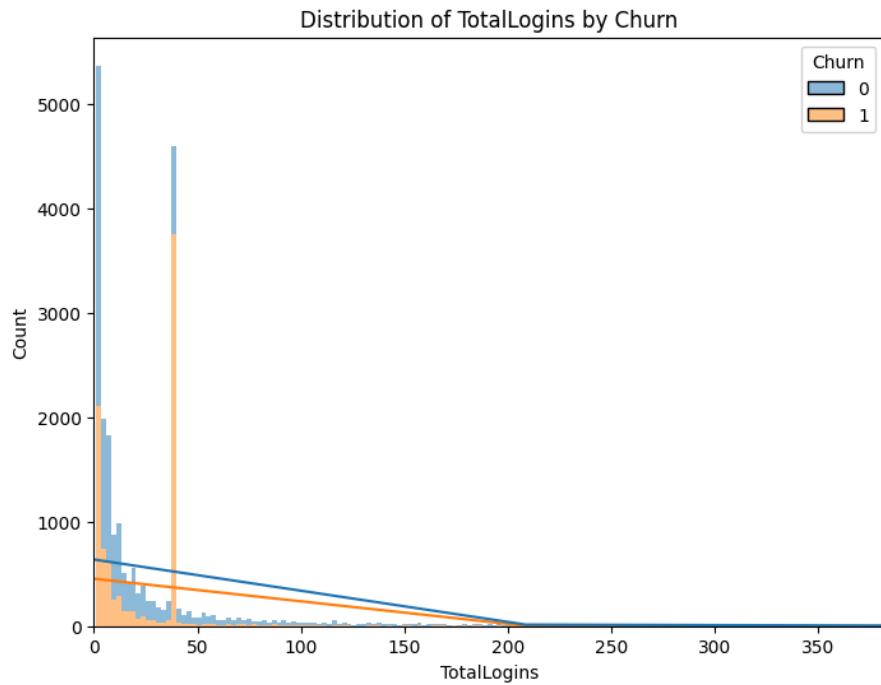


Figure 19: Histogram plot of churn distribution by TotalLogins

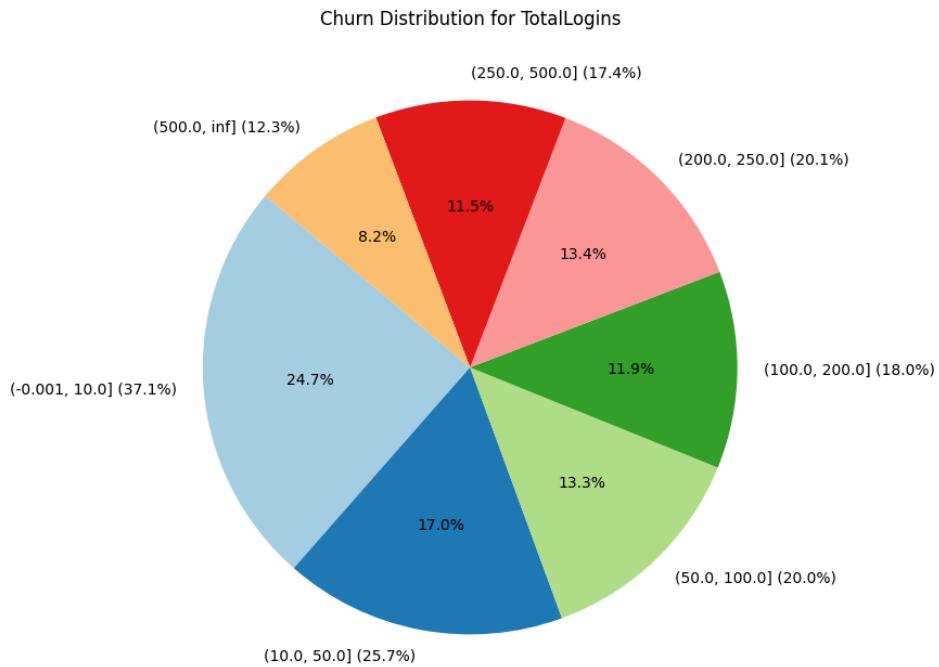


Figure 20: Pie plot of churn distribution by TotalLogins

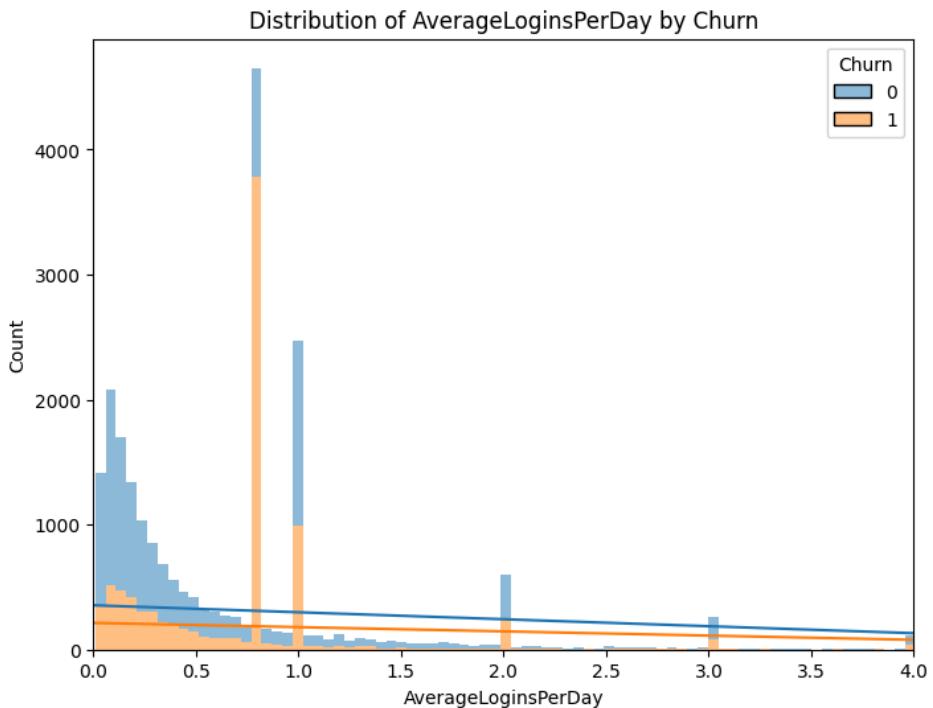


Figure 21: Histogram plot of churn distribution by AverageLoginsPerDay

The correlation between the number of unique devices and churn was as expected. Users who access the system from a greater variety of devices are likely more satisfied with the service and wish to access it more frequently, making them less likely to churn. On the below Figure

22 and Figure 23, we can observe the drop-off on the churn ratios while the number of devices increases.

