People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

YAHIA FARES University of Medea

Faculty of Science



Graduation Project Presented to Obtain the Master's Degree in Computer Science

Speciality: Systems Engineering & Web Technology

Theme :

Web Services Access Prediction Used for Recommender Systems Based on Association Rules Mining

---

Proposed & Supervised by:

- Mr BOUDJEBBOUR Karim

Prepared & Presented by:

- AICI Asmaa
- OTMANI Aya

Academic Year 2024-2025

Acknowledgment

First and foremost, we would like to thank ALLAH the Almighty for granting us the health, patience, and determination to carry out this work, and for guiding us throughout every stage of our studies up to this point, as we reach the end of this academic journey. We express our sincere gratitude to our professors at Dr. Yahia Fares University - Médéa, who were always available to support us and help us pursue our academic goals, despite their own busy schedules.

In particular, we would like to express our deepest thanks to our supervi- sor, Dr. BOUDJEBBOUR Karim, for his dedication, valuable guidance, and constructive criticism throughout the preparation of this research project. His continued support and encouragement have been essential to the success of our work. Furthermore, we would like to thank the members of the jury for agreeing to evaluate and review this work.

Finally, we express our heartfelt appreciation to all those who contributed, directly or indirectly, to the realization of this work. Though we cannot name everyone individually, please know that we are truly grateful to each and every one of you.

Dedication

To my dearest parents,

To my beloved sisters,

To my childhood friends, Chaima and Nada,

To Asma, my dear classmate,

For all the beautiful moments we shared. It was a pleasure working with you.

OTMANI Aya

Dedication

First of all, I thank Allah Subhanahu wa Ta'ala for His help and countless blessings. Allah who gave me the strength, willpower, and spirit to complete my studies.

I dedicate this work to my grandfather and grandmother. Their advice and prayers always gave me strength.

Thank you to my father and mother for your love, help, and all the things you did for me.

To my brothers Abdou, Abdeljalil and sisters Ikram, Rihab -your support and smiles always made me feel better.

Thank you to my uncle, his wife, and their son Abdelwadoud for your kindness and support.

To my aunts Somia, Hanan and Siham thank you for your love and care.

To my dear friend Lamis – thank you for always being there and helping me.

To Sofia, Aicha ,halima , abir – your support meant the world to me. I'm truly grateful to have friends like you by my side. Thank you for standing with me.

To my friend and classmate Aya – I'm happy we shared this journey together.

And finally, thank you to all my friends and everyone who supported me. I'm really grateful.

AICI Asmaa

Abstract

With the rapid expansion of web services available online, there is an increasing need for intelligent tools that can recommend relevant services to users without relying on personal data. A promising solution lies in web data min- ing, which enables the discovery of recurring and meaningful patterns within large-scale datasets.

This study presents a recommendation system that leverages association rule mining on historical web service usage data. Unlike traditional approaches, the proposed system does not require user profiles; instead, it infers recommendations based solely on the implicit relationships between services that are frequently accessed together.

Several algorithms were applied to extract these rules, and their performance was evaluated using standard metrics such as support, confidence, and lift. Experimental results across various datasets demonstrate the effectiveness of this approach in generating relevant service recommendations. Nonetheless, certain challenges remain-particularly in accounting for item frequency and service weighting-which will be addressed in future work.

Keywords: Association Rules, Data Mining, Recommender System, Web Service, Apriori, FP-Growth, ECLAT, AIS .

Résumé

Avec l'augmentation rapide du nombre de services web disponibles en ligne, il devient essentiel de développer des outils intelligents permettant de recommander des services adaptés aux utilisateurs, sans recourir à leurs données personnelles. Parmi les approches les plus prometteuses, l'exploration de données web permet d'identifier des motifs récurrents et pertinents à partir de grandes quantités de données.

Ce travail propose la conception d'un système de recommandation fondé sur l'analyse des règles d'association extraites de données historiques d'utilisation des services web. Le système fonctionne indépendamment de tout profil utilisateur, en s'appuyant uniquement sur les corrélations entre les services fréquemment consultés ensemble.

Différents algorithmes ont été appliqués pour l'extraction de ces règles, et leur efficacité a été évaluée à l'aide de métriques telles que le support, la confiance et le lift. Les résultats expérimentaux, obtenus à partir de jeux de données variés, ont démontré la faisabilité de cette approche pour générer des recommandations pertinentes, tout en soulignant certains aspects à améliorer, notamment la prise en compte de la fréquence d'apparition des éléments.

Mots clés : Règles d'Association, Data Mining, Système de Recommandation, Service Web , Apriori, FP-Growth, ECLAT, AIS .

ملخص

أمام النمو المتسارع في عدد خدمات الويب المتوفرة عبر الإنترنت، برزت الحاجة إلى أدوات ذكية تساعد المستخدمين في الوصول إلى الخدمات التي تتناسب مع اهتماماتهم دون الاعتماد على بياناتهم الشخصية. من بين أبرز التقنيات التي استُخدمت لتحقيق هذا الهدف نجد التنقيب في بيانات الويب، الذي يسمح باكتشاف أنماط متكررة ومفيدة داخل كمّ هائل من البيانات.

يقترح هذا العمل تصميم نظام توصية يعتمد على تحليل قواعد الارتباط المستخرجة من بيانات تاريخية حول استخدام خدمات الويب. هذا النظام لا يتطلب معرفة مسبقة بالمستخدم، بل يعتمد كليًا على العلاقات الخفية بين الخدمات التي يتم الوصول إليها بشكل مشترك.

تم تطبيق عدة خوارزميات لاستخراج هذه القواعد، مع تقييم فعاليتها باستخدام مقاييس كمية مثل الدعم والثقة والرفع. ومن خلال تجارب مختلفة على مجموعات بيانات متنوعة، أظهرت النتائج إمكانية تقديم توصيات دقيقة وفعالة، رغم أن بعض التحديات مثل اختلاف تواتر العناصر لم تُعالج بشكل معمق، مما يمثل فرصة للتحسين في الأعمال المستقبلية.

الكلمات المفتاحية : قواعد الترابط، تنقيب البيانات، أنظمة التوصية، خدمات الويب، Apriori, AIS. ECLAT, FP-Growth,

Contents

General Introduction

Chapter I: Web Services & Recommender Systems

I.1 Introduction..... 1

I.2 Web Services..... 2

I.2.1 Definition of Web Service..... 2

I.2.2 Standards used for web service..... 3

I.2.2.1 SOAP..... 4

I.2.2.2 WSDL..... 4

I.2.2.3 UDDI..... 5

I.2.3 Service Oriented Architecture..... 5

I.2.3.1 Definition..... 5

I.2.3.2 Web Services & SOA..... 6

I.3 Recommender Systems..... 7

I.3.1 History..... 7

I.3.2 Definition Of Recommender Systems..... 8

I.3.3 Recommendation Models..... 9

I.3.3.1 Content-Based Filtering..... 10

I.3.3.2 Collaborative Filtering..... 11

I.3.3.3 Hybrid System..... 13

I.3.4 Qualitative Evaluation Metrics of Recommendation Systems 15

I.3.5 Application Fields.....17
I.4 conclusion.....21

Chapter II: Association Rules 23

II.1 Introduction..... 24
II.2 Data Mining..... 25
II.2.1 Definition of Data Mining.....25
II.2.2 Importance of Data Mining..... 26
II.2.3 Applications of Data Mining.....27
II.2.4 Data Mining and KDD..... 29
II.2.5 Technique of Data Mining.....29
II.3 Association Rules..... 32
II.3.1 History of Association Rules..... 33
II.3.2 Definition of Association Rules.....34
II.3.3 Advantages and Disadvantages of Association Rule Mining 34
II.3.4 Application areas of Association Rule.....35
II.3.5 Association Rules Interestingness Measures..... 38
II.3.6 Association Rules Mining : Algorithms..... 41
II.3.6.1 AIS(Artificial Immune System).....41
II.3.6.2 Apriori algorithm..... 42
II.3.6.3 FP-Growth Algorithm (Frequent Pattern Growth algorithm)..... 44
II.3.6.4 ECLAT Algorithm.....47
II.4 Conclusion.....51

Chapter III: Proposed Solution & Implementation 52

III.1 Introduction..... 53
III.2 Related Work.....54

III.3 Problematic & Objectives.....56
III.4 Solution Overview.....57
III.5 Implementation & Tests.....58
III.5.1 Development Environment.....59
III.5.1.1 Materials.....59
III.5.1.2 Programming Language.....59
III.5.1.3 Integrated Development Environment.....60
III.5.2 Data Description.....60
III.5.3 Tests & Results.....61
III.5.4 Implementation.....75
III.5.5 Test Run :.....77
III.6 Conclusion.....81

General Conclusion

List of Figures

Figure 1: Architecture of A Web Service[3].....3
Figure 2: SOA and SOA Technologies [8].....6
Figure 3: Process of general recommendation system models.[12].....10
Figure 4: Recommendation principle of Contents-Based Filtering Model..11
Figure 5:Recommendation principle of Collaborative Filtering Model.[12]. 13
Figure 6: Recommendation principle of Hybrid recommendation model..14
Figure 7: List of recommendation fields [12].....18
Figure 8: Architecture of Data Mining [29].....26
Figure 9: Applications of Data Mining.....27
Figure 10: Classification - GeeksforGeeks.....30
Figure 11: Clustering diagram..... 31
Figure 12: Exemple of Association Rules.....33
Figure 13: Market Basket Analysis..... 36
Figure 14: Apriori algorithm example [24].....44
Figure 15: Framework of web service prediction..... 58
Figure 16: Execution time of the algorithms based on minimum support.. 61
Figure 17: Runtime of algorithms on datasets with 7 web services.....62
Figure 18: Runtime of algorithms on datasets with 15 web services.....63
Figure 19: Runtime of algorithms on datasets with 20 web services.....65
Figure 20: Runtime of algorithms on datasets with 28 web services.....66
Figure 21: Runtime with minimum support on Data with 30k..... 68

Figure 22: Runtime with minimum support on Data with 50k..... 69
Figure 23: Runtime with minimum support on Data with 75k..... 71
Figure 24: Runtime with minimum support on Data with 100k..... 72
Figure 25: Runtime of different minimum confidence values..... 74
Figure 26: Dataset Size Details..... 77
Figure 27: Frequent Itemsets Generated by the Apriori Algorithm..... 78
Figure 28: Association Rules Generation Code..... 78
Figure 29: Association Rules Extracted Using the ARM Function..... 79
Figure 30: Interactive Dashboard of Web Service Recommendations…. 80
Figure 31: Service Selection Sidebar in Recommendation System..... 80

List of Tables

Table 1: Evaluation Metrics for Recommendation Systems……16
Table 2: Sample Dataset [53]....45
Table 3: Comparison of Association Rule Mining Algorithms ……. … 47
Table 4: Tidset Representation of Items....48
Table 5: Generation of 2-Itemsets with Corresponding Tidsets....49
Table 6: Generation of 3-Itemsets with Corresponding Tidsets....49
Table 7: Generation of 4-Itemsets with Corresponding Tidsets....50
Table 8: The result....50
Table 9: Used datasets....60

Acronyms

ACO : Ant Colony Optimization

ACM : Association for Computing Machinery

AI : Artificial Intelligence

AIS : Artificial Immune System

ARM : Association Rule Mining

B2B : Business-to-Business

BERT4Rec : Bidirectional Encoder Representations from Transformers for Recom- mendation

BFS : Breadth-First Search

DFS : Depth-First Search

ECLAT : Equivalence Class Clustering and bottom-up Lattice Traversal

EDM : Educational Data Mining

FP : False Positives

FP-Growth : Frequent Pattern Growth

FN : False Negatives

GNP : Genetic Network Programming

Google Colab : Google Colaboratory

HTTP : HyperText Transfer Protocol

IBM : International Business Machines

IDE : Integrated Development Environment

IoT : Internet of Things

KDD : Knowledge Discovery in Databases

KNN : K-Nearest Neighbors

PCA : Principal Component Analysis

RSs : Recommender Systems

SMTP : Simple Mail Transfer Protocol

SNS : Social Network Services

SOA : Service-Oriented Architecture

SOAP : Simple Object Access Protocol

SVD : Singular Value Decomposition

SVM : Support Vector Machines

TF-IDF : Term Frequency-Inverse Document Frequency

TN : True Negatives

TP : True Positives

UDDI : Universal Description, Discovery, and Integration

URI : Uniform Resource Identifier

W3C : World Wide Web Consortium

WSDL : Web Services Description Language

XML : eXtensible Markup Language

General Introduction

In today's rapidly evolving digital landscape, efficiency and convenience have become crucial elements across all aspects of life. Technology continues to play a pivotal role in fulfilling these demands by enabling faster and more seamless access to information. Modern information systems now rely extensively on Internet technologies, which have revolutionized how individuals and organizations—whether academic, industrial, commercial, or governmental—communicate, collaborate, and operate.

One of the most significant innovations in this context is the development of web services, which offer a standardized means of delivering applications and data over the Internet. These services serve as essential building blocks for developing scalable, interoperable, and interactive applications that are accessible globally.

However, the vast amount of information available online can be both a strength and a challenge. Users frequently struggle to locate relevant content quickly due to the overwhelming volume of web resources. This information overload has spurred growing interest in analyzing user behavior and applying advanced data analysis techniques to enhance the efficiency of information retrieval.

Among the most impactful solutions to this problem are recommender systems. These systems leverage insights from user preferences, past interactions,

and behavioral patterns to generate personalized suggestions, thereby improving the overall user experience. In our research, we focus on a specific data mining technique known as association rule mining—a method designed to uncover hidden relationships and co-occurrence patterns among different web pages or services.

Association rule mining is widely applied to extract meaningful associations from large datasets by using metrics such as support, confidence, and lift to quantify the strength and relevance of the discovered patterns. By employing this technique in the context of web service recommendation, our goal is to identify services that are frequently accessed together through user navigation behaviors. This enables the system to suggest relevant services that align with users' inferred interests, without the need for explicit personal information.

Chapter I:

Web Services & Recommender Systems

I.1 Introduction

In recent years, various improvised approaches have been utilized in business-to-business (B2B) scenarios to capitalize on the foundational infrastructure of the Internet. However, the emergence of Web services has introduced a more structured and scalable framework for enabling interaction between applications. These services are built upon existing Web protocols and standardized eXtensible Markup Language (XML) technologies.

Web services represent a modern class of Web-based applications. They are modular, self-contained, and self-describing components that can be published, discovered, and invoked via the Web. Their functionality ranges from handling simple data requests to orchestrating complex business workflows. Once deployed, these services can be accessed and utilized by other applications or services.

Recommender systems leverage the collective input and preferences of users within a community to assist individuals in identifying content or products that align with their interests or needs. Initially known as collaborative filtering systems, they were designed to overcome limitations faced by traditional keyword-based filtering methods.

Specifically, they addressed two main issues: the first is the challenge of dealing with a large volume of relevant content where all results might match keyword queries but differ in quality by incorporating human evaluations. The second is the difficulty in filtering non-textual content, such as media, where personal preferences play a crucial role.

For instance, the Ringo system applied collaborative filtering techniques to

suggest music to users. This approach was later extended to other creative domains in both academic research and commercial applications.

I.2 Web Services

I.2.1 Definition of Web Service

Web services currently constitute the most widely recognized set of standards for Internet-based applications. This technology relies on widely adopted protocols, such as XML and HTTP (HyperText Transfer Protocol). Initially introduced by IBM and Microsoft, web services were later partially standardized by the World Wide Web Consortium (W3C).

According to the W3C, a Web service is a software system that has a Uniform Resource Identifier (URI) and XML-defined public interfaces and bindings. Other software systems can learn its definition. These systems can then use XML based messages sent via Internet protocols to communicate with the Web service in the way specified by its description.[1]

The world is rich with valuable data and services provided by computer programs, but much of it remains inaccessible. Websites are typically designed for human users, not for programs, and they often hide information within a constantly changing and complex structure of HTML.

I.2.2 Standards used for web service

Web services rely on a layered architecture structured into three core areas: communication protocols, service descriptions, and service discovery mechanisms. Several key standards are used to ensure interoperability, platform independence, and scalability of web-based interactions:

- SOAP (Simple Object Access Protocol): Enables structured message exchange between distributed systems using XML.
- WSDL (Web Services Description Language): Describes the functional and technical aspects of a web service in a machine-readable format.
- UDDI (Universal Description, Discovery, and Integration): Provides a centralized registry for publishing and locating web services.

  These standards work in tandem to support robust, dynamic, and loosely coupled service-oriented systems [2].

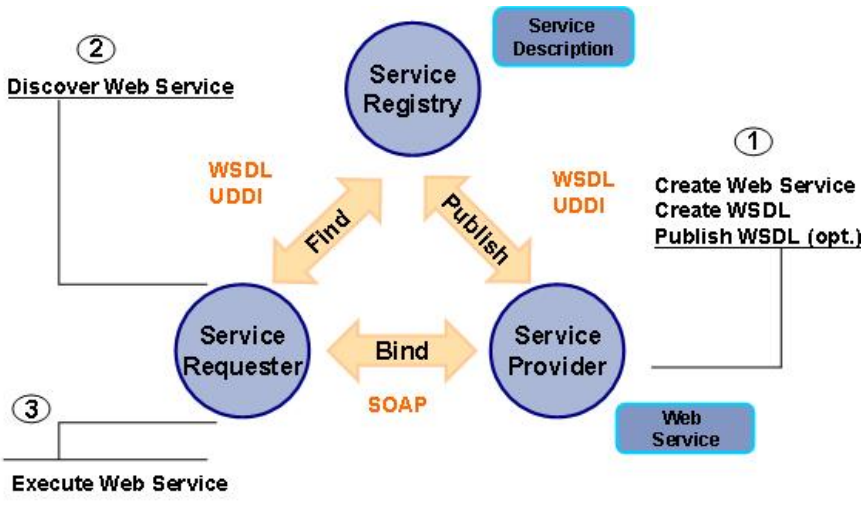


Figure 1: Architecture of A Web Service[3]

I.2.2.1 SOAP

SOAP is a W3C-defined protocol that provides a standardized format for messaging in distributed environments. It primarily uses HTTP and SMTP (Simple Mail Transfer Protocol) for message transport, with XML as its data representation language. Each SOAP message includes an envelope with a mandatory body and optional header:

SOAP messages are enclosed in an envelope consisting of a header and a body.

- Envelope: Wraps the message and defines namespaces.
- Header (optional): Carries metadata such as authentication or transaction info
- Body: Contains the payload, which may include service parameters (from the client) or execution results (from the server).

Thanks to XML, SOAP supports interoperability, extensibility, and platform neutrality across various systems .[4]

I.2.2.2 WSDL

WSDL is an XML-based specification for describing web services and how to access them. It defines:

- Abstract definitions: Operations, input/output messages, and data types.
- Concrete implementation: Binding protocols, physical addresses (URLs), and ports.

WSDL enables automated service discovery and integration, acting as a contract between service providers and consumers [5].

I.2.2.3 UDDI

UDDI is a platform-independent registry that stores descriptive metadata about web services. It enables service discovery based on industry type, provider, or interface definition. The registry content is structured into: The data stored in UDDI is structured in XML and organized into three parts known as pages:

- Blank pages: provide general descriptions of service providers, such as the name of the company providing the service, its business identifier, addresses, etc.
- Yellow pages: contain detailed descriptions of service providers listed in the blank pages to classify businesses and services by sector of activity.
- Green pages: provide technical information on the catalogued web services.

  This information includes the description of the service, its usage process and protocols used for its invocation. [6]

I.2.3 Service Oriented Architecture

I.2.3.1 Definition of Service Oriented Architecture

Service-Oriented Architecture (SOA) is a software design paradigm in which independent, self-contained services interact with each other over a network through standardized interfaces and protocols. Each service is responsible for a specific business function and can be reused, discovered, and invoked without knowing its internal implementation. SOA promotes:

- Loose coupling between services

- Interoperability across platforms
- Reusability of components
- Scalability and flexibility in enterprise systems

This architecture enables organizations to build distributed systems that are easier to integrate, extend, and maintain [7].

I.2.3.2 Web Services & SOA

Web services are one of the most widely adopted technologies for implementing SOA. They provide a practical and standardized way for services to communicate using:

- SOAP: for message exchange
- WSDL: for service description
- UDDI: for service discovery

Web services are platform-independent and use open standards (XML, HTTP), making them ideal for connecting heterogeneous systems. In modern enterprise environments, web services allow SOA to be realized effectively by enabling dynamic, automated, and interoperable interactions between services [7].

