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:

Prediction of web services access using Association Rule Mining

Proposed & Supervised by:

Mr ***BOUDJEBBOUR Karim***

Prepared & Presented by:

Mrs***LAACHEMI Nadjat***

Mrs***BENAISSA Chaima***

Academic Year 2020-2021


بسم الله الرحمن الرحيم

Acknowledgment

First, we would like to thank ALLAH the Almighty for giving us health, will as well as the patience to carry out this work, and guided us through every part of our studies until this moment as we are at the end of this journey.

We will continue to thank our dear professors at Dr. Yahia Fares, University - Médéa, who were always at our service, helping us to achieve our dreams despite how busy they are with their own work.

In particular, we would like to thank our supervisor and dear professor Dr. BOUDJEBBOUR Karim who was always by our side during the preparation of this thesis. We achieved this with his guidance, help and constant encouragement towards us at all times,

'We're sincerely very grateful'

At the end we thank all those who their service helped us to make this work come to life, but we cannot mention their names because they are very numerous.

Dedication

﴿وَقَضَىٰ رَبُّكَ أَلَّا تَعْبُدُوا إِلَّا إِيَّاهُ وَبِالْوَالِدَيْنِ إِحْسَانًا ۚ إِمَّا يَبْلُغَنَّ عِندَكَ الْكِبَرَ أَحَدُهُمَا أَوْ كِلَاهُمَا فَلَا تَقُل لَّهُمَا أُفٍّ وَلَا تَنْهَرْهُمَا وَقُل لَّهُمَا قَوْلًا كَرِيمًا (24) وَاخْفِضْ لَهُمَا جَنَاحَ الذُّلِّ مِنَ الرَّحْمَةِ وَقُلْ رَبِّ ارْحَمْهُمَا كَمَا رَبَّيَانِي صَغِيرًا ﴾

In memory of my loving dad

In honor of my precious mom

'Thank You Infinitely'

Laachemi Nadjat

Dedication

To My Beloved Parents

To My Dear Siblings: Hala, Asmaa, Sarah and Ismail

To My Friends: Karima, Aya and Maria

To My Husband Hamza

To My Partner Nadjat

Benaissa Chaima

Abstract

As the number of web services on the web grows massively, researchers realized that it is necessary to create a system that allows for discovering hidden patterns from existing data automatically and using these patterns to recommend specific web services to users. And with that idea in mind, web mining was born, a technique for identifying a user's need from the internet. Since then, association rules have always been an important part in many recommender systems' applications to detect interesting patterns. In our work, we propose an approach of web service recommender system using association rule mining based on previously saved data only, no need for user's profile and personal information as the regular systems usually work. The algorithms are used to study the available data and detect frequent itemsets, then generate association rules using the two measures of interestingness: minimum support and confidence. Using different synthetic datasets, the performance of the algorithms is compared and results are drawn and discussed. It proved that it is possible to use this approach to recommend web services to users. With the limitation of not taking weight into consideration.

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

Résumé

Comme le nombre de services Web sur le web se développe massivement, les chercheurs ont réalisé qu'il est nécessaire de créer un système automatique qui permet de découvrir les modèles cachés à partir des données existantes et utiliser ces modèles pour recommander des services web spécifiques aux utilisateurs. Et comme ça, le web mining est né, une technique permettant d'identifier les besoins d'un utilisateur à partir d'Internet. Et depuis cela, les règles d'association ont toujours été une partie importante dans les applications de nombreux systèmes de recommander pour détecter des modèles intéressants. Les algorithmes sont utilisés pour étudier les données disponibles et détecter des itemsets fréquents, puis générer des règles d'association en utilisant les deux mesures d'intérêt : support et confiance minimum. À l'aide de différentes bases des données synthétiques, les performances des algorithmes sont comparées et les résultats sont tirés et discutés. Il été prouvé qu'il est possible d'utiliser ce principe pour recommander des services web aux utilisateurs. Avec la limitation de ne pas prendre en compte le poids.

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

ملخص

مع تزايد عدد خدمات الويب عبر الانترنت بشكل كبير، أدرك الباحثون أنه من الضروري إنشاء نظام يسمح تلقائيًا باكتشاف الأنماط المخفية باستعمال البيانات الموجودة، واستخدام هذه الأنماط للتوصية بخدمات ويب معينة للمستخدمين. وبأخذ هذه الفكرة بالاعتبار، وُلد الويب ماينينغ، وهي تقنية لتحديد احتياجات المستخدم من الإنترنت. منذ ذلك الحين، كانت قواعد الارتباط دائمًا جزءًا مهمًا في العديد من تطبيقات أنظمة التوصية لاكتشاف الأنماط المثيرة للاهتمام. في عملنا هذا، نقترح فكرة لنظام توصية لخدمات الويب باستخدام التنقيب عن قواعد الترابط استنادًا إلى البيانات المحفوظة مسبقًا فقط، ولا حاجة للحساب الشخصي للمستخدم أو لمعلوماته الشخصية، التي تعتمد عليها الأنظمة العادية عادةً. تُستخدم الخوارزميات لدراسة البيانات المتاحة واكتشاف مجموعات العناصر المتكررة معا، ثم إنشاء قواعد الترابط باستخدام مقاييس الأهمية: الحد الأدنى للدعم والثقة. باستخدام مجموعات بيانات مختلفة، تتم مقارنة أداء الخوارزميات واستخلاص النتائج ومناقشتها. ثبت أنه من الممكن استخدام هذا النهج للتوصية بخدمات الويب للمستخدمين. لكن دون أخذ وزن العناصر وتكرارها بعين الاعتبار..

الكلمات المفتاحية:

قواعد الترابط، تنقيب البيانات، Apriori، FP-Growth، ECLAT، أنظمة التوصية، خدمات الويب.

Table of Contents

Acknowledgment

Dedication

Abstract in English

Abstract in French

Abstract in Arabic

Figures

Tables

Acronyms

General Introduction

General introduction 02

Chapter I: Web Services & Recommender Systems

1. Introduction 05

2. Web Services 06

2.1. History of Web Services 06

2.2. Definition of Web Service 07

2.3. Web Services Protocols 07

2.3.1. SOAP 08

2.3.2. WSDL 09

2.3.3. UDDI 10

2.4. Service Oriented Architecture 10

2.4.1. Definition 10

2.4.2. Web Services & SOA 11

2.4.3. Characteristics of SOA 11

3. Recommender Systems 11

3.1. History of Recommender Systems 12

3.2. Area of Use 14

3.3. Techniques 15

3.3.1. Collaborative Filtering 16

3.3.2. Content based Filtering 17

3.3.3. Hybrid Filtering 17

3.4. Purpose 18

3.5. Quality & Evaluation 19
3.6. Web Services & Recommender Systems 20
3.7. Benefits 20
3.8. Issues 21
4. Conclusion 23

Chapter II: Data Mining & Association Rules

1. Introduction 25
2. Data Mining 25
2.1. Definition of Data Mining 25
2.2. Application Domain of Data Mining 26
2.3. Data Mining in KDD Process 28
2.4. Techniques of Data Mining 30
2.4.1. Classification 30
2.4.2. Prediction 30
2.4.3. Association 31
2.4.4. Clustering 31
2.4.5. Regression 31
2.4.6. Outlier Detection 32
3. Association Rules 32
3.1. History of Association Rules 32
3.2. Definition of Association Rules 33
3.3. Advantages 33
3.4. Application Areas 33
3.5. Concept of Association Rule Mining 35
3.5.1. Itemsets & Association Rules 36
3.5.2. Metrics of Interestingness 37
a. Support 37
b. Confidence 38
c. Lift 38
d. Conviction 39
3.5.3. Alternative Measures of Interestingness 39
3.6. Algorithms 40

4. Conclusion 40

Chapter III: Proposed Solution & Implementation

1. Introduction 42
2. Related Work 44
3. Problematic & Objectives 47
4. Solution Overview 48
4.1.1. Apriori algorihm 48
4.1.2. ECLAT algorithm 52
4.1.3. FP-Growth algorithm 54
5. Implementation & Tests 58
5.1. Development Environment 58
5.1.1. Materials 58
5.1.2. Programming Language 59
5.1.3. Integrated Development Environment 59
5.2. Data Description 60
5.3. Tests & Results 62
5.3.1. Runtime with different minimum support values 62
a. Fixed number of web services with varying transaction number 62
b. Varying web services number with fixed transaction number 66
5.3.2. Runtime with different number of transactions 71
5.3.3. Runtime with different minimum confidence values 72
5.4. Discussion 73
5.4.1. Algorithm Performance 73
a. ECLAT 73
b. FP-Growth 73
c. Apriori 74
5.4.2. Effect of attribute size 74
5.5. Test Run 74
6. Conclusion 80

General Conclusion

General Conclusion 82
Bibliography & Webography 84

Figures

Figure 01: *The Web Services Roles, Operations and Artifacts* ____ *07*

Figure 02: *Web services detailed model* ____ *08*

Figure 03: *Recommendation filtering techniques* ____ *15*

Figure 04: *KDD process in Data Mining* ____ *29*

Figure 05: *Framework of web service prediction* ____ *48*

Figure 06: *Apriori algorithm's organizational chart* ____ *50*

Figure 07: *Apriori algorithm* ____ *51*

Figure 08: *ECLAT algorithm's organizational chart* ____ *53*

Figure 09: *ECLAT algorithm* ____ *54*

Figure 10: *FP-Growth algorithm's organizational chart* ____ *56*

Figure 11: *FP-Growth algorithm* ____ *57*

Figure 12: *A sample of the test dataset* ____ *61*

Figure 13: *Runtime of algorithms on datasets with 7 web services* ____ *63*

Figure 14: *Runtime of algorithms on datasets with 10 web services* ____ *64*

Figure 15: *Runtime of algorithms on datasets with 20 web services* ____ *65*

Figure 16: *Runtime with minimum support on Data with 10k* ____ *67*

Figure 17: *Runtime with minimum support on Data with 30k* ____ *68*

Figure 18: *Runtime with minimum support on Data with 50k* ____ *69*

Figure 19: *Runtime with minimum support on Data with 100k* ____ *70*

Figure 20: *Runtime with different number of transactions* ____ *71*

Figure 21: *Runtime of different minimum confidence values* ____ *72*

Figure 22: *Output of using FP-growth with large dataset* ____ *74*

Figure 23: *Details on the dataset's size* ____ *74*

Figure 24: *Apriori's function to get frequent itemsets* ____ *75*

Figure 25: *Results of Apriori's function* ____ *76*

Figure 26: *Length function to get the number of frequent itemsets* ____ *76*

Figure 27: *ARM's function for generating association rules* ____ *76*

Figure 28: *Results of ARM's function* ____ *77*

Figure 29: *Length function to get the number of association rules* ____ *77*

Figure 30: *A compact version of the association rules list of results* ____ *78*

Figure 31: *Length function to get the number of frequent itemsets* ____ *79*

Figure 32: *Length function to get the number of association rules* ____ *79*

Figure 33: *A compact version of the second association rules list of results* ____ *79*

Tables

Table 01: *Market basket transactions*_____*37*

Table 02: *Details of the used datasets*_____ *60*

Table 03: *Average Runtime of minimum confidence values*_____ *72*

Acronyms

AI: *Artificial Intelligence*

ARM: *Association Rule Mining*

CF: *Collaborative Filtering*

CRM: *Customer Relationship Management*

CSCW*: Computer-Supported Cooperative Work*

DNA*: Deoxyribonucleic Acid*

E-*: Electronic*

ECLAT: *Equivalence Class Clustering and Bottom-Up Lattice Traversal*

FP: *Frequent Pattern*

HTTP: *Hypertext Transfer Protocol*

HTTPS: *Hypertext Transfer Protocol Secure*

IBM: *International Business Machines*

KDD: *Knowledge Discovery in Databases*

KNN: *K-Nearest Neighbors*

ML: *Machine Learning*

OPTICS: *Ordering Points to Identify the Clustering Structure*

PARC: *Palo Alto Research Center*

POS: *Point-Of-Sale*

RS: *Recommender Systems*

SOA: *Service-Oriented Architecture*

SOAP: *Simple Object Access Protocol*

TV: *Television*

UDDI: *Universal Description, Discovery, And Integration*

URI: *Uniform Resource Identifier*

W3C: *World Wide Web Consortium*

WS: *Web Service*

WSDL: *Web Services Description Language*

XML: *Extensible Markup Language*

General Introduction

General Introduction

We are of in the era of speed, where everything is better if done faster and easier, and technology was and is still the way to make those needs possible to achieve. Access to information systems these days is now increasingly based on Internet technologies. Standardization efforts in this context have accentuated the enthusiasm of people and organizations (academic, industrial, commercial, or institutional) for the use of Internet and have enabled the emergence of web services as a support for application development accessible through it. Thus, the technologies associated with web services became essential for the development of interacting applications with users through the Internet.

As of January 2, 2021, there were 4,783,503,852 internet users extending worldwide[1], and whilst trying to find what you want online, sometimes it's overwhelming and hard due to the huge amount of resources and information's accessible on the world wide web, As it is estimated that in 2016 the number of Webpages available on the Internet is more than 4.4 billion [2], and approximately one million new pages are added to the web daily, so it takes time and effort to reduce the field of results on which a person can find what he needs .Therefore, analytics of user behavior on the web is constantly increasing; this increase facilitates access to the required information easily and in a short time span.

The usual recommender systems are based on different parameters that especially take in account user's profile, our work is based on a data mining technique called association rules, where the system recommends web services to users using this technique. Association rules that reveal similarities between web pages connected to each other using hyperlinks.

In association rule mining we uncover important relationships between variables in large a database Using some metrics [3], and here we use association rules to

reveal the similarities between web pages inferred from user's behavior, the goal of this is to suggest web pages that seem useful to the user [2].

Our presented work is constructed around the following structure:

- This general introduction to the subject of web services prediction on the web followed by the first chapter, in which we talk lightly about web services and recommender systems, their history and different techniques.
- In the second chapter, we explain data mining technique, focusing on its application and association rule analysis.
- The third chapter is devoted to the issue and the conception of the chosen method that we're using in order to solve said issue, by presenting the state of the art, the problematic, the methodology and the choice of algorithms to use, along with the implementation of it, followed by a discussion of this approach results and its evaluation in order to get a solid final approach
- Finally, we end our thesis with a general conclusion that summarizes the work done and the results obtained, putting into perspective the work that's carried out during this project, and the future possibilities to broaden the idea of research.

Chapter I:

Web Services & Recommender Systems

1. Introduction

The W3C introduced Web services to provide a standard way of communication between software applications. Web services are technologies designed to support interoperable computer-to-computer interactions over networks. UDDI enables applications to publish and find Web services. Usually, techniques to search for Web services in UDDI repositories are keyword-based. Even if the WSDL/UDDI contained semantic information in connection with the providers and syntactic information in connection with the web services, it remains difficult to decide how to find “relevant” web services, starting from simple keywords. Therefore, there is a need to design systems that assist users to locate web services.

Recommender systems allow entities providing items to guide the choices made by users. The recommendation can be issued by the service provider who possesses the items or by brokers. The recommender system we’re introduce here, is a search tool that can be anchored to any UDDI. [4]

In this chapter we will further explain web services and recommender systems.

2. Web services

2.1. History of Web Services

The World Wide Web was developed thanks to “Cern”, in the 1990's, from Time Berners-Lee’s Institute. W3C then adopted and it introduced Web services, the internet user can access the web more freely and link information sources in the world via the internet, Users were able to find the information without the need to know its whereabouts, through simple links , after 1993 Global networks was launched and the growth of the Internet has led to a new phenomenon in the world’s development of technology, increasing the numbers of users and creators of web services from a few thousand to about 300 million in 1990 [5].

2.2. Definition of Web Service

A Web service, as defined by the W3C Web Services Architecture Working Group, is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process-able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. [6]

It encapsulates discrete functionality identified by URI that allow for the communication and exchange of data between applications and systems and are distributed and programmatically accessible over the internet. They are self-contained, modular business applications that have open, Internet-oriented, implementing services from simple services to complex business processes.

