UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

ML APPROACH TO BUILD AN NBA MODEL TO OPTIMIZE CLIENT ENGAGEMENT

GERARD GÓMEZ JORBA

Thesis supervisor: ALEJANDRO MUÑIZ (BOEHRINGER INGELHEIM ESPAÑA, SA)

Tutor: MARTA ARIAS VICENTE (Department of Computer Science)

Degree: Master's Degree in Data Science

Master's thesis

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Contents

Abstract

When you are ill or sick, you normally go to the doctor or take a medicine. These medicines can be classified into 2 groups, retail and prescription. Depending on the group, your client also changes, the first group is the general public, and in the second group is the Health Care Professionals (HCPs). Pharmaceutical companies have customer-facing roles to promote the different products and services they provide. This study addresses 2 needs, one directed to help the sales reps in their day-to-day interactions with the HCPs, and the second is to generate customer journeys optimized by customer engagement. To approach this problem we harnessed the interaction data and used Markov Chains, Monte Carlo, and Genetic Algorithm to get the journey and optimize it. The results show that there is an improvement in the CTR of the mails so it would be interesting to further investigate this method for its implementation.

1 Introduction

When you are ill or feel sick, you normally go to the doctor or take a medicine. These medicines that you take can either be classified into two different big groups. The first group would be the ones you need a prescription to buy, so you need a doctor to prescribe them before going to the pharmacy and getting the medicine. The second group are the ones you can buy without a prescription. These two groups have major differences, like how often we can take them, the quantities, and the regulations on them. A big difference is the target group to whom you are selling the product, in the no-prescription medicine group, your target is the general public, who are the ones that are going to the pharmacy and buying the medicine. These kind of medicines are for when you have a headache or a fever, and the ones that fall into this group are ibuprofen, naproxen, paracetamol, acetylsalicylic acid (ASA), and many others. In the first group, the medicine needs to be prescribed by an HCP to be acquired, so you are selling the product to the health care professional, also known as HCP, instead of the general public. HCPs are anyone who maintains human health through the application of principles and procedures of evidence-based medicine and caring [Org13]. They can act with the ultimate goal of improving the human health and meeting the health needs. These professionals are medical doctors, nurses, midwives, dentists, and pharmacists. This kind of products are taken in doses and in the time prescribed by the doctor [Natnd]. In this group enter products such as insulin, opioids, and others.

1.1 Marketing in the pharmaceutical industry

To provide information about company products and services, including the benefits and how they can help specific patients, the cost or the potential side effects, and how to manage them pharmaceutical companies use customer-facing roles, such as sales reps, to interact and talk with the different HCPs. Customer-facing roles promote the different products and services the company has. These sales reps showcase the advantages the company has in front of the competition to promote the company portfolio in front of the HCPs. However, sales reps are not the only ones who are in contact with the HCP. There are the Medical Scientific Liaisons (MSL), who are relevant figures when representing the credibility and objectivity of the company and their products [CESnd]. They must be experts in the product they represent as the coordinate the flow of clinical information. Also, there are other important roles that are in contact with the HCPs, such as Market Access roles, which introduce the new products into the market before they are released. They also talk about political roles to change their perception of an illness.

There are several ways these sales reps can impact the HCP, the most obvious one are the face-to-face (F2F) meetings, but there are other ways such as follow-up emails, and emails sent from the central office. Other ways to impact the HCP is by inviting them to congresses or product launches. These events are normally congresses or presentations by experts to other HCPs about an active principle. These HCPs invited to this presentation need to be experts in the field, for example, a psychiatrist can't be invited to an event where a diabetes product is being promoted. These events are only educational or training. For the training events, the HCPs are invited by the Marketing team and are short courses or masters. Also, HCPs are impacted through phone calls, but they are not very popular, in-person interactions are much more appreciated. Also, in the Mediterranian countries the in-person meetings are much more valued than in other countries, as seen in figure 1, where the top 3 countries with the most F2F meetings are the Mediterranean ones and the Nordic countries are the ones with the least in-person 1 to 1 meeting and the most remote meetings. Also looking at the plot we can see how the in-person conferences, as well as the emails, are maintained with similar proportions in all the countries.

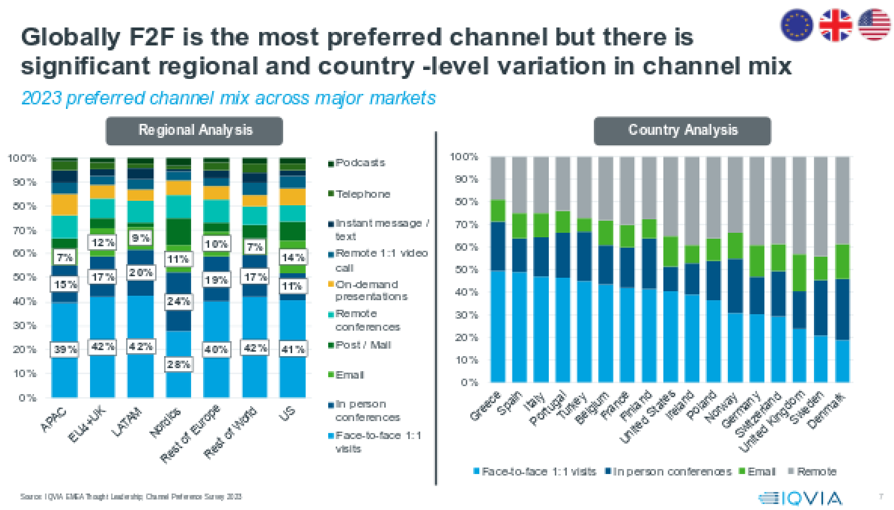

Figure 1: [IQV23]

Lastly, there are the screen-to-screen meetings (S2S), which have been given more importance since the COVID-19 pandemic, as the sales reps were not able to leave their homes. These F2F meetings are the core channel and are the most impactful way to make the HCP prescribe the product. This can be seen in figure 2, where it is seen that the 1 to 1 visits are the preferred ones in 2023 with a preference of a 41%, compared to the small preference the remote 1:1 video calls, which have a preference of a 4%. We also see that the HCP also has a big preference for in-person conferences with a 16%.

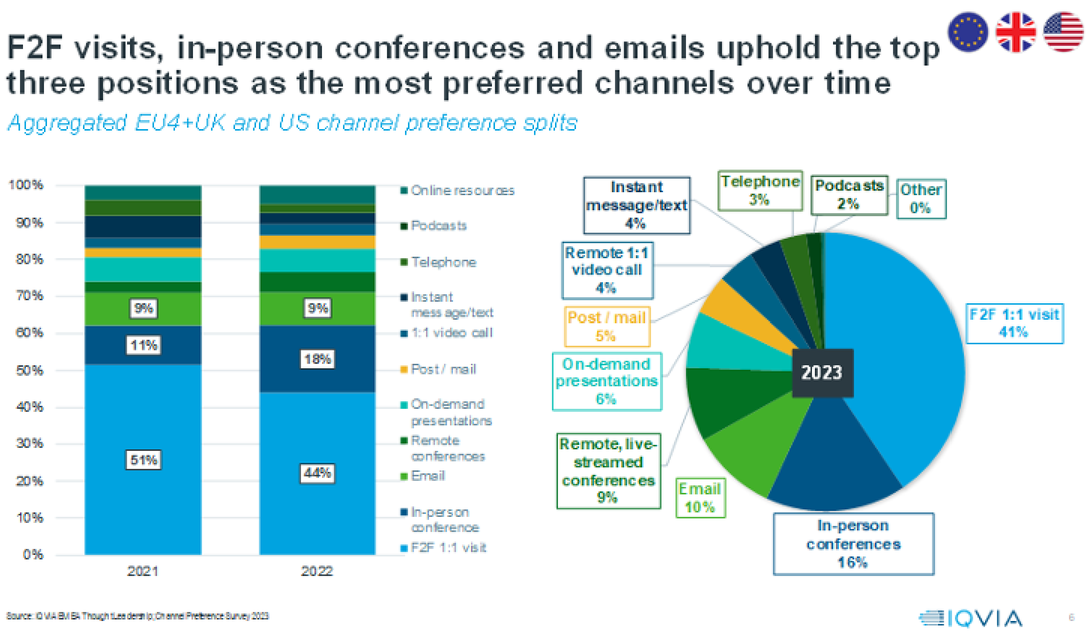

Figure 2: [IQV23]

These F2F meetings work in the following way: It is during these meetings that the sales reps promote the products they represent, show the latest updates on the product, give them the physical material they have on the products, such as the diptychs and triptychs, and also try to get a key message through related to the product they are promoting. The F2F meetings are very helpful as they help build long-term business relationships, as HCPs feel a better connection with the sales reps in in-person meetings rather than in screen meetings. But all these channels need to go together in a seamless way [Mednd]. Also, F2F meetings help increase the emotional relationship between the HCP and sales rep, which in the end is the company, so the better the relation between the two, the better the relation from the HCP with the company.

1.2 Regulation evolution

However, unlike consumer goods, prescription medicines face stringent advertising regulations that prohibit their promotion through mainstream channels such as television, social media, and billboards. These restrictions can be seen outlined in Article 4 of BOE-A-1994-17681 and Annex 3 of the 'Codigo Farmaindustria' [fe23]. This restricts very powerful channels such as social media, billboards, or even television. The European pharmaceutical industry follows a comprehensive set of guidelines known as the European Code of Good Practices for the Promotion of Medical Products, in Spain, Código Farmaindustria. This code, endorsed by major pharmaceutical entities across the continent, sets regulations aimed at ensuring ethical and responsible promotional practices within the industry. Consequently, the limitations on advertising mediums like social media, billboards, and television serve to uphold the integrity and credibility of pharmaceutical promotion, safeguarding public health interests. By following these regulations, pharmaceutical companies demonstrate their commitment to ethical marketing practices and prioritize accurate and unbiased information to HCPs. These restrictions accentuate the industry's dedication to maintaining transparency and integrity in its interactions with healthcare stakeholders.

