



FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGICAL CENTER
ELECTRICAL AND ELETRONICS ENGINEERING DEPARTMENT

Validation of a pre-clustering strategy for a recommender system

Undergraduate Thesis presented to the Federal University of Santa Catarina as a requisite for the bachelor degree of Electronics Engineering

Matheus Frata

Advisor: André Carvalho Bittencourt

Co-advisor: Felipe Campos Penha

Florianópolis, 2019.

MATHEUS FRATA

**VALIDATION OF A PRE-CLUSTERING
STRATEGY FOR A RECOMMENDER
SYSTEM**

Undergraduate Thesis presented to
the Federal University of Santa
Catarina as a requisite for the bach-
elor degree of Electronics Engineer-
ing. Advisor: André Carvalho Bit-
tencourt. Co-advisor: Felipe Cam-
pos Penha

**FLORIANÓPOLIS
2019**

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Frata, Matheus
Validation of a pre-clustering strategy for a
recommender system / Matheus Frata ; orientador,
André Carvalho Bittencourt, coorientador, Felipe
Campos Penha, 2019.
54 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro
Tecnológico, Graduação em Engenharia Eletrônica,
Florianópolis, 2019.


Inclui referências.

1. Engenharia Eletrônica. 2. Recommender Systems.
3. Clustering. 4. Principal Component Analysis. I.
Carvalho Bittencourt, André. II. Campos Penha,
Felipe. III. Universidade Federal de Santa
Catarina. Graduação em Engenharia Eletrônica. IV.
Título.

Matheus Frata

VALIDATION OF A PRE-CLUSTERING STRATEGY FOR A RECOMMENDER SYSTEM

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Engenharia Eletrônica e aprovado em sua forma final pelo Curso de Graduação em Engenharia Eletrônica.



Prof. Fernando Rangel de Sousa, Ph.D.
Coordenador do Curso

Banca examinadora:



André Carvalho Bittencourt, Ph.D.
Orientador
Neoway Business Solutions



Prof, Danilo Silva, Ph.D.
Universidade Federal de Santa Catarina



Felipe Campos Penha, Ph.D.
Co-Orientador
Neoway Business Solutions



Prof. Richard Demo Sousa, Ph.D.
Universidade Federal de Santa Catarina

To her, to them, and too all the ones that made me who I am.

Acknowledgments

First and foremost, I want to highlight the education from my parents, Adelar and Valdete. Their lessons and guidance enabled me to stand where I am today, and I cannot put in words how grateful I am for their support.

Secondly, I want to say "I love you" to the girl that I met way back when I still was in high school. She has been supporting me ever since. It has been 8 years of love, partnership and friendship. Thank you very much my Dear Caroline!

Thirdly, I want to give a shout out for all the folks that experienced UFSC life with me and still in contact: Lari, Zimpel, Griep, Ion, Karina, Roberto, Ruan, Claudio, Gui, Alevato, Elder, Eiterer. Also other folks that come and go: Giu, Penteado, Pedro Lemos, Kupas, Tortato. Deni you are going to deliver my certificate at my graduation!! (If you are reading this, and your name did not make it to this list, you should get in touch with me!!). Moreover, I want to thank the people from the institution: the faculty of EEL and other departments, the public employees of UFSC, and the labs that gave an opportunity to me: the Spacelab and the LCS.

Finally, I want to thank Neoway for the opportunity to develop this work. Specially the people at the Analytics Team! These people are the reason, for me, that makes Neoway a great place to work. Thank you very much for André, Penha, Igor, Breno, Mariana, Leandro, Victor, Roger, Yuri, Leonardo, Felix, and Manoel. Also, a shout out for other teams (like the girls from GG and guys of SEC), and other employees that are not in the company anymore, but somehow had their contributions to my career!

ABSTRACT

In this work we used a proof of concept to validate a pre-clustering strategy using firmographics data in order to improve the performance of Neoway's recommender system. The context of recommender systems is discussed, together with an introduction of the concept of clustering and principal component analysis. The terminology of the recommender system and its benchmark are explained, along with the internal blocks of the system. Clustering procedures are discussed and two experiments are proposed. The results showed a slight improvement on the performance of the system, but it is not enough to proceed with the further investing in the approach.

Keywords: Recommender Systems. Clustering. Principal Component Analysis.

Acronyms

B2B	Business to Business
DBSCAN	Density-Based Spatial Clustering of Applications with Noise.
IRS	Information Retrieval System
OPTICS	Ordering Points To Identify the Clustering Structure
OT	OnTarget.
PCA	Principal Component Analysis.
PoC	Proof of Concept
RS	Recommender System.
SVD	Singular Value Decomposition

List of Figures

1.1	Venn Diagram representing the sets involved in the recommender system.	2
1.2	Similarity score from a case at the benchmark. (a) shows a best case scenario one and (b) scenario to be investigated.	3
2.1	User-based logic on collaborative filtering. Source: [1]	8
2.2	Rectangular box with colored balls.	11
2.3	Reordering of the balls.	12
2.4	Lift plot for the reordered box with the random chance.	12
2.5	Steps of PCA for two variables. Source: Adapted from [2].	15
3.1	Block Diagram of the On Target.	18
3.2	Modifications on the OT done by the Benchmark.	19
3.3	Fake study data set that shows how both clustering strategies work.	21
3.4	Illustration of the cluster pairing. On the top is the OvO and on the bottom the OvA.	22
3.5	PCA plot for some studies. For each study (a column) the first row is only the portfolio data, second row both market and portfolio (market in orange and portfolio in blue).	23

3.6	Results of the clustering on the first two principal components of three studies.	25
3.7	Modifications on the OT for the experiment RpC.	26
3.8	Modifications on the OT for the experiment CaF.	27
4.1	Histogram plot of the studies' lift gain for experiment RpC.	30
4.2	Histogram plot of the studies' lift gain for experiment CaF.	32
4.3	Similarity distribution plot for Study 4 in experiment RpC.	33
4.4	Similarity distribution plot for Study 4 for experiment CaF.	33
4.5	PCA plot for Study 4. On the left there is just the portfolio, on the right both portofflio and market.	34
4.6	Similarity distribution plot for the clusters' runs of Study 22 on experiment RpC.	35
4.7	Similarity distribution plot for Study 5 on experiment CaF. An example of marginal increase on the lift.	36
4.8	Similarity distribution plot for Study 12 on experiment RpC. An example of marginal decrease on the lift.	36
4.9	Similarity distribution plot for Study 8 on experiment CaF. An example of considerable increase on the lift.	37
4.10	Similarity distribution plot for Study 9 on experiment RpC. An example of considerable decrease on the lift.	38
4.11	Similarity distribution plot for Study 24 on experiment CaF.	39
4.12	Lift plot for Study 24 on experiment CaF.	39
4.13	Similarity distribution plot for Study 25 on experiment CaF. An example of study with multiple high density areas in the portfolio.	40
4.14	Clusters' similarity distributions plots for Study 25 in experiment RpC.	41
4.15	PCA plot with portfolio and market data for Study 25. . .	42
4.16	Similarity distribution plot for Study 7 on experiment CaF. Lift increased but the curves skewed to the left.	43
4.17	Lift gain histogram without outliers of other clustering algorithm runs for experiment CaF	44

List of Tables

- 4.1 Summary of the first-decile lift gains for Experiment RpC . 30
- 4.2 Summary of the first-decile lift gains for Experiment CaF . 31

Contents

1	Introduction	1
1.1	Objectives	4
1.1.1	General Objectives	4
1.1.2	Specific Objectives	4
1.2	Work outline	4
2	Literature Review	7
2.1	Recommender systems	7
2.1.1	Types of Recommender Systems	8
2.1.2	Benefits to business	9
2.1.3	Evaluation	10
2.2	Clustering	13
2.3	Principal Component Analysis	14
3	Methodology	17
3.1	About the Product	17
3.1.1	On Target	17
3.1.2	Benchmark	18
3.2	Clustering	20
3.2.1	Clustering strategy and pairing	20
3.2.2	Number of clusters	21

3.2.3	Cluster algorithm	24
3.3	Experiments	26
3.3.1	Run per Cluster	26
3.3.2	Cluster as Feature	27
4	Experimental results	29
4.1	Experiment Run per Cluster	29
4.2	Experiment Cluster as Feature	31
4.3	Similarity distributions	32
4.3.1	Studies with no change to the lift	32
4.3.2	Studies with marginal increase or decrease on the lift	35
4.3.3	Studies with considerable increase or decrease to the lift	37
4.3.4	Outliers studies	38
4.3.5	Studies that had multiple profiles in its portfolio . .	40
4.3.6	Other relevant studies	42
4.4	Experiment CaF with other clustering algorithms	43
5	Conclusion	47
5.1	Takeaways	48
	Bibliography	51

CHAPTER 1

Introduction

Neoway Business Solution is a Big Data Analytics company with solutions to Sales & Marketing and Risk & Compliance that are suited to companies in a variety of business verticals and branches. One of its products is called On Target(OT), a lead recommendation system, based on scoring potential markets according to a given portfolio of clients. The OT will search for leads in a subset of the whole Brazilian's market space, which is composed by all active companies. The user can narrow down the search space based on a set of filters. The Figure 1.1 shows a Venn diagram that illustrates the subsets of the On Target. The Portfolio set is composed by the user's clients; The Market is where the On Target will search for the leads, it can vary from a set defined by the filters to all the Brazilian active companies (except the user's clients).

Recently, the OT was updated, and a benchmark was created to compare between different versions. Overall the new version showed an improvement in the accuracy performance and consistency of the recommendation. By analysing cases in the benchmark, we conjectured that the performance could be further improved in some cases if perhaps the user's portfolio had been pre-clustered before running the

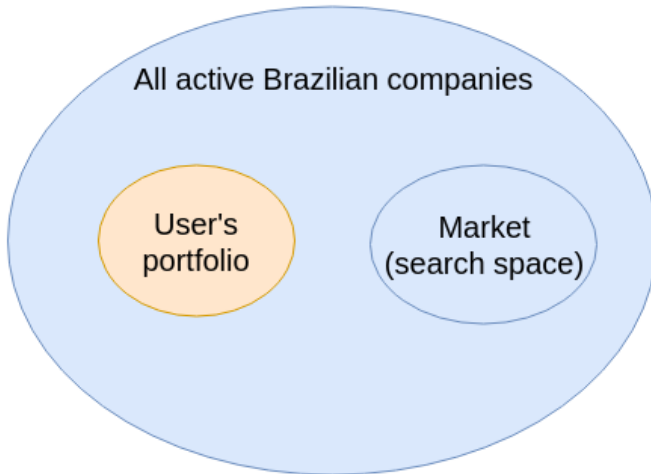


Figure 1.1: Venn Diagram representing the sets involved in the recommender system.

