End-of-study internship
Machine learning engineer intern

Internship report

Student :
Quoc-Hung TRAN
Contact :
quoc-hung.tran@etu.utc.fr

Company supervisor :
Reda CHAOUI
Academic supervisor :
Mokhtar ALAYA

Crtieo SA
32 Rue Blanche, 75009 Paris, France

09/2022 - 02/2023

Remerciements

Mon stage n'aurait pas été un grand succès sans le soutien de nombreuses personnes. Tout d'abord, je tiens à exprimer ma plus profonde gratitude à mon tuteur de stage, M. Reda Chaoui, pour ses excellents conseils et sa confiance sur de nombreux points au cours de mon stage. J'ai toujours été étonné par son souci sincère pour la réussite de mon stage et en effet, je n'ai été témoin d'aucune situation où il n'a pas rapidement cherché à m'aider. Je suis également reconnaissant pour tous ses précieux conseils non seulement pour mon stage mais aussi pour les domaines de développement de ma future carrière.

Je voudrais profiter de cette occasion pour exprimer mes remerciements à l'équipe de Delivery control, M. Jean-Denis Lesage, M.Nicolas Perrin, M.Hoang Do, M. Yacine Ben Baccar, M.Hai Nam Nguyen, M.Valerian Gonnot et Mme Colombe de Milly, pour m'avoir chaleureusement accueilli au sein de l'équipe pour ses commentaires détaillés sur mes revues de code et pour m'avoir aidé à devenir plus compétent dans les pratiques techniques utiles.

D'ailleurs, mon projet de stage aurait beaucoup moins de succès sans le soutien de nombreux développeurs de mon équipe, qui ont participé à toutes mes démos et m'ont apporté d'innombrables retours significatifs pour m'aider à améliorer considérablement mon projet. Je voudrais également profiter de cette occasion pour exprimer ma gratitude envers tous les employés et stagiaires de Criteo pour cette expérience unique. Enfin, je tiens à remercier Madame Jade Dong et l'Université de Technologie de Compiègne pour avoir facilité et fourni la structure nécessaire pour que les étudiants puissent avoir une première expérience dans un cadre professionnel.

Sommaire

Résumé

En tant que stagiaire chez Criteo, j'ai eu l'occasion de travailler sur un projet d'apprentissage automatique visant à améliorer le modèle de prévision des campagnes de l'entreprise. Le modèle de prévision joue un rôle essentiel dans la stratégie publicitaire de Criteo, car il permet d'optimiser l'allocation des ressources et d'améliorer la performance globale, qui visait à configurer les campagnes de manière autonome et à prédire l'impact des changements de configuration.

L'objectif principal de ce rapport est de fournir une compréhension complète du système de prévision des campagnes, y compris son état actuel, les défis auxquels l'équipe est confrontée et les suggestions sur la façon de relever ces défis à l'avenir. Il s'agit notamment de fournir des informations sur la méthodologie d'amélioration du modèle d'apprentissage automatique, ainsi qu'un aperçu de l'impact du système sur la stratégie publicitaire de l'entreprise. Le rapport vise à aider les lecteurs à comprendre la qualité du service de prévision et la façon dont il peut être amélioré afin d'optimiser l'allocation des ressources et d'améliorer la performance globale. Au cours de mon stage, j'ai travaillé au sein de l'équipe chargée des outils du service de prévision des campagnes et j'ai collaboré avec les équipes internes qui ont exprimé un besoin pour ces informations.

Ce document a pour but de résumer mon expérience de stage et est organisé comme suit : une présentation de l'entreprise et de l'équipe Delivery Control, une description du stage et un rapport sur mon expérience.

1 Enterprise and team introduction

1.1 Enterprise Presentation

In the beginning of the report, it would be beneficial to provide a high-level overview of the company's business, including its industry and main products or services.

1.1.1 Criteo SA

Criteo SA is an **advertising company** founded in 2005 in Paris, focused on delivering trusted and impactful advertising. Over its lifetime, from a small group of great minds at a start-up incubator Criteo has become a global leader in commerce marketing. To sastify its thriving range of products and clientle, Criteo's size has grown exponentially, reaching over 2.5 bilions users with nearly 4 billion ads served per day, supporter by more than 2500 emoployees from 27 offices worldwide.

As Criteo has expanded internationally, the company became a culture melting pot of rich diversity in gender, race and ethnicity. Indeed, creating an inclusive workspace is not an easy thing to check off. Thus, Criteo is making efforts every day to bring its employees worldwide together and embrace a set of core values : **Open, Together, and Impactful.**

1.1.2 Product and Technologies

Over the past decade, Advertising Technology (**AdTech**) has become a significant factor boosting the massive explosion in e-Commerce. To help Criteo's partners such as online marketers, retailers, and brands join this digital race, the company offers several products mainly centering around displaying ads to spread product or brand awareness to Internet users and thus pump up sales and revenue. Nowadays, the two most appreciable products that make Criteo become a pioneer in applying cutting-edge technologies to digital marketing are :

- **Customer Acquisition** : drives new customer conversions through powerful machine learning algorithms to identify new customers from existing ones based on their shopping patterns, browsing journeys, and individual interests.

- **Dynamic Retargeting** : brings online shoppers back to buy by delivering personalized ads at the right moment in their shopper journey based on real-time intent.

Most popular web browsers such as Chrome or Firefox support third-party data in recent years. The main goal of this within the AdTech environment is to assign each user an ID. These IDs allow Criteo to track user's data (for instance, shopping patterns, browsing history, and individual interests). In reality, user data plays a vital role in feeding Criteo's advanced machine learning models to create unique personalized ads for each user and ensure their highest engagement to Criteo's partner's products. As a global leader in

AdTech, Criteo is making efforts to develop and continuously improve its AI engine that comprises the following components :

- **Dynamic Creative Optimization** : creates the best visual elements for each ad display.

- **Product Recommendations** : selects products to displays in each ad that most likely to drive a customer to buy.

- **Predictive Bidding** : bids on the right ads at the right time with the highest probability for raising product awareness.

Overall, Criteo has a full funnel marketing strategy approach and works on 3 main states :

- **Awareness** : generating awareness among new audiences who may not be familiar with the brand, to influence their consideration.

- **Consideration** : getting people to think about the target brand rather than other competitors.

- **Conversion** : encouraging actions, such as sales, user visits or other forms of action that show user interest in a particular brand or product line.

1.1.3 The future of Advertising

As previously stated, digital advertisers rely on data to personalize and optimize ad displays. However, a new future of personalized advertising is on the cusp of dramatically changing the industry since tech giants such as Google and Apple announce the demise of third-party cookies, IDFA (the identifier for advertising) restrictions to protect user data privacy in 2022. Criteo is developing and optimizing a stable and privacy-driven solution that delivers results comparable to today's cookie-based one named Contextual Targeting to take on these concerning challenges. The main idea is to use Machine Learning and Natural Language Processing technologies to analyze texts, images, and videos to understand every web page's context and sentiment deeply. Then, the Criteo engine will select ads to display based on product affinity scores between shoppers' interests and these webpage insights.

1.2 Team Presentation

1.2.1 Delivery Control

To be honest, I am very glad to work with Delivery Control team which include Machine Learning engineer, Software Engineer and Data Scienctist staffs, during 6 months

of internship.

Delivery Control is in charge to control the volume of the ad campaigns. The team implements systems that control the spend and performance of all campaigns. We developed a system that paces the budget along a period (For example spend 100$ uniformely on a week). And a system that controls the performance (average cost of a clicks, sales, etc.)

To implement the systems, we have some feedback loops that observes the performance spend and modify the bid to impact the volume. As an extension of this scope, the team is also in charge of the forecasting. This tool predicts the performance of a ad campaign given a configuration. This tool is used to help clients to configure their campaign autonomously and predict the impacts of a configuration change or special events (such as Black Friday, sales periods etc.) The team is composed of 7 engineers (Machine Learning Software developers). They work in collaboration with the datascience team of Criteo. Main technos are spark, scala, python to write computatoin jobs. C and .netcore to build the API that will serve models.

1.2.2 Working Enviroment

The Delivery control team uses the **Scrum** framework to help the team work together. Ideally, the Scrum process is based on iterative cycles called Sprints which typically last two weeks. During each Sprint, the product is designed, coded, and tested while meeting every day to assess its progress (daily **stand-up**). The goals and tasks of each Sprint are determined in the Sprint **Planning** session. At the end of each Sprint, a **Restrospective** meeting is held to discuss what went well during the previous sprint cycle.

At Criteo, we use **Atlassian** tools (**Jira**, **Confluence**, etc.) to assign tasks between the developers. **Jira** is helping to schedule the work, while Confluence is a team workspace used to share knowledge and foster collaboration.

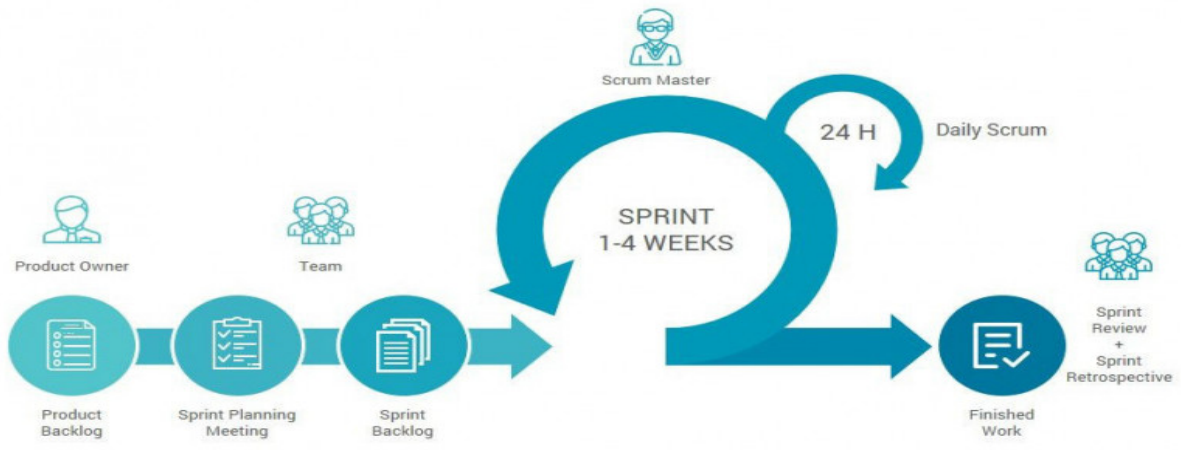

FIGURE 1.2 – Scrum Ceremonies

1.2.3 Technologies

Criteo deals with a massive volume of data with billions of unique adverts displayed at lightning- fast speed every day. Therefore, the company has one of the **largest Hadoop clusters** in Europe, with close to 171 Peta Bytes of storage and 42.000 cores. The two most common **SQL query engines** used to process data from clusters are **Presto** and **Hive**. The Metrics team also uses data analysis tools such as **Spark**, **Pandas**, **Numpy**, etc., to manipulate data and generate data analysis reports.

Additionally, to access Criteo's data, we need to use **virtual servers** or machines called **Mozart** that is free to use for everyone at Criteo. Depending on our needs, employees can connect to these machines remotely with various resources from small instances with 2 CPUs and 15GB of RAM up to 30 CPUs and 880 GB of RAM. Thanks to these virtual machines, Criteo's employees can flawlessly keep up their work and connect to the company's data remotely.

Besides, the three most commonly used programming languages in the Delivery Control team are **Python** (for data analysis), **C#** and **Scala**(for forecasting API) . The codebases are managed and served by Gerrit. With **Gerrit**, developers in the team can review each other's modifications on their source code using a Web browser and approve or reject those changes. It integrates closely with Git, a distributed version control system. At Criteo, we use the Jenkin automation server to help automate the parts of software development related to building, testing, deploying, and facilitating continuous integration. This tool helps maintain the robustness of the codebase ; for instance, the modifications that fail unit tests will not be allowed to merge to the codebase.

2 Context and Internship Topic

2.1 Motivation

First of all, it is essential to dive into the whole marketing service picture and fundamental aspects of my internship topic.

Criteo is a company that provides online advertising and marketing services, including targeted display advertising, retargeting, and personalized product recommendations. It is primarily focused on the e-commerce sector, and its goal is to help businesses drive sales and increase revenue through the use of data-driven advertising campaigns.

In terms of the specific campaigns that Criteo offers, there are several options available. These include :

1. **Display advertising** : Criteo offers targeted display advertising to help businesses reach the right audiences at the right time. This includes banner ads, video ads, and other types of visual ads that are displayed on websites and apps.

2. **Retargeting** : Criteo's retargeting campaigns are designed to help businesses bring back lost customers who have visited their website but haven't made a purchase. This is done by showing targeted ads to these users as they browse other websites, in an effort to remind them of the products or services they were interested in.

3. **Personalized product recommendations** : Criteo offers personalized product recommendations to help businesses increase the likelihood of making a sale. This is done by using data to show users products or services that are most relevant to their interests and needs.

Overall**, Criteo's campaigns** are designed to help businesses reach the right audiences with the right message, at the right time, in an effort to drive sales and increase revenue. In the context of online advertising, there are principals components which form a advertising campaign :

2.1.1 Campaign features

Advertiser, Publishers and Verticals

Advertisers are the main driving force behind campaigns, as they are the ones who create the campaigns and pay for ad placements. Advertisers typically have a specific goal in mind when creating a campaign, such as increasing brand awareness or driving sales. **Publishers** are a key part of the campaign equation, as they provide the platform on which the ads will be displayed. Publishers typically work with advertisers to find the best ad placements and formats to help the advertiser achieve their goals. **Verticals** play a role in campaigns by helping advertisers to identify the specific industry or market segment they want to target with their ads. For example, if an advertiser is selling fashion products,

they might create a campaign targeting the fashion vertical, while a travel vertical would include businesses that offer travel services, such as airlines and hotels.

Audience size and targeting options

Furthemore, the **audience size** is also an important feature of Criteo adverising campaign. The audience size refers to the number of people that the campaign is targeting. This number can vary widely depending on the specific goals of the campaign and the audience type that has been identified.

Criteo uses advanced algorithms and machine learning techniques to analyze data about users' online behavior in order to identify and target specific audience segments. This allows advertisers to reach the people who are most likely to be interested in their products or services, which can improve the effectiveness of their campaigns.

The size of the audience for a Criteo campaign will depend on the specific **targeting criteria** that are used. For example, if an advertiser is targeting a very specific and narrow audience segment, such as men between the ages of 25 and 34 who live in a certain geographic region and have an interest in a particular product or service, the audience size may be relatively small. On the other hand, if the targeting criteria are broader, such as all individuals who visit a particular website or use a specific type of device, the audience size may be much larger.

By carefully selecting the audience size and targeting criteria for their campaigns, advertisers can optimize their efforts to reach the right people at the right time, which can help to improve the effectiveness and results of their campaigns.

Business model

The problem is how the company generate revenue and profits from its advertising feasibly. Criteo 's **bussiness model** is the answer. It is an important aspect of a company's operations because it determines how the company will make money and sustain itself over the long term. By offering its campaigns on a pay-per-click or pay-per-action basis, Criteo is able to ensure that its services are financially viable for both the company and its clients. Additionally, the business model helps Criteo to focus on its core competencies and ensure that it is able to deliver value to its clients in a way that is sustainable over the long term.

In the context of online advertising, a business model is the understanding of the advertiser's business objectives and the terms of the agreement between the advertiser and the advertising platform (the "engine"). The business model specifies three key elements :

- **The value to optimize** : This refers to the specific goal that the advertiser is trying to achieve through the advertising campaign. This could be things like increasing sales, driving website traffic, or raising brand awareness.

- **The client constraint** : This refers to any limitations or restrictions that the advertiser has placed on the campaign. This could include things like budget constraints, target audience parameters, or specific metrics that the campaign needs to meet.