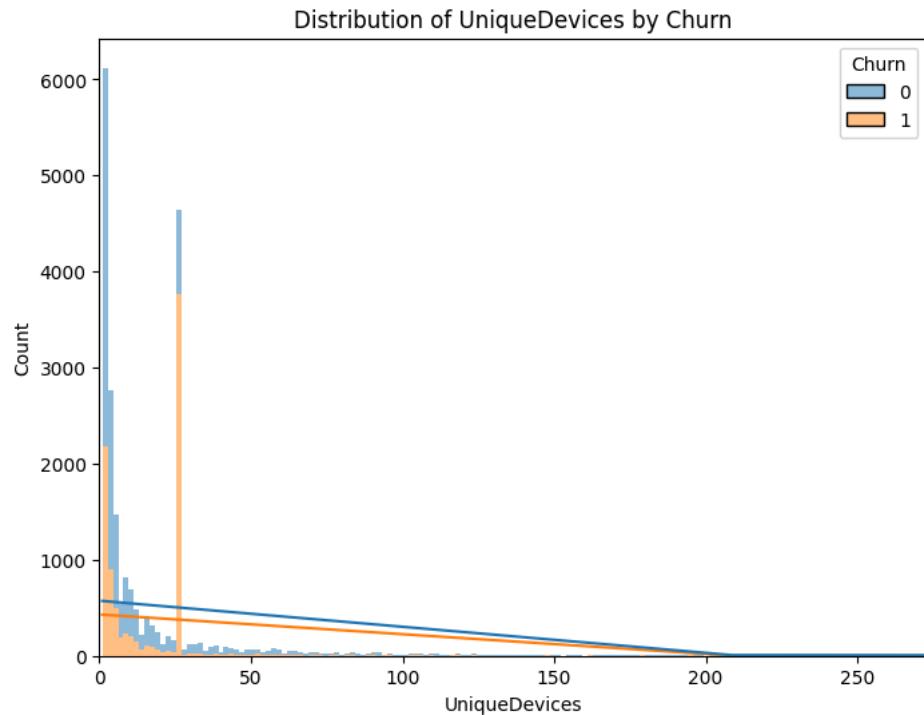


Figure 22: Histogram plot of churn distribution by UniqueDevices

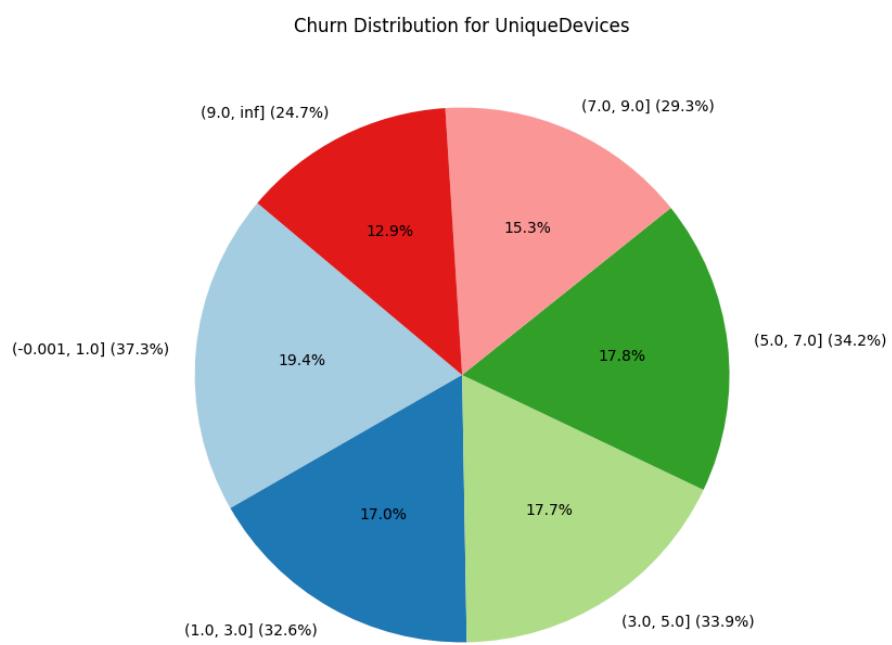


Figure 23: Pie plot of churn distribution by UniqueDevices

About the watch time behaviors of the users, we can observe correlations between the churn and the customer's complete ratio of the content in one watch and average usage duration. Users who churn tend to have lower content completion ratios and shorter usage durations compared to those who stay. Non-churned users show a longer distribution tail toward higher values in both features, suggesting greater engagement and satisfaction with the service. We can observe a simple skewed normal distribution on Figure 24 and Figure 25, but after one point (roughly 25 for both graphs) the churn is almost linear, as we suggested in our interpretation.

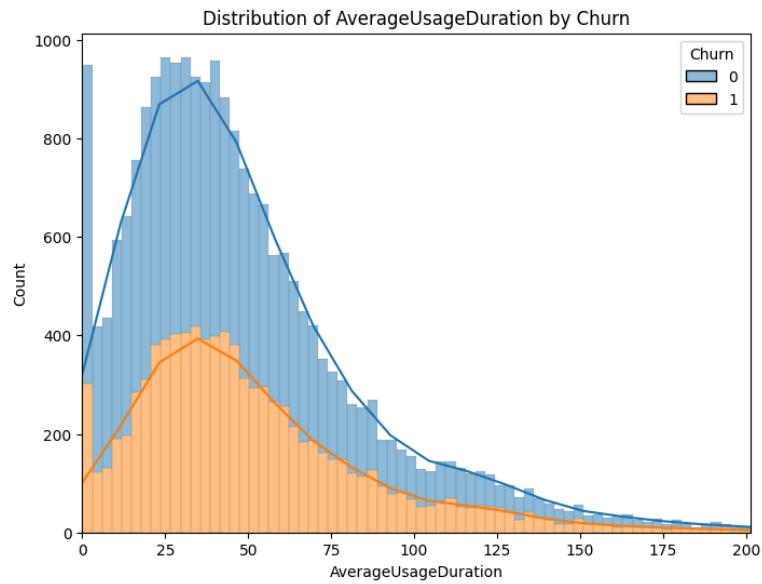


Figure 24: Histogram plot of churn distribution by AverageUsageDuration

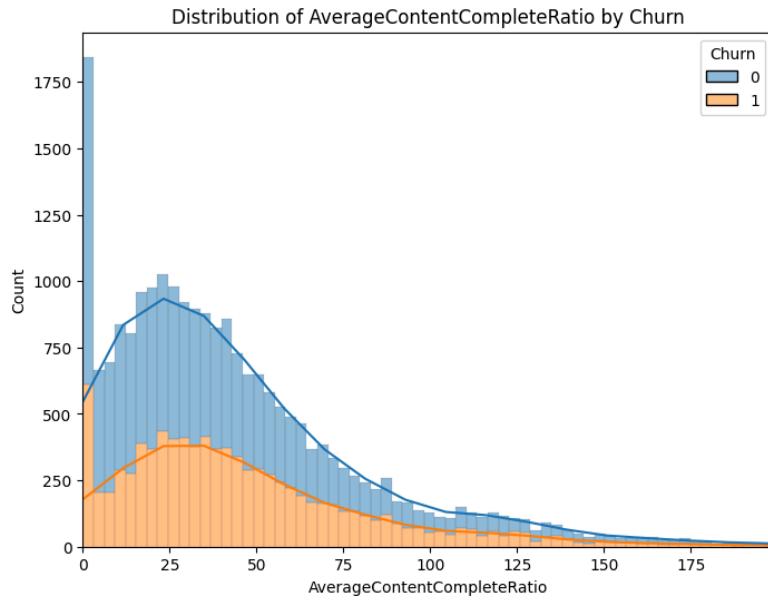


Figure 25: Histogram plot of churn distribution by AverageContentCompleteRatio

We can see how successful our features from the feature engineering phase in explaining the relationship between them and the churn label to the model thanks to the RandomForest's feature importance plot, which is provided below. It is clear that churn determination highly relies on the total usage period feature. Other than that, it can be observed that almost all of the other features were equally weighted. Also we can consider the app variety and numbers of user profiles features did not explain the churn as much as the others. The weighting of all features of our dataset can be observed on Figure 26.