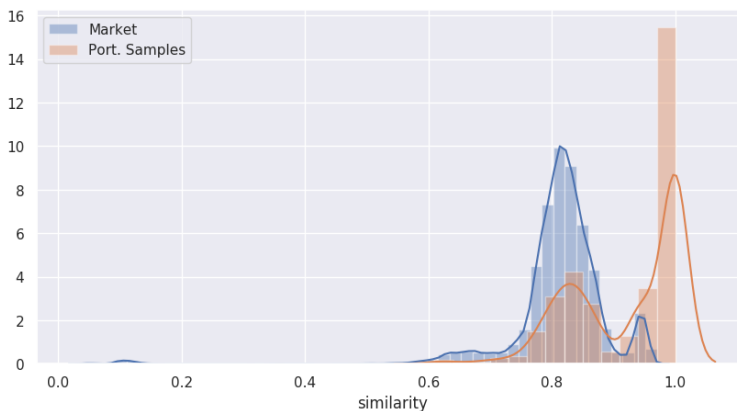
recommender system, leading to a more homogeneous and consistent scoring of the market. The last point can be understood through Figure 1.2 which shows the distribution of the recommender score in a benchmark case. The recommender system is trained with samples from the user's portfolio and market and its score allows for an interpretation of a similarity measure. The higher the score, going from 0 to 1, the more similar is a company to companies in the user's portfolio, i.e. the more similar they are, and are thus better qualified to be converted to a lead or client¹.

Figure 1.2a is the best case scenario of a study, where all the portfolio's samples are close to similarity 1.0. The market has two high density probabilities areas: one close to similarity 0 and other close to similarity 1.0, in others words, some companies have no fit with the portfolio and others are alike the portfolio. The latter is the high quality leads to recommend to the user. Figure 1.2b is an example of a study with a behavior that may be improved. The portfolio's sample distribution has two concentrations of mass along the x-axis which can be interpreted that there were two types of customers within the

¹More about the similarity distribution plot will be explained on the Chapter 3.



(a)



(b)

Figure 1.2: Similarity score from a case at the benchmark. (a) shows a best case scenario one and (b) scenario to be investigated.

user's portfolio set. The aim of this thesis is to investigate whether this behavior can be improved by a pre-clustering strategy.

One hypothesis of the On Target's team is that in this study (and others alike) the client has a heterogeneous portfolio, **meaning that it can have two or more distinct profiles in it**. The algorithm tries to optimize for the mean profile of the whole portfolio which may not be the best approach.

This work is one of the several improvements that have been considered in the On Target Product Roadmap. It is a proof of concept which its objective is to analyze whether the overall performance of the high quality leads generation improves by clustering the portfolio before running the recommendation algorithm.

In order to preserve interpretability, the clustering procedure will not include all available features. Only firmographics data will be used [3] which are data related to characteristics of a business, such as: company size, location, number of employees and others.

1.1 Objectives

1.1.1 General Objectives

The general objective of this work is to analyze whether the performance of the On Target improves by pre-clustering the portfolio with firmographics data before running the recommendation algorithm.

1.1.2 Specific Objectives

Specific objectives of this work are:

- defining the pre-clustering strategy;
- defining the pre-clustering algorithm;
- defining the number of clusters for the benchmark cases;
- running On Target against the benchmark with the pre-clustering approach; and
- analyzing the distribution of similarity scores and performance metrics against the benchmark.

1.2 Work outline

This thesis is organized in the following manner:

In the Chapter 2 some theoretical concepts are explained, so the reader can understand the what theory is behind this work.

Next, in the Chapter 3, we introduce the terminology used by the OT's team and a basic view of how it works, later in the chapter, a

discussion about the pre-clustering procedures and the conceived experiments.

In Chapter 4, is presented the results of these experiments alongside an extensive analysis in the metrics of several scenarios.

Finally, in Chapter 5, a recap of this work is presented to the reader, with the takeaways.

CHAPTER 2

Literature Review

In this chapter some basic concepts are presented to the reader to contextualize the work. First, it will be introduced the concept of recommender system (RS) and its importance in today's digital business strategies. Second, the metric used to evaluate the performance of the RS algorithm: lift. After that, a general discussion on clustering will be presented followed by one technique to visualize the clusters in a dataset: Principal Component Analysis.

2.1 Recommender systems

A RS is a software which its main purpose, as the name suggests, is to give suggestions [4] or recommendations to a user based on information about the user itself or the context of the items to recommend. They are classified as part of information filtering systems [4]. So, another way to interpret the recommendations is to think as the result of a filter applied on a search space, where only the most relevant information is given back to the user. It is important to emphasize that usually the search space on the RSs today are enormous. For example, there are more than 400 million products on Amazon US available [5]. Meaning

that, in terms of time, it is unreasonable for a human to search one by one, analysing multiple attributes of millions of items at the same time. Hence, the importance of this type of system on the applications today.

2.1.1 Types of Recommender Systems

There are three main types: based on **collaborative filtering**, the ones on **content-based filtering**, and **hybrid**.

The first one is **collaborative filtering**, where the recommendations come mainly from information generated by the user and its interaction with the items [4]. For instance, one type of logic is the **user-based**, where a item is recommended to the user based on what other similar users like, more known as *people like you, also like X*. Another logic is the **item-based**, where the RS acts based on the similarity between items, also known as *if you like X you may like Y*. Figure 2.1 shows an example of user-based collaborative filtering, where the RS wants to predict what is the preference on headphones of user E. Based on users B and C (they voted similarly to E), we can see that, probably its a *dislike*.











					
A		✓	✗	✓	✓
B			✓	✗	✗
C		✓	✓	✗	
D				✓	
E		✓	✓	?	✗

Figure 2.1: User-based logic on collaborative filtering. Source: [1]

The second one is **content-based filtering**, where the recommendations are based on the features of the items and more importantly:

an **user profile** [4]. The system needs an input from the user, be it interactively like in a sequential manner or historical data, or in batch where the user gives a lot information about itself at once. Having the profile, the RS can cross this information with the features of the items to look for items that are similar to the user's profile. It can remember the item-based logic from collaborative filtering, but here we have added the information of the user. One example that illustrate content-based, is an article reader service. Imagine when one signs up to the service, it will ask a lot of question to the new user: "What type of articles do you prefer: Sports, Politics, Health?", "What kind of Sports do you like: Soccer, Volleyball, Baseball...?". These questions build up a profile of the new user and it will be refined as the users utilizes the service. Therefore, if our hypothetical new user answers "Sports" and "Soccer" to the previous questions, the article service will start recommending articles of soccer. But if this new user only thumbs up Brazilian Soccer articles, the RS will learn that and narrow down the recommendations to Brazilian soccer.

Finally, there is the **hybrid** type, where the Recommendation System is based on a mixture of both previous types [6]. Combining them can improve the recommendations and at the same time deal with their constraints. Collaborative filtering has the problem of **cold start**, whenever a new item or user is added, it has no attributes [6]. So it will take some time until its attributes are filled up. This problem leads to another one which is the **sparsity** on the matrix of attributes. Meaning that, there are a lot of missing values. Content-based filtering can supply this data to the RS. And combining with the accuracy of collaborative filtering the system can achieve very personalized recommendations.

2.1.2 Benefits to business

The RSs are present on a myriad of online services today, bringing great value to them. Streaming platforms of music and videos have RSs on their business to increase user retention and engagement such as: 75% of what people watch on Netflix come from recommendations [7] or that 70% of people's time spent on YouTube comes from the recommendation of "the Algorithm" [8]. Social medias and reading platforms apply these same concepts on their "feed" for those same reasons, user

retention and engagement. E-commerce, on the other hand, want to increase their revenue by recommending similar products, or products that the "same type of customer purchased" when a potential customer is navigating on their online shop. Amazon is a great example of this: 35% of the purchases on their online shopping come from recommendations [7]. There is even the use of RSs on online dating services [9], where the RS improve the experience of the user in the search of potential partner while at the same time increasing the monetization of the service.

Netflix is one of the companies that are references on this type of system. They have a variety of algorithms on their RS. And due to the high user base, they can test their results using A/B testing and feedback from the users [10]. In 2006 they launched a competition, called the Netflix Prize, where the objective was to improve the accuracy of the recommendations by 10%. The winner would win one million dollars. There were more than 2.000 teams and more than 40.000 submissions on their platform. The Netflix prize was an important event to the RSs in general because it increased the awareness of this technology, and its importance to business, worldwide.

2.1.3 Evaluation

There are several ways to evaluate the performance of a recommender system: normalized Discounted Cumulative Gain (nDCG) [11], Precision [12], Recall [13] when "looking" to the RS with a ranking perspective; Mean Absolute Error (MAE) [14] and Root Mean Squared Error (RMSE) [15] when "looking" to the RS with a rating perspective; A/B testing when you have a high client base that guarantee statistical relevance and others [16]. To evaluate the Neoway's recommender system we chose the Lift.

Lift is a ratio between probabilities. The probability of selecting one of the target population after the RS sorting versus the random chance (before the RS sorting). The lift, is usually expressed in terms of quantiles. [17].

Suppose your market set is given \mathcal{M} with size $|\mathcal{M}|=N$ and let the subset $\mathcal{L} \subset \mathcal{M}$, with $|\mathcal{L}|=n$, represent all companies within the market that would lead to a conversion. The set \mathcal{L} is of course unknown and the problem is to chose a strategy that recommends companies in \mathcal{M}

such that it is also belongs to \mathcal{L} . The probability for a random strategy is uniform across the population and is given by n/N . Suppose now that to each company X_i you associated a score S_i and ordered the companies from the highest to lowest score, i.e. $S_i > S_{i-1}$ for all i and started picking from S_N to S_1 . For any k , the lift is how much better this strategy is up to k relative to the random strategy, i.e.

$$\text{lift}_k = \frac{P(X_{i:N-k+1 \leq i \leq N} \in \mathcal{L})}{n/N}. \quad (2.1)$$

To illustrate further, let us use an example of a box with colored balls. Imagine that there is a rectangular box where its height and depth allows only a single ball to fit, but its width allows up to 100 balls. There are two types of balls: the **red** ones that represent our **target population** and the blue ones. 90 of the 100 balls are blue and the remaining 10 are red. Moreover, this box is divided into sectors where each sector is a decile, meaning that there are 10 sectors, each one with 10 balls. The sectors, also, have a priority, they from left to right. Figure 2.2 shows hows this box looks like.

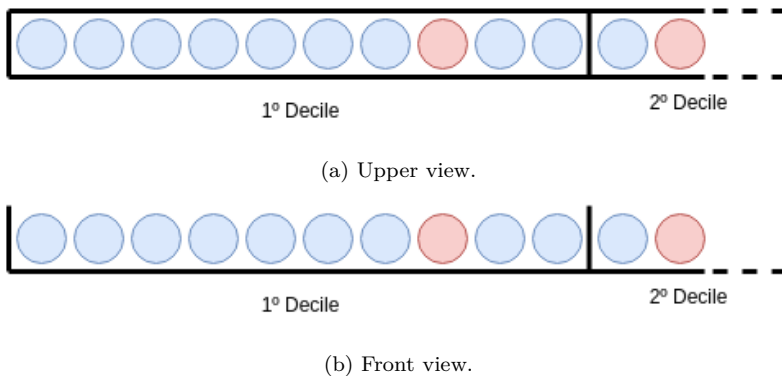


Figure 2.2: Rectangular box with colored balls.

