**AI-Driven Media Investment Plan Across Channels for E-commerce**

1. **Introduction:**

E-commerce businesses must carefully manage their budget allocation across various paid media channels to enhance customer acquisition and maximize conversion rates. This task involves a detailed analysis of historical performance data to understand how each channel contributes to overall success. Predictive modelling is employed to determine which channels have the highest potential for boosting return on investment (ROI), enabling businesses to make informed decisions about where to allocate their resources.

To effectively address this challenge, four datasets were leveraged to forecast revenue and support strategic budget reallocation. Machine learning techniques were utilized to analyse historical data and predict the outcomes of different budget scenarios. This data-driven approach facilitated the reallocation of budgets to channels that are projected to deliver the highest ROI. As a result, the overall campaign performance was optimized, leading to improved revenue and more efficient media investments.

1. **Libraries and Versions:**

Libraries used in our solution are mentioned below:

* %pip install pandas==1.3.3
* %pip install scikit-learn==0.24.2

**Pandas**: Version = 1.3.3

**scikit-learn**: Version = 0.24.2

1. **Input Section**

**New Budget as Input**: The total budget for media allocation.

**Select and Read Dataset**: The solution leverages the NetElixir Algnition Dataset 1, NetElixir Algnition Dataset 2 which encompasses performance data from Google, Microsoft, and Meta ad platforms, alongside website landing data.

1. **Approach and Methodology**
2. **Data Collection**

**Datasets**: Four datasets were collected, containing historical data on media spend, conversion rates, and revenue across various paid media channels:

* googleads-performance.csv
* microsoftads-performance.csv
* metaads-performance.csv
* website-landings.csv

1. **Data Processing:** In this data preprocessing step, we added columns, concatenated dataframes, mapped dataframes. Encoded categorical features like 'Campaign type' and 'Source\_Category' using **LabelEncoder,** and also performed data splitting, scaling etc.

* **Adding Source\_Category Column**

Added Source\_Category to google, microsoft, and meta DataFrames to indicate the source of the data.

* **Concatenating DataFrames**

Combined google, microsoft, and meta DataFrames into a single DataFrame named ads\_data.

* **Label Encoding**

Applied LabelEncoder to convert categorical columns into numerical format for further analysis.

Divide X and y. X contains independent features and y contains dependent feature. In our case y contains Revenue

* **Data Splitting:**

X: Contains independent features.

y: Contains the dependent feature, specifically the revenue.

* **Scaling:**

Standardizing the numerical features in X and the target variable y (Conversions) by scaling them to have a mean of 0 and a standard deviation of 1. This is done using StandardScaler, with the transformed data stored back in X and y.

* **Splitting:**

Splitting the dataset X and target variable y into training and testing sets, with 80% of the data used for training and 20% for testing. The random\_state=42 ensures that the split is reproducible.

1. **Model Building**

Here, we used the RandomForestRegressor from the sklearn.ensemble library to predict revenue based on input features.

Random Forest Regression is a Machine Learning algorithm which is very accurate, efficient and can handle complex datasets. It also reduces the risk of overfitting. It combines multiple decision trees to create a single model. Each tree in the forest builds from a different subset of data and makes it prediction independent.

For our model we choose random\_state value 42 which ensures that the split is reproducible.

The trained model predicts Conversions across different campaigns.

**Prediction and Evaluation**

Predict the target variable y for the test set and evaluate the model’s performance using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

**Budget Allocation by Source Category and Campaign Type**

The weight is calculated as the number of conversions for each source category divided by the total number of conversions across all source categories. The total budget is divided proportionally according to each source's conversion weight. If any source receives zero budget (likely due to having zero conversions), a minimum budget is allocated (1% of the total budget). After the initial allocation, if any budget remains, it is distributed proportionally among the source categories based on their conversion weight. The total allocated budget is checked to ensure it sums up to the original total budget. If there is a discrepancy, it adjusts the final budget.

**Calculating and Merging Total Conversions by Source and Channel**

In this step we done grouping data by Source\_Category and Channel to calculate the total number of conversions for each combination. Then, the total number of conversions for each Source\_Category is merged back into the grouped data, making it easier to apply further calculations.

**Calculating Conversion Weight and Initial Budget for Channels**

After merging the total conversions for each source and channel, the next step is calculating the budget for each channel within the Source\_Category. This is done based on the proportion of conversions within each channel relative to the total conversions of its source. Paid channels are subject to a minimum budget constraint to ensure they receive at least 10% of the source’s final budget. This section checks for channels labelled as paid or paid search and enforces the minimum budget requirement. Finally, we calculated the total budget allocated to each Source\_Category after applying the minimum constraints. Any remaining budget is then identified to ensure further adjustment.

