ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Department of Informatics

Master of Science (MSc)

in Information Systems

Master Thesis

"A Recommender System for Smart Energy Grids"

Alexandra Athanasakou

F3311901

Committee:

Associate Professor Iordanis Koutsopoulos (Supervisor)

Professor George C. Polyzos

Professor Vasilios Syris

Athens, April 2021

Contents

Dedication

To my beloved family and those who are always alive in my mind.

Acknowledgments

I would really like to thank my supervisor Iordanis Koutsopoulos for his support, trust, and guidance. His commitment and loyalty of being a professor and researcher really have a significant impact to those who collaborate with him and I admit that I am one of them. Also, special thanks to Spiros Chadoulos, my second supervisor with whom we had endless conversations and brainstorming ideas about the subject of this Thesis and he really helped me mature in the Recommender Systems field.

Also, I would like to thank Professors George C. Polyzos and Vasilios Syris for accepting to be part of the thesis committee and invest their time in studying this work.

Finally, I cannot forget to thank the people that I consider as a family who always support me and believe in me. A big part of me, I owe it to them.

List of tables

List of figures

List of abbreviations

The following table describes the significance of various abbreviations and acronyms used throughout the thesis.

Abbreviation	Meaning
RS	Recommender System
CB	Collaborative Filtering
MF	Matrix Factorization
SVD	Singular Value Decomposition
SGD	Stochastic Gradient Descent
ALS	Alternating Least Squares
RMSE	Root Mean Squared Error

Abstract

In recent years, there is a massive waste of energy production. This problem is not only from the side of electricity retailers but also from the side of users who consumes electricity. The everyday energy consumption routine causes the lack of energy resources because of the ignorance of the impact of our actions. But what if we change our energy consumption behavior? Is it easy to do such a change or is it difficult? The idea of developing a Recommender System for Smart Energy Grids is a great way to reduce energy consumption in our households and help Earth retain its resources.

As Recommender Systems developed for energy efficiency are among the most popular Smart City goals (Quijano-Sanchez, Cantador, & M Cortes-Cediel, 2020) there are many different implementations in the literature. Among others, Matrix Factorization has been consolidated as the best performing approach in many domains (Koren, Matrix Factorization Techniques for Recommender Systems, 2009); (Zeng & Wang, 2019), and it is believed by (Quijano-Sanchez, Cantador, & M Cortes-Cediel, 2020) that Model-based methods may have an important impact in Smart City initiatives, especially due to the huge amounts of sensor data generated at city scale in many applications.

This thesis aims to develop a Recommender System for Smart Energy Grids, that offers energy tips for energy efficiency, where a set of Matrix Factorization models using Singular Value Decomposition algorithm are trained with explicit feedback with the help of Clustering and especially K-Means algorithm, in order to offer predictions fast and memory-efficiently compared to a single Matrix Factorization model. Each of the Matrix Factorization models is trained with the respective subset of ratings of households that belong within the same cluster, where these households share a similar energy consumption behavior, while missing ratings of households are imputed with the mean rating of the cluster. In more detail, the different models offer faster response time and lower computational cost concerning predictions, while the training with ratings of cluster households and the imputation of missing ratings with the mean rating of each cluster offers more accuracy.

At the same time, in order that the proposed Recommender System offers a more personalized experience to the households, a set of “Real-Time rules” was created which filters the predictions list from Matrix Factorization models with respect to the recommendation time frame, the appliance of energy tips to electrical devices that are used from the target household and the interests of the target household and its similar users. Eventually, the final recommendation of the proposed Recommender System observes the real-time consumption of the target household’s electrical devices appeared in the filtered prediction list, in order to find possible increased consumption based on household’s past behavior.

In summary, this thesis highlights the importance of Clustering in both Matrix Factorization Models with SVD algorithm and the task of creating a dense ratings matrix, in order to offer accurate predictions in a fast and memory-efficiently way. Finally, as far as the personalization of recommendations is concerned, this thesis introduces a set of “Real-Time rules” that take into consideration the time frame where the recommendation takes place, the electrical devices used from the target household, the interests of the target household based on the interests of similar users, and, finally, the real-time energy consumption of the target household.

Keywords

Recommender Systems; Smart Energy Grid; Machine Learning; Matrix Factorization; Personalized Recommendations

Chapter 1 | Introduction

Today's electricity grid is used so that massive amounts of energy are being lost (Farhangi, 2010). On the other hand, lack of energy is a problem caused by our everyday routine in which big amounts of energy is used without even considering if there is a real need. Therefore, a way to sustain the energy resources is to introduce a smart way to convince people to decrease their energy consumption.

The next-generation electric power system is known as Smart Grid. A definition proposed by (Murphy, 2010) is the following:

"A smart grid is a modern electric system. It uses communications, sensors, automation and computers to improve the flexibility, security, reliability, efficiency, and safety of the electricity system. It offers consumers increased choice by facilitating opportunities to control their electricity use and respond to electricity price changes by adjusting their consumption. In short, it brings all elements of the electricity system production, delivery and consumption closer together to improve overall system operation for the benefit of consumers and the environment."

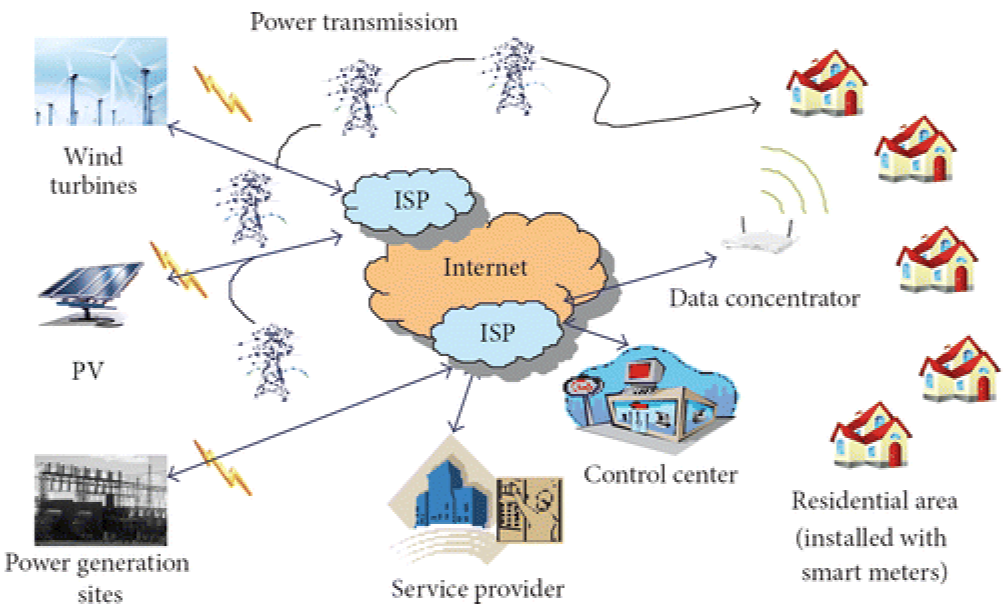


Figure 1: An example of communication architecture in Smart Grid (Bari, Jiang, & Saad, 2014)

So, with the help of smart meter data from Smart Energy Grids, a way to convince people to reduce their energy consumption is to take advantage of a Recommender System that offers personalized content. Of course, several different factors could influence energy usage, but the aim of Recommender Systems in Smart Energy Grids is to convince users to change their energy behavior by saving money and helping the planet retain its resources. (Martinez, Lairner, & Keating, 2009) suggests that the customers' behavior is an important factor. More specifically, the cost of energy is usually considered as the most important factor customers regard concerning how much energy they wish to spend. Another factor is the environmental footprint. These factors are considered as inner motivations for customers.

Another factor that might influence the household to reduce energy its consumption as described in (Agency, 2010) is the efficiency of electrical appliances in the household. By displaying the energy spent and how much the system predicts that the customer will spend, the customer is able to make a decision on how to handle this device, for example considering change this particular device, or initiate to handle it more efficiently.

According to (Crawley & Huang, 1997), external influences such as the weather and sunshine hours, are also factors that can affect a possible change in energy usage. If there is information about the energy consumption needed for heating or air conditioning, this can be viable in understanding how to use energy more cost-efficiently.

Finally, for an effective change of energy consumption pattern, there is a need to understand what type of information is needed so as to change the consumers' energy behavior. Do they care about saving money? Their environmental footprint? What will make the consumer behave differently and how it can be achieved?

With that in mind, Recommender Systems in Smart Energy Grids can have an impact on the energy consumption behavior of households. However, a Recommender System should handle smart meter data accordingly by getting only the most useful information, consider external factors such as the weather and listen to consumer needs, to successfully solve the energy consumption reduction problem.

1.1 | The Problem

In a Smart Energy Grid there are large amounts of information available in real-time. This information could be revealed to the consumer, but it would be very hard to make any use of the full spectrum of sensors, controls, and information technologies available. At the same time, users should be able to make fast decisions about how they want to spend their energy. For this reason, the Smart Energy Grid should be able to inform consumers about different ways that will help them reduce their energy consumption. Therefore, a Recommender System could effectively handle this problem by recommending energy tips that could help consumers reduce their energy consumption.

There exist Recommender Systems that use Collaborative Filtering (Kong & Liang, 2020) and especially Memory-based algorithms (Kwac, Flora, & Rajagopal, 2014); (Collaborative Recommendations and Adaptive Control for Personalised Energy Saving, H2020). However, this class of algorithms is memory and time insufficient when providing predictions for Big-Data. Also, in most real-world problems people tend to rate a small number of available items and the ratings matrix becomes sparse (Ricci, Rokach, & Shapira, 2015). In that case, Memory-based algorithms do not have enough feedback to provide accurate predictions. Matrix Factorization is a method that can handle both cases, namely effectively handle the data-sparsity problem and providing accurate predictions while being memory and time-efficient (Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, 2008).

From the above, Matrix Factorization successfully overcomes many of the problems of Memory-based algorithms. However, as data increases, the need for even more efficient ways to provide accurate predictions while being time and memory-efficient is apparent. So, the problem is to find a solution that extends the advantages of Matrix Factorization by improving the accuracy and the time and memory cost.

At the same time, we should try to offer to users real-time personalized recommendations taking into consideration the time frame where the recommendation takes place while at the same time observing the current energy consumption of the user. In the literature, there was not any approach of personalized recommendations that consider these two cases.

1.2 | Novelty

In this thesis, we address the problem of having an even more efficient way in terms of time, memory, and accuracy, to generate predictions using Matrix Factorization, and especially SVD algorithm, which is the model-based approach that implements the Collaborative Filtering method of RSs, with explicit feedback by combining the results derived from a Clustering algorithm. At the same time, we will present a novel way to offer personalized recommendations taking into consideration, mainly, the recommendation time frame and current energy consumption of the target user.

The contributions of this thesis are as follows:

- We propose a novel method for providing accurate predictions to households with less training and testing time and memory usage by grouping the households into clusters based on their electricity consumption using the K-Means algorithm and then create a Matrix Factorization model for each cluster, with fewer households that are also similar in terms of electricity consumption. To get even more accurate results, we consider imputing the missing ratings of users with the mean rating of the cluster they belong to. Finally, the algorithm used for Matrix Factorization is SVD, which is suitable for explicit ratings.
- We present "Real-Time rules", a novel approach for personalized recommendations which includes a sequence of steps applied in each prediction list generated from the Matrix Factorization models taking into consideration, at first, the time frame where the recommendation takes places, the electrical devices used from the target household, the interests of both target household and cluster's households, and, then, the real-time energy consumption of the target household that might be increased in certain electrical devices that smart meter data are collected. With this step, we offer the final recommendation to the target household.
- Our approach achieved an average RMSE of 0.812, an average training time of 0.12 s from 0.48 s, an average testing time of 0.06 s from 0.24 s, a Precision @ 15 of 70% which is 25% better than a dummy-static Recommender System and a Recall @ 15 which is 13% better, respectively. The same quality of predictions retained after applying "Real-Time rules" which offers a single, personalized recommendation.

1.3 | Literature Review

Key subjects as "User Profiling for Smart Energy Grid", "Recommender System for Smart Energy Grid", "Recommender Systems for Smart Cities" have been explored.

A recent paper by (Quijano-Sanchez, Cantador, & M Cortes-Cediel, 2020) stated the different Recommender Systems approaches that have been used in the context of smart cities. The purpose of this paper was to show current opportunities and challenges where personalized recommendations could be exploited as solutions for citizens, firms, and public administrations. This paper gave a clear and updated view concerning Recommender Systems in Smart Cities and how important is the reduction of energy consumption as a research topic. Also, this paper presented the different approaches of Recommender Systems that Smart City applications use, which Smart Energy Grids are part of them.

Regarding specific smart city actions and goals, among the most addressed objectives is energy efficiency. Moreover, as trending goals, saving energy in smart homes in a smart environment stands out among all.

Concerning the recommendations strategies on Smart City applications, Figure 2 shows their distribution in the papers. Most of the approaches are Collaborative Filtering and Content-based, representing 35% and 22% of the surveyed papers, respectively. However, in most papers, several recommendations strategies are performed following hybrid techniques.

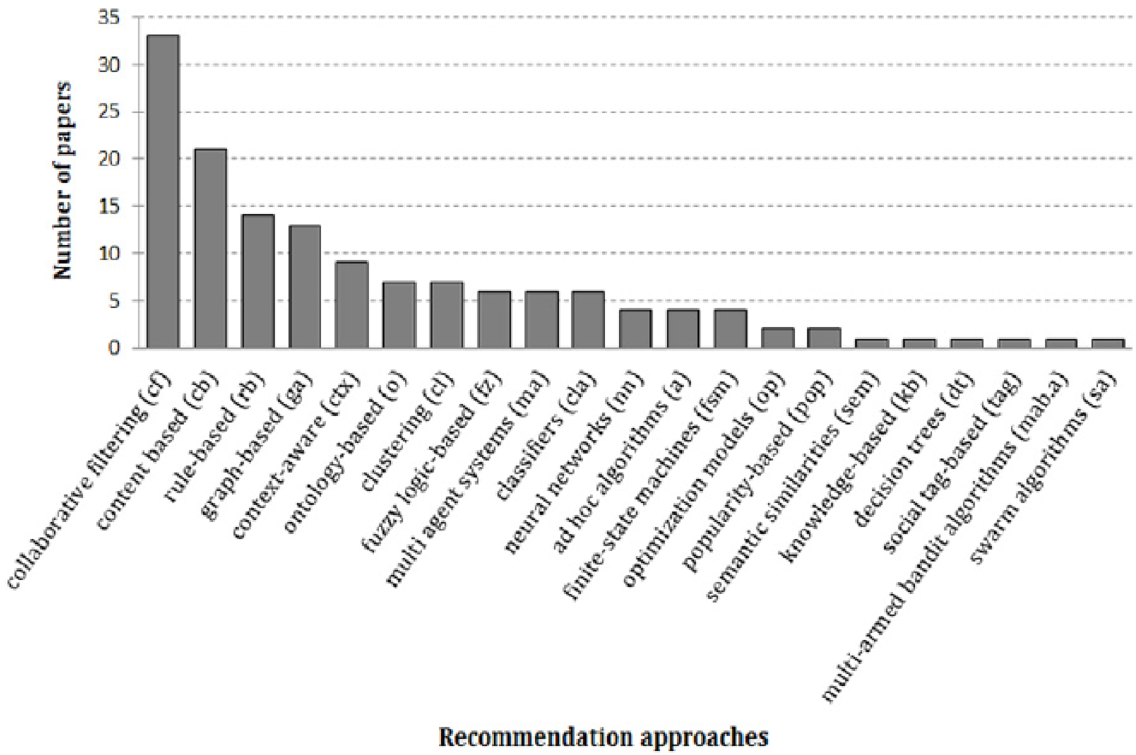


Figure 2: Recommendation Approaches (Quijano-Sanchez, Cantador, & M Cortes-Cediel, 2020)

In terms of the computed evaluation metrics, the majority (44%) make use of ranking quality metrics, mainly precision and recall. The next popular metrics are MAE and RMSE – which measure rating prediction errors and are reports in 18% of the cases – and user satisfaction, representing another 18%. System response time is considered in 8% of the cases, whereas systems effectiveness is analyzed in 6% of the evaluations. The rest of the used metrics are related to a variety of task dependent issues, such as time saved by users, energy consumption reduction and traffic congestion.

Also, a lot of different approaches to reduce the energy consumption of a user have been developed. In (Haben, Singleton, & Grindrod, 2016) paper, a combination of the Finite Mixture Model (FMM) of Gaussian multivariate distribution as a clustering method with the EM Algorithm has been made in order to distinguish user groups based on their electricity consumption and later help them to reduce demand in Low-Voltage networks.