Also, in Catalonia, there is the 'Guia Catalana del Medicamento' [Gen24] which is the regulatory law for the promotion of human-used medicines. In this guide, they explain regulations under which the F2F meetings are held in Catalonia, *Point 4.4 Visita Medica*. Also, it explains the characteristics of the Advertising Media needs to have, this content needs to be basically scientific and the basic objective does not need to be the prescription, dispensing, sale, or consumption of medications, but to provide technical information. Also explains how the publicity needs to be in a digital environment.

1.3 Multichannel and Omnichannel Markeiting

Nowadays, with all the different platforms there are available and ways to contact the HCPs, the companies need to be where the HCPs are, these are websites, phone calls, F2F visits, S2S visits, conferences, emails, and others [SAS]. It is under this statement that multichannel marketing started. It is a practice to impact the clients in a combination of direct and indirect communication channels helping attract more customers. One of the main ideas is that in multichannel marketing the customer can choose how to contact with the company, choosing their preferred channel. Multichannel helps get an image of the different kinds of clients the company has, so it can also segment them to impact the HCPs of interest [Mar]. Some of the challenges multichannel marketing has are:

- Efficient Management, as the more channels there are the more management there needs to be.
- Proper Marketing Attribution, which is to know which marketing channels are the ones contributing the most to the prescriptions. This is a challenge as the higher the number of channels, the higher the attribution is.
- Leveraging Marketing Analytics, due to lack of expertise in the organizations.

Have you ever received an email after visiting a website and looking but not buying an item? Or have you bought something on the internet, but the only way to return the item is to go to a physical store? All of these strategies are marketing strategies used to make you engage more with the brand. Here is where the concept of omnichannel comes into play. If sales reps just impacted the HCPs through different channels, with no strategy behind them, this would be multichannel, we just contact them through multiple channels, but without taking into account the data, the history behind them, their behavior towards the company, etc. What omnichannel tries to do is integrate all channels, including online ones, like websites, approved emails, ..., and offline ones, like face-to-face visits and events, into a unified customer experience, which means regardless of how customers interact with the company, they should receive consistent impacts, so they build trust and loyalty, which in the end it will translate to more sales. Also, omnichannel brings customer-centricity to the front, which prioritizes the needs of the customers aiming to have personalized experiences by leveraging data insights about customer behavior and preferences, all of this aiming to a seamless channel communication. So, to put this into context, an example of omnichannel marketing in the pharmaceutical context, is after an HCP is visited, there is a F2F meeting, and the next mail he/she will receive will be about things talked about during the F2F meeting. Another example would be if the customer goes to an event, during the next meeting the sales rep will know this and can center the meeting in another way knowing these facts. These series of actions are called customer journeys and lie at the heart of omnichannel marketing strategies. Optimizing this sequence of impacts is essential in the acquisition of new customers, being more profitable, and fostering lasting relationships.

Omnichannel marketing differentiates from multichannel marketing by integrating seamless experiences across each channel, while in multichannel, sales reps would just be impacting through different channels, with no integration between them. As the customer moves across platforms the transition between each channel is seamless and each channel is informed. So if an HCP went to a certain event, with multichannel, on the next visit the sales reps wouldn't have that information, on the other hand, in omnichannel, the sales reps have that information. This shows again that by implementing omnichannel, the company is really taking a customer-centric approach.

1.4 How do recommendation systems help the omnichannel approach?

With the introduction of data into our day-to-day lives and also with the internet enterprises have been able to get a lot of information about our us, and how we interact with them. From shopping for groceries to selecting entertainment, we are filled with options at every turn. With all these different options and information they are able to get a really good picture of our behavior and opinion towards them. This is where recommendation systems step in, serving as digital guides that tailor suggestions to our preferences, habits, and needs. By leveraging advanced algorithms and harnessing the data at their disposal, recommendation systems have revolutionized how we discover products, content, and experiences, enhancing both convenience and personalization in our digital interactions.

One example of a recommendation system are Boltzmann machine, is a bidirectionally connected network that excels in learning and adapting to complex patterns within data [AHS85]. Restricted Boltzmann machines, also known as RBM, are improved Boltzmann machines, by adding some restrictions they simplify the learning. RBM employs a form of unsupervised learning, allowing them to uncover latent features and associations within large datasets, thereby facilitating more accurate and personalized recommendations. Another well-known recommendation technique is Singular Value Decomposition (SVD), a mathematical method widely utilized in collaborative filtering systems. Collaborative filtering relies on the collective wisdom of user feedback to make recommendations, with SVD enabling the decomposition of the user-item interaction matrix into lower-dimensional representations. This enables recommendation systems to identify underlying patterns and relationships, ultimately enhancing the quality and relevance of suggested items. Also, it is worth mentioning that a combination of these two models, RBM and SVD++, a variation in the SVD, won Netflix one million dollar price, achieving a 10% increase in accuracy in the recommendations.[Gow14]

Next Best Action Models, also known as NBA models, are a sophisticated approach to the customer engagement approach that leverages data with artificial intelligence models to suggest the next best action in the customer journey through the analysis of preferences, necessities, and context. It works by building a large library of possible actions, which works as the bedrock upon which the model relies to make informed decisions. Through algorithms and machine learning techniques, these models shift through an array of possibilities to pinpoint the action most likely to resonate with the individual customer. These kinds of models have a seamless integration with the omnichannel framework. By synchronizing through various channels these models ensure a cohesive experience for the customer. This allows organizations to provide the best 360 experience to the customer, facilitating personalized interactions at each step. In the end, these kind of models are recommender systems that have a temporal term added into them, which acts as the context of the recommendation, as instead of recommending an item, it tries to recommend which will be the next action to perform. NBA models can be built to maximize whatever metric you are interested to increase. These metrics can be sales, engagement, and many others.

NBA models are used in various of different domains. For example in marketing or Customer Relationship Management (CRM), where they are used to optimize marketing campaigns by targeting users more strategically to optimize certain metrics or KPIs, like Cost per Click, revenue, customer satisfaction, or improve personalization, customer engagement, and customer experience by impacting the user at the right time, though the right channel, and with the right content.

The problem proposed is that with the addition of omnichannel, we want to impact the different HCPs with the correct channels and content in a seamless manner. But with the number of HCPs the sales reps have in their account plans, the universe of a sales rep is all the HCPs that the sales rep visits or interacts with, this number can go up to the 300s, and they need to keep track of all of a lot of information regarding them, like the number of visits they have done, and which content they have seen and haven't seen, and many other things. So with our model, what we are trying is to help them facilitate and give a bit of organization to the lives of the sales reps by recommending when to visit the different HCPs and also with which message, and will also recommend when to send follow-up emails. This way we can try to maximize the engagement of the HCP. Also, we are helping sales reps in decision-making, as it will tell them when it is best to visit those HCPs that are not as friendly with them and that normally they would leave for another day.

2 Background and State of the art

2.1 Data

Regarding the data available, we have information from all the different channels through which the company contacts HCPs or channels through which HCPs can interact with the company. These channels are Approved Emails, eMailings, F2F, S2S, events, phone, ..., also used personal data from the HCP, and lastly, territorial data, at sanibrick level. A sanibrick is a small region inside a city or a village, this is the way pharmaceutical companies divide their different territories.

For each content the following information is available:

- Date the interaction was done with the HCP
- ID of the HCP, also known as onekey
- some details about the content
- information about the interaction of the HCP with the content, ID
- as well as the name of the content, and channel through which the HCP was impacted.
- Also, by doing some preprocessing of the data, we can get the first time this content was used.
- After content has been sent to an HCP, we have information about how they interacted with it. There are three ways an HCP can interact with a content exposition, interaction, and conversion. An exposition is when for example an email has been sent to the HCP but not opened, an interaction with the content would be when the HCP opened the email, and lastly, there is the conversion, which refers to the HCP clicking the link on the email.

Regarding the web information, we decided not to include it, as this kind of interaction cannot be controlled by the sales reps. We can only incentivize the web impacts that come from Approved emails, as these have a landing web associated with it.

Regarding this data, we chose to look at 4 different products from 3 different franchises. A franchise is a group of products with the same selling idea, or that are in the same age of maturity. Products 1 and 2 are the key products, meaning these are the products that we will build customer journeys for the sales reps to follow. These two products are from the same franchise. Products 3 and 4, will be represented by sales reps that do not follow the same customer journey, so they will not be following the recommendations given by the model. A problem encountered is that some HCPs have an overlap of products. In that case, we need to take them into account, as when an HCP is impacted, even if they are impacted by different sales reps, they only see the impact as a company impact. So the model will have to react in response to Products 3 and 4, for example, if the HCP has a F2F programmed for week 4 but in that week he was already visited by the sales reps from another Product we will have to look for another alternative. Due to confidentiality issues, the name of these products cannot be disclosed.

Also, we have information about the HCP, this kind of information is personal information, like its specialty, and secondary specialty, when they got the medical degree, but also other information such as their sex, the hospital they work at, the zipcode, its sanibrik. In addition, we use the HCP behavior to get other data such as how is their behavior towards a product, how technological are they, or if they are new doctors. Another important characteristic is if the HCP is target or not target, depending on this the HCP is visited by a sales rep or not. Lastly, also have territorial data at a sanibrick level, and we have data about the ratio of people over a certain age, the total population, amount of hospitals, number of beds in a hospital, and also have the code of that region, and to which autonomous communities they are from.

Also, it is worth mentioning some of the data limitations there are on the pharmaceutical field, due to the strong regulations, that conditioned the analysis. The first, and probably the most important, is that there is no nominal prescription data, that is why we chose to optimize the engagement instead of the prescriptions, as we have data at that level. The most granular level the data is available at is at a sanibrik level.

