Case Study 2 - Applied AI - Group 9

- Himanish Tripathi, Hasan Sufi, Ho Ngoc An

1. Executive Summary

The Task: Predict daily transaction totals for Q1 2021 (Jan 1 – Mar 31) using 7 years (2014-2020) of historical data.

Process Overview:

- Analyzed over 67,000 individual transactions from 2014-2020.
- Built a baseline Random Forest model.
- Enhanced the model with explicit calendar features and proper train/validation split.
- Validated performance on 2020 data before forecasting 2021.

Key Insight: Weekends consistently show **zero transactions** across all 7 years. Capturing this weekly pattern is the most critical factor for model success.

Assumption: Observed patterns (weekly cycle, monthly variation) will continue into 2021.

2. Data Analysis: What We Found

(Connecting to Module 3: Supervised Forecasting - Understanding your time series)

2.1 Raw Data Overview

- **Total Transactions:** 67,732 individual transactions.
- **Date Range:** January 2014 → December 2020 (7 years).
- **Transaction Types:** Cash Parcels, Bill Payments, Transfers.
- **Data Structure:** Each row is a single transaction, not a daily total.

Our initial step was to aggregate these into ~2,500 days of daily net cash flow.

2.2 Observed Patterns

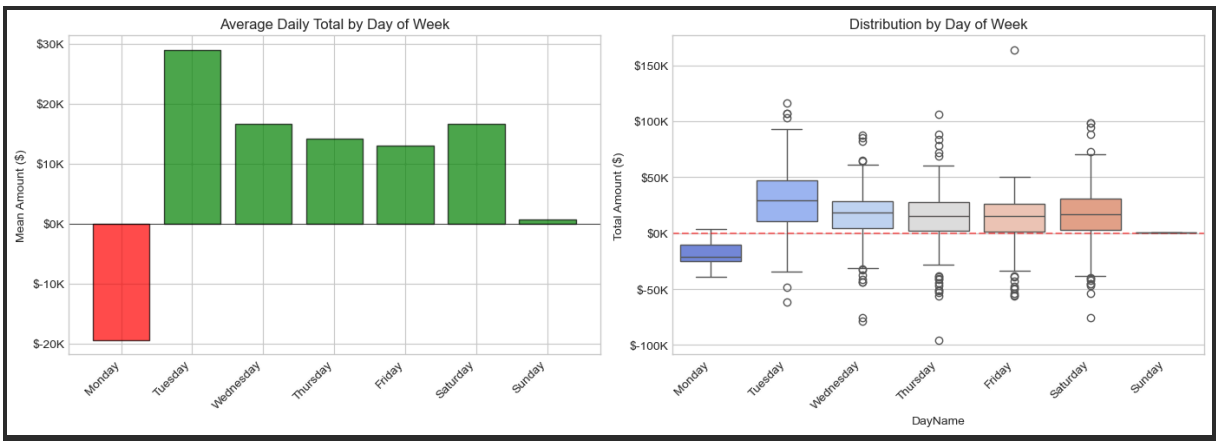
After plotting the data and calculating statistics, we found three main patterns:

Pattern 1: Weekends Are Dead

Day	Average Transaction Total
Monday	~$15,000
Tuesday	~$18,000

Wednesday	~$16,000
Thursday	~$14,000
Friday	~$12,000
Saturday	**$0**
Sunday	**$0**

This isn't missing data, it's real. The business doesn't process transactions on weekends. This pattern is 100% consistent across all 7 years.