2.3. Web Services Protocols

The web services protocols are used to define, locate, implement, and make web services interact with each other, and it aims to provide a mean of describing data and behaviour in a machine-process-able using XML-based messages over HTTPS. For IBM web services, SOAP is the standard protocol used to send messages WSDL files are central to testing SOAP-based services also from current technologies for publishing web Services UDDI working on assigning objects from a universe to two or more classes or categories

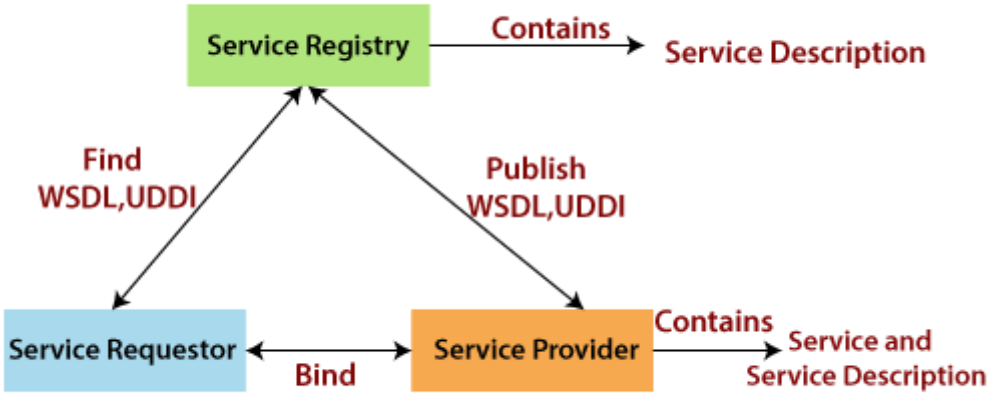


Figure 01: The Web Services Roles, Operations and Artifacts

The Web Services architecture describes how to instantiate the elements and implement the operations in an interoperable manner. The architecture of web service interacts among three roles: service provider, service requester, and service registry. The interaction involves the three operations: publish, find, and bind. [7]

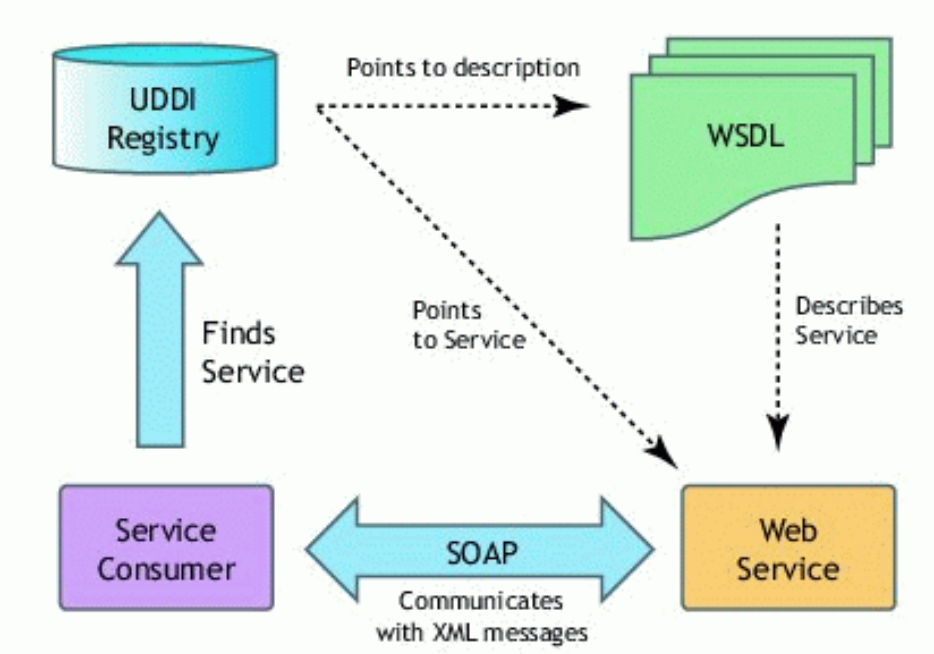


Figure 02: Web services detailed model

The Web Services specification includes:

Simple Object Access Protocol (SOAP)

Universal Description, Discovery, and Integration (UDDI)

Web Services Description Language (WSDL)

Figure 02 gives an architectural representation of the Web Services model. [8]

2.3.1. SOAP :

The SOAP protocol is one of the W3C products, "Simple Object Access Protocol" it provides remote procedure calls, SOAP is a method of transferring messages, or small amounts of information, over the Internet, it uses XML format over HTTPS. SOAP requires a WSDL (Web Services Description Language) file to describe the payload for XML messages. It’s a supported data transmission standard, which is the protocol that Web pages are sent over.

SOAP defines three parts to every message:

Envelope

Encoding rules

Communication styles

The SOAP envelope establishes a framework to define what is in a message and how it should be processed. A SOAP message is an envelope containing exactly one body and zero or more headers. The envelope is the top element of the XML document, providing a container for control information, the address of a message, and the message itself. Any control information, such as quality-of-service characteristics, is carried via headers. The message's identity and parameters are contained in the body. The headers and the body are both child elements of the envelope. [9]

2.3.2. WSDL :

WSDL files are central to testing SOAP-based services, it is a description language for web services in XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information it is essentially based on the mechanisms of SOAP, HTTP for invoking remote objects, the operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint, related concrete endpoints are combined into abstract endpoints.

WSDL is extensible to allow the description of endpoints and their messages regardless of the message formats or network protocols actually used to communicate.

2.3.3. UDDI :

Is an XML-based registry for businesses worldwide to list them on the Internet, it's a specification that defines how to publish and discover Web (web service). The registration of a service provider in the UDDI directory allows him to introduce himself and his services, UDDI may contain information about the providers and the services they publish and the ultimate goal is to stream online transactions. UDDI register can be accessed in SOAP.

2.4. Service Oriented Architecture

2.4.1. Definition

SOA is a style of software design where services are provided to the other components by application components via a communications protocol over a network. The principles of service-orientation are independent of any product, vendor or technology. it defines a way to make software components reusable and interoperable via service interfaces. Services use common interface standards and an architectural pattern so they can be rapidly incorporated into new applications. SOA just makes it easier for software components over various networks to work with each other.

Web services which are built as per the SOA architecture tend to make web service more independent. The web services themselves can exchange data with each other and because of the underlying principles on which they are created, they don't need any sort of human interaction and also don't need any code modifications. It ensures that the web services on a network can interact with each other seamlessly. [10]

2.4.2. Web Services & SOA :

Web service implementation is not the same as SOA implementation. The service layer is just one tier in any potential, and introducing one or more business functions as Web services would not assure that a full-fledged SOA can succeed. There are other relatively important levels within SOA to consider (application, business process, and service layer). A SOA architecture that is based on technology is unlikely to provide the best outcomes. It’s better to view a vendor's middleware product as part of a logical infrastructure service sub-layer in the SOA rather than the full story on its own. [11]

2.4.3. Characteristics of SOA :

These services have the following characteristics:

They are loosely coupled.

They support interoperability.

They are location-transparent.

They are self-contained. [11]

3. Recommender Systems

When you think of recommender systems, think of the restaurant your friend suggested to you, or that new Netflix show he said he have been watching, these recommendations are based on what he likes, and since not everybody has the same taste in food, TV shows, music and any other category, you may or may not like these recommendations. But you want to pick something different that you will for sure like, based on what you usually prefer, as if someone understood

your specific needs and presented them in an organized way, wouldn't that be convenient?

Recommender systems help users find products they may like when lacking personal experience or facing an overwhelmingly large set of items, it's a subclass the **information filtering system** that tries to predict what the users want. It's an automated system that filters some entities online to produce recommendations to the user, they are used in various application areas, including online retail, tourism and entertainment (it can be a product, ads, people, movies, songs, videos. etc.) And we see it everywhere, from Netflix, Amazon, YouTube to search engines like Google, you pick something and the system recommends the next one based on viewing history or personal information, it doesn't only work on products presented, but also the order of it.

3.1. History of Recommender Systems

Information saturation isn't new. In the early '90s, users were overwhelmed with the massive stream of web services. And the idea of a recommender system started to take place around that time to help sort that much information. However, the basics of recommender systems were founded by researches into cognition science and information retrieval, and its first manifestation was the Usenet communication system created by Duke University in the second half of the 1970s [12], where users were able to share textual content with each other. These were categorized into newsgroups and subgroups for easier search, but it was not directly built on or targeted the preferences of users.

There was an emergence of the first analysis paper on collaborative filtering in the mid-1990s, users were being flooded by a massive number of incoming

documents due to the increasing use of the internet. So, four researchers at the Xerox PARC decided to take on the challenge.

David Goldberg headed the community that created *Tapestry*, a pioneering mail and repository system. It enabled the user to make searches based on the contents of documents as well as user reactions. For example, “I want to find all the documents containing the words ‘Sunny day’, that the user ‘Sarah’ has rated ‘5 stars’,”.

The experiment was predicated on the belief that “information filtering can be more effective when humans are involved in the filtering process” [13].

They published their paper “Using collaborative filtering to weave an information tapestry” coining the term “Collaborative Filtering” and “Tapestry Filters”, which is implemented by having users annotate documents, and then allowing filtering using those annotations. Their approach was to recommend items to a user based directly on that user's similarity to other users [14].

In 1992, after hearing keynote speaker Shumpei Kumon talk about his vision for an information economy John Riedl and Paul Resnick started working on a collaborative filtering system for Usenet news. This time it was targeting the preferences of users, the system collected ratings from Usenet readers and used those ratings to predict how much other readers would like an article before reading it, it was basically invented to assist users in finding relevant items in the large amount of content posted in newsgroups. This meant they were able to add new newsgroups to its existing dataset so that the system could better predict user preferences for items. And so, the “GroupLens recommender” system was born, and the servers that collected the ratings and the performed the calculation were called the “Better Bit Bureau”. “GroupLens” is now used as a name for both this recommender system and the University of Minnesota research lab. It’s

considered was one of the few first automated filtering systems based on historical pattern of ratings.

A research paper was published including the algorithm, system design and results of a feasibility study (that was performed between Massachusetts Institute of Technology and the University of Minnesota), at the 1994 CSCW conference in the spring of 1996, the first collaborative filtering workshop was organized by Resnick and Hal Varian at the University of California, Berkeley.[15]

So, in that environment, you get the early days. The inspiration and the rapid growth period in today's recommender systems, particularly the automated collaborative filtering systems, was in the early 90s.

3.2. Area of use

A recommender system is used in a variety of areas, aiming to predict users' interests and recommend product items that are quite likely interesting for them. If that’s vague enough, you’d be surprised to see that recommendation systems are used abundantly in our everyday interactions, with apps and sites without us even realizing. For example, Google is using them to serve personalized advertisements and search results, YouTube to recommend related videos, Netflix to recommend movies, Amazon to recommend products, Spotify to recommend similar music based on our listening history, and many others to recommend a wide variety of different categories (music, movies, shows, books, games, podcasts, etc.) The value of recommender systems is massive and often drives a critical amount of revenue or traffic in businesses. For example, 35% of user purchases in Amazon and 75% of show/movie selections in Netflix are driven from the output of recommendation algorithms. They are among the most powerful machine learning systems that online retailers implement in order to

drive sales and views. The stated companies and many others leverage recommender systems to help users discover new and relevant items and content (products, videos, jobs, music), creating a delightful user experience while driving incremental traffic and revenue.

3.3. Techniques

A recommender system is used to suggest relevant items to users, and the use of efficient and accurate recommendation techniques is very important for a system that will provide good and useful predictions to its individual users. This emphasizes the importance of understanding the features and potentials of different recommendation techniques:

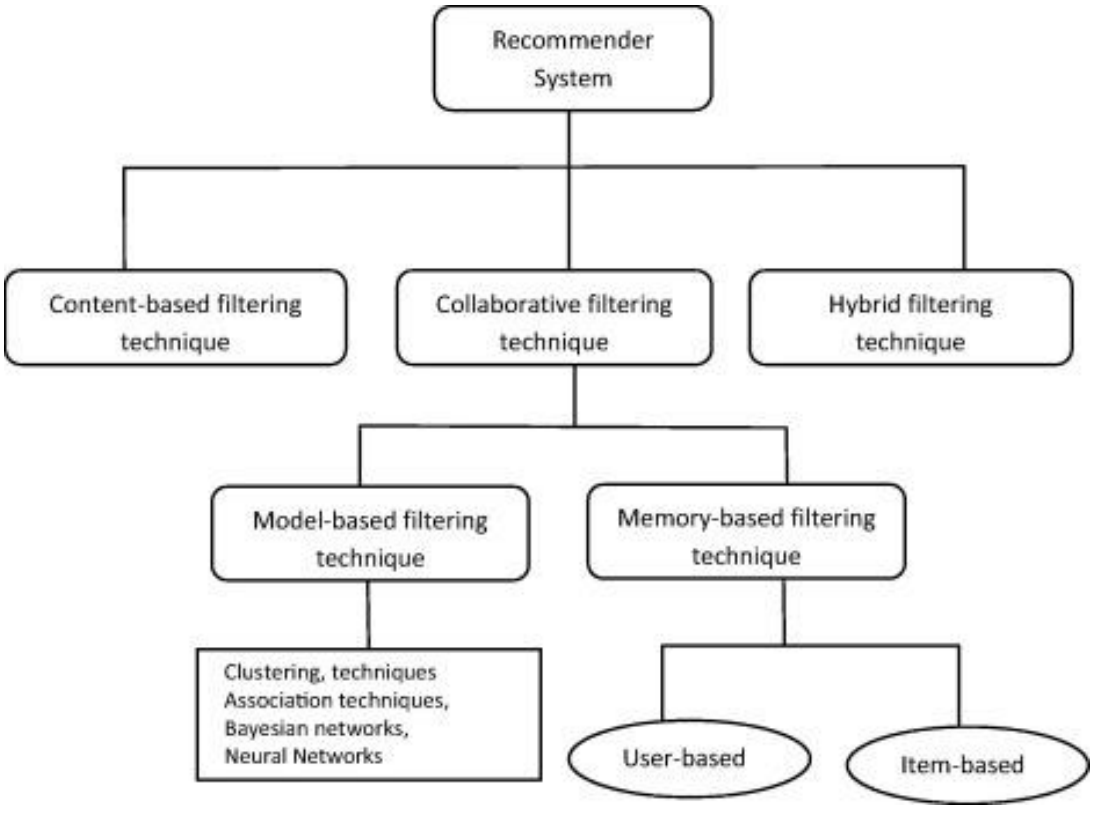


Figure 03: Recommendation filtering techniques

3.3.1. Collaborative Filtering :

Collaborative filtering is one of the most commonly used recommendation algorithms, it's an independent prediction tool used by recommender systems; it works by collecting and filtering users provided information (collaborating), then building a database of preferences for items by users. It then matches users with relevant interest and preferences by calculating similarities between their profiles. The process makes automatic predictions about the need of a user, to suggest personalized recommendations according to their taste. The head advantage of this filtering system is that it's independent of any machine analyzable content, therefore it's more accurate in the suggestion it makes.

Collaborative-filtering systems focuses on the relationship between users and items. The similarity of items is determined by the similarity of the ratings of those items by the users who have rated both items. So, the recommendations that are produced by CF can be of either prediction or recommendation. The items that were already rated by the user are considered to be those that the user will like the most.

As it gained a lot attention, deeper research has been done on collaborative filtering, and it branched out mainly to two approaches:

- **Memory based:** which can be divided into two classes:
  - *Item-Item Collaborative Filtering: "Users who liked this item also liked …"*
  - *User-Item Collaborative Filtering: "Users who are similar to you also liked …"*
- **Model based:** which contains more methods such as:
  - *Clustering based algorithms.*
  - *Association techniques.*

- *KNN.*
- *Bayesian networks.*
- *Neural networks/Deep learning…* [16]

3.3.2. Content Based Filtering :

Content-based filtering technique is also a domain-dependent algorithm. The user is recommended things that are primarily related to the positively rated items. To detect similarities across papers, Content-based approach uses a variety of models. These methods provide suggestions by learning the underlying model using statistical analysis or machine learning.

This technique is often used in recommender systems; the algorithms are designed to advertise or recommend items to users based on knowledge off of previous actions or explicit feedback. It relies on both implicit data such as browsing history and purchases and explicit data such as ratings provided by the user. It and bases the suggestions on that.

Since the systems works with the user's personal data, it doesn't need any other, and the results are more accurate on a larger number of users. Also, this model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

The recommendations of this system are based on attributes of the item and it works well when descriptive data on the content is provided beforehand.

3.3.3. Hybrid Filtering :

Recommending news articles based on browsing of news is useful, but would be much more useful when music, videos, products, discussions etc.

from different services can be recommended based on same news browsing. To achieve this, most content-based recommender systems now use some form of hybrid system.

Hybrid filters merge two or more recommender systems in different ways in order to benefit from all their structural and complementary advantages, and gain a better performance from the assembled system. We address the most relevant problems considered and present the associated data mining and recommendation techniques used to overcome them.

Most commonly, collaborative filtering is combined with some other technique in an attempt to avoid the ramp-up problem. [17]

There are many more types of recommendation techniques, some others are less common but just as effective in their own ways, like:

- Demographic based recommender system.
- Utility based recommender system.
- Knowledge based recommender system.

3.4. Purpose

Recommender systems help the users get personalized suggestions, and take correct decisions in their online experience, they are useful to both service providers and users, as they increase sales, retain customers, reduce transaction costs of finding and selecting items in an online shopping environment.

Some RSs use their knowledge about each user to recommend new products. While others base it on attributes of the item. Depending on what the web services provided is.

3.5. Quality & Evaluation

The quality of the performance of a recommender's system is assessed based on how accurate the recommendations are to the user's preference, using different types of metrics depending on the type of filtering technique that the system is based on.

Metrics of relevancy are usually used to determine the pertinence of a recommender's system. Recall and Precision are evaluation metrics that are commonly used in classification settings. In the context of recommender systems, we use metrics like Recall@k and Precision@k, since it is more common to provide k recommendations per example.

Recall@k is defined as the percentage of true labels captured when we recommend k labels per example. Its exact formula is:

$$recall@k = \frac{\#\text{ of true labels captured}}{\#\text{ of true labels}}$$

On the other hand, **Precision@k** is defined as the percentage of predictions we get right. Its exact formula is:

$$precision@k = \frac{\#\text{ of true labels captured}}{\#\text{ of predictions made}}$$

As a side note, metrics need not have a purely mathematical notion. A metric could be the cost per recommendation. [18]

3.6. Web services & Recommender Systems

Every business owner wishes to reach more of the customers who are interested in what he offers, on the other hand, users find it difficult to reach the relevant content that they seek with the huge amount of web services created every second. It is helpful to develop systems that assist clients in finding Web services.

The user profile, which summarizes what a user likes and dislikes, can be taken into account by such systems. The profile is typically used for online personalization, which is an Internet strategy for tailoring websites to specific users. [19]

3.7. Benefits

As stated earlier, the majority of websites rely on recommender systems, they've become essential for every person who uses the internet, we interact with them unconsciously on a daily basis, this massive use and growth of these systems is naturally caused by many advantages such as:

- ***Increasing revenue and sales :***

Which some might consider the most important outcome of having a business, resulting in extra profit to the business owner. The recommender systems were reported to increase revenue up to 300 %, conversion rates up to 150 %, and consumers' average order value up to 50 % according to a 2013 study, and by time the revenue surely gone up since then.