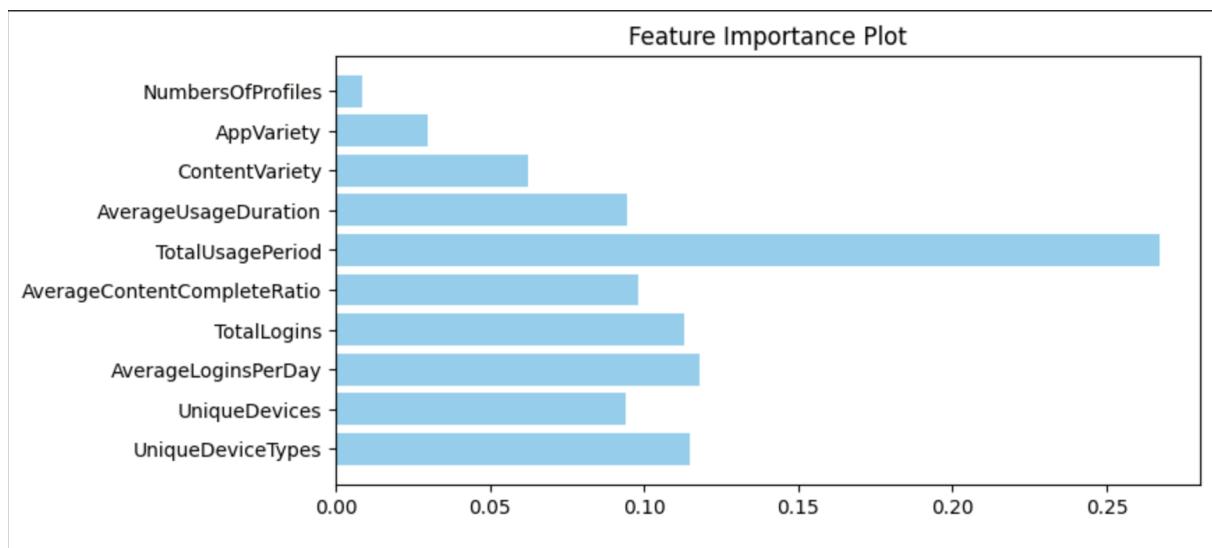


Figure 26: Feature Importance Plot of RandomForest Model

As we discussed previously, we used several machine learning algorithms in order to predict churn. The train-test accuracy results of the trained models can be seen on the following chart. As Figure 27 shows, the best performing models were tree-based Random Forests and Decision Trees. These two model's complexity was perfect for fitting our dataset, but very high train scores can be worrying and it will be handled in order to avoid over-fit situations via hyperparameter-fine tuning, adding new features and using more complex models.

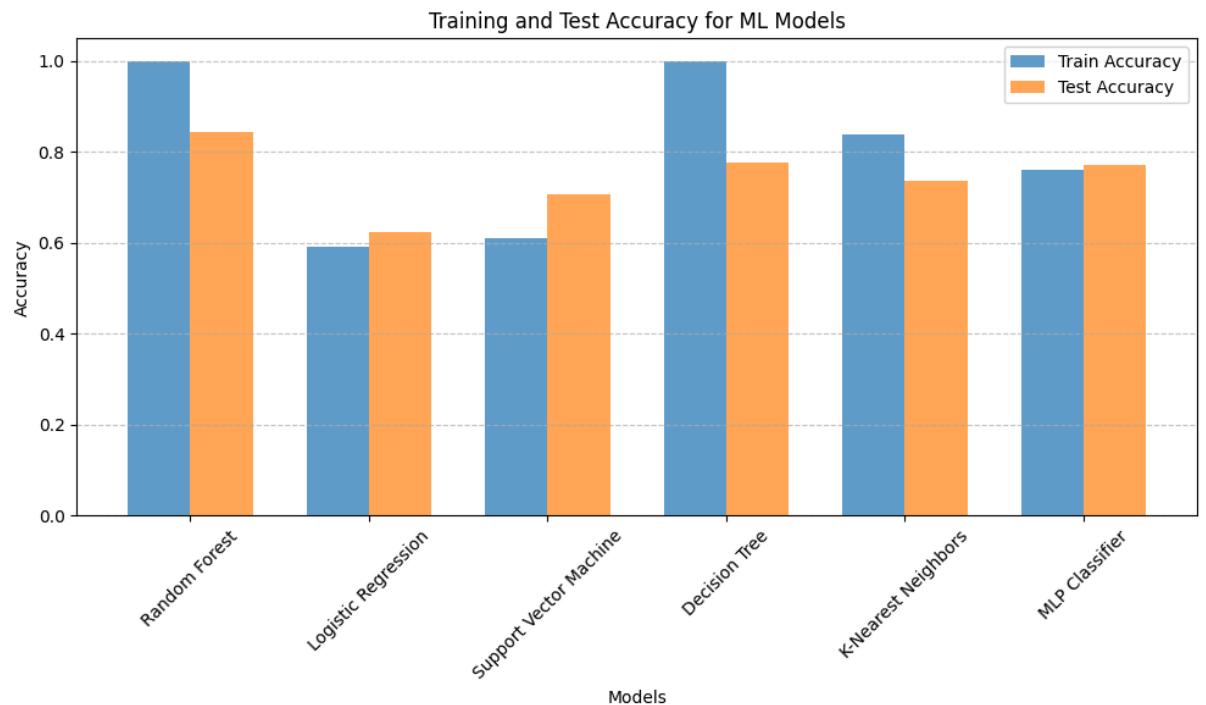


Figure 27: Train-Test Accuracy Comparison of ML Models

We can also use the confusion matrix plots in order to observe and compare our model's performance. Figure 28 also confirms that the best performing models were indeed Random Forests and Decision Trees since they have the highest True Positive and True Negative values with least False Negative and False Positive values.

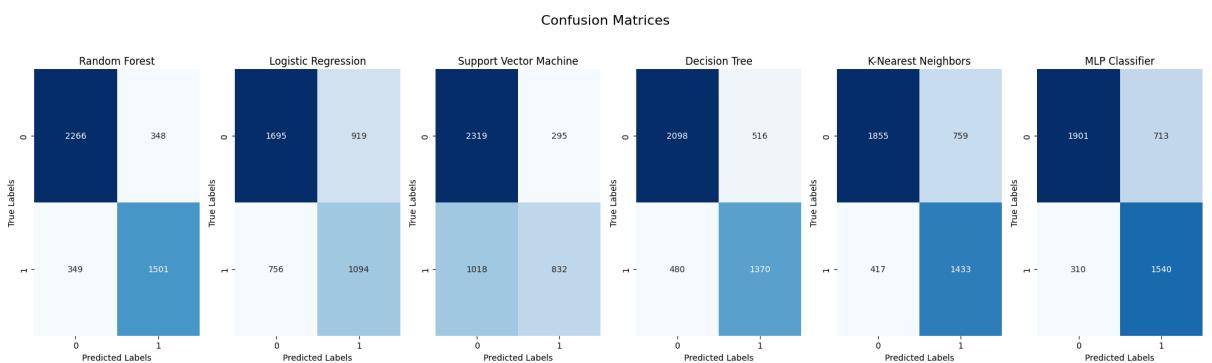


Figure 28: Confusion Matrices Comparison of ML Models

Finally, we can see the classification report of our best performing model on Figure 29. The model's accuracy (84.38%) indicates a promising performance in churn prediction. For class 0 (non-churners),

- Precision: 87% – 87% of the users predicted as non-churners are actually non-churners.
- Recall: 87% – 87% of the actual non-churners are correctly identified.
- F1-score: 87% – A strong balance between precision and recall.