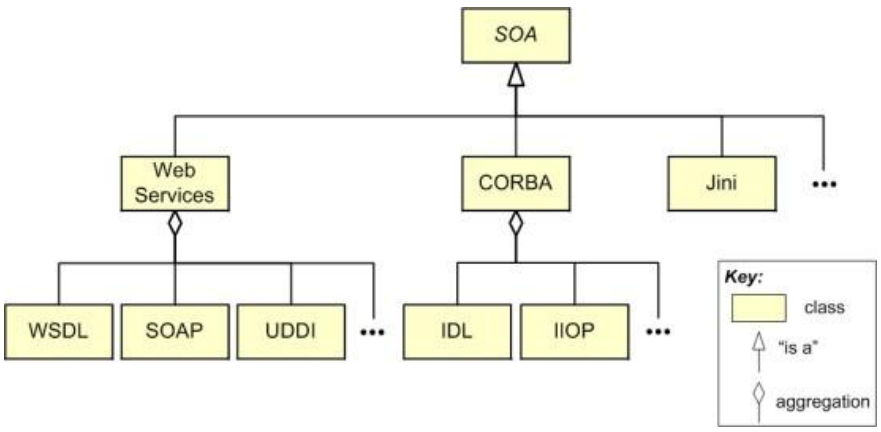


Figure 2: SOA and SOA Technologies [8]

I.3 Recommender Systems

I.3.1 History

Recommender systems have evolved significantly since the early 1990s. In 1992, Belkin and Croft explored the distinction between information filtering and retrieval, laying the conceptual foundation for recommendation technologies. The same year, Goldberg et al. introduced the Tapestry system, the first collaborative filtering model based on human evaluations. This inspired the development of GroupLens, a news recommendation service based on user-user collaborative filtering, later leading to the creation of MovieLens in 1997. During the 2000s, recommender systems gained traction in e-commerce platforms like Amazon, incorporating content-based and hybrid approaches to address cold-start problems. In 2007, the establishment of the ACM RecSys conference marked a milestone in formalizing the field. The 2010s witnessed a shift toward matrix factorization and latent factor models, especially following the Netflix Prize challenge. This era also saw the integration of deep learning, knowledge graphs, and sequence modeling. From 2018 onward, the field embraced advanced AI techniques such as transformer-based models like BERT4Rec, reinforcement learning for adaptive recommendations, and federated learning for privacy-preserving systems. Today, modern platforms like Netflix, TikTok, and Spotify rely on real-time, personalized, and explainable recommender systems to enhance user experience. [9]

I.3.2 Definition of Recommender Systems

Recommender Systems (RSs) are intelligent tools and methods developed to suggest items that might be relevant or useful to users. These recommendations support users in making various choices, such as selecting products to purchase, music to enjoy, or news articles to read online.

The word ”item” broadly refers to the content or product that the system recommends. Typically, each RS is specialized to handle a particular kind of item—such as music albums, movies, or news which affects the system’s structure, user interface, and the algorithm used to deliver suggestions tailored for that category.

These systems are particularly helpful for users who may not have the expertise or experience to assess a large number of available options.[10] Generally, there are two main operational approaches in recommendation systems:

A/ estimating how a user might rate an item they have not yet interacted with.

B/ producing a ranked list of items that are likely to meet the user’s preferences.

I.3.3 Recommendation Models

Since the introduction of the first collaborative filtering model in the 1990s, recommendation systems have become a major topic of research and are now widely used in both academic and commercial settings. These systems, which are a type of information filtering tool, aim to give users personalized suggestions by using available or collected data from the platform. By matching content with the user's individual preferences, the goal is to offer the most useful and relevant recommendations.

While users generally appreciate having multiple options, too many choices can create confusion and lead to decision fatigue. To avoid this, a good recommendation system must provide enough variety without overwhelming the user. This is done by studying both direct (explicit) and indirect (implicit) feedback from the user, as well as comparing their behavior to similar users, in order to generate a customized list of suggestions.[11]

Figure 1 summarizes the general processes of the recommendation system models, such as Collaborative Filtering, Content-Based Filtering, and Hybrid System, at a glance.

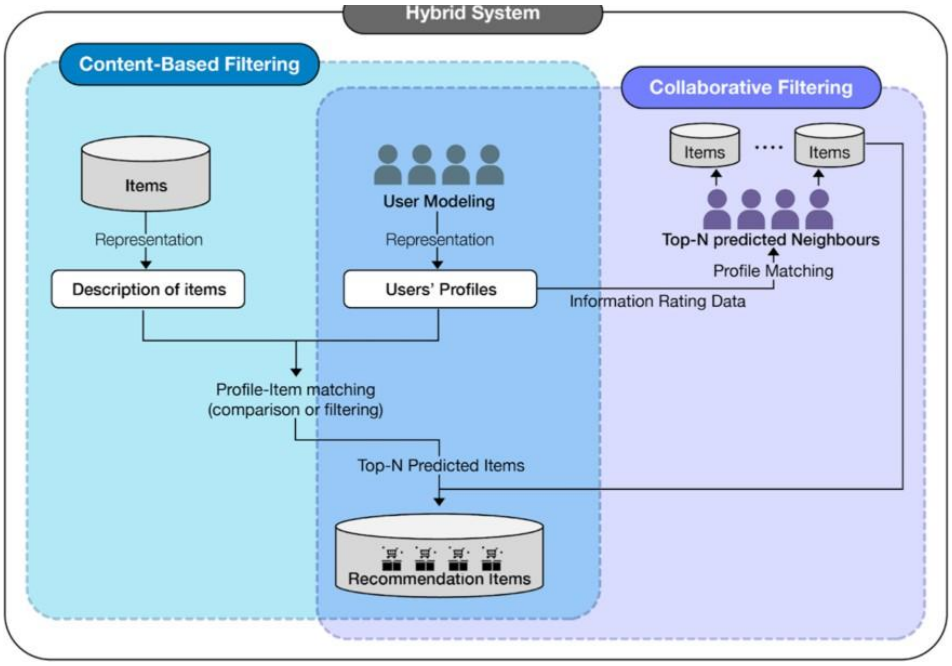


Figure 3: Process of general recommendation system models.[12]

I.3.3.1 Content-Based Filtering

Content-Based Filtering began gaining attention in 1992, following the work of Loeb and colleagues. This method recommends items that share similar features with those the user has previously shown interest in. In simple terms, it analyzes the characteristics of items that a user liked before and suggests other items with similar attributes.

This model was one of the earliest types used in recommendation systems and is considered the most basic. However, as noted by Salter and others, a key limitation of this method is that it only suggests items that are closely related to past user preferences. As a result, it does not help users discover new or diverse content, which can reduce the variety of experiences offered.

Despite these drawbacks, Content-Based Filtering has been widely used in areas where item and user profile information is easy to obtain. Examples include music and movie recommendation platforms, online shopping sugges-

tions, and educational content systems. The model often relies on text mining techniques, semantic analysis, and algorithms like TF-IDF (Term Frequency-Inverse Document Frequency), Neural Networks, Naive Bayes, and Support Vector Machines (SVM).

Since 2012, with the rise of hybrid recommendation models, the use of Content-Based Filtering alone has declined, and its application areas have become more limited.[12] Figure 6 is a visual summary of the item recommendation principle of the Content-Based Filtering Model.

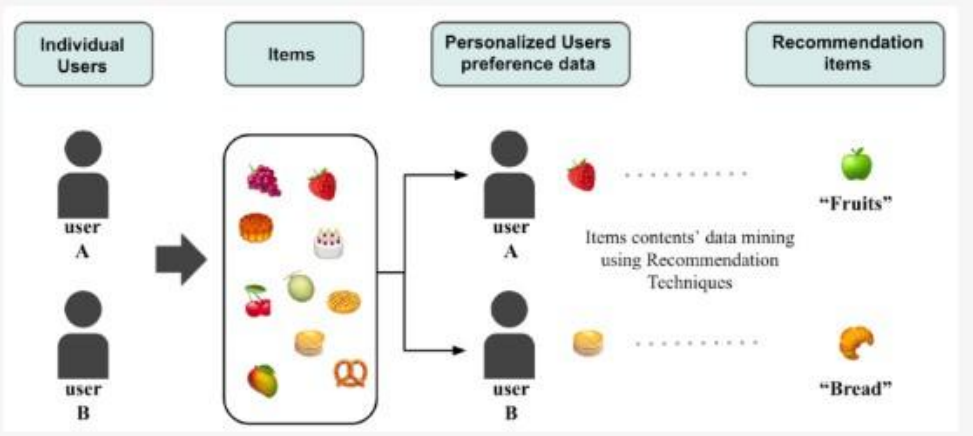


Figure 4: Recommendation principle of Contents-Based Filtering Model.[12]

I.3.3.2 Collaborative Filtering

Collaborative Filtering is an information filtering approach that first emerged in the 1990s, laying the groundwork for future developments in recommender systems. This method builds a database of user preferences by analyzing their rating or evaluation data, which is then used to suggest items aligned with their tastes. Collaborative Filtering techniques are broadly categorized into two types: Memory-Based Collaborative Filtering and Model-Based Collaborative Filtering. The memory-based approach is further divided into two main strategies: User-Based Collaborative Filtering, which finds users with similar preferences and recommends items they liked, and Item-Based Collaborative Filtering, which

identifies items similar to those previously selected by the user.[13]

User-Based Collaborative Filtering works by comparing user ratings on the same group of items to identify similar users. Based on this similarity, it suggests a top-N list of items that those similar users have rated positively. In contrast, Item-Based Filtering estimates a user's potential interest in a new item by calculating the similarity between it and previously rated items—usually using a user-item rating matrix. Both methods fall under the Memory-Based category, and they commonly use similarity measures like Pearson Correlation, Cosine Similarity, and the K-Nearest Neighbors (KNN) algorithm to group users with comparable behaviors and provide customized recommendations. [14]

The Memory-Based approach gained popularity through its application in E-Commerce platforms such as Amazon, where it contributed to improved customer retention and increased purchases. However, when the system lacks sufficient data, it faces three common challenges: sparsity, cold start, and the gray sheep problem. Sparsity happens when there are not enough user ratings available to generate recommendations. The cold start problem arises when new users or items are introduced into the system, leaving no prior data to build recommendations. The gray sheep problem occurs when a user's rating patterns are too unique, making it difficult to find others with similar preferences. [15]

To solve this problems, Model-Based Collaborative Filtering techniques were introduced. These involve learning predictive models from existing user evaluation data. Some of the widely used methods in this category include Clustering, Singular Value Decomposition (SVD), and Principal Component Analysis (PCA), all of which aim to enhance recommendation accuracy even in sparse data environments.

Compared to Content-Based Filtering, Collaborative Filtering has seen broader

adoption due to its strong capability in understanding and predicting user preferences. Nonetheless, this method still struggles with scalability and sparse data, which can impact the quality of recommendations. To counter these weaknesses, Hybrid Filtering models have been developed, combining both Collaborative and Content-Based techniques. Since 2010, research has increasingly focused on refining Collaborative Filtering by improving similarity calculations, incorporating user feedback, and applying techniques like tagging and clustering to enhance the depth and reliability of recommendations.[16], a study aiming to obtain more user preference data using tagging , a study using clustering , etc. Figure 7 is a visual summary of the item recommendation principle of the Collaborative Filtering Model.

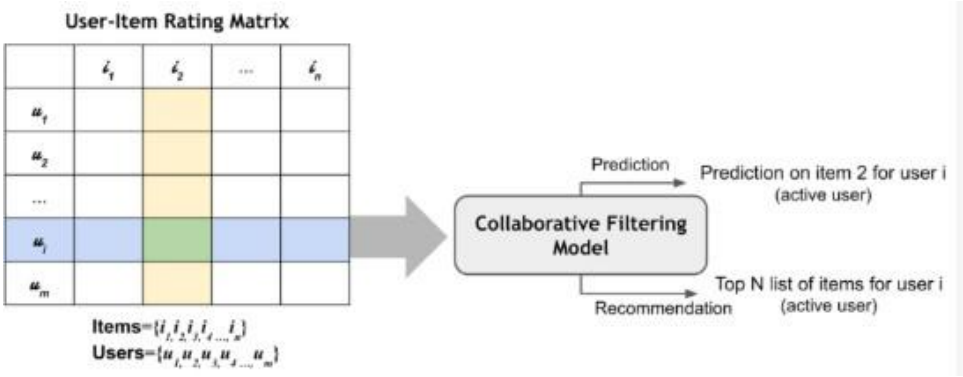


Figure 5: Recommendation principle of Collaborative Filtering Model.[12]

I.3.3.3 Hybrid System

Both filtering approaches have inherent limitations: Content-Based Filtering depends heavily on metadata associated with user-interacted items, while Collaborative Filtering relies on users' item rating data. To address the shortcomings of these individual models and enhance recommendation performance, Hybrid Recommendation Models have been proposed.

Primarily designed to mitigate the data sparsity problem, most research on hybrid models focuses on compensating for the lack of sufficient rating data by

integrating information derived from both Content-Based and Collaborative Filtering techniques. Notable studies in this domain include: the use of a Bayesian Probabilistic Matrix Factorization Framework to enrich user preference data; employing autoencoders to learn from limited user interaction data and extract nonlinear patterns while reducing noise; and constructing hybrid models that incorporate side information, such as social network connections and item attributes, in addition to explicit user ratings. [17]. These studies each attempted to alleviate the sparsity problem. Figure 4 is a visual summary of the item recommendation principle of the Hybrid Recommendation model.

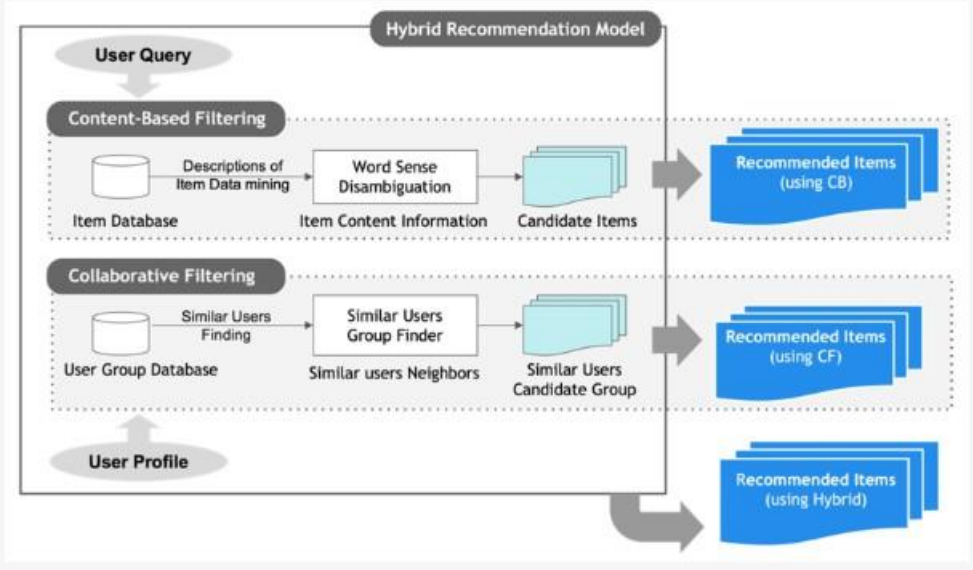


Figure 6: Recommendation principle of Hybrid recommendation model.[12]

I.3.4 Qualitative Evaluation Metrics of Recommendation Systems

To evaluate the performance of a recommendation system model, it is essential to quantify the characteristics of an effective recommendation system [18]. This study analyzes the commonly used measurement criteria for assessing the quality of models in recommendation system research. The most common qualitative evaluation metrics include Precision , Recall , Accuracy , F-Measure [19]. These metrics are calculated using the confusion matrix, which helps visualize the predicted and actual values, making it possible to measure the system's performance quantitatively.

The confusion matrix classifies items based on user preferences and the system's recommendations. The components of the confusion matrix are defined as follows:

True Positives (TP): The number of items that matched the user's preference when recommended by the system.

True Negatives (TN): The number of items the user does not prefer and that were not recommended.

False Positives (FP): The number of items recommended by the system that the user does not prefer.

False Negatives (FN): The number of preferred items that were not recommended by the system.

Accuracy is commonly used to evaluate the performance of recommendation models. It is calculated as the ratio of successful recommendations to the

total number of recommendations made. However, accuracy alone can be misleading when dealing with imbalanced data, where the number of preferred and non-preferred items is not equal .[20]

To ensure a more effective evaluation, Precision and Recall are widely used:

-Precision is the proportion of recommended items that are actually relevant to the user.

-Recall is the proportion of relevant items that have been successfully recommended by the model.

Evaluation Metrics	**Equation**	**Definition**
Precision	$\text{Precision} = \frac{TP}{TP + FP}$	The ratio of items that match the user's taste among all items recommended to the user.
Recall	$\text{Recall} = \frac{TP}{TP + FN}$	The ratio of items that match the user's taste among all items recommended to the user.
Accuracy	$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$	The ratio of successful referrals to total referrals.
F-Measure	$\text{F-Measure} = 2\frac{\text{Precision}\,\text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean value of Precision and Recall.

Table 1: Evaluation Metrics for Recommendation Systems

Since Precision and Recall often have a trade-off—where improving one may reduce the other—the F-Measure is used to balance both by calculating their harmonic mean [21]. The values of Precision, Recall, and F-Measure range from 0 to 1, with higher values indicating better model performance.

I.3.5 Application Fields

Recommendation systems have seen broad adoption across various service domains. This study aims to examine how the recommendation models and technologies previously described are implemented and tailored based on the characteristics and purposes of real-world services. Based on a review of relevant literature, the application domains of recommendation systems have been grouped into seven primary categories: Streaming Services, Social Network Services, Tourism Services, E-Commerce Services, Healthcare Services, Education Services, and Academic Information Services.

These categories were established by analyzing both the frequency of service mentions in the literature and their increasing user engagement or business value. The classification also considered search trends on platforms like Google Scholar using the keyword ”Recommendation System.” A summary of the key service areas where recommendation systems are frequently applied is illustrated in the study (Figure 8).d in the recommendation system to be described in this section.

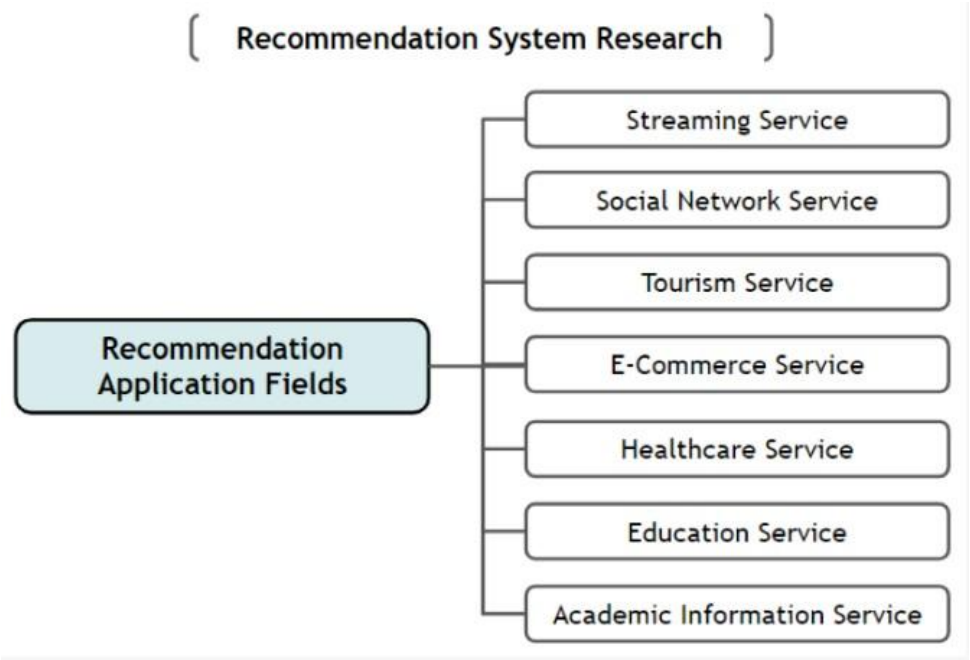


Figure 7: List of recommendation fields [12].