- **The billing unit** : This refers to the way that the advertiser will be charged for the advertising campaign. This could be on a pay-per-click basis, where the advertiser is only charged when a user clicks on an ad, or it could be on a pay-per-action basis, where the advertiser is only charged when a user takes a specific action (such as making a purchase).

Currently, the client constraint and value to optimize are enough to define a business model in online advertising, but this could change in the future as the industry evolves. The figure below show you the bussiness model correspoding to 2 citerias aboves :

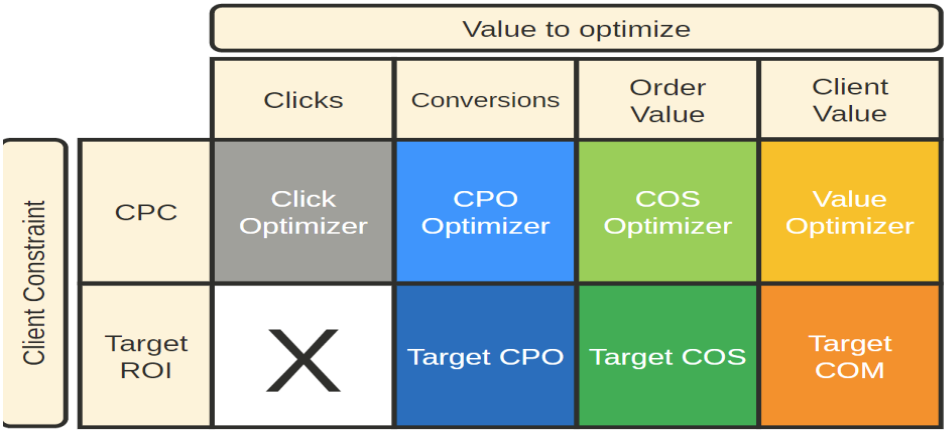

FIGURE 2.1 – Bussiness model

Each specific bussiness model that is used in an online advertising campaign will depend on the specific goals and objectives of the advertiser. Here is a brief explaination of the various business model terms :

Click Optimizer	A click optimizer business model is one in which the advertiser is charged based on the number of clicks that their ads receive. This model is often used when the advertiser's main goal is to drive traffic to their website or other online properties.
CPO Optimizer	CPO stands for "cost per order," and a CPO optimizer business model is one in which the advertiser is charged based on the number of orders that are placed through their ads. This model is often used when the advertiser's main goal is to drive sales or revenue.
COS Optimizer	COS stands for "cost of sale," and a COS optimizer business model is one in which the advertiser is charged based on the cost of the products or services that are sold through their ads. This model is often used when the advertiser's main goal is to drive sales or revenue, and the products or services being advertised have a high cost or margin.

In addition, Criteo have launched new bussiness model known as **AO (Adaptive Optimization)**. The goal of the feature is to offer our clients more convenience in managing their campaigns. Indeed, the feature gives to our clients the ability to directly edit their COS or CPO target in CPP and let our Engine automatically adapt CPCs on their behalf to reach their target. The feature is to be presented as an automated bidder (on behalf of the Client) : once the target (COS or CPO) is edited, the client doesn't need to do anything else to manage the campaign. There are two different business models depending on the client target :

- **Adaptive Revenue Optimization (ARO)** : optimises based on the client COS target.

- **Adaptive Conversion Optimization (ACO)** : optimises based on the client CPO target.

AO business model is used widely by big client that brings high-value to Criteo.

2.1.2 Forecasting

To deal with the growing demand from client, Criteo's campaign forecasting service would be a trusted and powerful tool that helps advertisers to estimate the expected reach and performance of their advertising campaigns with given features I mentioned above. This can be useful for a variety of purposes, including planning and budgeting, performance optimization, and campaign measurement.

- **Planning and budgeting** : Campaign forecasting helps advertisers to plan and budget for their campaigns by providing estimates of the expected reach and performance

of their ads. This can help advertisers to allocate their resources effectively and ensure that they have the necessary resources to meet their goals.

- **Performance optimization** : By forecasting the expected performance of their campaigns, advertisers can identify areas where they may need to make adjustments in order to optimize their results. For example, if a campaign is not performing as expected, advertisers can use the forecast to identify potential issues and make changes to improve the performance of the campaign.

- **Campaign measurement** : Campaign forecasting can also help advertisers to measure the success of their campaigns by comparing the actual results to the forecasted results. This can help advertisers to identify areas where their campaigns are performing well and areas where they may need to make improvements.

Overall, campaign forecasting is an important tool for advertisers to use in order to better understand the expected performance of their campaigns and make informed decisions about how to allocate their resources and optimize their efforts.

2.2 How the forecast done

Indeed, we adopted two methodology for forecasting in term of existing campaign and the new ones .

2.2.1 Exisiting campaign

An existing campaign in Criteo refers to a advertising campaign that has already been set up and is currently running. This could be a campaign that is being managed through the Criteo platform, which is a technology company that provides digital advertising solutions to businesses.

Forcasting existing uses primarily the historical features features flowed by time, beacause with some components :

- **Baseline** : The average performance over the last 4 weeks. For an existing campaign, the average number of displays in the past 28 days.

- **Seasonality factor** : the regular fluctuations that occur in business or economic activity over the course of a year. These fluctuations can be caused by a variety of factors, such as holidays, weather patterns, and cultural or societal events. In the context of an existing campaign, seasonality can have a significant impact on the performance of the campaign. For example, if the campaign is promoting a product or service that is particularly popular during a certain time of year, such as holiday gifts or summer vacation items, the campaign may experience a boost in performance during that time. On the other hand, if the campaign is promoting a product or service that is less popular during a certain time of year, such as winter coats in the summer, the campaign may experience a

drop in performance. Understanding and taking into account seasonality can be important for advertisers when planning and managing their campaigns.

We utilized this formula below :

$$\text{predict(metric, t)} = \text{avg\_28d(metric, } t_0) * \frac{seasonality\_factor(metric,t)}{seasonality\_factor\_avg\_28d(metric,t_0)}$$

t_0 is the latest time campaign lives, t is the future day ($t > t_0$)

The metric here is several KPIs such as the predicted number of displays, clicks, sales, that we would like optimize given existing campaigns settings. The performance of this forcasting formula is shonw in the section *here*.

2.2.2 New campaign

When it comes to this case, a machine learning algorithm was applied to new campaign forecasting in order to improve the accuracy and effectiveness of the displays forecasts. Particularly, we use a **Cascade model** to estimate other KPIs to mesure the success of a campaign, include a variety of metrics, such as

- All the metrics were based on displays predicted by **XGboost**. The number of displays is the first brick of cascade model.

$$Displays = XGBoost(F)$$

where $F = [X_1, X_2, ..., X_n]$ the set of features used to train the model

- Clicks : The number of times users click on an ad or link as a result of a campaign. This KPI can help to measure the engagement and interest of users in the campaign, computed in term of displays metric and CTR (click through rate) country benchmark value.

$$Clicks = Displays * CTR$$

- Visits : The number of times users visit a website or page as a result of a campaign, computed in term of Clicks metric and LR (Landing rate) country benchmark value.

$$Visits = Clicks * LR$$

- Sales : The number of purchases or conversions that are made as a result of a campaign. This KPI is often used to measure the effectiveness of a campaign in terms of generating revenue or achieving a desired action, computed in term of Visits metric and CR (Conversion rate) partner benchmark value.

$$Sales = Visits * CR$$

- OrderValues : computed in term of Sales metric and AOV (Average of values) partner benchmark value.

$$OrderValue = Sales * AOV$$

2.3 Internship Topic

Considering the context that Criteo wants to help its clients plan and optimize their campaigns in self-service : clients should be able to identify the best settings and inputs for their Ad Sets and make them live. Indeed in a near future we plan to show the forecasts directly in Cockpit (homepage of Management Center). **Forecasting tools** can be used to predict the expected performance of advertising campaigns based on a variety of input criteria, such as budget, displays, clicks, which can help advertisers to plan and budget for their campaigns, optimize their efforts to achieve better results, and measure the success of their campaigns. Therefore, we should ensure to have forecasts matching product expectations.

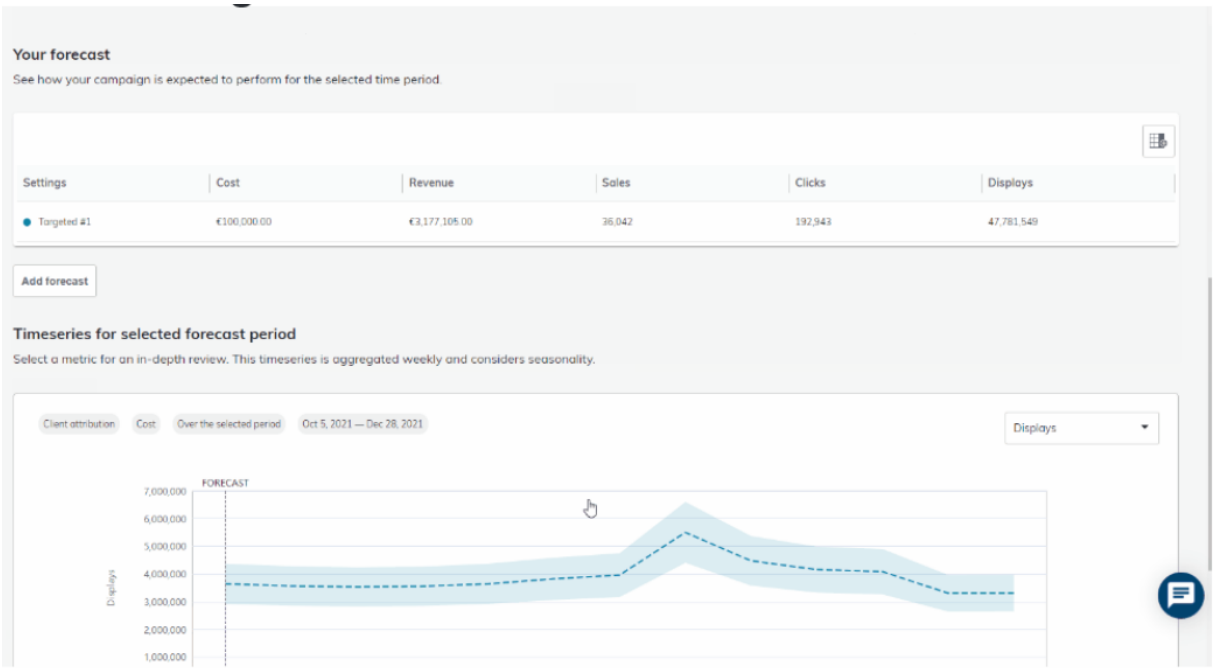

FIGURE 2.2 – Forecasting tool

My end-to-end machine learning project would involve a number of steps and activities, aim at improving the accuracy of campaign forecasting models on the **new campaigns scope** and standardizing the way we improve the models and define best-practices :

- Start with an offline analysis (Notebook)

- Demo the results to the team (R&D / PM / PAX)

- Decide if it should be added in prod

- Follow-up on live results using the dashboard and debug if needed

3 Organizational Aspect

3.1 Internship Objectives and Planning

During the third quarter of the year, a baseline model was developed in order to measure the performance of the machine learning project. However, a recent investigation revealed the need to not only focus on the accuracy of the model, but also on ensuring that the forecasts align with product expectations. As such, it is essential to make the model trustworthy in this regard in order to gain the confidence of our clients. To address this, my mentor created the "Improving Performance of Forecasting Model" page, which outlines a plan and goal for the future. This plan includes several prioritized tasks, such as data collection and preprocessing, model training, evaluation, deployment, and monitoring.

At the beginning of my internship, I was given the opportunity to familiarize myself with the team's technologies and processes. This included onboarding sessions, installation of the working environment, and exercises to help me understand the overall procedure for offline jobs, as well as how to collect data from various sources and organize notebook structures. I also had the chance to sync with team members and learn about the codebase, including how the forecasting model is deployed in real time, how to use Gerrit efficiently, and how to debug and implement various tasks.

In parallel with these tasks, I have spent time writing documentation for each task. This has allowed me to track progress and identify any potential issues more easily, improving project management and ensuring that the project stays on track. Additionally, this documentation has helped to ensure that the work can be easily reproduced by other researchers, which is especially important for research projects where reproducibility is a key requirement. The following figure lists all tasks that I completed during my internship in the order in which they were completed :

FIGURE 3.1 – Internship's important milestones.

3.2 Development Techniques and Technologies

3.2.1 Development Techniques

Throughout the internship, I followed an iterative development technique ; more specifically, I iteratively did many user-driven demos with the Delivery Control teams to let them try my products and get live feedback during the demo time. Thanks to this method, I received countless meaningful feedback from users on my deliverables. In some

first demos, sometimes I felt pretty overwhelmed by a massive amount of users' feedback, but then I realized that this helped me significantly in terms of better understanding my work from the users' standpoint. Hence, I could address the feedback at the early stage of the development to make my products more user-friendly and thus, enable clients to use them in their actual use cases.

In addition, I adapted to the team practices, for example, code reviews, unit tests. The code review process at Criteo helps me receive helpful feedback from other developers on my code and thus gain many good tips to write maintainable code. Also, to merge a new module into the team git repository, I had to make sure that it was well tested and the test sets that I wrote need to be reviewed by the team. Despite these practices being prevalent in the software development domain, it takes a long time to become a master for all developers.

3.2.2 Technologies and Tools

At Criteo, I had exposure to several tools and technologies mainly used in Data Analysis.

Programming languages

- Python is the primary programming language that I used during my internship. This language is highly versatile that can be used in multiple domains from Data Science, Machine Learning to Web Development, Automation Scripting, etc. In particular, I used Python to write a Data Analysis library and build user interface in Jupyter Notebook.

- SQL query : Since Criteo datasets are stored in Hadoop clusters, I used Presto to query data. This is a high-performance and distributed SQL query engine for big data.

- C# : This object-Oriented language to create an API (Application Programming Interface) that allows other software systems to access your forecasting functionality. This can be done by creating a web service or by creating a standalone application that exposes its functionality through an API. Launching and debugging help me understand how forecasting service deploy in production.

- Pyspark : the Python interface to Apache Spark, which allows to use Spark's data processing capabilities in the Python applications. Pyspark is maily used to create machine learning models that can scale to handle very large datasets.

- Data Analysis libraries : I mainly used Python libraries Pandas and Numpy to manipulate data and generate Data analysis reports. I also use work with other open-source libraries such as Matplotlib and Seaborn for data visualization.

Technologies

- Mozart : Mozart let me create a virtual instanceof a Linux system, and its pre-installed virtal terminal manager, I usually connect to these machines remotely with various resources from xlarge instances with 30 CPUs, 60GB of Memory and 2.6 Gbps of bandwiths .Thanks to these virtual machines, It's straightforward to handle concurent jobs, to store also a large file queried from database and to have fast communication between multiple instances for transfering large amounts of data or require.