2.2 Markov Chain algorithm

The Markov Chain, MC from now on, is one of the most well-known models. Markov chains were developed by Andrei A. Markov, a Russian Mathematician, and are stochastic models that describe sequences of possible events [Mah21]. The probability of transitioning from one state to another only depends on the current state of the system, meaning is stationary. These kind of models use only knowledge known of the present state, so no information from the past is used to predict the future. This characteristic is known as memoryless property. Nodes represent states and arrows represent transitions, this is how Markov Chains are represented. These arrows that connect the different states are weighted, meaning that the higher the weight the higher the chance of transitioning from one state to another. As they are probabilities, these weights must go between 0 and 1, and for each node, the sum of weights of all its outgoing edges is 1 [Hay13]. A simple example of a Markov chain can be seen in figure 3. In this example we can see that state E has a probability of 0.3 for staying in state E and a probability of 0.7 for moving to state A. On a further note, we can make sure these probabilities are okay by adding them up and making sure they add up to 1, for state E $0.3 + 0.7 = 1$, then the transition probabilities for state E are okay.

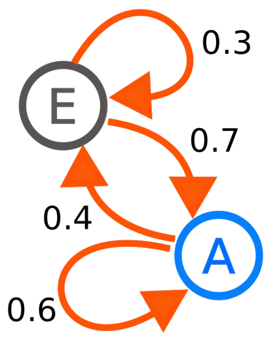

Figure 3: Example Markov Chain,[Wiknd]

2.3 Monte Carlo simulations

Monte Carlo simulation is a mathematical technique used to predict, understand, and analyze complex data. This includes decision-making, statistical analysis, resampling methods, and many [Mur03]. These kinds of methods are important as they add uncertainty and randomness to the model. In contrast to the conventional forecasting methods that give a more deterministic answer, the Monte Carlo simulation adds randomness to the model [Amand]. This model has three different components the input variables, the output variables, and the mathematical model, which in this case will use a Markov Chain shown in 2.2. The input and output variables used in our this implementation are explained in 3.1.3. In the context of our analysis, the Monte Carlo simulation serves as a powerful tool to forecast the optimal course of action to influence the HCP. By harnessing the principles of probability and randomness, this simulation technique offers a diverse array of potential pathways, ensuring that our approach is not bound by deterministic constraints. Through the exploration of multiple scenarios, Monte Carlo simulation empowers us to anticipate and adapt to varying outcomes, ultimately enhancing the robustness and efficacy of our strategic decisions. An implementation of the algorithm and how it is used in this case, can be seen in 2

2.4 Genetic Algorithm

Once we have the customer journey, we need to optimize it, that way we can maximize the engagement. This sequence optimization is done using a genetic algorithm. But what is a genetic algorithm? These algorithms were developed by John Holland in 1975 and they model the way biological evolution works following a Darwinian philosophy, meaning natural selection [Hol75]. These kinds of algorithms try to recreate the way evolution works genetically, therefore with mutations, where in genetics a nucleotide would change for another, crossover, recombination, which is the combination of two different alleles in the offspring, and survival of the fittest, only choosing the top offspring. These algorithms make modifications to an original model and get different models out of it. Then they test each of the results to find which are the best and get to go to the next generation. This is repeated until an objective is achieved.

3 Solution applied

In the end, the problem to be solved is to make a model that makes recommendations to the HCP based on 4 things, or the 4 Cs, the Client, the Channel, the Content, and the Cadence. Also, they are put in order of importance. First, as the most important thing in omnichannel marketing is customer-centricity, we are interested in the HCP and how he feels toward the company. Next, is the *channel*, so how are we going to interact with the HCP, so are we going to visit the HCP, or is it better to send them an email because we visited them last week? Next comes the *content*, so to give in the right information, and the one that is going to improve the most the engagement of the HCP. Lots of times, the information given on the follow-up emails will depend on the information said during the F2F meeting. Lastly comes the *cadence*, this is how often are we contacting with the HCP, as an overflow is deprecated. To sum up, we are interested in impacting the client with the right content, through the correct channel, and in a seamless manner.

3.1 Methodology

Regarding the solution applied, divided the main problem, recommending to the sales rep when to interact with the HCP, how and with which content, into two different problems, the first being when and through which channel is to impact the HCP, and the second being which content to give the HCP. Why a Markov Chain model? These kinds of models are based on previous actions, which is a more logical kind of behavior, than just choosing an action based on the present. As we can see in [IRCnd], we can see that sending an email after a phone call increases the chance of contacting them by 16%, which in the end increases the chance of closing a sale. In our case, this would apply to sending a follow-up email after a sales meeting, or visiting after the HCP went to an event. The decision to employ the MC model emerges from its simplicity, facilitating both training and the generation of customer journeys with a high degree of customization. This adaptability is crucial for incorporating specific business rules essential to the company's operations, as explained in further detail in 3.2. To seamlessly integrate these predefined rules into the model's operations, we employ a rule-based system alongside the MC framework, ensuring precise actions are taken in accordance with the established guidelines.

3.1.1 Data flow

The data flow depicted in Figure 4 outlines the meticulous journey of our data, from its extraction to its utilization in our model. We begin by extracting the data from Athena, a powerful analytical service provided by AWS [Ama24]. Athena allows to analysis a variety of data types, including structured, semi-structured, and unstructured data, all stored within the robust infrastructure of Amazon S3. Leveraging Athena's serverless architecture, we efficiently retrieve the necessary data, which is then meticulously stored in the landing zone within S3. All extracted data is saved in the high-performance Parquet format, ensuring optimal storage and processing capabilities. Once all data is saved in the landing zone, we read and reformat the date column to save it in the formatting zone, we also save the files in the parquet format. With the data appropriately formatted, our attention turns towards preparation for preprocessing. This pivotal step involves consolidating data from various channels into a unique dataset. Furthermore, ordering the data in ascending order based on date ensures chronological coherence, essential for the training of the sequence generation modeling we are interested in. This step is done in the prepareData.py script. Lastly, we have the optimization of model weights. In the weightOptimization.py script, techniques are employed to improve the action weights, to try to maximize the sequence generation performance. The resulting optimized weights are stored in the exploitation zone, and prepared to be input into our model framework.

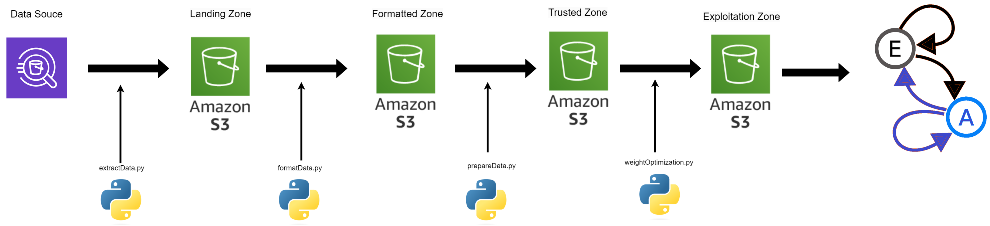

Figure 4: Data flow

3.1.2 Preprocessing

Regarding the preprocessing of the data. The first preprocessing step was to use only data from 2022 onward, this is due to a change in the behavior of the sales reps during the COVID-19 pandemic. Also, generated a weight for each of the interactions, when the HCP only had an exposition a weight of 1 was given, if there was an interaction a weight of 2 was given, and if there was an exposition a weight of 3 was given. These weights are useful to assess the interaction of the HCP with the company, the higher the better. These weights also will give a sense of the popularity of the channel in each situation. Then, as this model is built to be used to generate customer journeys for HCPs that are target, we also filtered for HCPs that are target and in the cycle that they were target. A cycle is a period of 4 months, cycle 1 from January to April, cycle 2 from May to August, and cycle 3 from September to December. So if for cycle 2 2022 the HCP was target but for cycle 3 the HCP was not target, we will have the customer journey for that HCP in cycle 2 2022 but not for cycle 3. In figure 5 we can see a representation on how the weights are determined for each kind of action.

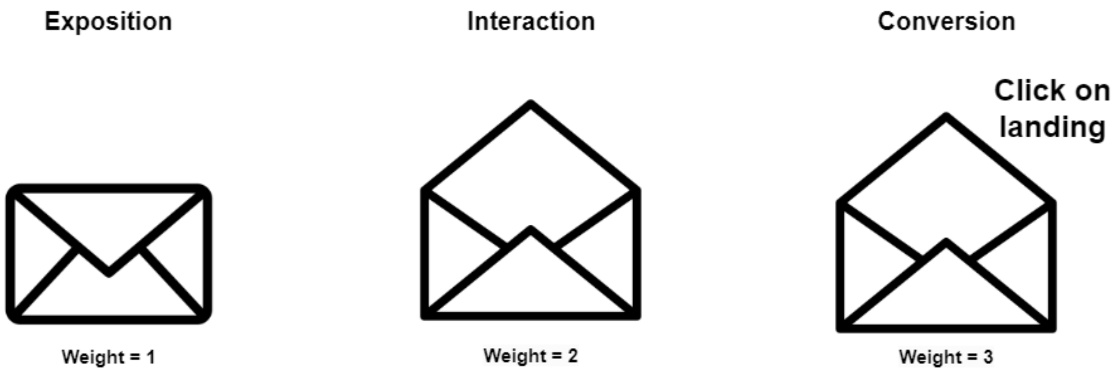

Figure 5: Weigh logics

The date preprocessing, as well as when getting the weight, was done in Athena. This was done as we are always interested in reducing the amount of data we have as soon as possible, that way there is a faster processing.

Next, we generated a new channel, this channel is the F2F+AoM channel. This channel represents a fusion of Face-to-Face (F2F) interactions and Approved eMails (AoM). The F2F+AoM channel is crafted to get instances where F2F impacts are closely followed by an AoM within a specified timeframe. Specifically, our criteria dictate that a follow-up email must occur within one week of a preceding F2F interaction or be scheduled one day before a scheduled F2F meeting. This stringent criterion ensures that only interactions demonstrating a proximate relationship between

F2F engagements and subsequent actions are included within the channel. This channel is created following the logic from [IRCnd], as we are interested in sending emails after the meetings. Also, they are used as a sanity check of the F2F meetings, as due to KPI completion some of them are not reliable. In figure 6 we can see the logic behind the F2F+AoM channel.