Another approach described in (Luo, Ranzi, Wang, & Dong, Service Recommendation in Smart Grid: Vision, Technologies, and Applications, 2016) provided recommendations concerning electricity retail plans, using Fuzzy C-Means as a clustering method in order to find users that have a similar energy consumption with the target user. Then, a distance metric found similar users with target user, collected their electricity retails plans choices implicitly and then employed a weighted aggregation strategy to aggregate the similar users' ratings for predicting the rating of the target user on a particular retail plan.

In (Schweizer, et al., 2015), a Window-sliding algorithm has been used, more specifically, an event stream of the current behavior data inside the smart home cooperates with an association rule database in order to match triplets of events with rules that exist in the association rule database. If there is a match, then a set of energy tips are being generated. In order to prioritize the most important ones, weights are assigned in each of these rules and then the final recommendation is being made.

Finally, a similar approach comparing to this thesis has been developed by enCOMPASS project (Collaborative Recommendations and Adaptive Control for Personalised Energy Saving, H2020). This Recommender System generated recommendations for energy tips obtained from a static database which was also used in this thesis. The main parts used in this project were Clustering and Time Series Analysis for determining the similarity between users depending on their energy consumption, while recommendations were generated using Collaborative Filtering and a Memory-based approach with implicit feedback. As far as Clustering is concerned, the features used apart from energy consumption, were related to demographics, house information, and categorical data. Additionally, a weight was assigned to each tip before proceeding to Collaborative Filtering.

1.4 | Thesis Structure

In the second chapter of this thesis, a detailed review of what is a Recommender System, what is the purpose of it and what are the different methods and approaches of implementation are presented. Additionally, significant emphasis is given to Matrix Factorization, and especially Singular Value Decomposition, because of the leading presence in the proposed Recommender System. Also, the same detailed information is given for Clustering and especially K-Means algorithm, which is a main part of the proposed Recommender System. Finally, a review concerning the different evaluation methods and metrics that have been used in this thesis is presented.

The third chapter has detailed information about the datasets used for the Recommendation process such as column names, descriptions, diagrams that highlight the distribution of users and ratings, and, finally, the use of each dataset in the main parts of the proposed Recommender System.

The fourth chapter describes all the components of the proposed Recommender System along with the final structure of the last, observes a set of experiments that led to the final decision concerning the Collaborative Filtering approach, and finally, presents a simulation. In more detail concerning the components of the proposed Recommender System, those are Clustering, which K-Means algorithm was used with features related to energy consumption of households, next is Matrix Factorization where SVD algorithm is used, and finally the "Real-Time rules" which consists of multiple filtering steps to the prediction list from Matrix Factorization models and leads to the final recommendation. All the aforementioned are accompanied with diagrams, charts and tables that provide relevant information.

The last chapter is the Conclusion which emphasizes the results of the proposed Recommender System, the problems faced while developing this Recommender System, how some general problems of Collaborative Filtering have been handled, and finally, present some ideas for future work.

Chapter 2 | Background

Recommender Systems (RSs) can be developed in multiple ways to best serve our purpose. In that section, focus is given on basic knowledge concerning RSs in order to understand why they provide value. Specifically, a detailed discussion will be presented concerning the definition of RSs and all the information needed to understand the purpose of RSs, their types, and how they work.

2.1 | Recommender Systems

As mentioned by (Portugal, Alencar, & Cowan), “Recommender Systems (RS) are used to help users find new items or services, such as books, music, transportation or even people, based on information about the user, or the recommended item”. These systems also play an important role in decision-making, helping users to maximize profits or minimize risks. Today, RSs are used in many information-based companies such as Google, Twitter, LinkedIn, and Netflix. The field of RS has its origin in the mid-1990s with the introduction of Tapestry, the first RS.

Figure 3: Popular platforms that make use of Recommender Systems

Recommendation problem has mainly three tasks:

1) Collecting information about users,
2) Learning from collected information and predicting users’ preferences for unknown items,
3) Applying a function or building a model that selects (and ranks) the items that are more likely to be preferred by users.

As far as the data collection and profiling task is concerned, in order for the Recommender System to provide personalized suggestions, past choices and preferences of users are used that reflect users’ tastes and interests. This information can be either explicitly provided or implicitly inferred. More information about these two types of feedback can be found in Section 2.1.2.

2.1.1 | Data for Recommender Systems

The data which RSs use are Items, Users, and Transactions. In more detail:

Items: Items are the objects to be recommended to users. The value of each item (positive or negative) is determined by users' interactions such as, for example, popular, unpopular, trending, etc. and that interaction determines the possibility to recommend or not. Depending on the type of RS, the item should be as descriptive as it could in order to relate it with other similar items.

Users: Users of an RS are the ones who feed RSs with data in order to get personalized recommendations and help others, with similar interests, find what best suits them.

Transactions: It is generally referred to a transaction, or as a recorder of interaction, that was given explicitly or implicitly. Those transactions store important information generated during the human-computer interaction which are useful for the algorithm that provides predictions.

Ratings are the most popular form of transaction data that an RS collects. These ratings may be collected explicitly or implicitly.

Explicit feedback is direct preference statements made by users about items they know (Quijano-Sanchez, Cantador, & M Cortes-Cediel, 2020). This knowledge is usually stored as ratings or as unary/binary values. Explicit feedback is the most preferred because it allows a precise control on what the system knows about the users' preferences. However, collecting that type of information requires time and effort from the side of users. Moreover, when including explicit interactions in real world application, there is a high risk of biases in rating distribution and thus in item relevance predictions, as users may tend to rate only what they like (Zhao, Harper, Adomavicius, & Konstan, 2018).

Implicit feedback, on the other hand, refers to user preferences that are inferred from implicit/indirect user interactions with the system and/or the environment without the need for users to actively inserting input. This form of preferences can be obtained by recording search queries, product purchases, and mouse actions, among others. While it allows capturing abundant information about users, it tends to obtain information that is noisier and may be biased to positive preferences (Zhao, Harper, Adomavicius, & Konstan, 2018).

Summarizing, when feedback is explicit, we refer to ratings:

- Numerical, such as the 1-5 stars provided in the movie RS of Netflix
- Ordinal, such as "Strongly Disagree", "Agree", etc. that the user is asked to select the term which best describe what he/she feels about an item
- Binary, such as 0 for Dislike and 1 for Like

When implicit, we refer to ratings:

- Unary, where a user has observed or purchased an item, or otherwise rated the item positively. In such cases, the absence of a rating indicates that there is no information relating the user to the item.

Whenever we are discussing about RSs, we refer to these variables, functions, and sets:

$$\boldsymbol{u}\text{: } user$$

$$\boldsymbol{i}\text{: } item$$

$$\mathbf{r}(\boldsymbol{u}, \boldsymbol{i})\text{: } rating\ (true\ function)$$

$$\hat{\boldsymbol{r}}(\boldsymbol{u}, \boldsymbol{i})\text{: } prediction\ \text{(based on true function)}$$

$$\boldsymbol{S}\text{: } set\ of\ possible\ values\ for\ a\ rating$$

$$\widehat{\boldsymbol{R}}\text{: } set\ of\ predictions$$

$,where\ u \in (u_1, u_n),\ \ i \in (i_1, i_n)$

After the prediction phase, the system will recommend the items i_{j1} … i_{jk} (k ≤ n) with the largest predicted utility.

2.1.2 | Recommendation Techniques

RSs can be classified by the degree of personalization, including the usefulness and accuracy of the recommendations (Schafer, Konstan, & Riedl, 2001). The degree of personalization can be defined from low to high. The first degree of personalization is a relatively simple system that does not take user's preferences into account when making recommendations. For instance, the RS only generated a list of the most popular items based on the number of review or number of purchases (i.e. editor's choices or top-sellers) (Ricci, Rokach, & Shapira, 2015).

Another way to classify recommender systems could be according to different principles depending on the task they are focused in, for example predicting item ratings and ranking item sets. Also, the approach to extract user preferences, implicit or explicit feedback, and the recommendation dynamics they follow (i.e. single shot or unique answer and conversational or iterative approaches) (Quijano-Sanchez, Cantador, & M Cortes-Cediel, 2020).

In summary, the different techniques of RSs are the following:

- Content-Based Filtering
- Collaborative Filtering
- Demographic
- Knowledge-Based
- Hybrid

The three most popular types of RSs are Content-Based Filtering (Lops, Gemmis, & Semeraro, 2011), Collaborative-Filtering (Ekstrand, Riedl, & Konstan, 2011) and Hybrid.

Recommendation techniques have different strengths and weaknesses. Some of the most common weaknesses are the rating sparsity problem and the cold start problem (Schein, Popescul, Ungar, & Pennock, 2002).

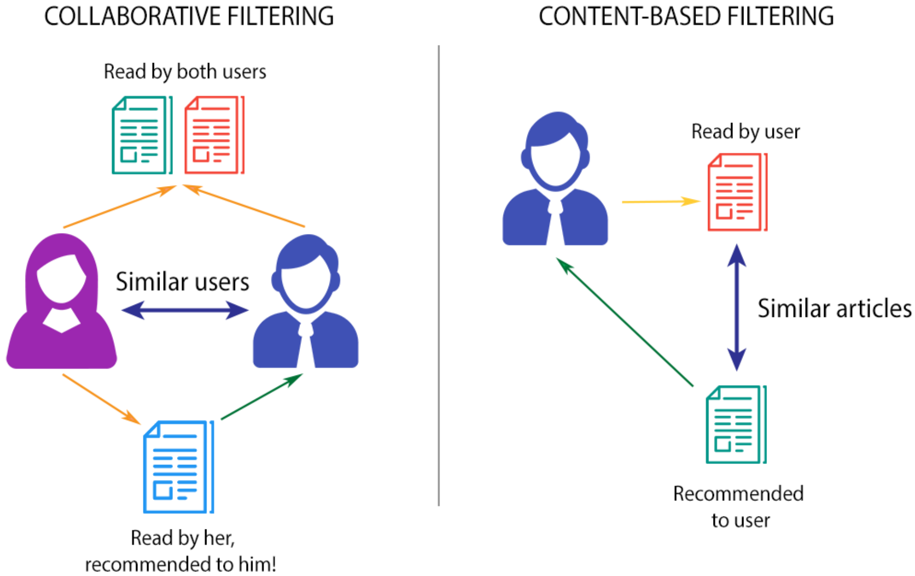


Figure 4: Collaborative Filtering vs Content-Based Filtering [1]

Collaborative Filtering

Collaborative filtering (also known as social filtering) is the most widely implemented recommendation system. It recommends popular items to users based on the feedback of other users who share the same interests. This approach suffers from the cold-start problem, whereby a new item or a brand-new user has not enough data available, namely ratings, in order to get accurate and relevant recommendations and Data-Sparsity problem. The most popular approaches of Collaborative Filtering are Memory-based and Model-based. The Memory-based approach compares a user's historical records to other records in the database (Schiaffino & Amandi, 2009).The Model-based approach uses statistical or learning methods, such as a Bayesian network (Huang & Bian, 2009), where a filtering technique classifies the user's historical records and builds a user model that is subsequently used in the recommendation process (Hsu, Lin, & Ho, 2012).

2.1.3 | Matrix Factorization

Matrix Factorization has become the predominant technique in Recommender Systems (Nobrega & Marinho, 2014). In its basic form, Matrix Factorization characterizes both items and users by vectors or factors in front of item rating patterns (Koren, Matrix Factorization Techniques for Recommender Systems, 2009). High correspondence between item and user factors leads to a recommendation. These methods have become popular in recent years by combining good scalability with predictive accuracy. Also, they offer much flexibility for modeling various real-life situations.

[1] https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd

Recommender systems rely on different types of input data, which are often placed in a matrix with one dimension represented users and in the other dimension representing items of interest. The most convenient data is high-quality explicit feedback, which includes explicit input by users regarding their interest in products. An example is Netflix which collects star ratings for movies, and TiVo users indicate their preferences for TV shows by pressing thumbs up and thumbs down buttons. Usually, expected feedback comprises a sparse matrix, since any single user is likely to have rated only a small percentage of possible items.

One strength of Matrix Factorization is that it allows incorporation of additional information. When explicit feedback is not available, Recommender Systems can infer user preferences using implicit feedback, which indirectly reflects opinion by observing user behavior including purchase history, browsing history, search patterns, or even mouse movements. Implicit feedback usually denotes the presence or absence of an event, so it is typically represented by a densely filled matrix.

Additionally, Factorization methods address the problems of limited coverage and sparsity by projecting users and items into a reduced latent space that captures their most salient futures. Because users and items are compared in this dense subspace of high-level features, instead of the “rating space”, more meaningful relations can be discovered. In particular, a relation between two users can be found, even though these users have rated different items. As a result, such methods are generally less sensitive to sparse data (Bell, Koren, & Volinsky, Modeling relationships at multiple scales to improve accuracy of large recommender systems, 2007); (Billsus & Pazzani, 1998)

Matrix factorization models map both users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. Accordingly, each item i is associated with a vector $q_i \in \mathbb{R}^f$, and each user u is associated with a vector $p_u \in \mathbb{R}^f$. For a given item i, the elements of q_i measure the extent to which the item possesses those factors positive or negative. For a given user u, the elements of p_u measure the extent of interest the user has in items that are high on the corresponding factors, again, positive or negative. The resulting dot product, $q_i^T p_u$, captures the interaction between user u and item i - the user’s overall interest in the item’s characteristics. These are approximate user’s u, rating of item i, which is denoted by r_{ui}, leading to the estimate:

$$\hat{r}_{ui} = q_i^T p_u$$

Equation 1: Predicted Rating (1)

The major challenge is computing the mapping of each item and user to factor vectors $q_i, p_u \in \mathbb{R}^f$. After the Recommender System completes this mapping, it can easily estimate the rating a user will give to any item by using Equation 1.

To learn the factor vectors ($q_i \ and \ p_u$), the system minimizes the regularized squared error on the set of known ratings:

$$\min_{q^*,p^*} \sum_{(u,i)\in K} r_{ui} - q_i^T p_u + \lambda(\left||q_i|\right|^2 + \left||p_u|\right|^2)$$

Equation 2: Minimization of error function where λ is a regularization parameter that weights the two terms so that the objective function is not dominated by one or the other. Common algorithms to minimize the objective function is Stochastic Gradient Descent and Alternating Least Squares.

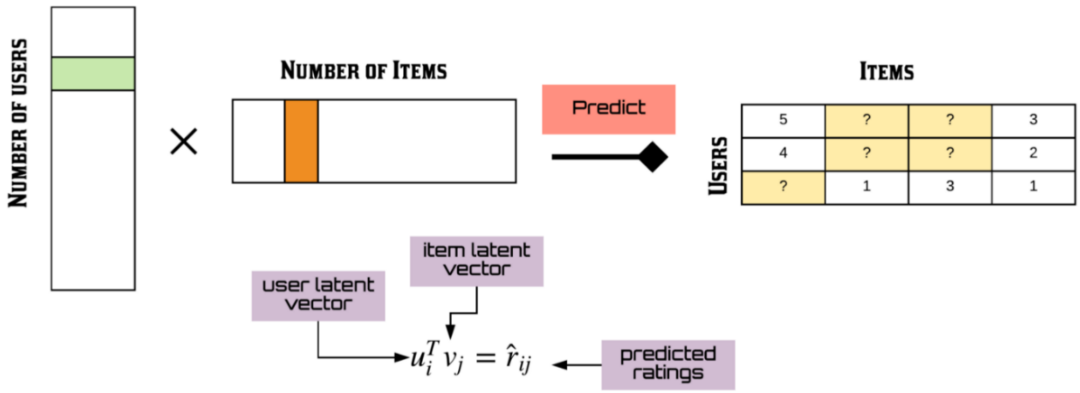


Figure 5: Matrix Factorization visualized[2]

2.2 | Clustering

Clustering is an approach of unsupervised learning that deals with the data structure partition in an unknown area and is the basis for further learning. The definition for clustering is described as follows (Jain & Dubes, 1988):

a) Instances, in the same cluster, must be similar as much as possible
b) Instances, in different clusters, must be different as much as possible
c) Measurement for similarity and dissimilarity must be clear and have the practical meaning

In simple words, clustering is an unsupervised learning method that tries to identify relations between data points where the area is unknown, and there is no prior knowledge.