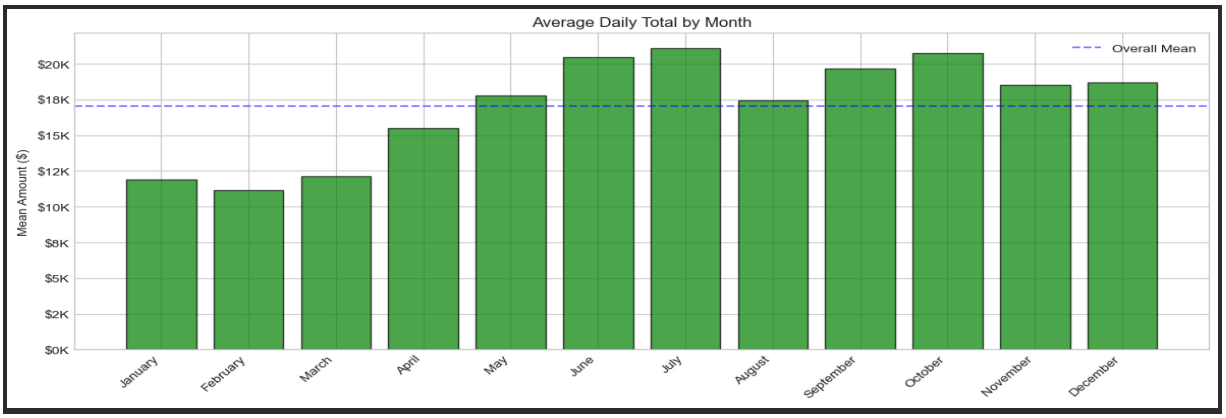


Why this matters: If our model predicts $10,000 on a Saturday, we're automatically way off. The weekend pattern is the #1 thing to get right.

Pattern 2: Moderate Monthly Seasonality

Some months are busier than others:

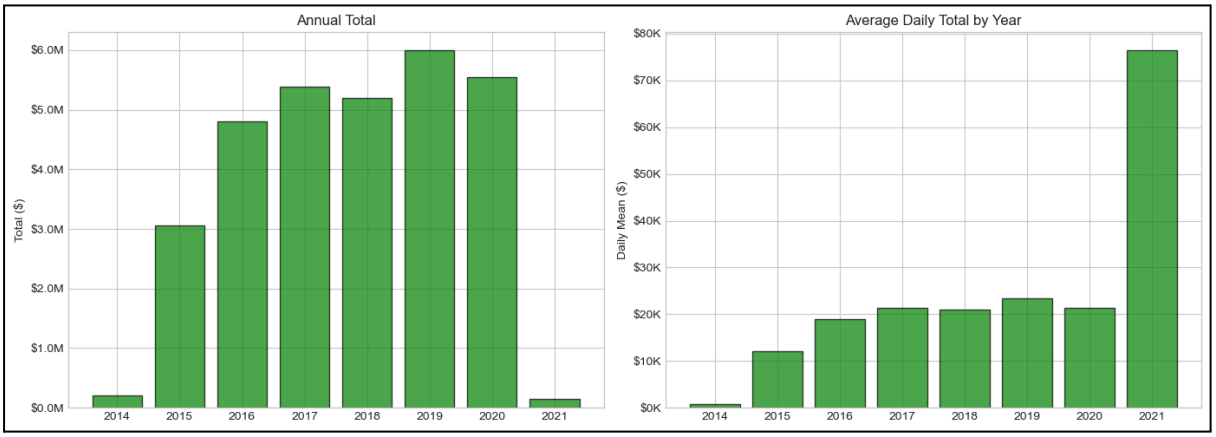
- **December** is elevated (year-end closings)
- **April** tends to be slower
- **Month-end days** (28th-31st) show higher activity (bills coming due)



Connection to lecture: This is the "seasonality" component Jeff talked about in the forecasting module. We can capture it with a `month` feature.

Pattern 3: No Strong Long-Term Trend

When we looked at yearly totals, we didn't see big growth or decline:The business is stable, no big trend to model. This actually simplifies things.



3. Himanish's First Model: The Starting Point

This section walks through our team's initial working model, built by Himanish.