3.5.1 Streaming Service

In the past, users accessed video content primarily through television or cinemas. However, the rise of platforms such as Netflix and YouTube has significantly shifted content consumption toward online streaming. Similarly, audio content, once downloaded onto personal devices, is now primarily accessed through streaming platforms like Spotify.

This shift in user behavior has necessitated the evolution of recommendation systems, which serve a crucial role in helping users navigate the overwhelming volume of content. These systems personalize suggestions based on individual consumption histories, leveraging user data to match preferences with items in the platform's content catalog.[22].

3.5.2 Social network services

Online social networking services (SNS) such as Facebook, Instagram, Twitter, and LinkedIn serve as major digital platforms where users can document aspects

of their daily lives, interests, and hobbies while engaging with others. The rapid growth in SNS usage has led to a significant increase in user-generated data. This includes both content users share through their posts and their interactions with other content, such as likes, comments, and ratings. Such rich data is not only leveraged to enhance recommendations within the SNS platforms themselves but is also valuable for recommendation systems in other domains.
In essence, the diverse data collected through SNS plays a vital role in advancing the effectiveness of recommendation technologies.[23]

3.5.3 Tourism Service

Social networking platforms like Facebook, Instagram, Twitter, and LinkedIn allow users to share their interests, daily activities, and engage with others' content. The exponential increase in SNS usage has led to a significant growth in user-generated data, including posts, likes, comments, and interactions.

This wealth of data enhances internal recommendation mechanisms within the platforms and can also be leveraged to improve recommendation systems in other domains. In essence, SNS platforms offer valuable contextual and behavioral data that supports more effective personalization.[24]

3.5.4 E-Commerce Service

In the past, items such as clothes, food, and books were mainly consumed by users through offline stores. However, e-commerce sites like Amazon, eBay, and Alibaba have altered the way people consume goods in recent years due to the growth of digital platforms like the web and applications. E-commerce gives customers a wide range of products and choices in an online setting while giving merchants a simple way to make sales. Specifically, the COVID-19 lockout

mechanisms prevented customers from going outdoors, which prevented them from using physical stores. As a result, consumption through digital channels skyrocketed. Additionally, the types of goods offered for sale on digital platforms started to expand. [25]

3.5.5 Healthcare Service

As interest in health increases, the number of users who use smart wearable devices has started to increase as the technology has become compatible with smartphones, and their convenience of use has also increased .Such wearable devices can efficiently monitor the user's bio-state. One example of a wearable technology is the smart watch, which collects bodily data on a regular basis, helps people without expert medical knowledge prevent ailments, and allows self-diagnosis. These wearable gadgets gather a great deal of user biometric information to aid in research on diseases or proper diagnosis based on certain bodily conditions. It has also proven used in studies that suggest treatments. In order to give consumers insight into better treatment alternatives, studies on health-related recommendation systems examined the connection between illnesses and patients' symptom patterns. The recommendation system utilized in the healthcare service industry was separated into two categories in this study: e-health and health recommendation systems., which support professional treatment according to the purpose of the system application.[26]

3.5.6 Education Service

From the traditional form of education in the classroom or lecture hall, a new education trend, called Smart Learning, has formed through e-learning, in which learning is conducted through an online environment . Smart education has

started to be gradually used in education due to the increase in the spread of various smart devices and the development of wireless networks. Smart education can access vast digital resources and seamlessly provide personalized learning tailored to the needs, goals, talents, and interests of learners without time and space constraints. In addition, the educational form has improved by reflecting the learning trend of the digital age . Therefore, the field of education services using the recommendation system provides learning resources in consideration of the learning style and knowledge level of learners, thus providing an effective and efficient learning experience. In other words, personalized learning content can be provided to learners.[27]

3.5.7 Academic Information Service

The exponential growth of academic content has made it increasingly challenging for researchers to find relevant resources efficiently. As a solution, recommendation systems have been developed to support academic workflows by suggesting suitable content.

These systems are commonly integrated into digital libraries, especially university-based platforms, to help scholars access relevant publications, manage data, write academic papers, and navigate the research lifecycle. The ultimate aim is to provide personalized, meaningful academic recommendations that assist researchers, academic institutions, and practitioners.[12]

I.4 conclusion

Recommender systems represent an emerging computational paradigm that employs a multidisciplinary approach. They analyze and model user behaviors

across various web services and platforms to identify patterns and trends. The goal of these systems is to develop applications that can understand user preferences and needs, offering personalized suggestions by leveraging data from users with similar behaviors. This approach enhances user experience by saving time and effort, making it easier for individuals to discover relevant content or products.

Chapter II:

Association Rules

II.1 Introduction

Data mining involves extracting meaningful patterns, relationships, and insights from vast datasets, uncovering previously unknown yet valuable information. This transformative process turns raw data into actionable knowledge, supporting decision-making across fields such as business, healthcare, and scientific research. By utilizing techniques from domains like machine learning, statistics, and database management, data mining reveals hidden knowledge that fosters innovation and enhances efficiency.

Within the sphere of data mining techniques, association rule mining stands out for its ability to uncover frequent patterns and correlations between variables in extensive datasets. These rules, such as ”Customers who purchase product A are likely to purchase product B,” play a critical role in applications like market basket analysis, fraud detection, and personalized recommendation systems. By identifying these associations, organizations gain deeper insights into their data, empowering them to make strategic, data-driven decisions.

II.2 Data Mining

II.2.1 Definition of Data Mining

Data mining refers to a collection of advanced techniques and algorithms designed to help analysts and users uncover solutions to complex problems that would otherwise demand significant manual effort or remain unresolved. It can also be viewed as a set of tools that assist users in identifying meaningful patterns and trends from selected portions of enterprise data. This analytical process is particularly valuable in various business contexts, such as forecasting future outcomes and gaining deeper insights into the factors that contributed to past performance [28].

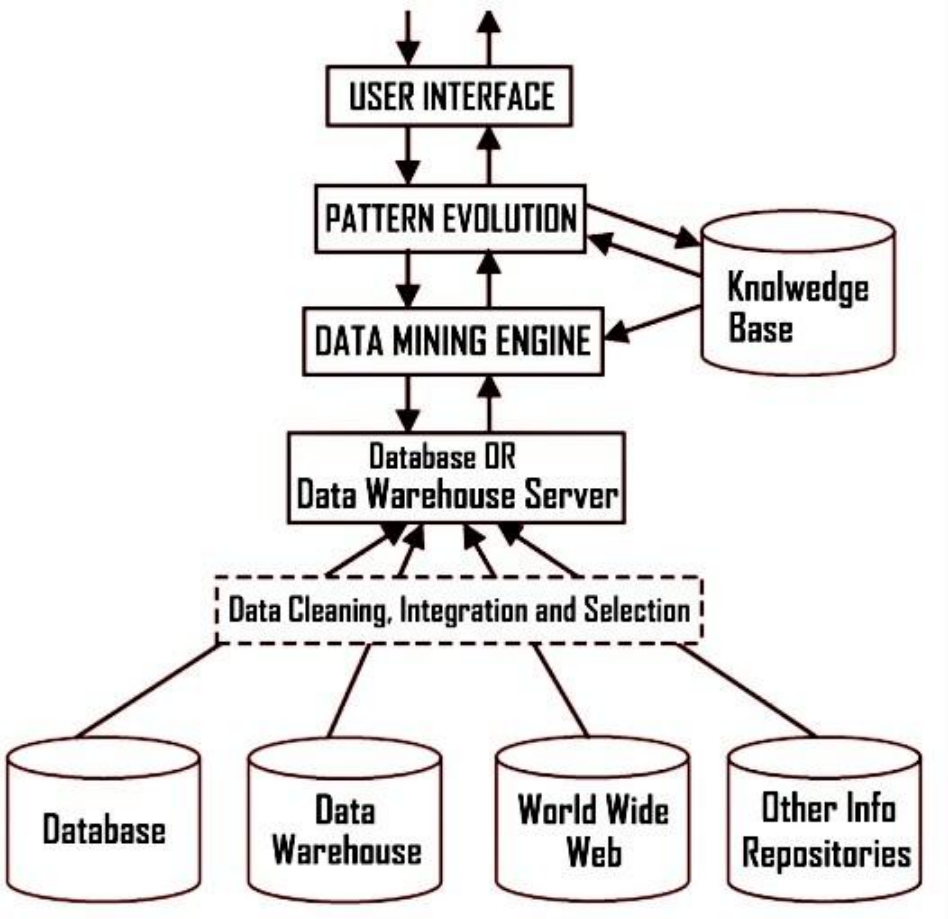


Figure 8: Architecture of Data Mining [29]

II.2.2 Importance of Data Mining

Data mining is the process of analyzing large datasets to detect unusual patterns, hidden relationships, and useful trends that can help in predicting future outcomes. By applying various techniques, this process enables organizations to make informed decisions that can lead to increased profits, cost reduction, better customer engagement, risk minimization, and other operational improvements [29].

II.2.3 Applications of Data Mining

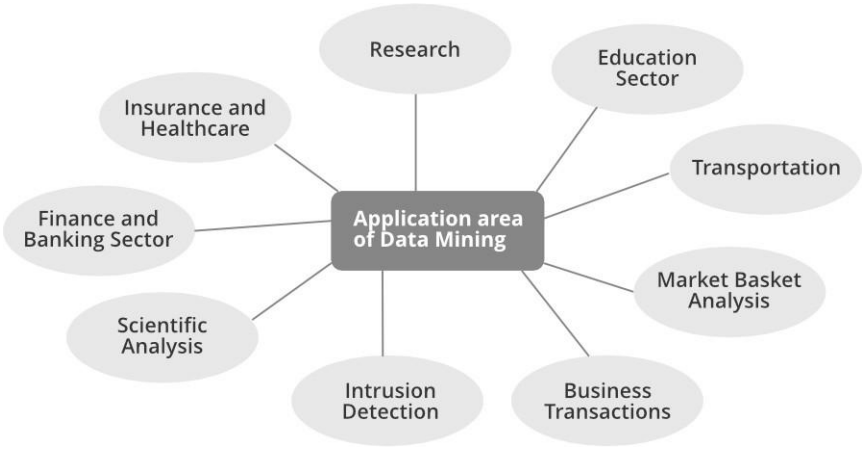


Figure 9: Applications of Data Mining

Scientific Analysis: Scientific simulations are producing massive volumes of data on a daily basis, including information from areas such as nuclear research and human psychology studies. Data mining techniques offer powerful tools to explore and analyze these large datasets. However, the pace at which new data is being collected and stored has outstripped our ability to analyze the existing data effectively.

Examples of scientific data analysis include:

- Analyzing biological sequences in the field of bioinformatics.
- Categorizing celestial bodies in astronomy based on observed data.
- Supporting medical decision-making through data-driven diagnostic and treatment tools[30].

Education: To analyze the education sector, data mining applies a method known as Educational Data Mining (EDM). This approach generates patterns

that can be used by both learners and educators. By using EDM, several educational tasks can be performed, such as:

- Predicting student admissions to higher education.
- Forecasting student profiles.
- Predicting student academic performance.
- Evaluating teaching effectiveness of instructors.
- Supporting curriculum development.
- Predicting students placement opportunities[30].

Research: Data mining techniques are capable of performing various analytical tasks such as prediction, classification, clustering, association rule discovery, and data grouping with high accuracy, particularly in research applications. The patterns or rules generated through data mining are often distinctive and valuable for extracting meaningful results. In most technical studies involving data mining, researchers use a model development approach based on training and testing phases. This methodology, known as the Train/Test strategy, involves dividing the dataset into two parts: one for training and another for testing. The training dataset is used to build or learn the model, while the testing dataset is applied to evaluate its accuracy and performance[30]. Example:

- Classification of imprecise or uncertain data.
- Clustering based on informational attributes.
- Decision support systems.

- Web mining for extracting insights from online data.
- Domain-driven data mining approaches.
- Applications in Internet of Things (IoT) and cybersecurity.
- Smart farming solutions using IoT technologies.

II.2.4 Data Mining and KDD

Knowledge Discovery in Databases (KDD) is the process of finding meaningful, reliable, and understandable patterns or structures in large datasets. This process includes several steps: selecting or sampling relevant data from a data warehouse, cleaning and preprocessing the data, transforming or reducing it if necessary, applying data mining techniques to extract useful patterns, and finally evaluating the results.Data mining is one key step within the KDD process. It focuses on the use of algorithms to discover patterns or structures in the data, while keeping computational resources within practical limits. For the extracted patterns to be considered as knowledge, they must satisfy certain criteria: they should be valid, easy to understand, useful, novel, and interesting[31].

II.2.5 Technique of Data Mining

Data mining is highly effective, as long as it draws upon one or more of these techniques:

2.5.1 Classification

Traditional neural network approaches often face several challenges, such as difficulties in understanding underlying issues, a tendency to overfit the data due to optimization constraints, and limitations in geometric strategies used for selecting the most appropriate model. As a result of these shortcomings, Support Vector Machines (SVMs) have gained increasing attention in recent years because of their strong theoretical foundations, robust generalization capabilities, and impressive empirical performance[5]. For example, SVMs can be applied in classification tasks such as distinguishing between spam and non-spam emails, or categorizing customers into ”high-risk” and ”low-risk” groups based on their behavioral data .

In the same way, if you were organizing clothing items, you could group them based on color—placing all green garments in one category, red in another, and blue in a third. In this case, the classification criterion would be color. Alternatively, you could categorize the clothing by size, grouping all small items together, all medium items together, and so on. Each approach reflects a different basis for classification depending on the chosen attribute[32].

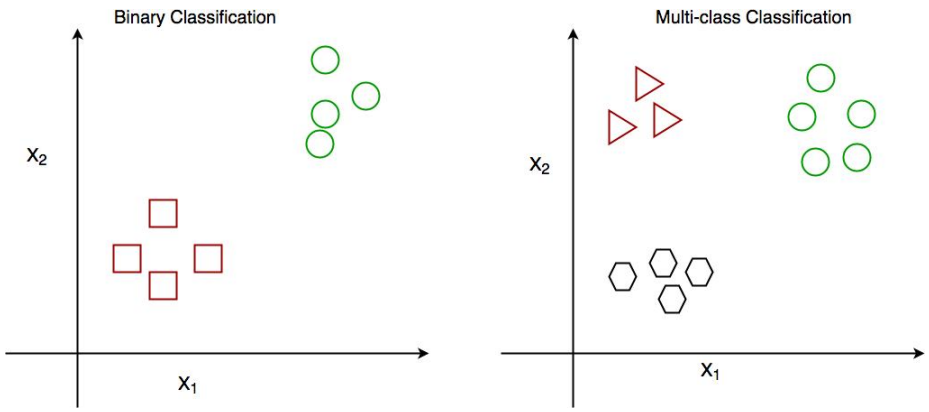


Figure 10: Classification - GeeksforGeeks

2.5.2 Clustering

Clustering is a data mining technique used to group similar data points into distinct clusters based on shared characteristics. The primary objective is to uncover patterns or similarities within the data without requiring any prior knowledge about its structure or predefined labels. This unsupervised learning approach is widely applied in various domains, such as market segmentation, image analysis, and anomaly detection[33].

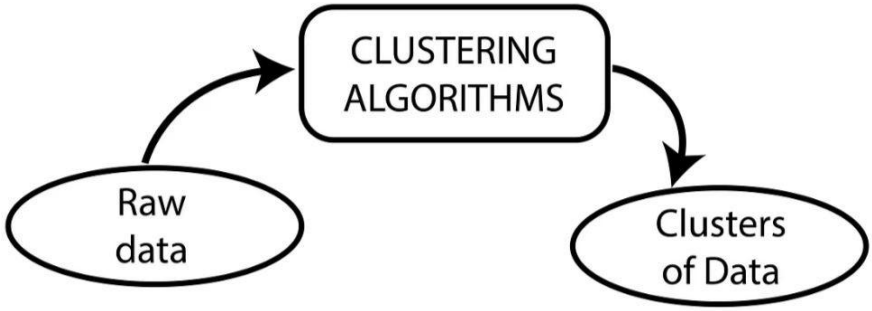


Figure 11: Clustering diagram

2.5.3 Association Rule Mining

This technique is used to uncover meaningful relationships or associations among different items or attributes within a dataset. It helps reveal patterns such as ”if item A occurs, then item B is likely to occur” or ”customers who purchase product X often also buy product Y.” Common algorithms used in association rule mining include Apriori and FP-Growth, which are effective in discovering such frequent itemsets and association rules [34].

2.5.4 Regression

Regression is a statistical method used in data mining to explore the relationship between a dependent variable and one or more independent variables. Its main

purpose is to estimate or predict the value of the dependent variable based on the values of the input features. In simple linear regression, the model involves only one independent variable, while in multiple linear regression, several predictors are used. This approach, known as multiple regression analysis, is valuable for both prediction and understanding the impact of independent variables on the outcome. Additionally, other types of regression models exist, such as logistic regression for categorical target variables and nonlinear regression for capturing non-linear relationships between variables [33].

2.5.5 Anomaly Detection

Anomaly detection is a data mining technique used to identify data points that significantly differ from the normal or expected patterns within a dataset. This method is effective for uncovering errors, fraudulent behavior, unusual patterns, or novel instances. For example, anomaly detection can be applied to identify missing or incorrect values in data, detect fraudulent transactions, recognize rare or unexpected events, or discover innovative and uncommon insights. By applying anomaly detection, organizations can enhance data quality, strengthen security, and foster creativity through the identification of outliers and unexpected behaviors[35].

II.3 Association Rules

Association rules are conditional statements in the form of ”if-then” that indicate the likelihood of relationships between items within large datasets stored in various types of databases. At its core, association rule mining uses machine learning techniques to examine data and uncover patterns known as co-

occurrences. The goal is to find frequently occurring item combinations, which are then expressed as association rules that reveal meaningful connections between different data elements[36].

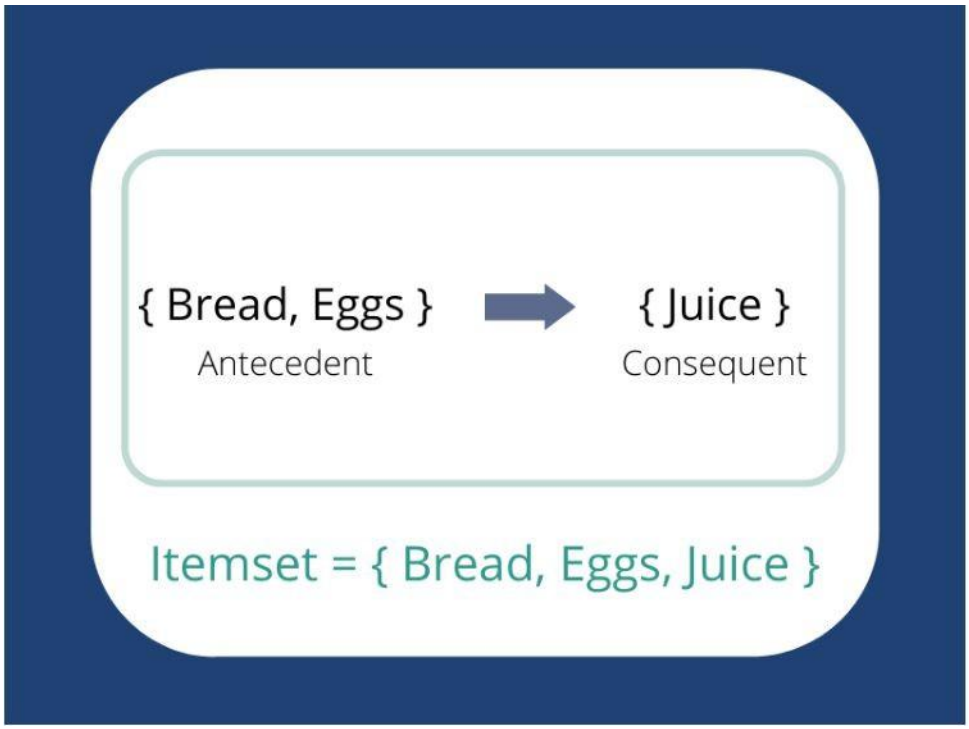


Figure 12: Exemple of Association Rules

II.3.1 History of Association Rules

Association rule mining, first proposed by Agrawal et al. in 1993, is a technique aimed at uncovering meaningful relationships among items within large transactional datasets. It is widely applied in domains such as retail to understand customer purchasing behavior. One of the main challenges in this process is efficiently identifying frequent itemsets, as their number can grow exponentially with the size of the dataset. To address this, algorithms such as Apriori and FP-Growth have been developed to minimize the number of database scans by pruning infrequent itemsets early in the process. Additionally, techniques such

as incremental updates and the use of user-defined constraints help narrow the focus to more relevant and useful rules. Various algorithms continue to improve the efficiency and accuracy of association rule discovery[37].