- ***User's satisfaction with the services performance :***

As you may have already guessed, the second advantage comes naturally with the first, when a user gets a personalized feed created by a recommendation engine catered especially for him and his interests – whether it's for music, books, news, movies, or e-commerce – he feels less stressed and more connected to the service. Therefore, if the service can provide this tailored feed and understand the need of its users (using a similar audience or each person's own data), it makes him use this service more, which leads to getting fulfilled perfectly in a lesser time.

- ***Turnover growth :***

Not only can the recommender system cater its services based on the user's personal data, it can also take into consideration the connections between several users too. It's a human nature to care what others think about us, recommender systems can provide very precise suggestions taking in account the reviews of our friends and close ones about a given service (of course the option is there if we choose to not have that). As a result, it will increase the on-site traffic and generates extra sales, that's why many much known businesses implement this feature to their algorithms. [5]

3.8. Issues

It is not conventional to discuss flaws, yet we shall do so. It should be noted that developing and gaining access to all of the benefits of recommendation systems is not a natural or simple procedure as it may appear. Here are a few common issues that are usually encountered when dealing with recommender's systems.

- ***Deficiency of data :***

The lack of high-quality data is perhaps the most critical challenge when it comes to recommender systems, it takes a huge amount of data to create the recommendations, the more data the system has to work with, and the more likely it is to provide accurate suggestions. The less there are, the looser the accuracy is.

So, to get great suggestions, it is better to provide accurate data, and spend more time collecting and perfecting it, than it is to retain the systems numerous times without knowing where the problem is.

- ***Information variability:***

Recommender systems are usually built on information gathered in the past or present. There are several industries where information is rapidly changing. If we relied on the previous collected data to complete the learning process, by the time the recommender's system is built, the situation may have already changed and the system becomes completely irrelevant.

- ***Unpredictable performance :***

It is easy to get excited seeing all the benefits a recommender system can provide to businesses, but estimating the financial revenue it can bring before implementing it to the business process is nearly impossible, so it’s a shot in the dark, unless you choose to stay safe and use the predictable technology that every other company uses but where is the originality in that?

4. Conclusion

Recommenders' systems are a new emerging computing paradigm that involves a multidisciplinary approach by analyzing and modeling user's behaviors on different web services and platforms in order to trace patterns and produce an app that helps understand what the users want and need, and suggests it to him when looking for something, by analyzing similar client's behavior, conveniently saving the time and effort of the user.

Chapter II :

Data Mining & Association Rules

1. Introduction

Data Mining in Today's World is a cornerstone of analytics, helping you develop the models that can uncover connections within millions or billions of records. It has applications in multiple fields, like science and research. We've seen the staggering numbers, the volume of the data produced is doubling every two years. Unstructured data alone makes up 90% of the digital universe. But more information does not necessarily mean more knowledge.[20] Data mining is widely used in business (insurance, banking, retail), science research (astronomy, medicine), and government security (detection of criminals and terrorists).

2. Data Mining

2.1. Definition of Data Mining

Data mining, also called knowledge discovery in database in computer science, is the process of turning raw data into useful information, using software to discover structures and patterns in large and complex datasets. [21] The field combines tools from statistics and AI (such as neural networks and machine learning). Using it, businesses can learn more about their customers to develop more effective marketing strategies, increase sales and decrease costs.

There are two aspects to data mining: model building and pattern detection, with database management to analyse large digital collections, known as datasets. [22] Data mining involves effective data collection and warehousing as well as computer processing. For segmenting the data and evaluating the probability of future events, data mining uses sophisticated mathematical algorithms.

2.2. Application Domain of Data Mining

Data Mining is primarily used today by companies with a strong consumer focus retail, financial, communication, and marketing organizations, to “drill down” into their transactional data and determine:

- Pricing
- Customer preferences
- Product positioning
- Impact on sales
- Customer satisfaction
- Corporate profits

With data mining, a retailer can use point-of-sale records of customer purchases to develop products and promotions to appeal to specific type of customers.

Here is the list of some important areas where data mining is widely used:

- ***Financial banking :***

The banking industry across the world has undergone tremendous changes in the way the business is conducted. With the recent implementation, greater acceptance and usage of “electronic” banking, the capturing of transactional data has become easier and, simultaneously, the volume of such data has grown considerably. It is beyond human capability to analyse this huge amount of raw data and to effectively transform the data into useful knowledge for the organization.[23] Data Mining can help by contributing in solving business problems by finding patterns, associations and correlations which are hidden in the business information stored in the databases.[24] By using data mining to analyse patterns and trends, bank executives can predict, with increased accuracy, how customers will react to adjustments in interest rates, which customers will be

likely to accept new product offers, which customers will be at a higher risk for defaulting on a loan, and how to make customer relationships more profitable [25]

- ***Electronic commerce :***

Data mining in e-commerce is a vital way of repositioning the company for supporting the enterprise with the required information concerning the business. Recently, most companies adopt e-commerce and being in possession of big data in their data repositories. The only way to get the most out of this data is to mine it to increase decision making or to enable business intelligence.

- ***Bioinformatics :***

Applications of data mining to bioinformatics include gene finding, protein function domain detection, function motif detection, protein function inference, disease diagnosis, disease prognosis, disease treatment optimization, protein and gene interaction network reconstruction, data cleansing, and protein sub-cellular location prediction. For example, microarray technologies are used to predict a patient's outcome. On the basis of patients' genotypic microarray data, their survival time and risk of tumor metastasis or recurrence can be estimated. [5]

- ***Telecommunications :***

The telecommunications industry was an early adopter of data mining technology and therefore many data mining applications exist. Several typical applications are described in this section. These applications are divided into three application areas: fraud detection, marketing/customer profiling and network fault isolation. [26]

- ***Fraud detection :***

Billions of dollars have been lost to the action of frauds. Traditional methods of fraud detection are time consuming and complex. Data mining aids in

providing meaningful patterns and turning data into information. Any information that is valid and useful is knowledge. A perfect fraud detection system should protect information of all the users. A supervised method includes collection of sample records. These records are classified fraudulent or non-fraudulent. A model is built using this data and the algorithm is made to identify whether the record is fraudulent or not. [27]

- ***Direct marketing :***

In direct marketing, data mining is a description of likely buyers or responders, and is useful in obtaining higher profit than mass marketing. As an example, banks and insurance companies may have a large database of customers to whom they want to sell certain products (such as loan products, retirement packages, and life insurance policies). [28]

- ***Pharmaceutical industry :***

The identification and quantification of pharmaceutical information can be extremely useful for patients, physicians, pharmacists, health organizations, pharmaceutical manufacturers, drug testing companies, etc. For example, information on interactions among prescription medicines, interactions between any kind of medicine and various foods, vitamins, and mineral supplements, common characteristics between certain drug groups and offending foods medicines, will yield very useful results using data mining techniques. [29]

2.3. Data Mining in KDD Process

The term Knowledge Discovery in Databases refers to the broad procedure of discovering knowledge in data and emphasizes the high-level applications of specific Data Mining techniques. It is a field of interest to researchers in various fields, including artificial intelligence, machine learning, pattern recognition,

databases, statistics, knowledge acquisition for expert systems and data visualization.

Data mining is usually applied to the operational procedure for obtaining new insights, whereas KDD is used by the research community developing new techniques. We use the terms interchangeably. The main objective of the KDD process is to extract information from data in the context of large databases. It does this by using Data Mining algorithms to identify what is deemed knowledge.

The Knowledge Discovery in Databases is considered as a programmed, exploratory analysis and modelling of vast data repositories. KDD is the organized procedure of recognizing valid, useful, and understandable patterns from huge and complex data sets.

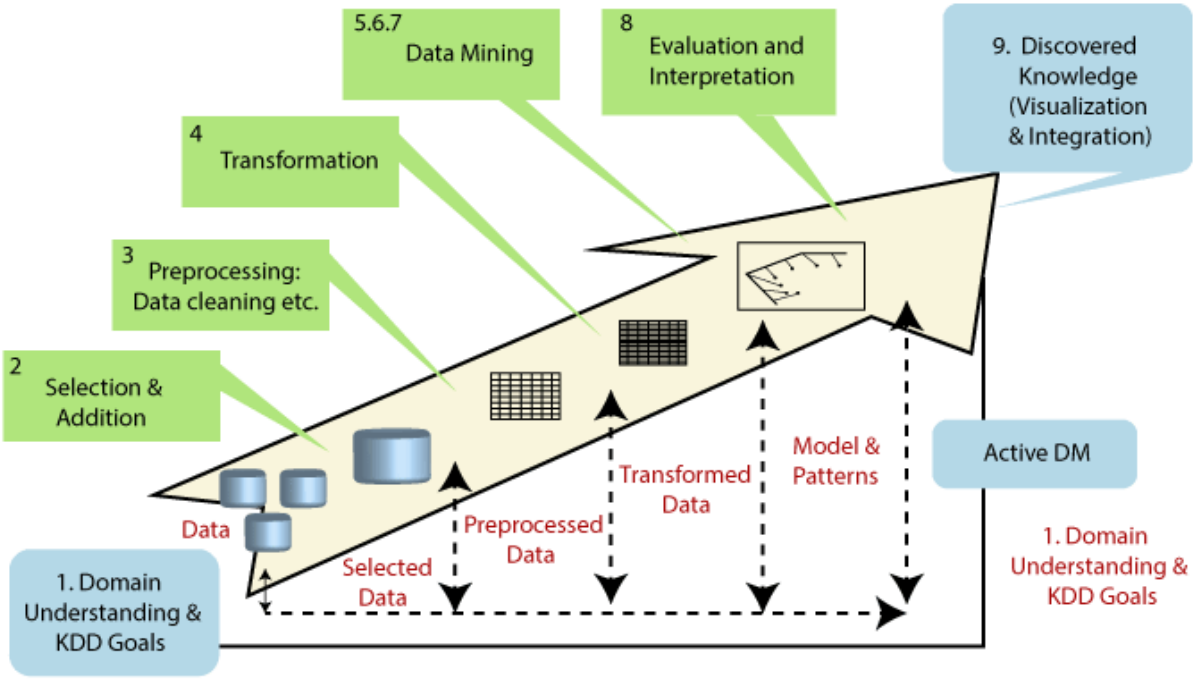


Figure 04: KDD process in Data Mining

Data Mining is the root of the KDD procedure, including the inferring of algorithms that investigate the data, develop the model, and find previously unknown patterns. The model is used for extracting the knowledge from the data, analyse the data, and predict the data. [30]

KDD involves looking in the data for such factors as:

- Associations: things done together (e.g., buying groceries).
- Sequences: events occurring over time (e.g., house, refrigerator).
- Classifications: rules for recognizing patterns (e.g., days on which restaurant Customers arrive and what they order).
- Clusters: defining new groups (e.g., market segments).
- Forecasting: predictions from time series (e.g., stock market fluctuation).[31]

2.4. Techniques of Data Mining

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

2.4.1. Classification :

Classification is concentrating on data mining techniques that are being used for such purposes. [32]

Being generic as it must try to integrate as many learning algorithms as possible. Meanwhile the system must be capable of generating, by means of meta-learning, a decision mechanism and so being able to decide the most adequate algorithm for each data mining task, depending on basic features of the data set, requirements of the user and the background knowledge acquired on previous data mining sessions. [33] For example, we can build a classification model to categorize bank loan applications as either safe or risky.

2.4.2. Prediction :

The prediction as its name implied is one of the data mining techniques that discovers relationship between independent variables and relationship between

dependent variables. [34] For example, in credit card fraud detection, history of data for a particular person's credit card usage has to be analysed. If any abnormal pattern was detected, then it should be reported as 'fraudulent action'.

2.4.3. Association :

Association is one of the best-known data mining techniques. In association, a pattern is discovered based on a relationship of a particular item on other items in the same transaction. For example, Fast-food chains learned very early in the game that people who buy fast food tend to feel thirsty due to the high salt content and end up buying Coke. They exploited this by creating combo meals that offer you food that is bound to make you thirsty with Coke as part of the meal.

2.4.4. Clustering :

Clustering is "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. [35] As an example. In a library, there is a wide range of books in various topics available. The challenge is how to keep those books in a way that readers can take several books in a particular topic without hassle.

2.4.5. Regression :

Regression is a data mining and ML technique used to fit an equation to a dataset, typically used to predict a range of numeric values or continuous values. The simplest form of regression, linear regression, it uses the formula of a straight line ($y = ax + b$) and determines the appropriate values for m and b to predict the value of y based upon a given value of x[1,6]. [36]

2.4.6. Outlier detection :

Outlier detection is one of the very interesting arising recently in the data mining research. Recently, a few studies have been conducted on outlier detection for large datasets [37], for example, Intrusion detection in computer networks.

3. Association Rules

Association rules in data mining gained a lot of attention from researchers of the field, and many methods and algorithms were developed for it, it indicates the probability of associations between items in massive datasets of wide range of data sources. It creates a link between grouped items, drawing rules that define these links among item sets. They are commonly used in medical data sets or to find sale correlations in transactional data. For example, after checking a supermarket's data, we noticed that if a shopper buys Meat and Rice together, they are likely to buy Juice, the rule {Meat, Rice} => {Juice} can be used in making decisions like promotional pricing or product placement, to raise sales.

3.1. History of Association Rules

While the concept can be traced back earlier, Association rule mining was created in the 1990s by computer scientists Rakesh Agrawal, Tomasz Imieliński and Arun Swami. The scientists developed an algorithm-based way to find relationships between items using point-of-sale systems (POS) and it was more popularized in the 1993 as their article "Mining Association Rules between Sets of Items in Large Databases" was published. [37]

3.2. Definition of Association Rules

Association rule mining is a technique for discovering and detecting common patterns, correlations, and frequent links in datasets stored in a variety of large datasets, including relational databases, transactional databases, and other repositories.

3.3. Advantages

- Understandable: Easy to understand
- Useful: Helps in making accurate decisions
- Efficient: Search algorithms
- Several Application fields

3.4. Application areas

There are various application areas in which association rules can be applied in order to extract useful information from huge dataset, “Association rule mining” or “mining associations” are terms used to describe the process of employing association rules.

Here are some examples of how they may be employed to solve the real-world problems:

- ***Market basket analysis (retail) :***

It’s probably the most know use of association rule mining. In supermarkets, for example, data is gathered using bar-code scanners. Managers would like to know if particular sets of products are purchased together on a regular basis. They might utilize this information to make changes to shop layouts

(placing items that are usually bought together next to each and making promotion according to that).

- ***Medical diagnosis:***

Applying association rules in the medical field to help doctors diagnose patients, since it can be difficult to have precise diagnosis in the presence of noise in training examples, Serban developed a system based on supervised learning methods and relational association rules, to assess the probability of an illness in certain conditions [38].

- ***Protein sequences:***

Proteins are essential components of any organism's biological machinery. Recombinant DNA technology have made it possible to quickly identify DNA sequences and, by extension, protein amino acid sequences from structural genes. Proteins are made up of 20 different amino acid sequences, with each having its own unique three-dimensional structure that is based on the protein's function, and any change in it may change the nature of the protein itself. After so much research in proteins' compositions, it is no longer thought to be random.

Nitin Gupta, Nitin Mangal, Kamal Tiwari, and Pabitra Mitra have identified the nature of the interactions between different amino acids in a protein. [38] Such association rules are improving our understanding of protein composition and providing information on global connections among certain groups of amino acids found in proteins.

- ***Census data:***

Researchers and the general public have the ability to access a wide range of general statistical data about society thanks to censuses. Data mining offers a lot of promise for assisting good public policy and supporting the effective running of a democratic society. [39]

- ***Customer Relationship Management:***

It's basically a central place where organizations could store customer and prospect data, track customer interactions, and share that data with colleagues. It enables businesses to manage customer relationships, allowing them to grow. [40]

3.5. Concepts of Association Rule Mining

In most cases, association rules should simultaneously fulfill especially a user-specified minimum support and a user-specified minimum confidence. The generation of association rules is typically broken into two steps:

First, the bare minimum of support: which is used to locate all frequently occurring itemsets in a database according to a minimum value of support.

Second, rules are formed using frequent itemsets and the minimal confidence constraint.

It's hard to locate all frequent itemsets in database since it requires searching all potential itemsets. The downward-closure feature of support ensures that for a frequent itemset, all of its subsets are equally frequent, and that for an infrequent itemset, all of its supersets must also be infrequent, allowing for efficient search. In the past, algorithms have been used to identify common itemsets through this feature. [41]

3.5.1. Itemsets and association rules:

They are the product of the process of extracting the association rules and they derive their origins from the items

- **Item**: A data item is a single trait of an object or idea in a specific place and at a certain time. Data items are subcomponents of data objects that reflect an object's or concept's atomic state within the provided constraints.[2]
- **Itemset**: A set of **n** items noted: {A, B, C, D …} the set of all possible itemsets formed by the item elements is $\mathbf{2^n}$. For example, all items bought by one customer during one visit to a department store.
- **Frequent itemset**: is a method for market basket analysis. It aims at finding regularities in the shopping behavior of customers of supermarkets, mail-order companies, on-line shops etc. More specifically: Find sets of products that are frequently bought together.[42]
- **Frequent itemset mining:** it was introduced as a means to find frequent groupings of items in a database containing transactions of these items. [43] The database is composed of a series of transactions that are analogous to websites visited by users. These orders are individual histories that are made up of some number of web services. Online retailers, make use of frequent itemsets to suggest additional web services that a user might be interested in, based on their past history and the history of others with similar interests. The following dataset is going to be used for frequent itemset mining, where each line represents a single user’s basket of items.

3.5.2. Metrics of interestingness:

Various measures are used to pick relevant rules, the best-known constraints are minimum thresholds on support and confidence.

