Junior Data Scientist Case Study

Topic: Personal Financial Management (PFM) Insights + Customer Churn Risk

Business context

You are working in a digital bank that offers a Personal Financial Management (PFM) feature inside the mobile app. The bank wants to use transaction data to generate useful PFM insights for customers and to proactively reduce churn. Your team has prepared a transactional dataset (single CSV) that contains both customer profile attributes and transaction history.

Goal

Deliver two outcomes: (1) PFM analytics that summarize customer spending and cashflow patterns, and (2) a baseline churn risk model that predicts whether a customer will churn in the next 30 days (**churn_next_30d**). You must show model performance using a confusion matrix (and any additional metrics you consider appropriate).

Files provided

- **pfm_churn_case_study_transactions.csv** (transaction-level table, ~26,793 rows)

Important note about data quality

This dataset is intentionally not clean. It contains common real-world issues that you are expected to detect and fix. Do not ignore data quality. Explain what you fixed and why.

- Null values in multiple fields (for example: merchant_category, merchant_name, city, txn_datetime, currency, and some customer profile fields).
- Mixed date-time formats in txn_datetime (for example: ISO format, DD/MM/YYYY, and ISO with +0500). Some values include extra whitespace.
- amount is not consistently numeric (some values include commas or leading/trailing spaces). A small fraction of rows contains negative amounts.
- Inconsistent casing in txn_type and currency (for example: DEBIT vs debit, PKR vs pkr).
- Duplicate transactions exist (duplicated rows and repeated txn_id values).
- Category noise exists (for example: 'Grocerries' as a misspelling of 'Groceries').
- Some city values are abbreviated codes (for example: KHI, LHE, ISB).

Tasks

1. **A. Data cleaning and preparation**
   1) Load the CSV and perform basic checks (shape, duplicates, missing values). 2) Clean txn_datetime into a proper datetime type. 3) Convert amount into a numeric field and ensure the sign logic is correct using txn_type. 4) Standardize categorical fields (txn_type, currency, merchant_category). 5) Decide how to handle duplicates and justify your choice.
2. **B. PFM analytics**
   Create a PFM summary at the customer level. At minimum include: 1) total inflow vs outflow, 2) net cashflow, 3) spend by category, 4) top merchants, 5) recurring payments detection (for example: rent, utilities, subscriptions). Present at least three insights that would be valuable inside a banking app.
3. **C. Feature engineering for churn**
   Aggregate transaction history into customer-level features (examples: average debit amount, variability of spending, share of spending on essentials, number of bill payments, frequency of cash withdrawals, decline in inflows). Avoid target leakage and explain your feature choices.
4. **D. Churn prediction model**
   Build at least one baseline classification model to predict churn_next_30d. Split data into train and test sets appropriately. Show a confusion matrix and explain precision, recall, and F1 (or any other metric you use). Recommend an operating threshold and justify it in business terms (for example: cost of contacting customers vs missing churners).
5. **E. Presentation**
   Create a short presentation (6 to 8 slides) summarizing your work. Include: problem statement, data quality fixes, key PFM insights, feature engineering, model results (confusion matrix), and recommendations.

Deliverables

- 1) Jupyter Notebook (.ipynb) with runnable code and explanations (markdown).
- 2) A presentation file (PPTX or PDF).
- 3) Optional: a short README (half page) with how to run the notebook.

Technical requirements

- Use Python. Recommended libraries: pandas, numpy, matplotlib, scikit-learn.
- Use simple, interpretable models (examples: logistic regression, decision tree, random forest, gradient boosting).
- Do not use deep learning.
- Your work must be reproducible (fixed random seed where relevant).
- Clearly state assumptions and decisions (especially for data cleaning and threshold selection).

Data dictionary

The provided CSV is a transaction-level table. Customer-level fields repeat across a customer's transactions.

Field	Type	Description	Example	Known issues / notes
`customer_id`	int	Unique customer identifier	10045	Some customer-level fields may be null for a small set of customers.
`account_id`	string	Customer account identifier	AC100451234	One account per customer in this dataset.
`txn_id`	string	Transaction identifier	T2025120110045001234	Duplicates may exist (duplicated rows and repeated txn_id).
`txn_datetime`	string	Transaction date-time (mixed formats)	2025-12-12 02:11:50	Mixed formats, some whitespace, small fraction of nulls.
`txn_type`	string	Transaction direction	DEBIT	Casing inconsistency (DEBIT vs debit).
`amount`	string/float	Transaction amount (absolute value in most rows)	12,685.08	Not consistently numeric; commas/spaces; small fraction negative.
`currency`	string	Transaction currency	PKR	Mostly PKR; some pkr; rare USD; small fraction of nulls.
`channel`	string	Payment channel	BillPay	Values include MobileApp, POS, ATM, IBFT, BillPay.
`merchant_name`	string	Merchant name if available	Imtiaz	Some nulls; some extra whitespace.
`merchant_category`	string	Merchant category	Groceries	Nulls; misspelling 'Grocerries'.
`counterparty_bank`	string	Bank name for transfer transactions	HBL	Usually filled for TransferIn/TransferOut only.
`narration`	string	Short description / memo	Utility bill payment - PTCL	Free-text; can be used for additional cleaning if desired.
`city`	string	Customer city	Karachi	Some nulls; some abbreviated codes (KHI, LHE, ISB).
`segment`	string	Customer segment	Retail	Retail, Affluent, Student.
`customer_age`	float	Customer age in years	32	Some nulls.
`onboarding_days_ago`	int	Days since onboarding	504	Integer.

Field	Type	Description	Example	Known issues / notes
`app_logins_30d`	float	App logins in last 30 days	23	Some nulls.
`support_tickets_90d`	int	Support tickets in last 90 days	1	Integer.
`churn_next_30d`	int	Target label: 1 churned within next 30 days, else 0	0	Customer-level label repeated for all transactions.

Evaluation rubric (what we look for)

- **Data handling:** correct parsing, thoughtful cleaning choices, clear justification.
- **PFM reasoning:** meaningful customer insights, sensible aggregation, recurring detection approach.
- **Modeling:** correct train/test split, appropriate metrics, confusion matrix interpretation, threshold reasoning.
- **Communication:** clean notebook structure and a presentation that a non-technical stakeholder can follow.
- **Practicality:** solutions that could realistically be implemented in a bank.