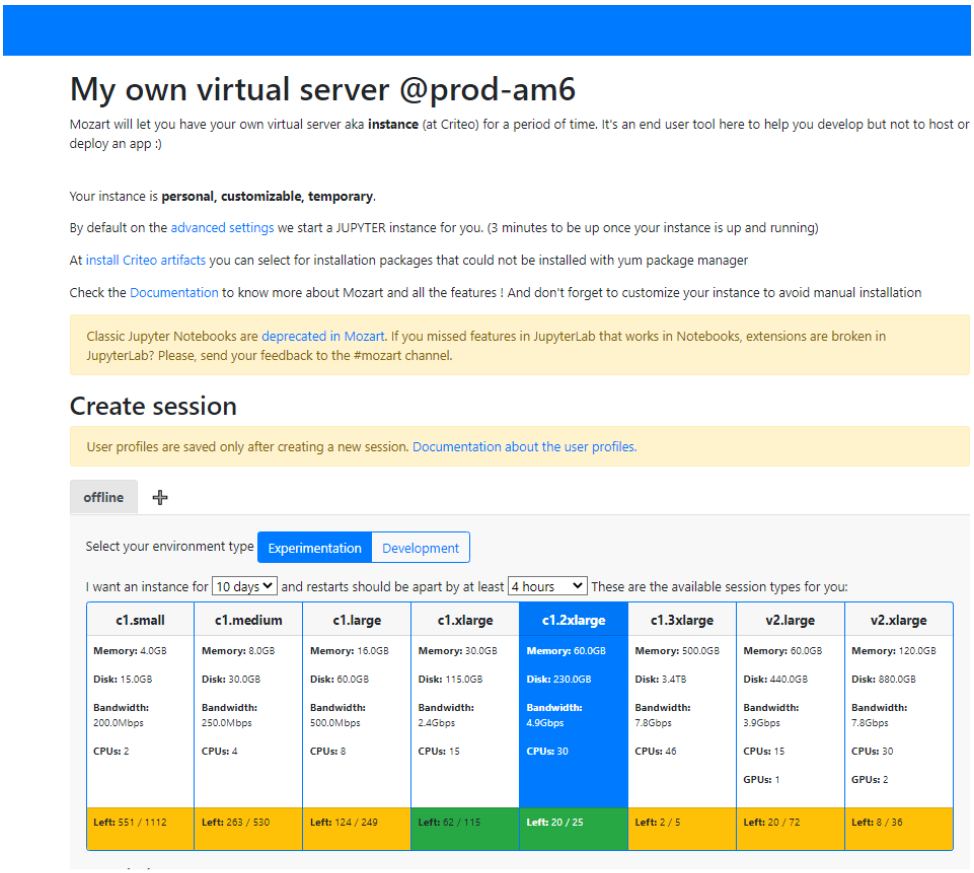

FIGURE 3.2 – Mozart

- MLFlow : open-source platform for managing the end-to-end machine learning life-cycle. It provides tools for tracking and managing machine learning experiments. It's very useful that it can help me keep track of the different variations of features that I have tried, and the corresponding results.

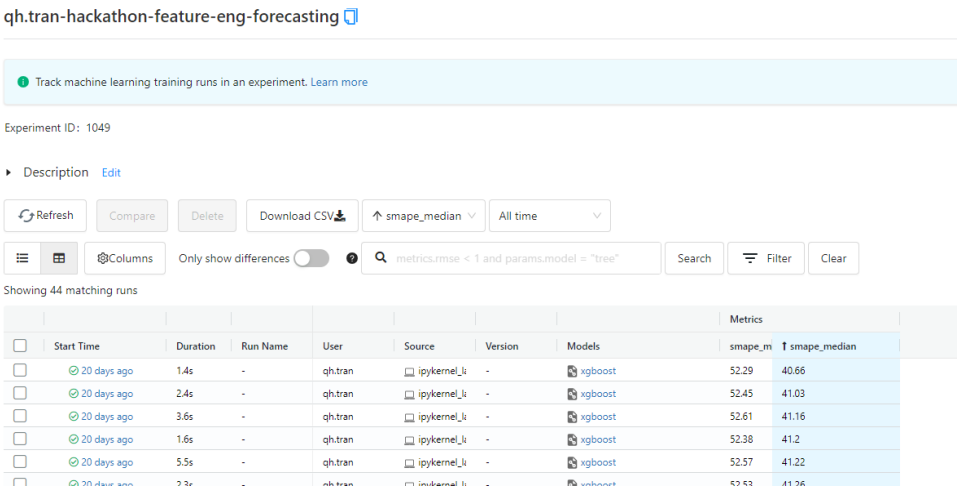

FIGURE 3.3 – MLFlow interface

- Hue : an open-source web-based interface for interacting with data stored in a Hadoop cluster. It allows users to manage and process data stored in HDFS (Hadoop Distributed File System) and other storage systems, as well as interact with a variety of Hadoop-related tools and technologies. I usually use Hue for evaluating SQL querying Databases before adding into the notebook.

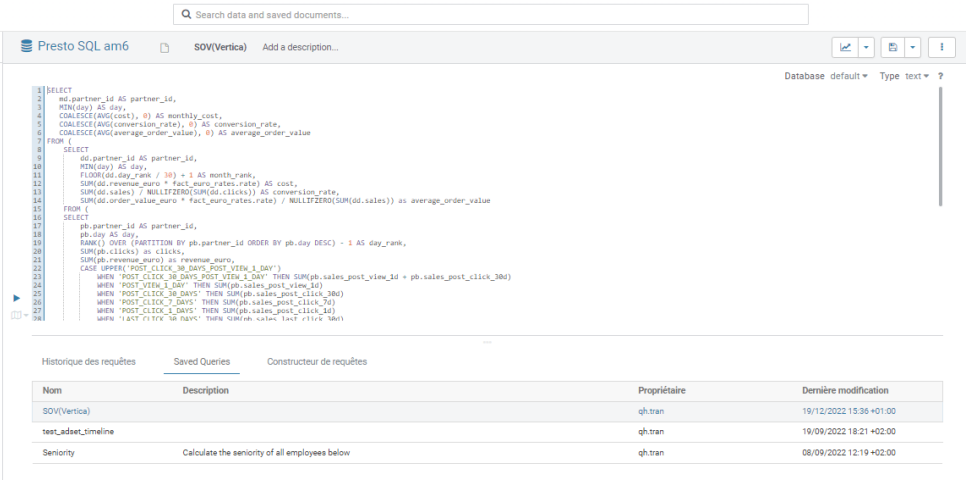

FIGURE 3.4 – Hue interface

- Postman : When working and debugging with forecasting API task, I usually use Postman. It's a popular tool for testing and interacting with Application Programming Interfaces (APIs). It is a web-based tool that allows users to send HTTP requests (such as GET, POST, PUT, and DELETE) to a server and view the response. Postman is often used for testing APIs during development to ensure that they are working correctly and to identify any issues that may need to be addressed.

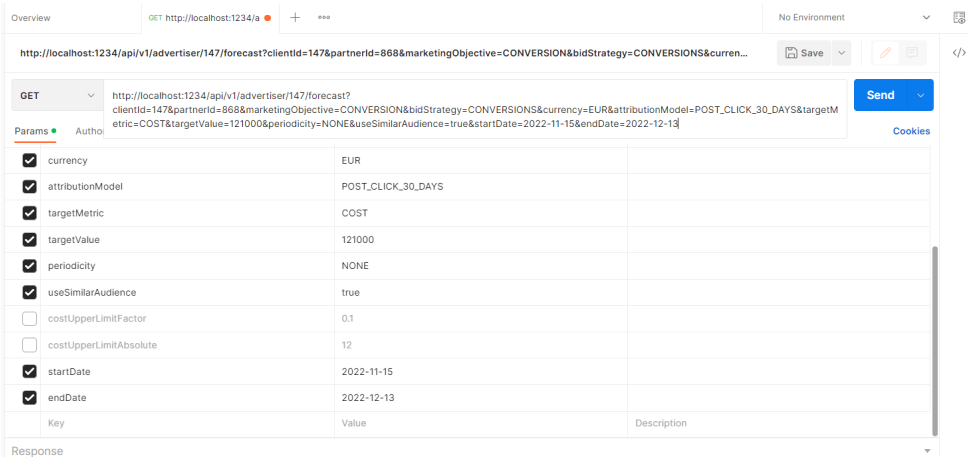

FIGURE 3.5 – Postman interface

3.2.3 Point of Contact

One of the most interesting facts when working at Criteo is that I had the opportunity to communicate with many people, especially from R&D teams. Reda Chaoui is my company supervisor and also my first point of contact when I needed helps in technical aspects. I had a 1-1 meeting with him twice a week to synchronize the progress of my internship and discuss some topics related to my work. Besides, I worked closely with M Nicolas Perrin - a machine learning staff of the Delivery Control team to ask for help if Reda is out of office. He also did code reviews for my work and gave me countless helpful feedback ; thus. Also, I met my team every morning as part of the Daily standup meeting of the Scrum methodology. This meeting was a time for me to share my current work with the team and seek help if I met any blockers.

Each Sprint, I have 1-1 meeting with M Jean-Denis Leasage, my manager about the work flow of the team, any blocked points and my task 's progress.

During the second part of my internship, I collaborated with M Louis Pruvot Carprioli- a data scientist to ask him about campaign's feature. Additionally, since I did many demos of my work with the Creator teams, I had a chance to meet many developers outside of the Metrics team to get helpful feedback for my work.

Beside, I have an apportunity to contact with Nam, Hoang and Colombe about software stuffs, deep more into how to compile the whole project in local machine, quality of code, etc

4 Enhance our data preprocessing pipeline

4.1 Context and objectives

Following the improvement in the performance of the forecasting model, the first priority is collecting and cleaning data. Data cleansing is critical because good results cannot be obtained from insufficient data, regardless of how sophisticated the ML algorithm is. There are many reasons to explain why data preprocessing is very important :

- Improving the quality of the data : Preprocessing can help to clean and normalize the data, which can improve the performance of the machine learning model.

- Reducing the volume of data : Preprocessing can help to filter out irrelevant or redundant data, which can reduce the resources required to run the machine learning model and improve its efficiency.

- Reducing the variability of the data : Preprocessing can help to transform the data into a more consistent format, which can improve the ability of the machine learning model to make accurate predictions.

- Improving model interpretability : Preprocessing can help to transform the data into a form that is easier for humans to understand, which can make it easier to understand how the machine learning model is making its predictions and identify any potential issues.

Current preprocessing data methods

Currenntly, we did several filtering rules such as :

- Cutting off the campaigns whose deliveried at least 500 displays which is a fix values.

- Filtering out the campaigns that are marked as being **contextual campaigns**.

- Filtering out the **video campaigns** : This means that any rows in the input data that represent campaigns that are classified as video campaigns will be removed from the dataset.

- Filtering **advanced segmentation campaigns** : This means that any rows in the input data that represent campaigns that have advanced segmentation enabled will be removed from the dataset. Advanced segmentation refers to the use of advanced targeting techniques to deliver ads to specific groups of users.

These campaigns are marked as outliers and were not covered in our scope because If they are not filtered out, they may have a disproportionate influence on the results of a machine learning model.

4.2 Collection Data

The dataset is collected by querying from the adset timeline table. For the first time, this dataset is a collection of all campaigns' information made in the last 28 days), CPX constraints(campaign COS, campaign CPM,...), etc., and a current day that the campaign lives. These are campaigns recorded in three months between July 01-08-2022 and 30-10-2022, containing 2843807 observations.

Data is typically stored in a pandas, which is built around DataFrame, a concept inspired by R's Data Frame, which is column-major. A DataFrame is a 2-dimensional tabular data structure with rows and columns.

	displays_avg28d	displays_min28d	displays_max28d	dayoftheweek	dayofmonth	month	daily_spend_strategy_smoothing_amount_euro	campaign_reve
0	5186.178571	448.0	21652.0	6	16	9	27.413784	
1	25805.928571	13803.0	39033.0	6	16	9	5.482757	
2	615495.142857	411233.0	780712.0	6	16	9	230.078403	
3	335.357143	44.0	541.0	6	16	9	4.112068	
4	6294.321429	2046.0	18822.0	6	16	9	NaN	

5 rows × 22 columns

FIGURE 4.1 – Training data

Depending on the dataset, different methods and procedures will be used to clean the data.

The idea here is to experiment with more intelligent filtering using quantiles and identify more complex outliers. For example, imagine we have a campaign with a nice budget and expect to deliver 100,000 displays without any problems. However, for some reason (e.g., BES), it ends up delivering only 3,000 displays. However, 3,000 would be above the threshold from quantile filtering, and we would then keep such campaigns.

4.3 Methodology

4.3.1 Check for duplicates in our logs

Firstly, it is essential to identify duplicates across the entire data set. Datasets that contain duplicates may contaminate the training data with the test data or vice versa. This step is help to visualize appropriately the distribution of each feature.

Forturnately, there are not duplicated values in our logs.

4.3.2 Data filtered with campaign in self service only

Some clients create their campaigns with advanced parameters on TOP (Criteo tool) and they don't use self-service, this means the set up of this campaigns is no the same as for self-service so we might want to filter them out.

We have about 3.5% not self -service campaigns filtered on the entire data.

After filtering out the first outlier campaigns above, now we would like to regard the methodology for **complex outliers**.

4.3.3 Complex outlier

As mentioned earlier, complex outliers in our dataset may represent unusual or unexpected events that fall outside the normal range of values for a specific business model. The current method of filtering our data set, which uses a fixed lower bound, may not be effective in identifying and removing these outliers. This is because the appropriate threshold for a campaign may vary depending on the business model, and a fixed threshold may not be suitable for all campaigns. In order to more accurately identify and filter complex outliers, it may be more effective to use a method such as calculating quantiles and the interquartile range (IQR).

Filtering methodology with IQR

First of all, We apply a **log transformation** to the label in order to normalize skewed data. Skewed data can be challenging for machine learning algorithms to accurately predict, but by using a log transformation, we can often make the data more symmetrical and improve the model's performance. Additionally, log transformations can make it easier to identify patterns and trends in the data, particularly when the values are very large or small. By bringing the values into a more manageable range, we can more clearly see patterns in the data.

As shown in the figure, the log transformation of the label (displays) does not follow a normal distribution. This can be seen in the log displays distribution and its Q-Q plot.

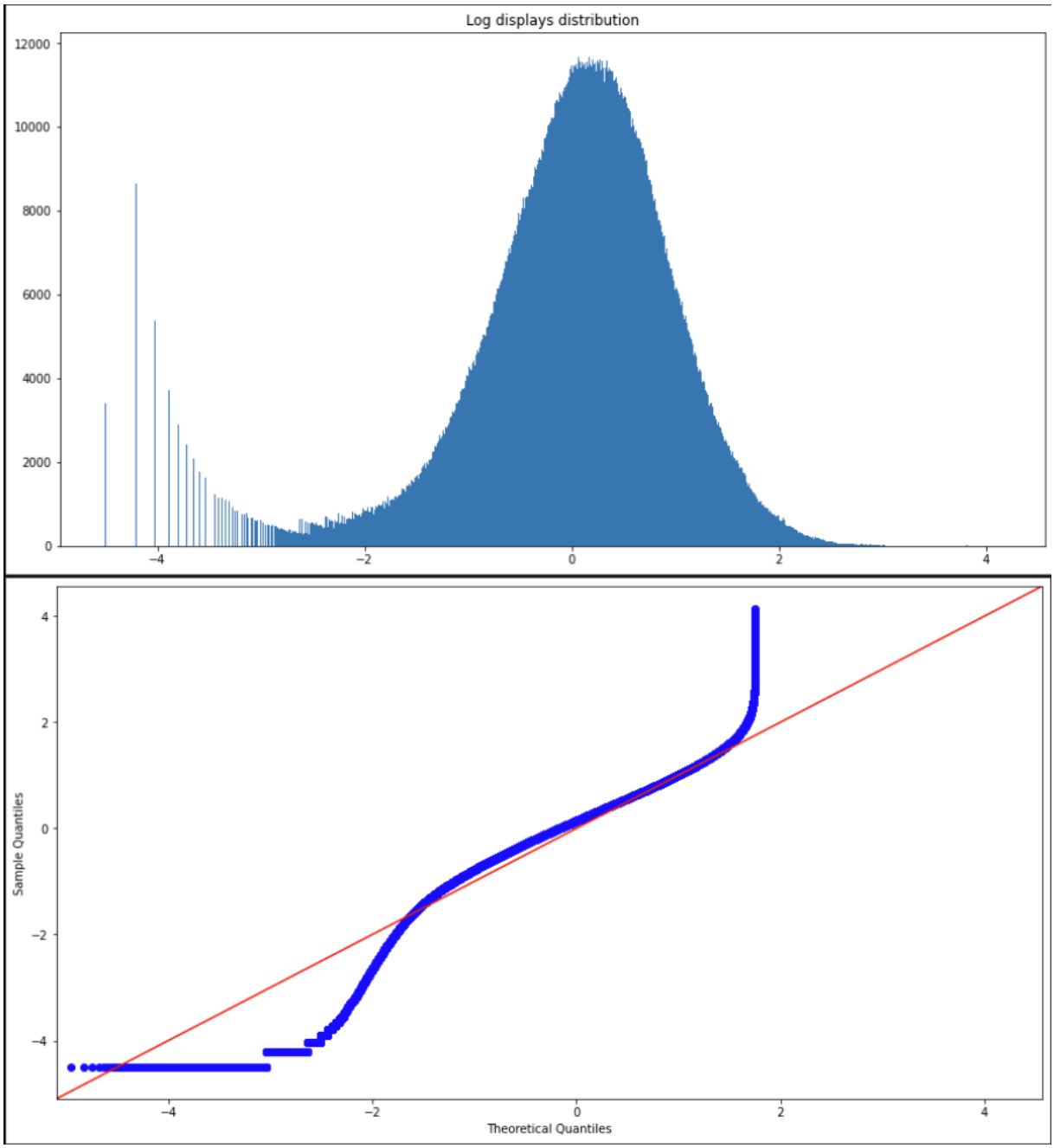

FIGURE 4.2 – Log Displays distribution

Therefore, It can be difficult to determine the lower and upper bounds based on quantiles alone. Therefore, we propose using a method called the Interquartile Range (IQR) to filter the data. The IQR is a measure of variability for skewed distributions or datasets with outliers. It is based on values from the middle half of the distribution, so it is less likely to be affected by outliers.