Let's the take the following database:

Let $I = \{i_1+i_2+\ldots+i_n\}$ be a set of n binary attributes called items.

Let $T = \{t_1+t_2+\ldots+t_n\}$ be a set of transactions called database.

T_{id}	Jus	Nuts	Diaper	Coffee	Eggs	Milk
1	1	1	1	0	0	0
2	1	0	1	1	0	0
3	1	0	1	0	1	0
4	0	1	0	0	1	1
5	0	1	1	1	1	1

Table 01: Market basket transactions

Let say X, Y are itemsets, X⟹Y is an association rule (set of frequent items), and T is a set of transactions of a database.

a. Support :

The frequency of simultaneous "if/then relationship" appearances of items in datasets.

The support of X with respect to Y, is defined as:

$$Supp(X) = \frac{|\{X \subseteq Y\}|}{|Y|}$$

It's an indicator of the "reliability" of the rule. A rule that has weak support can be observed only by chance. Support is used to eliminate uninteresting rules. [38]

Example:

X= {Diaper,Jus} is an itemset, $Supp${Diaper,Jus} is calculated as follows: $Supp(\{\text{Diaper, Jus}\}) = \frac{3}{5}$ as it appears in 60% of all transactions

b. Confidence :

It is the ratio between the number of transactions containing the item of the rule and the number of transactions containing the items of the condition. It is an indicator of the rule's "precision", of how often the rule has been found to be true.

It is defined as:

$$Conf(X \Longrightarrow Y) = \frac{Supp(X \cup Y)}{Supp(X)}$$

Example:

X ={Diaper}, Y = {Jus} are itemsets, $Conf(X \Longrightarrow Y)$ is calculated as follows:

$Conf(X \Longrightarrow Y) = \frac{3}{4} = 0.75 = 75\%$

Which means 75% of the times a customer buys Diapers, he also buys Jus.

c. Lift :

Lift is the ratio of the observed support to that expected if X and Y were independent.

$$Lift(X \Longrightarrow Y) = \frac{Supp(X \cup Y)}{Supp(X) * Supp(Y)}$$

Example:

The rule {Diaper, Jus}⟹{Eggs} has a lift of: $Lift(X \Longrightarrow Y) = \frac{0.2}{0.6*0.6} = 0.56$

d. *Conviction :*

Conviction can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions.

$$Conv(X \Longrightarrow Y) = \frac{1 - Supp(Y)}{1 - Conv(X \Longrightarrow Y)}$$

3.5.3. Alternative measures of interestingness :

In addition to confidence, other measures of interestingness for rules have been proposed. Some popular measures are:

- All-confidence
- Collective strength
- Leverage

Several more measures are presented and compared. Looking for techniques that can model what the user has known (and using these models as interestingness measures) is currently an active research trend under the name of "Subjective Interestingness."[44]

3.6. Algorithms

Data mining academics have given association rule mining a lot of attention, and as a result, several algorithms have been proposed for it. The three most famous and highly used ones are:

Apriori algorithm

ECLAT algorithm

FP-growth algorithm.

4. Conclusion

After a long period of research and development, association rule mining has become increasingly mature in the design and optimization of frequent pattern mining algorithms and is widely used in Internet, finance, biological information and other fields. But the direction of future research in this field is still challenging. As advances in data science, AI and ML, have occurred since the initial use case for association rules – and more devices generating data – association rules can be used in a broader range applications and areas. As more data is generated, new applications for association rules emerge, allowing for greater and more complex data sets to be analyzed and mined.

Chapter III:

Proposed Solution & Implementation

1. Introduction

Creators build a web service in order to sell to a specific type of client who needs it, and it wouldn't make sense to offer a web service to a client whose interest doesn't align with while showing the web service he needs to someone else, therefore, recommender system's intent is to show the clients relevant content based on their history and past data that they agree to share, but not always are these data available to the system, for a number of reasons: perhaps the client has been hacked before and doesn't feel safe giving access to his personal information or cookies, or maybe it's his first time visiting this particular website, in this case the system has no history, no information to work with, so there won't be any suggestions made, and if there were they would probably be very inaccurate. But the system will mostly have the history of similar past users' access who wanted the same service, and what it can do is use the pattern of similar users' access to narrow down the set of choices suggested to him based on the system's stocked data. Or better yet, use those data to give the most accurate prediction for this user with less effort and time consumed.

Association rules are very important in the data-mining field. They reflect regularities in the co-occurrence of the same items within a set of transactions, with numerous practical applications, including consumer market basket analysis, however it's rarely used in other domains, like recommender systems. It is intended to identify strong rules discovered in databases using different measures of interestingness. [46] Which means it can discover the interesting relations between web services based on a large database of past users and their interactions with the system.

In this section, we want to find a way to apply association rule mining to a set of web services in order to come up with as accurate as possible of a pattern of suggestions for the users, and help reduce the massive amount of web services offered for him to choose from. We compare association rule mining's most

relevant algorithms based on the publications discussed in previous chapters. All three algorithms are coded using Python, a dataset generator is used for our synthetic datasets. Case studies analysing the algorithms are carried out step by step using various synthetic datasets in order to: i) see their performance on datasets having different characteristics, ii) understand the causes of performance differences in different phases, iii) to find the most suitable one for our purpose, these algorithms are: Apriori, ECLAT, FP-Growth, which were previously detailed.

Our plan with this research is to determine the effect of each of Apriori, ECLAT, and FP-Growth algorithms, state the differences of performance and results of these algorithms on different datasets while predicting web service access.

This chapter consists the past related work in this field of research, then we move on to the tools needed for the implementation of the algorithms, the coding language and the datasets used. Then we will present the process implemented as well as comparisons between methods while running some models' comparison with different sized datasets and evaluating the results obtained. Finally, we draw a conclusion about what the best way to go about solving the issues at hand and which algorithm is best used and when, and with what metrics value.

2. Related work

In this part we review recent developments and published research related to our work stated in the introduction above, web service's mining using association rules has been receiving increasing attention from the Data Mining community, it all started with the creation (and evolution) of the Apriori algorithm **(R. Agrawal and R. Srikant in 1994)** it's purpose is to basically find and identify frequent itemsets in a dataset for Boolean association rule, and since then it was further extended and somewhat improved to other algorithms.

And in recent past years, a lot more researchers have shown interest in testing the ability of recommendation methods using association rules:

Renáta & Vajk (2006): The main benefit of their research was the ability to detect frequent patterns about the user's behavior and discover hidden information from large amount of web log data collected by web servers. From which the process of web log mining was born, it showed how frequent pattern discovery tasks can be applied on the web log data in order to obtain useful information about the user's navigational behavior.

Przemyslaw Kazienko (2009): the new approach presented is to discover indirect associations existing between pages that rarely occur together but there are other, "third" pages, called transitive, with which they appear relatively frequently. Two types of indirect associations rules are described in the paper: partial indirect associations and complete ones. The former respect single transitive pages, while the latter cover all existing transitive pages. The presented IDARM* Algorithm extracts complete indirect association rules with their important measure—confidence—using pre-calculated direct rules. Both direct and indirect rules are joined into one set of complex association rules, which may be used for the recommendation of web pages. Performed experiments revealed the usefulness of indirect rules for the extension of a typical recommender list. They also deliver

new knowledge not available to direct ones. The relation between ranking lists created on the basis of direct association rules as well as hyperlinks existing on web pages is also examined.

Santhisree & Damodaram (2010): they presented OPTICS algorithm to find density based clusters on a web usage data on MSNBC.COM website. Clustering web usage data is useful to discover interesting patterns pertaining to user traversals, behavior and their usage characteristics. Moreover, user accesses web pages in an order in which they are interested in. The average of inter cluster and intra cluster distance was calculated and the results were then compared with different similarity measures like Euclidean, Jaccard, projected Euclidean, cosine and fuzzy similarity to find the similarity between clusters and the results were visualized graphically to predict the user behavior.

Sanchez & Liu (2011): they used a Bayesian hierarchical model of the page counts per user to obtain posterior distributions of page access frequency that allow us to cluster user sessions in a relatively small number of groups. The proposed approaches were then generalized to different types of Web sites, different levels of aggregation of pages and different clustering schemes.

Mary & Malarvizhi (2012) proposed a combination of both weighted Apriori and dynamic programming to maintain navigation order, the dataset used in the study was MSNBC dataset. The authors extended association rule mining by assigning significant weights to the pages based on visiting order of each page. The system performance was evaluated based on time and space. The experimental result proved that the method guaranteed 35% support with 64% confidence, which was better than the conventional association rule models. Which showed that the proposed technique enhanced the web site effectiveness, increased the user browsing knowledge, improved the prediction accuracy and decreased the computational complexities.

Tassa (2014): he proposed a protocol for secure mining of association rules in horizontally distributed databases. This protocol, is based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. (PDIS 1996), which is a version of the Apriori algorithm that is not secured. It consists of two novel secure multi-party algorithms, one that computes the union of private subsets that each of the interacting players hold, and another that tests the inclusion of an element held by one player in a subset held by another. This protocol offers enhanced privacy. In addition, it is simpler and is significantly more efficient in terms of communication rounds, communication cost and computational cost.

Zhang & Chen (2014): they presented a forensics method of web browsing behavior based on association rule mining. The method aimed at providing the necessary data support to build the behavior pattern library for investigation. The obtained original data was pre-treated to transactional data. Frequent browsing time and frequent web browsing sequences were obtained from the transactional data by Apriori algorithm. The mining results proved helpful to identify and recognize anonymous or suspicious web browsing behavior patterns.

Timur Osadchiy, Ivan Poliakov, Patrick Olivier, Masie Rowland, Emma Foster (2018): unlike the usual Recommender systems which are based on methods such as collaborative and content-based filtering rely on extensive user profiles and item descriptors as well as on an extensive history of user preferences, they described a recommender algorithm that builds a model of collective preferences independently of personal user interests and does not require a complex system of ratings. The performance of the algorithm is analyzed on a large transactional data set generated by a real-world dietary intake recall system.

3. Problematic & Objectives

Web services became an effective way to navigate through the web and find what we're looking for. Regardless, there are millions and millions of web pages available, and it's never easy to find the service a user specifically wants, and that's where recommender systems come to play, their job is to make an accurate guess on what the user want, and provide suggestions based on patterns drawn from information in the web service's database. Predictions will be impossible to make if the information is insufficient or non-existent in some cases.

In this work, we address the following issue: how can we use the data a web service currently has, in order to detect patterns of past user's behavior and make predictions for first time user's access with no history, or users who won't feel comfortable giving the web service access to their profiles?

This project aims to identify consumer preferences and accurately suggest relevant items (e.g., products, services, content) Web pages that seem useful to the unknown user, and also help people finding information on huge online resources. Using association rules, it can help to assist service publishers in the categorization task, in order to reduce the effort required, and promote globally consistent classification decisions, even when several users are involved. Users will put a query and an automatic classifier will determine the most suitable categories where to look for the needed functionality. As a result, both service providers and consumers will be able to exploit Web Services technologies in a better manner.

4. Solution Overview

This research does not focus on user's profiles, in fact we don't need any personal information, no history nor cookies, no need for identification, what we need is a dataset with users' access history of throughout time, and use association analysis technique to find a model that predicts a user's access to existing web services, of course we use historical data user's access to web services, and association rule algorithms to find the patterns we need.

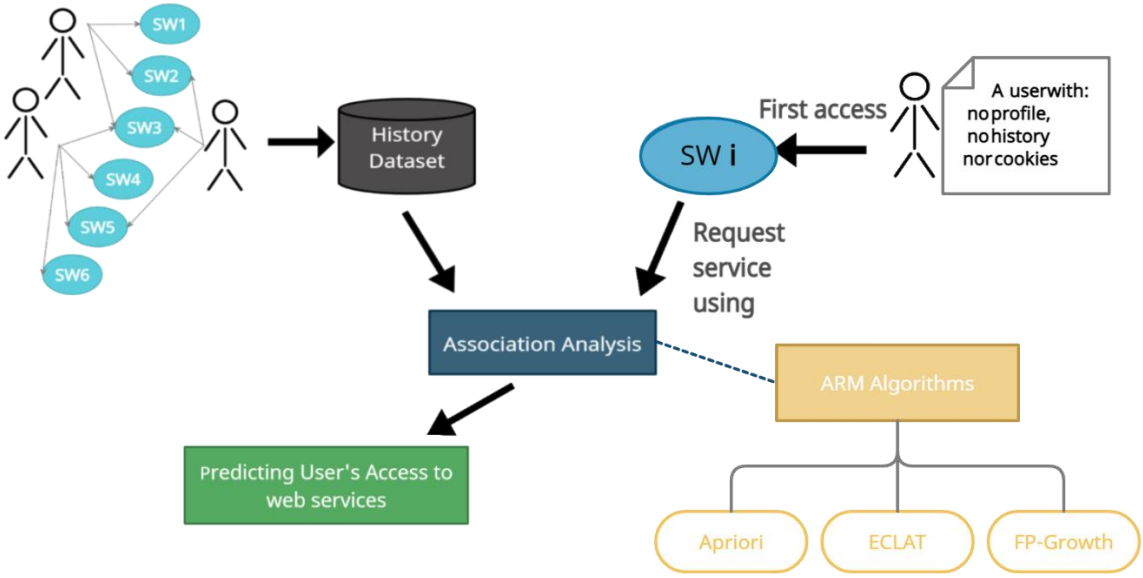


Figure 05: Framework of web service prediction

4.1.1. Apriori algorithm :

Apriori algorithm is a breadth-first search strategy, that utilizes the downward closure property of support.[45] Put simply, the Apriori principle states that. if an itemset is infrequent, then all its supersets must also be infrequent. This means that if {Diaper} was found to be infrequent, we can expect {Diaper, Jus} to be equally or even more infrequent. The Apriori principle can reduce the number of itemsets we need to examine.

- *Process:*

When using Apriori algorithm, first step is to find itemsets with support, starting with itemsets of a single item, then we eliminate the ones with support lower than the minimum support, after that we generate all the possible itemsets with the ones left, and repeat the process again and again until there are no newer itemsets.

The Apriori algorithm considers any subset of a frequent itemset to also be a frequent itemset. With this approach, the algorithm reduces the number of candidates being considered by only exploring the itemsets whose support count is greater than or equal to the minimum support count.

- *Apriori algorithm organizational chart:*

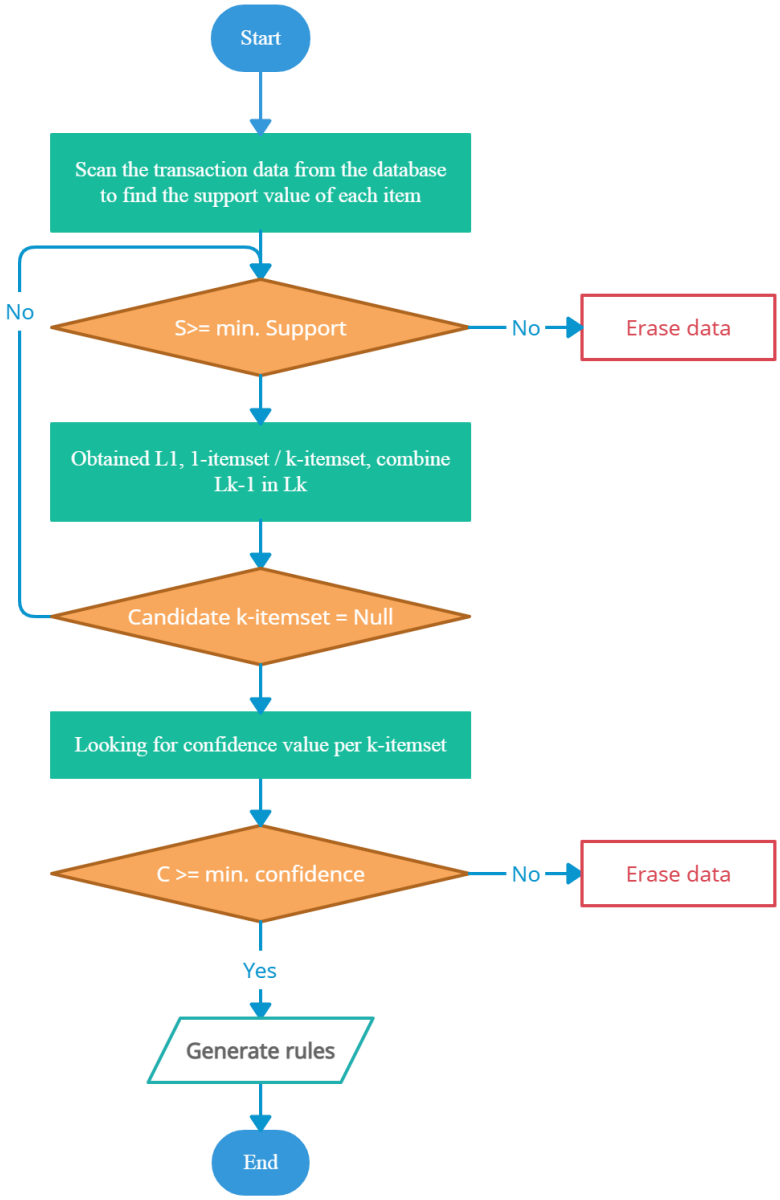


Figure 06: Apriori algorithm's organizational chart

- *Apriori Frequent itemset Algorithm:*

1: **INPUT:** A file $\mathcal{D}$ consisting of baskets of items, a support threshold σ.
2: **OUTPUT:** A list of itemsets $\mathcal{F}(\mathcal{D}, \sigma)$.
3: **METHOD:**
4: $C_1 \leftarrow \{\{i\} | i \in \mathcal{J}\}$
5: $k \leftarrow 1$
6: **while** $C_k \neq \{\}$ **do**
7: # Compute the supports of all candidate itemsets
8: **for** all transactions $\{tid, I\} \in \mathcal{D}$ **do**
9: **for** all candidate itemsets $X \in C_k$ **do**
10: **if** $X \subseteq I$ **then**
11: $X.support$++
12: # Extract all frequent itemsets
13: $F_k = \{X | X.support > \sigma\}$
14: # Generate new candidate itemsets
15: **for** all $X, Y \in Fi, X[i] = Y[i]$ for $1 \leq i \leq k-1$, and $X[k] < Y[k]$ **do**
16: $I = X \cup \{Y[k]\}$
17: **if** $\forall J \subset I, |J| = k : J \in F_k$ **then**
18: $C_{k+1} \leftarrow C_{k+1} \cup I$
19: k++

[46]

Figure 07: Apriori algorithm

4.1.2. ECLAT algorithm :

It’s stands for Equivalence Class Transformation; it’s considered a more efficient and scalable version of the Apriori algorithm. While the Apriori algorithm works in a horizontal sense imitating the Breadth-First Search of a graph, the ECLAT algorithm works in a vertical manner just like the Depth-First Search of a graph. This vertical approach of the ECLAT algorithm makes it a faster algorithm than the Apriori algorithm. It is suitable for both sequential as well as parallel execution with locality-enhancing properties.