II.3.2 Definition of Association Rules

Association rule mining is a data mining technique used to discover meaningful patterns within large datasets. It focuses on identifying relationships between variables and using these connections to support predictions or decision-making processes. The main objective of this approach is to generate rules that reveal how different items or attributes in the dataset are related to one another [38]

Association rule mining is a method used to uncover meaningful relationships and hidden patterns within large datasets. The process involves detecting frequent itemsets—groups of items that commonly occur together—and then generating association rules based on these itemsets. These rules help reveal how the presence of certain items in a dataset is related to the presence of others, providing valuable insights for decision-making and pattern recognition [39]

II.3.3 Advantages and Disadvantages of Association Rule Mining

Advantages of Association Rule Mining

- Reveals hidden patterns and relationships: Association rule mining enables the discovery of complex and previously unknown relationships among

variables that might not be identified using traditional data analysis methods.

- Improves customer insight: This technique helps businesses gain a deeper understanding of customer behaviors, preferences, and purchasing habits, allowing for more effective and personalized marketing strategies.
- Supports predictive modeling: Association rule mining can serve as a valuable preprocessing step in predictive analytics by identifying important features and associations that enhance model accuracy and relevance [40].

Disadvantages of Association Rule Mining

- High computational cost: When working with large datasets, association rule mining can become computationally demanding due to the exponential growth of potential item combinations.
- Risk of overfitting: There is a possibility of overfitting, where the derived rules fit the training data too closely and fail to generalize effectively to new, unseen data [40].

II.3.4 Application areas of Association Rule

Market basket analysis

Market Basket Analysis is a data mining approach commonly used in retail to gain insights into customer purchasing behavior with the aim of boosting sales. This method examines large-scale transaction data, such as purchase histories,

to identify groups of items that are frequently bought together. By revealing these product associations, businesses can optimize product placement, marketing strategies, and recommendation systems. Market Basket Analysis is a well-known and practical application of association rule mining [41].



Figure 13: Market Basket Analysis

Medical Diagnosis

Association rule mining is widely used in medical diagnosis to discover hidden patterns and relationships in patient data. By analyzing large medical datasets, such as symptoms, lab results, and diagnoses, this technique can identify frequent combinations of symptoms that may indicate specific diseases. These rules help doctors make faster and more accurate diagnoses, support decision-making, and even predict possible future health issues. As a result, association rule mining plays a valuable role in enhancing diagnostic systems and improving healthcare outcomes[42].

Protein Sequence

Association rules are valuable in identifying patterns that support the synthesis of artificial proteins, aiding research in bioinformatics and computational biology. Beyond scientific applications, they are also applied in areas such as catalog design and loss-leader analysis, among many other fields, to improve decision-making and strategic planning based on discovered item relationships [43] .Association Rule Mining (ARM) is applied to uncover relationships between amino acids within protein sequences. Identifying these associations plays a key role in predicting the three-dimensional structure and biological function of proteins. This insight is especially valuable for the design and synthesis of artificial proteins, contributing to progress in fields such as biotechnology and medical research [44].

Social network analysis

Many organizations apply association rule mining to extract meaningful patterns from social media data, which aids in understanding and analyzing social networks. For instance, analyzing data from platforms like Twitter can reveal that users who frequently post about a specific topic also tend to engage with related subjects. These discovered associations can help identify clusters or communities of users within the social network based on shared interests or behaviors[38].

Census data

Census data provides a broad range of statistical insights about society, which are valuable to both researchers and the general public. Information derived from population and economic censuses can be used to forecast and plan essential public services such as education, healthcare, transportation, and funding. It also plays a key role in guiding private sector decisions, such as the establishment of new factories, shopping centers, banks, or in targeting specific products for marketing.

Applying data mining techniques to census and other official datasets holds significant potential for informing public policy and supporting the effective operation of democratic systems. However, this process is complex and demands rigorous methodological approaches, an area of research that is still in the early stages of development[42].

II.3.5 Association Rules Interestingness Measures

Association rules are evaluated using key metrics that determine their relevance, strength, and reliability. These metrics include support, confidence, and lift, which quantify the frequency and strength of relationships between data items.

3.5.1 Support

Support is a key metric in association rule mining that quantifies how frequently a specific itemset appears within a dataset. It represents the proportion of transactions in which the particular combination of items occurs compared to the total number of transactions [45]. Support quantifies the frequency with which

an itemset—including both the antecedent and consequent—occurs within the dataset. It reflects how prevalent a specific association is among all transactions[46].

Formula:

$$\text{Support} = \frac{\text{Transactions containing both antecedent and consequent}}{\text{Total transactions}}$$

Example: If bread and butter appear together in 100 out of 1,000 transactions, the support is:

$$\text{Support} = \frac{100}{1000} = 0.10\,(10\%)$$

A higher support value indicates a more frequently occurring pattern in the dataset [46].

3.5.2 Confidence

Confidence is a metric used in association rule mining to measure the reliability or strength of a discovered association between two items in a dataset. It indicates the likelihood that an item Y is purchased when item X is purchased, representing the conditional probability of the consequent given the antecedent. Confidence for an association rule $X \rightarrow Y$ is calculated as the ratio of the number of transactions containing both X and Y to the number of transactions containing X:

Formula:

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cap Y)}{\text{Support}(X)}$$

Confidence measures the strength and dependability of association rules derived from frequent itemsets. Rules with high confidence suggest that the occurrence of the antecedent strongly impacts the likelihood of the consequent, making such rules more meaningful and trustworthy for decision-making [47].

3.5.3 Lift

Lift is a metric that evaluates how much more likely the consequent is to occur when the antecedent is present, compared to its expected occurrence if the two were independent. It is calculated as the ratio between the confidence of the rule and the overall frequency of the consequent in the dataset. For instance, in the rule bread, butter jam, lift is obtained by dividing the confidence of the rule by the proportion of transactions that include jam. A lift value greater than 1 indicates a strong and meaningful association between the items, making the rule potentially valuable and interesting [48].

Lift for an association rule $X \rightarrow Y$ is calculated as the ratio of the observed support of the itemset X and Y to the expected support under independence:

Formula:

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Support}(X \cap Y)}{\text{Support}(X)\text{Support}(Y)}$$

Lift interpretation

- A lift value greater than 1 indicates a positive association between the an-

tecedent and the consequent, meaning they occur together more frequently than expected under independence. This implies that the presence of the antecedent increases the probability of observing the consequent.

- A lift value less than 1 suggests a negative or inverse relationship, where the presence of the antecedent actually reduces the likelihood of the consequent occurring.
- A lift value equal to 1 denotes statistical independence, indicating that the occurrence of the antecedent has no influence on the occurrence of the consequent they appear together purely by chance [47].

II.3.6 Association Rules Mining : Algorithms

Data mining academics have given association rule mining a lot of attention, and as a result, several algorithms have been proposed for it.

II.3.6.1 AIS(Artificial Immune System)

The AIS algorithm performs multiple scans over the entire transaction database. In each pass, it examines all transactions. During the initial pass, it computes the support of individual items to identify those that are frequent, or large, within the dataset. These frequent itemsets are then extended in subsequent passes to generate candidate itemsets. After processing each transaction, the algorithm identifies itemsets that are common between the frequent itemsets from the previous iteration and the current transaction's items. AIS was the first algorithm introduced for generating all frequent itemsets in a transactional database. It

was designed to enhance database systems with capabilities required for decision support query processing and was primarily aimed at discovering qualitative association rules. However, a notable limitation of this technique is that it supports only a single item in the consequent of a rule [49].

Advantage:

- The AIS algorithm was utilized to explore potential associations between different departments based on patterns in customer purchasing behavior.

Disadvantage:

- A major limitation of the AIS algorithm is its tendency to generate a large number of candidate itemsets, many of which are eventually found to be infrequent and thus are discarded. This excessive generation increases computational complexity and decreases overall efficiency.
- In addition to restricting association rules to a single item in the consequent, another significant shortcoming of the AIS algorithm is the inefficiency of its data structures. These structures are not well-suited for large-scale data processing, resulting in suboptimal performance in terms of memory usage and execution speed [49].

II.3.6.2 Apriori algorithm

The Apriori algorithm is an unsupervised machine learning technique employed for association rule mining. It is designed to identify frequent itemsets and uncover meaningful relationships, correlations, or dependencies among groups of

items within large datasets. This algorithm is widely used in various applications, including disease prediction and recommendation systems. A notable example is market basket analysis in e-commerce, where Apriori helps determine which products are commonly purchased together, enabling businesses to optimize cross-selling strategies and customer experience[50].

How the Apriori Algorithm Works:

The Apriori Algorithm operates through a systematic process that involves several key steps:

- **Identifying Frequent Itemsets:** The Apriori algorithm starts by scanning the dataset to determine the frequency of individual items, known as 1-itemsets. Based on a predefined minimum support threshold, it then filters these itemsets to identify those that occur frequently enough to be considered significant for further analysis.

- **Creating Possible Item Groups:** After identifying the frequent 1-itemsets (individual items meeting the minimum support threshold), the Apriori algorithm proceeds to generate candidate 2-itemsets by pairing these frequent items. This iterative process continues, incrementally constructing larger itemsets (k-itemsets) by combining previously discovered frequent (k−1)-itemsets. The algorithm terminates when no further frequent itemsets can be generated.

- **Removing Infrequent Item Groups:** The Apriori algorithm utilizes a pruning strategy grounded in the Apriori Property, which asserts that if an itemset is infrequent, then all of its supersets must also be infrequent. By applying this property, the algorithm effectively eliminates unpromising candidate itemsets early in the process, thereby reducing the computational

complexity and the number of combinations that need to be assessed.

- **Generating Association Rules:**Once frequent itemsets are identified, the algorithm proceeds to generate association rules that describe the relationships between items. These rules are evaluated using metrics such as support, confidence, and lift to measure the significance and strength of the associations [51].

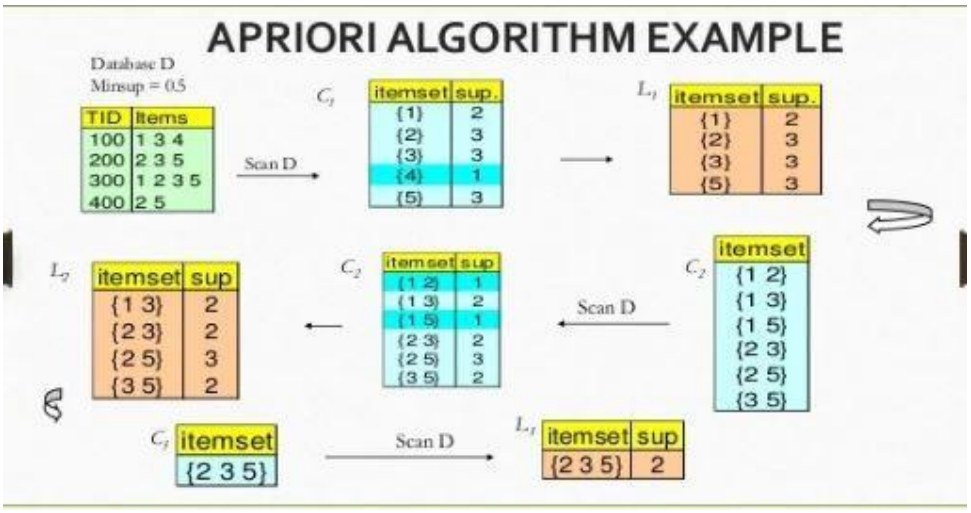


Figure 14: Apriori algorithm example [24]

II.3.6.3 FP-Growth Algorithm (Frequent Pattern Growth algorithm)

The FP-Growth algorithm builds a compact data structure called an FP-tree and employs a divide-and-conquer approach to mine frequent itemsets. Notably, it requires only two scans of the database, making it more efficient than algorithms that need multiple passes.

An FP-Tree is a specialized tree data structure constructed from transaction data during the process of mining frequent itemsets with the FP-Growth algorithm. To build an FP-Tree, the transaction dataset is first scanned to calculate

the support count of each item. Next, the tree is created where each node corresponds to an item and stores its frequency count. The tree has a root node that does not represent any item, often denoted as None or Null, which serves as the starting point. The child nodes of any given node represent items that frequently appear together with the parent item in the transactions, capturing the co-occurrence relationships within the dataset [52].

Examples with a Sample data set

Table 2: Sample Dataset [53]

Transaction ID	Items Purchased
T1	A, B, C
T2	B, D
T3	A, C, D, E
T4	A, D, E
T5	B, E
T6	A, B, C, E
T7	A, B
T8	C, D, E
T9	A, B, D
T10	D, E

Step 1:Define the Support Threshold

- Support Threshold: 20% (meaning an itemset must appear in at least 2 out of 10 transactions to be considered frequent)

Step 2: Count Item Frequencies

- Count the frequency of each item.
- Frequencies: A: 6, B: 6, C: 4, D: 6, E: 6.

Step 3: Sorting and Cleaning Data

- Remove all items that do not meet the minimum support threshold (none in this case).
- Sort items in each transaction by frequency.

Step 4: Building the FP-Tree

0. Initialize FP-Tree: Begin with a root node that is null (contains no item).
0. Process Transactions: Insert each transaction into the FP-Tree by creating or updating nodes according to the ordered items in the transaction.
    - For T1 (A, B, C): Add branches for A, B, and C. Node counts are A:1, B:1, C:1.
    - For T2 (B, D): Add branches for B and D. Node counts for new branches are B:1, D:1.
    - Continue this process for all transactions.

Step 5: Extracting Frequent Itemsets

0. Start from the Smallest Item: In our example, let's start with E.
0. Find Prefix Paths: Look for all paths in the tree ending in E.
    - Let's say, for E, we find prefix paths A-D-E, B-E, and C-D-E.
0. Create Conditional FP-Trees: For each prefix path, create a tree and count the frequency of items.
    - For A-D-E, the frequency might be A:3, D:3, E:3.
    - For B-E, the frequency might be B:3, E:3.

- For C-D-E, the frequency might be C:2, D:2, E:2.

0. Extract Frequent Itemsets: Any combination in these trees that meets the threshold is a frequent itemset.

   - From A-D-E: {A, E}, {D, E}, {A, D, E}.
   - From B-E: {B, E}.
   - From C-D-E: {C, E}, {D, E}, {C, D, E}.

0. Repeat for Remaining Items: Continue this process for D, C, B, and A.

This example illustrates how the FP-Growth algorithm efficiently discovers frequent itemsets by constructing and mining the FP-Tree. Unlike algorithms such as Apriori, it bypasses the expensive candidate generation process, making it highly effective for handling large datasets. The result is a collection of frequent itemsets, which serve as the foundation for generating valuable association rules, particularly in applications like market basket analysis [53].

Table 3: Comparison of Association Rule Mining Algorithms [52]

Characteristics	**AIS**	**Apriori**	**FP-growth**
Data support	Less	Limited	Very Large
Speed in initial phase	Slow	High	High
Speed in later phase	Slow	Slow	High
Accuracy	Very less	Less	More accurate

II.3.6.4 ECLAT Algorithm

The ECLAT (Equivalence Class Clustering and bottom-up Lattice Traversal) algorithm is a variation of the Apriori method that uses a top-down strategy. It

groups items into equivalence classes based on their support counts and generates association rules by combining these classes within a lattice structure. Compared to the traditional Apriori algorithm, ECLAT provides improved efficiency and better scalability [44]. The ECLAT algorithm uses a depth-first search (DFS) approach to explore itemsets, which allows it to handle vertical data formats efficiently. In contrast, the Apriori algorithm employs a breadth-first search (BFS) strategy, scanning the transaction dataset level by level to identify frequent itemsets. This fundamental difference in traversal methods contributes to ECLAT's superior performance on certain types of datasets, particularly when dealing with sparse or large-scale data [54].

How ECLAT Algorithm Works [55] :

Step 1: Build the Tidset

The first step in the process is to generate a Tidset for each item. A Tidset is a list that contains the IDs of all transactions in which the item is found. This helps in tracking where each item appears in the dataset. For example:

- $k = 1$
- Minimum support = 2

Table 4: Tidset Representation of Items

Item	Tidset
Bread	T1, T4, T5, T7, T8, T9
Butter	T1, T2, T3, T4, T6, T8, T9
Milk	T3, T5, T6, T7, T8, T9
Coke	T2, T4
Jam	T1, T8

Step 2:Determine Itemset Support Using Tidset Intersections

In this step, the ECLAT algorithm combines the tidsets of different items. The support of a given itemset is calculated by finding the intersection of the tidsets of its individual items. This process is repeated recursively to identify larger frequent itemsets. For example:

- $k = 2$

Table 5: Generation of 2-Itemsets with Corresponding Tidsets

Item	Tidset
{Bread, Butter}	{T1, T4, T8, T9}
{Bread, Milk}	{T5, T7, T8, T9}
{Bread, Coke}	{T4}
{Bread, Jam}	{T1, T8}
{Butter, Milk}	{T3, T6, T8, T9}
{Butter, Coke}	{T2, T4}
{Butter, Jam}	{T1, T8}
{Milk, Jam}	{T8}

Step 3: Recursive Processing and Creation of Larger Itemsets The algorithm proceeds by recursively combining existing itemsets (k-itemsets) in pairs. For each new combination, it calculates support by intersecting their tidsets. This recursive process continues until no more frequent itemsets can be found.

- $k = 3$

Table 6: Generation of 3-Itemsets with Corresponding Tidsets

Item	Tidset
{Bread, Butter, Milk}	{T8, T9}
{Bread, Butter, Jam}	{T1,T8,}

Step 4: Termination Condition The algorithm halts when it can no longer generate new itemsets that satisfy the minimum support threshold. At this point, all frequent itemsets have been identified.

Table 7: Generation of 4-Itemsets with Corresponding Tidsets

Item	Tidset
{Bread, Butter, Milk, Jam}	{T8}

- $k = 4$

We stop at $k = 4$ because there are no more item-tidset pairs to combine. Since the minimum support is 2, we conclude the following rules from the given dataset :

Table 8: The result

Items Bought	Recommended Products
{Bread}	{Butter}
{Bread}	{Milk}
{Bread}	{Jam}
{Butter}	{Milk}
{Butter}	{Coke}
{Butter}	{Jam}
{Bread and Butter}	{Milk}
{Bread and Butter}	{Jam}

II.4 Conclusion

Data mining plays a crucial role in extracting valuable insights from large datasets, facilitating data-driven decision-making across various domains. Among its diverse techniques, association rule mining stands out as a powerful method for uncovering meaningful patterns and relationships between data elements. By leveraging algorithms such as Apriori and FP-Growth, organizations can efficiently analyze transactional data to enhance business strategies, optimize operations, and improve customer experiences.

However, advancements in machine learning and artificial intelligence continue to refine these techniques, making them more efficient and scalable. As data continues to grow in volume and complexity, the significance of data mining in scientific research, healthcare, marketing, cybersecurity, and beyond will only expand.

Ultimately, data mining, and specifically association rule mining, serves as a cornerstone of modern analytics, helping businesses and researchers unlock hidden patterns, make informed decisions, and drive innovation in a data-centric world.

Chapter III:

Proposed Solution & Implementation

III.1 Introduction

The growing reliance on web services for accessing vast amounts of online content has underscored the importance of intelligent recommender systems capable of delivering relevant and personalized suggestions. However, a significant challenge remains: how to provide accurate recommendations in the absence of sufficient user data—particularly for first-time users or those unwilling to disclose personal information. This issue, known as the cold-start problem, continues to hinder the effectiveness of many recommender systems.

To address this, researchers have explored various techniques capable of inferring user preferences and patterns from limited or sparse data. Building on these advancements, the present study aims to enhance recommendation quality by leveraging association rule mining techniques on historical web service access data. This approach reduces manual effort, ensures consistency, and improves the overall efficiency of web service technologies for both consumers and providers.

To implement this, we apply well-established algorithms such as Apriori, ECLAT, FP-Growth, and AIS, and evaluate their applicability, efficiency, and scalability in processing diverse datasets. The experimentation was conducted on datasets composed of 100,000 transactions with varying numbers of services (7, 15, 20, and 28). The results reveal that each algorithm exhibits different performance characteristics depending on the dataset's structure and the support threshold used.