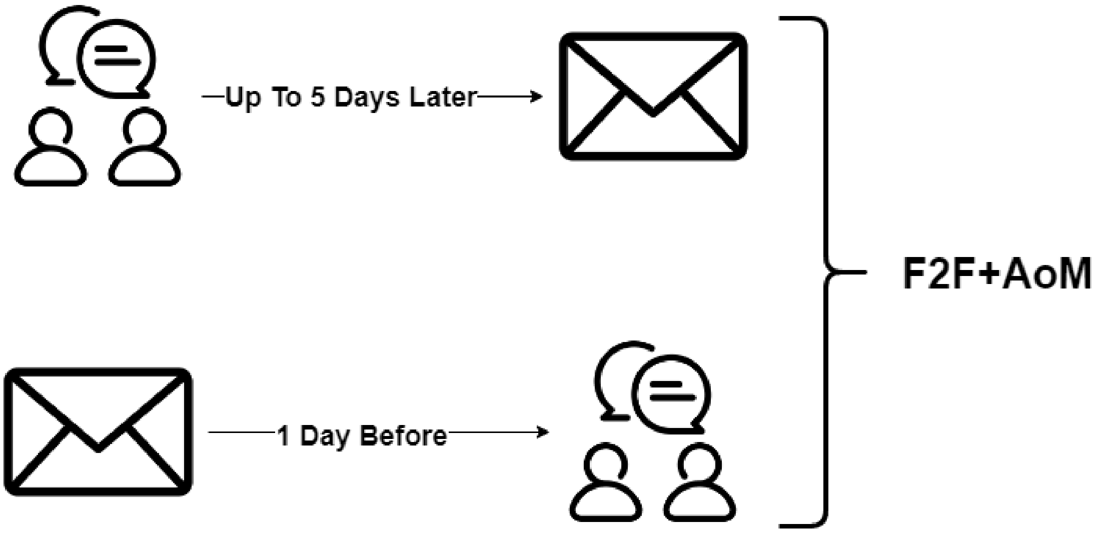

Figure 6: F2F+AoM channel logic

To have a deeper view into which was the behavior of the HCP with the company, some additional preprocessing was performed. First look at the fraction of interactions that weigh 3 for each channel of every HCP, which is the same as the CTR, as it is not the same to send 10 emails and interact with only 3, then to send 3 emails and interact with the 3 of them. So we multiplied this fraction by the weight. This step penalizes HCPs that do not interact with the company, so if an HCP hardly ever interacts with the company, it will have almost no effect on the model.

Another preprocessing step that was performed on the data, was to look into the average interaction in a cycle for an HCP, compare a cycle and the next, and add the difference between the two, then getting higher rates when the journey was better than the last, and lower when the journey was worst. In this preprocessing step, a filtering step was performed where journeys with only 1 action were filtered out as these are outliers. These journeys are considered outliers as from the customer journey side only contacting with an HCP once is not optimal in the first place.

3.1.3 Sequence Generation

For the first problem, the customer journey generation, a variation of an MC algorithm is used, instead of only looking at the past action, we look at the N past actions we have of the HCP, this is used as a context of where the HCP has been and how it has been impacted. This will help the model make more precise recommendations taking into account what the HCP has seen. The model has multiple modules which are, the fit module, this one is in charge of training the model, the sequence generation module, which is in charge of generating a customer journey, the evaluation module, in charge of evaluating the popularity of the sequence, and the unforeseen module, which is in charge of changing the model when something unforeseen has happened in the journey. Now, these modules are going to be explained, and also we are going to explain each of the parameters.

First comes the training module, this module is in charge of training the model and getting the probabilities that later will be used in the sequence generation and unforeseen modules. The fit module has 8 different inputs that are going to be listed an explained:

- The first one is the data that will use to train the model. This data will tell us which is the popularity of each action. This parameter is a pandas Dataframe.
- The `prev` parameter which tells how many actions will go into the context for the model to calculate the popularity for the next action.
- The third parameter are the predetermined probabilities, `predeterminedProb`,which are actions that we want to happen after another interaction, no matter the historical information. These actions have a business meaning, an example of this is we want the sales rep to visit the HCP 80% of the times after an event.

The fourth and fifth parameters are for the outlier treatment. Consider an action to be an outlier if they have a very low popularity. Popularity of an action is the calculated using the score of an action, using the preprocessed weight explained in 5 given the context of the action. The action will have positive popularity when it works and negative popularity when it doesn't work. So if an action for a certain historical information is shown not to work, for example, has a negative popularity, we will remove that connection from Markov Chain. For the outlier treatment, we have the following parameters:

- The `dropOutliers` is a `true` or `false` parameter that allows us to eliminate or not these outliers.
- The `outlierNumber`, is the minimum popularity an action needs to have after a certain context to not be considered an outlier.
- The next parameter is `removeInnerLoop`, which treats the possible inner loops there can be in the model and whether we want to allow them or not. So this parameter is again a boolean that will remove inner loops in the Markov Chain when `true` and keep them when `false`.

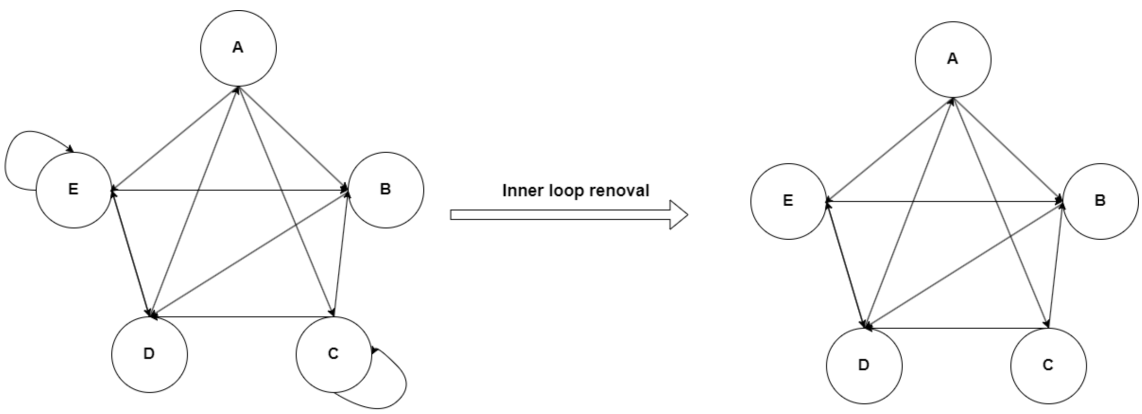

Figure 7: Inner Loop removal

Lastly, as mentioned before, the HCPs of interest can be impacted with 4 different products, 2 that we are controlling, through the customer journey, and 2 that we are not controlling, as they are not from the same franchise and are controlled by other sales reps. So the last 2 parameters are to control these products:

- The first is `productTrain`, which is a list of all the products that are key for the model, in this case, products 1 and 2.
- The `weightNoTrain`, which is a number by which we multiply the rates of these products that are not represented by the sales rep, therefore giving them less importance overall.

The training works in the following way, the model first iterates through each HCP and gets all its actions ordered by date. Before counting to the pairs, which are made of context and action, the model builds the historical information, the length of it will depend on the prev number, and stores the rate of these actions. The historical information is from the N past weeks before the action was done. Next, go through each action, and count the pairs of historic and action. If the action is not from one of the products stated in the productTrain list their rate is accumulated to be later added to the next action multiplied by the weight no train. This is done to take into account the actions done to the HCP that we do not control. At the moment all the context has been build, the count of the pairs, context+action, starts. These pairs are counted in the following way:

$$\sum historicRates \cdot rateAction + weightNoProduct$$

[1]

After adding the resulting weight to the pair, we update the context, by eliminating the oldest one and adding to the cue the new action. Here we multiply the weights instead of adding because we are interested in giving more weight when the action ends up in interaction or conversion. For example, if we do have only a prev of 1 and the previous action had a rate of 3, if we added with the next the difference between an exposition and an interaction would be 1, but if we multiply the difference between an interaction and an exposition would be of 3, therefore separating more the good from the bad actions. When the difference in date between two actions is greater than 7 days we place as many blackout weeks as groups of 7 days there are, except when the difference is greater than 16 weeks which is equivalent to a cycle. These blackout weeks are weeks where there was no interaction with the HCPs. When all the sequences have been traversed we eliminate those that are outliers, if we want outlier removal, can also remove the inner loops, if we want to remove them as well. Then we calculate the frequencies of each action after the historic. These frequencies give you a sort of popularity point of view to the action about the action after the context, as the higher the frequency is the higher the popularity of the action is. Lastly, update the weights so the precomputed probabilities are respected. In figure 8 we can see an example of a possible Markov Chain with all the different channels. This would have previous information of 1 as it only takes the last action done into account. A pseudocode explanation of the model can be found in 1.

[1] `historicRates` are the sum of all the rates in the context. `rateAction` is the rate of the action happening after the context. `weightNoProduct` is the sum of the rates of the products that are not in the list `productTrain`

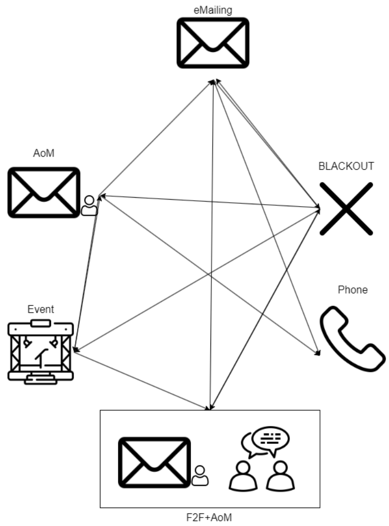

Figure 8: Example of a possible MC

Algorithm 1 fit function

```
procedure FIT(self, tabla, prev, predeterminedProb, dropOutlier, outlierNumber,
removeInnerLoop, removeChannels, productTrain, weightNoTrain)
    weightNoProduct ← 0
    for onekey in set(tabla["onekey"]) do
        tableonekey ← tabla filtered by onekey
        context ← empty list
        for action in tablaonekey do
            if action["Product"] not in productTrain then
                weightNoProduct += action["Rate"] · weightNoTrain
            else
                if context < prev then
                    build context
                else
                    add popularity of pair (context + action) 3.1.3
                    delete first action of context
                    add action to context
    if dropOutlier then
        dropOutliersFunction(outlierNumber)
    if removeInnerLoop then
        innerLoopRemovalFunction
    Transform probability of each pair to frequency after context
    transform Frequency using predetermined probabilities
```

Next comes the sequence generation module. This module is in charge of generating the customer journey, given some parameters, and following the probabilities calculated during the training. The parameters this module needs are the following:

- The first is `length`, which is the length of the sequence it is going to generate, depending on how the model wants to work the length can vary. By how the model is wanted to work meaning that if it wants to react to what is happening as it is going, for example, the cycle can be divided into 4 parts, 4 weeks each, and just build 4 weeks customer journeys. But if you want to have a pre-established customer journey a length of 16 can be given and the model will give you the customer journey for the whole cycle.