The standard process of clustering can be divided into the following steps:

1) Feature extraction and selection: Extract and select the most representative features from the original data set
2) Clustering algorithm design: Design the clustering algorithm according to the characteristics of the problem
3) Result evaluation: Evaluate the clustering result and judge the validity of the algorithm
4) Result explanation: Give a practical explanation for the clustering result

[2] https://towardsdatascience.com/recsys-series-part-4-the-7-variants-of-matrix-factorization-for-collaborative-filtering-368754e4fab5

2.2.1 | K-Means

K-means is a clustering algorithm that belongs into the category of **Clustering based on Partition**, where the main idea is to regard the center of data points as the center of the corresponding cluster (Xu & Tian, 2015). K-Means is one of the most popular of this kind of clustering algorithms. The core idea of K-means is to update the center of class, which is represented by the center of data points, by iterative computations, and the iterative browsers will be continued until some criteria for convergence is met. Also, K-Means, is strictly dependent on the value of K, which is the number of clusters. The number of clusters is not known and sometimes it becomes challenging to find the optimal K for the available data.

The advantages of such clustering algorithms are the relatively low time complexity and high computing efficiency in general.

The disadvantages of such clustering algorithms are that they are relatively sensitive to the outliers, easily drawn into local optimal, the number of clusters needed to be preset, and the clustering result is sensitive to the number of clusters.

In the clustering problem, there is a training set $\{x^{(1)}, \dots, x^{(m)}\}$ that its data need to be grouped into a few cohesive clusters k. Each data point $x^{(i)} \in \mathbb{R}^n$ belongs to a feature vector but there are no labels for $y^{(i)}$, making this an unsupervised learning problem. The goal is to predict k centroids and a label $c^{(i)}$ for each datapoint. The algorithm[3] is as presented:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence:

   {

   For every i, set

$$c^{(i)} = \arg\min_j \left|\left|x^{(i)} - \mu_j\right|\right|^2$$

Equation 3: Step 2a of K-Means algorithm

   For each j, set

$$\mu_j = \frac{\sum_{i=1}^{m} 1\left\{c^i = j\right\} x^i}{\sum_{i=1}^{m} 1\left\{c^i = j\right\}}$$

Equation 4: Step 2b of K-Means algorithm

   }

2.2.2 | The Elbow method

The Elbow method is a heuristic used in determining the number of clusters in a data set (Elbow Method (Clustering), 2021). The method consists of plotting the explained variation as a function of the number

[3] https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

of clusters and picking the elbow of the curve as the number of clusters to use. In case of K-Means, Elbow method finds the optimal k which refers to the number of clusters.

2.2.3 | Hopkins Statistic

The Hopkins Statistic as mentioned in (Hopkins Statistic, 2020), is a way of measuring the cluster tendency of a dataset (Hopkins & Skellam, 1954), or the feasibility of cluster analysis. It acts as a statistical hypothesis test where the null hypothesis is that the data is generated by a Poisson point process and are thus uniformly randomly distributed (Banerjee, 2004). A value close to 1 tends to indicate that the data is highly clustered, random data will tend to result in values around 0.5, and uniformly distributed data will tend to result in values close to 0 (Aggarwal, Data Mining, 2015).

The formula of Hopkins Statistic is as follows:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

Equation 5: Hopkins Statistic Formula

Where D is the dataset that contains n uniformly points $(p_1, \ldots, p_n)$,

$\sum_{i=1}^{n} x_i$ is the summary of distances between real points, $p_i \epsilon D$, and their nearest neighbors, $q_i \epsilon D$,

$\sum_{i=1}^{n} y_i$ is the summary of distances between artificial points generated randomly for dataset D with exact n points and the same variation as the original real dataset D, and their nearest artificial data points.

Finally, to understand if the dataset used for Clustering has meaningful clusters, then Hopkins Statistic above 0.5 will inform us that there is a high probability that the dataset is not uniformly distributed, meaning that the distance between real points and artificial ones is substantially larger.

2.2.4 | Silhouette coefficient

Silhouette coefficient is a way to evaluate the clustering result based on the average normalized distance between a data point and other data points in the same cluster and average distance among different clusters (Aggarwal, Data Mining, 2015).

The silhouette value ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters (Rousseeuw, 1987).

2.3 | Evaluation of Recommender Systems

In this section, the evaluation metrics used for detecting the accuracy of predictions and the quality of recommendations will be presented, so as the evaluation method where Collaborative Filtering was evaluated.

2.3.1 | Evaluation Metrics

One of the error metrics is **Root Mean Squared Error (RMSE)**. The way RMSE works is that the system has generated some predicted ratings $\hat{r}_{ui}$ for a test set I of user-item pairs (u, i) for which the true ratings r_{ui} are known. Typically, r_{ui} are known because they are hidden in an offline experiment, or because they were obtained through a user study or online experiment. The RMSE between the predicted and actual ratings is given by the following Equation 6:

$$\text{RMSE} = \sqrt{\frac{1}{|I|}\sum_{(u,i)\in I}(\hat{r}_{ui} - r_{ui})^2}$$

Equation 6: RMSE formula

As far as quality evaluation metrics is concerned, Recall and Precision are among the most popular. However, in the context of Recommender Systems, sometimes, is better to consider only the top-k recommendations to evaluate the recommendations. So, in Equations 7 and 8, Recall @ K and Precision @ K quality metrics are being presented where they consider only the top-k recommendations. In more detail, Recommended (u) is the items obtained from the top-K prediction list of user u where their predicted rating exceeds a certain threshold. At the same time, an item is considered as Relevant (u) for user u if the real rating exceeds a certain threshold. As far as the threshold is concerned, this value can be anything.

$$\text{Recall @ K (u)} = \frac{\text{Recommended (u)} \cap \text{Relevant (u)}}{Relevant\ (u)}$$

Equation 7: Recall @ K Formula

$$\text{Precision @ K (u)} = \frac{\text{Recommended (u)} \cap \text{Relevant (u)}}{Recommended\ (u)}$$

Equation 8: Precision @ K Formula

Essentially, Recall is the proportion of relevant items that are recommended, and Precision is the proportion of recommended items that are relevant.

2.3.2 | Evaluation Method

Cross Validation

Cross validation is an evaluation method used for offline evaluation (Aggarwal, Recommmender Systems - The Textbook, 2016). This method takes as input the ratings offered by users and divides them into q equal sets. Therefore, if S is the set of specified entries in the ratings matrix R, then the size of each set, in terms of the number of entries is |S|/q. One of the q segments is used for testing, and the remaining (q – 1) segments are used for training. In other words, a total of |S|/q entries are hidden during each such training process, and the accuracy is then evaluated over these entries. This process is repeated q times by using each of the q segments as the test set. The average accuracy over the q different test sets is reported. It is worth mentioning that this approach can closely estimate the true accuracy when the value of q is large.

Chapter 3 | Datasets used

In this chapter, information about each dataset used in the proposed Recommender System is provided. In more detail, column names and description of them is available, etc. Also, diagrams that describe the distribution of users or ratings are presented.

3.1 | The Pecan Street Dataset

The Pecan Street Dataset is a collection of hourly measurements in circuit-level electricity use and generation from nearly 1.000 volunteer homes from 2012 through 2019. Those measurements are anonymized, cleaned and curated into specific datasets while made available for free to university researchers through Dataport[4].

Essentially, some important columns of Pecan Street dataset that are presented in Table 1, are related to energy consumption data, while others provide information about the household such as the building type of the house, presence of photovoltaic panels, total square footage of the house and more.

Feature Name	**Description**
Data id	The unique identifier for the home-resident.
Building type	"Single-Family Home", "Town Home", "Apartment", "Mobile Home".
Pv	Denotes if the specific house has solar photovoltaic system installed.
Total square footage	The square footage of the first floor of the home.
Air	Air compressor circuit.
Air window unit	Window unit air conditioner circuit.
Aquarium	Aquarium circuit.
Bathroom	Bathroom circuit that includes only lights, fans, and wall outlets.
Bedroom	Bedroom circuit that includes only lights, fans, and wall outlets.
Car	Electric vehicle charger.
Clothes washer	Stand-alone clothes washing machine.
Clothes washer dry g	Clothes washing machine and natural gas-powered dryer circuit.
Dining room	Dining room circuit that includes only lights, fans, and wall outlets.
Dishwasher	Dishwashers circuit.
Disposal	Kitchen sink garbage disposal circuit.
Dry e	Electricity-powered clothed dryer (240V circuit).
Dry g	Natural gas-powered clothes dryer (120V circuit). Meters will only pick up the electricity use from the dryer's drum rotation, not the gas heating signature.

[4] https://www.pecanstreet.org/dataport/

Freezer	Stand-alone freezer circuit.
Furnace	Furnace and air handler circuit.
Garage	Garage circuit that includes only lights, fans, and wall outlets.
Gen	Power generated by a solar photovoltaic system.
Grid	Measurement of power drawn from the electrical grid (grid = use – gen)
Heater	Stand-alone heater circuit
House fan	Whole home fan circuit
Ice maker	Stand-alone icemaker circuit
Jacuzzi	Jacuzzi bathtub or hot tub
Kitchen	Kitchen circuit that includes only lights, fans, and wall outlets.
Kitchen App	First kitchen small appliance circuit that includes only wall outlets in the kitchen, and so may include toasters, coffee makers, blenders, etc.
Lights plugs	General lighting and plugs circuit that includes lights, fans, and wall outlets, often from multiple rooms in the home.
Living Room	Living room circuit that includes only lights, fans, and wall outlets.
Microwave	Microwave circuit.
Office	Home office circuit that includes only lights, fans, and wall outlets. Computers may be common devices plugged into included any wall outlets included on this type of circuit.
Outside Lights plugs	Exterior lighting and plugs circuit.
Oven	Oven circuit.
Pool	Combination pool pump and/or pool auxiliary power circuit.
Pool Light	Pool lighting circuit.
Pool pump	Pool pump circuit.
Pump	Any type of pump that is not a pool pump.
Range	Range (either a stand-alone cooktop or a cooktop and an oven) circuit.
Refrigerator	Refrigerator circuit
Security	Security system circuit
Shed	Shed circuit
Sprinkler	Sprinkler system circuit
Use	Whole home electricity use (use = gen + grid)
Utility room	Utility room circuit
Vent hood	Vent hood circuit
Water heater	Electric water heater
Wine cooler	Wine cooler circuit

Table 1: Feature Names and their Description for the Pecan Street Dataset

From the 1000 available households, only 612 of them used for training the proposed Recommender System.

Finally, Pecan Street Dataset is used for Clustering purposes and plays a significant role in the 2nd part of Real-Time rules.

3.2 | Dataset with Tips

The dataset with energy tips was originally taken from enCOMPASS Project (Collaborative Recommendations and Adaptive Control for Personalised Energy Saving, H2020) and is a part of the mobile application "Funergy" which tries to learn, in a fun way, the user how he/she can reduce energy consumption and change effectively his/her behavior in order to achieve energy savings, cost-efficiency and protect the environment. However, additional columns were added for the shake of the proposed Recommender System. The unique energy tips are 180.

The columns in the Tips dataset are those presented in Table 2. The new columns added are *appliance, whenTime,* and *whenSeason*.

Feature Name	Description
Tip id	The unique identifier of the energy tip.
Appliance	The electric device which the energy tip applies to.
When Time	The time of the day which is most suitable for the tip to be recommended.
When Season	The season of the year which is most suitable for the tip to be recommended.
Title	A general title of the energy tip
Description	A more informative text about the energy tip

Table 2: Feature names and descriptions for Tip Dataset

In Figure 6 we plot the number of energy tips that apply in each of the electrical devices which exist in the Pecan Street dataset is presented. It seems that the most energy tips apply in electrical devices that have a direct relationship with the air, such as air-condition, or air conditioning in windows, while others apply to all rooms which contain lights, fans, and wall outlets. Also, another category of appliance is the one that is rather general than to a specific electric device (use column). Less frequent are energy tips that refer to jacuzzi, freezer, oven, etc.

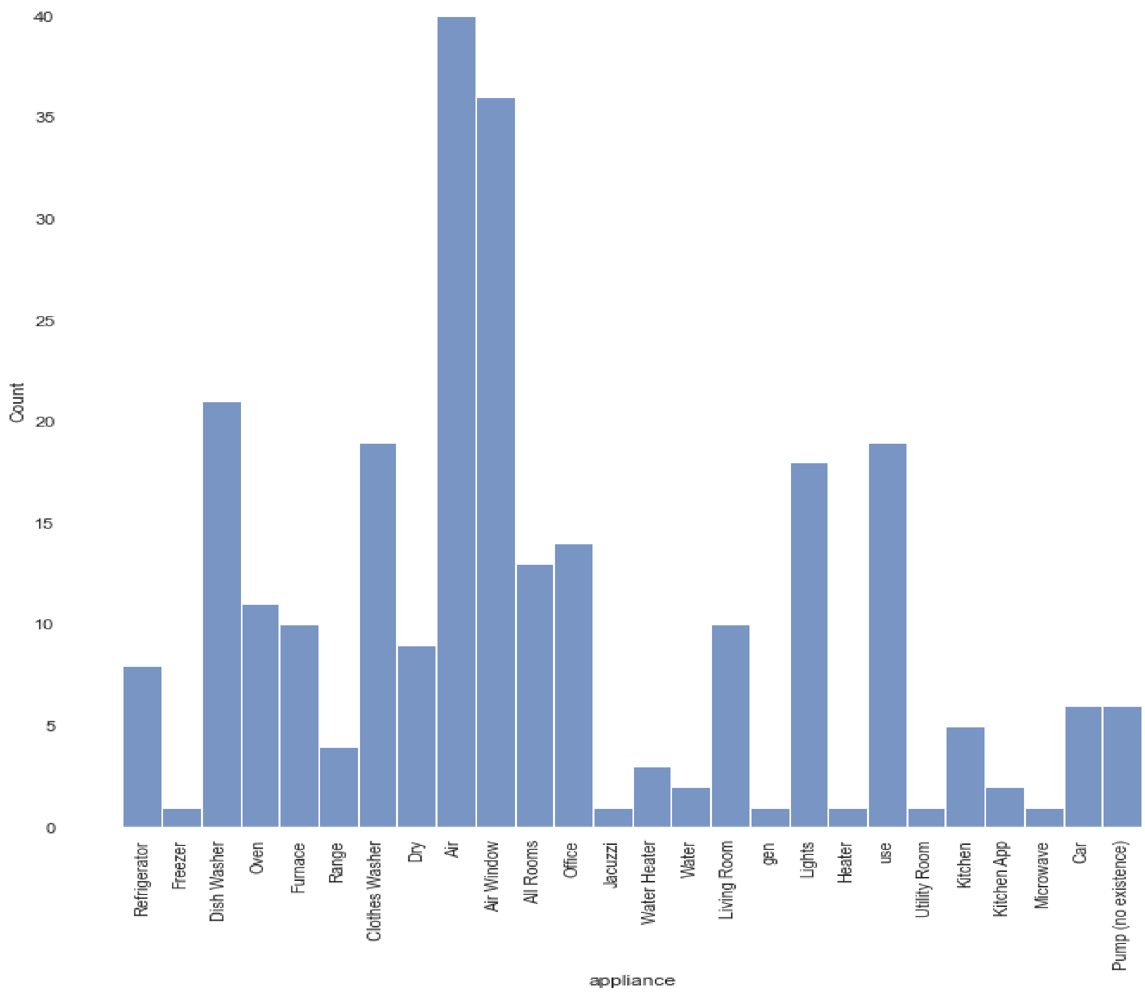


Figure 6: Frequency of appliance of electrical devices in available energy tips

3.3 | Ratings Matrix - Synthetic dataset

Since there is no explicit feedback from households of the Pecan Street Dataset, a synthetic ratings dataset was created intended to be used in Matrix Factorization with SVD. Considering that ratings are values from 1 through 5, we used the binomial distribution which is frequently seen in real-world problems and allows us to calculate whether an event happened due to random chance. The idea is that a set of 5 unique values is being created, namely 0 through 4, which are later translated to 1 through 5 for convenience, and then a random assignment of these values to households as a rating for a certain tip is being performed.

In more details, concerning the way the random ratings are generated, each cluster will have ratings from households that follow the binomial distribution but having a different success probability. Success probability in binomial distribution is how close we are to N, namely the "success" case which in our experiment is 4. The rest numbers, from 0 through 3 are handled as the "failure" case. In simple words, if the probability of success is low, then the frequency of numbers close to 4 will be less than those close to 0. So, for example, Cluster 1 has the success probability of 0.2, which means that the most frequently seen ratings will be 0 and 1 while rating values of 2,3, and 4 will occur less frequently. Then, Cluster 2 will have a probability of success of 0.4, Cluster 3 probability of success 0.6, and so on, meaning, also, that each Cluster's households share the same average rating.

The reason behind this idea is that if Cluster's households share similar energy consumption behavior, then they might rate similarly. For example, a group of households might give higher ratings for recommended tips than others, while some might give neutral ratings.