III.2 Related Work

2012 _ Min & Zhu

A fine-grained association rule mining approach was developed based on parametric rough set theory to address the cold-start problem commonly encountered in recommender systems. This technique demonstrated effectiveness in managing situations involving newly registered users or newly introduced items by identifying underlying patterns within complex many-to-many relationships, thereby enhancing recommendation accuracy in the absence of sufficient historical data. [56].

2017 _ Zhou & Hirasawa

A model was proposed that integrates Genetic Network Programming (GNP) with Ant Colony Optimization (ACO) to generate and evolve temporal association rules effectively. This hybrid method enables the recommender system to dynamically adjust to shifts in user interests and preferences over time. By leveraging GNP's capability to handle complex rule representations and ACO's strength in optimizing search paths, the system can continuously refine and improve recommendation quality. This adaptive approach enhances the system's ability to deliver more relevant and personalized suggestions, especially in environments where user behavior evolves frequently. [57].

2021 _ Benhamdi et al.

A recommender system based on Association Rule Mining (ARM) was developed to provide personalized learning materials. This system takes into account each learner's preferences, existing knowledge, and memory capacity in order to adapt the educational content to their specific needs. By analyzing these

factors, the system can deliver more relevant and effective learning resources, improving the overall learning experience and supporting better knowledge retention. [58].

2022 _ Srinadh V.

This study evaluated three major algorithms—Apriori, FP-Growth, and Eclat—for mining association rules. Using experimental datasets, the author compared these algorithms based on criteria such as processing time, memory consumption, and performance under varying support/confidence thresholds. The research concluded that FP-Growth outperformed Apriori in speed, while Eclat demonstrated high efficiency with smaller datasets.

2024 _ Nature Scientific Reports

This research proposed improvements to the traditional Apriori algorithm by integrating it with web log mining strategies. The enhanced algorithm was tested on datasets related to sports activity logs. The results revealed significant optimization in execution time and discovered previously hidden behavior patterns, demonstrating the power of combining domain-specific log mining with classical rule extraction.

2024 _ Annisa et al.

Developed a collaborative filtering recommendation system that combines clustering techniques with ARM. This hybrid approach aims to improve the accuracy and relevance of recommendations by grouping similar users and identifying frequent itemsets [59].

2025 _ Tandfonline

The authors explored how enhanced web usage mining can improve user experience across digital platforms. They incorporated association rule mining to track usage patterns and behaviors, aiming to personalize content delivery. Their findings confirmed that using pattern discovery techniques helped boost user satisfaction and interface efficiency, especially in high-traffic platforms.

2024 _ Mukesh Kumar & Dharminder Kumar The authors conducted an in-depth analysis of user navigation patterns on a government website by applying association rule mining and clustering techniques on web server logs. Their goal was to extract meaningful behavioral insights to enhance content delivery. The results showed that combining clustering with rule mining provided more accurate user segmentation and improved the recommendation quality.

III.3 Problematic & Objectives

Web services have emerged as a vital means for accessing and navigating the vast resources of the internet. Despite their utility, the immense volume of available web content presents a significant challenge in identifying specific services that align with users' needs. In response to this challenge, recommender systems have been developed to provide personalized suggestions by analyzing usage patterns and extracting meaningful insights from existing datasets. However, the accuracy of such systems is highly dependent on the availability of sufficient and relevant information; in the absence of adequate data, prediction capabilities are substantially diminished.

This research addresses the critical question of how existing web service data

can be utilized to identify patterns of user behavior and generate reliable recommendations for first-time users or individuals who are reluctant to share personal information. The primary objective of this study is to enhance the ability to infer user preferences and deliver accurate recommendations for relevant content, products, or services.

By employing association rule mining techniques, the proposed approach aims not only to support users in discovering pertinent resources but also to assist service providers in the efficient classification and organization of their offerings. This facilitates reduced manual effort and promotes consistency in categorization, even in multi-user environments. Ultimately, the integration of automated classification mechanisms will enable the system to match user queries with appropriate service categories, thereby improving the overall effectiveness of web service technologies for both providers and consumers.

III.4 Solution Overview

This study does not rely on users' personal profiles; indeed, no personal information, browsing history, cookies, or user identification is required. The essential requirement is a dataset containing users' access logs over a period of time. Utilizing association analysis techniques, the objective is to develop a predictive model capable of forecasting user interactions with existing web services. Historical access data serves as the foundation, while association rule mining algorithms are employed to uncover the underlying patterns necessary for accurate prediction.

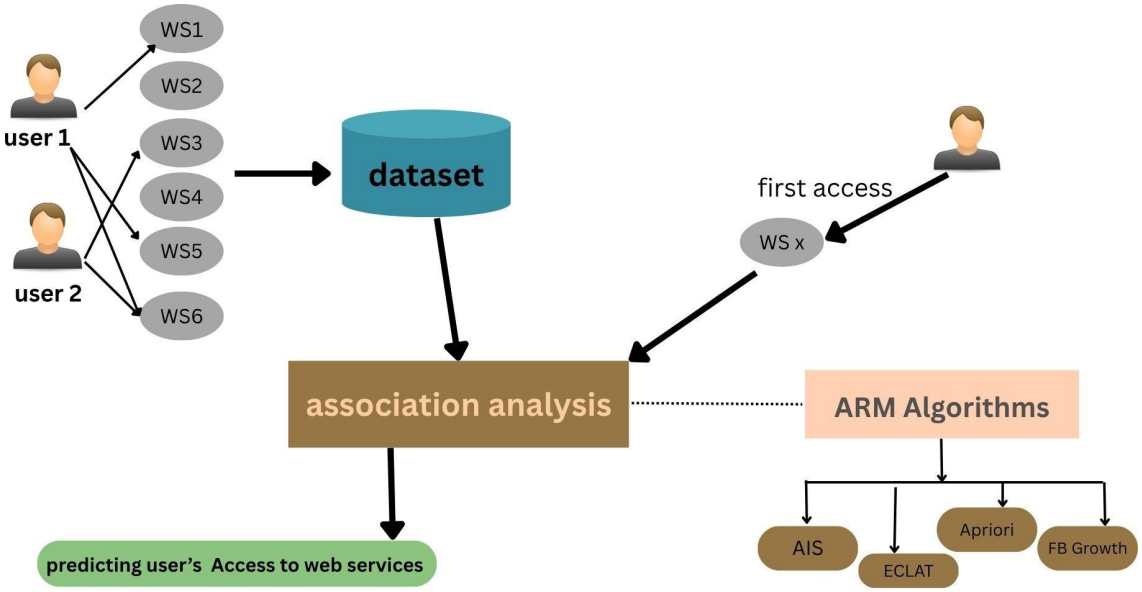


Figure 15: Framework of web service prediction

III.5 Implementation & Tests

This part presents a concise overview of how the applied algorithms were handled. Several techniques exist for extracting frequent patterns, and this study involves the application of AIS, Apriori, ECLAT, and FP-Growth to explore the addressed challenge. It also reflects on certain limitations associated with each method. These algorithms generally rely on evaluating specific measures for item combinations that could be considered significant. Among the key indicators used across the approaches are support and confidence, which serve as a basis for comparing their effectiveness.

III.5.1 Development Environment

To carry out the implementation, the project relies on Python — a widely adopted language within the domains of Data Science and Machine Learning. The selection of an appropriate programming environment plays a crucial role in ensuring the efficiency and flexibility of development processes. After evaluating various options, Python was identified as the most suitable tool to address the requirements of the current problem.

III.5.1.1 Materials

The implementation was carried out on a 64-bit Windows-based environment, utilizing a system equipped with an Intel(R) Core(TM) i6-6300U processor running at 2.40 GHz, alongside 8 GB of RAM. This setup provided a sufficient foundation for testing and executing the developed solution.

III.5.1.2 Programming Language

Python is widely recognized as one of the most versatile and accessible programming languages. Its clean and readable syntax, dynamic typing, and built-in support for object oriented programming make it ideal for both beginners and experienced developers. Python's interpreted nature allows for quick prototyping and testing, while its extensive ecosystem of libraries such as NumPy, Pandas, TensorFlow, and Flask enables development across domains like data analy- sis, machine learning, web development, and automation. These features make Python a strong candidate for implementing complex systems and research-based projects efficiently. [60]

III.5.1.3 Integrated Development Environment

For this project, we used Google Colaboratory (Colab), a cloud-based Integrated Development Environment (IDE) designed for Python programming. Colab allows users to write and run Python code directly from the browser without requiring any setup. It is particularly effective for tasks involving machine learning, data analysis, and educational purposes. Built on the Jupyter Notebook framework, Colab provides free access to computing resources, including CPUs, GPUs, and TPUs, making it a convenient and powerful platform for rapid development and experimentation.[61]

III.5.2 Data Description

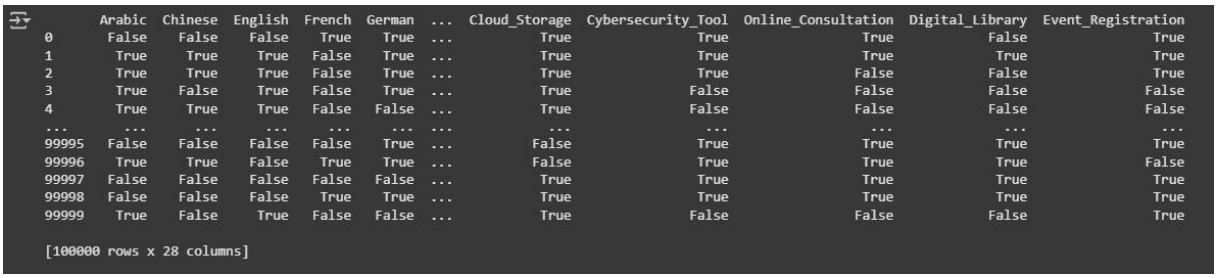
To better understand the algorithms employed and the significance of the dataset in building the prediction model, we utilized multiple datasets of varying sizes, as summarized in the table below. This approach allowed us to obtain results across a wide range of possible scenarios. It is worth noting that the dataset used in this study was not obtained from any external source. Instead, it was synthetically generated by one of the faculty members, based on simulated user access patterns. This allowed for full control over the dataset and enabled flexible evaluation of the algorithms performance.[62]

Users	**Web service**			
30k	7	15	20	28
50k	7	15	20	28
75k	7	15	20	28
100k	7	15	20	28

Table 9: Used datasets

Following the pre-processing phase, the datasets were saved in CSV format,

where each row corresponds to a user identified by a unique number starting from 1, and the columns represent the various web services accessed. Each cell thus indicates whether a specific user interacted with a particular service. To better understand how the applied algorithms function and how the dataset contributes to building the prediction model, multiple datasets of varying sizes were utilized. These datasets, outlined in the table above, were selected to ensure that the evaluation covers a wide range of possible scenarios.



	Arabic	Chinese	English	French	German	...	Cloud_Storage	Cybersecurity_Tool	Online_Consultation	Digital_Library	Event_Registration
0	False	False	False	True	True	...	True	True	True	False	True
1	True	True	True	False	True	...	True	True	True	True	True
2	True	True	True	False	True	...	True	True	False	False	True
3	True	False	True	False	True	...	True	False	False	False	False
4	True	True	True	False	False	...	True	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
99995	False	False	False	False	True	...	False	True	True	True	True
99996	True	True	False	True	True	...	False	True	True	True	False
99997	False	False	False	False	False	...	True	True	True	True	True
99998	False	False	False	True	True	...	True	True	True	True	True
99999	True	False	True	False	False	...	True	False	False	False	True

[100000 rows x 28 columns]

Figure 16: Execution time of the algorithms based on minimum support

III.5.3 Tests & Results

Comparison of Algorithm runtime based on minimum support variations

a. Fixed number of web services with varying number of users:

The initial analysis focuses on evaluating the impact of minimum support on runtime performance, given datasets with a constant number of services and a variable number of users. The following databases are generated with a diversity of items:

- Fixed number of web services: 7, 10,20, and 28.
- Varying number of users: 30 000; 50 000; 75 000 and 100 000.

The minimum support is varied in the range [0.1, 1] for each algorithm, and the corresponding runtime is measured.

- Starting with datasets of 7 web services.

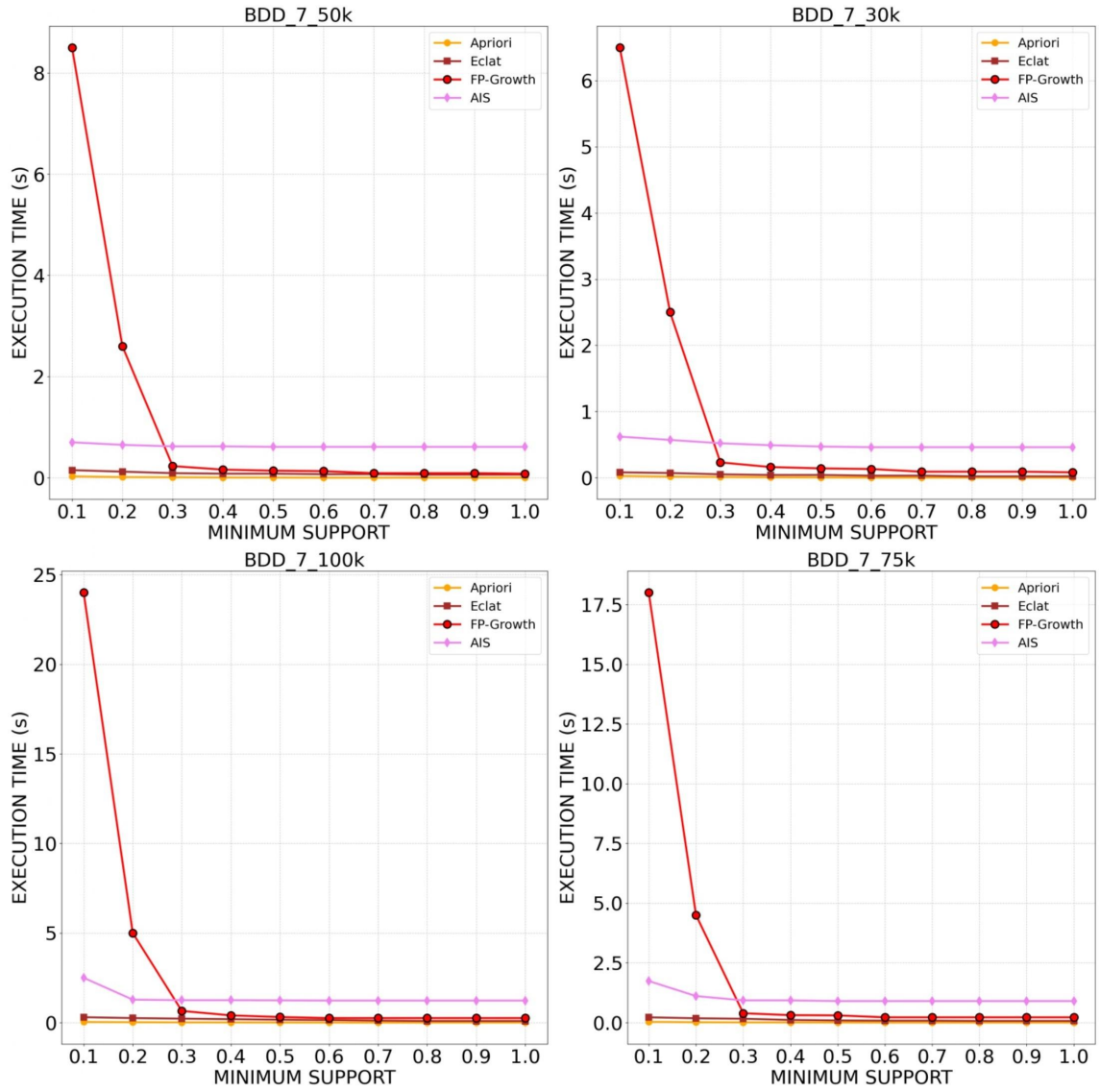


Figure 17: Runtime of algorithms on datasets with 7 web services

As the dataset size increases from 30K to 100K users, the execution time of all algorithms also increases, especially with low minimum support values. FP-Growth shows the highest execution time, reaching 24 seconds at 100K users and 0.1 support, while Apriori remains the fastest with execution times under

0.04 seconds in all cases. AIS performs better than FP-Growth but is slower than Apriori and Eclat. Eclat shows stable results with moderate times, increasing gradually with dataset size. For all algorithms, the execution time decreases as the minimum support increases, and becomes nearly constant above 0.3. Apriori and Eclat are the most efficient and scalable for larger datasets, while FP-Growth is more sensitive to low support thresholds.

- The next set of datasets is with 15 web services.

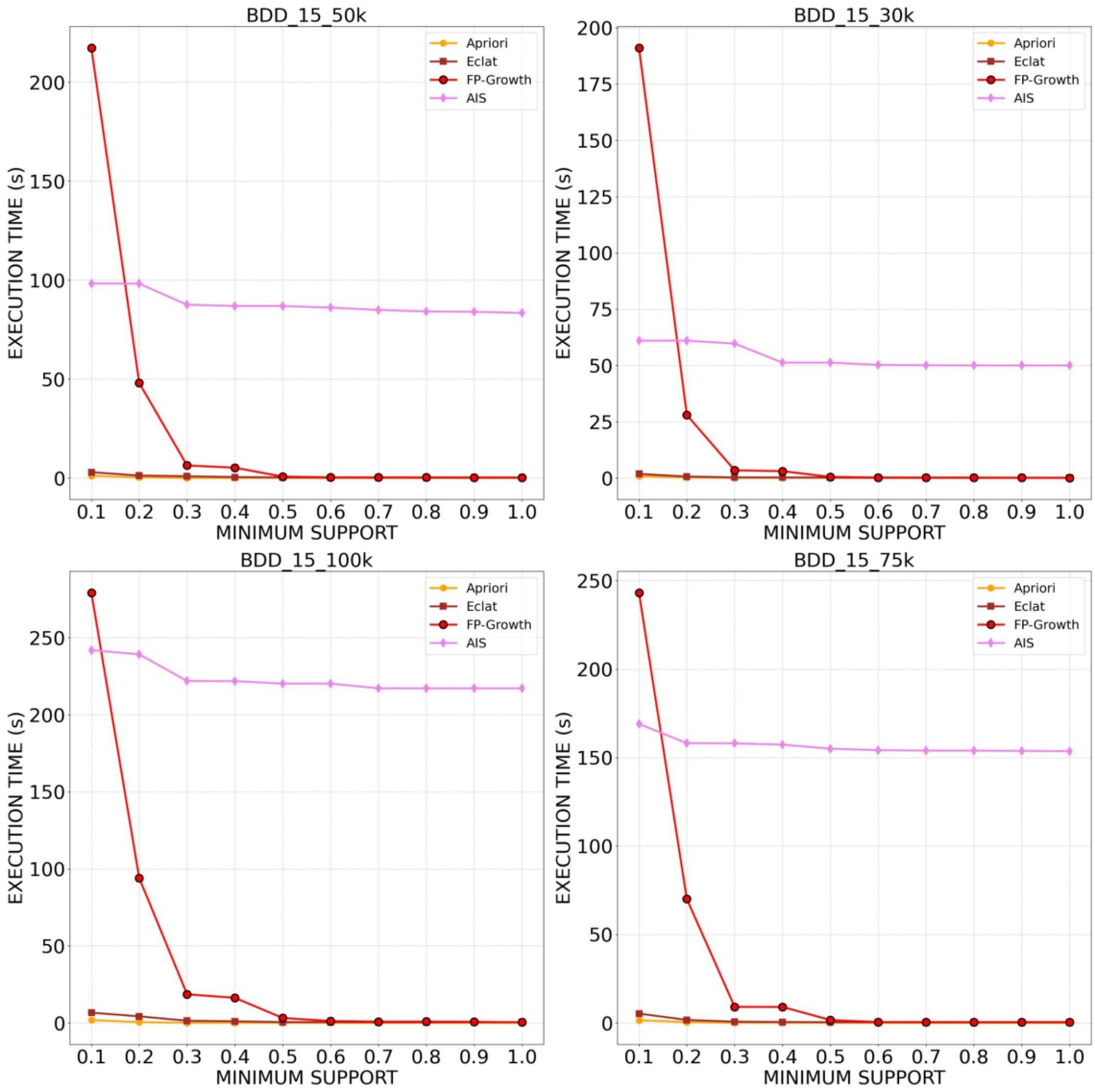


Figure 18: Runtime of algorithms on datasets with 15 web services

When we fix the number of services to 15 and increase the number of users (from 30K to 100K), we notice that the execution time of all algorithms goes up, but with different behaviors.

