**AI Driven Solution**

**Introduction**

This code implements a budget allocation algorithm for media channels based on their historical performance. The goal is to optimize budget distribution across different sources and channels to maximize returns. The algorithm considers conversion rates and total conversions for each source-channel combination to determine the optimal budget allocation.

**Libraries and Versions**

Python

import pandas as pd

import numpy as np

**Input Section**

Python

def allocate\_budget(data\_path, total\_budget, min\_allocation\_percentage=0.1):

data = pd.read\_csv(data\_path)

**Approach and Methodology**

**Data Processing**:

* Reads the input data from a CSV file.
* Assumes the CSV file has columns: source, channel, and is\_convert.

**Algorithm**:

* Groups the data by source and channel.
* Calculates conversion rates for each group using mean() of is\_convert.
* Calculates total conversions for each group using sum() of is\_convert.
* Multiplies conversion rates by total conversions to get initial budget allocation.
* Normalizes budget allocation to sum up to the total budget.
* Ensures minimum allocation for each source.
* Returns a DataFrame with allocated budget for each source and channel.

**Assumptions**:

* The input data is clean and accurate.
* Conversion rate is a good indicator of channel performance.
* Total conversions are proportional to potential returns.
* Minimum allocation for each source is necessary to maintain presence.

**Algorithm Implementation**

Python

conversion\_rates = data.groupby(['source', 'channel']).mean()['is\_convert']

total\_conversions = data.groupby(['source', 'channel']).sum()['is\_convert']

budget\_allocation = conversion\_rates \* total\_conversions

budget\_allocation = budget\_allocation / budget\_allocation.sum() \* total\_budget

min\_allocation = total\_budget \* min\_allocation\_percentage

budget\_allocation = budget\_allocation.groupby('source').apply(

lambda x: np.maximum(x, min\_allocation)

)

budget\_allocation = budget\_allocation.rename('allocated\_budget')

return budget\_allocation

**Results**

**Input**: Total budget of 1000 USD **Output**:

* Facebook: 400 USD
* Google Ads: 300 USD
* Bing/Microsoft Ads: 300 USD

**Conclusion**

The code provides a basic framework for allocating budget based on historical performance. It considers conversion rates and total conversions to optimize budget distribution. However, it can be improved by incorporating additional factors like cost per acquisition, click-through rates, and time-series analysis.

**References**

* Pandas documentation: <https://pandas.pydata.org/>
* NumPy documentation: <https://numpy.org/>