# Documentation Technique du Simulateur Marketing Mix Modeling (MMM)

## Introduction

Le simulateur de Marketing Mix Modeling (MMM) de MediaROI est une application développée avec **Streamlit**, permettant de simuler les impacts de différents budgets publicitaires sur les performances marketing du client. Le choix de streamlit est motivé par la facilité d’intégration avec la librairie lightweight et la rapidité de déploiement.

Cette application repose sur un modèle de MMM entraîné et des données spécifiques à chaque projet, stockés dans un dossier « mission ».

## Structure de l'Application

L'application est construite autour de plusieurs composantes essentielles, y compris le chargement dynamique des fichiers, le rendu visuel personnalisé, la gestion des états via session\_state, et la possibilité de télécharger les résultats au format CSV et PDF.

### Modèle et fichiers de données

Le modèle utilisé dans l'application est stocké sous la forme d’un fichier **model.pkl**, et les fichiers de données sont :

* **raw.xlsx** : Données brutes utilisées pour alimenter le modèle.
* **results.xlsx** : Résultats et coefficients obtenus lors de la modélisation initiale.

### Paramètre « mission »

Le paramètre mission dans l'URL (ex : ?mission=test) permet de spécifier le projet à utiliser. En fonction de ce paramètre, l'application récupère les fichiers **raw.xlsx**, **results.xlsx**, et **model.pkl** à partir du répertoire correspondant. Cela permet de réutiliser l'application pour différents projets sans modification du code, en ajustant simplement le répertoire des fichiers.

### Interface utilisateur et rendu HTML personnalisé

L'interface utilisateur est enrichie avec des éléments HTML et CSS pour un rendu visuel personnalisé. Par exemple, les résultats de la simulation (comme le ROI) sont présentés dans des cartes stylisées qui utilisent des bordures, des ombres et des tailles de texte spécifiques défini avec MediaROI et selon sa charte.

### Gestion de l'état avec session\_state

L'application utilise le mécanisme de **Streamlit session\_state** pour stocker et gérer les données tout au long des interactions utilisateur. Cela permet de charger le modèle une seule fois par session et de sauvegarder les budgets et les résultats des simulations.

* **Chargement du modèle et des données** : Lorsque l'application démarre ou que le paramètre mission est modifié, le modèle et les données associées sont chargés dans session\_state pour optimiser les performances et éviter les chargements répétés.
* **Stockage des budgets soumis** : Les valeurs saisies ou ajustées par l'utilisateur (budgets médias) sont enregistrées dans session\_state, garantissant la persistance des données entre les interactions.

### Gestion des canaux médias

L'application permet de simuler les budgets pour différents canaux médias prédéfinis. Les canaux disponibles sont fixés dans une liste, à savoir :

* TV
* OOH (Out-of-Home)
* Radio
* Print
* Search
* Display
* VOL (Vidéo Online)
* Social

L'utilisateur peut choisir d'ajuster les budgets de ces médias soit en **valeurs absolues**, soit en **pourcentages** grâce à un booléen nommé usePerc. Si un média n’est pas utilisé dans un projet particulier, il apparaît en **grisé** dans l’interface, indiquant qu'il n'est pas actif pour ce projet.

Cette approche permet à l'utilisateur de simuler des ajustements de budget tout en offrant la flexibilité d'utiliser ou non certains canaux en fonction des besoins du projet.

Dans la partie « Advanced settings », l’utilisateur peut aller encore plus loin en définissant l’allocation de chaque sous-canal (régies, chaines de TV ou autres en fonction de comment les canaux médias sont structurés).

### Optimal budget et simulated budget

Par défaut, l’application propose à l’utilisateur la même allocation que celle précédemment faite (pas de changement). En cliquant sur le bouton « Use optimal », les allocations de budget sont celles qui ont été calculés comme optimal à la fin du notebook de modélisation.

ATTENTION : Ces valeurs optimales sont valables pour un temps donné (10 semaines, 45 jours etc). L’application du simulateur ne permet pas de recalculer un optimal dans l’app car une itération d’optimisation prend entre 30 et 45 minutes.

### Exportation des résultats au format CSV et PDF

L’application permet à l'utilisateur de télécharger les résultats de la simulation sous deux formats :

1. **Exportation au format CSV** :
   * Une fois la simulation terminée, les résultats (les prévisions de performance en fonction des budgets ajustés) peuvent être convertis en fichier CSV.
2. **Génération d'un rapport PDF** :
   * L'application génère également un **rapport PDF** à partir des visualisations générées pendant la simulation. Les images des résultats (comme des graphiques) sont rassemblées dans un PDF.
   * Pour créer les visualisations du PDF,

### Visualisation des résultats

Les résultats de la simulation, qu'il s'agisse du ROI, des prévisions de ventes ou d'autres indicateurs, sont affichés dans des tableaux et des graphiques interactifs. L’utilisateur peut visualiser l’impact de ses ajustements de budget en temps réel, grâce aux recalculs automatiques réalisés par le modèle.

### Gestion des erreurs

L’application vérifie la présence des fichiers attendus (tels que **raw.xlsx**, **model.pkl**) et alerte l’utilisateur en cas de problème. Si un fichier est manquant ou mal configuré, un message d’erreur s’affiche pour permettre à l'utilisateur de corriger le problème.

## Pour le déploiement

Pour déployer le simulateur nativement sur GCP, il faut ajouter une partie dans l’app permettant de connecter celle-ci au storage google. Cela permettra ensuite de diriger l’app vers les fichiers issus des notebooks de modélisation en suivant l’arborescence voulu.   
  
S’il fallait rendre plus modulable la recherche des documents, on peut rajouter plus d’arguments dans la requête de l’url et ainsi récupérer des fichiers plus précisément.