Apriori remains the fastest overall. It starts strong and becomes even faster when the support increases. Eclat also gives good results and stays stable, but it is usually a bit slower than Apriori, especially with more users.

FP-Growth is very slow when the support is low, especially with large datasets, but it becomes much faster as support increases. So, it's not bad, but it needs a high support value to perform well.

AIS is always the slowest. Its execution time is high in all cases and doesn't improve much with higher support. This makes it the least efficient, especially when the dataset is large.

In conclusion, Apriori and Eclat are the best choices for large data. FP-Growth is only good when support is high. AIS is not recommended because it is too slow.

- The next set of datasets is with 20 web services.

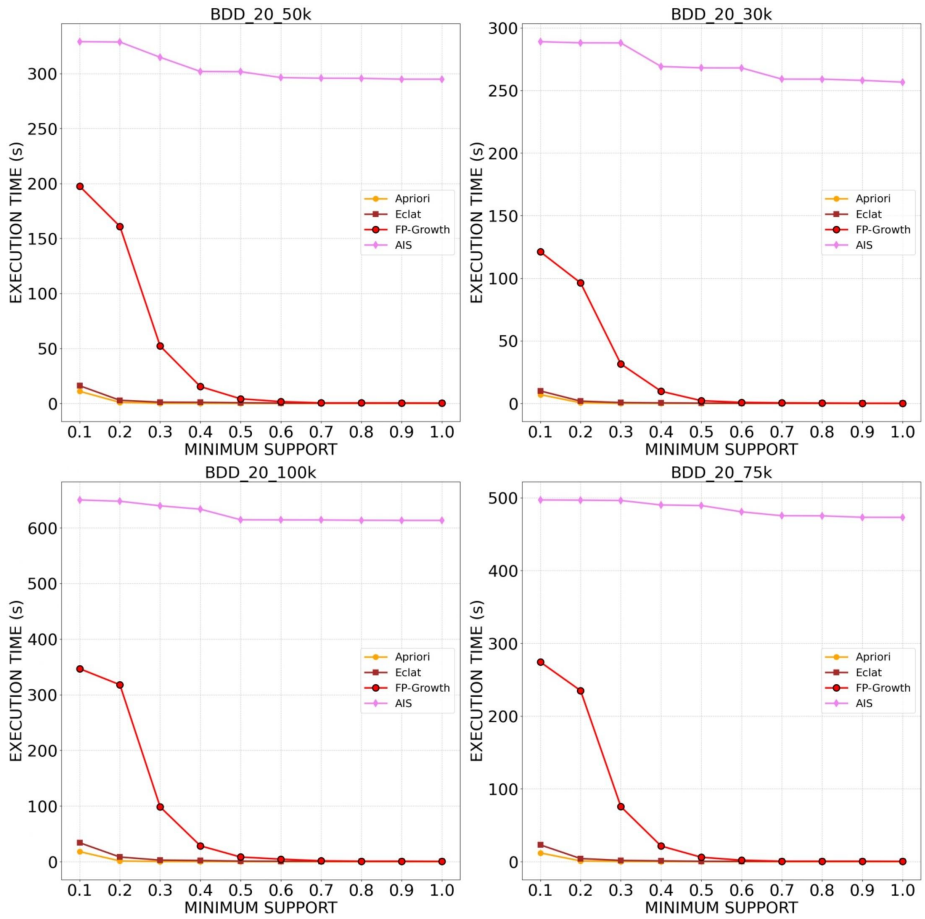


Figure 19: Runtime of algorithms on datasets with 20 web services

In the case of 20 services, we observe that all algorithms perform slower at lower support values, especially AIS, which consistently shows the highest execution time across all scenarios. As the support threshold increases, execution times drop significantly—particularly for Apriori and ECLAT, which become extremely fast and nearly reach zero at support values of 0.7 and above. FP-Growth improves progressively but does not outperform Apriori or ECLAT. AIS, on the other hand, remains slow even at higher support levels, making it the least efficient among the four algorithms.

- The last set of data is with the largest number of web services 28

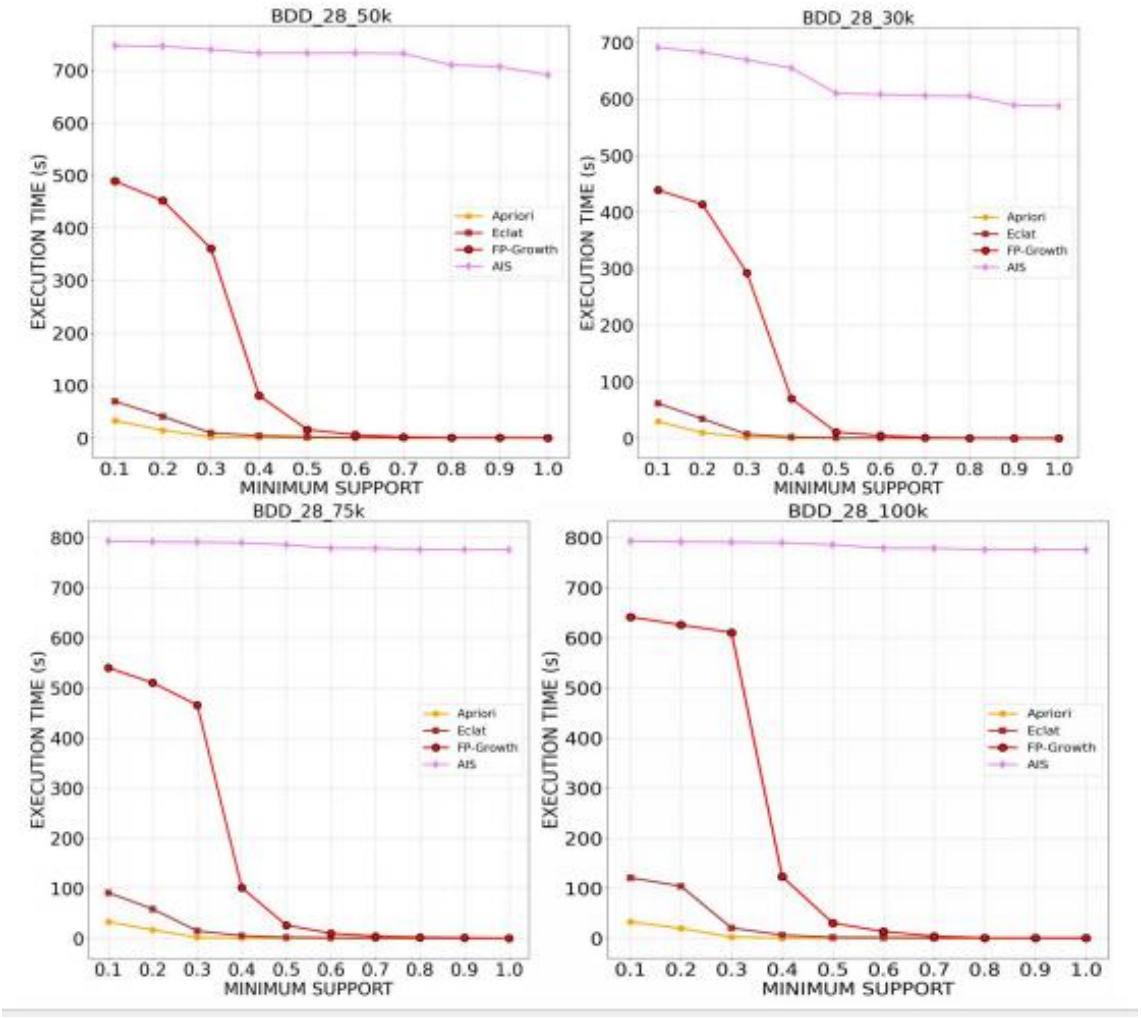


Figure 20: Runtime of algorithms on datasets with 28 web services

When the number of services rises to 28, all four algorithms still speed up as the minimum support threshold grows, but the absolute runtimes escalate sharply at low supports: with support = 0.1, Apriori finishes in roughly 30–33s, Eclat doubles that at ≈60–120 s, FPGrowth leaps to ≈440–640s, and AIS is the slowest at ≈690–793s. As the support passes0.3 the curves collapse Apriori drops below a second, Eclat settles around half a second, FP Growth falls to about one second, and AIS remains stuck in the tenminute range. Beyond a support of0.7 every curve plateaus: Apriori converges to nearzero time, Eclat hovers at ≈0.5s, FP

Growth at ≈1s, and AIS at ≈12–13minutes. Thus, even with the larger itemset, Apriori remains the fastest overall, followed by Eclat; FPGrowth is only competitive once the support is high, while AIS stays consistently the slowest.

a. Fixed number of users with varying number of web services:

The initial analysis focuses on evaluating the impact of minimum support on runtime performance, given datasets with a constant number of users and a variable number of web services. The following databases are generated with diverse itemsets:

- Fixed number of users: 30,000; 50,000; 75,000; and 100,000.
- Varying number of web services: 7, 10, 20, and 28.

qThe minimum support is varied in the range [0.1, 1] for each algorithm, and the corresponding runtime is measured.

- Starting with datasets of 30K users.

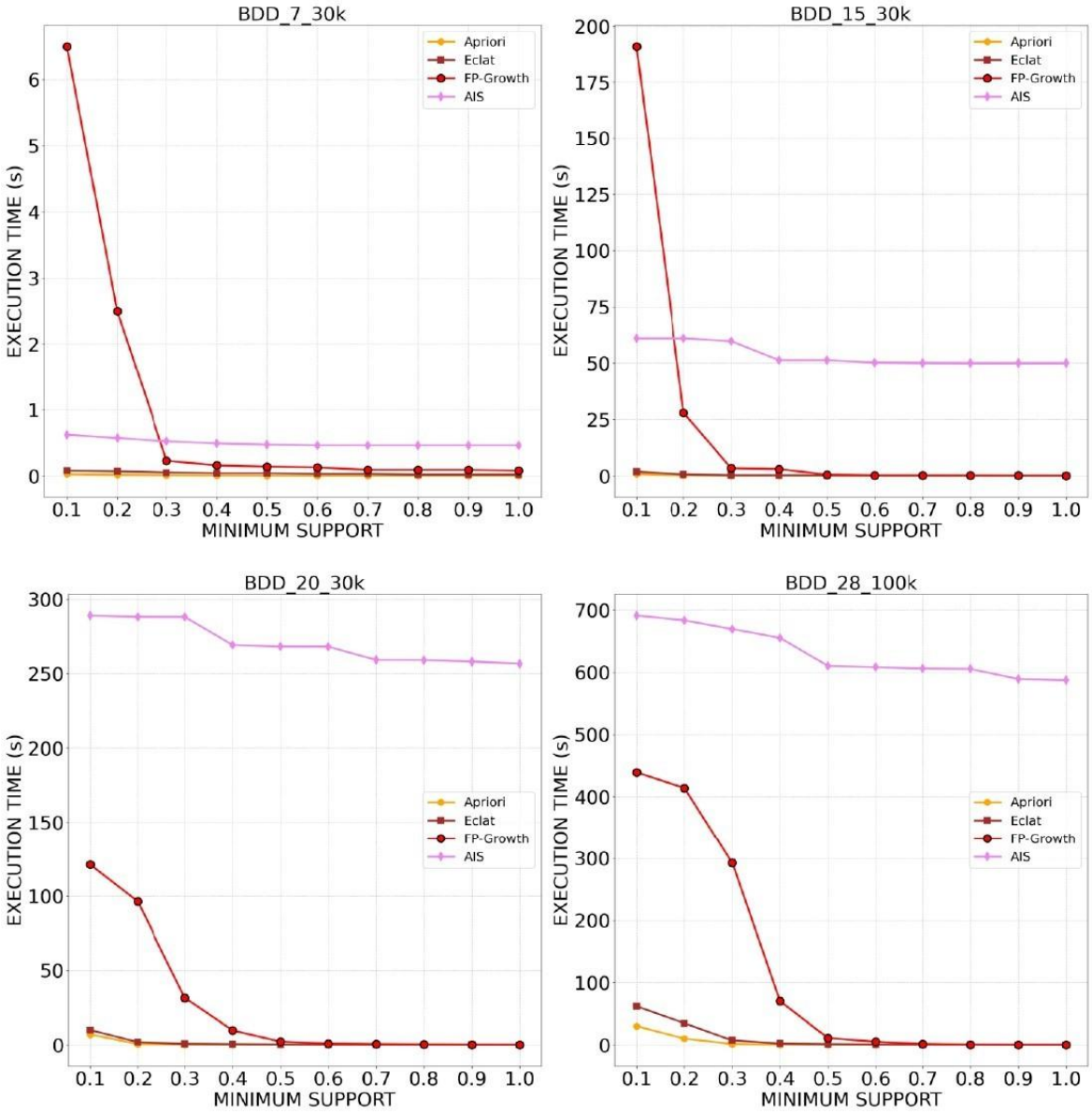


Figure 21: Runtime with minimum support on Data with 30k

When keeping the number of users fixed at 30,000 and increasing the number of web services from 7 to 28, we notice that all four algorithms show a rise in execution time, but at different levels. Apriori starts very fast with small service sets but becomes much slower as the number of services increases, reaching up to 30 seconds for 28 services. Eclat also shows an increase in time but remains more stable than Apriori and performs better at higher support thresholds. FP-Growth, on the other hand, becomes extremely slow when the number of ser-

vices is high and the support is low, taking over 400 seconds in some cases, although it improves when support increases. AIS consistently shows the highest execution time and does not scale well; even with more support or fewer services, it stays slow. Overall, increasing the number of services clearly impacts all algorithms, with Apriori and Eclat remaining more efficient compared to FP-Growth and AIS in such conditions.

- Starting with datasets of 50K users.

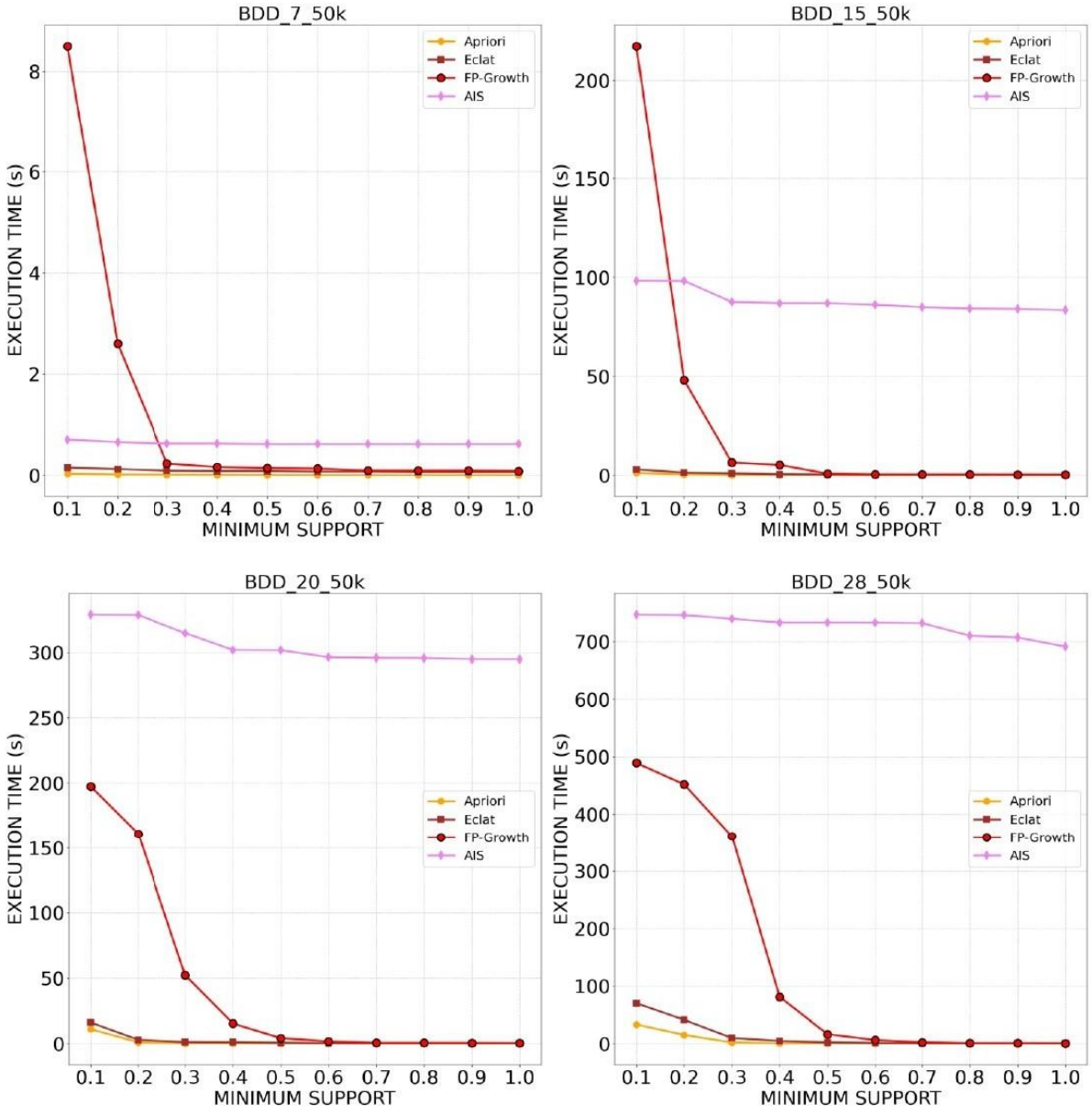


Figure 22: Runtime with minimum support on Data with 50k

As the number of services increases from 7 to 28 (with users fixed at 50,000), the execution times of all algorithms grow, but at very different rates. Apriori performs extremely well with fewer services (7 or 15), producing results in milliseconds, and remains efficient even as the number of services increases, though its time increases noticeably with 28 services. Eclat is slower than Apriori for smaller datasets, but it scales relatively well and maintains reasonable performance up to 20 services. However, its runtime becomes significantly longer at 28 services. FP-Growth, while fast at high support levels, suffers from very large runtimes at lower support thresholds, especially when the number of services is high. It becomes impractical with 28 services unless the support is high. AIS, on the other hand, shows consistently poor performance. Its execution time is significantly higher than all other algorithms in every setting, and it worsens drastically with more services, making it the least efficient option. Overall, Apriori is the best performer for smaller to medium datasets, while Eclat is suitable for moderate complexity, and FP-Growth is only viable with high support or when fewer services are used. AIS should be avoided due to its consistently high cost.

- Starting with datasets of 75K users.