To understand the IQR method, consider a box plot :

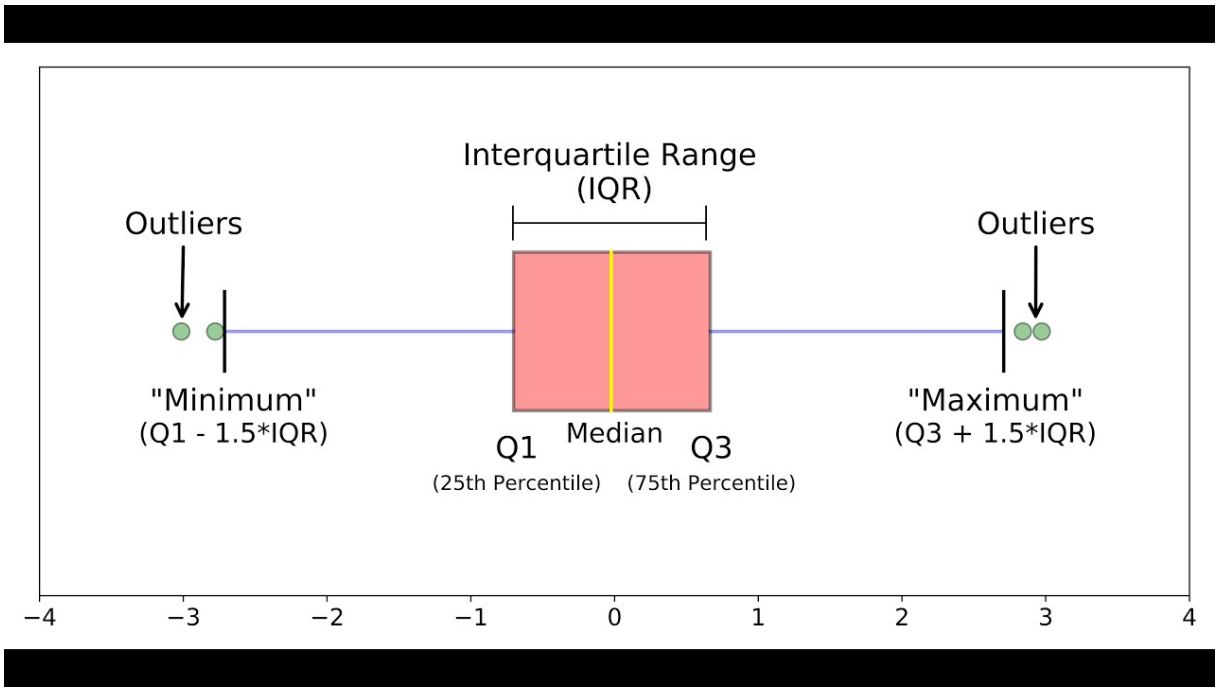

FIGURE 4.3 – Box plot

The median, or second quartile, is the center point of the data. The first quartile (Q1) represents the point at which 25% of the data lies between the minimum and Q1. The third quartile (Q3) represents the point at which 75% of the data lies between the minimum and Q3. The difference between Q3 and Q1 is the IQR.

$$IQR = Q3 - Q1$$

To detect outliers using this method, we define a range called the decision range. Any data point outside this range is considered an outlier. The lower bound is defined as $Q1 - 1.5 * IQR$, and the upper bound is defined as $Q3 + 1.5 * IQR$. Any data point less than the lower bound or more than the upper bound is considered an outlier.

For example, in this case, the IQR method filters data points as follows : if the number of displays value is less than $Q1 - 1.5 * IQR$, it is considered an outlier. If the number of displays value is greater than $Q3 + 1.5 * IQR$, it is also considered an outlier.

When scale is taken as 1.5, then according to IQR Method any data which lies beyond 2.7 from the mean (μ), on either side, shall be considered as outlier. A bigger scale would make the outlier(s) to be considered as data point(s) while a smaller one would make some of the data point(s) to be perceived as outlier(s).

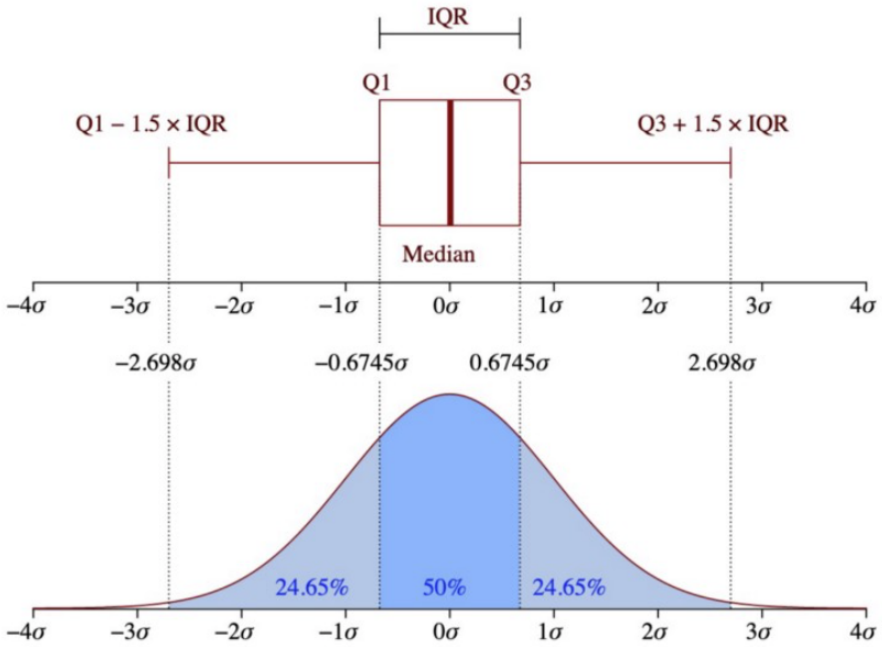

FIGURE 4.4 – Interpreting a box plot in relation to a histogram distribution

4.3.4 IQR on the whole data

Applying the IQR method to the entire dataset can be beneficial, but it is still important for the distribution to be normal. As shown in the Q-Q plot, the distribution tends to be linear but is not yet perfect across the entire dataset.

It is generally a good idea to have a normally distributed label (also known as the target variable or output variable) when using XGBoost, or any machine learning algorithm for that matter. This is because many algorithms assume that the label is normally distributed, and perform better when this assumption is satisfied.

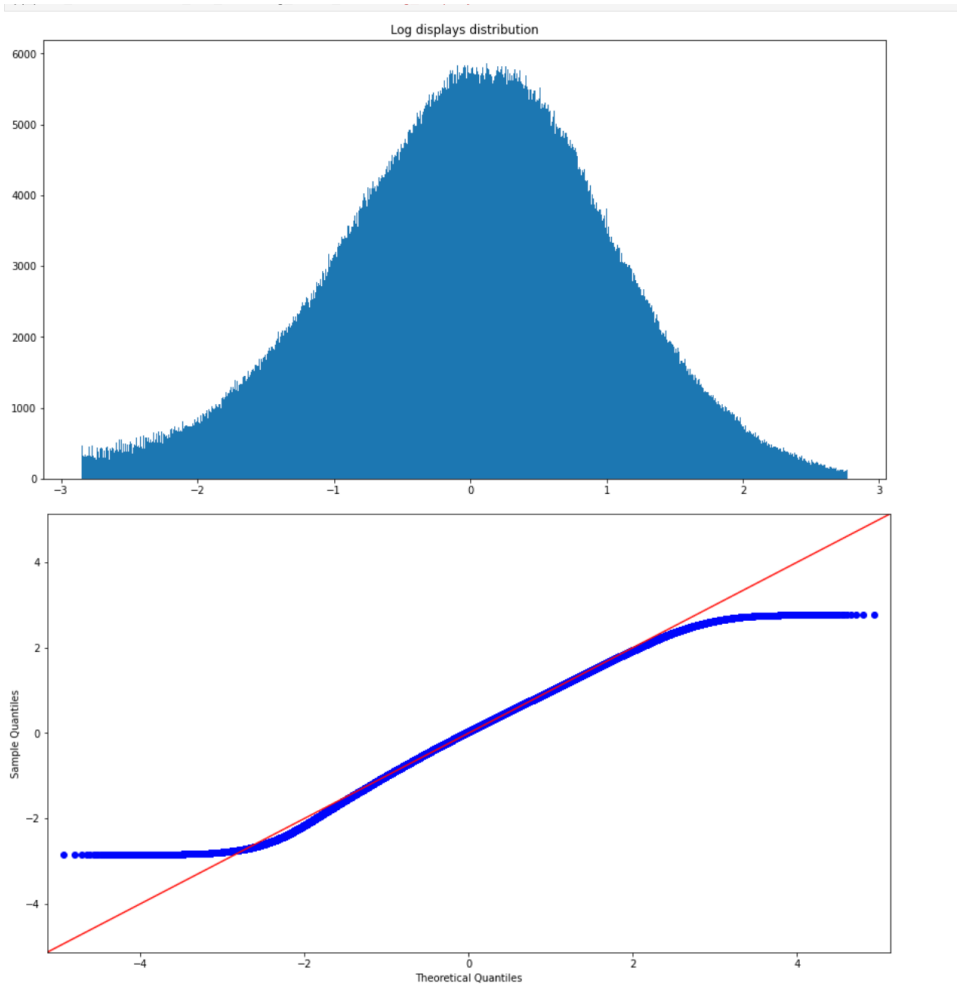

FIGURE 4.5 – Entire data distribution computed from IQR method

4.3.5 IQR by dimension

Therefore, we can also use the IQR method on the distribution of displays conditioned on the values of a categorical variable. This can be helpful for identifying complex outliers when analyzing the log displays distribution by dimension.

Campaign scenario

To do this, I would first need to group the data by the values of the campaign scenario. Next, I can calculate the IQR for each group separately. By using the lower and upper bounds formula, I can then identify and filter out any outliers in each group. This will allow us to identify and remove any unusual or unexpected data points that may be influencing the distribution of displays for each campaign scenario. The log displays distribution for each sub-dataset is shown in the following figure :

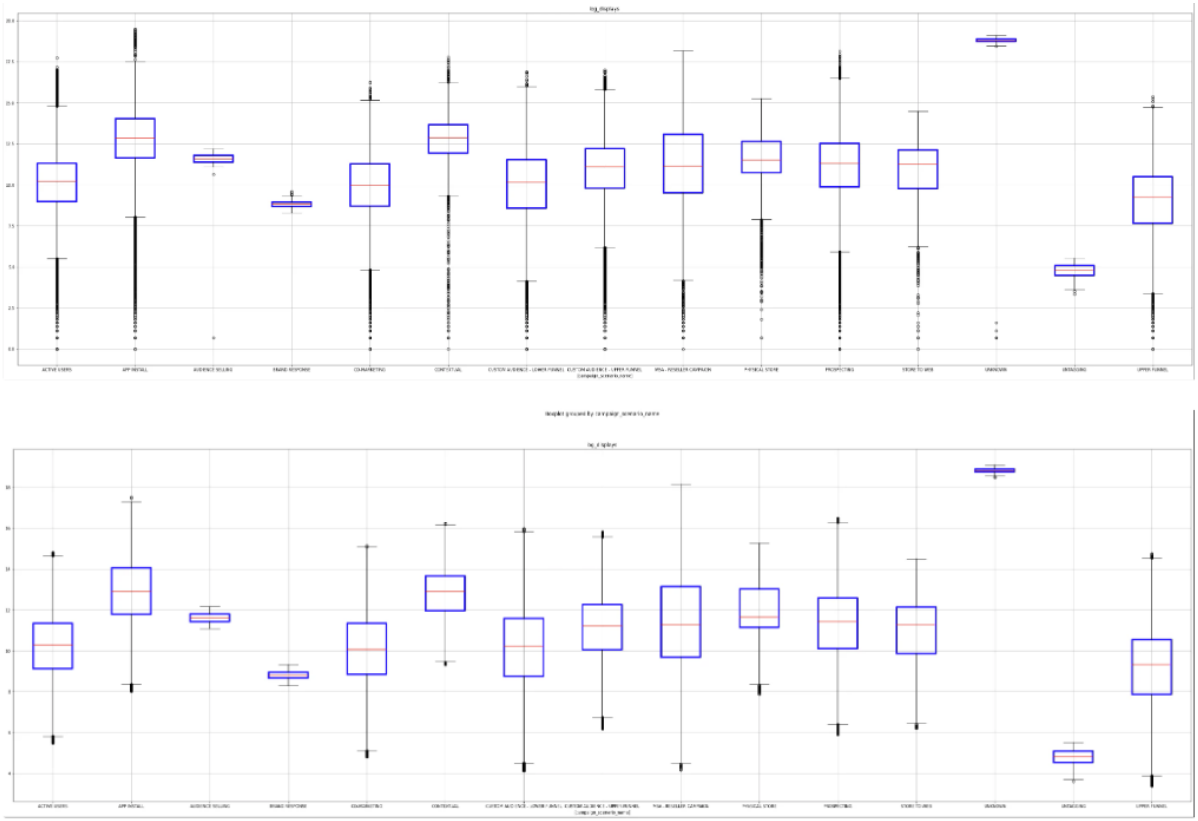

FIGURE 4.6 – log_displays distribution by each audience type after IQR filtering by campaign scenario type

Scenario campaign name	**lower bound (number of displays)**	**upper bound (number of displays)**
Upper funnel	29	2,535,178
Untagging	35	397
Unknown	104,884,542	201,074,391
Store to Web	493	6,450,100
Prospecting	369	14,431,045
Physical store	2,639	5,516,818
MSA - Reseller campaign	64	100,242,578
Custom audience - Lower funnel	60	8,699,958
Custom Audience - Upper Funnel	483	7,374,104
Contextual	11,080	11,463,796
Co-marketing	123	3,844,236
Brand response	3,829	11,805
Audience Selling	46,235	248,837
App install	3,067	45,132,248
Active users	243	2,674,304

This table shows that for each campaign scenario, we have different thresholds of log transformation of displays. It can be practiced because if we fix the lower bound,

for example, like, 3828 displays on the entire data, It may be suitable for the Brand Store scenario. However, other strategies like Prospecting could not be better because the number of displays between [368.57, 3828] is still available for the forecasting model.