- The next parameter is the `mccp` which will tell the amount of F2F visits the sales rep needs to make in the whole cycle. This completing this number is normally a key performance indicator of the sales rapes. This number is calculated using different parameters like the hospital where the doctor works, ...

- Third, we have the historical information or `context`, the length of this parameter will depend on the prev parameter determined in the training and shows the impacts the HCP received in the last n weeks. This will be used to know which are the next best actions to be done by the sales rep depending on the popularity they had given that history.

- The fourth parameter is the `maximum_content`, this parameter will tell the model how much content is available for an HCP, as no repeated content is going to be sent. Also, the content is updated each cycle, so each HCP will have different amounts of available content and also will fluctuate throughout the cycle as each content can also be added in the middle of the cycle.

- The fifth parameter is the `fixed_events`, which are actions, normally events, that happen at a certain week and cannot be moved. Also, this parameter can be used to input the festivity weeks or when the HCP is on sick leave, as BLACKOUT weeks.

- Also there are the 2 parameters that tells us if we have the permisions to send an email or call the HCP. This parameters are booleans that when `true` is that we have the permision and when `false` we do not have the permision. This parameters are `ePermission` for the eMails and `mPermission` for the phone.

- Lastly, we have the `probVar`, which is the probability of varying the mccp, this is a small probability of increasing or decreasing the mccp by a bit. This is done to try to recreate the reality as the sales reps normally do not do the whole mccp, or if they want they also make more visits.

Also, there are some objectives associated with this module, first, there is an interest in maintaining the business rules from the precomputed probabilities. Also, the F2F needs to be separated, because in history we have seen all of them clamped together in the end or the middle, and this is not the most optimal way to do it as you want to maintain the contact throughout the cycle and not put it all in a short period.

The sequence generation works in the following manner. First is randomly chosen if the mccp is increased or decreased, with the probability given as a parameter, in the case that is chosen, it will increase, or decrease by 0'1, 0'2, or 0'3 randomly. Next given the mccp and the overall length of the sequence, the frequency of the F2F is determined, so they are spaced out, and also complete. Then for each week of the sequence that is being generated, first look at the fixed events, if there is one it is put into the sequence, otherwise, we move one, then look at the frequency of F2F, and determine if one needs to be set or not. Lastly, look at the historical information and make a choice using the probabilities set during the training, and remove 1 from the content available for that action. In the case that the chosen action has no content available, another choice is done, this is done 10 times, if there is no change, then one action is chosen randomly from all the different available channels. It is worth mentioning that F2F, F2F+AoM, and Events were taken out of the possible actions to take, as the Events are always on fixed dates and the F2F and F2F+AoM will always be set by probability metrics. The output of this module is a sequence composed of the history and customer journey built together.

Algorithm 2 sequence generation

```
procedure SEQUENCEGENERATION(self,length,mccp,context,maximum_content,fixed_events,ePermission,mP
    sequence ← list()
    F2FDone ← 0
    number ← random number between 0 and 1
    if number < probVar/2 then
        mccp ← round(mccp+mccp*randomChoice(0.1,0.2,0.3))
    else if number < probVar then
        mccp ← round(mccp-mccp*randomChoice(0.1,0.2,0.3))
    for week in range(length) do
        probOfF2F ← (mccp-F2FDone)/(length-week)
        if week in fixedEvents.keys() then
            add the fixed action to sequence
            update the context
        else
            randNum ← random number between 0 and 1
            if randNum < probF2F and F2FDone < mccp then
                F2FDone ← F2FDone + 1
                if ePermission and maxContent[follow-up email] > 0 then
                    add F2F+AoM action to sequence
                    maxContent[follow-up email] ← maxContent[follow-up email] - 1
                    update context
                else
                    add F2F to sequence
                    update context
            else
                Added ← False
                if context in the MarkovChain then
                    nextStep ← choose the next step using the probabilities
                    if nextStep maintains the business Rules then
                        add nextStep to sequence
                        Added ← True
                        update context
                if not Added then
                    while not Added do
                        choose randomly between one of the available channels
                        if nextStep maintains the business Rules then
                            add nextStep to sequence
                            Added ← True
                            update context
    return initial context + sequence
```

Next is the evaluation module, which is in charge of evaluating the popularity of the sequence and also looks at whether it maintains the stipulated events. This module has 3 different parameters which are:

- The `sequence` that is going to be evaluated, this sequence has the context already added to it.
- The `maximum_content` available. This parameter is already explained before in the sequence generation.
- The `fixed_events`, this parameter is also already explained in the sequence generation module.
- And lastly the `not_seen` parameter, which gives the popularity an action has when it never

happened before.

This module goes through the sequence and uses the first N actions as historical information. Then goes through the sequence, updating at each step the historic info, looking at the popularity score it had, and multiplying it by the previous. While iterating through the whole sequence, the module looks at whether the sequence has the fixed actions where they have to be and also if the sequence does not exceed the maximum content available, if any of these is not followed then the score is 0, otherwise the popularity score is given.

Algorithm 3 sequence evaluation

```
procedure EvaluateSequence(self,sequence,maximum_content,fixed_events,not_seen)
    score ← 1
    previous ← first self.prev actions from the sequence
    for action in sequence do
        if previous in self.Markov-chain and action in possible edges then
            prob ← probability of moving from action to previous, retrieving probability from Markov Chain
            score ← score * prob
            update previous
        else
            score ← score * not_seen
        if bussine rule not complete then
            return 0
    return score
```

The last module is the unforeseen module, this module is in charge of evaluating how the popularity of the sequence is affected once an unforeseen action is done. One example of an unforeseen action would be in week one it was expected for the HCP to go to an Event, but in the end, the HCP did not go. Then this module would be in charge of evaluating and remaking the customer journey, in the case it is needed. This module takes the following parameters

- The customer journey
- The week that the unforeseen action happened
- The action that was done
- The mccp of the HCP, explained before in 3.1.3
- The maximum content
- The threshold, this threshold is the minimum allowed difference in the popularity between the original customer journey and the modified customer journey with the action replaced at the said week.

This module works in the following manner, first, it looks at the difference between the two customer journeys. If the customer journey difference is lower than the threshold, then change is needed. In the case that the difference is higher than the threshold, then a new sequence is generated from the week the unforeseen action was done.

Once the sequence is gotten form the sequence generation module, it is optimized through the genetic algorithm, explained in 2.4. For the implementation used in this problem, the algorithm is divided into 2 parts, the mutation in which 2 actions are permutated no matter the location, with the exception that it is a fixed action in which case it is not permutated, and the crossover, in which given a location and 2 mutated sequences that come from the same sequence are crossed over. An iteration of this algorithm is a mutation for every position and all the possible crossovers. After an iteration, all the resulting sequences are evaluated and the 200 best are chosen. This value is a parameter and can be changed, the time of the optimization with the genetic algorithm will increase exponentially as this parameter goes up. This high number was decided because of the thought that to get to the customer journey with the best score you probably have to go through not-so-optimal sequences. This process is done 15 times. This value is a parameter and can be changed depending on the amount of changes you want to make. After this, the best sequence is chosen out of all, and this is the customer journey that will be used by the sales rep for that HCP. An example of the genetic algorithm can be seen in figure 9.

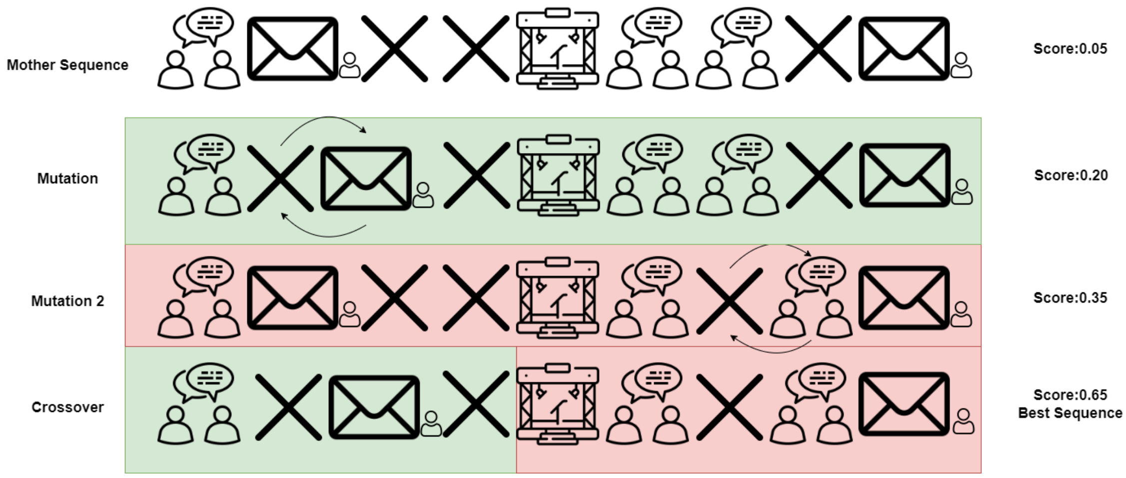

Figure 9: Example of a genetic Algorithm

3.1.4 Sequence generation architecture and sequence examples

The model architecture is shown in figure 10. This figure shows which is the actual life of each sequence. First, the MCMC algorithm is trained using the fit module, and then a sequence is gotten using the sequence generation module, 1 sequence per HCP, will call this sequence the mother sequence. Next, the sequence goes through the genetic algorithm and it will be optimized for the engagement to get customer journey with the highest popularity.

