Gulf Bank Datathon 2025 Problems

Data Science Track

1. ATM Cash Demand Forecasting

1.1 Overview and Problem Definition

Gulf Bank operates a large and growing network of ATMs across Kuwait, located in branches, malls, and off-site areas. Managing these machines efficiently is critical:

- Too little cash causes customer frustration and potential revenue loss.
- Too much cash leads to unnecessary costs and idle liquidity.

Your task is to develop a data-driven forecasting model that predicts daily cash withdrawals for each ATM.
The forecasts will enable the bank to plan cash replenishment schedules and enhance its liquidity management.

The dataset simulates real-world ATM behavior, complete with natural challenges such as missing days, delayed transaction reporting, and changing customer patterns.
You are expected to build robust and explainable models that work under these realistic conditions.

Each team must forecast:

- Withdrawal Amount (KWD)
- Withdrawal Count

The goal is to strike a balance between predictive accuracy, generalization, and interpretability.

1.2 Challenge Structure

The challenge consists of three main tasks and one optional extension:

1️⃣ Baseline Forecasting (Statistical Models)

2️⃣ Feature Engineering & Machine Learning Models

3️⃣ Explainability & Business Insights

⭐ Optional Extension: Scenario Visualization

All submissions will be automatically run and scored by the AI judging system. The AI judge will execute train.py and predict.py to generate and score results, and must be able to do so.

Human experts will review submissions in cases of rule violations, errors, or technical issues.

1.A Sub-Task 1 - Baseline Forecasting

Objective

Develop simple, interpretable forecasting models that establish a performance baseline for each ATM.

What You Should Do

- Use atm_transactions_train.csv to forecast the next 14 days in atm_transactions_test.csv.
- Predict both withdrawal amount and withdrawal count.
- Build at least three baseline models from the list below:
    - Naive (last-day value)
    - Moving average (7, 14, or 28 days)
    - Exponential smoothing
    - ARIMA / SARIMA models

Judge Evaluation

The judge will verify schema and code correctness, compute RMSE, and assign scores for accuracy and completeness.

What to Submit

- predictions.csv - forecasts for all (dt, atm_id) pairs in the test set.
    - Columns: dt, atm_id, predicted_withdrawn_kwd, predicted_withdraw_count
    - Must match test data 1:1 (no missing or extra rows).
- train.py and predict.py - runnable scripts to train your model and make predictions.
- requirements.txt or environment.yml - to recreate your environment.
- README.md - short text explaining your baseline models, validation steps, and parameters used (≤1 page).

Evaluation (30 pts)

CRITERION	POINTS	DESCRIPTION
BASELINE IMPLEMENTATION AND METRIC SCORE	20	Three or more valid baseline models and the quality of your RMSE metric
CODE CLARITY, QUALITY, AND COMPLETENESS	10	Code readability (including proper documentation, proper layout, programming best practices, etc.)

1.B Sub-Task 2 - Feature Engineering & Model Enhancement

Objective

Design advanced models that combine statistical and machine learning approaches, incorporating richer contextual features.

What You Should Do

Use the following datasets to improve predictive performance:

- calendar.csv - weekends, public holidays, Ramadan, and salary cycles

- atm_metadata.csv - region, location type, installation and decommission dates
- cash_replenishment.csv - cash-out events and replenishment timing
- Transaction data - lagged, aggregated, or cleaned features

Train your models on the training data and generate predictions for the test period.

The datasets are intentionally realistic, containing noise, shifts, and minor inconsistencies between sources.

Successful teams will treat these imperfections as opportunities to engineer features that connect patterns across files, rather than trying to clean them away entirely.

Account for real-world data imperfections such as:

- Missing or duplicated records
- Reporting delays
- Seasonal or regional shifts in behavior
- Newly installed ATMs with limited history

You may use regression, time-series, or hybrid methods. Any form of machine learning model that predicts the withdrawal amount and withdrawal count is acceptable.

Clearly document how you validate and select your models and feature engineering steps.

What to Submit

- Updated predictions.csv - forecasts for all (dt, atm_id) pairs in the test set.
    - Columns: dt, atm_id, predicted_withdrawn_kwd, predicted_withdraw_count
    - Must match test data 1:1 (no missing or extra rows).
- Updated train.py and predict.py - runnable scripts to train your model and make predictions.
- Updated requirements.txt or environment.yml - to recreate your environment.
- README.md - short text explaining your baseline models, validation steps, and parameters used (≤1 page).
- A short report (1–2 pages) describing your features, model types, and validation approach

Evaluation (30 pts)

CRITERION	POINTS	DESCRIPTION
MODEL IMPLEMENTATION AND METRIC SCORE	15	Three or more valid models and the quality of your RMSE metric
CODE CLARITY, QUALITY, AND COMPLETENESS	5	Code readability (including proper documentation, proper layout, programming best practices, etc.)

ENGINEERED FEATURES	10	Feature quality, testing of all potentially valuable features, and meaningful selection of the best features

1.C Sub-Task 3 - Explainability & Business Insight

Objective

Explain your model's decisions and highlight the operational insights it provides.

What You Should Do

- Use explainability tools (e.g., SHAP, permutation importance) to identify main drivers of withdrawals.
- Compare global factors (e.g., holidays, salary periods) with local ones (specific regions or ATMs).
- Translate findings into actionable insights for cash planning and replenishment decisions.

Communicate your conclusions in a clear, visual format that a business audience could understand.