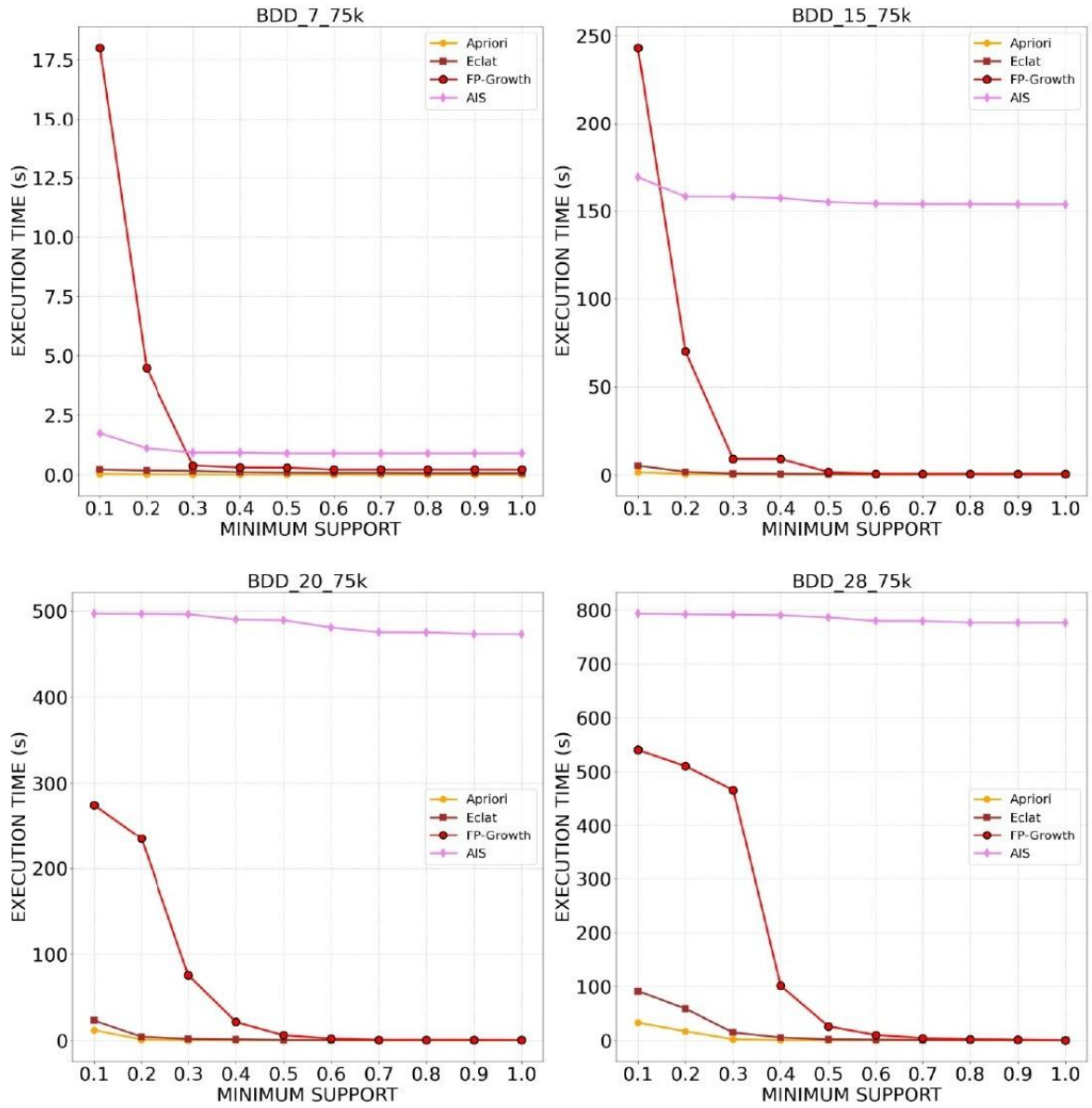


Figure 23: Runtime with minimum support on Data with 75k

With 75 000 users, raising the number of services from 7 to 28 amplifies the contrasts among the four algorithms. Apriori stays quickest overall: with seven services it finishes in a few milliseconds and—even at 28 services drops from 33 s at support 0.1 to essentially zero once support exceeds 0.6. Eclat is a steady second; it is only a fraction of a second with seven services, rises to about a minute at the lowest support with 28 services, and then falls below one second at higher supports. FP-Growth struggles whenever both the service count is large

and support is low, soaring past five minutes at 28 services and support 0.1, yet, like Apriori, it accelerates sharply as the support threshold climbs, sliding under a second when support ≥ 0.9. AIS remains by far the slowest: it already needs almost two seconds for seven services and balloons to more than thirteen minutes for 28 services, showing little improvement as support grows. In short, for 75 k users, Apriori and Eclat remain the practical choices, FP-Growth is viable only when high support values are acceptable, and AIS is unsuitable for time critical tasks.

- Starting with datasets of 100K users.

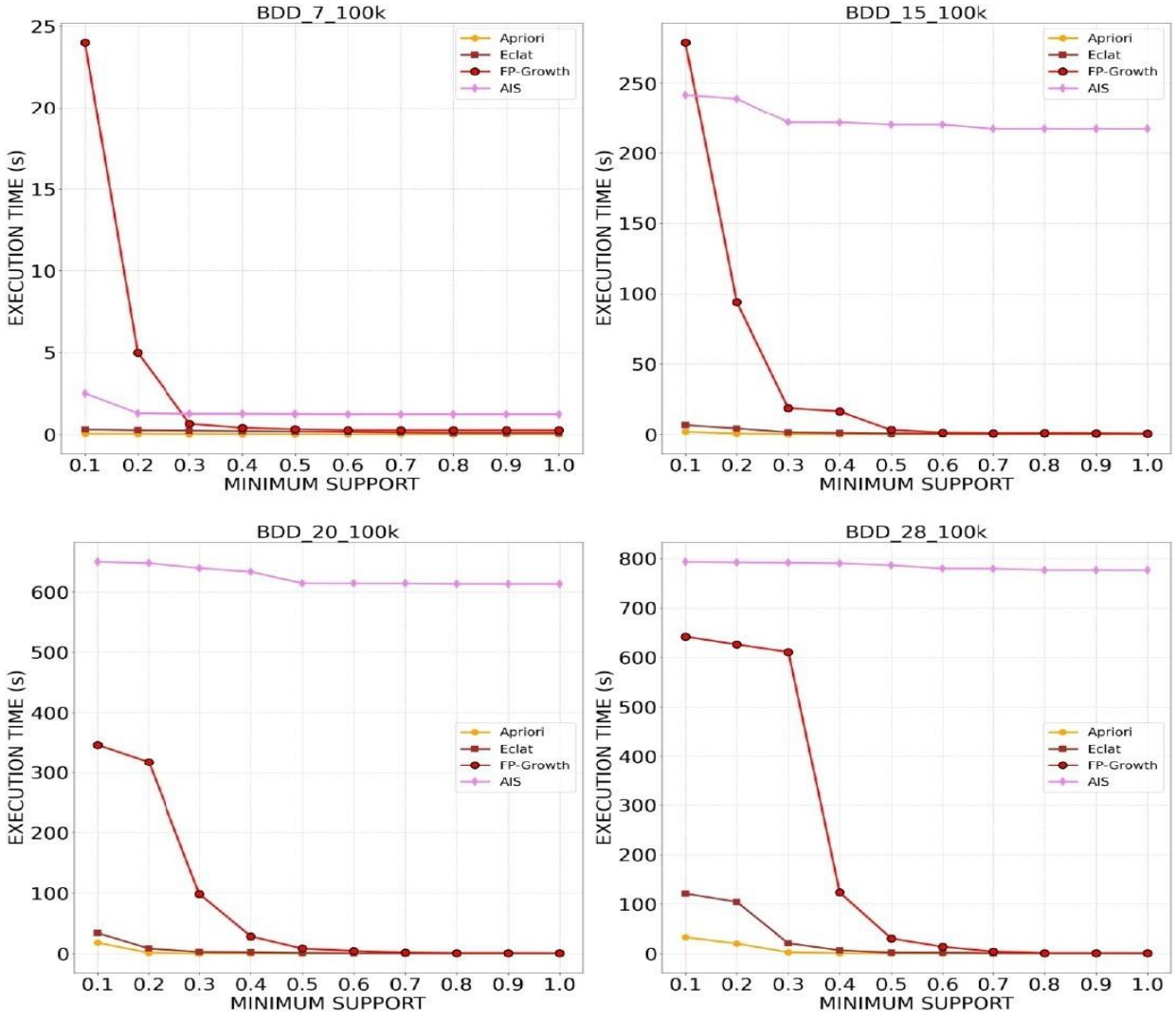


Figure 24: Runtime with minimum support on Data with

For the 100 000-user dataset, the four algorithms show a clear performance hierarchy across all service counts (7, 15, 20, 28) and support levels. Apriori is consistently fastest, peaking at only 33 s in the worst case (28 services, 0.1 support) and dropping to near-zero time once the support reaches 0.6. Eclat ranks second: it remains sub-second for 7 services and, even at 28 services, falls from 121 s at very low support to under one second when support ≥ 0.8. FP-Growth is a distant third; its heavy tree-building cost makes it extremely slow at low

supports but it accelerates sharply when support is high, eventually matching Eclat only above 0.8. AIS is decisively the slowest, taking several minutes in every configuration and showing little improvement as support increases.

Runtime with different minimum confidence values :

To compare the performance of the three algorithms in generating association rules, we used a dataset containing 20 web services and 100,000 transactions, with a minimum support threshold set to 0.6.

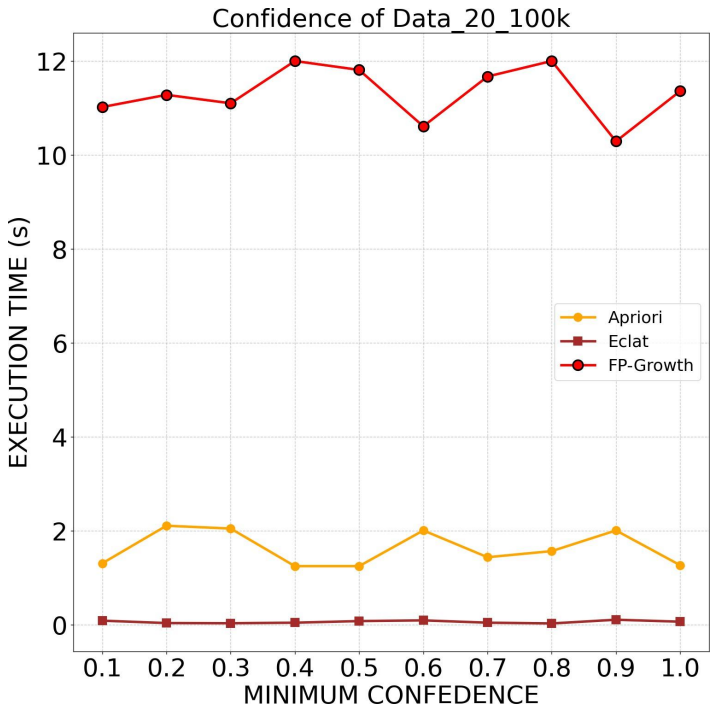


Figure 25: Runtime of different minimum confidence values

From the figure, it is clear that the average execution times of Apriori, FP-Growth, and ECLAT remain relatively stable across different minimum confidence thresholds. As expected, the differences are minimal since all three algorithms use the same evaluation metric confidence to generate the rules. However, FP-Growth shows consistently higher execution times compared to Apriori and ECLAT, which are both significantly faster, especially ECLAT, whose exe-

cution time is nearly negligible throughout. This confirms that while confidence influences rule filtering, it has little impact on performance variation among the algorithms.

III.5.4 Implementation

As shown in the experimental results, each algorithm responds differently depending on the dataset used.

Algorithm performance:

A comparative analysis of the four association rule mining algorithms Apriori, Eclat, FP-Growth, and AIS was conducted using datasets of 100,000 transactions and varying numbers of services (7, 15, 20, and 28). The results clearly show significant differences in performance under different configurations of minimum support and itemset size.

(a) **Apriori:** Apriori consistently performed well on datasets with smaller itemsets (≤15 services), achieving very fast runtimes (e.g., 0.008 s with 7 services and 0.4 support). Its performance starts to degrade as the number of services increases due to the exponential candidate generation. For instance, with 28 services and 0.1 support, runtime peaked at 33 seconds, and it consumes a large amount of memory. It is simple to implement and works well with low-dimensional data, but slows down significantly with more services and lower support thresholds.

(b) **Eclat:** Eclat showed more stable scalability than Apriori for large itemsets due to its vertical database layout and set intersection operations. However, it is still considerably slower in almost all scenarios. For instance, with 7

services and 0.1 support, it took 0.30 seconds, while Apriori needed only 0.032 seconds. The gap widens with more services: at 28 services and 0.1 support, Eclat reached 121 seconds, nearly 4× slower than Apriori. Although Eclat performs better than Apriori on high-dimensional data, the intersection of long TID lists is computationally heavy, resulting in slower runtimes.

(c) **FP-Growth:** theoretically the most efficient algorithm due to its tree-based pattern mining, gave mixed results. On datasets with high support thresholds, it showed competitive performance (e.g., only 0.25 seconds with 7 services and 0.6 support). However, when support was low and the number of services increased, runtime exploded. With 28 services and 0.1 support, it reached 641 seconds—the slowest among all algorithms tested. This is attributed to the complexity of FP-Tree construction and significant memory limitations at scale. While it performs well with high support values or large datasets with few distinct items, it suffers from heavy memory usage and severe slowdowns in low-support, high-dimensional scenarios.

(c) **AIS:**AIS turned out to be the least efficient of all tested algorithms in terms of runtime. It showed minimal sensitivity to changes in the support threshold, hovering around 600–790 seconds regardless of configuration. This behavior is expected due to its single-pass candidate generation and counting, which leads to redundant computations and poor scalability. Although conceptually simple and requiring no multiple passes over the data, AIS is inherently inefficient, with consistently long runtimes that make it unsuitable for large-scale datasets.

III.5.5 Test Run :

The dataset used in this example is consisting of 100k users (rows) and 28 distinct web services (columns).

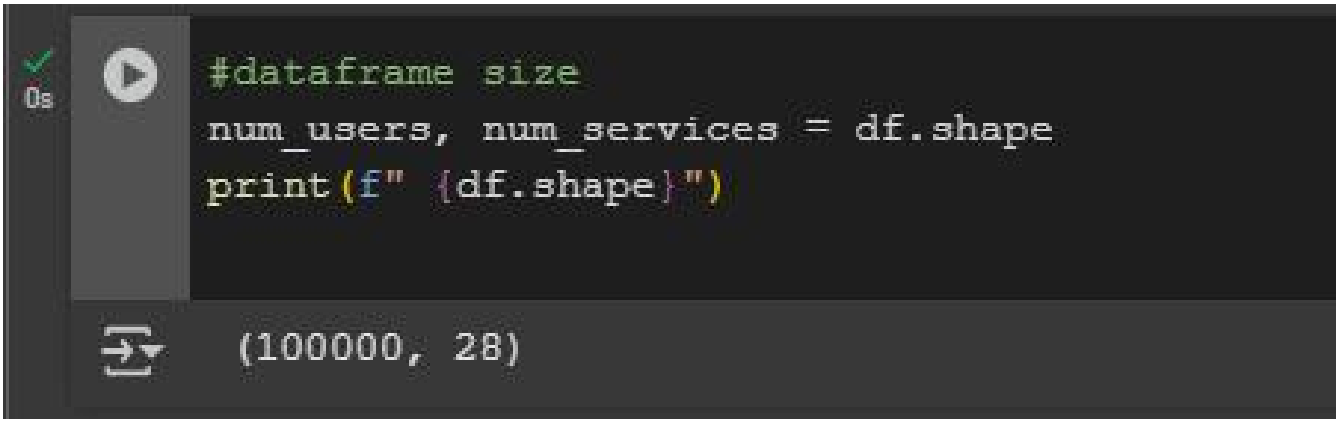


Figure 26: Dataset Size Details

The dataset contains binary cell values, where ”True” indicates that a user has accessed a specific web service, while ”False” indicates otherwise. Given the size of the dataset, the Apriori algorithm was identified as the most suitable for implementation in this case. Therefore, the example utilizes the Apriori function provided by the mlxtend library to extract frequent itemsets based on a defined minimum support threshold. This function outputs results in a clear dataframe format with two columns: one for the frequent itemsets and the other for their corresponding support values.

Typically, the optimal support threshold ranges between 60% and 70% of the total transactions. However, as the threshold increases, the number of frequent itemsets tends to decrease. In the case of large datasets, it becomes necessary to lower the support threshold to obtain meaningful patterns. Based on prior evaluations, a minimum support value of 0.55 (or 55%) was found to yield the most suitable results.

‘

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
# Generate frequent itemsets using Apriori
frequent_itemsets = apriori(df, min_support=0.55, use_colnames=True)
print(frequent_itemsets)
```

```
    support                                           itemsets
0   0.93843                                             (News)
1   0.96877                               (E_Learning_Platform)
2   0.60840                                    (Online_Banking)
3   0.63012                                           (Health)
4   0.63937                                        (Job_Search)
..      ...                                                ...
73  0.55997  (E_Learning_Platform, Government, News, Cybers...
74  0.60327  (E_Learning_Platform, Cloud_Storage, Academic_...
75  0.59298  (E_Learning_Platform, Academic_Platform, News,...
76  0.57245  (E_Learning_Platform, Cloud_Storage, News, Cyb...
77  0.55950  (Cloud_Storage, Government, Academic_Platform,...

[78 rows x 2 columns]
```

Figure 27: Frequent Itemsets Generated by the Apriori Algorithm

- Next, we proceed to generate the association rules from the identified frequent itemsets, using the confidence metric provided by the same library, with a threshold set at 0.7 (or 70%)

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules

# Generate frequent itemsets
frequent_itemsets = apriori(df, min_support=0.55, use_colnames=True)
# Generate association rules with confidence threshold of 0.7
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)

len(rules)
```

```
259
```

Figure 28: Association Rules Generation Code

- result of Association Rules Generated Using ARM function



```
=== Association Rules  ===
```

	Antecedents	Consequents	Support	Confidence	Lift
0	E_Learning_Platform	News	0.91	0.94	1.00
1	News	E_Learning_Platform	0.91	0.97	1.00
2	Online_Banking	News	0.57	0.94	1.00
3	Health	News	0.61	0.97	1.04
4	Job_Search	News	0.60	0.93	0.99
5	Social_Community	News	0.62	0.97	1.04
6	Government	News	0.75	0.93	1.00
7	News	Government	0.75	0.80	1.00
8	Academic_Platform	News	0.72	0.95	1.01
9	News	Academic_Platform	0.72	0.77	1.01
10	Cloud_Storage	News	0.78	0.95	1.02
11	News	Cloud_Storage	0.78	0.83	1.02
12	News	Cybersecurity_Tool	0.72	0.77	1.00
13	Cybersecurity_Tool	News	0.72	0.94	1.00
14	Online_Consultation	News	0.60	0.93	0.99
15	Digital_Library	News	0.56	0.94	1.00
16	Online_Banking	E_Learning_Platform	0.60	0.99	1.02
17	Health	E_Learning_Platform	0.61	0.97	1.00
18	Job_Search	E_Learning_Platform	0.62	0.97	1.00
19	Social_Community	E_Learning_Platform	0.62	0.97	1.00
20	Government	E_Learning_Platform	0.79	0.98	1.01

Figure 29: Association Rules Extracted Using the ARM Function

Interface test:

The user interface of this project was built using Streamlit, a Python library that allows quick creation of interactive web apps with minimal code. Streamlit enables us to create a responsive sidebar for user input, display dynamic text and data, and customize the page style easily.

For this application, we reused the same dataset from before, loaded from the CSV file , which contains records of users and the web services they used. The data was preprocessed to keep only relevant service usage columns in boolean format.

The interface lets users select the services they have already used via a sidebar multiselect widget. Based on this input, the backend runs the Apriori algorithm to mine association rules and generate personalized recommendations.

The results are displayed clearly on the main page, showing suggested rules and final recommended services. Additionally, some CSS styling was applied to improve the visual look and feel of the app.