Without seeing inside the box, the probability of retrieving a red ball is 10 out of 100, or 10%. This is our **random chance** and it is the same probability for each decile. Now lets consider that someone reordered the balls (representing the RS), trying to place the red ones in first deciles. The Figure 2.3 shows the box before and after the

reordering.

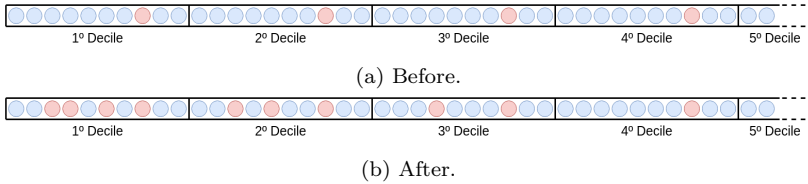


Figure 2.3: Reordering of the balls.

After the reordering, we recalculate the probabilities of retrieving a red ball for each decile. We can see on Figure 2.3b, that there are 4 red balls in the first decile, 3 in the second, 2 in the third and 1 in the fourth. With this information we have the following probabilities, respectively: 40%, 30%, 20%, 10% and 0% to the remaining deciles. Now the lift for the deciles can be calculated, for the first one is:

$$lift_{1^{\circ} decile} = \frac{0.4}{0.1} = 4 \quad (2.2)$$

Using the same approach, we calculate the lift for the others deciles, respectively: 3, 2, 1, and 0 to the remaining ones. Figure 2.4 shows a plot of the values of the lift for each decile.

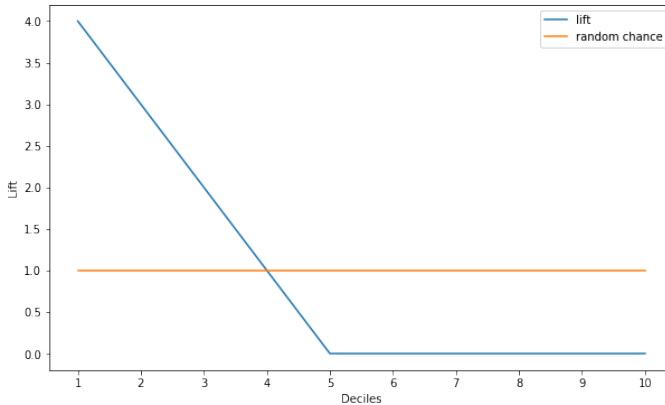


Figure 2.4: Lift plot for the reordered box with the random chance.

We can see that it is more likely to pick a red ball on the firsts

sectors of our box (deciles) than in the last ones. Actually, there is a zero probability to get a red ball after the fourth decile, since all balls are to the its left. And this is the expected behavior, since in an information retrieval system (IRS), we want all the relevant information on the top (in our case: on the left). Hence, only the first N -quantiles (in our case: $N = 10$) are considered and the rest is discarded. The designer of the system should decide how many quantiles are included and it also depends on how much information is retrieved and how much can be processed by the user.

With this example, another way to interpret the lift is: how many times better than the random chance it is to get relevant information in the first N -quantiles ordered by an IRS.

2.2 Clustering

Cluster Analysis is an unsupervised learning technique where its objective is to find a finite and discrete set of "hidden" structure in a dataset [18]. These structures or patterns can help better understand data or serve as a preprocessing step for algorithms. Clustering can also be a way to represent large amounts of information which is equivalent to compressing data.

This technique is applied in several areas [19]. For instance, in Engineering and Computer Science, clustering is used in speech recognition, information compression, noise removal. In biology, the applications are: genetics, taxonomy, microbiology and others; It is also used in Astronomy and Geography on classification of galaxies, planets for the former and classification of regions, areas, vegetation and land for the latter.

There are various approaches used to cluster a dataset The Connectivity-based method looks for the closeness of two objects, meaning that it forms the clusters based on their relative distance which can be computed in different ways like the complete (maximum distances) or simple (minimum distance) linkage methods. The Hierarchical Clustering is an example of algorithm that uses this approach [20]; On Centroid-based clustering the data is represent by a vector of centroids - we can think of them like center of mass - one for each cluster. The KMeans algorithm (and its variations), uses this approach [20]. The Distribution-

based approach focus on the statistics of the dataset. It groups together objects that have similar distributions. The algorithms to exemplify this are the Gaussian Mixture and its Bayesian variation. Another one is the Density-based clustering where the data is grouped on the high density areas. Points that are too far away are considered outliers and do not go to any cluster. Common algorithms are the DBSCAN and the OPTICS [20].

There is no single best algorithm on cluster analysis [21]. All of these mentioned, have pros and cons and are fit to different kind of problems. Some are suited to find the number of clusters but at the same time do not scale with a high volume of data (Hierarchical) [22], others are fast and simple but they have to know the number clusters in advance (KMeans). Hence, it is important to test different algorithms and their parameters to see if the analysis perform on a given problem.

2.3 Principal Component Analysis

Principal Component Analysis, or PCA for short, is an orthogonal linear transformation [23] where you "break down" a dataset - that can have correlation between the dimensions - into an independent set of variables called **Principal Components**. These components maintain the original data variance, or in practical terms, the amount of information. They are sorted in a descending fashion, where the first principal component (PC_1) has the most variance and PC_N has the least variance. N goes up to the number of the smallest dimension (rows or columns), the minimum between these two.

To understand how the PCA works in an intuitive way, Figure 2.5 illustrates the steps of the PCA algorithm on a dataset with two variables.

From the original data: (I) we calculate the mean point (center of mass), by getting the means of the x-axis and y-axis. Then we make this point our new origin (II). After that, a line that goes through the origin is fitted on the data on the direction of the highest variance (III). This direction is obtained by maximizing the sum of squared distances from the origin of each point's projection on the line (represented by the green Xs). The best fitted line is the Principal Component 1 (PC_1). From a projection orthogonal to PC_1 , the process repeats from (I) to

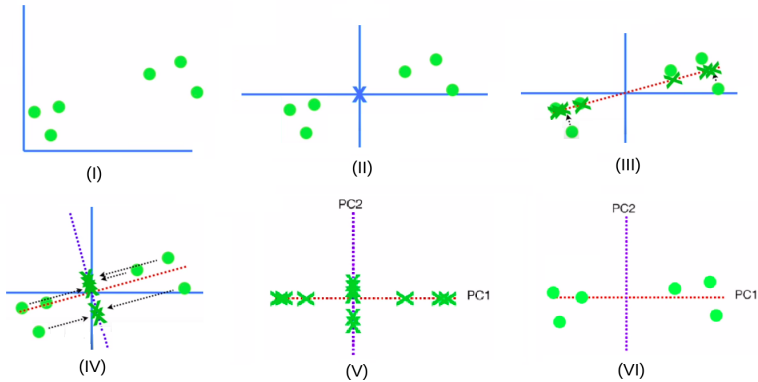


Figure 2.5: Steps of PCA for two variables. Source: Adapted from [2].

find PC_2 which is the second-highest-variance direction. However, since this example there are only two variables, there is only one possible direction (IV). With PC_1 and PC_2 calculated, when can make them our new axis (V) and use the points' projections (V) to find where they go on the PCA plot (VI). For a dataset with more variables the process is the same: data is centered; each PC_N line should point to the direction of N -highest variance and every PC_{N+1} must be orthogonal to PC_N .

PCA can be applied to any numerical dataset [24] (first it must be transformed or scaled). It can be a useful tool to analyse any multi-variable data. One use of this technique is the **PCA plot** which consists on getting the first two or three principal components (PCs) and plotting the transformed data. Since most of the variance (information) is on the first PCs, we can use them to look for patterns on the dataset with 2-D or 3-D plots. One of the patterns that can be found are clusters [25].

Another use of PCA is **dimensionality reduction**. Imagine that there is a dataset with more than 200 variables. Due to the curse of dimensionality [26] or even computing power a reduction on the number features is needed. If by applying PCA on this dataset we get, for instance, 99% of the variance on the first 20 PCs, the remaining PCs can be discarded. In practice, we are getting a 90% data reduction with only 1% of the information lost.

CHAPTER 3

Methodology

In this chapter it will be explained how the cluster analysis and the experiments were conducted. The first part of the chapter is dedicated to define some concepts related to the On Target product: the OT itself and its benchmark project. The second part is about the cluster analysis. And the third about the experiments.

3.1 About the Product

Before getting to the clustering and experiment procedures, it is important to define some concepts about the OT product. A high level explanation of the OT and how its benchmark was built will be described in this first part of the chapter.

3.1.1 On Target

The On Target is one of Neoway's offerings to Sales & Marketing, focused on the B2B public. Neoway's customers use the OT to look for other business that have the potential to become their customer. It achieves that through content-based recommendations. To create the recommendations the following inputs are needed:

- the **Portfolio**, a list of companies that represent a user’s customer base. It can range from a few dozens to dozens of thousands of companies;
- the **Market**, a list of companies that the user wants to target and which the OT will help recommend. It can range from dozens to tens of millions companies; and
- the **Features**, data used by the OT to find patterns and perform the recommendation.

And it outputs the sorted Market based on a score called **Similarity**. Figure 3.1 illustrates a block diagram of the OT.

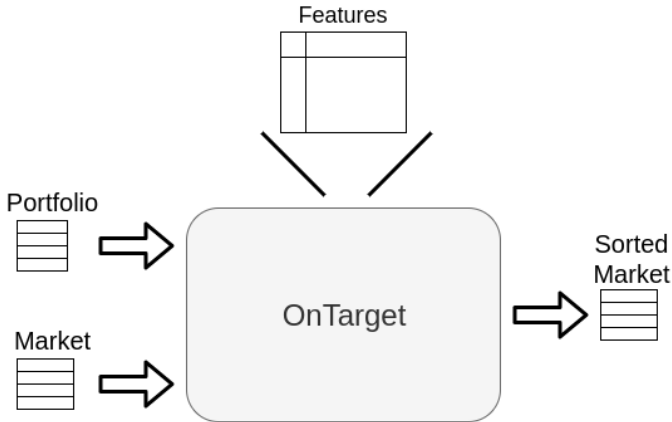


Figure 3.1: Block Diagram of the On Target.

3.1.2 Benchmark

As mentioned on Chapter 1, we developed a benchmark to guarantee that changes made to the algorithm will lead to improvements. The Benchmark is a project that compares different versions of the OT, by running these versions on almost thirty different business scenarios. They can be: one retail customer with a huge portfolio and huge market size; or a bank with a small portfolio and medium market size; or even a service provider with small portfolio and small market size.

A run of the OT for a single combination of Portfolio and Market is called a **Study**. An **Experiment** is the run of all of these scenarios for new versions of the OT and comparison to a base version of the algorithm. Every modification on the algorithm generates a new experiment. For instance, if the number of features is increased, an experiment (or more) is generated. If a parameter of the algorithm is changed, a new experiment is generated.

The Benchmark does some minor modification to the OT pipeline. It removes a random sample of companies from the portfolio (it can be 10%, 30% or 50%) and it places them on the market. The idea is to use this sample as a holdout set [27], which is a separated part of the data used for blind test. For consistency, it is expected that the companies in the holdout set will be scored highly. Note that the OT does not know a priori which companies are in the holdout set. Figure 3.2 shows these modifications to the OT.