The good result that data is to be getting the normal distribution after filtering out :

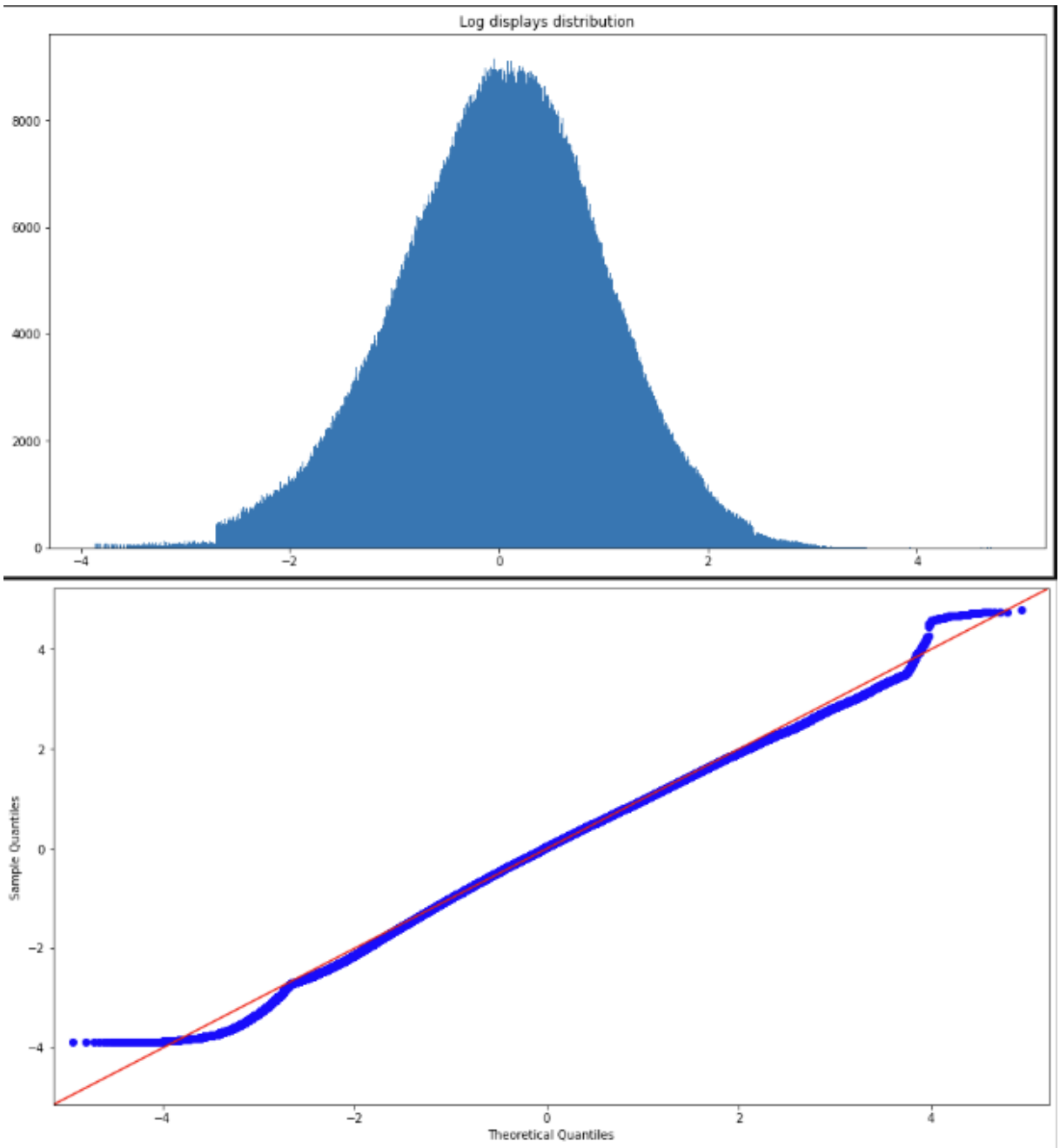

FIGURE 4.7 – Training data tends to become normal distribution after IQR filtering by campaign scenario

Revenue type

We applied the same methddology with revenue type categorical features.

Campaign revenue type	**lower bound (number of displays)**	**upper bound (number of displays)**
CRO :	123	4,620,003
Visit Optimizer_CPC :	13	8,146,556
Target COS - ARO :	1,040	2,505,866
Budget Conversion Optimization (BCO)	425	1,557,955
COS Optimizer	305	4,384,377
Target CPI - AIO	5,751	67,527,057
Budget Revenue Optimizer (BRO)	752	1,324,136
Budget Visit Optimization (BVO)	65	16,913,313
Visit Optimizer_CPM	570	16,695,754
Target CPO - ACO	923	3,751,816
Install Optimizer	3,017	11,144,832
CPM Completed View(Video) :	830	5,356,996
Value Optimizer	0.211	8,530,648
CPC	709	27,610,551
Budget View Optimization	8,844	20,921,996
CPM	73	113,568,510
Budget Store visit Optimization	368,460	9,801,581

The table shows the distribution of displays for each business model. As can be seen, the high number of displays were delivered within the Target CPI-AIO group, while the number of displays delivered within the CPM model was smaller but still compatible. These results suggest that the campaigns within the Target CPI-AIO group were more successful in delivering a large number of displays, while the campaigns within the CPM model were still able to deliver a significant number of displays despite having a smaller distribution.

The good result that data is to be also getting the normal distribution after filtering out in this category :

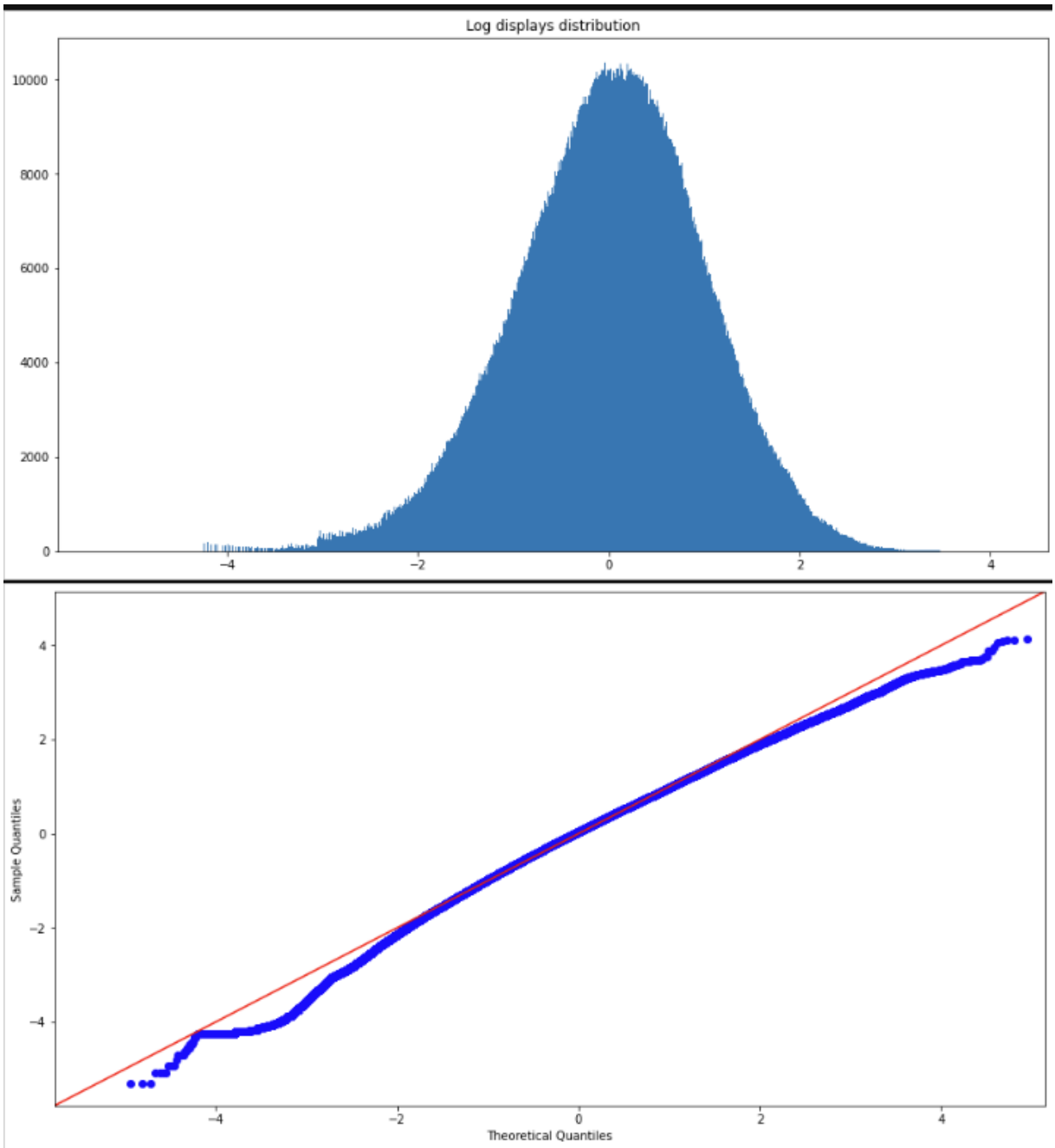

FIGURE 4.8 – Training data tends to become normal distribution after IQR filtering by revenue type

4.4 Results

Eventually, we decided to choose IQR filtering by revenue type. We used the inter-quartile range (IQR) to transform the training dataset, and applied the same transformation to the test dataset using the same scaling parameters that computed from the training data. This ensures that both datasets are being compared in a meaningful way and that any differences between the two datasets are due to actual differences in the

data, rather than differences in the scaling.

By using all filtering rules, we have 25% data points filtered in the entirity of data. After positive result, we deicide to implemenet this methodology into production.

5 Model training

5.1 Context and objectives

After having worked on enhancing our data preprocessing pipeline (e.g., campaigns from the desired scope, not using deprecated tables), feature engineering is the next step to experiment. In our case, feature engineering is the process of creating new features from existing data to improve the performance of machine learning models, that provide XGBoost model with more information about the campaign in our dataset and potentially improve its performance.

Besides, we dediced to kick off a mini hackathon by creating two generic notebooks for everyone. The idea is for everyone to experiment with more features related to campaigns and more general features available in self-service.

5.2 Baseline model

In the context of a machine learning pipeline in production, it is important to have a summarize of the various components that make up the system. One key component is the machine learning algorithm(s) being used, in this case, it is **XGBoost**, which is a powerful and widely used gradient boosting library that is effective at handling structured or tabular data. Additionally, it is important to consider data quality, how traning.testing dataset could be splitted.

Another important aspect to consider is the evaluation metric used to measure the performance of the forecasting model. In this case, we are using **SMAPE**(Symmetric Mean Absolute Percentage Error), which is commonly used in time series forecasting, it allows to measure the accuracy of the forecasted value by comparing it to the actual value (with several benefits detailed in *here*).

Additionally, understanding the **features** of the data being used for the model is crucial for effective modeling. The model's performance can be impacted by the quality and type of features being used. In this case, it is not specified which features are being used.

Lastly, **monitoring the performance of the pipeline** in real-time via a monitoring dashboard is essential for identifying trends or potential issues and making necessary adjustments. It can give us an idea of how well the model is performing over time and make it easier to identify when it's time to retrain it.

We would like to take a look of these principals components in the following sections.

5.2.1 XGBoost model

Why using XGboost ?

In fact, after the first iteration XGBoost is an implementation of gradient boosting that uses decision trees as its base model (detailed explaination in sec :xgboost*Annexe*) was choosed by being fast and easy to use, and has great synergy with our MLPlatform. The model is able to handle both categorical and continuous variables, and it is able to deal with missing values, which is crucial for new campaigns. Additionally, it is very easy to add new features to the model in case we want to fine tune it. Furthermore, it can handle (enforce) **monotonous behavior** (refer this article) of input parameters with respect to the forecasted metric. For example, if we increase our spend, we can expect to see an increase in our displays as well.

5.2.2 Current features

In production, The model takes a number of features as input, including :

- **seasonality_display_factor** : a factor that adjusts the expected number of displays based on seasonal trends. For example, if this factor is higher during the holiday season, the model may expect more displays during this time.
- **dayoftheweek** : numerical feature indicating the day of the week. For example, if dayoftheweek is 3, it may indicate Wednesday.
- **dayofmonth** : numerical feature indicating the day of the month. For example, if dayofmonth is 15, it may indicate the 15th of the month.
- **month** : numerical feature feature indicating the month. For example, if month is 6, it may indicate June.
- **daily_spend_strategy_smoothing_amount_euro** : a numerical feature representing the amount of money (in euros) that is used to smooth out daily spend as part of the campaign's spending strategy.
- **campaign_revenue_type** : numerical feature indicating the type of bussiness model.
- **campaign_CPI_euro** : a numerical feature representing the cost per install (CPI) of the campaign in euros.
- **campaign_COS** : This may be a numerical feature representing the cost of sale (COS) for the campaign.
- **campaign_CPO_euro** : This may be a numerical feature representing the cost per order (CPO) of the campaign in euros.
- **campaign_CPM_euro** : a numerical feature representing the cost per thousand impressions (CPM) of the campaign in euros.
- **campaign_CPC_euro** : a numerical feature representing the cost per click (CPC) of the campaign in euros.
- **similar_audience_size** : a numerical feature indicating the size of a "similar" audience for the campaign. This could refer to an audience that shares similar characteristics with the target audience of the campaign.
- **visitors_30d** : a numerical feature indicating the number of visitors to a

website or landing page in the past 30 days.
— **custom_audience_size** : a numerical feature indicating the size of a custom audience for the campaign. This could refer to an audience that has been specifically selected or defined for the campaign.
— **displays_avg28d_targeting** : a numerical feature indicating the average number of displays over the past 28 days, targeting same operating system (iOS/Android), devices (desktop/mobile/tablet) and environment (app/web).

These factors are presented for the performance of a campaign, make a decision the deliveried displays should be targeted as lable.

5.2.3 Data quality

Data are recorded in three monthsfor the purpose of training a model and making predictions querying from the *preprocessing_displays* table including campaigns. Training/ testing datset are splitted by a column called "**is_train_set**" containing a binary value (0 and 1) indicating whether a given row should be included in training set or the testing set.

Currently, we have 50/50 splited ratio between two datasets with **971,920** training samples and **980,695** testing samples.

5.2.4 Current performance in monitoring dashboard

Currently, our monitoring dashboard for a forecasting campaign that can be used to track the performance of the forecast model in other KPIs over time and detect any changes in the underlying data generating process. This can be useful for identifying areas where the model may need to be updated or improved, and for making informed decisions about forecasting strategy.

Checking that the performance match the performance dashboard, we have currently 48.5% SMAPE error in monitoring dashboard.