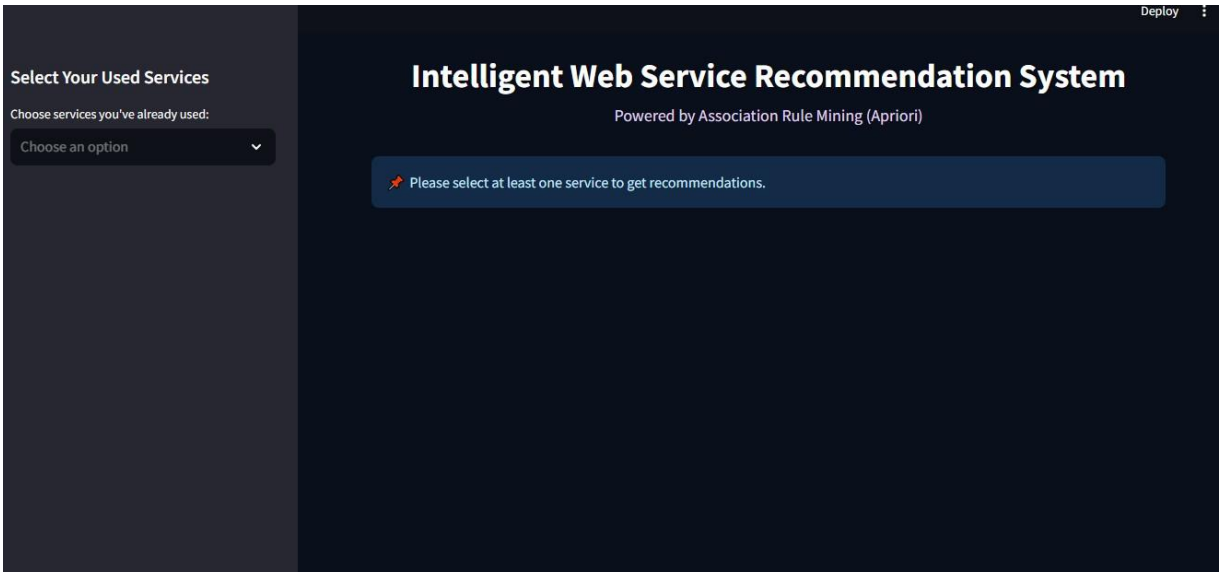


Figure 30: Interactive Dashboard of Web Service Recommendations

After selecting your used services, the system processes the input to generate association rules and provides personalized recommendations based on these rules :

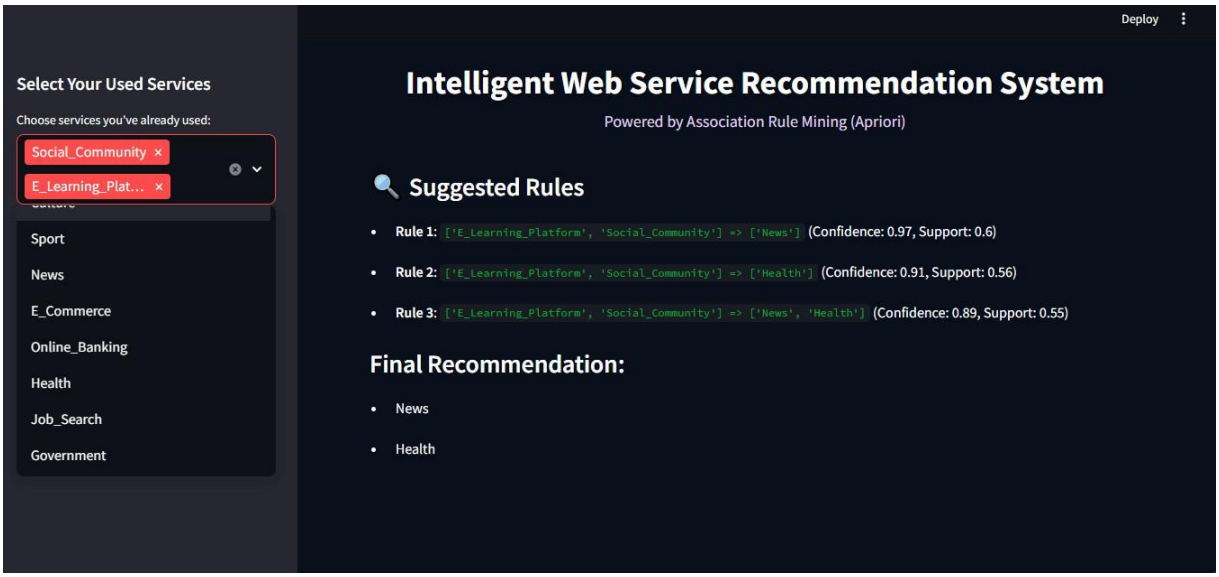


Figure 31: Service Selection Sidebar in Recommendation System

III.6 Conclusion

This study explored the effectiveness of various association rule mining algorithms Apriori, Eclat, FP-Growth, and AIS in addressing the cold-start problem and improving the accuracy of recommender systems based on web service usage data. By leveraging historical access logs and applying frequent pattern mining techniques, the research demonstrated that meaningful user behavior patterns can be extracted even in the absence of rich personal data. Through extensive testing across different dataset sizes, item counts, and support thresholds, Apriori and Eclat emerged as the most reliable and efficient algorithms for generating recommendations in large-scale environments. Apriori consistently delivered fast and accurate results in low-dimensional data scenarios, while Eclat provided better scalability for more complex datasets. FP-Growth showed promise at high support levels but suffered from memory overhead and instability with larger itemsets and lower thresholds. AIS, although conceptually simple, was found to be inefficient and unsuitable for high-volume applications due to its excessive runtime and poor scalability. The experimental findings reinforce the importance of algorithm selection based on the characteristics of the data and system constraints. Moreover, the integration of automated classification mechanisms using association rules offers significant potential for enhancing the usability and responsiveness of web service platforms, particularly for new or anonymous users. Future work may focus on hybrid approaches that combine multiple techniques or incorporate contextual and semantic information to further improve recommendation quality and adaptability.

General Conclusion

Frequent pattern mining is widely recognized for its broad range of applications, including software bug detection, data clustering, classification, and many other analytical domains. While this study takes a different direction by focusing on the prediction of online web service access, frequent pattern mining remains a foundational technique for uncovering meaningful relationships within large datasets, supporting various data-driven decision-making processes.

The primary objective of this work is to demonstrate the practical implementation and comparative analysis of four well-known algorithms in frequent pattern mining: Apriori, FP-Growth, AIS, and ECLAT. Using multiple real-world datasets, we evaluated the performance, strengths, and limitations of each algorithm in the generation of association rules.

These algorithms have evolved significantly over time—transitioning from tools primarily used for data exploration to essential components of modern predictive analytics. Given that frequent itemset mining continues to be an active research area, ongoing innovation and optimization are critical, especially in applications where performance, scalability, and accuracy are of paramount importance.

Our findings confirm that association rule mining can effectively power recommendation systems, even in scenarios with limited or no explicit user profile data. By identifying co-occurrence patterns in user interactions, the system

can make relevant predictions about future service accesses. However, several limitations persist, such as the inability of traditional algorithms to account for transaction weights, item frequency variations, or duplicate items within transactions—factors that can reduce their effectiveness in real-world business environments.

Looking ahead, future research could enhance these approaches by integrating clustering techniques, improving scalability for large-scale databases, and incorporating weighted or repeated transactions into the mining process. Such advancements would not only improve the accuracy of recommendations but also broaden the practical applicability of association rule-based predictive systems.

Bibliography

[1] World Wide Web Consortium. *W3C*. Accessed: 2025-05-11. 2024. URL: https://www.w3.org.

[2] Advicescout Editorial Team. *Web Services Standards: A Comprehensive Overview*. https://www.advicescout.com/web-service-standards-and-protocols/. Accessed: April 24, 2025. 2023.

[3] Abhishek Ghosh. *Web Services: What Are Web Services?* Consulté le 9 mai 2025. 2013. URL: https://thecustomizewindows.com/2013/08/web-services-what-are-web-services/.

[4] Xinyu Zhang, Wenjun Li **and** Hui Wang. "SOAP and REST in web service communication: A comparative study". **in***International Journal of Web Information Systems*: 16.4 (2020), **pages** 429–448. DOI: 10.1108/IJWIS-01-2020-0002.

[5] E Thirumal **and** E Duraisamy. "WSDL-based service composition: A survey and classification". **in***Journal of King Saud University - Computer and Information Sciences*: 33.4 (2021), **pages** 447–455. DOI: 10.1016/j.jksuci.2018.03.014.

[6] Wesam Alghabban, Bilal Qureshi **and** Sami Bashir. "Service discovery mechanisms for Web services: A comparative analysis". **in***Journal of Systems and Software*: 152 (2019), **pages** 123–137. DOI: 10.1016/j.jss.2019.02.007.

[7] Yaroslav Petrashko, Alexander Grigoriev **and** Sergey Kuznetsov. "A survey on service-oriented architecture and its applications in cloud computing". **in***International Journal of Cloud Computing*: 10.2/3 (2021), **pages** 165–182. DOI: 10.1504/IJCC.2021.116982.

[8] Paulo Merson, Rick Kotermanski **and** Phil Bianco. *Evaluating a Service-Oriented Architecture*. https://www.researchgate.net/figure/SOA-and-SOA-Technologies_fig2_237571793. 2010.

[9] Shuai Zhang, Yujie Zhao **and** Aixin Sun. “A Brief History of Recommender Systems”. **in***arXiv preprint arXiv:2209.01860*: (2022). URL: https://arxiv.org/abs/2209.01860.

[10] Francesco Ricci, Lior Rokach **and** Bracha Shapira. “Introduction to Recommender Systems Handbook”. **in***Recommender Systems Handbook*: **byeditor**Francesco Ricci **andothers**. Boston, MA: Springer US, 2011, **pages** 1–35. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_1. URL: https://doi.org/10.1007/978-0-387-85820-3_1.

[11] Meng-Lun Wu, Chia-Hui Chang **and** Rui-Zhe Liu. “Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices”. **in***Expert Systems with Applications*: 41.6 (2014), **pages** 2754–2761. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.10.008. URL: https://www.sciencedirect.com/science/article/pii/S0957417413008130.

[12] Hyeyoung Ko **andothers**. “A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields”. **in***Electronics*: 11.1 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11010141. URL: https://www.mdpi.com/2079-9292/11/1/141.

[13] Sung-Hyuk Park **and** Sang Pil Han. “Empirical analysis of the impact of product diversity on long-term performance of recommender systems”. **in***Proceedings of the 14th Annual International Conference on Electronic Commerce*: 2012, **pages** 280–281.

[14] Mehrbakhsh Nilashi, Othman Ibrahim **and** Karamollah Bagherifard. “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques”. **in***Expert Systems with Applications*: 92 (2018), **pages** 507–520.

[15] Benjamin Gras, Armelle Brun **and** Anne Boyer. “Identifying grey sheep users in collaborative filtering: a distribution-based technique”. **in***Proceedings of the 2016 conference on user modeling adaptation and personalization*: 2016, **pages** 17–26.

[16] Yue Shi, Martha Larson **and** Alan Hanjalic. “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges”. **in***ACM Computing Surveys (CSUR)*: 47.1 (2014), **pages** 1–45.

[17] Huan Zhao **andothers**. “Side Information Fusion for Recommender Systems over Heterogeneous Information Network”. **in***ACM Trans. Knowl. Discov. Data*: 15.4 (**june** 2021). ISSN: 1556-4681. DOI: 10.1145/3441446. URL: https://doi.org/10.1145/3441446.

[18] Zeshan Fayyaz **andothers**. “Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities”. **in***Applied Sciences*: 10.21 (2020). ISSN: 2076-3417. DOI: 10.3390/app10217748. URL: https://www.mdpi.com/2076-3417/10/21/7748.

[19] Yingbo Liu **andothers**. “Research on the Matthews correlation coefficients metrics of personalized recommendation algorithm evaluation”. **in***International Journal of Hybrid Information Technology*: 8.1 (2015), **pages** 163–172.

[20] Denis Parra **and** Shaghayegh Sahebi. “Recommender systems: Sources of knowledge and evaluation metrics”. **in***Advanced Techniques in Web Intelligence-2: Web User Browsing Behaviour and Preference Analysis*: Springer, 2013, **pages** 149–175.

[21] Manami Kawasaki **and** Takashi Hasuike. “A recommendation system by collaborative filtering including information and characteristics on users and items”. **in***2017 ieee symposium series on computational intelligence (ssci)*: IEEE. 2017, **pages** 1–8.

[22] Carlos A Gomez-Uribe **and** Neil Hunt. “The netflix recommender system: Algorithms, business value, and innovation”. **in***ACM Transactions on Management Information Systems (TMIS)*: 6.4 (2015), **pages** 1–19.

[23] Jianming He **and** Wesley W Chu. “A social network-based recommender system (SNRS)”. **in***Data mining for social network data*: Springer, 2010, **pages** 47–74.

[24] Kraisak Kesorn, Wanarat Juraphanthong **and** A Salaiwarakul. “Personalized attraction recommendation system for tourists through check-in data”. **in***IEEE access*: 5 (2017), **pages** 26703–26721.

[25] Bhavna Galhotra **and** Ayushi Dewan. “Impact of COVID-19 on digital platforms and change in E-commerce shopping trends”. **in***2020 fourth international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*: IEEE. 2020, **pages** 861–866.

[26] Arshia Rehman, Saeeda Naz **and** Imran Razzak. “Leveraging big data analytics in health-care enhancement: trends, challenges and opportunities”. **in***Multimedia Systems*: 28.4 (2022), **pages** 1339–1371.

[27] Jinjiao Lin **andothers**. “Intelligent recommendation system for course selection in smart education”. **in***Procedia Computer Science*: 129 (2018), **pages** 449–453.

[28] Jiawei Han, Micheline Kamber **and** Jian Pei. “Mining Stream, Time-Series, and Sequence Data”. **in***Data Mining: Concepts and Techniques*: Springer, 2011, **pages** 387–414. DOI: 10.1007/978-1-4302-3325-1_14. URL: https://link.springer.com/chapter/10.1007/978-1-4302-3325-1_14.

[29] SAS Institute Inc. *Data Mining | Analytics*. Accessed: 2025-01-30. 2025. URL: https://www.sas.com/en_us/insights/analytics/data-mining.html.

[30] GeeksforGeeks. *Applications of Data Mining*. Accessed: 2025-01-30. 2025. URL: https://www.geeksforgeeks.org/applications-of-data-mining/.

[31] Jiawei Han, Micheline Kamber **and** Jian Pei. *Data Mining: Concepts and Techniques*. 3rd. Accessed: 2025-01-30. Morgan Kaufmann, 2011. URL: https://books.google.dz/books?id=dH2KQhJboSYC.

[32] Bowie State University Library. *Classification-Division*. https://bowiestate.libguides.com/english_100/Classification_Division. Accessed: 2025-02-15.

[33] LearnWithWhiteboard Digest. *All Major Data Mining Techniques Explained with Examples*. Accessed: 2025-01-30. 2024. URL: https://medium.com/@learnwithwhiteboard_digest/all- major- data- mining- techniques- explained- with- examples-1f63d61481e4.

[34] IIP Series. *Titre du document (remplacez par le vrai titre)*. Accessed: 2025-01-30. 2025. URL: https://iipseries.org/assets/docupload/rsl20246E3BB8DE57D5DD9.pdf.

[35] Auteur Inconnu. *What are the most effective data mining techniques used in academia?* Accessed: 2025-01-30. 2025. URL: https://www.linkedin.com/advice/1/what-most-effective-data-mining-techniques-used-academia-butde.

[36] TechTarget. *Association Rules in Data Mining*. Accessed: 2025-01-30. 2025. URL: https://www.techtarget.com/searchbusinessanalytics/definition/association-rules-in-data-mining.

[37] R. B. Aggarwal **and** R. K. Gupta. *Techniques of Data Mining*. Springer, 2002. ISBN: 978-0792375430. URL: https://books.google.dz/books?hl=fr&lr=&id=dH2KQhJboSYC&oi=fnd&pg=PR11&dq=technique+of+data+mining&ots=zN0FhZ7PDc&sig=TLYjJxM3hAJ8ADxAxqjjXn1nZGY&redir_esc=y#v=onepage&q&f=false.

[38] DataCamp. *Association Rule Mining in Python*. Accessed: 2025-02-01. 2021. URL: https://www.datacamp.com/tutorial/association-rule-mining-python.

[39] Educative. *What is association rule mining?* https://how.dev/answers/what-is-association-rule-mining. Accessed: 2025-02-15.

[40] Massed Compute. *What are the advantages and disadvantages of using association rule mining in data analysis?* Accessed: 2025-02-01. 2025. URL: https://massedcompute.com/faq-answers/?question=What+are+the+advantages+and+disadvantages+of+using+association+rule+mining+in+data+analysis%3F.

[41] TechTarget Contributor. *What is market basket analysis?* https://www.techtarget.com/searchcustomerexperience/definition/market-basket-analysis. Accessed: 2025-02-15.

[42] Akash Rajak **and** Mahendra Kumar Gupta. "Association Rule Mining: Applications in Various Areas". **in***Proceedings of the International Conference on Data Management*: Accessed: 2025-02-15. 2008. URL: https://www.researchgate.net/profile/Akash-Rajak/publication/238525379_Association_rule_mining-_Applications_in_various_areas/links/5d861ec6458515cbd1a88146/Association-rule-mining-Applications-in-various-areas.pdf.

[43] Javatpoint. *Association Rule Learning*. Accessed: 2025-02-01. 2025. URL: https://www.javatpoint.com/association-rule-learning.

[44] Jagadish F. *What Is Association Rule Mining with Respect to Data Mining?* https://blog.fabrichq.ai/what-is-association-rule-mining-with-respect-to-data-mining-1eb7da248db9. Accessed: 2025-02-15. **january** 2024.

[45] Codefinity. *Python Basics Course*. https://codefinity.com/courses/v2/a7e17f02-2cc9-4b92-abe0-cc8710d7011e/d2fce24c-4b70-427a-815a-afeb7ae6e604/bf978c9d-0090-4671-a662-a7277eb34e7d. Accessed: 2025-05-02. 2025.

[46] Applied AI Course. *Association Rule Mining*. Accessed: 2025-02-02. 2025. URL: https://www.appliedaicourse.com/blog/association-rule-mining/.

[47] Codefinity. *Course on Association Rule Mining*. Accessed: 2025-02-02. 2025. URL: https://codefinity.com/courses/v2/a7e17f02-2cc9-4b92-abe0-cc8710d7011e/d2fce24c - 4b70 - 427a - 815a - afeb7ae6e604 / bf978c9d - 0090 - 4671 - a662 - a7277eb34e7d.

[48] LinkedIn. *How do you measure lift and confidence in rule mining?* Accessed: 2025-02-02. 2025. URL: https://www.linkedin.com/advice/0/how-do-you-measure-lift-confidence-rule-mining-skills-data-mining.

[49] Ms. Kavita Kathuria Himani Bathla. "Association Rule Mining: Algorithms Used". **in***International Journal of Computer Science and Mobile Computing*: 4.6 (2015). Accessed: 2025-02-02, Page Numbers. URL: https://www.ijcsmc.com/docs/papers/June2015/V4I6201552.pdf.

[50] Rakesh Agrawal **and** Ramakrishnan Srikant. "Fast algorithms for mining association rules". **in***Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*: Morgan Kaufmann, 1994, **pages** 487–499.

[51] GeeksforGeeks. *Apriori Algorithm*. https://www.geeksforgeeks.org/apriori-algorithm/. Accessed: 2025-05-02. 2024.

[52] Prof. Santosh V. Chobe. "An Overview of Association Rule Mining Algorithms". **in***International Journal of Computer Science and Information Technologies*: 5.1 (2014). Accessed: 2025-02-02, Page Numbers. URL: https://www.ijcsit.com/docs/Volume5/vol5issue01/ijcsit20140501201.pdf.

[53] Rounak Dey. *FP-Growth Algorithm in Data Mining*. https://medium.com/image-processing-with-python/fp-growth-algorithm-in-data-mining-e1064accf6a3. Accessed: 2025-05-02. 2021.

[54] Coding Infinite. *ECLAT Algorithm with Numerical Example*. https://codinginfinite.com/eclat-algorithm-numerical-example/#htoc-what-is-the-eclat-algorithm. Accessed: 2025-05-02. 2023.

[55] GeeksforGeeks. *ECLAT Algorithm in Machine Learning*. https://www.geeksforgeeks.org/ml-eclat-algorithm/. Accessed: 2025-05-02. 2023.

[56] Tomas Mikolov **andothers**. "Efficient estimation of word representations in vector space". **in***arXiv preprint arXiv:1305.1372*: (2013).

[57] Huiyu Zhou **and** Kotaro Hirasawa. "Evolving temporal association rules in recommender system". **in***Neural Computing and Applications*: 31.8 (2019), **pages** 2605–2619. DOI: 10.1007/s00521-017-3217-z. URL: https://link.springer.com/article/10.1007/s00521-017-3217-z.

[58] Soulef Benhamdi, Abdesselam Babouri **and** Chiky Raja. "An association rule based recommender system for learning materials recommendation". **in***2021 International e-Engineering Education Services Conference (e-Engineering)*: IEEE, 2021. DOI: 10.1109/e-Engineering47629.2021.9470635. URL: https://doi.org/10.1109/e-Engineering47629.2021.9470635.

[59] Siti Annisa, Dian Palupi Rini **and** Abdiansah Abdiansah. "Collaborative Filtering Recommendation System Using A Combination of Clustering and Association Rule Mining". **in***Journal of Information Systems and Informatics*: 6.3 (**september** 2024), **pages** 1–10. DOI: 10.51519/journalisi.v6i3.802. URL: https://journal-isi.org/index.php/isi/article/view/802.

[60] Mark Lutz. *Learning Python*. 5 **edition**. Sebastopol, CA: O'Reilly Media, 2013. ISBN: 9781449355739.

[61] Google Research. *Google Colaboratory*. https://colab.research.google.com/. Accessed: 2025-05-18. 2023.

[62] GITHUB . https://github.com/bkarim78/Communities_Discovery_Based_Service.