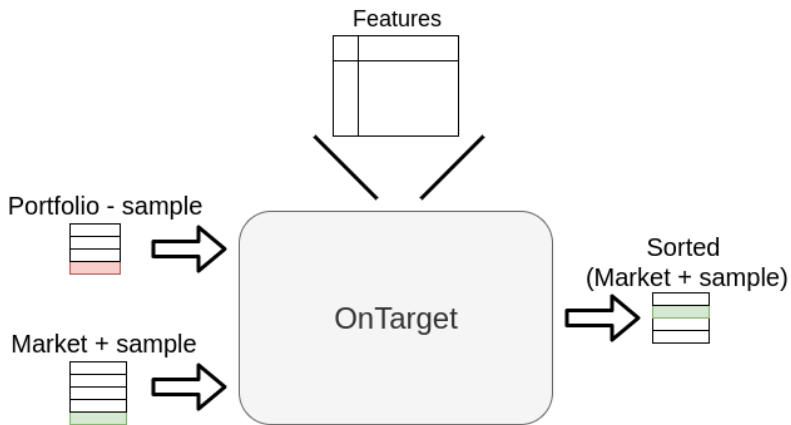


Figure 3.2: Modifications on the OT done by the Benchmark.

These changes to the pipeline are to generate the metrics to evaluate the experiments in two ways. The first one, is the performance with the **lift**, usually only the first decile. The second one is consistency, with the **similarity distribution** plot. On the former, the holdout set is used to calculate the probabilities after the sorting of the OT - similar to the red balls analogy discussed in Chapter 2.1.3. The latter, plots the distribution of the scores assigned to the market and to the

holdout set. As seen on the Introduction chapter of this work, Figure 1.2 shows an example of this plot, where the orange are the holdout set and the blue curve is the market.

3.2 Clustering

In this second part of the chapter it will be explained how the cluster analysis were conducted. How and what are the clustering strategies and clusters pairing. Also, a discussion about definition of number of clusters and the choice of the clustering algorithm.

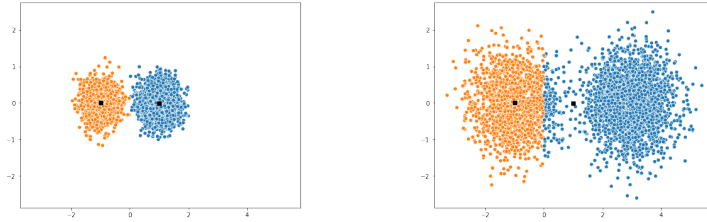
3.2.1 Clustering strategy and pairing

The first thing to tackle was the definition of the clustering strategy and pairing. The clustering strategy dictates how the clustering algorithm will be applied to the data, and the cluster pairing, determines how the clusters from the portfolio will be paired with the clusters from the market.

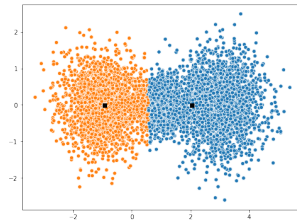
There are two possible clustering strategies: training the cluster algorithm on the portfolio and applying it to the market, which will be called Train on Portfolio (ToP), and training and applying to all the data, which will be called Train on All (ToA). Let us take the example of the KMeans algorithm using these two strategies applied on a fake study dataset, which is illustrated on Figure 3.3. Assuming that all sub plots are on the same scale, in Figure 3.3a we see the portfolio clustered with its two centroids, in Figure 3.3b we can see the KMeans "predicting"¹ the clusters of all data with the centroids learned in Figure 3.3a. This is the ToP. The ToA is represented in Figure 3.3c where the KMeans trains and predicts to all data. We can see that the centroids are on different positions, as a result some companies are allocated on a different cluster relative to Figure 3.3b.

For the pairing there are also two possible ways: each cluster in the portfolio will be matched with its cluster in the market, let us call it One vs One (OvO), and each cluster in the portfolio will run with the whole market, let us call it One vs All (OvA). For instance, if we have a study with three clusters in the portfolio each with 50 companies

¹this is a fairly common name for the inference step of an algorithm in scikit-learn implementation [28]



(a) KMeans learning the centroids from portfolio data. (b) KMeans "prediction" to all data, with centroids learned from (a).



(c) KMeans learning and predicting to all data.

Figure 3.3: Fake study data set that shows how both clustering strategies work.

each, and 150 companies for each cluster in the market, in OvO the OT will run three times with a portfolio size of 50 and market size of 150; for OvA the OT will run three times with a portfolio size of 50 and a market size of 450 (whole market for each cluster). Figure 3.4 shows a block diagram of this scenario.

Through a business view, it makes more sense to adopt the **ToP** since we are interested on the recommendations based on the user profile, in other words, the user's portfolio, regardless of the market data.

On cluster pairing ToA the market is replicated for each cluster, leading each company to be scored N times (N being the number of clusters). Hence, there could be cases where one company has a high score in a cluster while, at the same time, have a low score in other one. To not deal with these cases the **OvO** was chosen for this work.

3.2.2 Number of clusters

The second aspect of the cluster analysis was the number of clusters for each study, since most of the cluster algorithms need this information

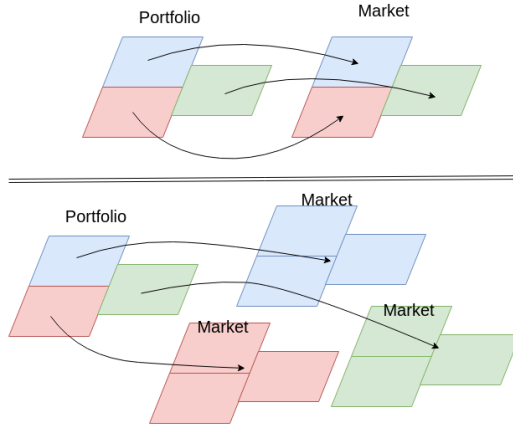


Figure 3.4: Illustration of the cluster pairing. On the top is the OvO and on the bottom the OvA.

upfront. Even though there are some research on evaluating the number of clusters automatically [29], there are no production-ready solutions. There are, however, methods like Elbow method and Silhouette [30] that help with the determination of the number of clusters. Considering that the objective is to analyse the impact of the clustering on the RS and not the cluster analysis itself and that there are only up to thirty studies, it was decided to set the number of the clusters **manually**.

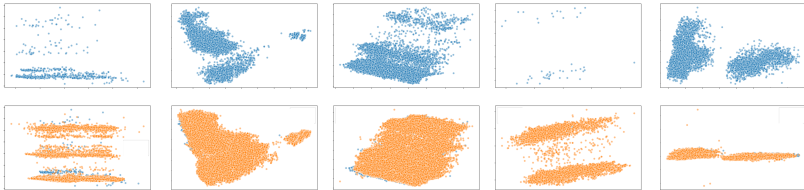
To visualize the clusters on the studies, it is necessary to plot the data. But OT uses too many variables on production to analyse all of it at once. Hence, it was applied PCA transformation on the studies and used the first two principal components to plot the transformed data and to see the patterns in it.

Before applying the PCA, a preprocessing step was needed. Categorical features were converted to numeric using One Hot Encoding and all features were scaled using the Z-score Normalization².

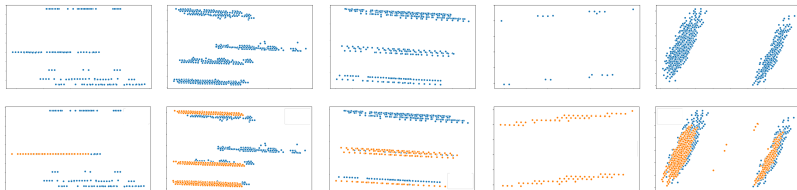
Figure 3.5 shows the results of the PCA plot for some studies using all features that OT uses on production environment (3.5a), and PCA plot using only firmographics data for the same studies (3.5b). Each study is a column, where the first row is the plot of the PCA for the

²all of the data transformations functions used on this theses are from the Scikit-learn Python library [28]

portfolio data, and the second row is for both, using different colors to distinguish portfolio from market (blue is the market and orange the portfolio).



(a) PCA plot using all features



(b) PCA plot using firmographics data

Figure 3.5: PCA plot for some studies. For each study (a column) the first row is only the portfolio data, second row both market and portfolio (market in orange and portfolio in blue).

We can see that it is much more difficult to visualise the clusters on the all-features PCA plot. There are some examples where you see a clear boundary on the data, but if we look to the firmographics plot these boundaries are much more defined. Moreover, if you take into account the business perspective, it makes sense to group the companies by their firmographics. For example, a study with one cluster with big companies from the state of São Paulo or another one with small companies from the state of Santa Catarina.

Using the firmographics PCA plot, each study was assigned with a number of clusters manually. For instance, in Figure 3.5b, from left to right, the number of clusters in the portfolio assigned are, respectively: 3, 4, 3, 2, 2.

3.2.3 Cluster algorithm

The last aspect of the cluster analysis was the choice of the algorithm. Six clusters algorithms were applied to all the studies: KMeans, Gaussian Mixture, Bayesian Gaussian Mixture, Agglomerative Clustering, Spectral Clustering, DBSCAN³. All of them were computed to the firmographics data transformed by the PCA with default parameters.

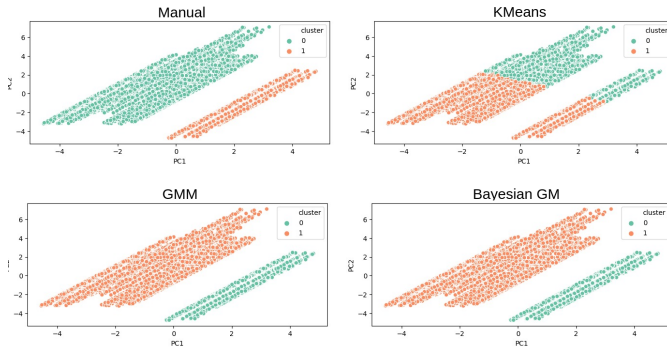
The last three did not run on some studies that have a market size in an order of magnitude of dozen of thousands companies, due to memory error⁴. Because of their high memory complexity [22], [31], [32], they were discarded.

The studies were also clustered manually in order to be used as a benchmark for the algorithms. Since most of the PCA plots present a regression characteristic, these boundaries were lines equations. Figure 3.6 shows the results for some of the studies using "Manual Clustering", KMeans, Gaussian Mixture Model (GMM) and Bayesian Gaussian Mixture (Bayesian GM).