- *Process:*

The main idea is to compute a candidate's support value using Transaction Id Sets (tidsets) intersections rather than generating subsets that don't exist in the prefix tree. All single items, as well as their tidsets, are utilized in the function's initial call. The function is then executed repeatedly, with each recursive call verifying and combining item-tidset pairings with additional item-tidset pairs. This method is repeated until there are no more item-tidset pairings to merge.

- *ECLAT algorithm organizational chart:*

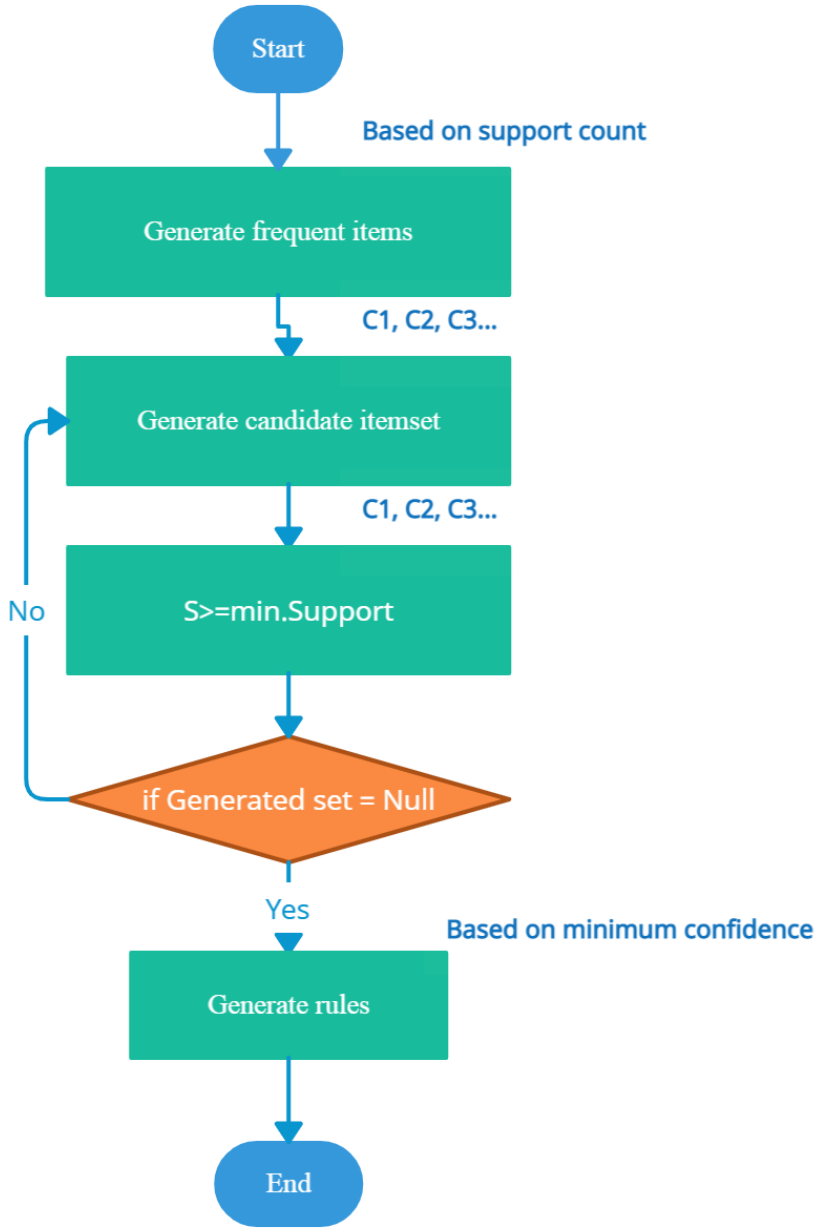


Figure 08: ECLAT algorithm's organizational chart

- *ECLAT Frequent Itemset Algorithm:*

1: **INPUT:** A file $\mathcal{D}$ consisting of baskets of items, a support threshold σ, and an item prefix I, such that $I \subseteq \mathcal{J}$.
2: **OUTPUT:** A list of itemsets $\mathcal{F}[I](\mathcal{D}, \sigma)$ for the specified prefix.
3: **METHOD:**
4: $\mathcal{F}[i] \leftarrow \{\}$
5: **for** all $i \in \mathcal{J}$ occurring in $\mathcal{D}$ **do**
6: $\mathcal{F}[I] := \mathcal{F}[I] \cup \{I \cup \{i\}\}$
7: # Create $\mathcal{D}_i$
8: $\mathcal{D}_i \leftarrow \{\}$
9: **for** all $j \in \mathcal{J}$ occurring in $\mathcal{D}$ such that $j > i$ **do**
10: $C \leftarrow \text{cover}(\{i\}) \cap \text{cover}(\{j\})$
11: **if** $|C| \geq \sigma$ **then**
12: $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{j, C\}$
13: # Depth-first recursion
14: Compute $\mathcal{F}[I \cup i](\mathcal{D}_i, \sigma)$
15: $\mathcal{F}[I] := \mathcal{F}[I] \cup \mathcal{F}[I \cup i]$

[46]

Figure 09: ECLAT algorithm

4.1.3. FP-growth algorithm :

The FP-Growth Algorithm, proposed by Han in, is a step forward from Apriori approach, without the need for candidate generation, a common pattern is generated. The database is represented by the FP growth method as a frequent pattern tree a frequent or FP tree.

A new way of searching for frequent itemsets in a database has been developed by the National Science Foundation. The method involves creating a tree structure that will maintain the association between the itemsets. This tree structure is called a "pattern fragment" and breaks down the frequent items into fragments.

FP tree: Frequent Pattern Tree is a tree-like structure that is made with the initial itemsets of the database. The purpose of the FP tree is to mine the most frequent pattern. Each node of the FP tree represents an item of the itemset.

- *Process*

The root node represents null while the lower nodes represent the itemsets. The association of the nodes with the lower nodes that is the itemsets with the other itemsets are maintained while forming the tree.

The construction of a FP-tree is subdivided into three major steps.

Step 1 — Scan the data set to determine the support count of each item, discard the infrequent items and sort the frequent items in decreasing order.

Step 2 — Scan the data set one transaction at a time to create the FP-tree. For each transaction:

a. If it is a unique transaction form a new path and set the counter for each node to 1.
b. If it shares a common prefix itemset then increment the common itemset node counters and create new nodes if needed.

Step 3 — Repeat this process for all the transactions in the dataset. Finally, after traversing the whole dataset, we will get an FP-Tree and a header table.

- *FP-Growth algorithm organizational chart:*

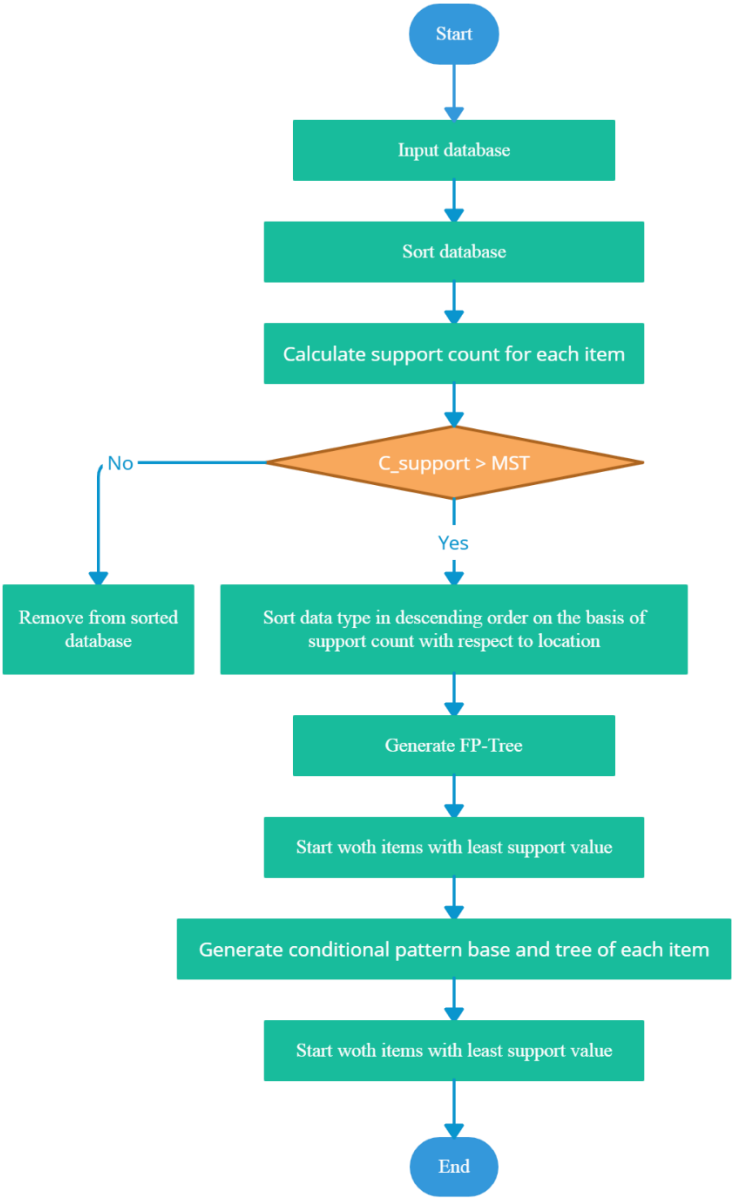


Figure 10: FP-Growth algorithm's organizational chart

- *FP-Growth Frequent Itemset Algorithm:*

1: **INPUT:** A file $\mathcal{D}$ consisting of baskets of items, a support threshold σ, and an item prefix I, such that $I \subseteq \mathcal{J}$.
2: **OUTPUT:** A list of itemsets $\mathcal{F}[I](\mathcal{D}, \sigma)$ for the specified prefix.
3: $\mathcal{F}[i] \leftarrow \{\}$
4: **for** all $i \in \mathcal{J}$ occurring in $\mathcal{D}$ **do**
5: $\quad \mathcal{F}[I] \leftarrow \mathcal{F}[I] \cup \{I \cup \{i\}$
6: # Create $\mathcal{D}_i$
7: $\quad \mathcal{D}_i \leftarrow \{\}$
8: $\quad H \leftarrow \{\}$
9: $\quad$ **for** all $j \in \mathcal{J}$ occurring in $\mathcal{D}$ such that $j > i$ **do**
10: $\quad\quad$ **if** support$(I \cup \{i, j\}) \geq \sigma$ **then**
11: $\quad\quad\quad H \leftarrow H \cup \{j\}$
12: $\quad$ **for** all $(tid, X) \in \mathcal{D}$ with $I \in X$ **do**
13: $\quad\quad \mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(tid, X \cap H)\}$
14: # Depth-first recursion
15: $\quad$ Compute $\mathcal{F}[I \cup \{i\}](\mathcal{D}_i, \sigma)$
16: $\quad \mathcal{F}[I] \leftarrow \mathcal{F}[I] \cup \mathcal{F}[I \cup \{i\}]$

[46]

Figure 11: FP-Growth algorithm

Others not-as-famous algorithms also include.

- *ASSOC:*

The ASSOC procedure which is a mining method for generalized association rules using fast bitstrings operations. The association rules mined by this method are more general than those output by Apriori. For example, "items" can be connected both with conjunction and disjunctions, and relations between antecedent and consequent can be arbitrary.

- *OPUS search :*

A fast rule discovery technique that does not rely on monotone or anti-monotone constraints like minimal support. It has been extended to find rules with any item as a consequent. [45]

5. Implementation & Tests :

In this section, we give brief information about the implementation of algorithms. Algorithms. The different algorithms that are used to mine frequent itemsets are many. This work shows how Apriori, ECLAT and FP-Growth address the issue at hand and compares the shortcomings of them. All three algorithms must calculate statistics about itemsets that might ultimately be included in the final collection of frequent itemsets. One statistic that is common to all three of these algorithms is support. Another is confidence, both of which we're going to use to calculate the performance of algorithms.

5.1. Development Environment:

In order to make our plan come to life, we're using Python, the most used language in the field of Data Science and Machine Learning. Choosing a good programming environment is very important for the development of projects. And after much research we found that python is the best choice to sole the issue at hand.

5.1.1. Materials :

We operated and implemented our program in a Windows operating system of 64bits, 7th version, Intel Core i3-5005U CPU @ 2.00 GHz, 8.00 GB of Ram.

5.1.2. Programming Language:

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. With its extensive standard -freely available- library and easily extended functions and data types [47], it is perfectly suitable for a project like the one at hand.

5.1.3. Integrated Development Environment:

We used Colaboratory, or "Colab" for short, which is a web IDE for python from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs. [48]

5.2. Data Description

For the experimental study, the datasets were generated using an application of work done in 2017 [49] with various sizes. After processing the databases, each sequence corresponds to the history of a user's access and each event in the sequence corresponds to a user's request for a page. Datasets are clear and processed efficiently for handling missing or redundant attributes, to get the best outcome from the Data Mining process.

The construction of the dataset containing the users and the web services they have accessed to, is done in four processes:

- Profile creation process.
- Process for calculating similarity between users and web services.
- Processes for generating and filtering user access to web services.
- Community generation process.

All these processes are described in [49]

To have a clear idea of the algorithms used and the role of the dataset in the generation of the prediction model, we have used several datasets with different sizes described in the table below comprises so that we get results for most possible cases:

Users	Web service		
10k	7	10	20
30k	7	10	20
50k	7	10	20
100k	7	10	20

Table 02: Details of the used datasets

After pre-processing, the datasets are stored in "csv" files, the users are numbered from 1 in the rows, while the visited web services fill the columns. The cells pertaining to each row provide information on all the web services that each user visited.

To have a clear idea of the algorithms used and the role of the dataset in the generation of the prediction model, we have used several datasets with different sizes described in the table above comprises so that we get results for most possible cases.

	Blog	Books	Business	Culture	Economy	Entertainment	Finances	Games	Health	Lifestyle
0	True	False	False	True	True	False	True	False	True	True
1	False	True	False	True	False	True	True	True	True	False
2	True	True	True	False	True	True	False	True	True	False
3	True	False	True	False	True	False	False	True	True	False
4	True	False	True	False	True	False	False	True	False	False
...	...	...	...	...	...	...	...	...	...	...
99995	True	True	False	True	True	False	True	True	True	False
99996	True	False	True	True	True	True	True	True	True	True
99997	False	True	True	True	False	True	True	True	True	False
99998	False	True	True	True	False	False	True	True	False	True
99999	True	False	False	False	True	False	False	True	True	False

100000 rows × 20 columns

Figure 12: A sample of the test dataset

5.3. Tests & Results

5.3.1. Runtime with different minimum support values:

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

The first step is to check how the minimum support infers the runtime with datasets of fixed number of services and varying number of transactions.

The following databases are generated with diversity of items:

- ✓ Fixed web service number: 7, 10, and 20.
- ✓ Varying transactions number: 10000, 30000, 50000, 100000.
- ✓ Average size of transactions: 4.
- ✓ Average size of patterns: 2.

While we vary the minimum support [0.1, 1] with each algorithm and we calculate the runtime.

- Starting with datasets of 7 web services.