After generating the synthetic ratings dataset, 20% of ratings were excluded in order to have a sparse ratings matrix and were kept as a separate column for evaluation purposes. Finally, each household does not have a specific number of ratings, namely a household might have ratings for a large amount of tips e.g. 170/180, while others might have fewer ratings for available tips.

Overall, the ratings matrix has the following information in Table 3 where the first column references to household id, which is the same as the data id in Table 1, the second one to tip id, and the last columns reference the corresponding rating. In the case of **Rating** column, we have the rating of the household to a tip but 20% are missing. In the case of **All_Ratings** column, we have all the ratings of the household to a tip in order to use it for evaluation purposes.

Feature name	**Description**	**Count of Unique Values**	**Count of values**
Household ID	The Household ID from Pecan Street Dataset	612	-
Tip ID	The Tip ID from Tips Dataset	180	-
Rating	The rating that the household ID gave for the respective tip ID. 20% of the values are missing.	5	88.128
All_ratings	The rating that that the household ID gave for the according tip ID. 0% of the values are missing.	5	110.160

Table 3: Feature names, their descriptions and further information for the Ratings dataset

As presented in Figure 7, the rating value 1 occurs ~10.000 times (10.5%), 2 occurs ~20.000 times (17.5%), 3 and 4 values ~21.000 times (19.6 and 19.7 %, respectively), and 5 ~13.000 times (12.6%). Additionally, the mean rating is 3.079 and the standard deviation 1.27.

Finally, the Ratings matrix is used in Matrix Factorization and in the first part of Real-Time rules.

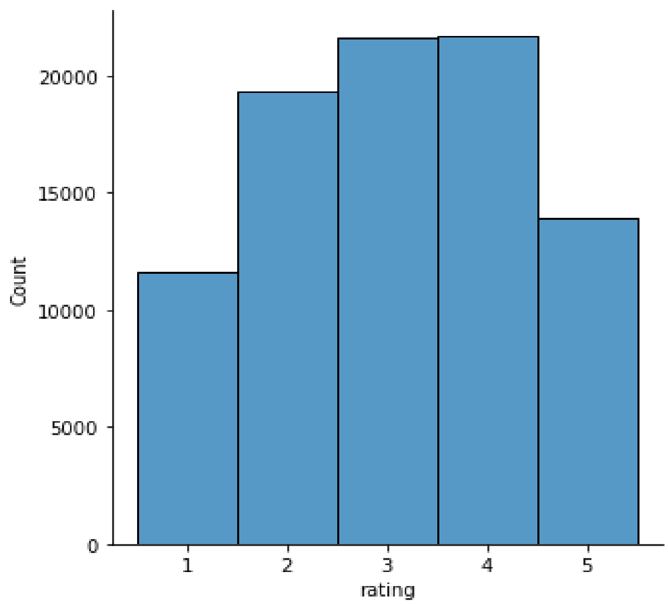


Figure 7: Distribution of ratings. It is obvious that the frequency of appearance of each rating value follows a distribution like normal distribution.

3.4 | Use of datasets within the Recommender System

The use of each dataset in each step of the Final Recommender System is presented in Table 4 below.

Dataset	**Use in Step**
Pecan Street Dataset	Clustering, Real-Time Rules
Synthetic Dataset with Ratings	Matrix Factorization, Real-Time Rules
Tips Dataset	Matrix Factorization, Real-Time Rules

Table 4: Use of datasets in each step of the Final Recommender System

The Pecan Street Dataset is used in Clustering and in “Real-Time rules”. While in Clustering, the overall system considers the user’s past energy consumption behavior and extracts some features that train the K-Means algorithm. While in “Real-Time rules”, past energy consumption behavior and devices used by the target household are being used to filter out irrelevant tips resulted from the Matrix Factorization Prediction list.

The Synthetic ratings matrix was generated after Clustering due to the finalization of the total households that will train the Matrix Factorization models for predictions.

Last but not least, the dataset with tips is mandatory for creating the Synthetic ratings matrix because it contains the item that the Recommender System provides recommendations. Additionally, information about each energy tip is significant in both Matrix Factorization and “Real-Time rules” in order to provide the description of the recommendation.

Chapter 4 | Recommender System Approach

This chapter consists of three parts. The first part showcases the steps involved in the proposed Recommender System, the second part presents the conducted Experiments for deciding the most optimal approach for implementing Collaborative Filtering, and the third part provides a simulation of the proposed Recommender System.

In more detail, the first part provides detailed information about the basic structure of the proposed Recommender System. More specifically, a detailed description of the training features for K-Means algorithm and the number of clusters is provided, while evaluation of Clustering is conducted through several evaluation metrics. Then, figures and diagrams present information about the ratings distribution of the cluster's households. As for the second part of the proposed Recommender System, namely the Matrix Factorization, Singular Value Decomposition algorithm is presented. Finally, a detailed explanation of "Real-Time rules" is given.

The second part of this chapter is dedicated to the experiments. The conducted experiments highlight the significance of Matrix Factorization in Recommender Systems. At the same time, the experiments are divided by 3 methods of imputation of missing ratings in order to make the ratings matrix denser and provide even more accurate predictions. The evaluation of error in predictions in each of the experiments is done through 5-Fold Cross Validation method and the evaluation metric used for the best approach is Root Mean Squared Error (RMSE), while the quality evaluation of the provided recommendations is done using Precision @ 15 and Recall @ 15 comparing with a dummy-static Recommender System. Additionally, the same quality evaluation was done after applying the 1st Part of the "Real-Time rules" due to filtering of the prediction lists from MF models.

The third part showcases a simulation of the proposed Recommender System for two households.

4.1 | Parts of the proposed Recommender System

In this section, detailed information about each step of the proposed Recommender System, as shown in Figure 8, can be found with the respective evaluation. In more detail, information will be given about the Clustering task with K-Means algorithm, the prediction phase from Matrix Factorization models, and, finally, the appliance of "Real-Time Rules". Additionally, the final version of the proposed Recommender System is presented along with diagrams with detailed information.

In summary, the proposed Recommender System predicts, with satisfactory accuracy, the unknown ratings of households using the respective Matrix Factorization model depending on the cluster where the target household belongs to and then, the "Real-Time rules" are applied in the sorted prediction list of each household in order to filter out irrelevant energy tips, prioritize the remaining ones based on "weight" and, finally, provide the top-1 recommendation after observing the real-time energy consumption of the target household's electrical devices for increased consumption. Graphically, the aforementioned are presented in Figure 9, so the interaction between datasets used and the households within the RS.

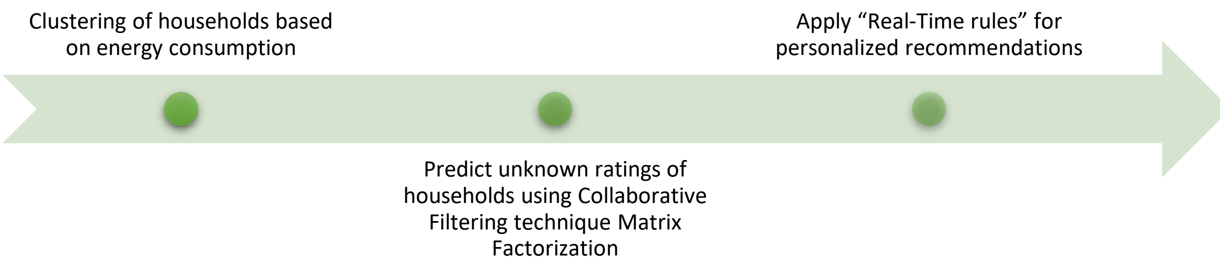


Figure 8: The three parts of the proposed Recommender System

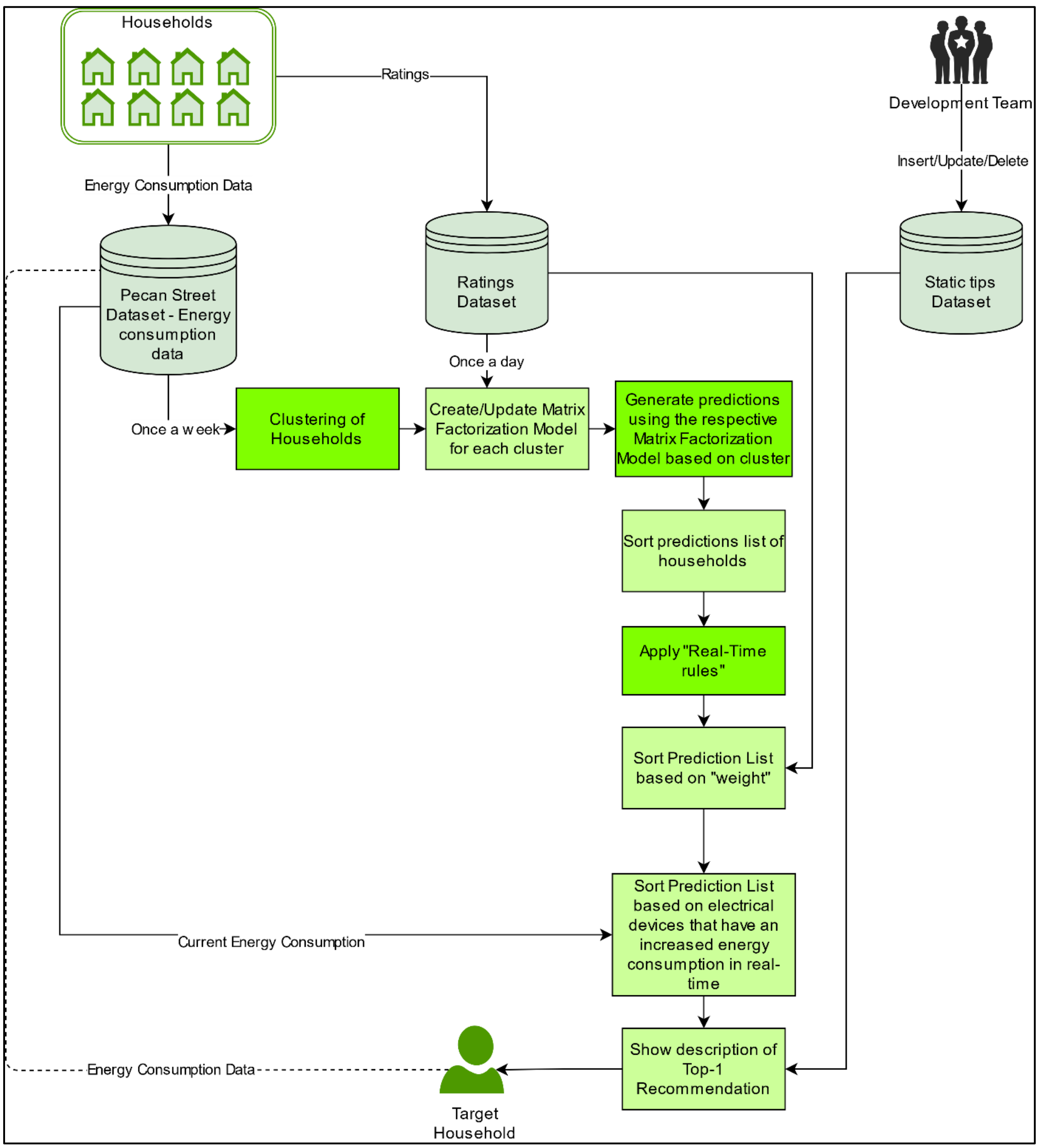


Figure 9: Data Flow and Step-by-Step diagram of Final Recommender System

4.1.1 | Clustering – Feature Selection and Optimal K

After loading the Pecan Street Dataset, the creation of the energy profile of each household begins. This energy profile consists of features that take into account the total energy consumption drawn by the electrical grid and the photovoltaic system of a household in specific cases e.g. by season, by weekends, etc. In more detail, the features are the following:

- Average energy consumption for each household for Weekdays (Monday through Friday).
- Average energy consumption for each household for Weekend days (Saturday and Sunday).
- Normalized energy consumption per hour for each household (Use per hour divided by the average consumption of the household)
- Average energy consumption for each household
- Average energy consumption of each household for each month

The other two features assume that a day is divided by time periods. The first time period is Overnight, started from 22:00 through 6:00, the second one is Breakfast from 6:00 through 9:00, the third one is Daytime from 9:00 through 15:00 and the final one is Evening from 15:00 through 22:00. This idea was first mentioned by *(Haben, Singleton, & Grindrod, 2016)* which analyzed multiple time series of energy consumption data from smart meters and conclude in these 4 time periods within a day.

Time period 1	**Time period 2**	**Time period 3**	**Time period 4**
Overnight 22:00 – 6:00	Breakfast 6:00 – 9:00	Daytime – 9:00 – 15:00	Evening – 15:00 – 22:00

Table 5: Time periods

Taking the above into consideration, there are two features that were found from the work of (Haben, Singleton, & Grindrod, 2016):

1) **Seasonal Score**
2) **Weekend vs Weekday Difference score**

$$Seasonal\ Score = \left|\frac{p_{w_1} - p_{s_1}}{P_{1_{mean}}}\right| + \left|\frac{p_{w_2} - p_{s_2}}{P_{2_{mean}}}\right| + \left|\frac{p_{w_3} - p_{s_3}}{P_{3_{mean}}}\right| + \left|\frac{p_{w_4} - p_{s_4}}{P_{4_{mean}}}\right|$$

Equation 9: Seasonal Score

$$Weekend\ vs\ Weekday\ Difference\ Score = \left|\frac{p_{wd_1} - p_{we_1}}{P_{1_{mean}}}\right| + \left|\frac{p_{wd_2} - p_{we_2}}{P_{2_{mean}}}\right| + \left|\frac{p_{wd_3} - p_{we_3}}{P_{3_{mean}}}\right| + \left|\frac{p_{wd_4} - p_{we_4}}{P_{4_{mean}}}\right|$$

Equation 10: Weekend vs Weekday Score

Here for each household and time period, a calculation for both is being made:

- The Average consumption of household $P_{1_{mean}}, P_{2_{mean}}, P_{3_{mean}}, P_{4_{mean}}$
- The Average consumption in Summer $p_{s_1}, p_{s_2}, p_{s_3}, p_{s_4}$
- The Average consumption in Winter $p_{w_1}, p_{w_2}, p_{w_3}, p_{w_4}$

- The Average consumption on Weekdays $p_{wd_1}, p_{wd_2}, p_{wd_3}, p_{wd_4}$
- The Average consumption on Weekend days $p_{we_1}, p_{we_2}, p_{we_3}, p_{we_4}$

The above two features were normalized by the average consumption in each equivalent period (in seasonal score per season, in weekend-weekday score per the observed weekday or weekend day) so that households with low average energy consumption do not take into account as households with a higher change in energy consumption behavior in comparison with households with high average energy consumption (Haben, Singleton, & Grindrod, 2016). At the same time, due to differentiation in demand that was not included in seasonal or weekday vs weekend scores, the standard deviation was included in calculations to measure the household's behavioral diversity.

All these features create an energy profile for each household to train K-Means and if a household does not have data for a specific feature, then it is excluded. So, even if the dataset contains data for 1.000 households, the above features rejected some of them end up having 612 households for training the Clustering model and train, at a later stage, the MF algorithm used for generating recommendations.

As far as the number of clusters is concerned, the Elbow Curve indicated that the preferable k number for k-Means, concerning the training data, is 4.

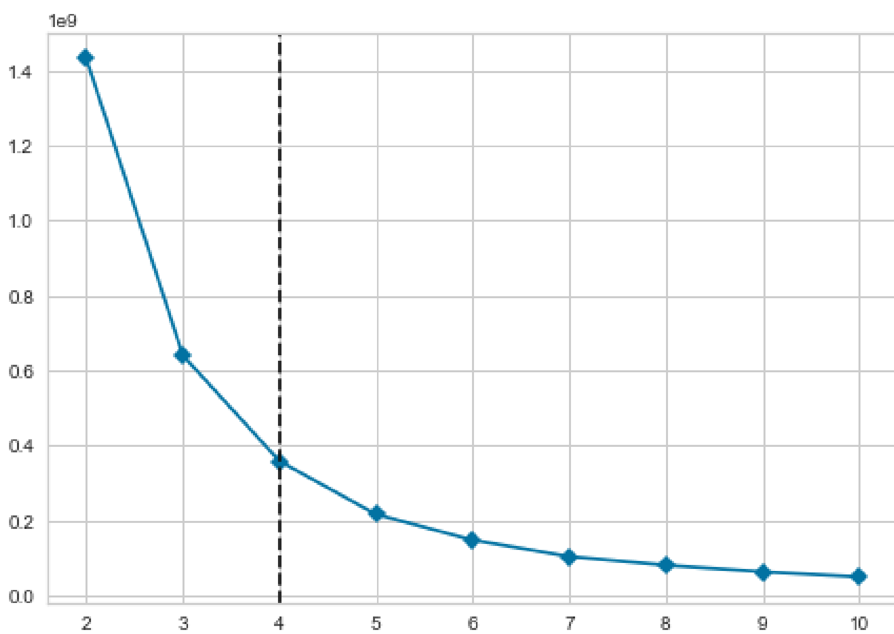


Figure 10: Results for optimal number of clusters based on K-Elbow Curve

As far as the Hopkins Statistic is concerned, which is used to assess the clustering tendency of a dataset by measuring the probability that a given dataset is generated by uniform data distribution, the resulted score is ≈0.87, which indicates that the training dataset, namely the energy profiles created, has meaningful clusters to get created, so Clustering is a reasonable task. As mentioned in (Haben, Singleton, & Grindrod, 2016), "choosing the correct attributes and the number of them is potentially the most important aspect of a successful clustering".