The code checks whether the total allocated budget exceeds the original budget. If it does, the allocations are scaled down proportionally to ensure the final budget doesn't exceed the defined limit. Otherwise, the calculated channel budget is kept as it is.

**Estimating Conversions:**

Here we estimated the conversions for ads that have non-zero clicks by calculating relevant metrics such as the **Cost to Clicks Ratio** and **Cost per Impression**. These ratios are calculated for further analysis. Then, merges ad data with budget allocation data for further analysis. It also handles missing or infinite values in the dataset to ensure smooth calculations. We ensured that the budget ratios are applied and capped where necessary to avoid extreme values. This ensures that the budget is proportionally allocated based on the campaign’s previous performance while applying a limit to the highest budgets. After data cleaning and transformation, the next step is to create the final Data Frame that contains the input features necessary for analysis. These features are critical for the model to make predictions.

Once the data is prepared and cleaned, a trained machine learning model is used to predict conversions. The predicted conversions are then inverse-transformed to return them to the original scale. This final step groups the predicted conversions by Source\_Category and Campaign type to display a summary of the total predicted conversions for each group.

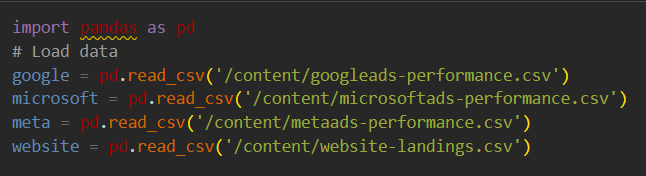
1. **Assumptions**

For the NetElixir Algnition Dataset 1, which contains four datasets, the Meta Ads dataset did not include a 'Campaign type.' To address this, the 'Campaign type' for Meta Ads was assumed to be 'General.'

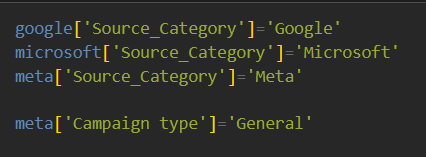
1. **Algorithm Implementation:**

The code for data cleaning, preprocessing, and algorithm implementation is as follows:

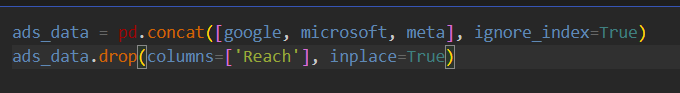
* **Data Loading:**



* **Data Preprocessing:**

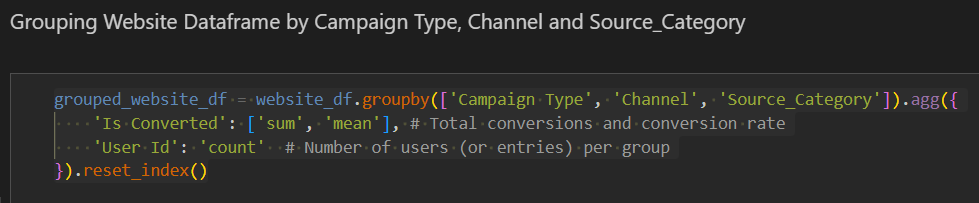
****

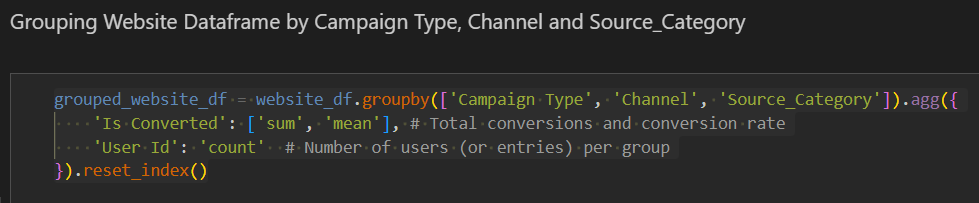
**Concatenating dataframes:**

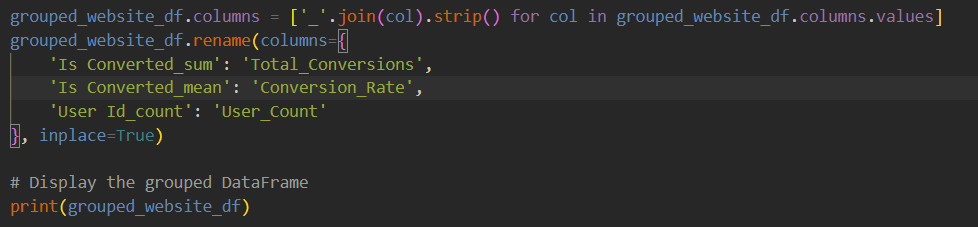
****

**Mapping Source\_Category in website DataFrame:**

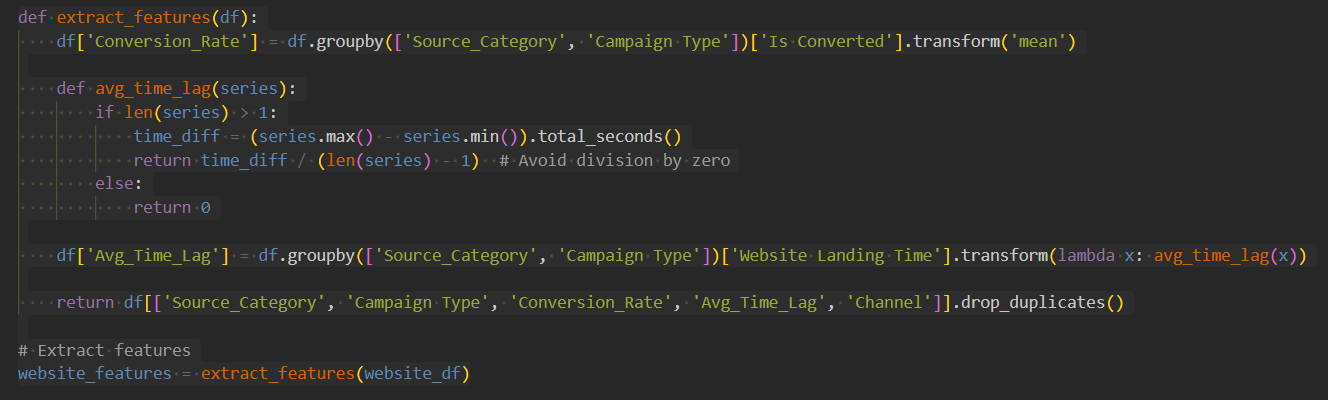




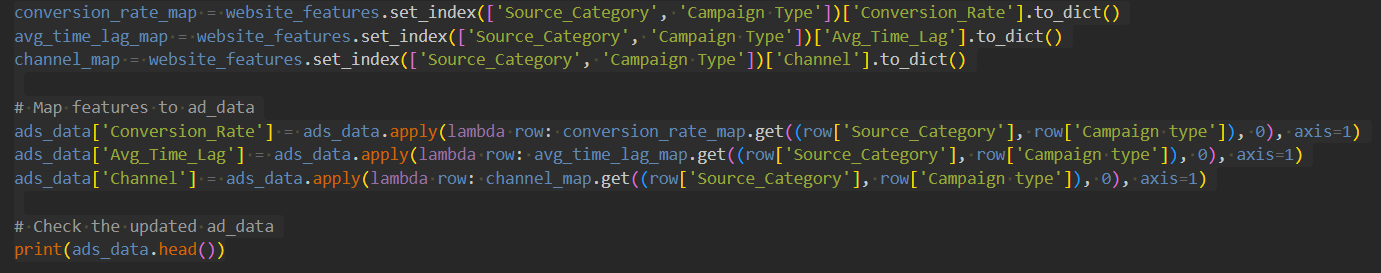




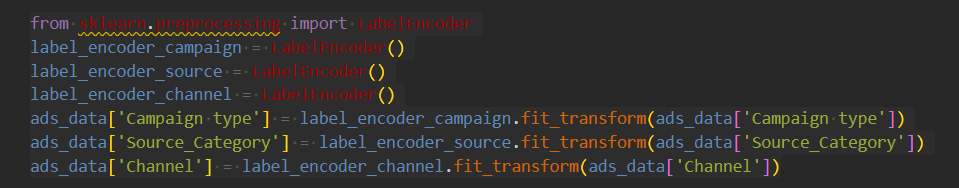
**Feature Extraction from Website Data:**