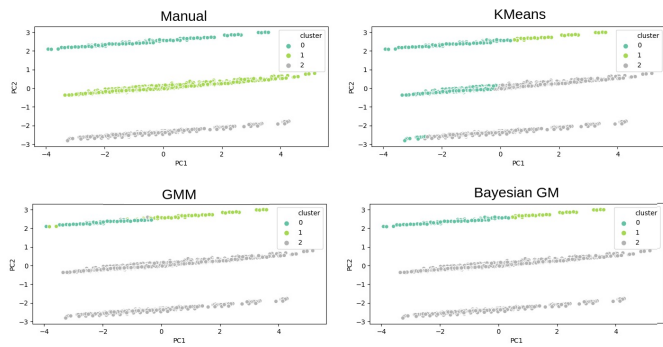
The algorithms did not get the expected shape of the data in some of the studies. The Bayesian and GMM had better results relative to the KMeans, 3.6a illustrates this example. But they did not assign the clusters as expected to other studies, as seen in 3.6b and 3.6c. Due to these reasons and the same argument explained in section 3.2.2 - we are interested on the results on the RS and not in the cluster analysis itself - it was decided by the team to start to the experiments using the "Manual Clustering".

³all of these implementations came from the scikit-learn python library [28]

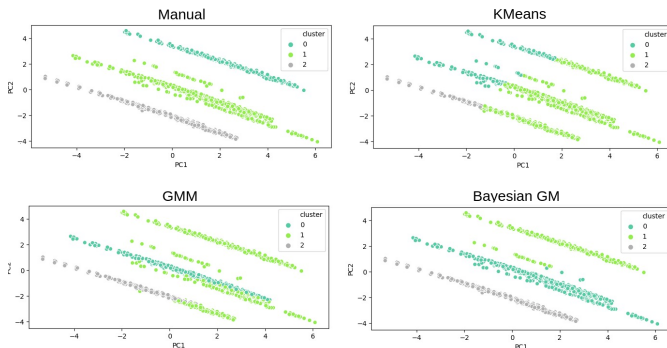
⁴all the algorithms were running on a machine with 250 GB of RAM



(a)



(b)



(c)

Figure 3.6: Results of the clustering on the first two principal components of three studies.

3.3 Experiments

In this last part of the chapter, it will be discussed about the experiments that were developed to test the clustering in the OT. After the analysis of the clusters, two experiments were performed, as detailed next: Run per Cluster (RpC) and Cluster as Feature (CaF).

3.3.1 Run per Cluster

The experiment RpC consists on using the clustering strategy ToP and cluster pairing OvO. A cluster in a portfolio will be matched with its pair on the market, and each cluster, in a study, will generated a run of the OT. After the runs, all of the outputs will be joined into a single one. So, for the user, the interface still the same. But, internally, OT will split the study in N^5 smaller studies and then will aggregate the outputs. These modifications can be better understood when looking to the diagram on Figure 3.7.

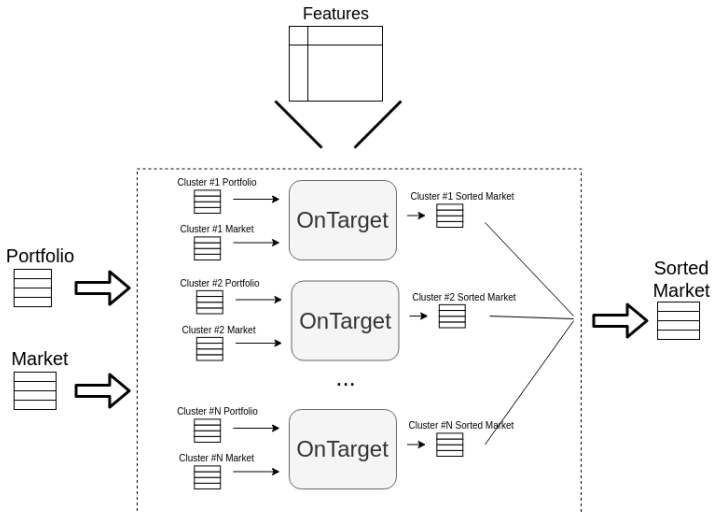


Figure 3.7: Modifications on the OT for the experiment RpC.

To clarify this idea of the Experiment RpC, let us take the example of Figure 3.3. Since there are two clusters, OT will match the dark blue

⁵ N is the number of clusters

cluster on the portfolio with its pair on the market, generating one run. Another run will be generated by using the same analogy for the light blue cluster. Then the OT will aggregate both outputs and return as only one sorted market.

Another important modification made to the OT pipeline due to the RpC was the definition of a minimum size to a study. There could be cases where, in a portfolio, a cluster has a small number of companies that is not enough for the RS algorithm to create the score. If a cluster does not have the minimum data in a set to run, all of the data is discarded.

3.3.2 Cluster as Feature

The second experiment consists in using the information of the clusters as an extra column in the features table used by the OT. It will not use any of the cluster pairing strategies (it uses the clustering strategy ToP, though). The experiment CaF is much more simple than RpC: there is no split on the study; it is not necessary an aggregation of the output; and it is not needed to define a minimum number to a study to run. The information of the clusters is joined with the features table that feed the RS algorithm. Figure 3.8 shows its simple modification.

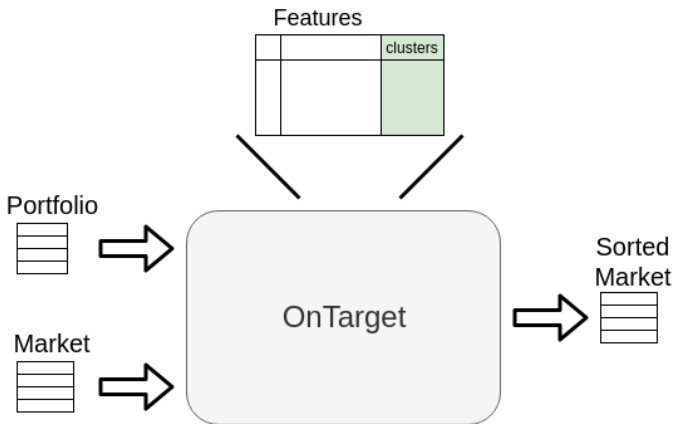


Figure 3.8: Modifications on the OT for the experiment CaF.

CHAPTER 4

Experimental results

In this chapter we will present the results of the experiments RpC and CaF, along with a discussion of the impact on the lift and similarity distributions of these experiments relative to the OT without the clustering.

4.1 Experiment Run per Cluster

As seen on 3.3.1 this experiment was the one that creates sub-runs of the OT for each cluster in the portfolio. Also, it had to be created some heuristics to deal with clusters that did not have enough data to run. Using the "manual clustering", as discussed on 3.2.3 the summary of the lift gain (how much the lift on the first decile increased or decreased relative to the OT without clustering) is presented on Table 4.1. Figure 4.1 shows another perspective for this data through a histogram, excluding the outliers studies.

The cells are colored to better understand the gains. The dark colors (green and red) represent a considerable positive or negative gain (more than 5%), the light colors a slight positive or negative gain (less than 5%). The white, represent that the lift did not change. Finally, the

orange ones are the outliers which were defined using the interquartile range rule [33].

Study	Lift Gain (%)	Study	Lift Gain (%)
1	-12,09	15	-13,26
2	16,67	16	-20,00
3	0,89	17	-4,28
4	0,00	18	-13,06
5	-0,14	19	-15,42
6	-17,28	20	0,57
7	-12,44	21	-33,95
8	-5,17	22	0,00
9	-27,78	23	-28,16
10	6,25	24	200,01
11	-17,75	25	-12,77
12	-1,01	26	-29,35
13	-6,93	27	-0,08
14	-1,16		

Table 4.1: Summary of the first-decile lift gains for Experiment RpC

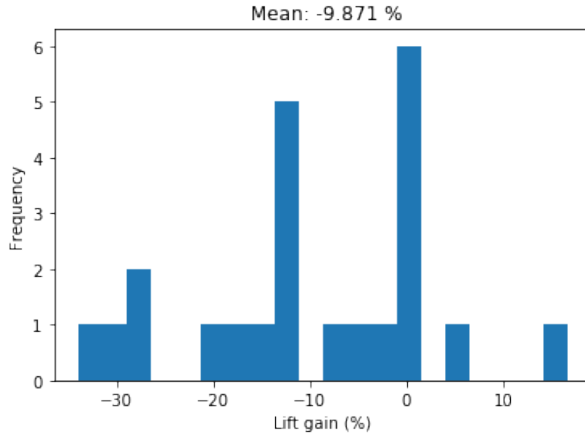


Figure 4.1: Histogram plot of the studies' lift gain for experiment RpC.

There are also some studies that are bolded. They represent the studies that brought up the hypothesis of this work, as discussed at the

Introduction, they had two or more high density areas on the similarity distribution plot, meaning that, they can have more than one profile in their portfolio.

We can see that only five studies had positive impact on the lift, and one of them is an outlier. The majority of the studies had a negative impact, six of them were slightly negative and fourteen worsened the lift considerably. It is important to notice that two studies did not present any changes on the lift with the clustering. The overall mean lift gain for RpC is -9.871% .

4.2 Experiment Cluster as Feature

Now for the second experiment, which was the simple use of the clustering information as another feature. Using the same color scheme as Table 4.1, Table 4.2 shows the summary of the lift gains for the experiment CaF. Figure 4.2 this data in a histogram (without outliers studies).

Study	Lift Gain (%)	Study	Lift Gain (%)
1	4,40	15	0,46
2	-8,33	16	-6,58
3	23,65	17	-46,08
4	0,00	18	-1,87
5	1,49	19	3,58
6	-28,33	20	-11,28
7	12,50	21	-2,64
8	16,84	22	1,23
9	12,80	23	-0,14
10	18,59	24	300,02
11	-7,84	25	4,90
12	-0,15	26	-0,77
13	0,68	27	1,21
14	7,30		

Table 4.2: Summary of the first-decile lift gains for Experiment CaF

We can see that, differently from RpC, the majority of the studies had positive gain (16 studies, to be exactly). Only four studies were considerably worse on the lift gain and five slightly worse, a total of nine

studies. Only one study did not present any change to the lift and, this time, four were outliers (two positives and two negatives). The mean lift gain for CaF is 2.016%.

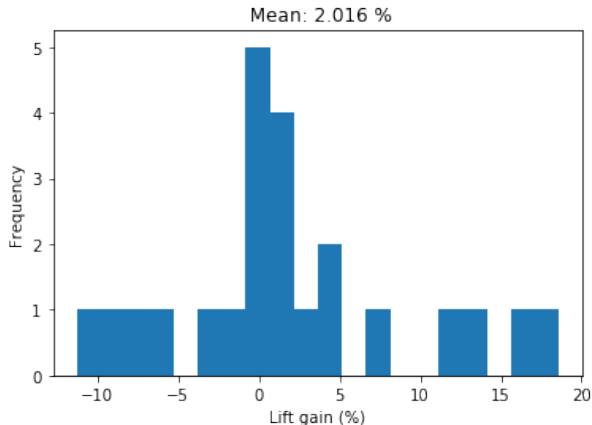


Figure 4.2: Histogram plot of the studies' lift gain for experiment CaF.

4.3 Similarity distributions

Now let us look at some of the similarity distributions plots for both experiments. These plots will have three curves: the market similarity in green, the holdout set in orange, and the portfolio in blue. Each topic of this section will address a group represented by the colors in the lift gain summary tables.

4.3.1 Studies with no change to the lift

In both experiments there were studies that did not lead to any change to the lift. Study 4 appeared on both. Figure 4.3 shows the similarity distribution plot for experiment RpC, and 4.4 for experiment CaF. In both figures, the first row is the OT run without the clustering, and the second one is with the clustering.