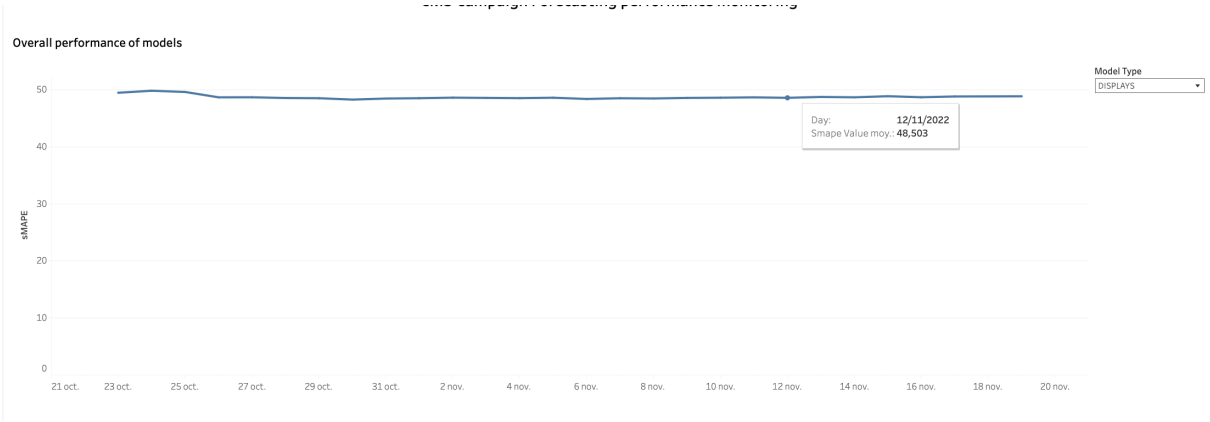

FIGURE 5.1 – Overall performance of model in testing dataset in real time

By the result below, we gain better performance with **new preprocessing** steps mentioned above (44% of SMape)

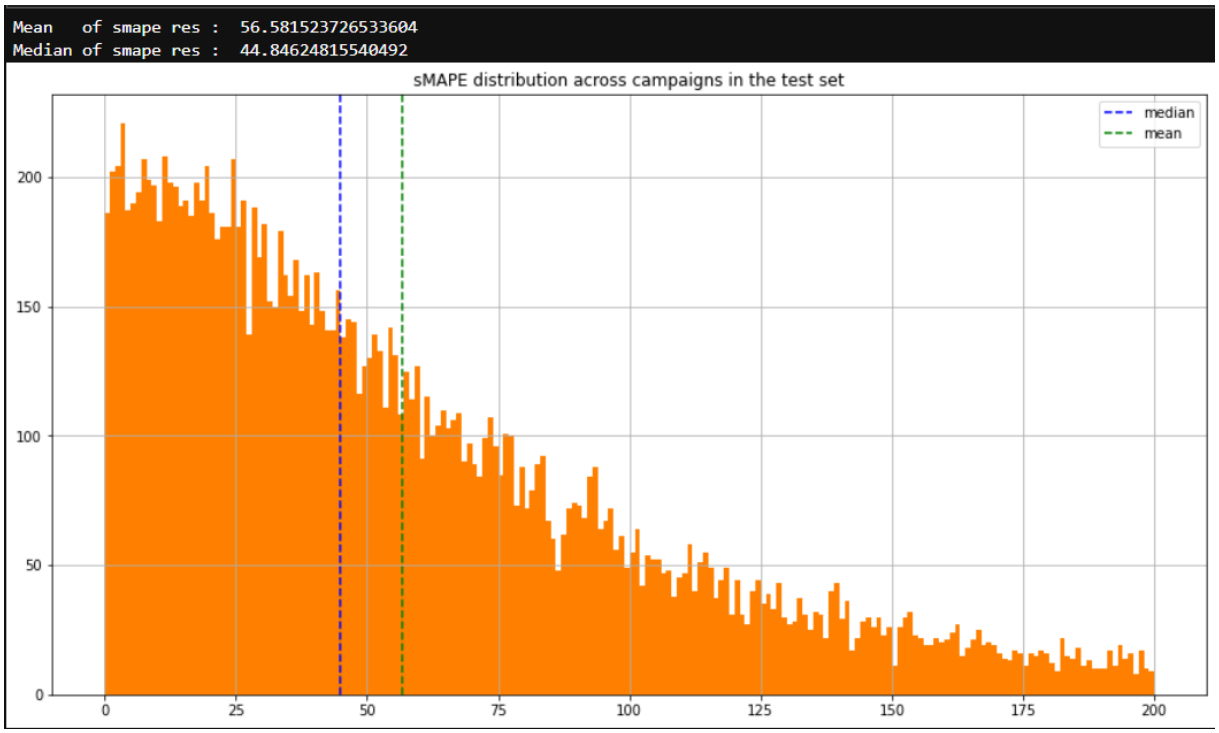

FIGURE 5.2 – Overall performance of model after new preprocessing rules

Features importance

XGBoost calculates feature importance by training a model and then evaluating the importance of each feature based on how much the model's performance improves when the feature is included. This is done by training the model multiple times, each time using a different subset of the features. The improvement in the model's performance when using a particular feature is then used to calculate the importance of that feature.

The benefit of this approach is that it provides a way to identify which features are most important for making accurate predictions with the model. This can be useful for a number of purposes, such as identifying which features to include in the model, selecting the most important features for a particular use case, or simply understanding which features are driving the model's predictions.

It is important to note that the calculated feature importances are relative to one another, so it is important to look at the relative difference in importance between features rather than the absolute importance values. Additionally, the calculated feature importances are specific to the trained model and the data it was trained on, so they may not be directly comparable to feature importances calculated for a different model or dataset.

Let's see how the importance features computed currently in prod :

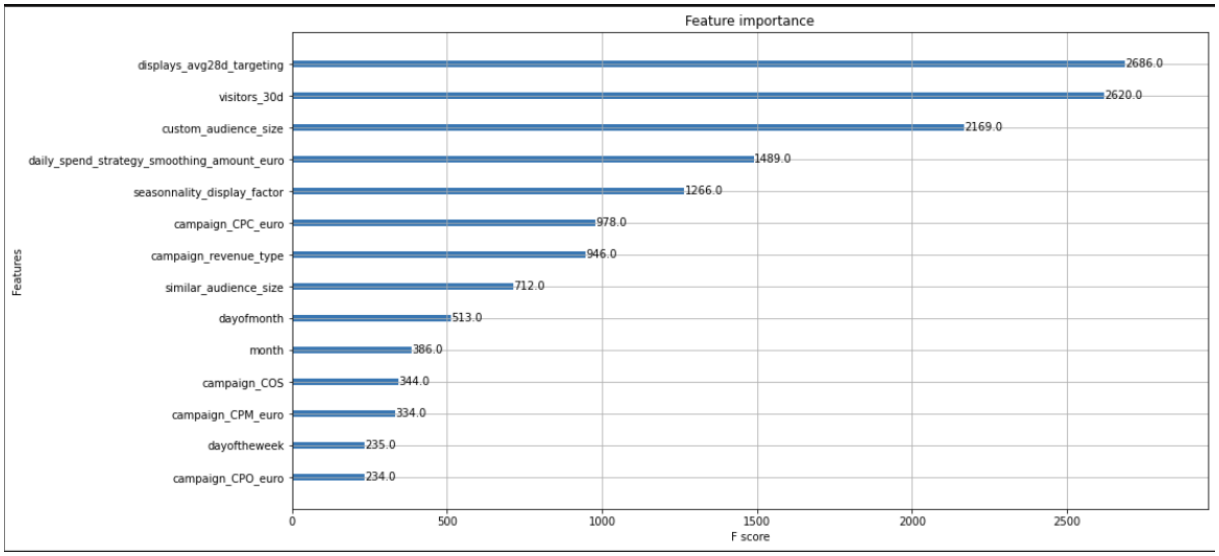

FIGURE 5.3 – Baseline feature importances

I measured the importance of the feature **display_avg28d_targeting** by removing it from the model and re-training the model with the remaining features. This will allow me to see the difference in performance and determine the importance of this specific feature in the both training and testing datasets.

Model	**Mean SMape in training dataset**	**Mean SMape in testing dataset**
Without displays_avg28d_targeting	46%	51%
With displays_avg28d_targeting	40%	44%

As the result, The feature **display_avg28_targeting** is a strong predictor for the target variable and has a significant impact on the model's performance. However, it does not contribute to the overfitting problem in our training process. It is important to carefully monitor the performance of significant features like this one.

5.3 Why using MLFLow for feature engineering offline analysis

In feature engineering analysis process, we decided to access **MLFlow** which is an open-source platform for managing the end-to-end machine learning lifecycle. It is designed to help manage and track experiments, package and deploy models, and provide reproducibility. One potential use of MLFlow in the context of feature engineering is to track and compare the performance of different feature sets or feature engineering techniques on a particular modeling task. This can help us identify the most effective features to include in your model and improve its performance.

Using MLFlow for offline analysis of feature engineering can also be useful for sharing my work with others. By storing experiment results and artifacts in a centralized location, I can easily share my findings and allow others to reproduce your work. Additionally, MLFlow provides tools for visualizing and comparing the results of multiple experiments, which can help you identify trends and patterns in your data and make more informed decisions about your model.

We want to use two of the main concepts of MLflow to store :

- **Metrics** : It is a set of (key, step) => double that stores metrics for one run. A step is a learning step, and the framework defines it. In our case, the median and the mean of the SMAPE error of testing data are chosen.
- **Artifacts** : All the files related to the experiments (XGBOOST model and importance features).

5.4 Methodology

Feature creation involves deriving new features from existing ones. This can be done by simple mathematical operations such as aggregations to obtain the mean, median, mode, sum, or difference and even product of two values. These features, although derived directly from the given data, when carefully chosen to relate to the target can have an impact on the performance(as demonstrated later).

The idea is for everyone to experiment with more features related to campaigns and more general features available in self-service : frequency capping, geoloc, domains white/black list, placement_targeting, client_segmentation.

Furthermore, I attempted to engineer features, for example :

- Add more projections of categorical features (ex : campaign_revenue_type).

- Cross features (ex : displays_avg28d_targeting for campaigns of the same country and client segmentation)

5.4.1 Implementation

Capping features

This feature is considered to the amount of budget the adset is allowed to spend during one day. Adding this new feature into our model as "**daily_capping**", which is queried from **dim_campaign** (this table prensensts all the features of a campaign could release) that is merged into our **entired training data**, on each **campaign_id** (refering to a unique identifier for each campaign in the data).

It does not have a significant impact on the model's performance, as shown by its low importance on the feature importance dashboard. It only slightly reduces the Smape by 0.5% in both datasets.

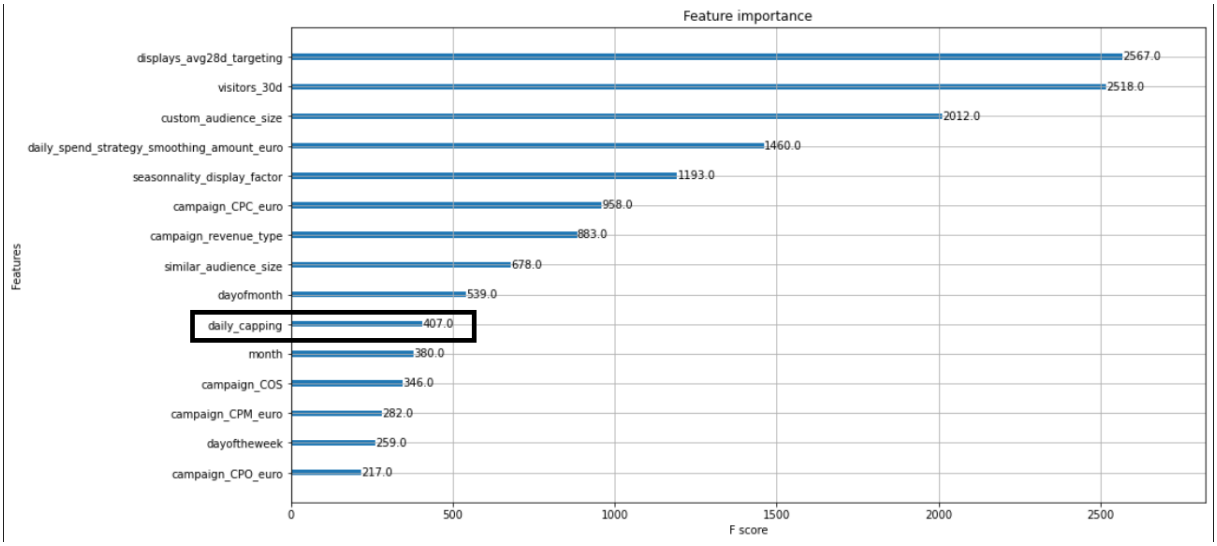

FIGURE 5.4 – Capping feature importances

Target placement Domain and/or app white-black listing

In the context of a marketing campaign, target placement refers to the specific websites, platforms, or apps where an advertisement will be shown. White-listing is the process of specifying a list of approved websites, platforms, or apps where an advertisement is allowed to be shown, while black-listing is the process of specifying a list of websites, platforms, or apps where an advertisement is not allowed to be shown.

We are trying to calculate the number of white-listed and black-listed domains and applications for a marketing campaign. So far, we have made some progress, but it is taking longer than expected.

The main idea is to get the domain/apps by day with the maximum timestamp on this day and sum up the number of these domains and join them into **entire data**.

This feature is not stable and reduced slowly the model performance also.

General features available in self-service

As we can see, **displays_avg28d_targetting** is an essential feature currently in our model, by a combination of country_level and audience targeting, demonstrating that historical feature impact in our model. Two following formulas are used for these new feature implementations.

$$displays\_avg\{window\}d\_dimensions_{dims,day} = mean(displays\_\{window\}d_{dims,day})$$

$$displays\_avg\{window\}d_{day,campaign\_id} = \sum_{day=day}^{d-window} (displays_{day,campaign\_id})/window$$

where *dims* is categorical feature values by each dimensions as Country level, client segment,... etc projected by each campaign_id, *day* is specific day that campaign alive and *window* is the number previos days, maybe in [7, 14, 28].

We applied this methodology on certain dimensions and different windows which are shown in the following sections.

Firstly, we applied the methodology in term of set of dimensions exsting in our log :

- **Country level** : Advertisers can select specific countries or regions in which their ads will be shown.
- **Campaign scenario name** : Advertisers can give their campaigns a specific name to help them keep track of different campaigns and differentiate them from one another.
- **Campaign revenue type** : Advertisers can choose the business model they will use to generate revenue from their campaigns.
- **Client segment** : an indicator of the amount of money a advertiser has spent at Criteo, allowing for a distinction between larger and smaller clients in terms of financial investment.

Window	Median Smape	Mean Smape
7	42.05%	52.92%
14	41.55%	52.93%
28	41.94%	52.95%

These features impact the model performance that reduce SMape error from 44% to 41%.

Next step, we attempted to take a look at how **cross feature** affect our model value because XGBoost should find cuts to "naturally" do the crossing, we have several dimensions combined : 'country_level' and 'campaign_revenue_name', 'country_level' and 'campaign_scenario _name', 'country_level' and 'client_segment'.

Window	Median Smape	Mean Smape
7	42.35%	53.52%
14	42.15%	53.64%
28	42.35%	53.52%

And then, we experienced by including all features mentioned above in our model to see what happen :

Window	Median Smape	Mean Smape
7	40.09%	52.42%
14	41.12%	52.46%
28	41.31%	52.43%

As the result, we can deduce that these features have impacted to the model performance that help minimize SMape error from 44% to 40% in term of testing dataset.

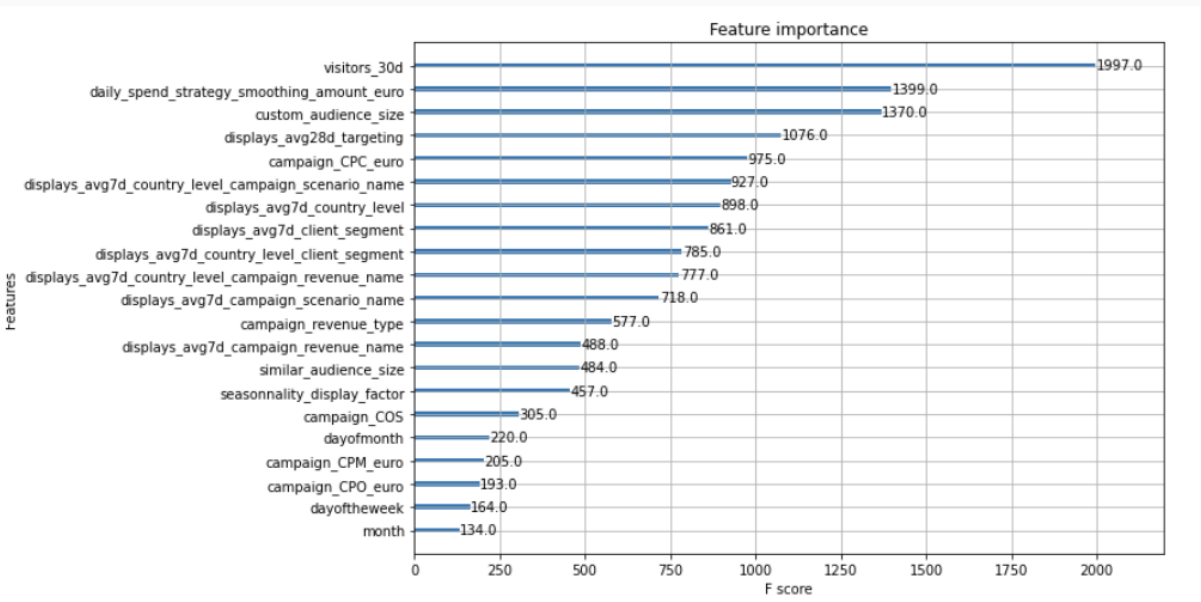

FIGURE 5.5 – feature importances

Remark : Considering the features importance dashboard, it's evidence that adding these certain features causes the importance of **display_avg28d_targeting** features (F1 scores) to decrease. However, these historical new features have affected to our model. We should care about the **trade-off** between the F1 score and the SMAPE error, but in our case, it's very good idea to include as many relevant features as possible in your model, as this can help to improve its performance.