****

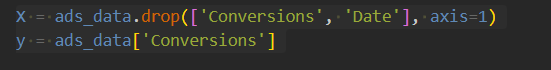
**Mapping Features to Ads Data**

****

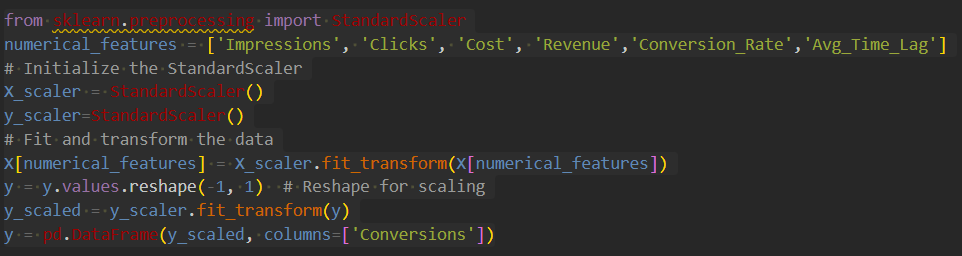
**Label Encoding:**

****

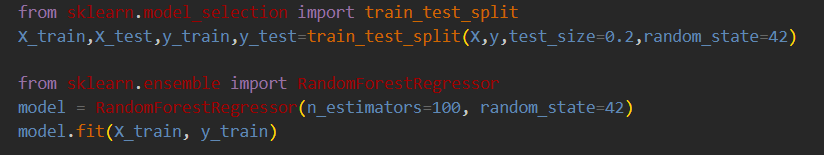
**Data Splitting:**

****

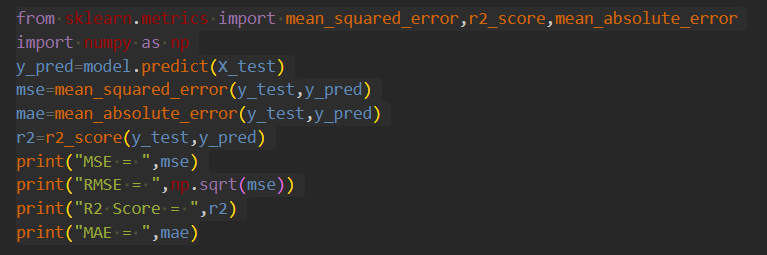
**Scaling:**

****

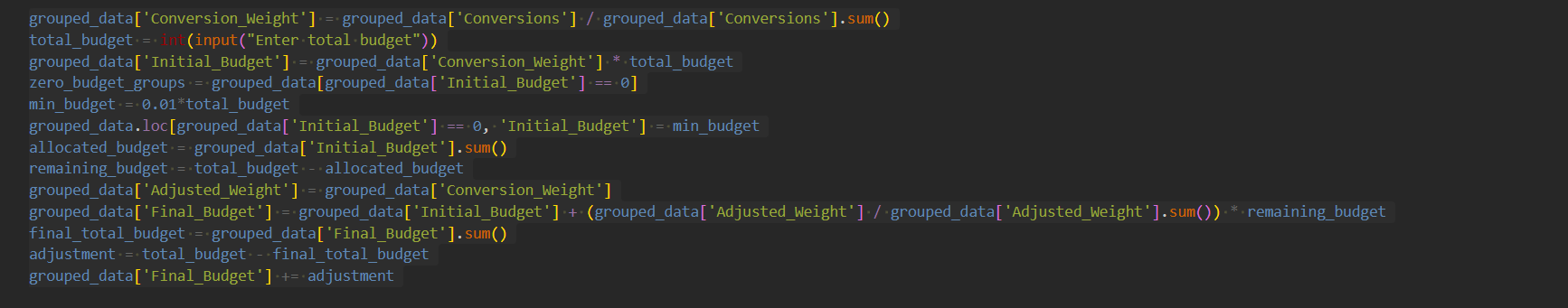