Finally, the average value of silhouette score is ≈0.59 which indicates that the clusters well represent each household based on features given.

Table 6 has all the information concerning the "rating behavior" of each group. Cluster 1 has 141 households and ratings are generated with low success probability, so ratings are generally low. Next, Cluster 2 has 142 households that rate neutrally, and Cluster 3 households tend to give higher ratings. Finally, Cluster 4 has a lot of households (180) that give high ratings and for that reason, the overall mean rating is greater than 2.5.

Cluster	Success probability	Count of users	Mean rating
1	0.2	141	1.8
2	0.4	142	2.6
3	0.6	149	3.38
4	0.8	180	4.19

Table 6: Cluster's average ratings

4.1.2 | Matrix Factorization – Singular Value Decomposition (SVD)

Matrix Factorization models map both users and items to a join latent factor space of dimensionality, such that user-item interactions are modeled as inner products in that space (Equation 1) (Ricci, Rokach, & Shapira, 2015). However, much of the observed variation in rating values is due to effects associated with either users or items, known as *biases* or *intercepts*, independent of any interactions (Koren, Matrix Factorization Techniques for Recommender Systems, 2009).
So, if biases for items and users added, then the baseline estimator changes to Equation 11:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

Equation 11: Predicted Rating formula

Where μ is the global average rating, b_i indicates the observed deviations of item i, or so-called item bias, and b_u the user's u, or so-called user bias, respectively, from the average. Also, q_i is a vector which is associated with an item i and p_u with a user u, respectively. As far as $p_u q_i^T$ is concerned, this is the resulting dot product that captures the interaction between user u and item i.

In order to learn the model parameters $b_i, b_u, q_i, p_u,$ the minimization of the regularized squared error changes to Equation 12:

$$\min_{b^*,q^*,p^*} \sum_{(u,i)\in K} r_{ui} - \mu - b_i - b_u - q_i^T p_u + \lambda(b_i^2 + b_u^2 + \left|\left|q_i\right|\right|^2 + \left|\left|p_u\right|\right|^2)$$

Equation 12: Minimization of the regularized squared error

Here, the observed ratings broken down into its four components: global average μ, item bias b_i, user bias b_u, and user item interaction $q_i^T p_u$. This allows its component to explain only the part of a signal relevant to it. Also, the constant λ avoids overfitting by penalizing the magnitudes of the parameters and is usually determined by cross validation. The system learns by minimizing the squared error function, namely, to have a lower value of error between true and predicted rating (Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, 2008); (Paterek, 2007). Because both q_i and p_u are unknowns, Equation 12 is not convex. However, if we fix one of the unknowns, the optimization problem becomes quadratic and can be solved optimally. Minimization, in the case of Singular Value Decomposition, is typically performed by Stochastic Gradient:

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u)$$

$$q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i)$$

where $e_{ui} = r_{ui} - \hat{r}_{ui}$. These steps are performed over all the ratings of the trainset and repeated $number\ of\ epochs$ times. Baselines are initialized to 0, user and item factors are randomly initialized according to a normal distribution, while learning rate γ and the regularization term λ.

Finally, in case of the proposed Recommender System, the regularization term λ is set to 0.02, the learning rate γ to 0.005. After several experiments, those values generate more accurate results based on RMSE. An observation was that high values of regularized term of items led to poor performance for the dataset used, while higher number of iterations did not offer higher accuracy at all.

Detailed evaluation of SVD should be found in 4.2 section where experiments are presented concerning the most accurate algorithm and approach based on the data available.

4.1.3 | Real-Time Rules

Having in mind that energy consumption has a direct relationship with the time of the year, day of week, and hour within the day, a set of “Real-Time rules” were created to filter the prediction list of Matrix Factorization Models and provide even more personalized recommendations based on real-time conditions concerning the time frame of the recommendation taken place and the energy consumption of the target household.

In more detail, “Real-Time rules” are divided into two parts, Part 1 and Part 2. The first one aims to exclude irrelevant energy tips that are not applicable due to the season, time period, and hour of the day where the recommendation takes place. Also, it excludes energy tips that apply in electrical devices that are not used by the target household. Finally, a “weight” is calculated that considers the high ratings of the target household and the Cluster’s households, so that it prioritizes the energy tips left on the prediction list. As for the second part, the final recommendation results are based on the current energy consumption of the target household and its past energy consumption behavior. The past energy consumption is, essentially, the average energy consumption of the target household for the current season, day of week i.e. weekday or weekend, and hour of the recommendation for each electrical device that the energy tips of the filtered prediction list apply to. These averages are considered as a threshold concerning the current energy consumption and if just one of the resulting electrical devices surpasses this threshold, then the top-1 energy tip that applies in this electrical device is being recommended to the target household. However, if no electrical device expresses an energy consumption concern, then the energy tip that applies to the electrical device that the target household has given the highest ratings, essentially based on “weight”, is considered as the top-1 and is being recommended.

“Real-Time rules” were created having in mind the features used in Clustering. In more detail, as features of Clustering has to do with the season, month, hour, and day of the week, we thought that similar filtering cases of the predictions list from Matrix Factorization models may offer a more personalized experience to the target household in real-time.

In more detail concerning the “Real-Time rules”, the steps are the following:

While offline, the proposed Recommender System should:

1) Create Clusters with available households based on their energy consumption behavior
2) Predict a list of recommendations $predictions_u$ from Matrix Factorization models using Singular Value Decomposition algorithm for user u where $u \in K$, which K contains all households in ratings matrix R
3) Find $devices_u$ used by household $u \in K$
4) Find $households_c$, namely all the households $u \in K$ that appear in the same cluster c

While in real-time:

Part 1

1) Find $month$
2) Find $season$
3) Find $time\ period$ (i.e. 12:00 p.m. then Morning)
4) Find $day\ of\ the\ week$
5) Find $week\ day$, namely if it is a week day or a weekend day
6) Keep only the prediction list generated from Matrix Factorization models $predictions_u$ for target household u
7) Find the $cluster_u$, namely the cluster of target household $u \in K$

Having found the information above, we continue by filtering the predictions list of the respective Matrix Factorization model in order to keep only relevant energy tips based on the time frame where the recommendation takes place and the electrical devices that smart meter data are collected for the target household:

8) $Results_u$ = $predictions_u$ having only $tips$ that apply on $devices_u$ of target household u
    a. If $Results_u$ has left with one tip, then revert action of step 8 and go to step 9
9) Update $Results_u$ having only $tips$ that apply on $season$ = {current season | All Seasons}
    a. If $Results_u$ has left with one tip, then revert action of step 9 and go to step 10
10) Update $Results_u$ having only $tips$ that apply on based on $time\ period$ = {current time period | All Time Periods} (e.g. it’s 21:00 so it’s Evening)
    a. If $Results_u$ has left with zero tips, then the recommendation procedure is set to trigger again in an hour that corresponds to time period of energy tips left
    b. If $Results_u$ has left with one tip, then Recommend the Top-1 from $Results_u$

After filtering out the prediction list with “irrelevant energy tips” based on time frame of the recommendation, we continue by finding the interests of the target household based on similar households, namely the households that belong to the same Cluster with the target household. A tip that has been already rated from the target household and similar households and holds a 4 or 5 rating value for both households, is considered as interesting.

11) For $households_c$ where c is the cluster where target household u belongs:
    a. $count(r)_{c_{device}}$ = Count ratings r for each $device$ in $Results_u$
    b. $countHigh(r \in \{4{,}5\})_{c_{device}}$ = Count ratings r with 4-5 value for each $device$ in $Results_u$
12) For target household u
    a. $count(r)_{u_{device}}$ = Count ratings r for each $device$ in $Results_u$
    b. $countHigh(r \in \{4{,}5\})_{u_{device}}$ = Count ratings r with 4-5 value for each $device$ in $Results_u$
13) Compute $Device\ Ratio_{c_{device}} = \frac{countHigh(r \in \{4{,}5\})_{c_{device}}}{count(r)_{c_{device}}}$
14) Compute $Device\ Ratio_{u_{device}} = \frac{countHigh(r \in \{4{,}5\})_{u_{device}}}{count(r)_{u_{device}}}$

Finally, a "weight" is calculated for each electrical device, so that each tip from the filtered prediction list that applies in each of the devices will be ranked higher than others if the target household finds its electrical device more interesting:

15) Compute $weight_{device_u} = \frac{Device\ Ratio_{u_{device}}}{Device\ Ratio_{c_{device}}}$
    a. If $Device\ Ratio_{c_{device}} is\ 0$ continue to the next device
    b. If $r_{u_{device}} = 0$, namely target household u do not have ratings for the specific $device$ then $weight_{device_u} = \max\,(weight_{device_u})$
16) $SortedResult_u\ =\ Results_u$ sorted based on $weight_{device_u}$

Part 2

In this part, past energy consumption behavior and real-time energy consumption play a vital role to detect possible increased consumption of the target household's electrical devices. The most important note in each of the conditions below is that if there is no data for a certain electrical device "Real-Time rules" will try to minimize the "loss" by keeping, at least, one case where the time frame of recommendation is respected. For example, if there are no data for the current day of the week, then we consider energy consumption of household on weekdays or weekend days, depending on the day of week. Similarly, if there is not enough energy consumption data to represent a season, then only data of the current month are considered.

17) $data_u$ = Energy consumption data of the target household u for all devices in $SortedResult_u$ for current $day\ of\ week$ *and hour*
    a. If $data_u = 0$ due to current $day\ of\ week$ then:
        i. $data_u$ = Energy consumption data of target household u for all devices in $SortedResult_u$ for current $week\ day$ ($Weekday\ or\ Weekend\ day$) and $hour$
    b. If $data_u = 0$ due to current $week\ day$ then:
        i. data$data_u$ = Energy consumption data of target household u for all devices in $SortedResult_u$ for current $hour$

18) Filter $data_u$ with data for Current $month$
    a. $average\ consumption\ of\ user\ month\_hour_{device}$ = Average consumption of target household u for $SortedResult_u$ **devices** for the Current $month$, $hour$ and ($week\ day\ or\ day\ of\ week$) based on **Past Data** (e.g. Saturday 10:00 a.m. average consumption on June)
19) Filter $data_{u_{device}}$ with data for Current $season$
    a. $average\ consumption\ of\ user\ season\_hour_{device}$= Average consumption of target household u for $SortedResult_u$ **devices** for the Current $season$, $hour$ and ($week\ day$) based on **Past Data** (e.g. Monday 9:00 a.m average consumption on Summer)
    b. $MonthsAvailable_{u_{device}}$= Number of months that data are available
20) If $average\ consumption\ of\ user\ month\_hour_{device}$ and $average\ consumption\ of\ user\ season\_hour_{device}$ are ≠ 0 and $Months\ available_{u_{device}} > 1$ then:
    a. $average\ consumption\ of\ user\ season\_month\_hour\ _{device}$ = $average\ (average\ consumption\ of\ user\ month\_hour_{device},$ $average\ consumption\ of\ user\ season\_hour_{device})$
21) If values from step 18a, 19a and 20a are 0 then:
    a. $average\ consumption\ of\ user\ for\ hour\ _{device}$ = Average consumption of target household u for $SortedResult_u$ **devices** for the Current $hour$ based on **Past Data**

So, as $average\ consumption_{device}$ we consider one of the following variables mentioned before:

$$average\ consumption\ _{device} = \begin{cases} average\ consumption\ of\ user\ season\_month\_hour\ _{device} \\ average\ consumption\ of\ user\ month\_hour_{device} \\ average\ consumption\ of\ user\ season\_hour\ _{device} \\ average\ consumption\ of\ user\ for\ hour\ _{device} \end{cases}$$

22) Find $current\ consumption\ of\ user_{device}$ for $SortedResult_u$ **devices**
23) $difference = current\ consumption\ of\ user_{device} - average\ consumption\ _{device}$

This is the last step of Part 2 of "Real-Time rules". This step will inform us if there is at least one electrical device that exceeds the average consumption that we found earlier so that top-1 energy tip which applies in such electrical device to be recommended. If there is no such case, then the top-1 from the filtered list of Part 1 is recommended.

24) If there's a $difference < 0$, recommend the tip that applies to the $device$ with $\max\ (difference)$
    a. Else recommend the top-1 of $SortedResult_u$

4.1.4 | Final Recommender System

The basic steps of the proposed Recommender System and the execution timeline without applying the "Real-Time rules" can be found in Figure 11. More specifically, Clustering should be done once a week, while training of the Matrix Factorization Models should be done once a day in order to predict unknown ratings of households.

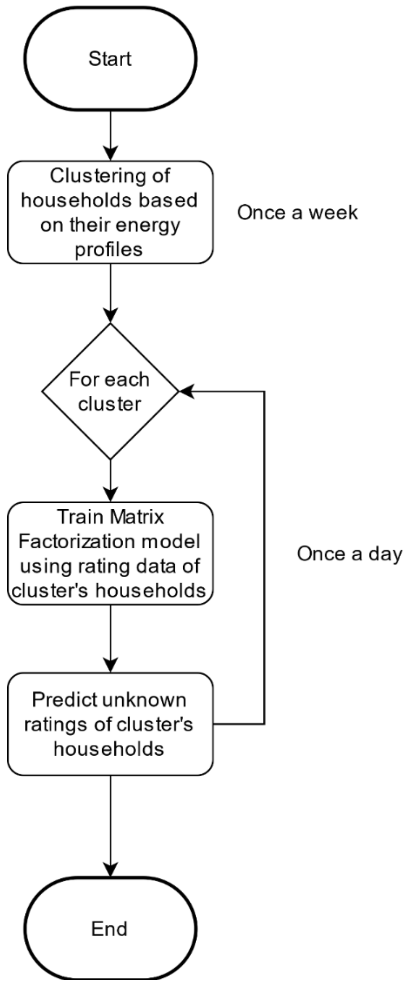


Figure 11: Flowchart of basic steps for Recommendation process

As far as the frequency of iterations about each basic step of the Recommender System is concerned, since the RS handles a lot of data, it is important to consider first the computational cost for each step. For this reason, Clustering should be done only once a week and in times where most of the households do not have an active session in the RS, while predictions with MF Models using SVD should be generated once a day. The reason behind each of these decisions is justifiable as follows. In clustering, we deal with energy consumption data, which is more difficult to change especially within a week. Of course, the energy consumption should positively change over time, so the weekly update should acquire changes in household's energy consumption behavior, if any. In Matrix Factorization models, due to ratings added all the time, updated information should be introduced.

The Final Recommender System has specific rules from applying some of its basic steps. More specifically, a household cannot belong to a Cluster if the RS does not have any energy consumption data collected from smart meters. The reason is that Clustering cannot provide the assignment of a household to a cluster because when calculating the energy profile of a user, it will be empty. Moreover, we will not know the Matrix Factorization model that should predict the unknown ratings of a target household. Eventually, the household will not get recommendations until providing smart meter data for at least one device. However, in case energy consumption data is available, so the target household belongs to a Cluster, the total absence of ratings is not a significant problem due to the imputation of missing ratings with the mean rating of Cluster that the target household belongs to as a first step. Thus, the respective Matrix Factorization model can provide a list of predictions for the target household, overcoming the Cold Start Problem. As far as the 1st Part of "Real-Time Rules" is concerned, it is evident that the lack of rating feedback from the target household will affect the effectiveness of the "weight" calculation. "Weight" is calculated based on rating data of the target household and households that belong within the same cluster as the target household. So, this step would not change the prediction list by prioritizing energy tips that apply in interesting, for the target household and cluster households, electrical devices. However, the filtering of the prediction list will function as expected due to the available time frame of the recommendation. As far as the 2nd Part of "Real-Time Rules", the absence of rating feedback from the target household is not a problem and will not affect its functionality.

The most important step is the trigger of the Recommender System for recommendation to the target household. If this is the first time in the specific day that a trigger has been fired, then the RS will proceed with all the steps below, otherwise, some steps will be avoided because they have been already executed and should continue from where we have left, namely the step where the trigger was set to fire again in the preferred time period.