For Class 1 (churners):

- Precision: 81% – 81% of the users predicted as churners are actual churners.
- Recall: 81% – 81% of the actual churners are correctly identified.
- F1-score: 81% – A satisfactory balance between precision and recall.

Macro average and weighted average metrics indicate balanced performance across the two classes. These results show that the model effectively predicts churners and non-churners with similar levels of accuracy for both classes. Even though, further improvements could focus on enhancing performance for the chunner class.

--- Random Forest ---				
Accuracy: 0.8438620071684588				
	precision	recall	f1-score	support
0	0.87	0.87	0.87	2614
1	0.81	0.81	0.81	1850
accuracy			0.84	4464
macro avg	0.84	0.84	0.84	4464
weighted avg	0.84	0.84	0.84	4464

Figure 29: Classification Report of the Random Forest Model

6.2. Recommendation Engine Results

For our recommendation engine, the dataset includes several columns, as shown in Figure 30, which we used to develop our model.

```

Dataset: ratings.csv
['userId', 'movieId', 'rating', 'timestamp']
-----
Dataset: tags.csv
['userId', 'movieId', 'tag', 'timestamp']
-----
Dataset: movies.csv
['movieId', 'title', 'genres']
-----
Dataset: links.csv
['movieId', 'imdbId', 'tmdbId']
-----
Dataset: ubcfv2.csv
['Movie ID', 'Movie Name', 'Description', 'Genres', 'Made Year', 'Audio Language']
-----
Dataset: dsmartFavorites_report.csv
['userid', 'profileid', 'contentid', 'insertdate', 'name']
-----
Dataset: dsmart_liked_report.csv
['userid', 'profileid', 'contentid', 'insertdate', 'name']
-----
Dataset: ibcf.csv
['Movie ID', 'Movie Name', 'Description', 'Genres', 'Made Year', 'Audio Language']
-----
Dataset: ubcfavorites1.csv
['Movie ID', 'Movie Name', 'Description', 'Genres', 'Made Year', 'Audio Language']
-----
```

Figure 30: Overview of Datasets Used for the Recommendation Engine.

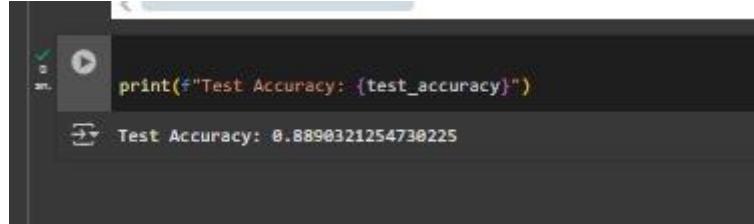
After merging the data appropriately, various models were tested within the recommendation system, and their outputs were compared across specific metrics. The Figure 31 below provides examples of key measures, such as Recommendation Score, Precision, and RMSE, which demonstrate the variations in performance amongst models like Hybrid, Hybrid NN, IBCF, SVD, and Standard.

Model	Recommendation Score	Precision	RMSE
Hybrid	4.697333333	0.9739746667	0.1428146667
Hybrid NN	4.708	0.9938	0.0584
IBCF	4.68	0.932145	0.342635
SVD	4.76	0.9206	0.393855
Standard	4.681632653	0.9942102041	0.2599510204

Figure 31: Comparison of Models based on RS, Precision and RMSE

The recommendation engine employs a deep neural network model with user and movie embeddings, in conjunction with a softmax activation layer, to predict personalized movie ratings. User identities, movie identifiers, and genres were among the user-movie interaction

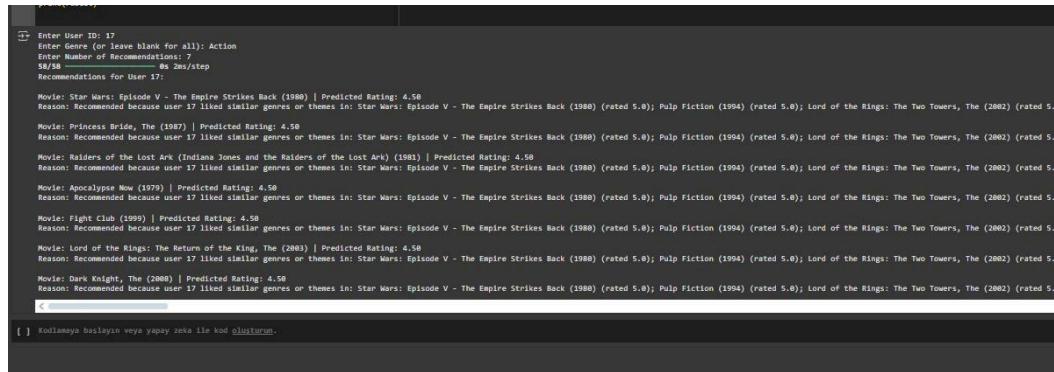
data used to train the model. With an 88.9% test accuracy represented in Figure 32, it showed good generalization across user-movie pairs that were not visible.



```
print("Test Accuracy: " + str(test_accuracy))
Test Accuracy: 0.8890321254730225
```