Historical features based on the partner level, client level.

Historical features like "**displays_avg28d**" by campaign_id are specific to individual campaigns and only apply to existing campaigns. Using this feature in the model can introduce bias, as a new campaign would not have any historical data available in the training dataset.

To mitigate this issue, It's possible to implement these features with an additional dimension, such as **client_id** or **partner_id**, in the second iteration of feature engineering. This allowed us to capture the number of displays in the past 28 days for a partner, etc, who may be connected to multiple campaigns, particularly those that are high-value for Criteo.

$$displays\_avg\{window\}d_{day,partner\_id} = \sum_{day=day}^{d-window} (displays_{day,partner\_id})/window$$

For new campaigns don't present the same partner_id with training_set, we need to set in Nan values, beacause for the new campaigns assigned to new partners in dataset, we don't get any these information.

- training SMape : 19.1% - testing Smape : 80%

We can deduce that leads overfiting problem in our data set. The feauture is not incorrect computed because training and testing data have conflicted information in term of the time, so information from testing set could be leak into trainng process.

5.4.2 Computing sMAPE on "valuable" subsample of clients

In addition to the global sMAPE we're computing, we should look at a specific subsample of clients that are considered of high value at Criteo : for example, retailers doing AO campaigns. So, in this case, we have 12801 samples in our log with 209 campaigns.

To be detail, we select in our dataset that AO is associated to these campaincampain_revenue_types :

10 : ACO

12 : ARO

24 : AIO (this types belongs to video campaigns which we have filtered out in preprocessing process).

We decrease the SMape error in this subset from 57.2% to 53.7% median of SMape.

Remark : It is essential to assess a model's performance on a targeted subset of

the data, particularly if it will be used to make decisions or predictions in a specific scenario. By doing so, we can better understand how relevant features impact the model's performance and adapt accordingly and propose any other solutions.

5.5 Conclusion

Typically, engineering feature step plays an important role in our offline training process. After offline analyst of this part, we gain the best current model which enhance the model performance from 48% to 40%.

Based on previous research, we realize that historical feature plays the important feature and affect significantly in our model. However, we will have to take a look at these important features because they can cause the data leakage problem. There are several causes, for example :

- Some features like **display_avg28d_targeting**, etc,... As mentioned above, these features from the same dimensions (or similar settings) in each day manipulated from **displays_avg28d** values of each campaign, and they were scaled before splitting. There might be a risk of data leakage, which may allow the model to adjust its predictions based on information from the testing dataset. It is important to verify whether this is the case in order to avoid data leakage and ensure the accuracy of the model ; because **displays_avg28d** may bring the campaign performance from testing dataset into training process.

- If we want to experiment with historical feauture like historical features like **the number of displays in the past 28 days for a particular partner**, splitting our training data randomly rather than by time may pose a risk. For example, if a partner A creates a campaign on 10-01-2022 and it is included in the training dataset, but there is also a new campaign created on 05-01-2022 and it is included in the testing dataset, the feature **displays_avg28d_partner** in the training set will gather information from the campaign in the testing dataset on **05-01-2022** for instance. This can lead to a leakage of future information into the training process and potentially result in unrealistic or biased evaluation when the model is tested.

- We need to care about the features distribution drift. For example if one campaign x a targeting option exist in test set but not in training set, it should be fulfill as null values.

Remark : In summary, **data leakage** can be a challenging issue to address as it is not always obvious. It poses a significant risk as it can cause our models to fail unexpectedly, even after thorough evaluation and testing. Thus, it's crucial to carefully examine and assess our pipeline to ensure that it adheres to best practices for preventing data leakage. Additionally, as previously mentioned, it's essential to verify two key ideas based on the analysis performed :

- The feature **display_avg8d** is a crucial metric for assessing the performance of a

campaign over the past 28 days. It is commonly used in conjunction with various historical features, such as targeting options and other dimensions, to provide a comprehensive understanding of campaign performance. However, it's important to note that scaling this feature prior to data splitting for analysis may lead to the importation of testing dataset's performance into the training model. To circumvent this issue, it would be beneficial to supplement the use of display_avg28d with the historical feature by one feature like **displays_avg28d_same_settings**, which is the average number of displays delivered in the same settings within self-services, to provide a more generalized perspective on campaign performance.

- The second option is to analysis about the splitting training data. We split data by time, filtering out existing campaigns in testing data set. Keeping an eye data quality in a both training and testing datasets like the imbalanced catacorical features need to be taken into account. By doing this, we could experiment correctly historical features affecting in our model. I believe that ensuring the quality of the data should be a top priority from the outset.

6 Offline analysis : Existing campaigns model accuracy

6.1 Context and objectives

After having analyzed the accuracy of models for new campaigns, it is now time to take a back look at existing campaigns' outcomes. Indeed in a near future, we plan to show the forecasts directly in Cockpit (homepage of Management Center), meaning that we should be confident in the forecasted values for this use case.

This analysis also aims to benchmark the formula we use in prod with a XGBoost model to see if there are some big differences in terms of accuracy and decide whether we should have a specific model at some point.

6.2 Methodology

6.2.1 Training Data

Generating our train and test sets

The dataset is collected by querying from the adset timeline table. For the first time, this dataset is a collection of all existing campaigns' information, including baselines' details (for instance, how many displays were made in the last 28 days), CPX constraints(campaign COS, campaign CPM,...), etc., and a current day that the campaign lives. These are campaigns recorded between July 20-07-2022 and 2022-09-18.

Contrary to new campaigns framework, where we split data per campaign_id, we did a split by time because we use historical data for each campaign. The test set corresponds to all the data in September.

FIGURE 6.1 – Training/ testing dataset splitting by day

6.2.2 Applying filters on the data

In the UI, we only forecast campaigns that have been live for 28 days at least. Also, we don't want to test our model on campaigns with just a few days active in the test set. Therefore firstly, we apply the following filters :

- Train set campaigns should be live for at least 28 days.

- Test set campaigns should be present in the training set and should have at least 10 existing days in the test set.

Secondly, we do a bit of preprocessing and apply several filters to the data :

Filtering out campaigns with less than a minimum number for each label (label can be displays, clicks or sales). To determine each value, we considered the 10% quantiles of each labeled data set to avoid deleting excessively multiple data. For example, we attempt to filter out 10% of data regarding the clicks metric, then map 10% 's quantile to the number of displays set to get the relevant values (for an instant, 1000), meaning that 10% of campaigns gain at most 1000 displays.

- Filtering out video campaigns purely contextual campaigns.

6.2.3 Modelling

Formula

For each metric (displays, clicks, we compute its values at time **t** using the following formula)

$$\text{predict(metric, t)} = \text{avg\_28d(metric, t}_0\text{)} * \frac{seasonality\_factor(metric,t)}{seasonality\_factor\_avg\_28d(metric,t_0)}$$

t_0 is the latest time campaign lives, t is the future day ($t > t_0$)

Each seasonality factor is mapped by its country and vertical and we reweight the baseline with a seasonality ratio.

However, we should be careful and not introduce bias while using the formula. Indeed, in the test set, the values of the baseline is taking into account the displays values that we shouldn't be aware of. This means that in the test set, the baseline value we should use is the "last information" we have the right to access, which is the first value of displays_avg28d.

	day	displays_avg28d	displays_avg28d_prod
0	2022-09-01	1.336365e+08	1.336365e+08
1	2022-09-02	1.341201e+08	1.336365e+08
2	2022-09-03	1.344778e+08	1.336365e+08
3	2022-09-04	1.351353e+08	1.336365e+08
4	2022-09-05	1.359235e+08	1.336365e+08
5	2022-09-06	1.365976e+08	1.336365e+08
6	2022-09-07	1.374791e+08	1.336365e+08
7	2022-09-08	1.379767e+08	1.336365e+08
8	2022-09-09	1.384608e+08	1.336365e+08
9	2022-09-10	1.385377e+08	1.336365e+08
10	2022-09-11	1.389311e+08	1.336365e+08

FIGURE 6.2 – The feature "displays_avg28d_prod" that is fullilled by the last information from training dataset, is simulated as a historical feature manipulated in production

XGBoost model

In this section, we would like to train a XGBoost model for this use case and compare its performance with the formula used in prod. It was also the occasion for us to try to tune hyperparameters of the model, and to do so, we implemented a Gridsearch.

Gridsearch implementation

Instead of cross-validation, we split training set into 2 two subset : training and validation test. In the "validation" part, it's only the data points that are the more "old" possible, hence, in this case, we just have one validation test.

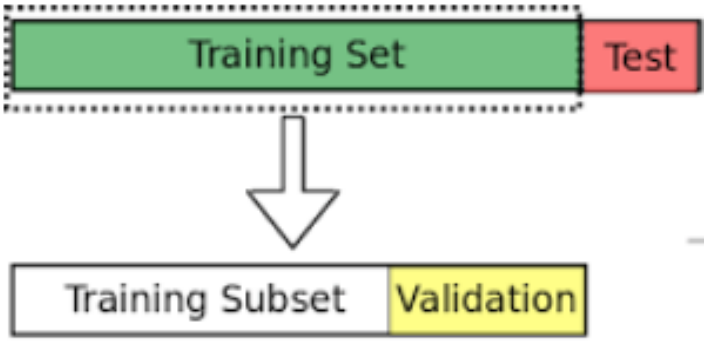

FIGURE 6.3

We tried to tune two influential hyperparameters :

- **max_depth** : increasing this value make model more complex and likely to overfit

- **eta** : how much we update predictions after each tree.

A simple way to estimate how the model might change with each modifiable parameter is to use learning curves. A learning curve, in this case, is a plot of its performances, training loss, training accuracy, validation accuracy, and test accuracy against each parameter. This experiment help us to consider how impactful eta and max_depth are

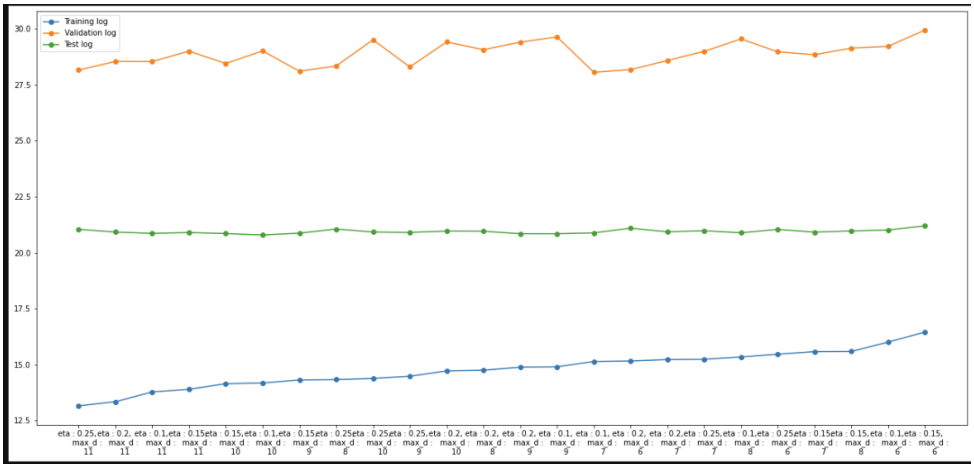

FIGURE 6.4 – Learning curve express the model performance in each specific datasets to gain the best hyper-parameters

As we can see the figure above, eta was used in update to prevents overfitting against the complexity of the model (maximum depth of tree). Increasing eta means that decreasing max_depth give us a generative model. In this case eta = 0.25 and max_depth = 11 is our choice.

We didn't deep into this part, the main objective is to see how XBoost express the performance with new hyper-parameters.

6.3 Results

In this section, we will summarize the results we obtained :

6.3.1 Displays

Model	Median sMAPE	Mean sMAPE
Formula	18.2	29.7
xGBoost	14.8	24.3

By taking a look at the Feature importance of XGBoost dashboard in this usecase, we can deduce that it was expected to observe here that the best features from the "new campaigns" use case (displays_avg28d_targeting, visitors_30d) are outperformed by historical data.

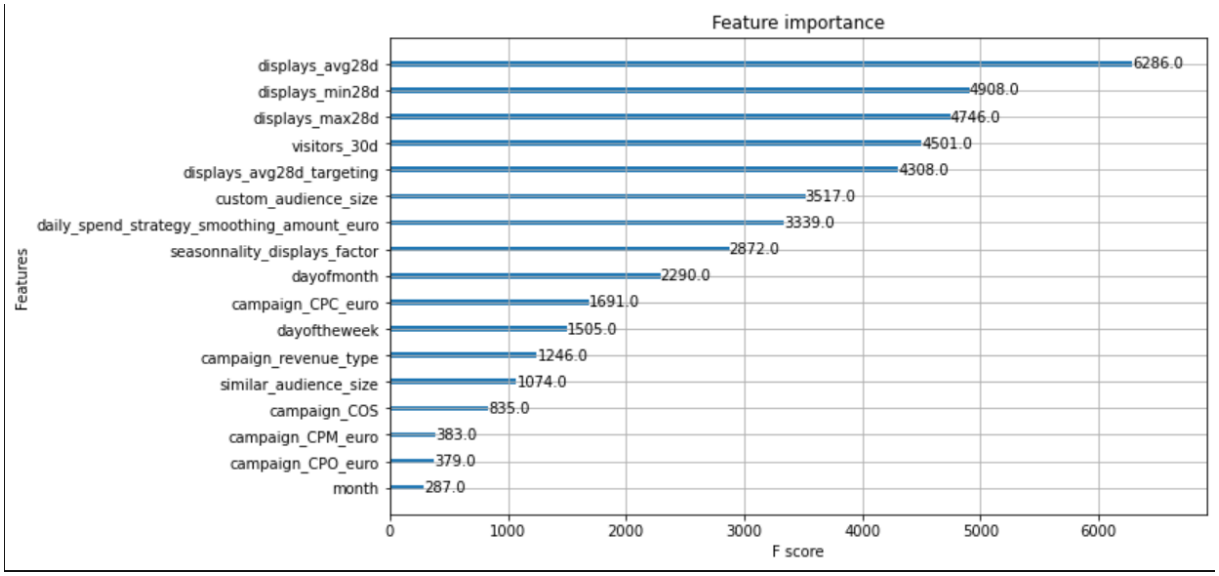

FIGURE 6.5 – Feature imporatance dashboard of XGboost with features in term of existing campaign forecasting number of displays

6.3.2 Clicks

Considering clicks Metric performance, we got the pretty performance in term of this KPI.