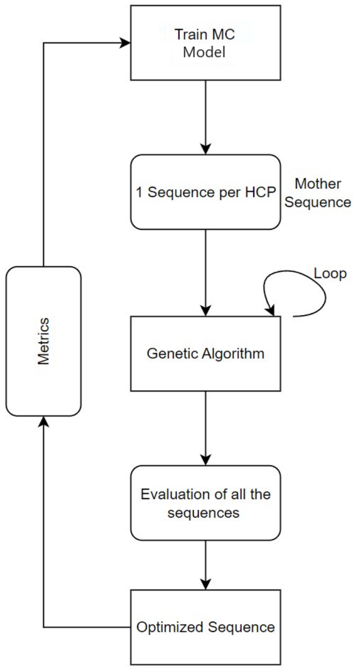

Figure 10: Example Sequence live cycle

One example of a sequence live cycle would be the following, first, we would train the model and get a sequence for one of the HCPs with its characteristics. The output sequence can be seen in figure 11. This is the mother sequence, and to optimize it, goes through the genetic algorithm several times to get a better distribution of the channels. The output sequence from the genetic algorithm can be seen in figure 12.

Figure 11: Example mother sequence

Figure 12: Example optimized sequence

In the optimized sequence we can see how the popularity of doing F2F+AoM visits in a row is better, also we can see that there is also a change in the order of the channels at the end of the sequence.

3.1.5 Content recommendation

Regarding the content recommendation, we implemented a simple recommendation only based on the key messages seen during a certain cycle and on the not-seen content. For this implementation, we defined 4 different data frames 2 for each product. One of the data frames will save if a content has been seen by an HCP, the other has the information for the number of times a KM has been seen by an HCP.

The content recommendation has the following parameters:

- `contentSeenProduct1` Data frame with boolean, with the rows being the onekeys and the columns being the different contents for product 1. When `True` the content has been seen by the HCP, otherwise `false`.
- `keyMessagesProduct1` Data frame where we count the number of times an HCP has seen a certain key message. On the columns there are the different km there are an on the rows there are the onekeys.
- `contentSeenProduct2` and `keyMessagesProduct2` same as product 1 parameters but for product 2.
- `contentKMdict`, this dictionary holds the key messages every content has.
- `percetageProduct1`, this parameter tells the parentage of parameters that are going to product 1. This is necessary as depending on the cycle we are in one product has more preference than another.

So the content recommendation was done in the following way:

1. We take the channel the impact is going to be done with this week.
2. Then we make the recommendation for one product. This recommendation takes into account the channel, as well as the number of times the HCP has been impacted with a certain KM. When choosing the content will choose the one that has the key messages least seen.
3. Then step 2 is repeated for the second product.
4. Then will choose with which product we will impact the HCP. In the case there is no content available for one product we will impact with the other product.

3.2 Problem particularities

Due to business and several benchmarks, this solution has to be fulfilled, it generated some particularities within it that now will be explained. The first problem particularity is the difference between the F2F and S2S channel, and the fight for digitalization within the field. Since the COVID-19 pandemic, there has been a change of mentality, and try to start introducing more digital visits, as they are more comfortable if you need to visit several clients that are far apart from each other. However, the S2S channel has not had the best reception from the sales reps, as it gives the impression to the HCP that you are not giving him as much importance as before. Here is where the first problem is, as the S2S meetings are not popular, they will never be recommended, therefore they will need to be forced into the sequence. The model will have a digitalization module, which will change one of the F2F meetings to a S2S meeting. That way incentivising the HCPs to have more S2S visits.

The next problem encountered is the need for some actions to happen after something happened, for example, there is a need for F2F meetings to happen after an event, which is good for this seamless interaction between the different channels there are, as after an event the HCP will have the concepts recent and the sales reps can get feedback about how it went, so a module was added that regulates this probabilities. Another business-related problem is the number of visits a sales rep needs to make within a cycle, this cycles are 16-week intervals, so there are 3 in a year. This amount of visits will vary depending on the cycle, but the sales reps must fulfill them, as they are chosen depending on several factors. This is something we will need to take into account when generating the customer journeys, so the amount of visits needs to also be forced into the customer journey. Something that also needs to be taken into account is the amount of content available, as it is limited, and something that does not exist cannot be recommended. So beforehand, we know approximately how much content is available. This will be used during the journey generation. Also, due to regulations, the content expires and after that expiration date, it cannot be used anymore.

Another constraint that needs to be looked after during the generation of journeys is what will be called fixed events, these are actions that happen in certain weeks and that cannot be changed. An example of these are the events, that you cannot move from one week to another, they are programmed for a week and can't be moved. Lastly, other things need to be looked after, and these are the special dates, like the beginning of a cycle, or the holidays. These are periods where the sales reps might have a different behavior, for example, during the summer holidays, normally in August, there won't be any interaction with the HCP, and this is because the sales reps are on vacation, and they won't contact with the HCP.

3.3 Solutions (ad-hoc)

The solution ad-hoc that is working at the moment is a trigger-based system. This is based on what are called micro journeys, which are a small sequence of impacts of a length of 2 or three, which follow the following characteristics after an HCP shows interest in a certain product impacts them in a predetermined way. An example of these micro journeys would be, in a video conference an HCP shows interest in a certain product, after this videoconference the HCP will be impacted with an AoM which is related to that product. This has been shown to improve the CTR in the AoMs which are the emails that are sent by the sales reps. The difference with the MCMC version is that we are centered on macro journeys, so long journeys try to maximize the whole cycle journey. In figure 13 we can see an example of this kind of solution. We can see when the HCP visits a company website the sales rep is notified. The sales reps then send a follow-up email regarding the content seen on the website. This way we can get omnichannel interactions.

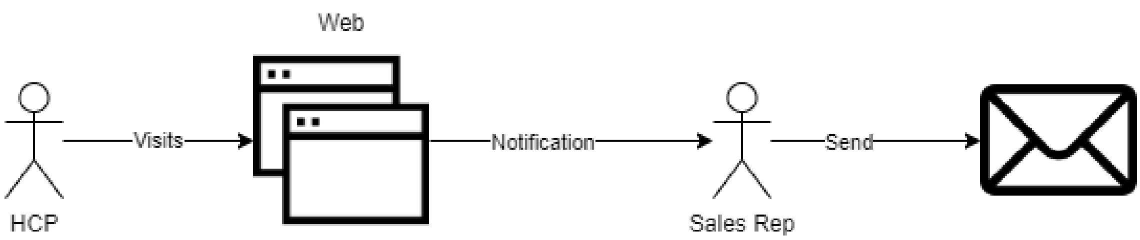

Figure 13: Trigger-based solution example

4 Evaluation and results

In machine learning, model evaluation stands as a crucial step in the development and deployment of effective predictive models. This step centers on the idea of seeing how well the model performs on unseen data. Model evaluation works as a systematic assessment of a machine learning model performance on a given data set, giving you an idea of the model's generalization capabilities. Within the scope of model evaluation, there are two fundamental types of problems, regression and classification. In regression tasks, where the aim is to predict continuous values, metrics like the squared error give you a great insight into the difference between the predicted and the actual values. In classification tasks, which deal with categorical outcomes, metrics such as the 0—1 loss and the Cross-Entropy Loss quantify the model's misclassifications or divergence from the true label distribution.

So to choose the best model to make the best recommendations an evaluation of the different models is needed. This is a difficult task as we cannot use any of the aforementioned metrics. So we conducted some research to investigate what is being done in other artificial intelligence areas. We ended up finding the generative AI areas very helpful, as in some way both jobs encounter similar problems, as you are generating something new every time. More specifically we looked into how image generation models and Large Language Models, also known as LLMs, are evaluated. For the image generation models, we looked into the Inception Score (IS). Introduced in [Sal16] and named after the Inception Network, it is a metric for automatic evaluation of the quality and realism of image generation models. It uses the inception model to every generated image to get the conditional probability distribution $p(y|x)$. More information about this metric can be found in the aforementioned paper. Regarding the LLM, we looked into the benchmark system for evaluating the different results. What is done several prompts are prepared and input into the LLM and the results from the LLM are saved for a later evaluation of them. It is worth mentioning that these benchmarks are useful for the comparison between two models and the result alone does not give you much information. These benchmarks are normally libraries of prompts built to get how the LLM performs on a specific task. One example of this is GLUE, also known as General Language Understanding Evaluation, which is centered on nine English sentence understanding tasks [Wan19]. A good performance in GLUE requires the model to share substantial knowledge about different tasks with different difficulties. Another benchmark is HellaSwag [ZHB+19], which is a dataset of sentences with the following format "Given a context and an ending, return a logit for that ending".

To evaluate the different models, inspiration was obtained from the aforementioned state-of-the-art evaluation techniques for generative models. So the two problems we encountered, assessing if a customer journey generation is better than another, and the evaluation of the sequence by the model, were solved by dividing the problem into two subproblems. The former problem was tackled by getting inspiration from the benchmarks and the inception score. We looked into the business rules the sequences need to fulfill and made a function that evaluated to which degree they were fulfilled. The latter problem was solved as if the problem was a time series problem, the training of the model was with all the data until a date, and the rest was used as test data. In this case, the Click Through Rate was used as an evaluation metric. Finally, a proof of concept, PoC from now on, is also being performed with the model to asses its performance in real life.

4.1 Evaluation approach

The evaluation approach was done as mentioned before in two different steps. This was a difficult task, as there are no metrics to evaluate the quality of a sequence telling you if the model is making good or bad recommendations. To try to solve this, first looked into the generation and quality of the sequences generated. The quality of the sequences is important as we want them to have some business sense and still follow the business rules. Secondly looked at how the model predicted the historical actions.