Figure 32: Test Accuracy for DNN-Softmax Recommendation Engine

The figure 33 given below is an example given for User 17 that demonstrates that the model's suggestions are accurate and understandable. In this instance, the model recommended films like "The Dark Knight" and "The Empire Strikes Back" because of their thematic and general resemblance to highly regarded movies. By matching the recommendations with the user's particular interests, the engine effectively delivers material that is tailored to the user, increasing user engagement. This is made possible by the combination of predictive accuracy and easily understandable reasons.



```
Enter User ID: 17
Enter Genre (or leave blank for all): Action
Enter Number of Recommendations: 7
Recommendations for User: 17

Movie: Star Wars: Episode V - The Empire Strikes Back (1980) | Predicted Rating: 4.50
Reason: Recommended because user 17 liked similar genres or themes in: Star Wars: Episode V - The Empire Strikes Back (1980) (rated 5.0); Pulp Fiction (1994) (rated 5.0); Lord of the Rings: The Two Towers, The (2002) (rated 5.0)

Movie: Princess Bride, The (1987) | Predicted Rating: 4.50
Reason: Recommended because user 17 liked similar genres or themes in: Star Wars: Episode V - The Empire Strikes Back (1980) (rated 5.0); Pulp Fiction (1994) (rated 5.0); Lord of the Rings: The Two Towers, The (2002) (rated 5.0)

Movie: Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981) | Predicted Rating: 4.50
Reason: Recommended because user 17 liked similar genres or themes in: Star Wars: Episode V - The Empire Strikes Back (1980) (rated 5.0); Pulp Fiction (1994) (rated 5.0); Lord of the Rings: The Two Towers, The (2002) (rated 5.0)

Movie: Apocalypse Now (1979) | Predicted Rating: 4.50
Reason: Recommended because user 17 liked similar genres or themes in: Star Wars: Episode V - The Empire Strikes Back (1980) (rated 5.0); Pulp Fiction (1994) (rated 5.0); Lord of the Rings: The Two Towers, The (2002) (rated 5.0)

Movie: Fight Club (1999) | Predicted Rating: 4.50
Reason: Recommended because user 17 liked similar genres or themes in: Star Wars: Episode V - The Empire Strikes Back (1980) (rated 5.0); Pulp Fiction (1994) (rated 5.0); Lord of the Rings: The Two Towers, The (2002) (rated 5.0)

Movie: Lord of the Rings: The Return of the King, The (2003) | Predicted Rating: 4.50
Reason: Recommended because user 17 liked similar genres or themes in: Star Wars: Episode V - The Empire Strikes Back (1980) (rated 5.0); Pulp Fiction (1994) (rated 5.0); Lord of the Rings: The Two Towers, The (2002) (rated 5.0)

Movie: Dark Knight, The (2008) | Predicted Rating: 4.50
Reason: Recommended because user 17 liked similar genres or themes in: Star Wars: Episode V - The Empire Strikes Back (1980) (rated 5.0); Pulp Fiction (1994) (rated 5.0); Lord of the Rings: The Two Towers, The (2002) (rated 5.0)
```

Figure 33: Example output for User 17

By examining the characteristics of programs, such as genres, categories, tags, authors, and directors, the content-based recommendation engine utilizes cosine similarity to offer customized recommendations. This approach effectively establishes a connection between user preferences and the relevant database content. For instance, recommendations such as "Vesikalı Yarım" and "Çılgınlar" were generated with high similarity scores (e.g., 0.826 and 0.809, respectively) when "Selvi Boylum Al Yazmalım" was selected. The thematic content of

the chosen show, characterized by its romantic and dramatic elements, aligns with these suggestions, thereby substantiating the efficacy of the method.

Furthermore, Google Generative AI's AI assistance has been shown to increase the system's accuracy by refining similarity scoring. A comparison of recommendations with and without AI assistance reveals that the former exhibits more individualized and sophisticated matching. The system's capacity to contextualize programs based on comprehensive metadata ensures a customized and engaging experience for users. These results are illustrated in Figures 34, 35, and 36.

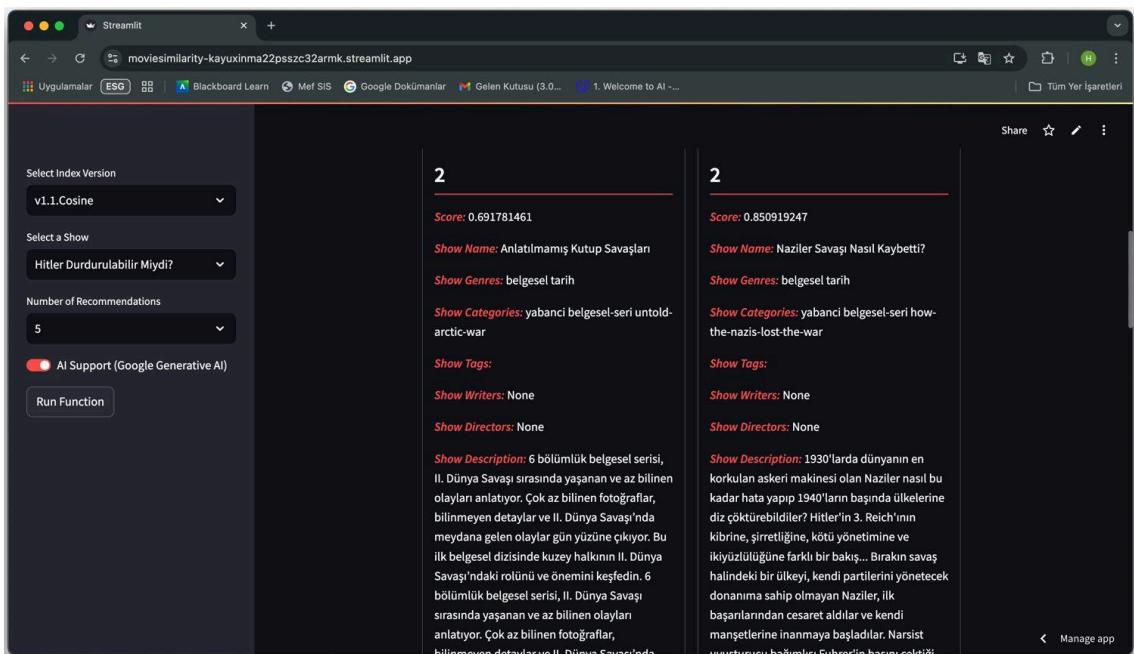


Figure 34: The top content-based recommendations for the selected show ranked by similarity scores.

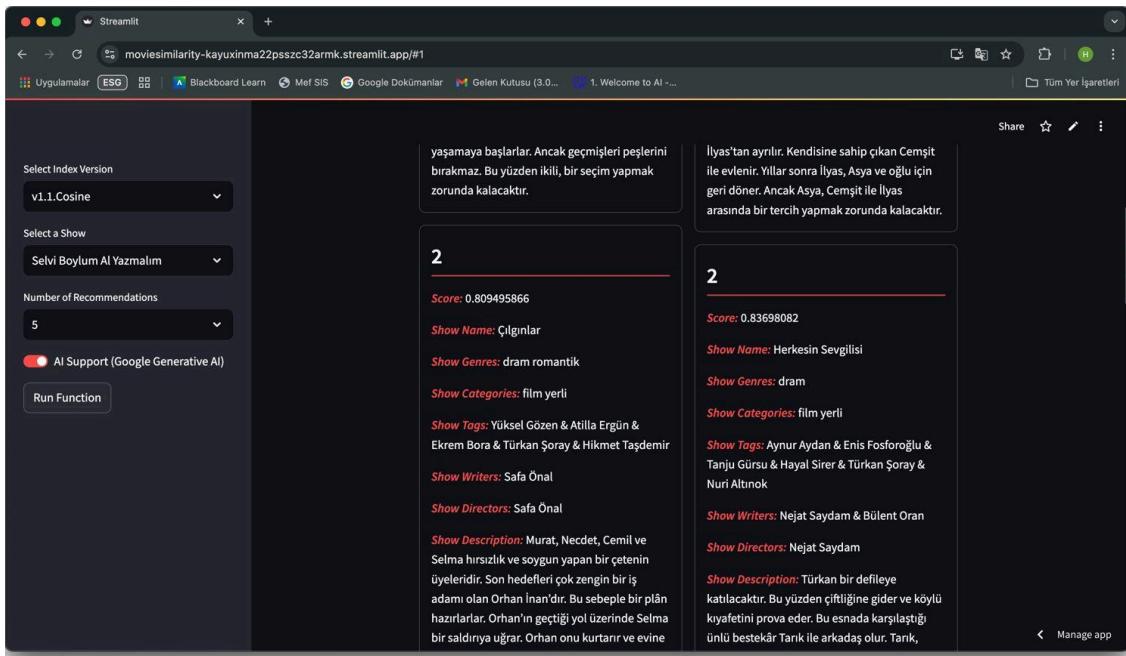


Figure 35: The top content-based recommendations for the selected show "Selvi Boylum Al Yazmalim" ranked by similarity scores.

