**Source Partner Opportunities- Corridor Expansion Analysis**

Overview:-This project is a two-part initiative that focuses on both the source-partner side and the destination-partner side of the remittance process. The remittance flow involves several key entities: the sender customer, source-partner, TerraPay, destination-partner, financial institution, and the receiver customer. The project outlines the cross-border transfer process, where funds are securely transferred from one country to another. It aims to streamline and enhance the efficiency of cross-country payments by improving collaboration between financial institutions and partners across different regions. Through this process, we ensure smooth, secure, and prompt transactions that meet regulatory standards and customer expectations. Additionally, the project will find any potential gaps or challenges in the workflow, providing solutions to perfect the overall remittance experience.

Financial Institution

Sender Customer

Cu

Source Partner

Destination Partner Side

Source Partner Side

TerraPay

Receiver Customer

Destination Partner

Problem Statement:- **Partner Sustenance - Source Partner Side**

The aim of this project is to find and expand potential remittance corridors for the existing source-partner in a specific country, in collaboration with TerraPay. While the source-partner has set up operations in several corridors, there is a need to explore and evaluate new corridors for expansion, as well as assess and improve existing corridors with underperforming transaction volumes. A corridor, in this context, refers to the relationship between the source-partner in the origin country and the destination country.

This initiative will focus on identifying corridors with robust growth potential, based on factors such as the gross transaction value and the current volume of TerraPay's business in the destination country. For corridors that are already operational but have lower transaction volumes, the project will look to identify strategies to increase their value. This may include enhancing partnerships, improving marketing efforts, improving service offerings, or addressing barriers that could be limiting the growth of these corridors.

Destination Country

Source Partner

Corridor

APPROCH:-

Data Collection:-

The first step of the project involved collecting data on TerraPay’s remittance flows from the source country to the destination country using the Superset dashboard and the database table (Db) datasets. This data provided insights into the Gross Sent Value (GSV) and transaction volume activity across existing corridors. After gathering the data, we saved it as a CSV file named TP\_B.csv for further analysis. This file will serve as the foundation for evaluating and improving TerraPay’s remittance corridors.

Continuing the data collection process, we gathered additional data from the source-partner side. This included information on the source-partner, source country, destination country, and the Gross Sent Value (GSV) of transactions. This data provided a clearer picture of the remittance flow from the source country to the destination country, focusing on the source-partner's involvement. The collected data was then saved in a CSV file named P\_B.csv for further analysis, which will help in evaluating and optimizing TerraPay's remittance corridors.

As part of the data collection process, we also gathered information on source\_country and the corresponding Regional Director name in CSV format named as R\_D.csv from the database table (Db) in Superset. This was critical for understanding the regional management and context of the remittance corridors.

**Methods Used:**

1. **Sorting Data**:

The data from both CSV files (**TP\_B** and **P\_B**) was sorted by the **source\_country**. This sorting step allowed for a more organized and efficient comparison and analysis of the remittance data across the source and destination countries.

1. **Filtering Rows Based on GSV**:

A condition was applied to filter out rows where the **Gross Sent Value (GSV)** was less than **10,000**. Rows with GSV values below this threshold were either dropped or retained based on the specific analysis requirement. This filtering step was applied consistently across both **TP\_B** and **P\_B** datasets to focus on higher-value remittance data.

1. **Matching and Dropping Common Pairs**:

For each selected **source\_partner** and **source\_country**, we checked for matching **source\_country-destination\_country** pairs between **P\_B** and **TP\_B**. Any rows in **TP\_B** that had a matching **source\_country-destination\_country** pair from **P\_B** were dropped to eliminate duplicate data.

1. **Selecting Top 10 Rows**:

After removing the common pairs, the top 10 rows from **TP\_B** were selected based on the highest **GSV** values. For each of these top rows, the **source\_partner** name was added to the data for clarity and reference.

1. **Managing Duplicate Source-Destination Pairs**:

If any of the top 10 rows had the same **source\_country** and **destination\_country** pair, that row was skipped to ensure diversity. The next available row with a unique **source\_country-destination\_country** pair was selected.

1. **Joining Regional Director Data**:

Finally, the **Regional Director** names were joined to the data from a separate **RD.csv** file. This file contains mappings of **source\_country** and their respective **Regional Director** names. The **Regional Director** name was added at the end of each selected row to provide additional context for the analysis.

