**Objective**

Hyper-personalization: From Thoughts to Reality. Our goal is to use LLMs on users transaction data to determine their spending habits. With this additional information, we offer personalized services such as financial health assessments, product suggestions, and more.

**Rationale**

1. Users experience personalized services in every other aspect of life such as Online shopping, Streaming etc.
2. ABN AMRO bank doesn’t offer such experience at the moment.

**Description**

The basic approach is to

1. Prepare sample data: Here account statements are used as base. Using simple excel formulas, we extract the Payment method and Merchant name. Then payment type is manually categorized ( Update how to do this for larger set of 5000 -10K entries). Output is Transaction date, Amount, Debit/Credit, Payment method, Merchant and Type.
2. The sample data is used to train the model. The model makes a link between the Merchant and the type. It also use other patterns such as Date, Time etc.
3. The model is created as a utility to be used in any service.
4. Once the model is ready, customer data for x months is used as a base and categorized the expenses and store it a database.
5. The financial health API used this DB to create a personalized overview containing details such as Total income, Planned expenses, expense type with optional/mandatory indicator and overall health indicator.
6. The product proposition API uses predefined rules to make product suggestions to the customers. One such example
   1. If customer has a sudden hike for 3 months in buying products associated with having kids, then they get info on Jongenrekening, Toekmost rekening etc.

**Technical Approach**

1. SQL DB
   1. Main database to store account overview. Fields are

|  |  |  |
| --- | --- | --- |
| **Col** | **Filled** | **Remarks** |
| AccountNumber | Input file |  |
| Date | Input file |  |
| D-C-ind | Input file  D – Debit  C- Credit |  |
| Amount | Input file ( 0,00) | All amounts are in Euros only |
| PaymentMethod | Input file | Determined in the file with a simple excel logic |
| Merchant | Input file | Determined in the file with a simple excel logic |
| DescriptionLine | Input file | Actual description line |
| ExpenseType | Derived | Using the trained model |
| ExpenseCategory | Derived | Using the configuration |

1. LLM Service:
   1. Objective

This is main services which uses sample data to train the model. In the sample data, the expense Type is manually determined. The model trains not only on the Description line but also on Amount/Date ( **to be checked if its possible and add values**).

* 1. Logic

1. Transaction analysis service
   1. Objective

The service reads the actual transaction data and then each entry uses the LLM service to determine the Expense type. The category is then added based on the configuration. The complete output is stored in the database for further usage by the API.

1. Financial health API
   1. Objective

The financial health API uses the transaction analysis service output and give a detailed view of the financial health of the customer. The API provides a comprehensive view as well as a granular view of various expense type and categories.

* 1. Input

Account Number

Date Range ( Start/ End) -> For simplicity we can always consider the start date as first date of the month.

* 1. Logic

Using the input account number and date range, the API retrieves all the entries corresponding to these keys. The values are determined as mentioned below.

* 1. Output

|  |  |
| --- | --- |
| Field | Description |
| AccountNumber | Filled with value provided in the input |
| Date Range | Filled with value provided in the input |
| HealthStatus | Overall Health status. Based on the value of Total expense, Total planned expense and total actual expense.  Red - Expense more than income by x%  Amber – Slight overrun  Green – No overrun |
| Total Income | Sum of all Credits on the account for the month  ( Fixed start/end date) |
| Total planned expense | Average sum of all debits on the account for last x months.  ( Example -If account had data for 8 months with total debit of 16000 then 2000 as value) |
| expense Type | Sub element under Total planned expense. Example – Grocery, Dining etc |
| expense Type  amount | Sub element under Total planned expense, actual value of the expense Type |
| expense category | Sub element under Total planned expense, Category of the expense type such as Mandatory/Optional |
| Total Actual expense | Sum of all debits of that particular month |
| expense Type | Sub element under Total actual expense. Example – Grocery, Dining etc |
| expense Type  amount | Sub element under Total actual expense, actual value of the expense Type |
| expense category | Sub element under Total actual expense, Category of the expense type such as Mandatory/Optional |
| Advise | “To be determined” |

Example:

accountNumber : NL72ABNA0547587987

dateRange: 01/09/2024 - 18/09/2024

healthStatus: Amber

totalIncome: 4000

totalPlannedExpense: 3800

   type: grocery

   amount: 500

category: mandatory

   type: dining

   amount: 400

category: optional

   type: utilities

   amount: 600

category: mandatory

   type: others

   amount: 2300

category: //

totalActualExpense: 3100

   type: grocery

   amount: 400

category: mandatory

   type: dining

   amount: 500

category: optional

   type: utilities

   amount: 500

category: mandatory

   type: others

   amount: 1700

category: //

advise: moderate saving needed

1. Product proposition API
   1. Logic
   2. Input
   3. Output

**Other considerations**

1. Rabo bank has a similar API to give you an insight in your expenses.
2. A good addition could be to give customers an option to categorized and set their own preference.