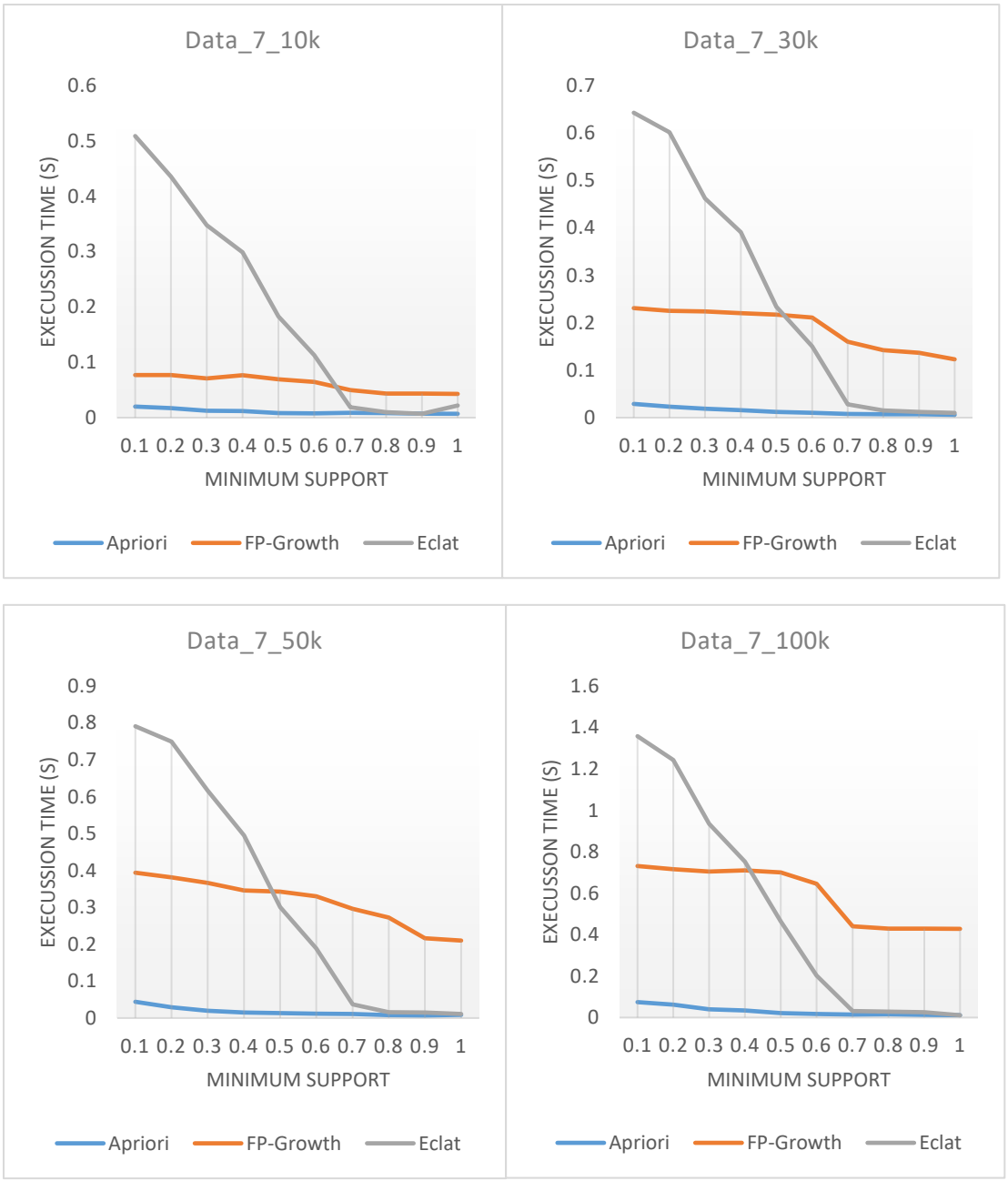


Figure 13: Runtime of algorithms on datasets with 7 web services

It is clear that increase in minimum support decreases the time of execution of all three algorithms. The total performance, we can see that Apriori performance is the fastest followed by FP growth, while ECLAT's performance is the slowest. It is seen that their performance becomes constant on the second phase of execution for minimum support values above 0.7 but at different levels of time.

- The next set of data is with 10 web services:

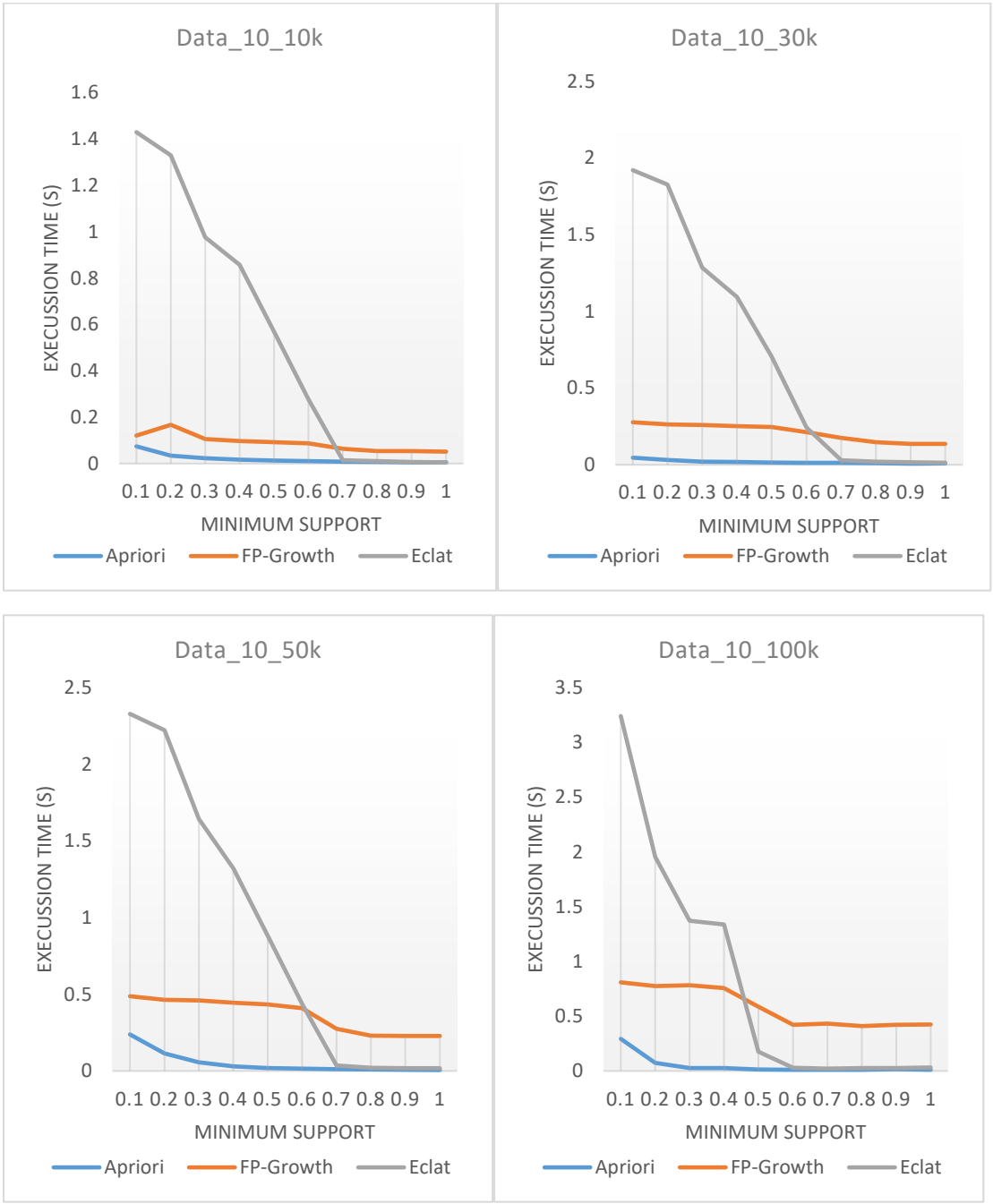


Figure 14: Runtime of algorithms on datasets with 10 web services

The runtime starts slow with all algorithms, and it gets faster as the minimum support increase, and the performance time gets constant as it hits the 0.7 mark with each algorithm, with Apriori being the fastest of all three, the results are almost the same as the charts above.

- The last set of data is with the largest number of web services: 20

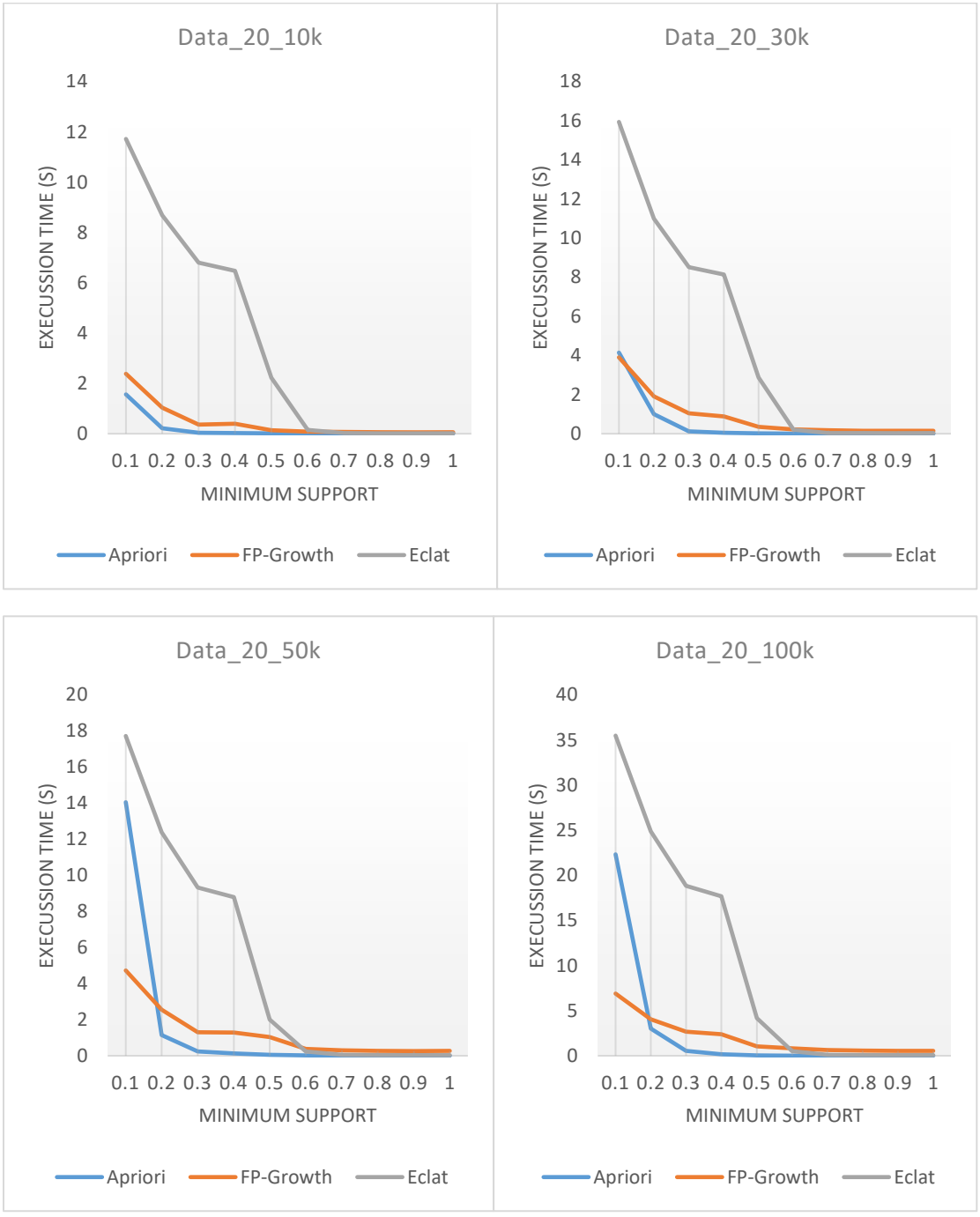


Figure 15: Runtime of algorithms on datasets with 20 web services

In this case we can see a difference in performance, FP-growth started faster as the support value is low, as the support value increases FP-growth becomes slower and Apriori becomes faster, ECLAT is the slowest in both cases.

FP-Growth better performance can only mean that it works faster with bigger datasets and with low minimum support value.

b. Varying web services number with fixed transaction number:

The second comparison is to check how the minimum support infers the runtime with datasets of varying number of web services and fixed number of transactions.

The following databases consist of a fixed transaction number of 75000, and a varying web service number of 7 and 20, average size of transactions is still 4, average size of patterns=2, while minimum support value increases from 0.1 to 1.

The following databases consist of:

- ✓ Fixed transactions number: 10000, 30000, 50000, 100000.
- ✓ Varying web service number: 7, 10, 20.
- ✓ Average size of transactions: 4.
- ✓ Average size of patterns: 2.

While we vary the minimum support [0.1, 1] with each algorithm and we calculate the runtime.

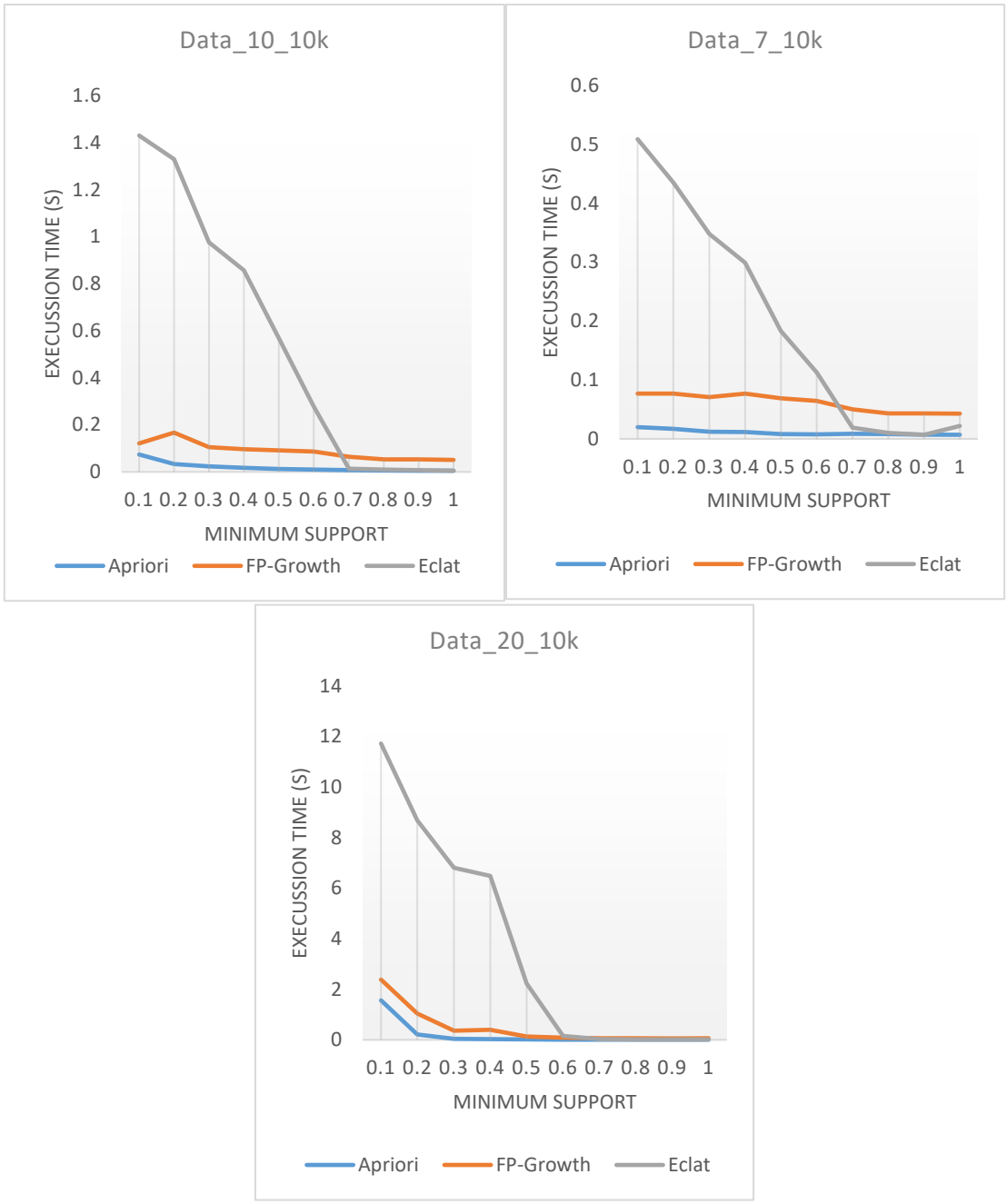


Figure 16: Runtime with minimum support on Data with 10k

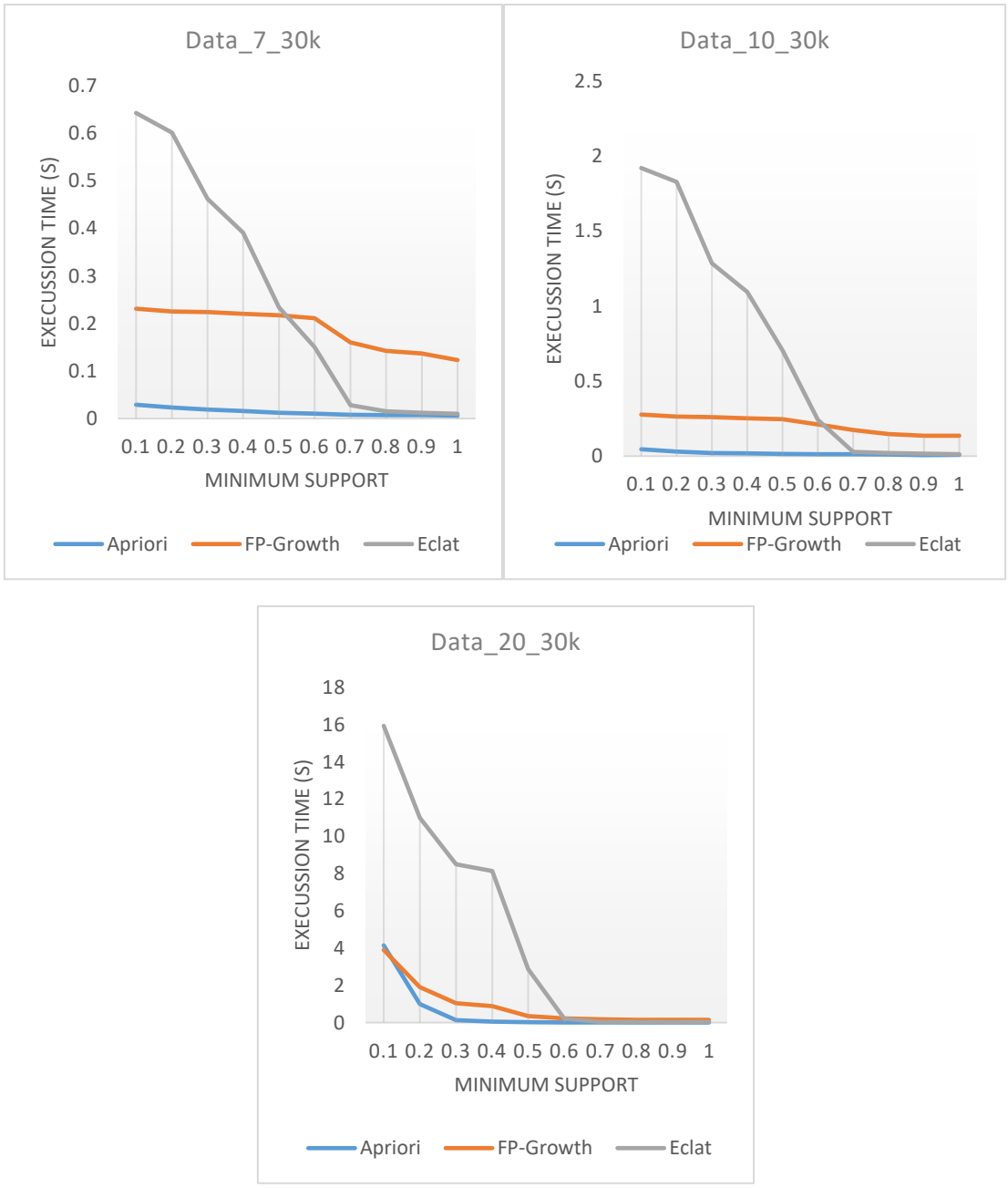


Figure 17: Runtime with minimum support on Data with 30k

The total performance in figures 14 and 15 shows that even with the same transaction numbers, the data with a big number of web services take longer to process.