We can see that in experiment RpC (4.3) the similarity distributions with and without clustering are virtually the same. This is better understood when you look at the number of clusters in its portfolio.

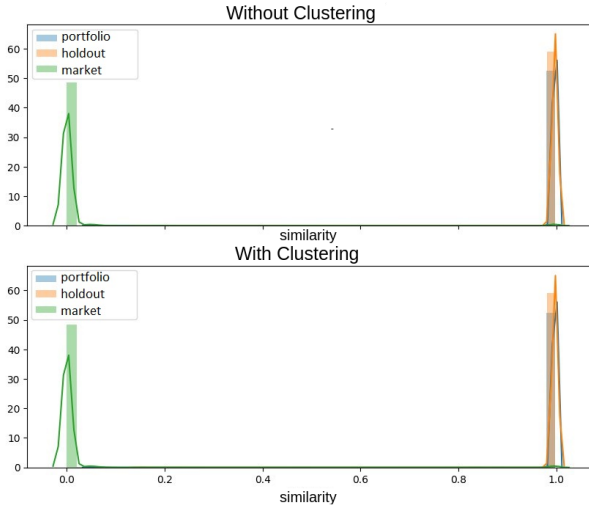


Figure 4.3: Similarity distribution plot for Study 4 in experiment RpC.

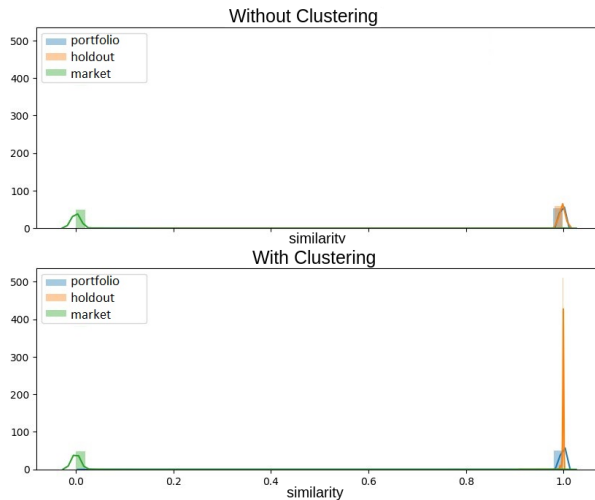


Figure 4.4: Similarity distribution plot for Study 4 for experiment CaF.

Figure 4.5 shows its PCA plot. There is only a single cluster on the portfolio, consequentially, the result of this cluster is the same regardless of using the clustering approach.

In CaF (4.4), however, there is a difference on the distributions be-

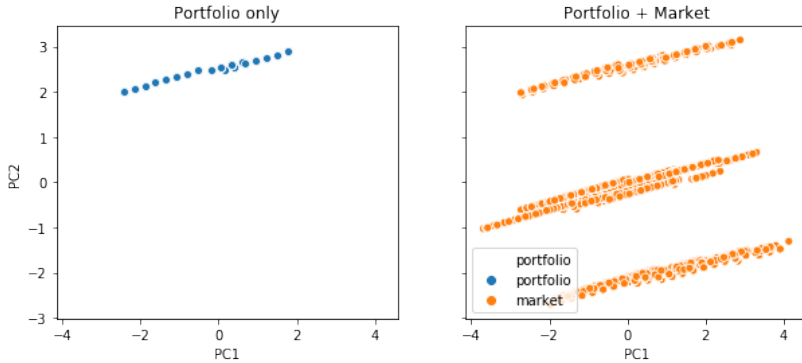


Figure 4.5: PCA plot for Study 4. On the left there is just the portfolio, on the right both portfolio and market.

tween the with and without the clustering. Basically, the holdout set changed its distribution. In this experiment the OT scored these companies with really close scores, in other words, their standard deviation decreased. But, even though the scores of the companies (and possibly the ordering) of the holdout set changed, the lift kept the same because of how it is calculated. The same number of companies (of the holdout set) on the first decile occurred in the run without the clustering and in with the clustering, thus the value of the lift is the same for both runs.

Another study that had a zero lift gain was Study 22, in the RpC experiment. Figure 4.6 shows the similarity distribution plots for the clusters' runs of this study.

Although this study has three clusters in the portfolio, only Cluster 1 has the minimum data to a valid run. Cluster 0 and Cluster 2 did not have companies on the Market, so the sixteen companies of the former and thirteen companies of the latter were discarded. Since they represent less than 0.001% of the overall data of the study, the behavior is practically the same as Study 4, thus, for the same reason, the lift is the same for both runs (with and without clustering).

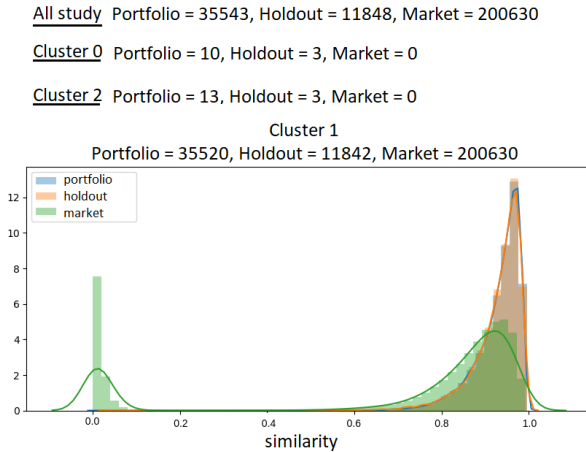


Figure 4.6: Similarity distribution plot for the clusters' runs of Study 22 on experiment RpC.

4.3.2 Studies with marginal increase or decrease on the lift

Both experiments had studies that changed the lift in a minor way, positively and negatively. These studies improved or lowered the lift up to approximately 5%. They are presented in the light green and light red colors in both tables previously mentioned. Experiment RpC had two positives and five negatives. CaF had seven and five, respectively.

All of these studies had the same prevalent behavior: the overall distributions of the three sets (portfolio, holdout, and market) before and after the clustering were almost the same. There is a small difference to the holdout distribution. Figures 4.7 and 4.8 illustrate, respectively, examples of a small increase and decrease on the lift. In the former, the similarity distributions plot for Study 5 in experiment CaF, and in the latter the same plot for Study 12 for experiment RpC.

We can notice that the peak near similarity 1.0 of the holdout set goes from approximately 37.0 to beyond 40.0 in Figure 4.7. The opposite happens in Figure 4.8, the peak goes from almost 25.0 to 20.0.

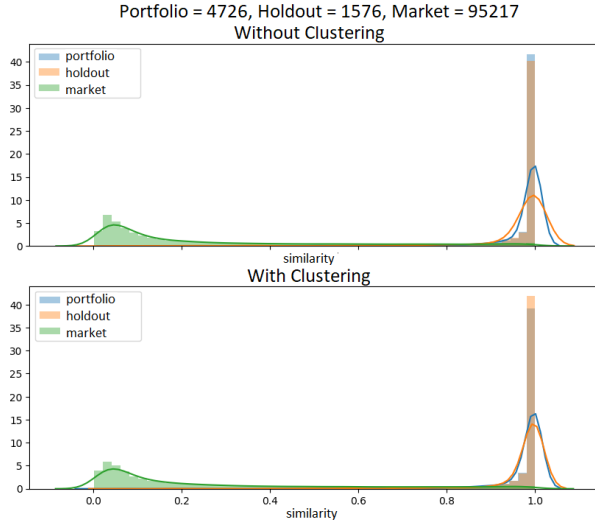


Figure 4.7: Similarity distribution plot for Study 5 on experiment CaF. An example of marginal increase on the lift.

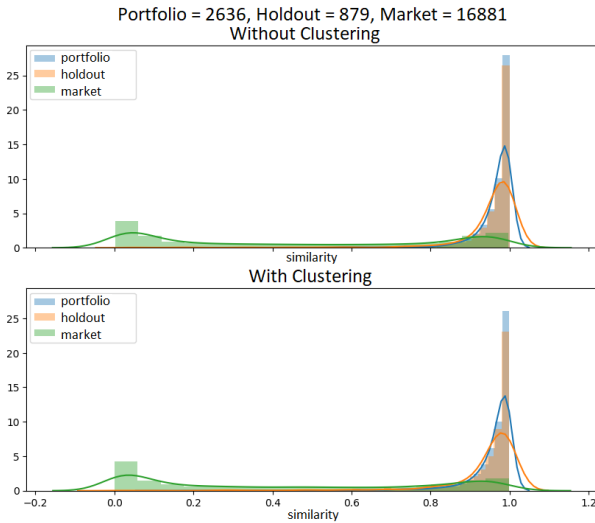


Figure 4.8: Similarity distribution plot for Study 12 on experiment RpC. An example of marginal decrease on the lift.

4.3.3 Studies with considerable increase or decrease to the lift

Most of the studies on both experiments had more than 5% variation to the lift (positive and negative). In RpC, fourteen of them were negative and only Study 2 was positive. However, in experiment CaF, five were positive and four were negative.

The behavior of the similarity distributions with the clustering follows the same idea as section 4.3.2, but with more exacerbate results. Figure 4.9 shows the similarity distributions with and without clustering for Study 8 in experiment CaF, which is an example of a considerable increase on lift. In contrast, Figure 4.10 displays the same plot for Study 9 in experiment RpC, an example of considerable decrease on the lift.

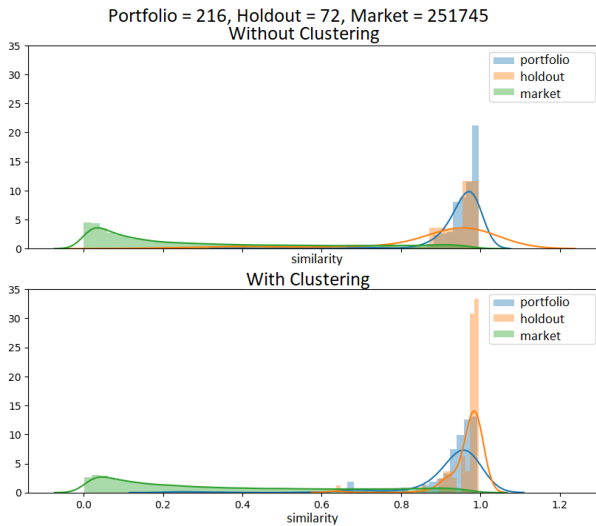


Figure 4.9: Similarity distribution plot for Study 8 on experiment CaF. An example of considerable increase on the lift.

Now the difference between the distributions, in these cases, is easier to spot. In Figure 4.9 we can see that the peak of the holdout set rose from 10.0 to more than 30.0. Consequentially, the curve became more thin, meaning that its variance decreased. The contrary to this, is showed on Figure 4.10. The peak of the holdout set distribution went from 20.0 to less than 10.0. Also, the portfolio distribution changed.