The first problem solution was inspired by the inception score and the benchmarks. To tackle this task, Excel was first designed with different parameters such as fixed events, different contexts, and others. All the parameters that can be used for the sequence generation are listed and explained in 3.1.3. The parameters in the Excel file were then used to generate different sequences, which were later optimized and used to evaluated the model with two metrics that have a business sense. First, we want the sequences to follow the precomputed probabilities, these are business rules that should be followed for the omnichannel experience. So we look at if these precomputed probabilities are being maintained after the sequence generation and optimization. This is important as, explained before, doing some actions after another is known to improve sales. Another characteristic from the sequence we looked at, is a good spread of the F2F meetings throughout the cycle. We saw a tendency for the F2F meeting to be all crumbled up at the end of each cycle, which is not efficient. So we look at whether the F2F meetings are spread throughout the cycle. To look into these characteristics, a function was designed to look into them and give them a score, in this case, the score we are looking to be the smaller the better. So this evaluation works in the following way:

1. Train the model with all the data

2. Generate a list of sequences using different parameters

3. Evaluate the different sequences, looking at whether they follow the business rules to get a score

The second evaluation performed was in a time series manner, and looking at the historical data we have. In this case, the Click Through Rate (CTR) was the metric of interest, as it is the only metric that we have that can measure the success of an action. CTR is the rate by which people click on the link inside the email. Our evaluation is based on the success of the emailing actions, so the channels involved are the follow-up email and the emails sent from the central office. As the F2F actions are always expected to be a success. So when evaluating the model, a first preprocessing step was done, we separated the data into train and test, the training dataset are all observations from the start of 2022 to 31st August 2023, then the test data went from 1st September 2023 to 31st December 2023, these dates are corresponding to the beginning and the end of cycle 3 2023. Also, filtered the data according to the ePermission of the HCP. If the ePermission is provided emails from central and follow-up emails can be sent, otherwise we can not. So to start the evaluation, the model is first trained with the train observations. Then, stored all the individual customer journeys done to each HCP, and lastly traversed through each of them looking for emails from central and follow-up emails. When traversing through each of the customer journeys when an email from central or follow-up emails was encountered, considered True Positive when the model could have recommended sending the email from the central office/follow-up emails and the HCP opened it, False Positive when the model could have recommended sending the email form central office/follow-up emails and the HCP did not open it. Considered True Negative when the model did not recommend sending the email from central office/follow-up emails and the HCP did not open it, and False Negative when the model did not recommend it and the HCP opened it.

		User Action		
		p	**n**	**total**
Action Recommended	**p′**	An email was recommended and the user opened it	An email was recommended and the user didn't open it	P′
	n′	An email was not recommended and the user did open it	An email was not recommended and the user didn't open it	N′
	Total	P	N	

Table 1: Logic for the confusion matrix

This methodology of looking at each action allowed us to make the confusion matrix, as seen in the table above, calculate the CTR of recommended actions $\frac{TP}{TP+FP}$, and the CTR of the not recommended actions $\frac{FN}{FN+TN}$. Also, we have these measures for each kind of mail, so we have the 4 metrics for follow up emails and emails sent from central office individually. Having them like this allows us to see if our model makes better recommendations for one than for the other. For this historic evaluation, a score was also taken and this is the difference between the CTR recommended and the CTR not recommended, the closer to 1 and positive the better the model recommends. In this evaluation, an action could be recommended if the probability in the model of it happening was higher than 0.2, otherwise, we consider it not to be recommended and also considered a mail to be opened if the rate was higher than 1.5, as after the preprocessing steps the value of 2 may have been lowered.

4.2 Results

4.2.1 Historical Results

To look at the performance of the model with the historical evaluation we plotted the follow-up email CTR, as it is the one that the sales reps are sending, and we compared it with the actual CTR for the follow-up emails. The CTR in the present for the follow-up email is 7.79%. We plotted the different `prev`, going from 3 to 9, and different `weightNoTrain`, and tried with 0, 0.3, and 0.5.

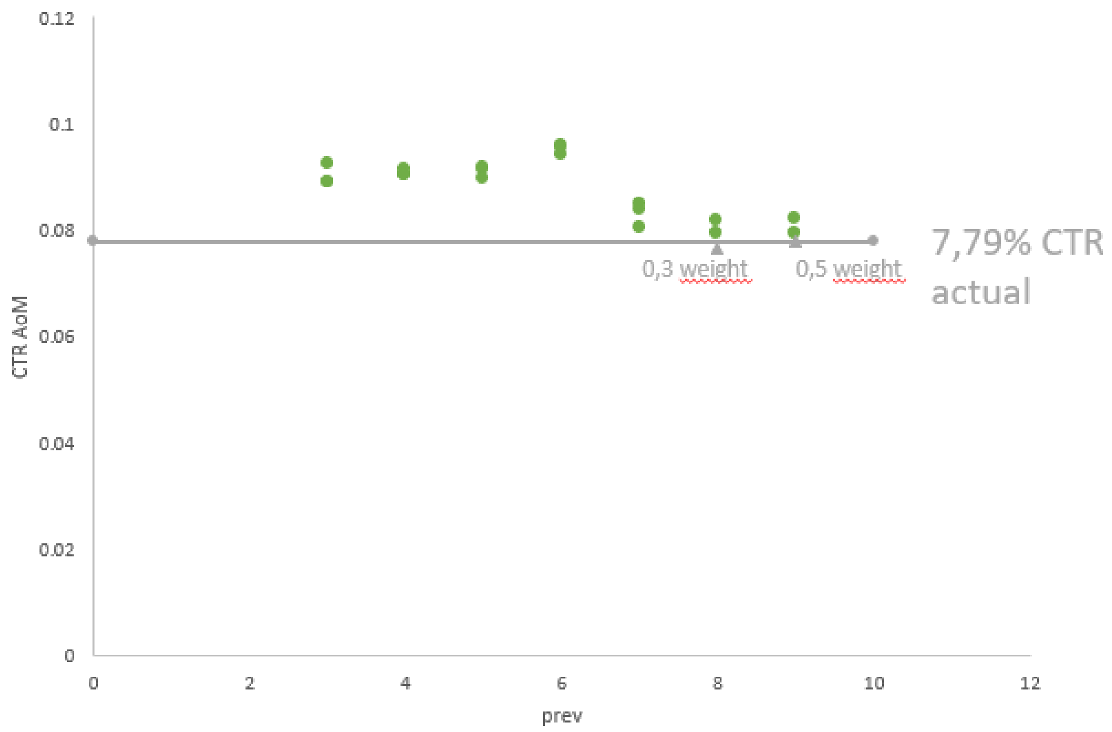

Figure 14: CTR gotten by the model in follow-up emails

In figure 14 we can see the follow-up CTR got with the recommendations, which are the different dots, compared to the one we have at the moment, which is the gray line. Looking at the plot we can see that almost all models, with the exception of the model with a `prev` of 8 and a weight no train of 0.3, and also the model with a `prev` of 9 and a weight no train of 0.5, are improving the CTR. We can see that the models with the best performance are the ones with the lowest `prev` parameter, up to 6 which is the one with the best performance.

In figure 15 we can see a plot to help us choose parameters. We can see on the x-axis the historical score. This historical score is calculated with the difference in CTR between the recommended and the not-recommended actions and follows the following formula:

$$\frac{TP}{TP+FP} - \frac{FN}{FN+FP}$$

[2] In the y-axis, we can see the F1 score, and for the size of the dots, we have the `wightNoTrain` parameter which is explained in 3.1.3.

[2]The meaning of the different TP, FP, TN, and FN can be seen in the table 4.1

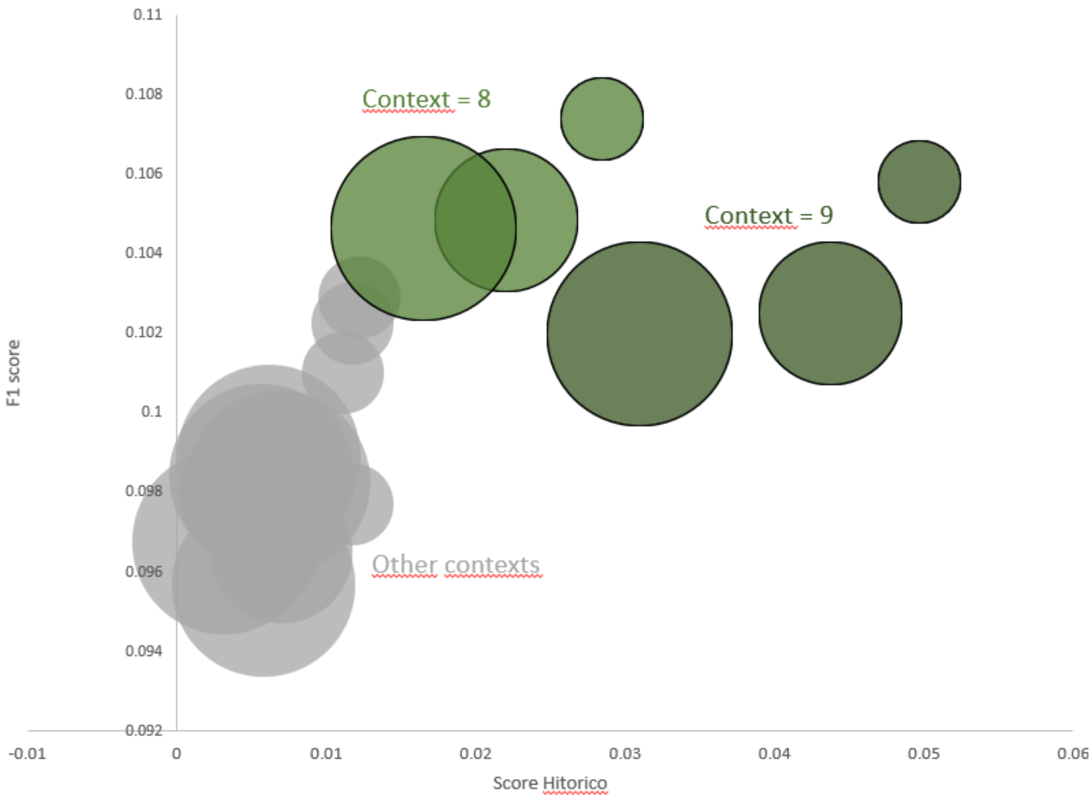

Figure 15: CTR gotten by the model in follow-up emails

At first glance, we can see that long context models perform better, but only up to a certain point, as also more context makes the model be much more specific. Another characteristic we can see is that the `weightNoTrain` makes the model perform a little worse than without it. Also, can be seen that the best-performing models are the ones with a context of 8 and 9 when considering the F1.