DRY RUN:

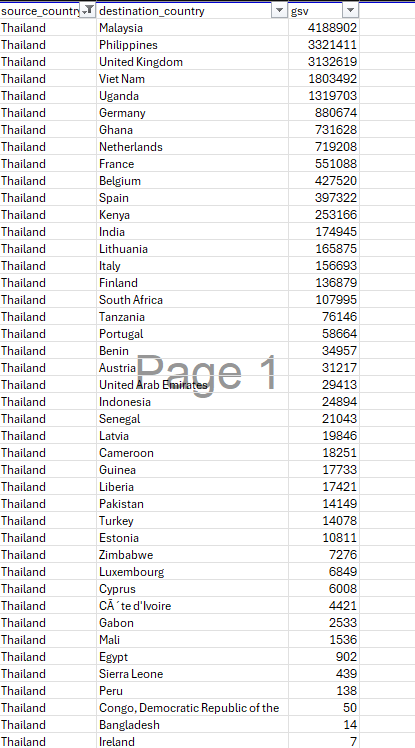
To validate and prove our approach, we conducted a **dry run** using a specific **source\_partner**: “**Smart”** . We followed the exact steps outlined in our approach to ensure the correctness and effectiveness of the methods applied.

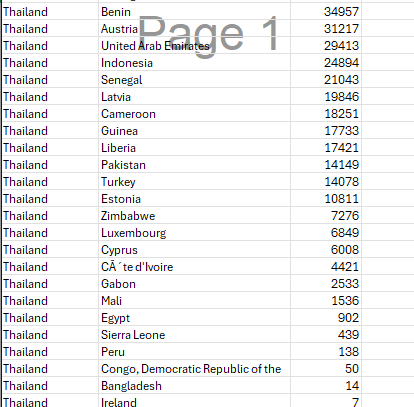
1. **Data Filtering for Source Partner**: From P\_B.csv



1. **Sorting the Data**:

From TP\_B sort by Source\_country of source\_partner i.e., Thailand





1. **Filtering Rows Based on GSV** :i.e. IF GSV>=10,000 then we keep it

A screenshot of a computer

Description automatically generated From P\_B .csv after filtering

 From TP\_B after filtering

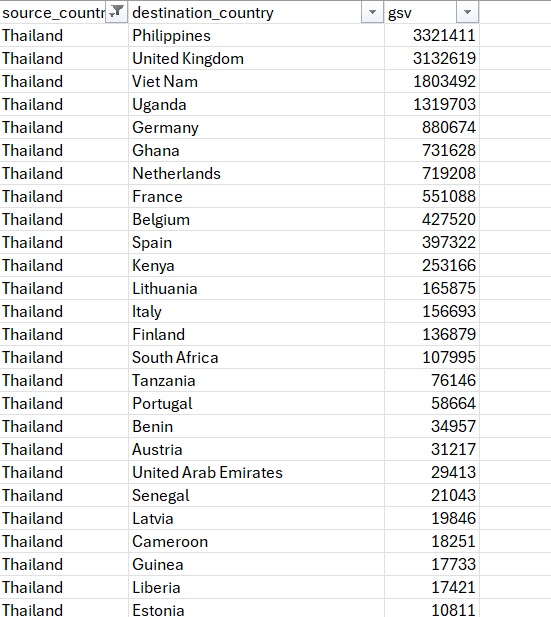
1. **Matching and Dropping Common Pairs**:

**Matching**:

 Rows Highlight in Yellow indicate that source-destination Countries are common in both TP\_B and P\_B(“Smart”)

That indicates that their corridor exists for the partner.

**Dropping:**

**** This is the list of potential Source-Destination Countries were TerraPay is currently dealing with remittance which has same source\_country as the Source-partner has.

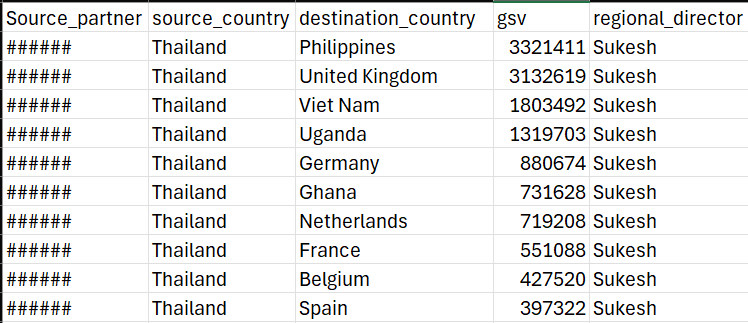
1. **Selecting Top 10 Rows**:

A screenshot of a computer

Description automatically generated The top 10 Destination countries are the potential for the source\_partner to establish new corridors with TerraPay to expand its growth.

1. **Overseeing Duplicate Source-Destination Pairs**: As there are now duplicate source and destination countries in the list.
2. **Joining Regional Director Data**:

SMART - Sukesh From RD.csv



This is the result that has the potential Destination-Countries where the source\_partner can start business with TerraPay with the regional director’s name.

**Final Approach: Final Approach**

1. **Data Loading and Cleaning:**

This documentation outlines the process for loading and cleaning CSV files for datasets **P\_B** and **TP\_B** using the **clean\_and\_load\_csv()** function, which removes any leading or trailing whitespace from column names for consistency. The function takes a file path as a parameter and returns a cleaned panda DataFrame. In a Jupyter Notebook, users can upload the CSV files directly to the Jupyter directory and load them accordingly.