4.2 | Experiments

In this section, a set of experiments are presented with their results concluding with the most accurate algorithm to use for the Matrix Factorization model(s) based on the available ratings matrix. The RMSE is obtained through 5-Fold Cross-Validation, a method that ensures that our predictions are not generated from an overfitted algorithm/model.

Finally, the quality evaluation of the produced recommendations of the final approach used in the proposed RS is presented in comparison with a dummy-static RS. With this, we highlight the significance of the proposed Recommender System in providing accurate recommendations in a fast and memory-efficient way.

4.2.1 | Experiment 1: Find the most optimal CF approach for the ratings matrix

The purpose of this experiment is to identify between K-Nearest Neighbors (Memory-based CB) and Matrix Factorization (Model-based CB) using SVD, which approach results in the most accurate predictions based on the ratings matrix while offering fast and memory-efficient training and testing times. In more detail, K-Nearest Neighbors is a Memory-Based algorithm that predicts using a distance or similarity metric. This class of algorithms provide a very fast training time since it just stores the data and their labels, but it is heavy in terms of storage. Also, the prediction phase is slow since it calculates all the distances to determine the k-Nearest Neighbors. Thus, is not practical for a large-scale system. Moreover, a frequent problem is that they do not generalize the data at all, namely, they tend to overfit the data. On the other hand, Matrix Factorization overcome the data sparsity problem, which is common in real-world datasets. Also, SGD is a relatively fast algorithm (Koren, Matrix Factorization Techniques for Recommender Systems, 2009) that learns based on the learning rate given as a hyperparameter.

Each of the aforementioned algorithms needs some pre-defined parameters to work. As seen in Table 7, K-Nearest Neighbor considers only 5 neighbors, the comparison is made between items and the predictions are calculated based on cosine similarity. In Table 8, SVD needs to have pre-defined the number of latent factors, the number of iterations where SGD will execute, the learning rate and the regularization term which is part of the cost function that needs to get minimized.

Parameters	Value of parameter
Number of Neighbors	5
Comparison between	Items (Item-based)
Similarity Metric	Cosine Similarity

Table 7: Parameters of K-Nearest Neighbors

Parameters	Value of parameter
Number of factors	2
Number of epochs	5
Learning Rate	0.005
Regularization term for the cost function	0.02
Add biases	Yes

Table 8: Parameters of Singular Value Decomposition

Finally, the unknown ratings of households are deleted.

Algorithm	RMSE	Train Time (s)	Test Time (s)
SVD	0.898	0.39	0.22
K-Nearest Neighbors	0.97	0.79	2.10

Table 9: Results of Experiment 1

As presented in Table 9, the needed time for the algorithm to learn is the training time, while the time for generating predictions is the testing time. So, SVD seems to result to the best RMSE score, while having a relatively small train and test time. Similar results were observed for SDG with minor differences. Finally, k-Nearest Neighbor provides neither good prediction accuracy nor good training and testing times.

4.2.2 | Experiment 2: Recommend using SVD algorithm with all ratings

The purpose of this experiment is to test four different methods of imputing missing ratings in order to find which one offers the most accurate predictions. Data imputation approach has been widely used, because it does not require additional data from other sources (e.g., trust networks and crowdsourcing) (Lee, Kim, Xie, & Park, 2018). Since the imputed value is not a real rating but an inferred rating, however, there may exist an error. Inferring the imputed value of a missing rating accurately is the key to reducing such errors and eventually improving the accuracy of CF.

As far as the different methods of imputation of missing ratings is concerned, the first one is to delete the missing ratings, the second and the third one is to replace them with the mean rating or median, respectively, while the fourth one is to replace the unknown ratings of each household with the mean rating of the cluster it belongs to.

Imputation method	RMSE	Train Time (s)	Test Time (s)
Delete unknown ratings	0.899	0.37	0.19
Fill unknown ratings with the mean rating	0.881	0.49	0.26
Fill unknown ratings with the median rating	0.88	0.58	0.19
Fill unknown ratings with the mean rating of each cluster	0.812	0.48	0.24

Table 10: Results of Experiment 2

From Table 10, it is obvious that the RMSE score is the same either imputing with mean rating or with median for both algorithms. Also, in the case of deleting unknown ratings of households, the RMSE score worsened because of the decreased amount of data. Finally, the best RMSE score is observed in the case of imputing missing data for each household with the mean rating of the cluster it belongs to.

Additionally, the train and test times for both algorithms have minor differences, except in the case of deleting the unknown ratings which is justifiable because the ratings data decreased.

4.2.3 | Experiment 3: Recommend using SVD algorithm with ratings from each cluster

The purpose of this experiment is to use the results of Clustering when creating a Matrix Factorization model with SVD algorithm. The idea is that a household belongs in the same cluster with other households that share similar energy consumption patterns. In that way, the ratings of households might be similar in real-world scenarios. In more detail concerning the experiment, a predictive model is created for each cluster, trained with the respective ratings of households, and tested with four different methods to impute missing ratings. The methods for imputing missing ratings are the same as in Experiment 2.

In more detail, the number of models is the same as the number of clusters, which is 4, and each household gets recommendations from the respective model, namely the one that represents the cluster where it belongs. Each model is trained with rating data from households that belong in the same cluster. The difference in the RMSE score of each model is due to the different training data.

This experiment is divided into two parts; The first one provides the RMSE score of each model for each imputation method concerning missing ratings, while the second one provides the average RMSE score of all MF models for each tested algorithm and each imputation method of missing ratings.

In Table 11, the RMSE score for each MF model is presented for each of the tested algorithms. From the results obtained, it is apparent that SVD provides the best results while imputing missing ratings with the mean rating of each cluster. The most significant part of this experiment is observed in training and testing times. This time the same prediction accuracy score was obtained but the time to train and test data reduced significantly. In more detail, both training and testing time of SVD is reduced by 75%. The reason is that the data used in each of these MF models are less due to the reduced number of households that appear in each training dataset. For example, 612 households train the MF model in Experiment 2, but in this experiment, households appeared in each cluster and will train the respective model is about 140.

Imputation method	Model 1 RMSE	Model 2 RMSE	Model 3 RMSE	Model 4 RMSE	Average RMSE	Average Training Time	Average Testing time
Delete unknown ratings	0.806	0.99	0.98	0.803	0.894	0.09	0.04
Fill unknown ratings with the mean rating	0.869	0.898	0.898	0.869	0.883	0.12	0.06
Fill unknown ratings with the median rating	0.869	0.898	0.899	0.868	0.883	0.15	0.07
Fill unknown ratings with the mean rating of each cluster	0.726	0.898	0.898	0.726	0.812	0.12	0.06

Table 11: Results of Experiment 3

From the results, it is obvious that even if we dealt with a synthetic dataset, this approach is promising and is a reliable way to reduce computational cost while keeping the prediction accuracy high.

4.2.4 | Results of Experiments

The final approach for generating predictions is presented in Figure 12.

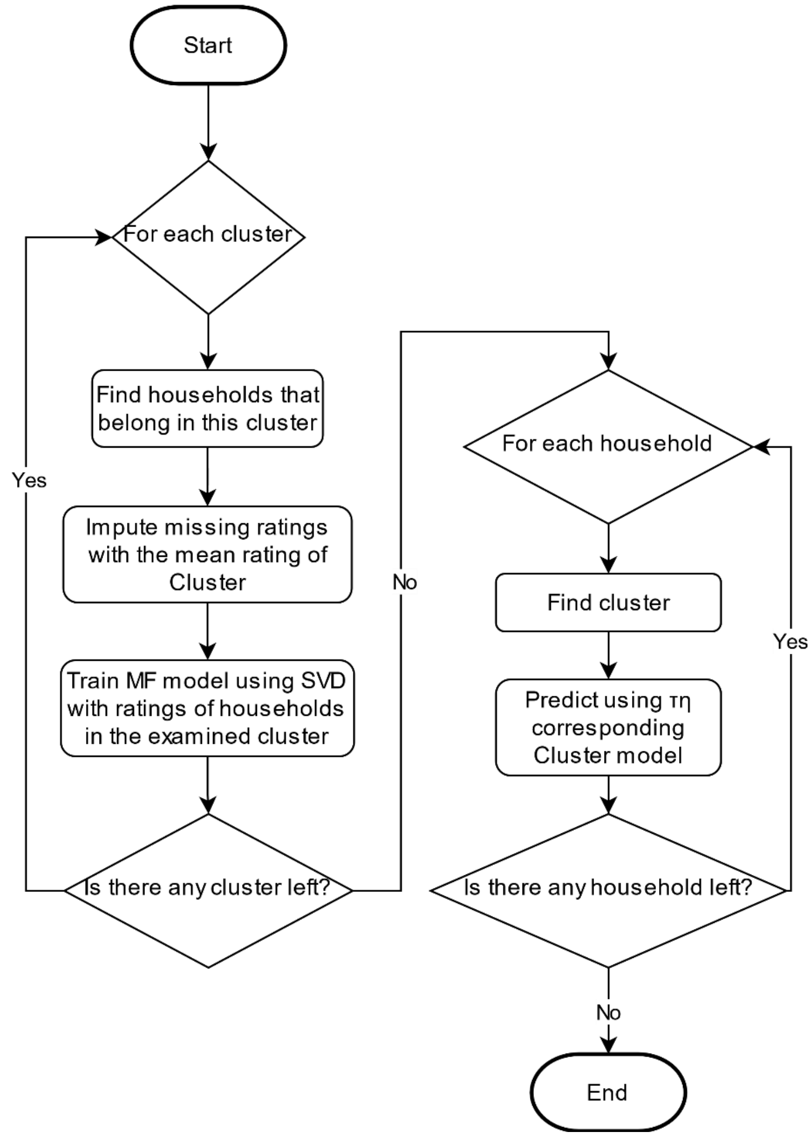


Figure 12: Final approach for generating predictions

To summarize, SVD gave the best results when we tried to create a denser matrix by imputing missing ratings with the mean rating of each cluster.

As far as the quality of the predictions of the proposed approach is concerned, the calculation of Precision @ 15 and Recall @ 15 for each MF model was calculated (essentially the average of the Precision @ 15 and Recall @ 15 of each household in each of the clusters) and presented in Table 12. In order to calculate these quality metrics, remember that when generating the synthetic dataset with ratings, 20% of them were made invisible in order to be considered as missing ratings of households and then kept as a separate column for evaluation purposes. In more detail, the unknown ratings of energy tips that their real rating is greater or equal to the mean real rating of the cluster where each household belongs to, from now on threshold, is considered as a **relevant** recommendation.

Additionally, the same quality evaluation of recommendations was conducted for a dummy-static Recommender System as shown in Table 13. This dummy-static RS assign to each unknown rating a value of 3, which is the mean rating of all households. What we want to prove with this dummy-static RS is the usefulness of the proposed RS in providing accurate and relevant recommendations in comparison with a simple Recommender System that generates dummy predictions.

Finally, as presented in Table 14, the average Precision @ 15 of the proposed RS resulted in 0.70, meaning that about 10 of the top-15 predictions obtained from Matrix Factorization Models using SVD are relevant, and the average Recall @ 15 resulted in 0.42 means that 42 % of the relevant tips of a household appear in the top-15 recommendation list. As far as the dummy-static RS is concerned, the dummy top-15 for households of each cluster generated an average Precision @ 15 of 0.45 and an average Recall @ 15 of 0.29. In summary, the proposed Recommender System has a Precision @ 15 25% better than a simple, ransom RS and a Recall @ 15 13% better. In other words, the proposed RS achieve to include 25% more tips that are relevant and 13% more tips of the proportion of relevant tips.

MF Model based on Cluster	**Precision @ 15**	**Recall @ 15**
1	57 %	41 %
2	53 %	42 %
3	81 %	41 %
4	83 %	42 %

Table 12: Precision @ 15 and Recall @ 15 for each MF Model - Proposed Recommender System

MF Model based on Cluster	**Precision @ 15**	**Recall @ 15**
1	58 %	41 %
2	52 %	41 %
3	82 %	42 %
4	0 %	0 %

Table 13: Precision @ 15 and Recall @ 15 for each MF Model – Dummy-static Recommender System

	Average Precision @ 15	**Average Recall @ 15**
Proposed RS	70 %	42 %
Dummy-static RS	45%	29 %

Table 14: Average Precision @ 15 and Recall @15 for both RSs

The reason why we observe a relatively good Precision @ 15 and Recall @ 15 from the dummy-static RS, is that the synthetic ratings matrix has a mean rating of 3, so that the most households give a rating of 3. But, what about if a cluster with households behave differently? What if none of the real ratings are 3? Can the dummy-static RS achieve to find a way to cope with this? No. This is the reason why the proposed Recommender System offers quality recommendations than a dummy-static Recommender System.

The same evaluation was done after applying Part 1 of "Real-Time rules". In short, Part 1 of "Real-Time rules" take as input the sorted prediction list and filters out irrelevant energy tips based on the recommendation time frame, the electrical devices that a target household use and the interests of a target household with its similar households. However, due to many possible combinations that can apply in Part 1 of "Real-Time rules", it was decided to test a single case to observe the quality of recommendations of the proposed Recommender System. The test case references to Winter season, Evening time period and Weekend day. Finally, Precision @ 15 and Recall @ 15 for both the RSs (proposed and dummy-static) result in the same evaluation score. That means that we achieve to retain the same quality of recommendations and at the same time keeping up with the recommendation time frame, the devices that each target household use and the interests of each household based on the "weight".

Finally, the proposed Recommender System succeed in improving the time cost (average training time of 0.12 s from 0.48 s and testing time 0.06 s from 0.24 s) and memory usage (each model is getting trained with about 27.500 ratings than 110.160) for providing predictions by 75%, having a Matrix Factorization model for each cluster. Moreover, it succeeded improving the prediction accuracy (average RMSE of 0.813) and quality (25% Precision @ 15 and 13% Recall better than a dummy-static RS) by imputing missing ratings with the mean rating of each cluster for each MF model and finally, introduce a set of filtering rules that offers personalized recommendations by taking into consideration the recommendation time-frame, the electrical devices used by each household, the interests of each household and its respective neighbors and, finally, by observing the real-time energy consumption of the target household in order to identify possible increased consumption so that to provide the most relevant recommendations in real-time.

4.3| Simulation

In this section, a simulation of the proposed Recommender System for two households will be presented. The analyzed energy consumption data from Pecan Street dataset will be from 2012 through 2013. Also, the current energy consumption for each household used in Part 2 of "Real-Time rules" is synthetic and aims to highlight the behavior of "Real-Time rules" through the Recommender System.

Detailed information for the first household #4352 is presented in Table 15, while for household #4298 in Table 16. In more detail, household #4352 has energy consumption data (e-gauge) from May through December of 2012 and January through June of 2013 for 16 devices. On the other hand, household #4298 has data for just one year beginning from late February 2013 through June 2013 for 9 devices.

User ID	4352
Cluster	3
Mean rating of Cluster	3.8
Unknown ratings	39/180
Start date - Collection of e-gauge data since	May 2012
End date - Collection of e-gauge data since	June 2013
Count of devices with smart meters	16

Table 15: Information about household #4352

User ID	4298
Cluster	3
Mean rating of Cluster	3.8
Unknown ratings	42/180
Collection of e-gauge data since	February 2013

End date - Collection of e-gauge data since	June 2013
Count of devices with smart meters	9

Table 16: Information about household #4298

The first part of the proposed Recommender System is Clustering. In this part, households that have available energy consumption data will be assigned to a cluster based on their energy consumption behavior. As far as the households of this simulation are concerned, both households were assigned to Cluster 3. The households that belong to this cluster are in total 149 and have a mean rating of 3.8 so they tend to give high ratings.

The second part of the proposed Recommender System is the prediction one. In this part, each of the available Matrix Factorization models will be trained with data from the households that belong to the respective cluster. Additionally, the missing ratings of households in this MF model will be replaced with the mean rating of cluster households, namely 3.8. In our case, the Matrix Factorization model No 3 will provide the prediction list for each of the households.

Finally, the top-5 prediction list of unrated tips for household #4352 is as presented in Table 17, while for household #4298 in Table 18. As far as the real rating is concerned, remember that when the synthetic ratings dataset was created, 20% of the ratings made invisible and kept as a separate column for evaluation purposes.

Tip id	**Appliance**	**Predicted rating**	**Real rating**
431	Use	3.42	3
23	Office	3.38	4
429	Lights	3.37	4
20	Office	3.36	4
53	All Rooms	3.31	4

Table 17: Top-5 Prediction list of MF Model for household #4352

Tip id	**Appliance**	**Predicted rating**	**Real rating**
101	All Rooms	3.44	3
431	Use	3.44	4
13	Range	3.41	4
20	Office	3.41	3
456	Refrigerator	3.40	5

Table 18: Top-5 Prediction list of MF Model for household #4298

After generating predictions for both households, it is time to apply "Real-Time rules". Remember that in "Real-Time rules" we do not only make use of the ratings of each household, but also the current and past energy consumption.