4.2.2 Performance

The performance and the execution time of the model for different parameters and sequence lengths in the Genetic Algorithm were also evaluated. These would help us choose the parameters to execute the Genetic Algorithm for the different lenghts of the sequences. In figure 16 we can see the results.

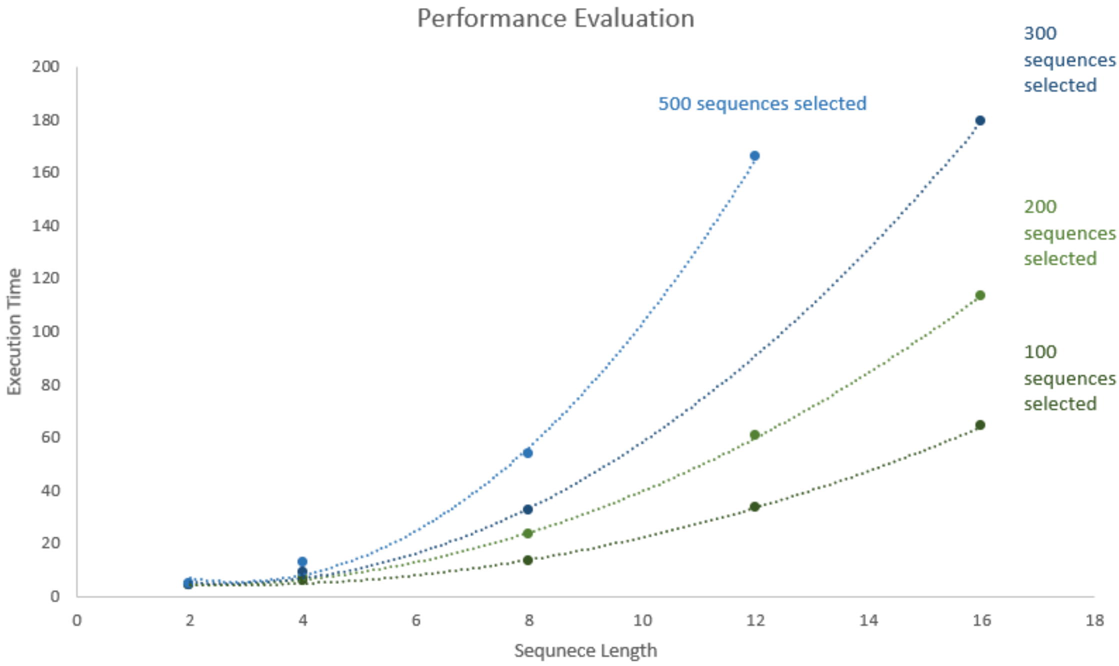

Figure 16: Performance of the model depending on the sequences chosen

We can see that for a certain number of sequences selected, the execution time of the model increases exponentially. Also, the higher the number of sequences selected is, as expected, the execution time is higher. Again, looking at the figure, we can also see that the difference in execution time between the 100 sequences selected compared to the 200 sequences selected and the 200 sequences selected compared to the 300 sequences selected have the same difference.

5 Conclusions and future steps

This model will be able to recommend the best action to perform to an HCP at the right moment and with optimal content. So it will provide a lot of help to the sales reps when having to organize their week or even their decisions throughout the week. Also, as they will be impacting the HCPs more optimally, this will translate into an improvement in the engagement of the client towards the company.

As seen in section 4.2.2, the bottleneck in the performance of the model is the genetic algorithm and trying all the combinations of the mother sequence. The genetic algorithm part, even though it might be the slowest, is also one of the most important parts of the whole model, as it optimizes the different sequences, this is due to the way the sequences are made, as when the context is not seen before the action is chosen randomly. So by using the genetic algorithm and rearranging the actions, the sequence is improved.

As future steps, to improve the preprocessing steps, it would be worth looking into incorporating co-visitiation matrices. These matrices are explained here and were developed by @vslaykovsky. These are based on close actions that happened within the same session, and give you an idea on the popularity of each action after another.

As seen in 15, we the model performed better in with the weightNoTrain being 0. A new way to take into account these actions performed by sales reps of the other products would be interesting to look at. Also, another thing that we want to implement is to improve the decision when the context was never seen before. Now when the context is never seen, the decision is made randomly, so unoptimal decisions may be done. So what we propose to do is to calculate the similarity between the different contexts with a decaying function implemented, so the further the action was done from the present the less it will influence the decision. This will help the model make more improved decisions.

5.1 Proof of Concept

Regarding the Proof of Concept, is another evaluation of the model to see if it translates to the real world and also see if it makes good predictions. The objective of the PoC is to improve the CTR of the HCPs we are impacting with the model. Before the PoC starts, the CTR of the HCPs is 3% in emails from the central office and 7% in follow-up emails. So we are looking to increase these numbers, which would mean that the HCP are engaging more with the company. Thanks to another project that is being developed, it is showing that more engagement is being translated into more sales, so by improving the HCPs engagement we will be also impacting the sales.

For this PoC, the sales reps of the company will follow the recommendations from the model and give feedback about them. These sales reps are from 2 different territories, which are from two different parts of Spain. Also, another thing we should take into account, is that each sales rep is visiting a territory, and in these territories, there are around 300 HCPs. So the model would be giving around 300 recommendations per week to each sales rep. As this amount of recommendations is not manageable by the sales reps, a selection algorithm was made to choose which HCP the sales reps receive a recommendation of. This selection algorithm has several parameters which are:

- The amount of recommendations we want to send, this value will vary depending on how many recommendations the sales rep wants to receive that week
- The fraction of "Alto Potencial"(AP), which are the high potential HCPs, which will be in

this week's recommendations.

- These week's recommendations are the recommendations given by the MCMC algorithm.
- Last week's recommendations were sent to the sales rep.
- Lastly the overall recommendations are given in the cycle. These last 2 parameters are useful as we want to impact all the different HCPs with a certain frequency, and also we don't want to repeat last weeks recommendations.

We want the sample of recommendation to represent the real amount of recommendations, so first we are going to divide the number of recommendations into 2 groups, the amount of recommendations we want to make to the AP group, and the amount of recommendations we want to make to the other group. Also, we want to maintain the proportion of channels equal to the real one, so before choosing the recommendations we also measure the proportion of each channel and calculate how many recommendations we need for each channel. Also, it is worth mentioning that before this we remove the recommendations we can't make, which are the BLACKOUT weeks, as it is a week where no interaction with the HCP will be made, the Event, as the HCP will be in an event. Another thing we need to take into account is that the sales rep has 2 weeks to perform a F2F meeting, so if last week we recommended making a F2F meeting to an HCP, the following week we cannot make a recommendation to that HCP. Lastly, we want to recommend to all the HCPs, so we will take that into account at the moment of choosing which HCPs were recommended last week, as well as how many recommendations we made overall in that cycle to each HCP. After taking these facts into account, the algorithm chooses which are the recommendations that are going to be made that week, and then we are going to proceed to choose the content to send or show to each HCP.

Regarding the results for this PoC, they are still not available as it started in cycle 2 2024, May 2024, therefore it is still ongoing. So no results will be available until September 2024, which is the date when the PoC will end. Once the PoC has ended we are going to evaluate the how it went, looking at if we improved the engagement of the different HCP towards us. Also another important point is to get the feedback of the sales reps and to know how they felt using the suggestions the model gave them, as this is an important factor as they are the ones that are going to use the tool.

References

[AHS85] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[Ama24] Amazon Web Services. Amazon athena documentation: When should i use amazon athena?, 2024. Accessed on April 14, 2024.

[Amand] Amazon Web Services. What is monte carlo simulation?, n.d. Accessed on April 14, 2024.

[CESnd] CESIF. Perfil profesional del medical scientific liaison (msl), n.d. Accessed on April 14, 2024.

[fe23] Industria farmaceutica española. Anexo iii - guía de actuación en comunicación y relaciones con los medios sobre medicamentos de prescripción, 2023.

[Gen24] Generalitat de Catalunya. Guia de publicitat de medicaments 2024, 2024. Accessed on April 14, 2024.

[Gow14] Stephen Gower. Netflix prize and svd. http://buzzard.ups.edu/courses/2014spring/420projects/math420-UPS-spring-2014-gower-netflix-SVD.pdf, 2014.

[Hay13] Brian Hayes. First links in the markov chain. *American Scientist*, 2013. Accessed on April 7, 2024.

[Hol75] John Holland. Genetic Algorithms. *Scientific American*, 1975.

[IQV23] IQVIA. Choosing the right channel for healthcare professional engagement, 2023. Accessed on April 14, 2024.

[IRCnd] IRC Sales Solutions. Sales follow-up statistics, n.d. Accessed on April 14, 2024.

[Mah21] Fariha Mahfuz. Markov chains and their applications, 2021.

[Mar] Marketing Evolution. What is multi-channel marketing? Accessed on April 14, 2024.

[Mednd] Medical Affairs Academy. Why face-to-face meetings are important for medical affairs, n.d. Accessed on April 14, 2024.

[Mur03] Krishnamurty Muralidhar. Monte carlo simulation. In Hossein Bidgoli, editor, *Encyclopedia of Information Systems*. Elsevier, 2003.

[Natnd] National Institute on Drug Abuse. Overview of misuse of prescription drugs, n.d. Accessed on April 14, 2024.

[Org13] World Health Organization. Definition and list of health professionals. https://www.ncbi.nlm.nih.gov/books/NBK298950/, 2013.

[Sal16] Tim Salimans. Improved Techniques for Training GANs. *Advances in Neural Information Processing Systems*, 2016.

[SAS] SAS. Multichannel marketing. Accessed on April 14, 2024.

[Wan19] Alex Wang. GLUE: A MULTI-TASK BENCHMARK AND ANALYSIS PLATFORM FOR NATURAL LANGUAGE UNDERSTANDING. *arXiv preprint arXiv*, 2019.

[Wiknd] Wikipedia. Markov chain, n.d. Accessed on April 14, 2024.

[ZHB+19] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.