In a production environment, users should specify the full path to the CSV files based on their location in the filesystem. By following these steps, users can effectively load and clean the **P\_B** and **TP\_B** datasets, ensuring that the data is ready for analysis with consistent column names

A computer code with text

Description automatically generated with medium confidence

1. **Filtering Based on GSV:**

Filter rows from both DataFrames based on the GSV column. Any row with a GSV less than 10,000 is excluded using the filter\_gsv() function.

**filter\_gsv(df1, df2, threshold=10000)**

Filters two DataFrames based on the 'gsv' column.

**Parameters:**

* **df1** (**pd.DataFrame**): The first DataFrame to be filtered.
* **df2** (**pd.DataFrame**): The second DataFrame to be filtered.
* **threshold** (**int**, optional): The threshold value for filtering the 'gsv' column. Default is 10,000.

**Raises:**

* **KeyError**: If either DataFrame does not contain a 'gsv' column.

**Returns:**

* **tuple**: A tuple containing the filtered DataFrames.

**Example:**



1. **Filtering Top Rows:**

For each source\_partner, filter the rows in TP\_B to get the top 10 rows for each source\_country, based on the highest GSV values using the filter\_and\_display() function. This ensures that rows with the same source\_country and destination\_country pair are skipped.

**filter\_and\_display(df5, source\_partner)**

Filters **df5** to get the top 10 rows for each source country based on 'gsv'.

**Parameters:**

* **df5** (**pd.DataFrame**): The DataFrame to be filtered.
* **source\_partner** (**str**): The source partner to be added to the DataFrame.

**Raises:**

* **KeyError**: If required columns do not exist in **df5**.

**Returns:**

* **pd.DataFrame**: The top rows for each source country.

**Example:**



1. **Comparison with Filtered Data:**

Compare the filtered rows with another DataFrame (filtered\_df1) to ensure that only matching source\_country and source\_partner pairs are retained using the compare\_and\_filter() function.

1. **compare\_and\_filter(top\_rows, filtered\_df1)**

Compares **top\_rows** with **filtered\_df1** and keeps only matching rows.

**Parameters:**

* **top\_rows** (**pd.DataFrame**): The DataFrame containing top rows to be compared.
* **filtered\_df1** (**pd.DataFrame**): The DataFrame to compare against.

**Raises:**

* **KeyError**: If required columns do not exist in **filtered\_df1**.

**Returns:**

* **pd.DataFrame**: The filtered top rows.

**Example:**



1. **Merging Duplicate Rows:**

Merge duplicate rows from the filtered results by aggregating GSV values for identical source\_partner, source\_country, and destination\_country pairs using the merge\_duplicate\_rows() function.

1. **merge\_duplicate\_rows(final\_results)**

Merges duplicate rows based on **source\_partner**, **source\_country**, and **destination\_country**.

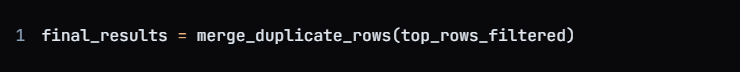
**Parameters:**

* **final\_results** (**pd.DataFrame**): The DataFrame containing results to be merged.

**Returns:**

* **pd.DataFrame**: The merged results with summed 'gsv' values.

**Example:**



1. **Adding Regional Director Information:**

Add Regional Director information to the final results by merging the filtered and merged rows with the R\_D.csv file, which contains source\_country and regional\_director data, using the add\_regional\_director() function.

1. **add\_regional\_director(merged\_results, r\_d\_file\_path)**

Adds a 'regional\_director' column to **merged\_results** based on matches in **R\_D.csv**.

**Parameters:**

* **merged\_results** (**pd.DataFrame**): The DataFrame to which the regional director will be added.
* **r\_d\_file\_path** (**str**): The path to the **R\_D.csv** file.

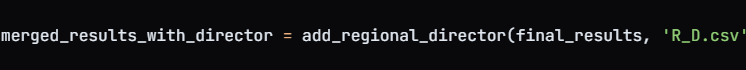
**Raises:**

* **KeyError**: If required columns do not exist in **R\_D.csv**.

**Returns:**

* **pd.DataFrame**: The merged DataFrame with the regional director information.

**Example:**



1. **Final Output:**

Iterate through the list of source\_partners, applying the above steps for each, and store all results in a single DataFrame. Finally, save the combined results to a CSV file (Potential\_partner.csv).

Output:

A CSV file containing the top GSV data for each source partner with added Regional Director information, free from duplicates and irrelevant data.

**Final result :** The result is stored in the Csv file named merged\_results\_final that contains top 10 potential corridor grouped by source\_partner it can necessarily be top 10 for every case depending in the existing TerraPay businesses .