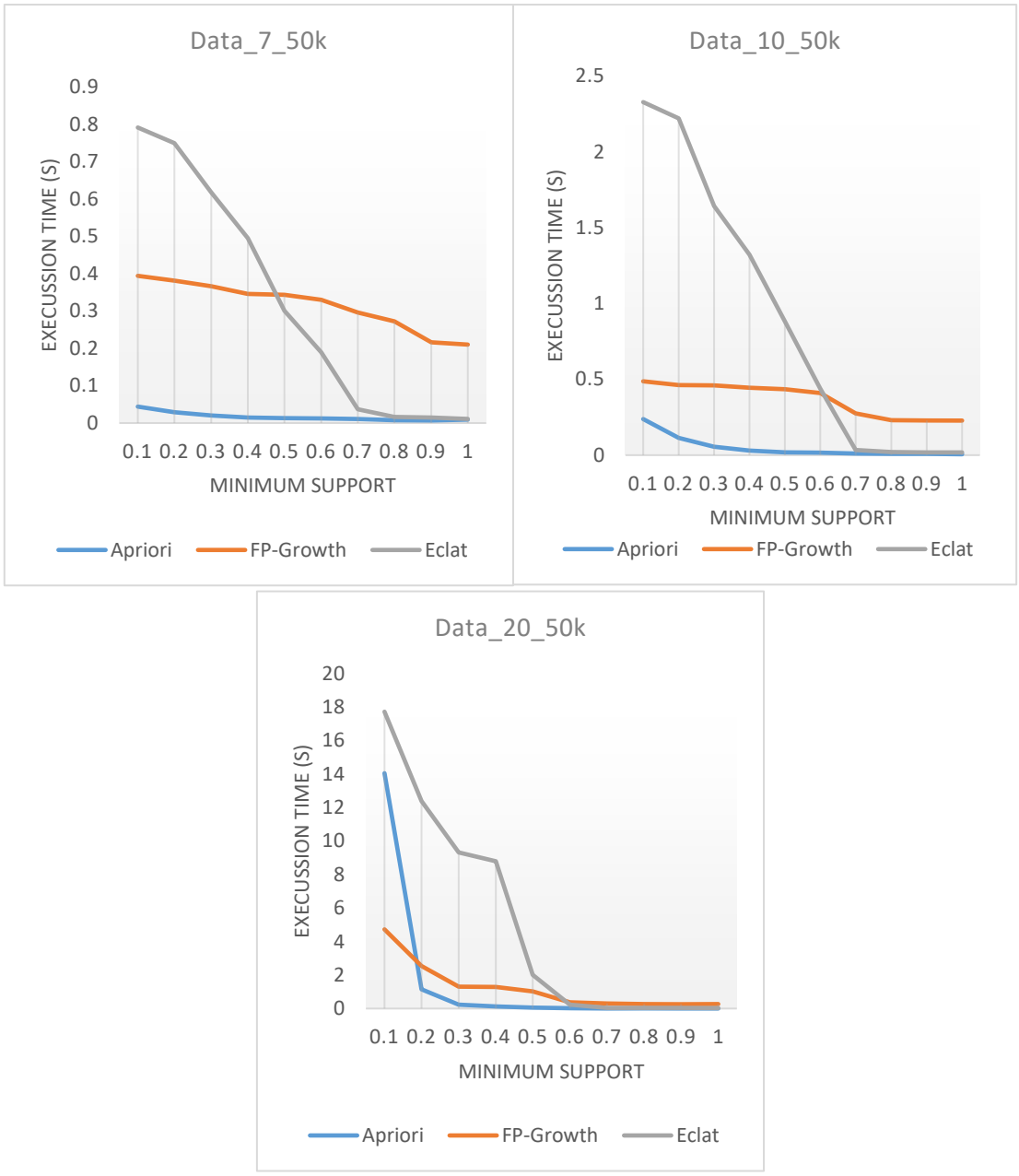


Figure 18: Runtime with minimum support on Data with 50k

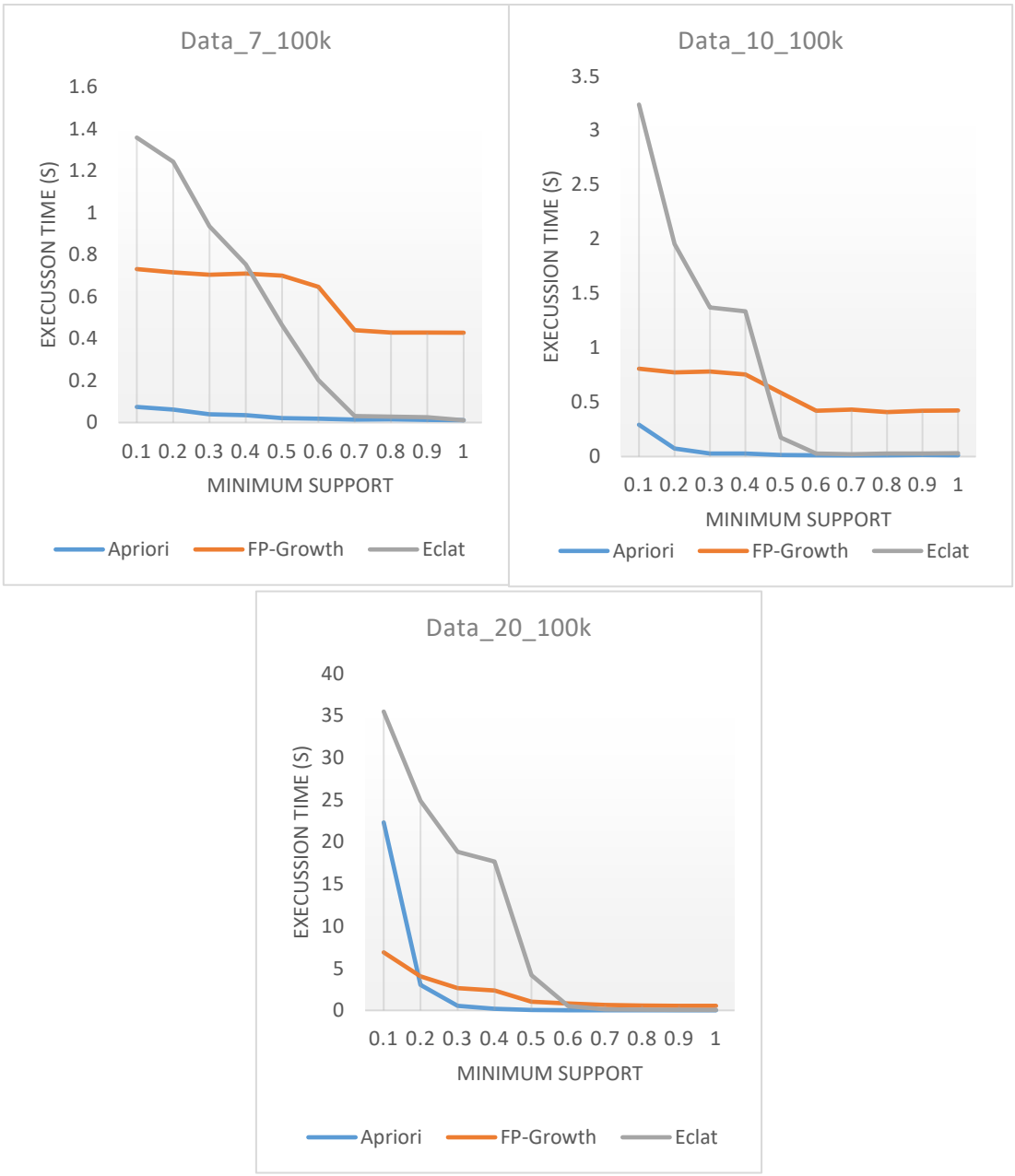


Figure 19: Runtime with minimum support on Data with 100k

With the larger sets, and unlike the previous charts, we can see a change in performance of FP-Growth on dataset with both large number of web services (20) and large number of transactions (50k and 100k), FP-Growth got faster especially at beginning of the execution as the support value was low. And it got slower again as the value increases.

5.3.2. Runtime with different number of transactions:

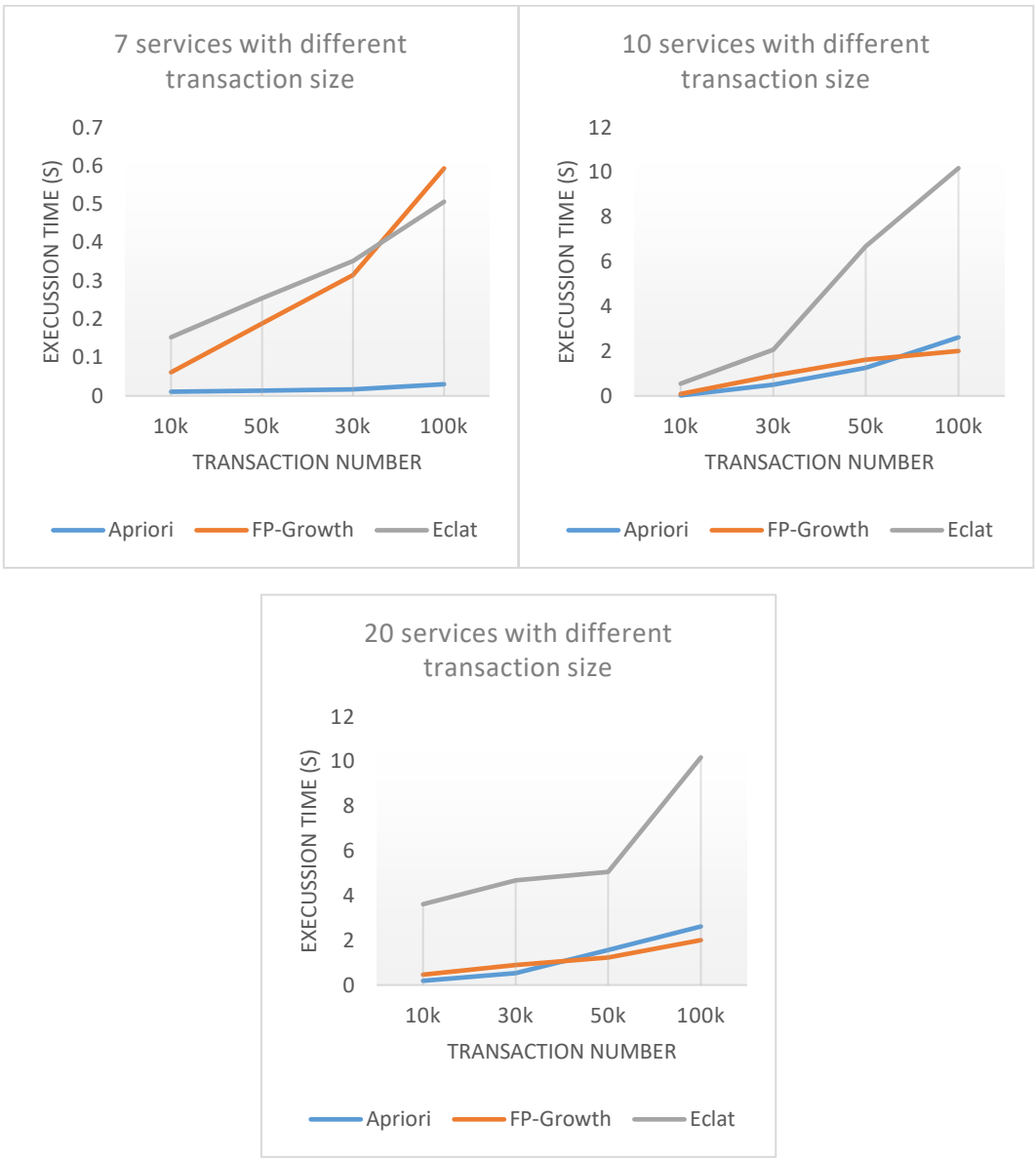


Figure 20: Runtime with different number of transactions

The runtime of all three algorithms increases as the number of itemsets goes up. However, we can clearly see that the slope of the increase ECLAT is way higher. And that of FP-Growth runs overall runs faster when the dataset is large. While the rise in Apriori's slope was constant with time. These observations show how both FP Growth and ECLAT algorithms are more sensitive to the itemsets and data size compared to Apriori.

5.3.3. Runtime with different minimum confidence values :

Using the dataset of 20 web services and 100000 transactions with a minimum support threshold of 0.6 to compare the performance of the three algorithms when it comes to generating association rules

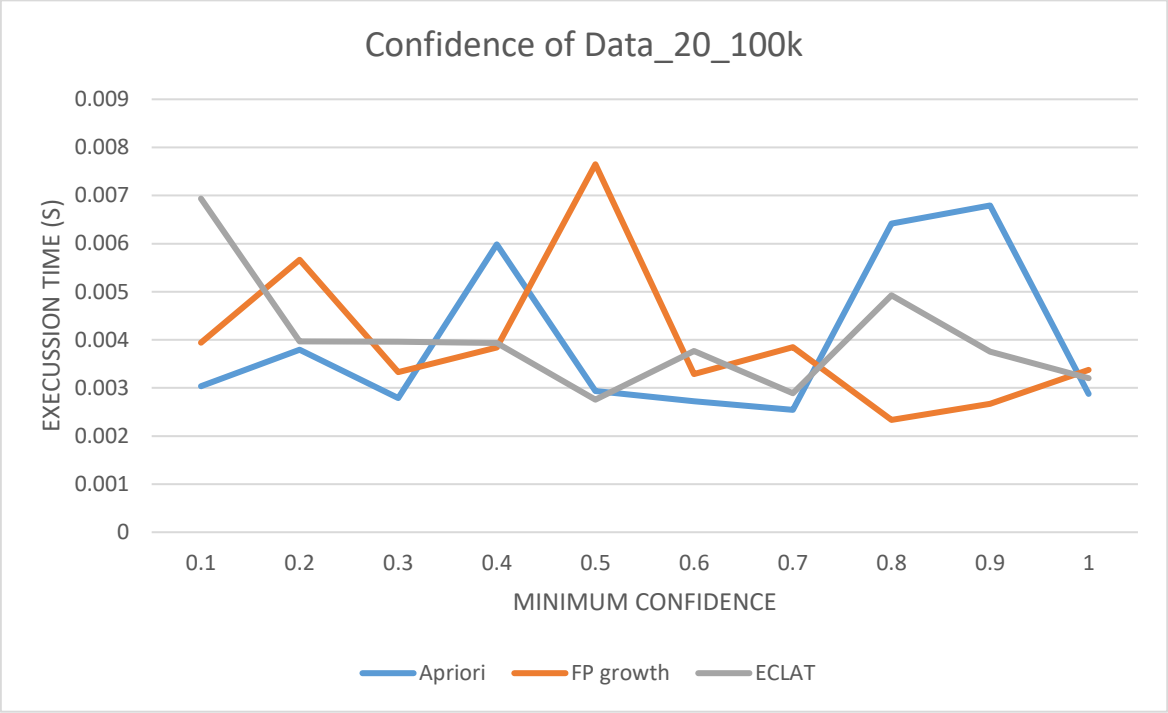


Figure 21: Runtime of different minimum confidence values

Algorithm	Average Runtime
Apriori	**0,0039893**
FP-Growth	**0,0039941**
ECLAT	**0,0039996**

Table 03: Average Runtime of minimum confidence values

From table 03, we can see that the average execution time of Apriori, FP-Growth and ECLAT are almost the same, which was expected as all three algorithms use the same function and metric “confidence”.

5.4. Discussion

As a result of the experimental study of the datasets above, it's clearly revealed that all three algorithms behave differently around different datasets.