What to Submit

- analysis.py - runnable script to produce explainable results.
- requirements.txt or environment.yml - to recreate your environment.
- README.md - short text explaining your baseline models, validation steps, and parameters used (≤1 page).
- A report with visualizations (≤3 pages) describing feature importance and insights

Evaluation (40 pts)

CRITERION	POINTS	DESCRIPTION
INSIGHT QUALITY	15	Identifies strong, data-backed drivers
CLARITY AND ACCURACY	10	Interprets patterns and relationships correctly
BUSINESS VALUE	10	Connects findings to operational improvements
CODE CLARITY, QUALITY, AND COMPLETENESS	5	Code readability (including proper documentation, proper layout, programming best practices, etc.)

1.D Optional - Scenario Visualization (Bonus +10 pts)

Objective

Build a simple dashboard or visualization to explore "what-if" scenarios:

- How would changes in salary dates affect cash demand?
- What happens to withdrawals during holidays or Ramadan?

- How do new ATMs affect regional demand?

Use any visualization tool or platform.

Focus on clarity and storytelling rather than technical complexity.

What to Submit

- dashboard/ folder containing:
    - Code for your visualization (e.g., Dash, Streamlit, Power BI export, or Jupyter notebook).
    - One static preview image (screenshot.png).
- summary.md - a 1-page explanation of what scenarios your visualization explores (e.g., salary-day shifts, regional changes).

Evaluation (10 bonus pts)

CRITERION	POINTS	DESCRIPTION
CLARITY	4	Easy to interpret and visually clear
CREATIVITY	3	Engaging or interactive approach
RELEVANCE AND VALUE	3	Direct link to business decision-making and the potential business value

1.3 Dataset Description

You will receive five CSV files covering the period from January 2020 to November 2025.

FILE	DESCRIPTION
ATM_METADATA.CSV	ATM details - region, location type, installation, and decommission dates
ATM_REGION_LOOKUP.CSV	A simplified mapping of each atm_id to its assigned region. This file can be used for quick joins or regional aggregation, but it may not always align perfectly with the metadata. A few ATMs may be missing or displaying inconsistent data due to relocations or changes in behavior, which is a natural occurrence.
CALENDAR.CSV	Daily calendar with weekends, holidays, Ramadan, and salary cycle indicators
ATM_TRANSACTIONS_TRAIN.CSV	Historical withdrawals and deposits for model training
ATM_TRANSACTIONS_TEST.CSV	Same structure, used for prediction and scoring
CASH_REPLENISHMENT.CSV	Replenishment operations, ending balances, and cash-out flags

The data captures a realistic operating environment with:

- Missing days and maintenance outages
- Delayed transaction reporting

- Occasional duplicates and anomalies
- Behavioral changes across time and regions
- Newly installed ATMs with short histories

Use metadata to determine each ATM's active period.

Note: Each dataset reflects a different aspect of ATM operations - customer behavior (transactions), timing and seasonality (calendar), physical and regional context (metadata), and operational activity (replenishment).

Strong solutions will combine information from all files to uncover richer patterns and build more accurate forecasts.

1.3.1 Data Quality and Common Issues

The dataset is designed to reflect the challenges of real banking data.

While it is clean enough to use directly, you'll notice several realistic imperfections that require thoughtful handling.

Identifying and addressing these issues is an essential part of the challenge.

Potential data issues and how to handle them:

ISSUE TYPE	DESCRIPTION	SUGGESTED APPROACH
MISSING DAYS	Some ATMs may have days with no recorded transactions or replenishment events due to outages or reporting gaps.	Use interpolation, forward-filling, or model features that can handle gaps (e.g., lag features with null awareness).
REPORTING DELAYS	A few transactions are recorded late or with mismatched report dates.	Compare dt and reported_dt to detect lags; aggregate or adjust accordingly.
DUPLICATE RECORDS	Certain ATM-day entries appear more than once (e.g., repeated uploads).	Remove exact duplicates or consolidate by summing daily totals.
REGION INCONSISTENCIES	Some ATMs exhibit behavioral changes inconsistent with their listed region, or their region may vary across different files.	Treat the region as a helpful hint, not an absolute truth. Detect and manage regional drift by comparing ATM-level vs. region-level behavior.
COLD-START ATMS	New ATMs appear midway in the data with limited history.	Use region- or type-level averages to estimate initial patterns.
DECOMMISSIONED ATMS	A few ATMs stop appearing after their decommission date.	Exclude them after deactivation or fill with zeros depending on the model design.
DATA SHIFTS / SEASONAL DRIFT	Customer behavior changes over time (e.g., during Ramadan or due to macroeconomic changes).	Include time-based or seasonality features and retrain models over sliding windows.

UNBALANCED ACTIVITY	Urban ATMs handle far higher volumes than rural ones.	Consider scaling, stratified validation, or volume-based weighting when training.

💡 Tip: These imperfections are intentional; they simulate the reality of ATM operations.

Effective solutions will detect, clean, and model around these issues rather than assuming perfect data.

1.4 Scoring and Leaderboard

Your predictions will be evaluated using a weighted score based on:

RMSE with a 2x penalty applied to your errors for under-predicting the amount.

The Cash-out Penalty adds a higher cost to underestimating demand on days when ATMs run out of cash.

Scores are normalized by ATM activity level and averaged across all ATMs.

A live leaderboard will display team rankings and performance metrics.

1.5 Gen AI Usage Policy

Participants are encouraged to utilize generative AI tools to aid in problem-solving. However, all solutions submitted are the sole responsibility of the participant, including the accuracy, validity, and originality of the final outputs.

1.6 Summary

This challenge is designed to reflect real-world data science work in the banking industry.

You will need to handle imperfect data, model dynamic patterns, and clearly explain your results.

Success will depend not only on predictive accuracy but also on robustness, interpretability, and a clear understanding of the business.

Good luck, and may your ATMs always stay stocked with cash! 💸