* **Algorithm:**



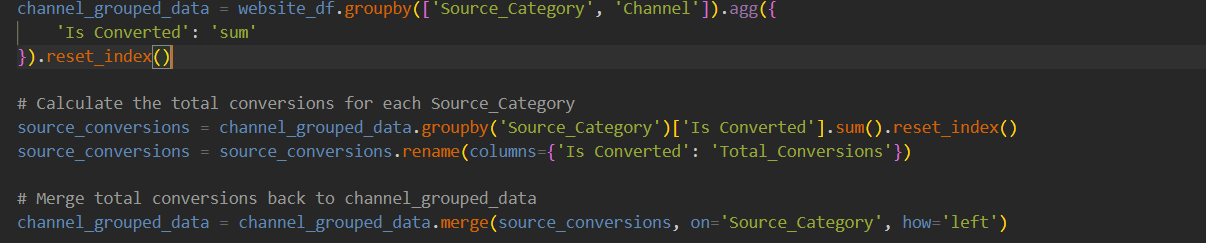
**Evaluation:**

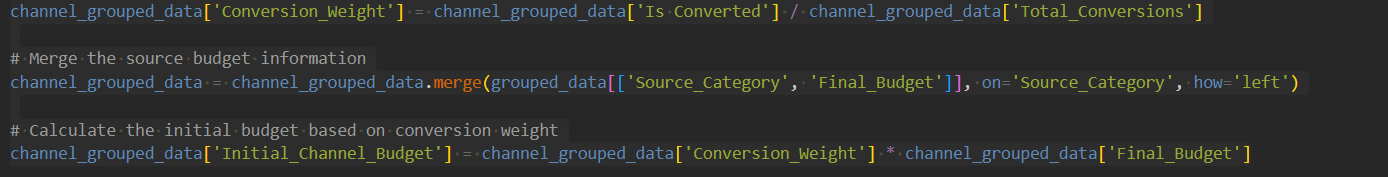
****

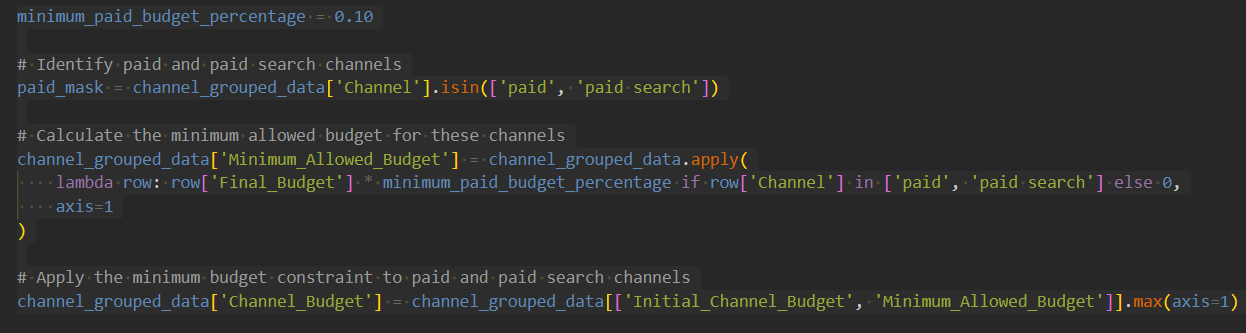
**Budget Allocation:**

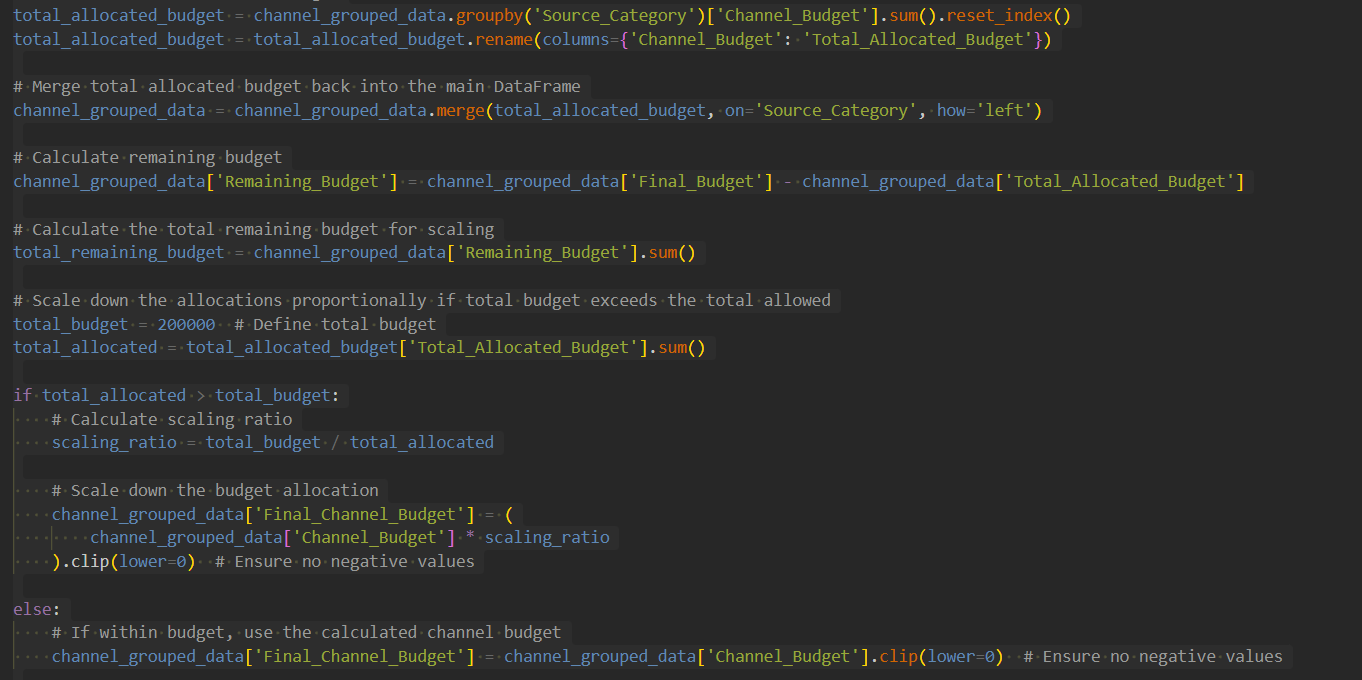
****

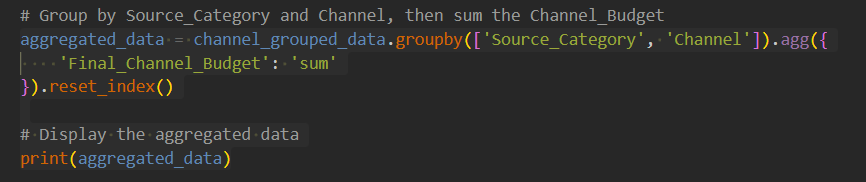
**Total conversions by Source and Channel:**

