

ACCOUNT STATEMENT ANALYSIS - PYTHON

Objective

The objective is to comprehensively analyse an account statement by examining balances, transactions, and patterns to identify cash flow status, spending habits, significant transactions, and recurring payments. And transforming the statement tables to its maximum information extractable format.

Statement Overview

The statement table contain 10 named columns. They are 'Tran Date', 'Value Date', 'Tran Particulars', 'Type', 'Sub', 'Insr', 'Insr No', 'Debit', 'Credit', 'Balance'. And there is an unnamed column which have values like 'Cr'.

Tran Date is the column that contain the date of transaction happen. Value date is the date in which the transaction validated. In some cases, these two dates will be different. But in most times, those were same. Tran Particulars are the core part of this data. They were different for different transaction categories. But they have same pattern for the same transaction category. This transaction category can be found from this Tran Particulars. The Tran Particular contain some word tokens and numbers and their mixed forms. These parts were separated by a character '/'. When splitting it with the character we get some information like Transaction Category, Reference Table Number, MCC codes, UPI ID, Time Stamp, Location. For each transaction categories it has different combinations of these. For transaction the maximum number of elements in the Tran Particulars is found to be 5.

The column named 'Type' contain the information of tran type. They were found to be T, C. T for Transaction, C for Cash. 'Sub' column contains values like CI, BI which is the short form for Customer Initiated and Bank Initiated. The column 'Instr', 'Instr No' is for the Instrument name and Instrument number. The other columns are Debit, Credit, Balance which indicate the information's of Debit Amount, Credit Amount and the balance of the account.

Expanding statement records

I have separated the Tran Particular columns and created columns for each part in it. It helps to filter out the transaction of different categories. There is an mcc code in it. It were used to expand the transaction category by adding the details form the mcc table like code_desc, merchant, group_category.

Function To Analys

I have created some functions to analyse the statement:

1. **preprocess_statement_excel_file:** This function preprocesses the Excel file data. It drops specified rows, sets headers, removes unnecessary rows and columns with all NaN values, and assigns a specified header name to columns with NaN headers.
2. **expand_data:** This function expands the data by splitting the 'Tran Particulars' column into multiple columns, reorders columns, sorts the data based on 'Particulars_1', updates the 'Particulars_1' column for specific conditions, and matches codes to merchant groups based on the particulars columns.
3. **acc_statement_Report:** This function generates an account statement report. It calculates various metrics such as total transaction count, statement period, number of months within the period, counts of transaction categories, types, subtypes, total credit and debit transactions, and counts of credit and debit transactions for each category.
4. **calculate_credit_debit_counts:** This function calculates the total count of credit and debit transactions in the dataset.
5. **calculate_credit_debit_counts_by_transaction_category:** This function calculates the count of credit and debit transactions for each transaction category.
6. **separate_single_multi_transaction_per_days:** This function separates transactions into two tables based on whether there is a single or multiple transactions per day.
7. **separate_credit_debit:** This function separates credit and debit transactions into two tables.
8. **divide_data_by_tran_categories:** This function divides data into separate tables based on transaction categories.
9. **divide_data_by_months:** This function divides data into separate tables based on specified months.
10. **opening_closing_balance_month:** This function calculates the opening balance, closing balance, and balance difference for each month.
11. **add_credit_amount_bin_column:** This function adds a new column 'Credit Amount Bin' based on specified credit amount bins.

12. **add_debit_amount_bin_column:** This function adds a new column 'Debit Amount Bin' based on specified debit amount bins.
13. **get_credit_amount_bin_counts:** This function calculates the counts of each bin in the 'Credit Amount Bin' column.
14. **get_debit_amount_bin_counts:** This function calculates the counts of each bin in the 'Debit Amount Bin' column.
15. **calculate_average_monthly_balance:** This function calculates the average monthly balance.
16. **calculate_average_monthly_balance_with_counts_and_total_amount:** This function calculates the average monthly balance along with counts and total amounts credited and debited.
17. **filter_high_credit_transactions_quantile:** This function filters high credit transactions based on a specified quantile threshold.
18. **find_repeating_transactions_across_months:** This function identifies repeating transactions across different months based on transaction characteristics.