The time frame where the recommendation procedure takes place is Saturday (Weekend day) February 27 20:00 where the season is Winter, and the time period is Evening. For both households, the energy consumption data for some devices is for just one year (essentially one month) and for some others for two years (e.g. February 2012 and 2013).

As previously described, the 1st part of "Real-Time Rules" aims to filter the prediction list from the respective Matrix Factorization model based on devices used by the target household and the time frame where the recommendation takes place. Thus, from the energy tips that were generated as predictions from SVD, only 29 from 39 apply on devices used from household #4352, while for household #4298 only 25 from 42. So, in each case, we can proceed by filtering the prediction list with devices used by the target household. If there was only 1 tip left, due to the collection of smart meter data for the household for just one device from the available ones in the prediction list, then this rule would have not been taken into consideration. In this case, the predictions list would have more general energy tips meaning that some energy tips may apply in devices that no smart meter data are collected for the target household.

Next, we consider keeping only the energy tips that apply to the current season, namely Winter, or in all seasons. Again, if just one tip had left, we would not consider this filtering rule. For household #4352, the recommendation list was reduced by 7 energy tips resulting in 22 energy tips left to be recommended. For household #4298, the recommendation list was reduced by 6 energy tips resulting in 19 energy tips.

The third filtering condition keeps only energy tips that apply in the current time period, namely Evening, or in all time periods. If there were no such energy tips, so all energy tips that were left apply in the Morning, then the recommendation procedure would have set to trigger the next morning, namely from 6 a.m. through 5 p.m. In this case, both households have data that apply in the Evening. Household #4352 has now 19 energy tips left to be recommended, while household #4298 has 17 energy tips left.

Finally, the filtered list for household #4352 has 19 energy tips left to be recommended that apply in 14 devices used by the household, while household #4298 has 17 energy tips left that apply in 7 devices.

Right after, a weight is assigned to each device in the filtered list, so that later sorted in descending order to find the devices that the target household has the most interest in. In the calculation of weight, we do not take only into consideration the target's household ratings, but also ratings of households that belong in the same cluster with the target household. After the calculation, we find out that household #4352 has a special interest in energy tips that apply in **Cars** with a weight of 1.39 as show in Table 19. As for household #4298, its special interest belongs to devices related to **plugs and lights in exterior places** of a household with a weight of 1.18 as shown in Table 20.

Tip id	**Appliance**	**Predicted rating**	**Real rating**
397	Car	3.12	3
396	Car	3.18	4
251	Kitchen/Kitchen App/Refrigerator	3.18	3
474	Dishwasher/Clothes washer	3.20	3
394	Air/Air Window	3.30	2

Table 19: Top-5 filtered list for household #4352

Tip id	**Appliance**	**Predicted rating**	**Real rating**
63	Lights	3.27	5
246	Lights	3.27	3
220	Lights	3.25	5

18	Air/Air Window	3.32	4
215	Air/Air Window	3.30	4

Table 20: Top-5 filtered list for household #4298

After applying the steps described in Part 1 of "Real-Time rules", Part 2 begins. The goal is to find the device that has an increased energy consumption in real-time taking into consideration past energy consumption data and specifically the average consumption data of the target household.

As mentioned before, we have a small sample of energy consumption for both households in this simulation. A significant note that highlights the efficiency of the proposed Recommender System in handling unavailability of data is that whenever there is no data available for a device for the current conditions where the recommendation takes place, e.g. current season, day of the week, hour, month, then the RS considers only what is available. For example, if we do not have energy consumption data for at least 2 months of Winter, then the average consumption of the target household is not considered based on season but rather based on the rest available cases. Another example is when there is no data for a device for the current day of the week. In that case, we consider the week day, e.g. if Saturday (day of the week 6), then we consider the target household's average energy consumption data on Weekend days.

In more detail concerning the current energy consumption data, this refers to the energy consumption that the target household has in each of the used devices appeared in the recommendation list. As previously mentioned, depending on the energy consumption data available for the target household, the calculation of average consumption in each of the devices used by the target household is different. With that in mind, an observation of the average consumption of the target household in each of the used devices has been made in order create synthetic values as the real-time consumption for each device used from the target household that either surpass the average consumption or not.

So, the current energy consumption for some devices of household #4352 is presented in Table 21 and for household #4298 in Table 22:

Use	**Air 1**	**Bedroom 1**	**Car**	**Clothes washer**	**Dishwasher**	**Dryer**	**Furnace 1**	**Refrigerator**
1.97	1	0.08	0.87	0.18	0.0012	1.08	0.21	0.05

Table 21: Current energy consumption of household #4352

Use	**Air 1**	**Bathroom 1**	**Car**	**Clothes washer**	**Furnace**	**Jacuzzi**	**Outside Lights and Plugs**
0.85	0.8	0.01	1.31	0.7	0.01	0.12	0.186

Table 22:Current energy consumption of household #4298

Even though a household has installed smart meters in some devices, there are cases where no data is available from the date of enrolment in the Pecan Street Dataset but in a later time. Among different households, this case is apparent for household #4352, for example, for Car device where the collection of energy consumption data begun in May 2013 and through June 2013. So, in this case, where no data is available for the observed season (Winter) or the observed month (February), the **average energy consumption of household #4352** is considered based on observed hour (20:00) and on observed day (Saturdays). On the other hand, **household #4298** has energy consumption data for plugs and lights of exterior places of a household (a device) for the observed Month (February), so the **average energy consumption of household #4298** is considered based on observed month.

As mentioned before, real-time energy consumption has synthetic data created for this simulation. So, for household #4352 it was considered to give a higher value than the average consumption for Use, Car, Kitchen App, Dishwasher, Bedroom, Furnace, and Air devices. From the above mentioned, the device with the highest difference between average consumption and real-time consumption will be recommended. As far as the household #4298, none of the devices appeared in the filtered prediction list exceeds the average energy consumption in real-time. So, the top-1 from the filtered list of Part 1 of "Real-Time rules", which is sorted based on "weight", is recommended.

Finally, as shown in Table 23, household #4352 has its highest difference, based on the average consumption during Saturdays at 20:00, in real-time consumption based on past data for "**Use**", namely the summary of photovoltaics energy and energy drawn from the electric grid. In this case, the energy tips are more general. In Table 24, household #4298 does not exceed average consumption in any of the devices appeared in the filtered predictions list, so the top-1 recommendation from Part 1 of the "Real-Time rules" is recommended.

Household	4352
Tip Description	Irritated by the blinking light on the electric devices in stand-by mode? Turn off your gaming console instead of leaving it on stand-by while it is not in use
Appliance	Use
Time Period of appliance	General
Season of appliance	All
Real-Time Consumption	1.98
Average Consumption	1.80
Weight	0.99
Predicted Rating	3.41
Real Rating	3

Table 23: Recommendation for Household #4352

Household	4298
Tip Description	Remember to switch off lights when rooms are unoccupied.
Appliance	Lights
Time Period of appliance	Evening
Season of appliance	All
Real-Time Consumption	0.186
Average Consumption	0.1863
Weight	1.18
Predicted Rating	3.27
Real Rating	3

Table 24: Recommendation for Household #4298

Additionally, Figure 13 presents the average consumption at 20:00 o'clock for **use** for all days of the week for household #4352 and similar households that belong to the same cluster with the target household.

The red dot expresses the real-time consumption of the household concerning **use** and we can observe a significant increase.

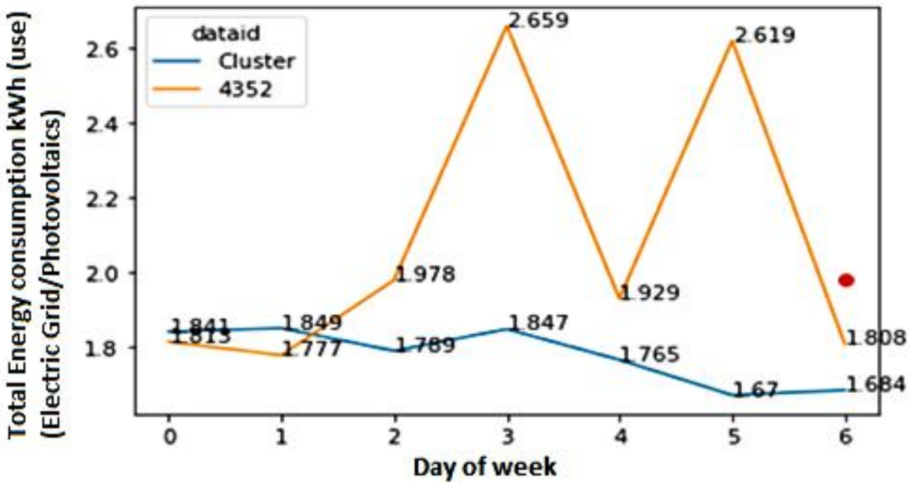


*Figure 13: Average energy consumption of "**use**" for household #4352 and corresponding cluster per Weekday at 20:00 o'clock. Red point points on current consumption.*

Figure 14 presents the average consumption on Saturdays and at 20:00 o'clock for **"outside lights and plugs"**, namely the energy consumption of the exterior lights and plugs of a household, for each month that energy consumption data are available for household #4298, in comparison with similar households that exist within the same cluster with the target household. As in Figure 13, the red dot expresses the real-time consumption of the household concerning **"outside lights and plugs"** and what we can observe is that the real-time consumption is almost equal with the average energy consumption, but the "Real-Time rules" do not raise concern about this. Essentially, Part 2 of the "Real-Time rules" is an extra procedure that aims to find if there is a device that exceeds, even a little, the average consumption of a specific device.

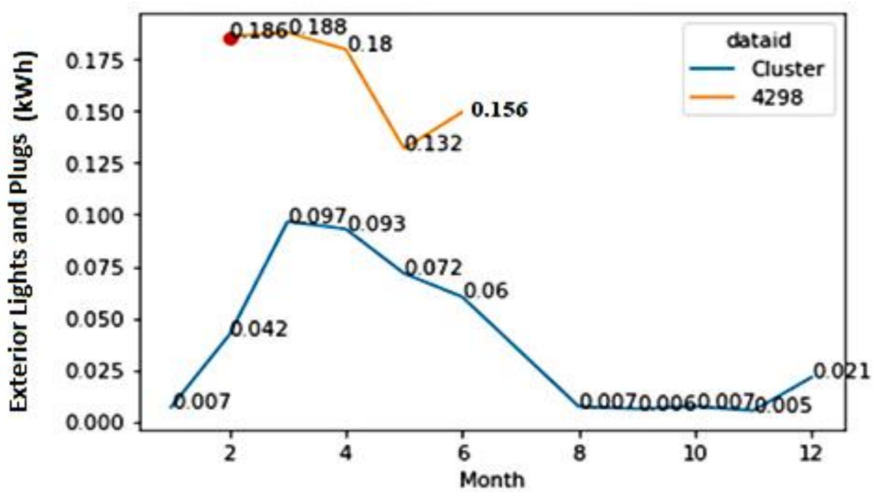


Figure 14: Average energy consumption of exterior plugs and lights for household #4298 and corresponding cluster per Month on Weekday 6 (Saturday) and 20:00 o'clock. Red point points on current consumption.

Finally, the appliance of “Real-Time rules” offers something personalized. We did manage to keep the energy tips that apply at least in one case concerning the recommendation time frame, we keep in tips that apply in devices used by the target household, we prioritize the filtered predictions list based on the “weight” that takes into account the interests from both target household and similar households in the same cluster, and finally, Part 2 of “Real-Time rules”, helped us to find what is the most suitable tip looking at the real-time consumption of the target household. In each case, if there were not a concern, we managed to have a list that considers many parameters that offer a personalized experience to each of the households in this simulation.

Chapter 5 | Conclusion

The proposed Recommender System succeed in improving the time and memory cost for providing recommendations by having a Matrix Factorization model for each cluster, improve the prediction accuracy and quality by imputing missing ratings with the mean rating of each cluster, introduce a set of filtering rules that offers personalized recommendations by taking into consideration for each household the recommendation time-frame, the electrical devices used, the interests of the household and its similar households, and finally, by observing the real-time energy consumption in order to identify possible increased consumption so that to provide the most relevant recommendations in real-time.

5.1 | Problems faced

The first challenge faced in developing such Recommender System in Smart Energy Grids is data acquisition. Lately, there are a lot of datasets[5] that share the energy consumption of households and demographic information, however, what is missing from these datasets is the collection of feedback for tips recommended to households when energy consumption is higher than usual for certain electrical devices. To resolve this, a synthetic dataset was created based on some assumptions that it is believed that reflect reality.

The second challenge was the lack of feedback even after households have rated a certain energy tip. It is not known if the recommended tip was useful at all after applying the real-time rules either by not having a “Done/Not Done” feedback from the household or by implicitly infer the energy consumption after the recommendation.

Another problem is the lack of energy consumption data. If the System does not have a lot of data, for example, it has data for just one day, then:

- Clustering will not work well
- Predictions will be irrelevant because the target household might belong in the wrong cluster and so the wrong model will predict unknown ratings

Also, if a device does not have a tip that applies to it, then households will not get recommendations for the specific device, neither the RS detect a possible increased energy consumption in this device. So, there is a need for enrichment of the static database with the energy tips in order that all the devices have an equal probability of getting recommended to households that smart meters collect data.

5.2 | Solution to common problems

As far as the Cold Start Problem is concerned, two solutions are adapted in the proposed Recommender System. First and foremost, Matrix Factorization technique can effectively provide accurate predictions

[5] https://data.mendeley.com/datasets/n85kwcgt7t/1,
https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households,
https://www.kaggle.com/jaganadhg/house-hold-energy-data,
https://www.pecanstreet.org/dataport/

even when a ratings matrix is sparse, essentially having a few data available for a new user. So, if a household belongs to a cluster, then even if there are a few ratings, the respective Matrix Factorization model can effectively provide accurate recommendations. Additionally, the imputation of missing ratings of households for tips with the average rating of the cluster it belongs significantly improve the accuracy of the predictions so as to provide to households that belong in a cluster, an accurate prediction. Essentially, cluster households that share similar energy consumption data, share also similar ratings based on the synthetic ratings dataset that tries to imitate a real one, and that could help a new household get accurate recommendations until achieving a denser rating matrix.

For the data sparsity problem, the solution was to use Matrix Factorization. This problem arises due to the lack of feedback which is common in most of the items that an RS uses.

Finally, computational and memory cost has been resolved using, at first Matrix Factorization and especially SVD, and therefore by introducing for each cluster an MF Model where a smaller set of users can provide even better predictions than all the users together.

5.3 | Future work

The proposed Recommender System consists of many state-of-the-art algorithms and methods that could provide personalized and accurate recommendations. However, we believe that the overall Recommender System can improve its accuracy and relevance by introducing some new features in the Clustering procedure.

As far as the ratings dataset is concerned, the first significant change is to introduce implicit feedback from the smart meters. An example of implicit feedback should be a decrease of energy consumption in a specific electric device after a sequence of recommendations concerning that electrical device. Also, if the Recommender System has a User Interface that provides a categorization of energy tips based on the appliance of an electric device, then if a household/user has a lot of views in the specific category, then this is a sign of interest.

Moreover, in case where implicit feedback is introduces, Alternating Least Squares (ALS) is the state-of-the-art algorithm used in Matrix Factorization. ALS is favorable because the system can use parallelization (Zhou & al., 2008). In ALS, the system computes each q_i independently of the other item factors and computes each p_u independently of the other user factors. Moreover, ALS is an algorithm that can be used in a Big Data environment, an advantage that gives a solution to the problem of data explosion where the time and space complexities are critical (Luo, Ranzi, Wang, & Dong, Service Recommendation in Smart Grid: Vision, Technologies, and Applications, 2016). For example, it is already implemented in Spark, a unified analytics engine for large-scale data processing[6].