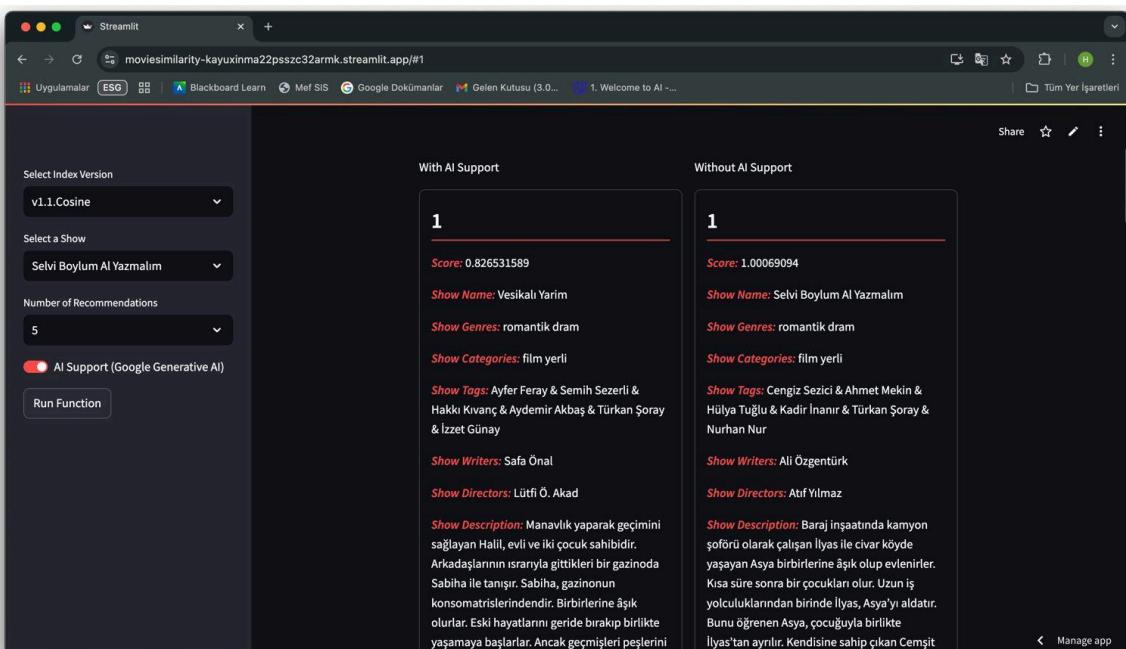


Figure 36: The top recommendations for "Selvi Boylum Al Yazmalim" with and without AI support.

6.3. Movie Recommendation Chatbot Results

Performance of the chatbot is evaluated by human feedback. Several scenarios have been tried. The retrieved documents and the response of the chatbot have been evaluated. Here are some example user prompts and chatbot responses:

Example 1: Query by Director

Query: "Christopher Nolan'in yönettiği filmlerden önerir misin?"

Response: (Figure 38)

Retrieved Documents: (Figure 37)



Figure 37: Retrieved documents for Example 1

As you can see in Figure 37, the movie documents that contain Christopher Nolan as director are retrieved. In addition, our chatbot provided relevant information about the movies.

Chatbot Response: (Figure 38)

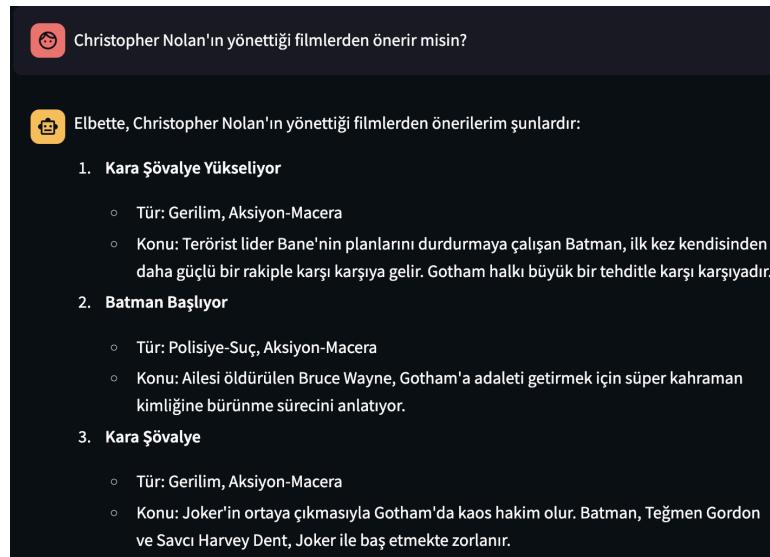


Figure 38: Chatbot Responses for Example 1

We can also request more information about a specific movie that our chatbot recommended. Here is an example:

Query: "Kara Şövalye filmini daha detaylı şekilde yorumlar misin? Konu, oyuncu performansı, eleştiriler vs."

The chatbot response for this query is in the section of Appendix A represented in Figure 41.

Example 2: Query by Genre and Cast

Query: "Tom Cruise'un oynadığı aksiyon filmlerinden önerir misin?"

Retrieved Documents: (Figure 39)

```

İsim: Mission: Impossible – Yansımalar
Tür: aksiyon-macera
Konu: Ethan Hunt, birlikte çalıştığı IMF ekibi ve birkaç dostuyla beraber sıkıntılı bir sürecin
Oyuncular: Ving Rhames & Henry Cavill & Tom Cruise
Yönetmenler: Christopher McQuarrie

İsim: Top Gun: Maverick
Tür: dram aksiyon
Konu: Top Gun: Maverick, pilot Pete "Maverick" Mitchell'in hava kuvvetlerine geri dönüşü sonrası
Eleştiri: 26 Temmuz 2019 olarak duyurulan vizyon tarihi pandemi nedeniyle 3 yıl ertelenen Top G
Oyuncular: Tom Cruise & Miles Teller & Jennifer Connolly
Yönetmenler: Joseph Kosinski

İsim: Gerçeğe Çağrı
Tür: aksiyon-macera
Konu: Colin Farrell ile gizem ve macera yüklü bir aksiyon... 21. yüzyıl sonları.. Dünya ikiye b
Oyuncular: Bryan Cranston & Bokeem Woodbine & Colin Farrell
Yönetmenler: Len Wiseman

```

Figure 39: Retrieved Documents for the Example 2

Chatbot Response: (Figure 40)

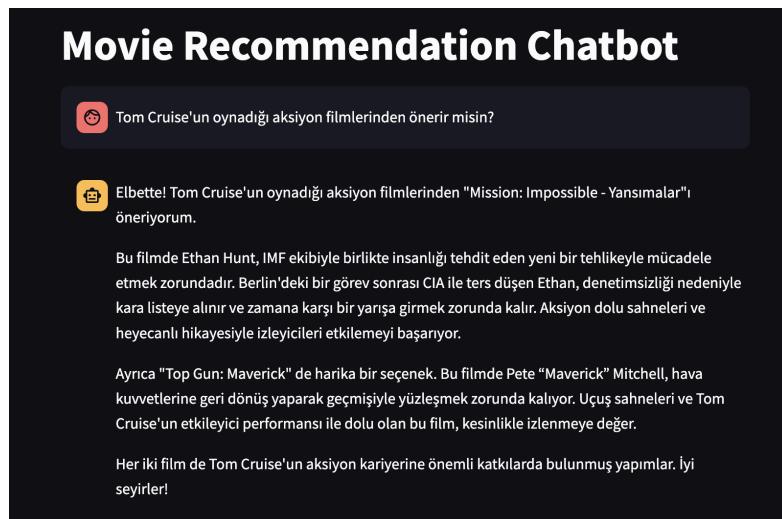


Figure 40: Chatbot Response for Example 2

As you can see in Figure 40, the action movie documents that Tom Cruise is in the cast are retrieved. In addition, our chatbot provided relevant information about those movies.