Model	Median sMAPE	Mean sMAPE
Formula	18.8	29.7
xGBoost	14	20.8

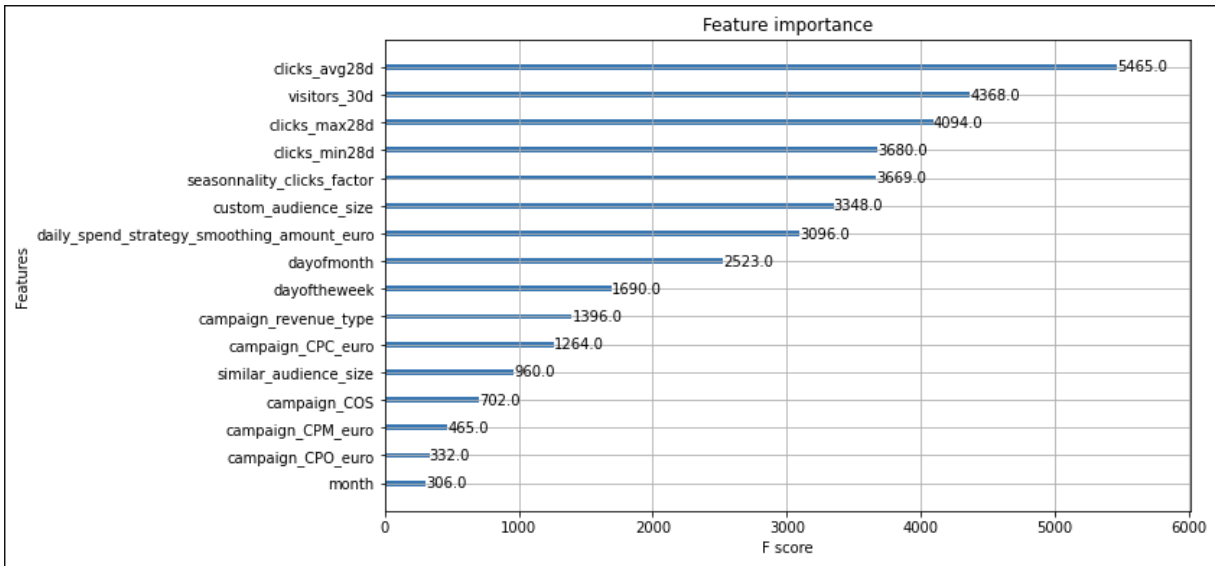

FIGURE 6.6 – Feature imporatance dashboard of XGboost with features in term of existing campaign forecasting number of visits

6.3.3 Sales

Model	Median sMAPE	Mean sMAPE
Formula	32.2	50.5
xGBoost	37.4	64.21

Considering sales, we see that this metric is tricky to predict. 44.6% of data contains 0 sales, so SMAPE errors are higher than other metrics ; with 0 at the label, we have SMAPE errors like 200* (x-0)/(x+0) = 200 in this use case, which leads to the lousy accuracy of the forecasting model.

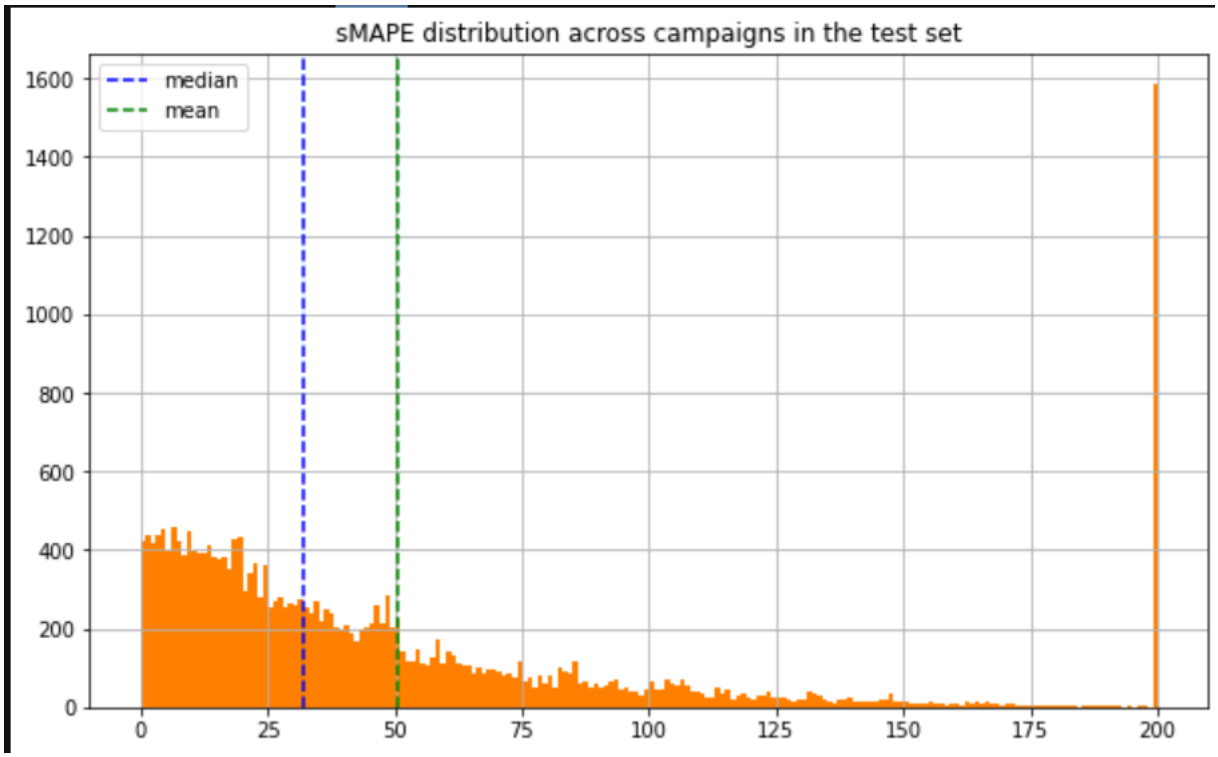

FIGURE 6.7 – Sales Smape distribution (formula)

6.4 Conclusion

In our production environment, we use a formula that has very similar performance to XGBoost. This formula has been evaluated in our existing campaign use case, and it has shown a pretty good accuracy of around 18% sMAPE. it's very nice to have such close performance of a simple model vs XGBoost because this easy formula is super easy to maintain and to debug.

It is worth noting that this analysis also confirms the importance of filtering outliers when it comes to improving the model's performance. This is an important aspect that should be considered in any model development and optimization efforts. Furthermore, performance of forecasts on existing campaigns are way better than on new adset because we have access to historical data which helps a lot in such use case. Overall, our results show that the formula we use in production is effective in the given use case and it has a similar performance to XGBoost, a widely used model in various applications.

7 Conclusion and Takeaways

This internship has been on of the most enriching experiences during my COmputer Science curriculum at UTC, especially in Data Science domain. During this period at Criteo, I had the opportunity to work on a forecasting model project and gain hands-on experience in developing and implementing machine learning models in a real-world setting. My experience included working with large datasets, experimenting with various machine learning techniques and tools, and learning the importance of data pre-processing, feature engineering, and model evaluation.

One of the key takeaways from the internship was the importance of considering the business domain when developing forecasting models. Additionally, I learned the importance of effective communication and collaboration within a team when working on a large-scale project. While my internship provided me with valuable experience, there were some tasks and aspects of the project that I did not get to explore in-depth, such as deploying machine learning models in production and understanding its use cases.

In addition, I had a chance to work with all members in the team, I figured out that besides the technique skills, communication takes an essential role to go further un Machine Learning engineer. Although I did improve this skill sonsiderably during my time at Criteo, I also need to keep practicing it more often to show off my work efficiently to future co-workers and make other skills more visibles.

Nonetheless, I am grateful for the guidance and support provided by my mentors and colleagues during this internship, and I look forward to applying what I have learned in future projects.

8 Bibliographie

[1] *XGBoost docmumentation* https://xgboost.readthedocs.io/en/stable/

[2] *Criteo- Global Leader in Commerce Marketing* https://www.criteo.com/company/

[3] *Interquartile range* https://en.wikipedia.org/wiki/Interquartile$_r$*ange*

[4] *How to interpret sMAPE just like MAPE* https://medium.com/@davide.sarra/how-to-interpret

A Annexes

A.1 XGBoost Algorithm

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

The basic idea behind gradient boosting is to iteratively add new models to an ensemble, where each model is trained to correct the mistakes of the previous models. In XGBoost, the models are decision trees, and the ensemble is trained using gradient descent.

The XGBoost algorithm can be broken down into several key steps :

1. Initialize the ensemble with a single decision tree : This can be done using any decision tree algorithm, such as CART or C4.5.

2. Fit the ensemble to the training data : The ensemble is trained to minimize the loss function, which is typically the mean squared error (MSE) for regression problems or the cross-entropy loss for classification problems.

3. Iteratively add new decision trees : At each iteration, a new decision tree is added to the ensemble, with the goal of correcting the mistakes of the previous trees. The new tree is fit to the negative gradient of the loss function, with respect to the ensemble's current predictions.

4. Prune the decision trees : XGBoost uses a cost complexity parameter, known as "gamma," to control tree pruning. This helps to prevent overfitting, as smaller trees are less likely to overfit the data.

5. Regularization : XGBoost includes both L1 and L2 regularization for the leaf weights of the decision tree, which helps to prevent overfitting.

6. Weighted Quantile Sketch : XGBoost uses a novel data structure called the "Weighted Quantile Sketch" to approximate the gradient statistics, which makes it faster and more memory-efficient.

7. Cross-validation : XGBoost supports cross-validation, which is important for model selection and hyperparameter tuning.

8. XGBoost algorithm is highly efficient and scalable, it is able to handle large datasets and missing values, and it has built-in regularization to prevent overfitting. Additionally, it has many hyperparameters that can be fine-tuned to optimize the performance of the model.

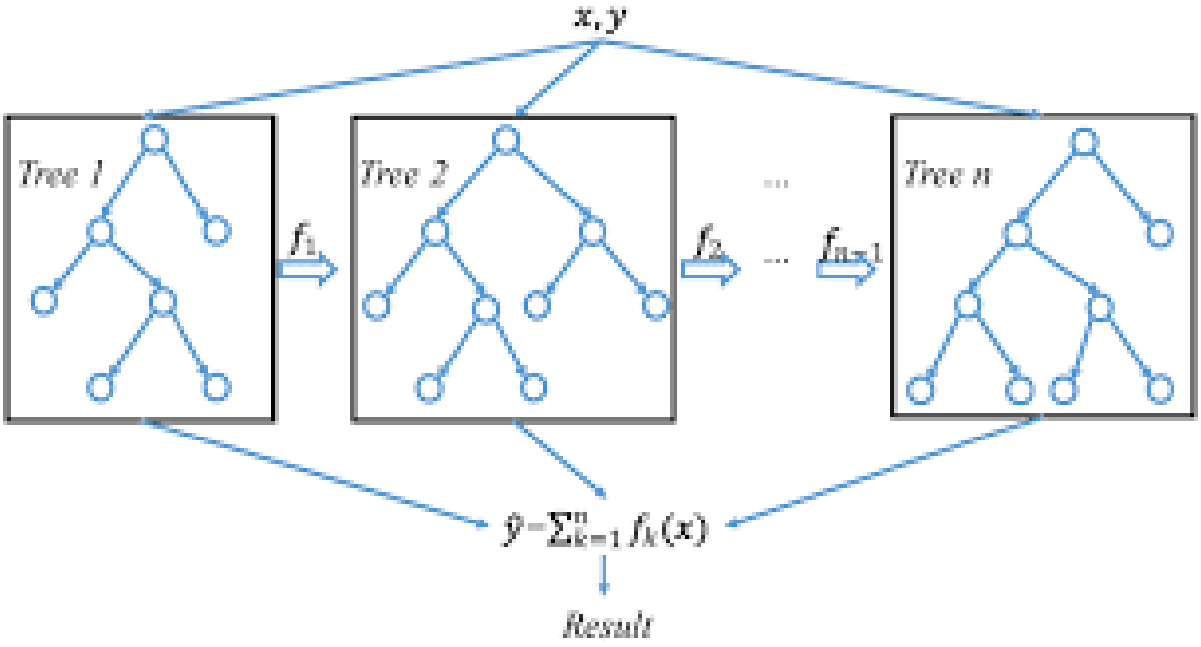

FIGURE A.1 – A general architecture of XGBoost

In conclusion, XGBoost is a powerful and efficient gradient boosting algorithm that is well-suited for large datasets and offers a wide range of features for both training and evaluating models. It is a popular choice among data scientists and machine learning practitioners and is widely used in industry and academia.

A.2 SMape

SMAPE (Symmetric Mean Absolute Percentage Error) is a commonly used metric for evaluating the accuracy of forecast models, particularly in time series analysis. It is a symmetric version of the mean absolute percentage error (MAPE) and is defined as the average percentage difference between the forecasted and actual values, with the percentage calculated based on the actual value.

When it comes to measuring accuracy relative to the actual values, the most popular metric is MAPE, the mean absolute percentage error.

$$MAPE = (\frac{100\%}{n}) * \sum_{i=1}^{n} |\frac{Ai - Fi}{Ai}|$$

where :

Ai is the actual value, Fi is the forecasted value, n is the number of data points

So, effectively, MAPE is meaningful only if all observations have relatively large actual values. When this is not the case, a sibling of MAPE, sMAPE — the symmetric

mean absolute percentage error — can be used instead.

$$SMAPE = (\frac{100\%}{n}) * \sum_{i=1}^{n} |\frac{Ai - Fi}{(|Ai| + |Fi|)/2}|$$

To get better sense fot this, we can plot these metrics for various levels.

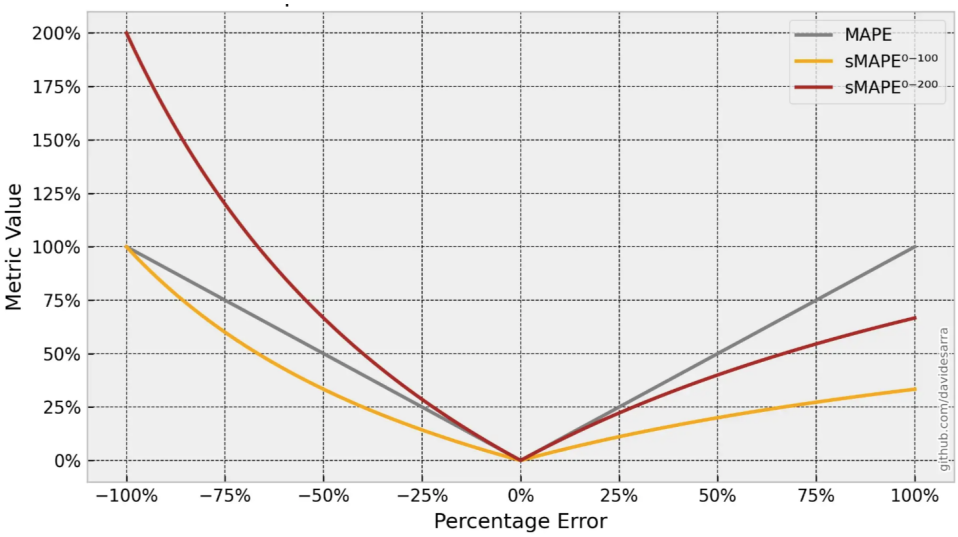

FIGURE A.2 – Comparasion between MAPE and sMAPE metrics

sMAPE, approximates MAPE also around 0%, that is when the scale of actual and predicted values is similar.

One of the advantages of SMAPE is that it is scale-independent and can be used to compare forecasts across different data sets, regardless of their range of values. Additionally, SMAPE is bounded between 0 and 200, with a lower value indicating a better forecast.

However, SMAPE has some disadvantages as well, one of them is when the actual value is zero or close to zero, the SMAPE is undefined or very big. Therefore, it is not suitable for datasets with low or zero values.

In conclusion, SMAPE is a widely used metric for evaluating the accuracy of forecast models, particularly in time series analysis. It is symmetric, scale-independent, and easy to interpret, but it has some disadvantages such as when the actual value is zero or close to zero. It is important to consider the characteristics of the data when selecting a metric for model evaluation.

Table des figures

Table des matières