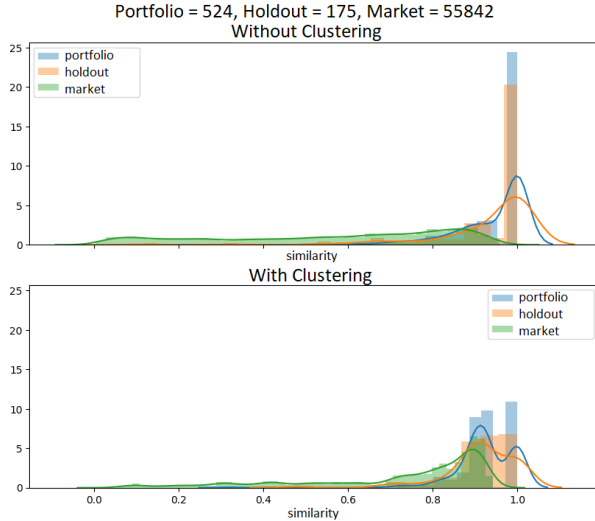


Figure 4.10: Similarity distribution plot for Study 9 on experiment RpC. An example of considerable decrease on the lift.

Its single peak decreased by 15 units, and it was splitted in two peaks, one near 0.9 similarity and the other near similarity 1.0. Moreover, the distribution of the market skewed to the right on this run, creating a high density area near 0.9 similarity.

4.3.4 Outliers studies

Study 24 was clearly the outlier of all studies. It got more than 200% increase on the lift in both experiments. In experiment CaF more three studies were considered outliers due to the interquartile range rule - the lower and upper bound in this experiment were -14.77% and 18.62% , respectively. These studies had similar shift on the distributions as the ones seen in 4.3.3. Hence, just Study 24 is presented here. Figure 4.11 exhibits the similarity distribution plot of this study in CaF, and Figure 4.12 its lift plots, which contains the lifts of all deciles of the runs with and without the clustering.

The main aspect of this study is presented in Figure 4.11. The Figure shows unusual distributions with multiple modes (without clustering), the highest one located at similarity 0.8. The OT failed to

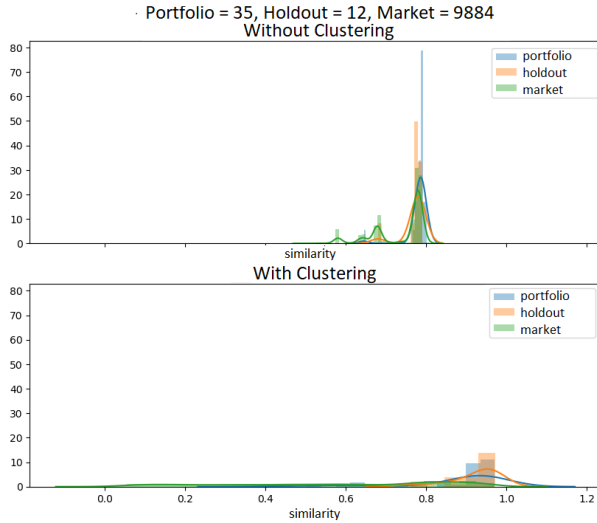


Figure 4.11: Similarity distribution plot for Study 24 on experiment CaF.

classify the portfolio of this study close similarity 1.0. Moreover, the number of companies for each of the sets of the study is unbalanced. There are only 35 companies in the portfolio and twelve in the holdout set against a market with two orders of magnitude higher. This discrepancy can be one of the issues that led this study to have odd similarity distributions.

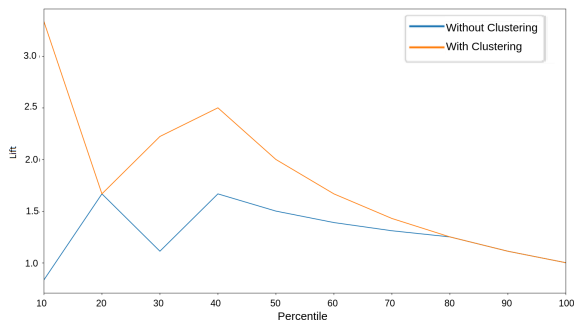


Figure 4.12: Lift plot for Study 24 on experiment CaF.

This issue is reinforced when we look at Figure 4.12. The expected

shape of a lift curve is in the form of a decreasing exponential, in other words, the lift of the decile on the left is greater than the one in the right, but this is not what happens on both runs of the experiment for this study. In the run with the clustering, the lift of the second decile is lower than the fourth one. On top of that, the lift of the first decile in the run without the clustering is lower than 1.0.

With the clustering, the distributions skewed to the right, with the holdout set being the closest to similarity 1.0. This explains why the lift of the first decile increased over 200% in the run with the clustering.

4.3.5 Studies that had multiple profiles in its portfolio

The last group of studies to be analyzed are the bolded ones in both tables (4.1 and 4.2). These are the studies that sparked the idea of clustering the portfolio before running the OT. However, most of them did not had the expected behavior. Let us use one example that illustrate this result. Figure 4.13 displays the similarity distributions plots for Study 25 for the runs in experiment CaF.

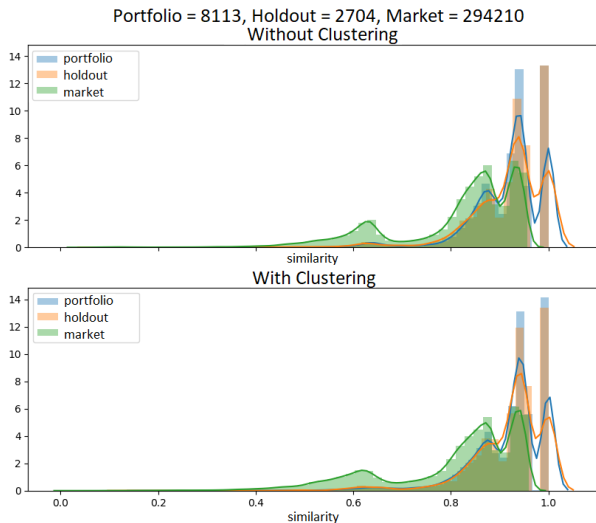


Figure 4.13: Similarity distribution plot for Study 25 on experiment CaF. An example of study with multiple high density areas in the portfolio.

Study 25 is a study with a positive lift gain, that is, the performance

of the study improved. However, looking at Figure 4.13 we notice that the three modes of the portfolio distributions in the run without the clustering are preserved in the run with the clustering. We expected that each one of these peaks would shift to the right, close to similarity 1.0.

If we take a look at each clusters' similarity distributions plot we can see more clues for this behavior. Figure 4.14 shows these plots along with the clusters' sets sizes of the runs that did not had enough data. Figure 4.15 shows its PCA plot with the portfolio and the market.

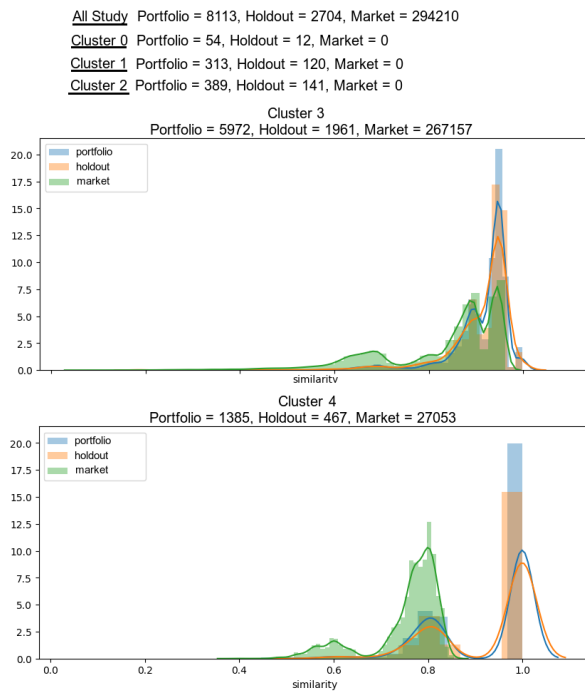


Figure 4.14: Clusters' similarity distributions plots for Study 25 in experiment RpC.

There are five cluster in the portfolio, but only two (Cluster 3 and 4) had a matching pair with the market. In these cases we see that they did not present a "one peak area" in the portfolio. Cluster 4 has two peaks: one near similarity 1.0 and the other near 0.8, and Cluster 3 two close peaks near similarity 0.9. The other clusters with not enough data

$(0, 1, 2)$, the portfolio and holdout sets correspond to approximately 10% of the study portfolio size. This is a small portion of the data and it only affects RpC, after all, in CaF there is no cluster pairing, since this information is used as a feature.

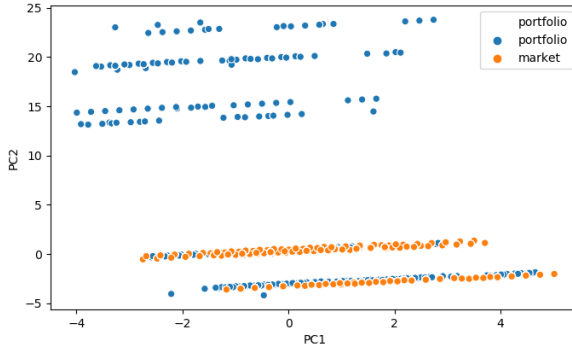


Figure 4.15: PCA plot with portfolio and market data for Study 25.

The outcome of these studies demonstrate that, perhaps, **our initial hypothesis is not on point**. There still other analysis to be done, such as, analyzing the companies from the similarity perspective, for instance, seeing what the companies in the high density area near similarity 0.9 of some study have in common. These new perspectives emerged as the work was being developed. However, because of the time constraint, they could not be prioritized in this proof of concept.

4.3.6 Other relevant studies

There is another study that is not in one of the specifics lift-gain groups aforementioned, but its analysis of the similarity distribution is worth to be considered. Figure 4.16 shows the similarity distribution plots for Study 7 in the experiment CaF.

This study had a lift gain of 12.5%, so its performance improved, thus the expected impact in the distributions is the holdout set increase its peak near similarity 1.0 or to shift to this region when the distribution is located more on the left of the axis. Nonetheless, the opposite happens in this study. In Figure 4.16, we see that the distributions keep its shape but shifted from similarity 1.0 to around 0.85. But since the

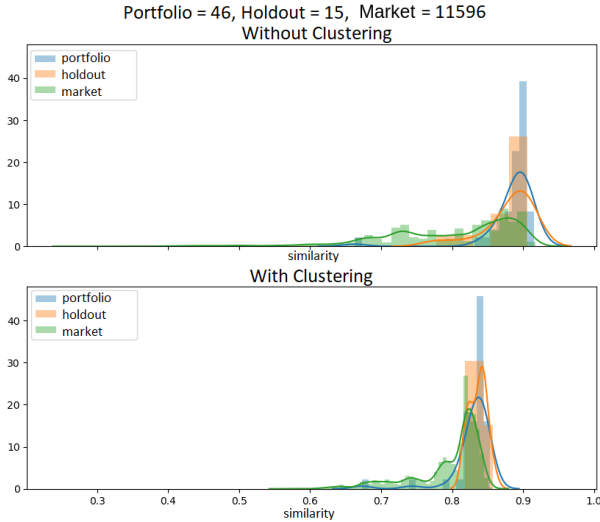


Figure 4.16: Similarity distribution plot for Study 7 on experiment CaF. Lift increased but the curves skewed to the left.

holdout distribution stayed more in the right, in the first decile there are more companies from the holdout set, hence the increase on the lift

The cause of this shift was not investigated. But the main point of mentioning this study is to raise the awareness of the OT's team in the choice of metrics. This example indicate that only the lift is not enough to verify the output of the OT. As seen throughout section 4.3, the lift and the similarity distribution plot are correlated, specially the distribution of the holdout set. The lift is a metric of performance and the similarity distribution plot shows the consistency of the study. The shift of Study 7 was not critical, but if the similarity that the distributions went was 0.3 instead of the 0.85, the recommendations would not make sense for the OT's user due to its lack of consistency. That is why a new metric, that take into account consistency and performance, should be included in the overall benchmark of the studies.