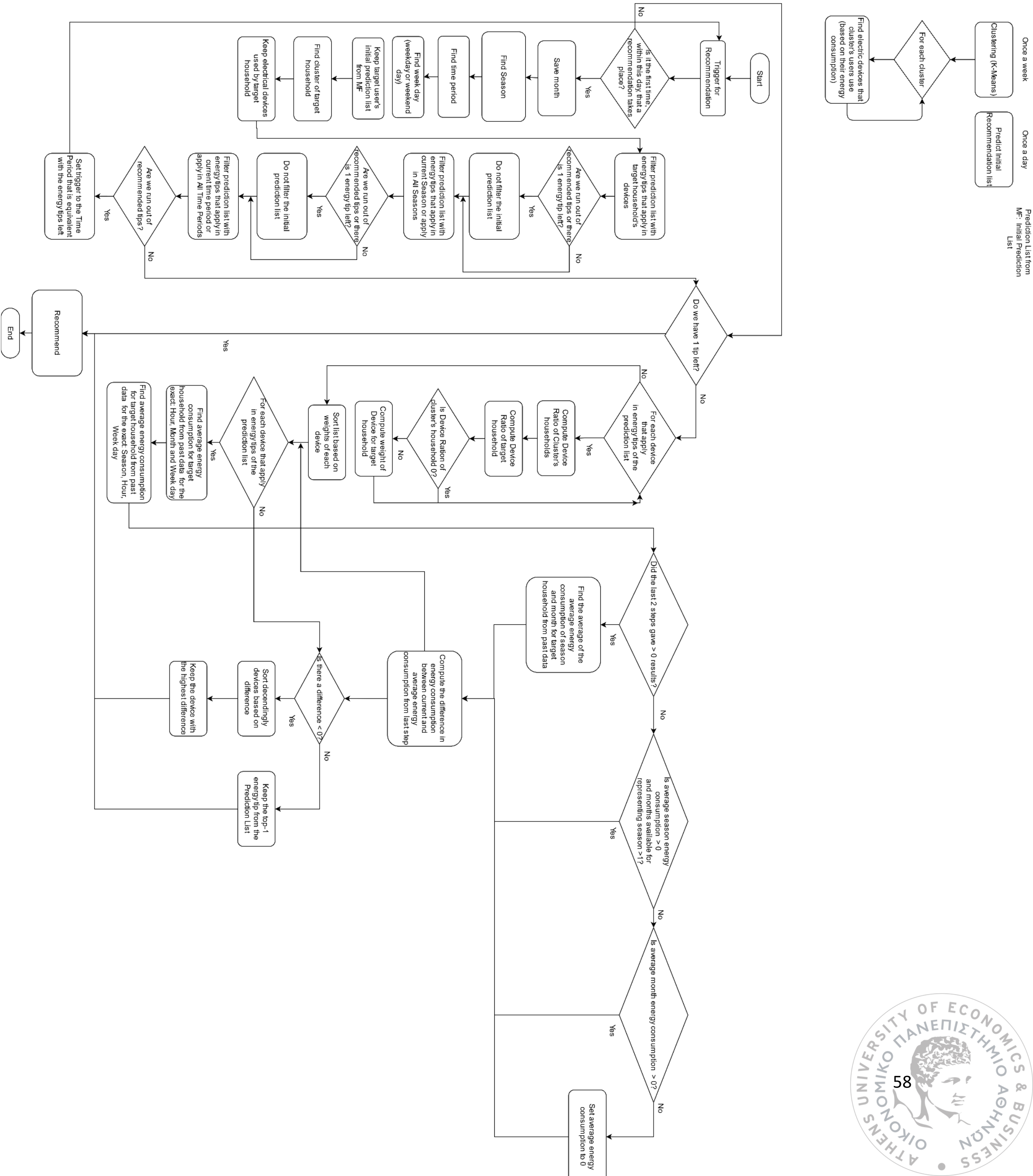
Also, a Recommender System is not only useful because of its accuracy, but also when presented accordingly within an elegant graphical user interface. With that in mind, in case this Recommender System integrates into a Mobile Application, some ideas after the recommendation is presented to the target household are presented:

[6] https://spark.apache.org/

1) Show the percentage of:
    a. All users in the application that find this energy tip useful
    b. Users in the same cluster with the target user that find this energy tip useful
2) Show a range of the reduction of energy consumption that users succeed after applying the recommended energy tip
3) Show a range of the amount of money that users save after applying the recommended energy tip

Additionally, the enrichment of the database with energy tips should be the next improvement of the proposed Recommender System including tips for public buildings such as schools, companies, etc. so that if the RS integrates within an application, would be accessible to a wider range of users.

Appendix


Once a week
Clustering (K-Means)
For each cluster
Find electric devices that cluster's users use (based on their energy consumption)
Once a day
Predict Initial Recommendation list
MF: Initial Prediction List from Prediction List
Start
Trigger for Recommendation
Is it the first time, within this day, that a recommendation takes place?
Yes
No
Save month
Find Season
Find time period
Find week day (weekday or weekend day)
Keep target user's initial prediction list from MF
Find cluster of target household
Keep electrical devices used by target household
Filter prediction list with energy tips that apply in target household's devices
Are we run out of recommended tips or there is 1 energy tip left?
Do not filter the initial prediction list
Filter prediction list with energy tips that apply in current Season or apply in All Seasons
Filter prediction list with energy tips that apply in current time period or apply in All Time Periods
Are we run out of recommended tips?
Set trigger to the Time Period that is equivalent with the energy tips left
Do we have 1 tip left?
Recommend
End
For each device that apply in energy tips of the prediction list
Compute Device Ratio of Cluster's households
Compute Device Ratio of target household
Is Device Ration of cluster's household 0?
Compute weight of Device for target household
Sort list based on weights of each device
For each device that apply in energy tips of the prediction list
Find average energy consumption for target household from past data for the exact Hour, Month and Week day
Find average energy consumption for target household from past data for the exact Season, Hour, Week day
Did the last 2 steps gave > 0 results?
Find the average of the average energy consumption of season and month for target household from past data
Is average season energy consumption > 0 and months available for representing season >1?
Is average month energy consumption > 0?
Set average energy consumption to 0
Compute the difference in energy consumption between current and average energy consumption from last step
Is there a difference < 0?
Sort decendingly devices based on difference
Keep the device with the highest difference
Keep the top-1 energy tip from the Prediction List

References

Agency, I. (2010). Energy Efficiency Governance. *Interantional Energy Agency*.

Aggarwal, C. (2015). *Data Mining.* Springer.

Aggarwal, C. (2016). *Recommmender Systems - The Textbook.* NY, USA: Springer.

Anandhan, A., Shuib, L., Ismail, M., & Mujtaba, G. (2019). Social media recommender systems: Review and open research issues. *IEEE Access 6*, 15608-15628.

Banerjee, A. (2004). Validating clusters using the Hopking statistic. *IEEE International COnference on Fuzzy Systems*, 149-153.

Bari, A., Jiang, J., & Saad, W. (2014). Challenges in the Smart Grid Applications: An Overview. *International Journal of Distributed Sensor Networks*. doi:https://doi.org/10.1155%2F2014%2F974682

Bell, R., & Koren, Y. (2007). Scalable Collaborative Filtering with Joinly Derived Neighborhood Interpolation Weights. *IEEE Int. Conf. Data Mining (ICDM 07)* (pp. 43-52). IEEE CS Press.

Bell, R., Koren, Y., & Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. *13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 95-104). New York, USA: ACM.

Billsus, D., & Pazzani, M. (1998). Learning collaborative information filters. *15th Int. Conf. on Machine Learning* (pp. 46-54). San Francisco, CA, USA: Morgan Kaufmann.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *Use rModel User-Adap Inter*, 331-370. doi:https://doi.org/10.1023/A:1021240730564

Canadian Electricity Association. (2010). The Smart Grid: A pragmatic Approach. Retrieved from http://www.electricity.ca/media/SmartGrid/SmartGridpaperEN.pdf

Collaborative Recommendations and Adaptive Control for Personalised Energy Saving. (H2020, 6 10). Retrieved from Cordis Europa: https://cordis.europa.eu/project/id/723059

Crawley, D., & Huang, Y. (1997). Does it matter which weather data you use in energy Simulations. *User News*, pp. 1-11.

Davoli, F., Repetto, M., Tornelli, C., Proserpio, G., & Cucchetti, F. (2012). Boosting Energy Efficiency through Smart Grids. *ITU*. Retrieved from http://www.itu.int/dms_pub/itu-t/oth/4B/01/T4B010000050001PDFE.pdf

Ekstrand, M., Riedl, J., & Konstan, J. (2011). Collaborative Filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 175-243.

Elbow Method (Clustering). (2021, 1 2). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Elbow_method_(clustering)

Farhangi, H. (2010). The path of the smart grid. *IEEE power and energy magazine*, 18-28.

Fischer, J., Ramchurn, S., Osborne, M., Parson, O., Juynh, D., Alam, M., . . . Jennings, N. (2013). Recommending Energy Tariffs and Load Shifting based on Smart Household Usage Profiling. *IUI.* Santa Monica, USA.

Funk, S. (2006). Netflix Update: Try this at Home. Retrieved from http://sifter.org/~simon/journal/20061211.html

Gunawardana, A., & Shani, G. (2015). Evaluating Recommender Systems. In *Recommender Systems Handbook* (pp. 265-308). Springer.

Haben, S., Singleton, C., & Grindrod, P. (2016). Analysis and Clustering of Residential Customers Energy Behavioral Demand using Smart Meter Data. *IEEE Transactions on Smart Grids*, 136-145.

Häubl, G., & Trifts, V. (2020, 19). Consumer Descision Making in Online Shopping Environments: The effects of Interactive Decision Aids. *Marketing Science*.

Hopkins Statistic. (2020, 1 13). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Hopkins_statistic#cite_note-1

Hopkins, B., & Skellam, J. (1954). *A new method for determining the type of distribution of plan individuals.* Annals of Botancy Co.

Hsu, M., Lin, T., & Ho, T. (2012). Design and implementation of an intelligent recommendation system for tourist attractions: The Integration of EBM molde, Bayesian Network and Google Maps. *Expert Systems with Applications*, 3257-3264.

Huang, Y., & Bian, L. (2009). A Bayesian Network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet. *Expert Systems with Applications*, 933-943.

Isinkaye, F., Folajimi, Y., & Ojokoh. (2015). Recommendation Systems: Principles, methods and evaluation. *Egyption Informatics Journal 16*, 261-273.

Jain, A., & Dubes, R. (1988). *Algorithms for clustering data.* Prentice-Hall, Inc, Upper Saddle River.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender Systems: An Introduction. New York: Cambridge University Press.

Jeeninga, H., & Huenges, B. (2007). Domestic Electricity Consumption and Life Style. *Netherlands Energy Research Foundation*.

Keim, D., Andrienko, G., Fekete, J., Gorg, C., Kohlhammer, J., & Melancon, G. (2008). Visual Analytics: Definition, Process, and Challenges. *Information Visualization*, pp. 157-175.

Knijnenburd, B., & Willemsen, M. (2015). Evaluating recommender systems with user experiments. In *Recommender Systems Handbook* (pp. 309-352). Springer.

Kohavi, R., Longbotham, R., Sommerfield, D., & Henne, R. (2009). Controlled experiments on the web: Survey and practical guide. *Data Mining Knowledge Discovery 18*, 140-181.

Kong, W., & Liang, G. (2020). A Personalized Residential Energy usage Recommendation System based on load minitoring and Collaborative Filtering. *IEEE Transactions on Industrial Informatics*.

Koren, Y. (2008). Collaborative Filtering for Implicit Feedback Datasets. *Proc. IEEE Int. Conf. Data Mining (ICDM 08)* (pp. 263-272). IEEE CS Press.

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. *Proceeding of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 426-434). New York, USA: ACM.

Koren, Y. (2009). Collaborative Filtering with Temporan Dynamics. *Knowledge Discovery and Data mining (KDD 09)* (pp. 447-455). ACM Press.

Koren, Y. (2009). Matrix Factorization Techniques for Recommender Systems. *IEE Computer Society*, 42-49.

Koren, Y. (2010). Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. *ACM Trans. Knowl. Discov. Data*, (p. 24).

Kwac, J., Flora, J., & Rajagopal, R. (2014). Household Energy Consumption Segmentation. *IEEE TRANSACTIONS ON SMART GRID*, 420-430.

Lee, Y., Kim, S.-W., Xie, X., & Park, S. (2018). How to Impute Missing Ratings? Claims, Solution, and Its Application to Collaborative Filtering. Lyon, France.

Leskovec, J., Rajaraman, A., & Ulman, J. (n.d.). Mining massive datasets. Cambridge: Cambridge University Press.

Litos Strategic Communication. (2008). The Smart Grid: An Introduction. Retrieved from http://energy.gov/oe/downloads/smart-grid-introduction-0

Lops, P., Gemmis, M. d., & Semeraro, G. (2011). Content-Based recommender systems: State of the art and trends. In *Recommender Systems Handbook* (pp. 73-105). Springer.

Luo, F., Ranzi, G., Wang, X., & Dong, Z. (2016). Service Recommendation in Smart Grid: Vision, Technologies, and Applications. *9th International Conference on Service Science.*

Luo, F., Ranzi, G., Wang, X., & Dong, Z. (2019). Social Information Filtering-Based Electricity. *IEEE Transactions on Smart Grid*, (pp. 95-105).

Martinez, E., Lairner, J., & Keating, K. (2009). Pursuing energy-efficient behavior in a regulatory environment: Motivating policymakers, Program Administrators, and Program Implementers. *California Institure for Energy and Environment*.

McLaughlin, M., & Herlocker, J. (2004). A Collaborative Filtering Algorithm and Evluation Metric that accurately model the user experience. *27th annual internation ACM SIGIR conference on Research and development in information retrieval.* ACM Press.

Montaner, M., Lopez, B., & Rosa, J. d. (2003). *A taxonomy of recommender agents on the internet.* Artificial Intelligence Review.

Murphy, P. (2010). *Enabling tomorrow's electricity system.* Ontario, USA: Ontatio Smart Gird Forum.

Nobrega, C., & Marinho, L. (2014). Predicting the learning rate of Gradient Descent for Accelerating Matrix Factorization.

Paterek, A. (2007). Improving Regularized SVD for Collavorative Filtering. *KDD Cup and Workshop* (pp. 39-42). ACM Press.

Pazzani, M., & Billsus, D. (2007). Content-Based Recommendation Systems. (Springer, Ed.) *The adaptive web*.

Pecan Street Dataset. (n.d.). Retrieved from https://www.pecanstreet.org/dataport/

Portugal, I., Alencar, P., & Cowan, D. (n.d.). *The use of Machine Learning Algorithms in Recommender Systems: A Systematic Revie.* Canada.

Quijano-Sanchez, L., Cantador, I., & M Cortes-Cediel, O. G. (2020). Recommender systems for smart cities. *Information Systems 92*.

Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook.* New York: Springer.

Rousseeuw, P. (1987). Silhouettes: A Graphical aid to the Interpretation and Validation of Cluster Analysis. *Computational and Applied Mathematics*, 53-65.

Sarwar, B. M., & al., e. (2000). Application of Dimensionality Reduction in Recommender System - A case study. *Workshop on Web Mining for e-Commerce: Challenges and Opportunities (WebKDD).* ACM Press.

Schafer, J. B., Konstan, J., & Riedl, J. (2001). E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 115-153.

Schein, A., Popescul, A., Ungar, L., & Pennock, D. (2002). Methods and metrics for cold-start recommendations. *25th Annual International ACM SIGIR Congerence on Research and Development in Information Retrieval* (pp. 253-260). ACM.

Schiaffino, S., & Amandi, A. (2009). Building an expert travel agent as a software agent. *Expert Systems with Applications*, 1291-1299.

Schweizer, D., Zehnder, M., Wache, H., Witschel, H., Zanatta, D., & Rodriguez, M. (2015). Using consumer behavior data to reduce energy consumption in smart homes. *IEEE 14th International Conference on Machine Learning and Application.*

Takács, G., & al., e. (2007). Major Components of the Gravity Recommendation System. *SIGKDD Explorationgs*, (pp. 80-84).

U.S. Department of Homeland Security. (2011, November). Blueprint for a secure cyber future.

Xu, D., & Tian, Y. (2015). A comprehensive survey of Clustering Algorithms. *Ann. Data. Sci.* Springer.

Xu, R., & Wunsch, D. (2005). Survey of Clustering algorithms. *IEEE Trans Neural Netw*, (pp. 645-678).

Zeng, T., & Wang, D. (2019). A kind of Recommendation Algorithm of Matrix Decomposition based on Power Grid user rating characteristics. *IEEE Innovative Smart Grid Technologies - Asia*, 4180-4185.

Zhao, Q., Harper, F., Adomavicius, G., & Konstan, J. (2018). Explicit or implicit feedback? Engagement or satisfaction? A field experiment on machine-learning-based Recommender Systems. *33rd Annual ACM Sympsium on Applied Computing*, (pp. 1331-1340).

Zhou, Y., & al., e. (2008). Large-Scale Parallel Collaborative Filtering for the Netflix Prize. *Proc. 4th Int. Conf. Algorithmic Aspects in Information and Management* (pp. 337-348). Springer.