5.4.1. Algorithm performance:

a. ECLAT:

It is prominently clear that ECLAT algorithm is not the way to go, as the experiments showed, compared to the other two, ECLAT took the longest time. The reason for that according to researchers is that when the dataset is large it takes more memory space to store candidate set, and more processing time is required for intersecting long TID sets. [45] So ECLAT algorithm is not practical for the purpose that we want to achieve.

b. FP GROWTH:

As FP growth is widely used, it was expected to be the best algorithm performing wise (according to previous studies), however we have found a few flaws within it:

i) Even though we established that it performs best with large numbers, we also found that as the attribute number got high, the algorithm may not fit in the shared memory. A problem we have faced with all three algorithms but especially with FP-Growth, when we increased the dataset size to get more info on the algorithm behavior.

ii) The FP Tree is more cumbersome and difficult to build than Apriori, which can be a problem if we're building the program from scratch, luckily there are a few pre-defined libraries that programmers can use. [50]

iii) FP growth is slow with smaller data (Figure 13) and fast with larger ones (Figure 15).

a. APRIORI:

For a reasonably sized dataset, Apriori's performance was good considering the performance of the other two, even though it's known to be slow, it performed well with medium size data, and it slowed down as the data gotten bigger, which was expected (Figure 13 and 14).

5.4.2. Effect of attribute size :

Attribute size specifies the maximum number of items per row. A larger attribute size means that the frequent itemsets will also be larger, this increases the size of the data structures used to hold these itemsets.

These larger data structures require more memory for storage (Figure 22) and greater processing time to traverse them.

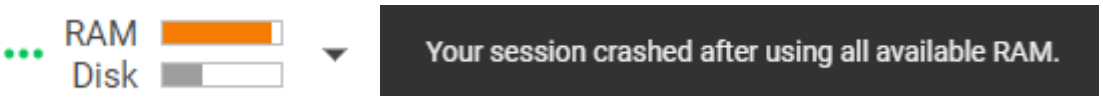


Figure 22: Output of using FP-growth with large dataset

Figure 20 illustrates the effect of increasing transaction size on the performance of the three algorithms.

5.5. Test Run

The data we're using as an example is a synthetic dataset, with 10000 users (rows) by 20 different web services (columns)

```
#Datafarame size // number of users and web services
df_shape = bdd.shape
n_of_users = df_shape[0]
n_of_services = df_shape[1]
df_shape, n_of_users, n_of_services
```

```
((100000, 20), 100000, 20)
```

Figure 23: Details on the dataset's size

The cell values are binary, the value “True” indicates that the users have visited this web service, and the value “False” means that he has not.

For a dataset this size we found that Apriori is the most suitable algorithm to implement so it is going to be the algorithm applied in this example.

We’re using the pre-defined Apriori function in the mlxtend library to find the frequent itemsets using the minimum support as the metrics of elimination, it gives very clear results in the form of dataframe of two columns representing all the frequent itemset found and the minimum support threshold value of each,

The best minimum support threshold is usually between 60% and 70% of the transaction numbers. However, the number of frequent itemsets decreases with the increase in support threshold. Therefore, when the dataset is large it is required to lower the minimum support value to get reasonable results, according the previous comparisons, we’ve found that the best minimum support value is 0.45 or 45%.

```
#Finding frequent itemsets using minumum support threshold

from mlxtend.frequent_patterns import apriori

frequent_itemsets=apriori(df, min_support=0.45, use_colnames=True)
frequent_itemsets
```

Figure 24: Apriori’s function to get frequent itemsets

	support	itemsets
0	0.64172	(Blog)
1	0.50037	(Books)
2	0.49830	(Business)
3	0.60840	(Culture)
4	0.63012	(Economy)
5	0.49910	(Entertainment)
6	0.49998	(Finances)
7	0.76184	(Games)
8	0.50161	(Health)
9	0.49982	(Lifestyle)
10	0.50048	(Media)

Figure 25: Results of Apriori's function

- To get the number of itemsets that were found we use the len() function:

```
len(frequent_itemsets)
```

```
30
```

Figure 26: Length function to get the number of frequent itemsets

- The number of frequent itemsets found with the 45% minimum support threshold is 30.

Now moving along to generating the association rules from these frequent itemsets, using the metric of confidence from the same library, with a value of 0.8 or 80%:

```
#Generating association rules using minumum confidence threshold

from mlxtend.frequent_patterns import apriori,fpgrowth, association_rules

rules=association_rules(frequent_itemsets, metric ="confidence", min_threshold=0.8)
rules
```

Figure 27: ARM's function for generating association rules

- The results were as follows:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Blog)	(Economy)	0.64172	0.63012	0.58071	0.904927	1.436119	0.176349	3.890500
1	(Economy)	(Blog)	0.63012	0.64172	0.58071	0.921586	1.436119	0.176349	4.569103
2	(Media)	(Blog)	0.50048	0.64172	0.49128	0.981618	1.529667	0.170112	19.490432
3	(Books)	(Games)	0.50037	0.76184	0.47640	0.952095	1.249732	0.095198	4.971553
4	(Books)	(Restaurants)	0.50037	0.63937	0.49159	0.982453	1.536595	0.171668	20.552213
5	(Finances)	(Culture)	0.49998	0.60840	0.47370	0.947438	1.557262	0.169512	7.450235
6	(Media)	(Economy)	0.50048	0.63012	0.48514	0.969349	1.538357	0.169778	12.067636
7	(Blog, Economy)	(Media)	0.58071	0.50048	0.47623	0.820082	1.638592	0.185596	2.776381
8	(Blog, Media)	(Economy)	0.49128	0.63012	0.47623	0.969366	1.538383	0.166665	12.074063
9	(Media, Economy)	(Blog)	0.48514	0.64172	0.47623	0.981634	1.529692	0.164906	19.507964
10	(Media)	(Blog, Economy)	0.50048	0.58071	0.47623	0.951547	1.638592	0.185596	8.653454
11	(Books, Games)	(Restaurants)	0.47640	0.63937	0.46788	0.982116	1.536068	0.163284	20.164804
12	(Books, Restaurants)	(Games)	0.49159	0.76184	0.46788	0.951769	1.249303	0.093367	4.937877
13	(Games, Restaurants)	(Books)	0.50780	0.50037	0.46788	0.921386	1.841410	0.213792	6.355514
14	(Books)	(Games, Restaurants)	0.50037	0.50780	0.46788	0.935068	1.841410	0.213792	7.580244

Figure 28: Results of ARM's function

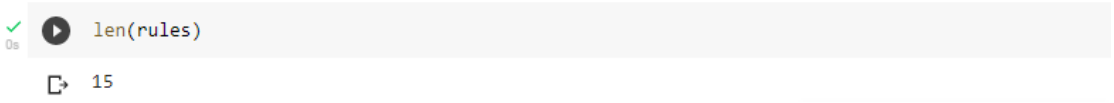


Figure 29: Length function to get the number of association rules

- The table above (Figure 28) is very detailed with many calculations and numbers, to make it more accommodating to our needs, we'll extract only the parts we need:

```
rules= rules[['antecedents', 'consequents', 'support','confidence']]
rules
```

	antecedents	consequents	support	confidence
0	(Blog)	(Economy)	0.58071	0.904927
1	(Economy)	(Blog)	0.58071	0.921586
2	(Media)	(Blog)	0.49128	0.981618
3	(Books)	(Games)	0.47640	0.952095
4	(Books)	(Restaurants)	0.49159	0.982453
5	(Finances)	(Culture)	0.47370	0.947438
6	(Media)	(Economy)	0.48514	0.969349
7	(Blog, Economy)	(Media)	0.47623	0.820082
8	(Blog, Media)	(Economy)	0.47623	0.969366
9	(Media, Economy)	(Blog)	0.47623	0.981634
10	(Media)	(Blog, Economy)	0.47623	0.951547
11	(Books, Games)	(Restaurants)	0.46788	0.982116
12	(Books, Restaurants)	(Games)	0.46788	0.951769
13	(Games, Restaurants)	(Books)	0.46788	0.921386
14	(Books)	(Games, Restaurants)	0.46788	0.935068

Figure 30: A compact version of the association rules list of results

- Taking row number 5 as an example:

The user who visits the web service “Finances” is 94% more likely to visit the web service “Culture”

- Taking row number 12 as another example:

The user who visits both web services “Books” and “Restaurants” is 95% more likely to visit the web service “Games”, and so on.

The decrease of the two metrics will affect the number of results by increases it, however the lower the values of the support and confidence metric the less precise the predictions will be.

- If we change the minimum support threshold to 30% and the confidence to 75%, the number of frequent itemset increases to:

```
len(frequent_itemsets)
```

```
125
```

Figure 31: Length function to get the number of frequent itemsets

- And the number of association rules becomes:

```
len(rules)
```

```
173
```

Figure 32: Length function to get the number of association rules

- And the result table will look like this:

	antecedents	consequents	support	confidence
0	(Blog)	(Economy)	0.58071	0.904927
1	(Economy)	(Blog)	0.58071	0.921586
2	(Blog)	(Media)	0.49128	0.765568
3	(Media)	(Blog)	0.49128	0.981618
4	(Books)	(Games)	0.47640	0.952095
...	...	...	...	...
168	(Games, Tools)	(Restaurants, Books, Shopping)	0.34039	0.778017
169	(Games, Shopping)	(Restaurants, Books, Tools)	0.34039	0.861660
170	(Books, Tools)	(Games, Restaurants, Shopping)	0.34039	0.780443
171	(Books, Shopping)	(Games, Restaurants, Tools)	0.34039	0.892241
172	(Tools, Shopping)	(Games, Restaurants, Books)	0.34039	0.828119

173 rows × 4 columns

Figure 33: A compact version of the second association rules list of results

- Taking row number 170 as a third example:

The user who visits both web services “Books” and “Tools” is 78% more likely to visit the following web services “Games”, “Restaurant”, “Shopping”. However, the chance of that is less likely than the examples mentioned above.

6. Conclusion

In this chapter we presented the different tools used along with the proposed methods of detecting user patterns in detail, we specify the techniques and algorithms used as well as the methods in order to achieve different results by comparing different datasets, we implemented and discussed the obtained results.

As for the algorithms' performance, Apriori is an algorithm for mining frequent itemsets and association rules that is straightforward to understand. As a result, Apriori is a preferred starting point for frequent itemset analysis, it's the safest option. Apriori. However, it has significant scaling difficulties. As a result, Apriori should be avoided when dealing with big datasets. FP-Growth should be considered for most itemset applications. But it exhausts and occupies memory considerably much faster than ECLAT and Apriori.

General Conclusion

General Conclusion

Today there is no second thought about the fact that frequent pattern mining has broad applications which incorporate software bug detection, clustering, classification, and several other problems. And even though the view that we took going into this study is different from all those - web services prediction online-, the major utility of frequent pattern mining is an intermediate tool for providing pattern-centered insights for diverse problems. The purpose of the work was to provide the reader with the complete working circumstances of the Apriori, FP-Growth and ECLAT algorithms with a variety of datasets. The Analyzing of the Algorithms for Frequent Pattern Mining and association rule generating and elaborated advantages and disadvantages associated with algorithms. They have been well utilized to explore the past and now it has reached the stage where its application can effectively predict the future.

Frequent itemset mining is a major topic in the area of active research, new algorithms are often created, as well as old algorithms are improved. For an application where performance quality is critical, it's important to evaluate the dataset with newer algorithms when they're developed and proved to perform better than existing ones.

With this modest work, it has been proven that association rules can be used in many fields including recommendation systems despite the lack of user information.

"**Interesting, but not realistic**." an engineer said about association rule lining, he went on to explain that a transaction with many identical objects is not taken into consideration by these algorithms, they do not take weight into consideration. [50] Also, there are many more subtle circumstances that this method does not account for. That is why businesses tend to not implement it in their operations.

For the upcoming researches, it is possible to extend the research by using the different clustering techniques and also the Association Rule Mining for large number of databases, while taking weight as a key factor. Which is expected to provide a better and more precise results.

Bibliography & Webography

[1]https://www.statista.com/statistics/617136/digital-population-worldwide/ , January 2021.

[2] R. GEETHARAMANI, P. REVATHY, and S. H. O. M. O. N. A. G. JACOB, "Prediction of users webpage access behaviour using association rule mining," Sadhana - Acad. Proc. Eng. Sci., vol. 40, no. 8, pp. 2353–2365, 2015, doi: 10.1007/s12046-015-0424-0.

[3] P. Kazienko, "Mining indirect association rules for web recommendation," Int. J. Appl. Math. Comput. Sci., vol. 19, no. 1, pp. 165–186, 2009, doi: 10.2478/v10006-009-0015-5.

[4] E. Aïmeur, D. Daboue, F. Serge, and M. Onana, "Wsrs: a Web Service Recommender System," pp. 284–289, 2011, doi: 10.5220/0001288102840289.

[5] https://www.marefa.org/Web_services/

[6] https://www.w3.org/TR/ws-arch/, 2004.

[7] https://www.javatpoint.com/restful-web-services-architecture-of-web-services

[8] https://support.novell.com/techcenter/articles/dnd20030304.html, March 1, 2003.

[9]https://www.ibm.com/docs/en/scbn?topic=SSRJDU/gateway_services/soap/SCN_WSG_Protocols.html

[10] https://www.guru99.com/soa-principles.html, August27, 2021.

[11] L. E. Moser and P. M. Melliar-Smith, "Service-Oriented Architecture and Web Services," Wiley Encycl. Comput. Sci. Eng., 2008, doi: 10.1002/9780470050118.ecse510.

[12] https://www.javatpoint.com/service-oriented-architecture

[13] https://onespire.hu/sap-news-en/history-of-recommender-systems/

[14] https://medium.com/the-graph/how-recommender-systems-make-their-suggestions-da6658029b76, March 9, 2017.

[15] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to Weave an Information tapestry,” Commun. ACM, vol. 35, no. 12, pp. 61–70, 1992, doi: 10.1145/138859.138867.

[16] https://en.wikipedia.org/wiki/GroupLens_Research, March 17, 2021.

[17]https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0, December 28, 2017.

[18] X. Li, J. Xing, H. Wang, L. Zheng, S. Jia, and Q. Wang, “A Hybrid Recommendation Method Based on Feature for Offline Book Personalization,” no. Xx, pp. 1–15.

[19]https://medium.com/qloo/popular-evaluation-metrics-in-recommender-systems-explained-324ff2fb427d, April 4, 2019.

[20] https://azati.ai/recommendation-systems-benefits-and-issues/, June 3, 2021.

[21] https://www.sas.com/en_us/insights/analytics/data-mining.html

[22] Hand, David J., and Niall M. Adams. "Data mining." Wiley StatsRef: Statistics Reference Online: 1-7, 2014

[23] https://www.britannica.com/technology/data-mining, December 20, 2019.

[24] Dr. Madan Lal Bhasin, “Data Mining: A Competitive Tool in the Banking and Retail Industries”, The Chartered Accountant October 2006

[25] Vivek Bhambri “Application of Data Mining in Banking Sector”, International Journal of Computer Science and Technology Vol. 2, Issue 2, June 2011

[26] K. Raza, “APPLICATION OF DATA MINING IN BIOINFORMATICS,” vol. 1, no. 2, pp. 114–118.

[27] G. M. Weiss, “Data mining in telecommunications.”

[28] https://bigdata-madesimple.com/14-useful-applications-of-data-mining/, August 20, 2014.

[29] X. Ling, “Data Mining for Direct Marketing: Problems and Solutions,” pp.

73–79.

[30] Hian Chye Koh and Gerald Tan. Data Mining Applications in Healthcare. Journal of Healthcare Information Management — Vol. 19, No. 2, 5-11

[31] https://www.javatpoint.com/kdd-process-in-data-mining

[32] Chung, H. M., & Gray, P. (1999). Special Section: Data Mining. Journal of Management Information Systems, 16(1), 11–16. doi:10.1080/07421222.1999.11518231

[33] Harshna, NavneetKaur, "Survey paper on Data Mining techniques of Intrusion Detection", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 4, April 2013.

[34] J. Botía, M. Velasco, and A. Skarmeta, "A Generic Datamining System. Basic Design and Implementation Guidelines." no. October, 1998.

[35] Rahul Diwate, "Data Mining Techniques in Association Rule: A Review", January 2014.

[36] Kalyani M Raval, "Data Mining Techniques" ISSN: 2277 128X, Ijarcsse, Volume 2, Issue 10, October 2012.

[37] Jiawei Han and Micheline Kamber (2006), Data Mining Concepts and Techniques, published by Morgan Kauffman, 2nd Ed.

[38] R. Agrawal, T. Imieliński, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," ACM SIGMOD Rec., vol. 22, no. 2, pp. 207–216, 1993, doi: 10.1145/170036.170072.

[39] N. Gupta, N. Mangal, K. Tiwari and P. Mitra, "Mining Quantitative Association Rules in Protein Sequences", In Proceedings of Australasian Conference on Knowledge Discovery and Data Mining – AUSDM, 2006

[40] A. Rajak and M. K. Gupta, "Association Rule Mining: Applications in Various Areas," Int. Conf. Data Manag., no. January 2008, pp. 3–7, 2008, [Online]. Available: https://pdfs.semanticscholar.org/e5db/a964a791763d7cd9c9d8979f2c00604e7b9a.pdf.

[41] https://www.salesforce.com/eu/learning-centre/crm/crm-systems/

[42] https://analystanswers.com/data-item-definition-examples-use-in-coding-languages/

[43]http://www.lirmm.fr/~lazaar/imagina/NL-DM-IMAGINA1819-part1.pdf

[44] V. Monbet, “Règles d’association,” pp. 29–35.

[45] https://en.wikipedia.org/wiki/Association_rule_learning/, July 12, 2021.

[46] J. Heaton, “Comparing dataset characteristics that favor the Apriori, ECLAT or FP-Growth frequent itemset mining algorithms,” Conf. Proc. - IEEE SOUTHEASTCON, vol. 2016-July, 2016, doi: 10.1109/SECON.2016.7506659

[47] https://docs.python.org/3/tutorial/, September 5, 2021.

[48] https://research.google.com/colaboratory/faq.html

[49] Multi-criteria discovery of service-based social networks, Master's 2 end of study thesis – “Yahia Fares” university Medea - Algeria, June – 2017 [Online]. Available: https://github.com/Drlilou/pfe-Master

[50] https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/, August 27, 2021.