4.4 Experiment CaF with other clustering algorithms

The poor performance of the studies in experiment RpC, which as the mean lift gain of -9.871% , made the team of the OT to not proceed

the testing of the clustering with this experiment. Although CaF had a positive mean lift gain of +2.016%, it was conducted with the "manual clustering". Thus, it was determined to oversee the results of this experiment with the other clustering algorithms previously presented in Figure 3.6. Figure 4.17 shows the histogram of the lift gains of the re-runs for CaF with KMeans (4.17a), Gaussian Mixture (4.17b), and Bayesian Gaussian Mixture (4.17c).

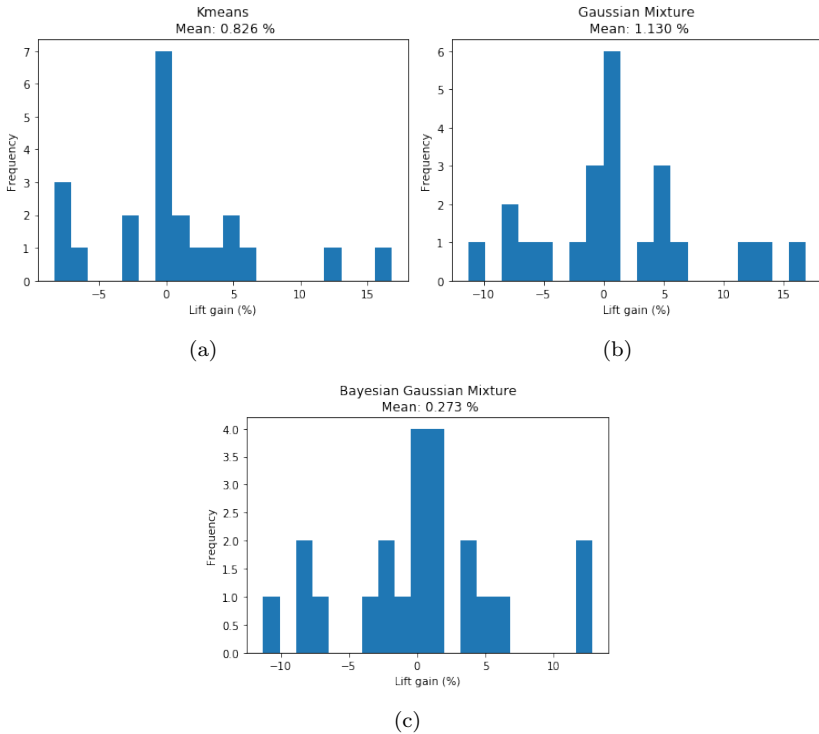


Figure 4.17: Lift gain histogram without outliers of other clustering algorithm runs for experiment CaF

We notice that none of them reached the lift gain of the "manual clustering". The closest one was the Gaussian Mixture with 1.13%. The other two resided below 1% (0.8 for KMeans and approximately 0.3 for Bayesian Gaussian Mixture). The outcome of these results consolidate the strategy of cluster the studies manually, meaning that the groups in the PCA plots make sense, at least in the performance perspective

of the OT.

Despite of this confirmation on the strategy adopted by the team, this is not feasible for the OT, since it is impossible to the system run in production with human interference during the leads scoring. The clustering done manually could be use as a success criteria for other clustering algorithms, but that would turn the type of problem to a classification, due to the presence of the labels.

Another alternative to improve the performance in experiment CaF without the manual clustering, is to tweak the hyper-parameters of the Gaussian Mixture and see if it the lift gain approaches the achieved mark of 2%. For instance, in the Scikit-learn Python package, this algorithm has up to seven different hyper parameters that go from initialization, weights settings up to convergence settings [28].

CHAPTER 5

Conclusion

The idea of this work emerged with the benchmark of the On Target (OT). Through the similarity distribution plots, the team noticed that some studies had multiple modes in the holdout set distributions, leading to the hypothesis of multiple profiles in the portfolios. Based on this scenario we used a proof of concept to validate a pre-clustering strategy, where we cluster leads by using firmographics data in order to improve the performance of the OT.

We started this thesis by discussing the context of recommender systems and their relevance for business today. Next, we introduced the concepts of clustering and principal component analysis, which were applied in the upcoming chapters.

After that, we introduced some terminology of the On Target and its benchmark, along with the explanation of how the leads are scored through block diagrams. Then, we tackled the clustering procedures. Concerning the choice of the algorithm and number of clusters we decided to adopt a "manual" approach. The clustering strategy and pairing chosen were, respectively, "Train on Portfolio" (ToP) and "One versus One" (OvO). From both of these, two experiments were designed to test the clustering methods: "Run per Cluster" (RpC) and "Cluster

as Feature" (CaF). We analysed their results seeing that the former had a negative gain of approximately -10% and the latter had a slight improvement of 2% . We also, further investigate their results by examining the similarity distribution plots of the studies of the groups former based on the lift gain tables. In this examination we investigated the correlation between the lift and the similarity plot, and other studies' cases that did not yield performance improvement as expected. Finally, we repeated experiment CaF with other clustering algorithms and verified that none of them surpassed the improvement of the "manual" approach.

Outlined all the steps of this work and its outcomes we conclude that it is not worth to pursue, for now, with the idea of clustering with firmographics data before running the OT. The overall improvement of 2% on the lift is not enough to prioritize the further development of this work on the On Target Product Roadmap at this moment.

5.1 Takeaways

In spite of the unsatisfactory result in the performance attained, this work brought some insights for the team.

Focus on the value first. Clustering the data manually, saved a lot of time and resources from the company. It would take too much from the team to develop a solution that would cluster the data in an automated manner. The value is the impact on the performance of the recommender system, clustering is a way to achieve that. The use of a "Proof of Concept" was an opportunity to quickly test this idea before putting the necessary effort to put it on production;

Better understanding about the problem. We noticed that some of the studies have distinct clusters in their portfolios while others were the case of a single cluster with outlier companies. Moreover, there were studies with a same magnitude of portfolio and market size while others had an unbalanced configuration. This variability of configurations of a study adds a layer of complexity on the OT. It is not a simple model that fulfils the requirement of consistent recommendations for all of the verticals and branches that Neoway addresses. Knowing more about the data in this problem empowers the team to deliver better solutions;

Choice of metrics. The benchmark already brought up the discussion of new metrics for the OT. This work, reinforced that when we discussed in Section 4.3.6 about the study that had a positive lift gain while the sets in the similarity distribution shifted to lower values. This result shows that the lift by itself is not sufficient to evaluate a study.

Future work

To become a production-ready solution, the suggested future work is:

- to further develop the clustering module. The manual solutions still need to be automated. One possible way is to test the Gaussian Mixture with the tuning of its hyper parameters. Since, as seen in Chapter 4.4, it got the best lift gain from the three algorithms;
- to add more features or to change the context of them. The addition of more features or testing different feature contexts is a valid experiment for this work.
- to include suitable metrics. We discussed the limits of the lift on checking the consistency of the study in Chapter 4.3.6. To better evaluate the studies a new metric that measures the consistency of the output from the OT is necessary;
- to get user feedback. To be able to really understand if the proposal solution improved the OT's recommendations, it is necessary to get feedback from the user.

Bibliography

- [1] Collaborative filtering: an ace up cartamundi's sleeve - delaware / a.i. <https://www.delaware.ai/customer-stories/collaborative-filtering-an-ace-up-cartamundis-sleeve>. (Accessed on 2019-10-30).
- [2] StatQuest with Josh Starmer. "statquest: Principal component analysis (pca), step-by-step". <https://www.youtube.com/watch?v=FgakZw6K1QQ>. (Accessed on 2019-11-11).
- [3] A. Weinstein. *Handbook of Market Segmentation: Strategic Targeting for Business and Technology Firms, Third Edition*. Taylor & Francis, 2013.
- [4] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.
- [5] (2015) how many products does amazon sell? - export x. <https://export-x.com/2015/12/11/how-many-products-does-amazon-sell-2015/>. (Accessed on 2019-11-15).
- [6] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-

- art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.
- [7] How retailers can keep up with consumers | mckinsey. <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers>. (Accessed on 2019-10-30).
- [8] Ces 2018: Youtube’s ai recommendations drive 70 percent of viewing - cnet. <https://www.cnet.com/news/youtube-ces-2018-neal-mohan/>. (Accessed on 2019-10-30).
- [9] Lukas Brozovsky and Vaclav Petricek. Recommender system for online dating service. *arXiv preprint cs/0703042*, 2007.
- [10] Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, 2016.
- [11] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
- [12] Quoc V. Le and Alexander J. Smola. Direct optimization of ranking measures. *CoRR*, abs/0704.3359, 2007.
- [13] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.
- [14] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.
- [15] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA., 2007.

- [16] Denis Parra and Shaghayegh Sahebi. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2*, pages 149–175. Springer, 2013.
- [17] Lift analysis – a data scientist’s secret weapon. <https://www.kdnuggets.com/2016/03/lift-analysis-data-scientist-secret-weapon.html>. (Accessed on 2019-11-06).
- [18] Rui Xu and Don Wunsch. *Clustering*, volume 10. John Wiley & Sons, 2008.
- [19] B.E.M.L. Sabine Landau, B.S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. A Hodder Arnold Publication. Wiley, 2001.
- [20] Amandeep Kaur Mann and Navneet Kaur. Survey paper on clustering techniques. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 2(4):803–806, 2013.
- [21] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.
- [22] Pasi Franti, Olli Virmajoki, and Ville Hautamaki. Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1875–1881, 2006.
- [23] Jianqing Fan, Qiang Sun, Wen-Xin Zhou, and Ziwei Zhu. Principal component analysis for big data. *Wiley StatsRef: Statistics Reference Online*, pages 1–13, 2014.
- [24] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [25] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.
- [26] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 2010.

- [27] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14, 03 2001.
- [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [29] Hong Yu, Zhanguo Liu, and Guoyin Wang. An automatic method to determine the number of clusters using decision-theoretic rough set. *International Journal of Approximate Reasoning*, 55(1):101–115, 2014.
- [30] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.
- [31] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [32] Donghui Yan, Ling Huang, and Michael I Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 907–916. ACM, 2009.
- [33] Graham Upton and Ian Cook. *Understanding statistics*. Oxford University Press, 1996.