7. CONCLUSION

In the phase of the project we worked on developing a machine learning model to predict customer churn, our aim was to address the issue of customer retention for subscription-based services. The solution phase involved preparing a strong dataset, exploring and engineering features and their relations, and applying multiple machine learning algorithms to identify the best-performing model. Lastly, improving the performance of the selected model with several techniques such as hyperparameter fine-tuning. The experience provided a deep understanding of the lifecycle of a predictive modeling project, from data preprocessing to model evaluation and improvement.

During the recommendation engine stage of the project, the development of the engine to improve user-specific content delivery was important. To identify the most suitable model, a methodology was employed that relied on the creation of an extensive dataset, the investigation and design of pertinent characteristics, and the implementation of a range of recommendation strategies, including content-based approaches and collaborative filtering. Finally, methods such as embedding refinement and hyperparameter optimization were employed to improve the performance of the selected model. This process, from data preparation to model assessment and refinement, generated valuable insights into the entire recommendation system development cycle.

Retrieval-Augmented Generation (RAG) was used in this project's design to create a movie recommendation chatbot that would provide contextually relevant, personalized suggestions based on reviews and movie metadata. The system used OpenAI's GPT models for natural language production in conjunction with a Pinecone vector database and a retrieval module. Web scraping, vector-based retrieval, and synthetic data synthesis were important techniques that demonstrated RAG's potential for clever and user-focused applications.

7.1. Life-Long Learning

In order to complete this project, we acquired new knowledge and skills in several areas. We increased our data visualization skills using programs like Matplotlib and Seaborn, and we strengthened our feature engineering skills by developing useful features like AverageUsageDuration and TotalUsagePeriod. Effective comprehension and analysis of data

required these abilities. In order to maximize model performance, we also gained a deeper grasp of hyperparameter tuning techniques like grid search and random search, as well as machine learning algorithms like Random Forests, SVM, and Logistic Regression.

We studied complex natural language processing (NLP) methods and used Retrieval-Augmented Generation (RAG) for the chatbot component. This required experimenting with tools like OpenAI's GPT-based models and learning from technical documentation and web resources. Self-learning presented a variety of difficulties, with subjects like RAG and hyperparameter tweaking requiring a great deal of work and repetition. We handled this by allocating tasks, using internet resources such as Stack Overflow and Coursera, and exchanging ideas during team meetings.

7.2. Professional and Ethical Responsibilities of Engineers

We followed MEF University's academic rules and the IEEE Code of Ethics' professional and ethical guidelines when designing and developing this project. We placed a high value on availability, equity, and honesty, making sure that user privacy and data security were given top priority in all decisions in accordance with laws like the GDPR.[16]

As a team, we also signed two separate privacy agreements with Demirören Media (D-Smart GO Platform), further underscoring our commitment to safeguarding sensitive user information. We minimized biases in data and algorithms, maintained thorough documentation, and held people accountable for the models' correctness. The design of our system was heavily influenced by ethical concerns, especially when it came to promoting inclusive and accessible technology and protecting sensitive user data through encryption. These obligations helped to create a system that is consistent with the more general ethical precepts of security, trust, and the good of society.

7.3. Contemporary Issues

On the churn prediction side of the project, we have leveraged contemporary tools such as Python's Scikit-learn and Pandas libraries for machine learning and data manipulation. Emerging technologies such as big data analytics and artificial intelligence played a pivotal role in shaping the approach to customer retention analysis.

7.4. Team Work

Three people made up our team: Berkay Arat, who led the creation of chatbots; Hamza Gözükara, who focused on churn prediction; and Beyzanur Yıldız, who specialized in recommendation systems. In order to confirm that our work complied with industry standards, we worked together with project managers from Demirören Media (D-Smart GO Platform). We collaborated on every aspect of the project as a team, including documentation, testing, and system integration. Using our collective knowledge and abilities, tasks were divided equally among participants. Frequent meetings promoted an effective and productive team environment that included efficient communication, equitable job sharing, and steady development.

APPENDIX A

The project codes have been compiled into a zip file and will be shared through the MEF BlackBoard system. While the datasets cannot be provided due to privacy agreements, the zip file contains detailed information and code implementation for the system.

 Kara Şövalye filmini daha detaylı şekilde yorumlar misin? Konu, oyuncu performansı, eleştiriler vs.

 "Kara Şövalye" (The Dark Knight), Christopher Nolan'ın 2008 yılında yönettiği ve süper kahraman türünün en önemli yapıtlarından biri olarak kabul edilen bir filmdir. İşte filmin detaylı analizi:

Konu

"Kara Şövalye", Gotham City'de Joker'in (Heath Ledger) ortaya çıkmasyla başlayan bir kaos döneminin anlatır. Joker, Gotham'ın suç dünyasını alt üst ederek Batman'ın (Christian Bale) ve polisin gücünü sorgulamasına neden olur. Teğmen Jim Gordon (Gary Oldman) ve Savcı Harvey Dent (Aaron Eckhart) ile birlikte çalışarak Joker'e karşı koymaya çalışan Batman, ahlaki ve etik değerlerini sorgulamak zorunda kalır. Joker'in planları, sadece fiziksel bir mücadele değil, aynı zamanda psikolojik bir savaş da beraberinde getirir. Film, suç, adalet, kaos ve düzen temalarını derinlemesine işler.

Oyuncu Performansları

- **Heath Ledger (Joker):** Ledger'in performansı, film tarihinin en ikonik ve unutulmaz karakterlerinden birine hayat vermesiyle dikkat çeker. Joker'i canlandırdıktan gösterdiği çığırılık, karanlık ve karmaşası, karakterin derinliğini artırmıştır. Ledger, rolde o kadar başarılıydı ki, bu performansıyla En İyi Yardımcı Erkek Oyuncu dalında Oscar ödülünü kazandı.