****

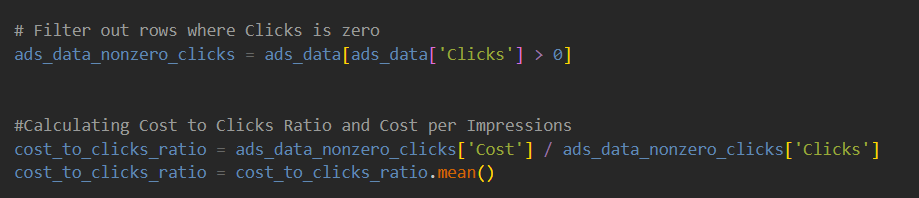
****

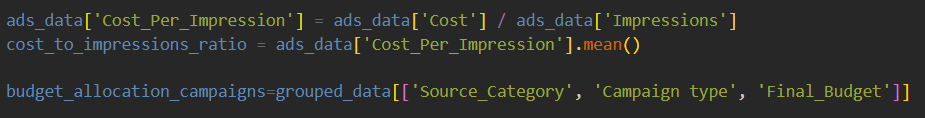
****

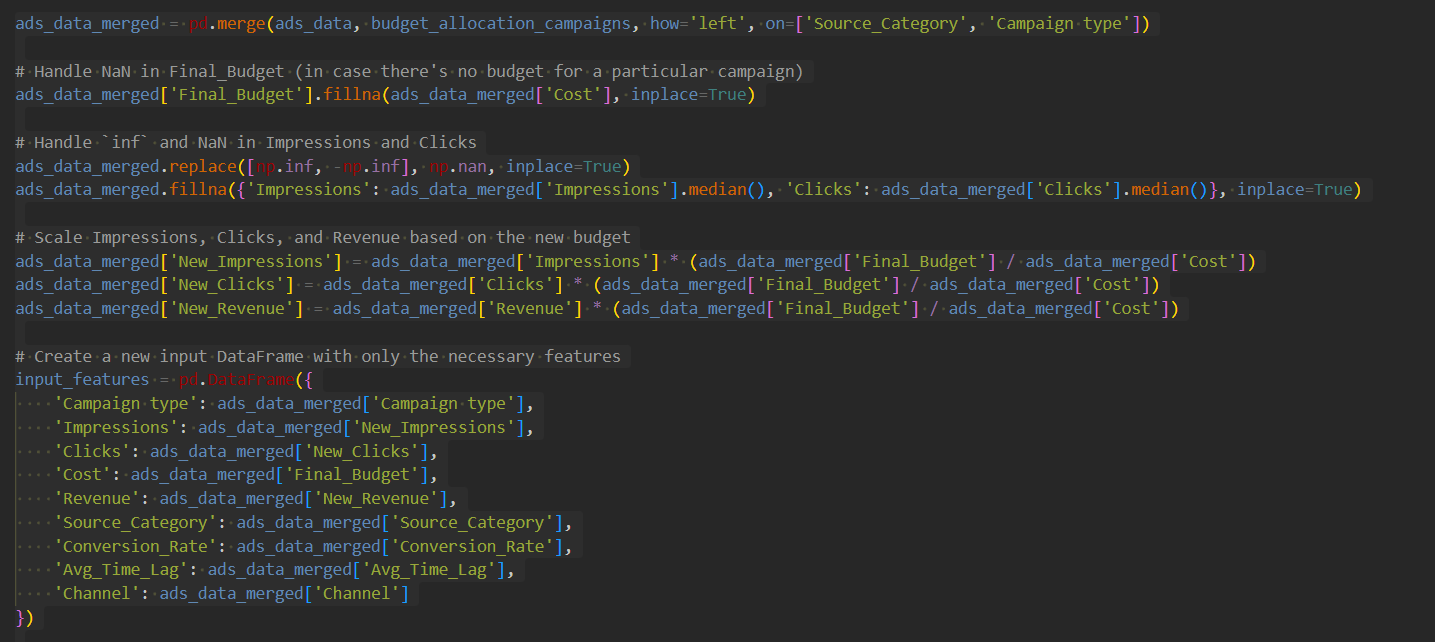
****

****

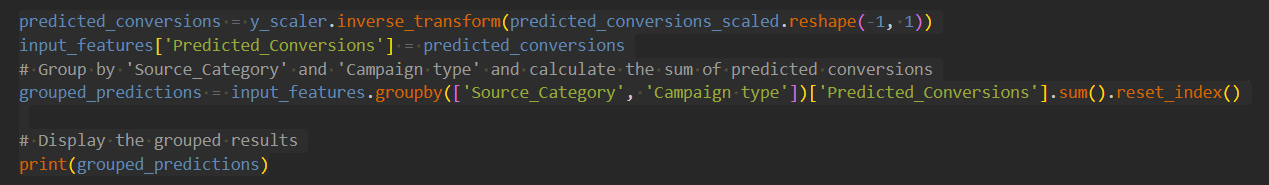
**Estimating Conversions:**

****

****

****

****

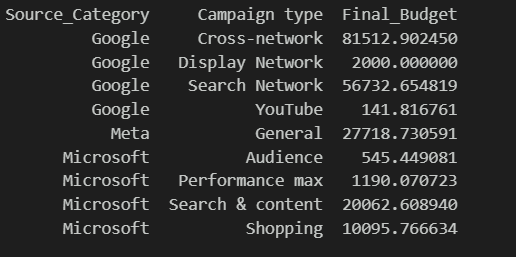
****

1. **Results:**

**Input:** 200,000

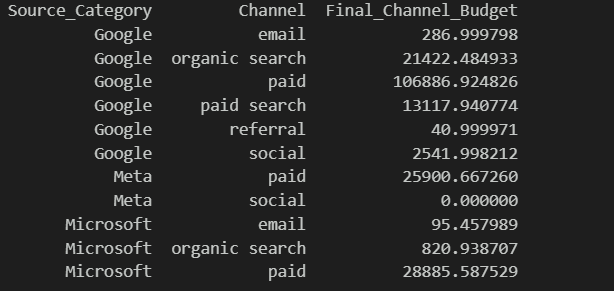
**Output:**

Budget Estimation by source category and campaign type:



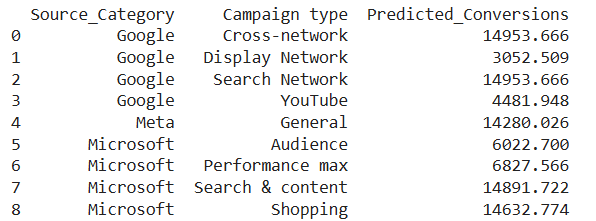
Total Budget: 200000.0

Budget Estimation by source category and Channel:

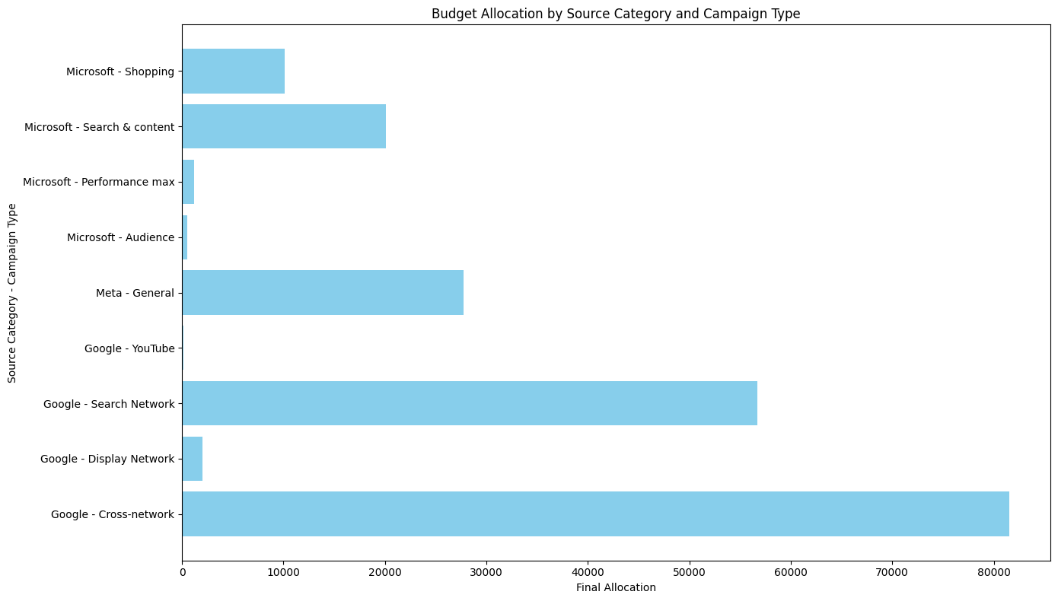


Total Budget: 200000.0

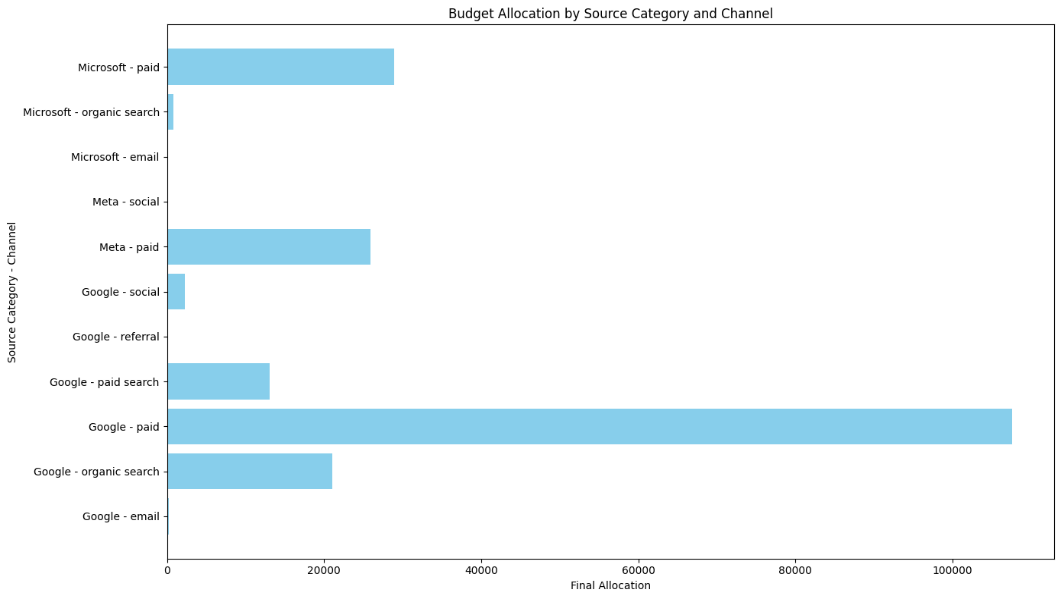
**Estimated Conversions:**

****

**Bar Chart for Budget Allocation for Source and Campaign Type:**



**Bar Chart for Budget Allocation for Source and Channel:**



1. **Conclusion:**

This project successfully optimized budget allocation across various paid media channels to maximize conversions and revenue by integrating and processing data from multiple sources. Using a Random Forest Regressor model, we predicted conversions and allocated the budget based on conversion weights, ensuring minimum funding for all channels and adhering to constraints for paid channels. The approach included comprehensive data preprocessing, model evaluation, and conversion estimation, which resulted in actionable insights and improved budget distribution. Overall, this project demonstrates the power of data-driven decision-making in enhancing and optimizing media spend.

1. **References:**
2. Scikit-learn Documentation: <https://scikit-learn.org/stable/>
3. The Current State of Media Budgeting: <https://www.ismartcom.com/blog/optimizing-ad-spend-how-ai-is-transforming-media-budgeting>
4. Pandas Documentation: <https://pandas.pydata.org/pandas-docs/stable/>
5. Matplotlib Documentation: <https://matplotlib.org/stable/index.html>
6. <https://www.researchgate.net/publication/380178156_Machine_Learning_in_Marketing_Analytics>
7. Ma, L. and Sun, B. "Machine learning and AI in marketing–Connecting computing power to human insights." International Journal of Research in Marketing, 2020.

**Note:**

We created some visualizations using PowerBi and we added those files in “Visualizations” folder (Visualizations/PowerBi). to this zip file. Those PowerBi files will be accessed only when the user also has PowerBi in their system. So, we also included the output images of the PowerBI files in “Visualizations” folder (Visualizations/Images).