- **Christian Bale (Batman/Bruce Wayne):** Bale, Bruce Wayne karakterinin karmaşıklığını ve içsel çatışmalarını başarıyla yansıtmıştır. Batman'ın karanlık ve yalnız ruh halini izleyiciye etkili bir şekilde aktarır. Bale, Bruce Wayne'in hem süper kahraman hem de bir insan olarak yaşadığı ikilemi ustalıkla sergiler.
- **Aaron Eckhart (Harvey Dent):** Eckhart, Dent karakterinin dönüşümünü mükemmel bir şekilde yansıtır. Başlangıçta adaletin simgesi olan Dent, Joker'in oyunları sonucunda nasıl karanlık bir yola sürüklendiğini gösterir. Bu dönüşüm, filmin en çarpıcı noktalarından biridir.
- **Gary Oldman (Jim Gordon):** Oldman, Gordon karakterini güçlü ve kararlı bir lider olarak çizer. Gotham'ın adaletini sağlamak için Batman ile iş birliği yaparken, ahlaki değerleri de sorgular.

Eleştiriler

"Kara Şövalye", eleştirmenler ve izleyiciler tarafından büyük beğeni topladı. Filmin güçlü yönleri arasında:

- **Senaryo:** Jonathan ve Christopher Nolan tarafından yazılan senaryo, karmaşık ve çok katmanlı bir yapıya sahiptir. Joker'in eylemleri, Gotham'daki mevcut düzenin sorgulanmasına neden olur. Filmin sona ermesiyle birlikte izleyiciye bıraktığı sorular, tartışmalara yol açar.
- **Görsel ve Teknik Yön:** Filmin görsel estetiği ve aksiyon sahneleri, IMAX formatında çekilmesi sayesinde büyük bir etki yaratır. Özellikle Joker'in yaptığı korkutucu eylemler, izleyiciyi

- **Senaryo:** Jonathan ve Christopher Nolan tarafından yazılan senaryo, karmaşık ve çok katmanlı bir yapıya sahiptir. Joker'in eylemleri, Gotham'daki mevcut düzenin sorgulanmasına neden olur. Filmin sona ermesiyle birlikte izleyiciye bıraktığı sorular, tartışmalara yol açar.
- **Görsel ve Teknik Yön:** Filmin görsel estetiği ve aksiyon sahneleri, IMAX formatında çekilmesi sayesinde büyük bir etki yaratır. Özellikle Joker'in yaptığı korkutucu eylemler, izleyiciyi derinden etkiler.
- **Müzik:** Hans Zimmer ve James Newton Howard'ın müzikleri, filmin atmosferini güçlendirir. Müzik, gerilim anlarında izleyiciyi daha fazla içine çeker.
- **Temalar:** Film, adalet, kaos, ahlak ve insan doğasının karanlık yönleri gibi derin temaları işler. Joker'in "bir insanın, zor durumda ne kadar çürüyebileceğini" gösteren felsefesi, izleyiciyi düşündürür.

Sonuç

"Kara Şövalye", sadece bir süper kahraman filmi olmanın ötesine geçer. Karanlık temaları, derin karakter analizleri ve unutulmaz performanslarıyla izleyiciyi derinden etkileyen bir yapım olarak öne çıkar. Bu film, modern sinemada süper kahraman türünün nasıl ciddi ve derin bir anlatıma dönüşebileceğini en iyi örneklerinden biridir. Sinema tarihindeki yeri ise, hem eleştirmenler hem de izleyiciler tarafından hala tartışılmaktadır.

Figure 41: Response of the Query ‘Kara Şövalye filmini daha detaylı şekilde yorumlar misin? Konu, oyuncu performansı, eleştiriler vs.’’

REFERENCES

- [1] J. Chen, X. Wang, S. Zhao, F. Qian, and Y. Zhang, "Deep attention user-based collaborative filtering for recommendation," *Neurocomputing*, vol. 383, pp. 57–68, 2020. [Online]. Available: <https://doi.org/10.1016/j.neucom.2019.09.050>.
- [2] M. Majka, "Understanding churn rate," <https://www.researchgate.net/>, Jul. 2024. https://www.researchgate.net/publication/382592085_Understanding_Churn_Rate
- [3] Xiaojun Wu and Sufang Meng, "E-commerce customer churn prediction based on improved SMOTE and AdaBoost," *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, Kunming, 2016, pp. 1-5, doi: 10.1109/ICSSSM.2016.7538581.
- [4] I. Jahan and T. Farah Sanam, "An Improved Machine Learning Based Customer Churn Prediction for Insight and Recommendation in E-commerce," *2022 25th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar, Bangladesh, 2022, pp. 1-6, doi: 10.1109/ICCIT57492.2022.10054771.
- [5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, 2001, pp. 285–295. doi: 10.1145/371920.372071.
- [6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 173–182. doi: 10.1145/3038912.3052569.
- [7] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002. doi: 10.1023/A:1021240730564.
- [8]
- P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv.org*, 2020. Available: <https://arxiv.org/abs/2005.11401>. [Accessed: Jan. 10, 2025]

[9] Llama Team and AI at Meta, “The Llama 3 Herd of Models.” Available: https://scontent.fist10-1.fna.fbcdn.net/v/t39.2365-6/463020162_522238820565582_8192401983671993921_n.pdf?_nc_cat=108&ccb=1-7&_nc_sid=3c67a6&_nc_ohc=5ek67KPfK6gQ7kNvgFv6PIK&_nc_zt=14&_nc_ht=scontent.fist10-1.fna&_nc_gid=AvupuY3nOUS3gXYxhyc99kD&oh=00_AYD0xGQ4gGm6gD0SUe3FVXZ6shXZnQH0xM9-28Mni-CuHA&oe=67498599. [Accessed: Nov. 25, 2024].

[10] Detection of the chronic kidney disease using XGBoost classifier and explaining the influence of the attributes on the model using SHAP - Scientific Figure on ResearchGate. Available from:

https://www.researchgate.net/figure/Confusion-matrix-and-performance-metrics-formula_fig3_370070277 [accessed 24 Nov 2024]

[11] Varun, “Cosine similarity: How does it measure the similarity, Maths behind and usage in Python,” *Medium*, Sep. 27, 2020. Available:

<https://towardsdatascience.com/cosine-similarity-how-does-it-measure-the-similarity-maths-behind-and-usage-in-python-50ad30aad7db> . [Accessed: Jan. 10, 2025]

[12]

RAG, “RAG from scratch: Retrieval,” *Google Docs*, 2020. Available:

https://docs.google.com/presentation/d/124I8jlBRCbb0LAUhdmDwbn4nREqxSxZU1RF_eTGXUGc/edit#slide=id.g267060cc54f_0_69 . [Accessed: Jan. 10, 2025]

[13] RAG, “RAG from scratch: Generation,” *Google Docs*, 2020. Available:

https://docs.google.com/presentation/d/1eRJwzbSv71e9Ou9yeqziZrz1UagwX8B1kL4TbL5Gc/edit#slide=id.g2b46f2cb556_0_101 . [Accessed: Jan. 10, 2025]

[14] Google Developers, "Training a deep neural network-based recommendation model," [Online]. Available:

<https://developers.google.com/machine-learning/recommendation/dnn/training?hl=tr>. [Accessed: 10-Jan-2025].

[15] PlantUML, "Open-source tool that uses simple textual descriptions to draw UML diagrams," [Online]. Available: <https://plantuml.com>. [Accessed: 10-Jan-2025].

[16] GDPR Info, "General Data Protection Regulation (GDPR) – Complete guide to GDPR compliance," [Online]. Available: <https://gdpr-info.eu>. [Accessed: 10-Jan-2025].