3.1 Code Breakdown (5 Main Steps)

1. **Load the Data:** Loads the 67K transaction history and the Q1 2021 output template.
2. **Fix the Dates:** Converts the Excel serial date format (e.g., 41638) in `Report_TransactionEffectiveDate` into standard date objects.
3. **Aggregate to Daily Totals (Smart Step):**
   - Groups and sums transactions by date (net flow).
   - Uses `.resample('D').sum().fillna(0)` to explicitly ensure weekends and days with no data are marked as **0**, not missing.
4. **Create Calendar Features:** Extracts the following features from the date index:
   - `dayofweek` (0=Mon, 6=Sun)
   - `month` (1-12)
   - `dayofmonth` (1-31)
   - `year` (2014-2020)
5. **Train Random Forest:**
   - Model: `RandomForestRegressor(n_estimators=100)`.
   - Choice Rationale: Random Forest handles non-linear patterns (like the weekday/weekend jump) and categorical-like features well, and provides feature importance.

3.2 Initial Model Assessment

Decision	Why It Was Right
`.fillna(0)`	Correctly represents the real-world weekend zero-transaction pattern.
4 Calendar Features	Captures the main weekly and monthly seasonality identified in analysis.
Random Forest	Suitable algorithm for the identified non-linear, categorical-like patterns.

Issue	Opportunity for Improvement
No validation set	Prevents measuring performance before final submission.
No explicit flags	Requires the model to implicitly learn that `dayofweek` 5 & 6 = weekend.

4. Enhanced Model: Targeted Improvements

We made targeted improvements to Himanish's solid foundation to increase robustness and measure performance.

4.1 Proper Train/Validation Split (Module 4: Model Validation)

Before (Himanish)	After (Enhanced)	Why It Matters
Train on 2014-2020.	Train on **2014-2019**, Test on **2020**.	Allows us to **measure** model quality (RMSE) on recent, unseen data (2020 is a "practice run" for 2021).

4.2 Added Explicit Features

- **Explicit Weekend Flag:** `df['IsWeekend'] = (df.index.dayofweek >= 5).astype(int)`
    - *Why it helps:* Makes the critical zero-transaction pattern explicit, allowing the Random Forest to split on it directly.

- **Month-End Flag:** `df['IsMonthEnd'] = df.index.is_month_end.astype(int)`
    - *Why it helps:* Directly captures the observed higher activity near the end of the month.

4.3 Final Feature Set Comparison

Feature	Himanish's Model	Enhanced (Final)
`dayofweek`	✅	✅
`month`	✅	✅
`dayofmonth`	✅	✅
`year`	✅	✅
`IsWeekend`	❌	✅ **(New)**
`IsMonthEnd`	❌	✅ **(New)**

4.4 Why Lag Features Were Rejected

We considered features like Lag_1 (yesterday's value) and Lag_7 (same day last week).

- **Decision:** Rejected.
- **Reasoning (Connection to Lecture: Iterative Forecasting):** Forecasting 90 days out requires using our *own predictions* as inputs for the next day, leading to compounding errors. The simpler calendar features already capture the dominant patterns without this complexity and risk.

5. Results & Model Comparison

5.1 Validation Results (Test Set: 2020)

The Enhanced Random Forest was compared against simple benchmarks to prove its value.

Model	RMSE	Notes
Mean Baseline	$23,432	Ignores all weekly and monthly patterns.
Random Forest (Enhanced)	$20,160	Lowest error.

Improvement: The Enhanced Random Forest model beats the simple Mean Baseline by **14.0%**, confirming its predictive power.

5.2 Feature Importance

The Random Forest model ranks the importance of features for predicting the transaction totals.

Rank	Feature	Importance	What This Means
1	`DayOfMonth`	0.197	Position within the month is the strongest predictor (likely capturing mid-month/month-end payment cycles).
2	`Year`	0.183	Some variation across the 7 years matters.
3	`IsMonthEnd`	0.173	Explicitly confirms the month-end billing cycle effect.
4	`IsWeekend`	0.160	Confirms the critical weekend pattern is well-captured.
5	`Month`	0.157	Seasonal effects (e.g., December being busier) are real.
6	`DayOfWeek`	0.130	Specific weekday matters (e.g., Tuesday is the busiest).

5.3 Visual Validation (January 2020)

Comparison of predictions vs. actuals for January 2020 showed:

- Weekends are **correctly predicted at ~$0**.
- Weekday activity levels match reasonably well.
- Some day-to-day noise remains unpredicted

6. Limitations & Reflections

6.1 Gaps and Future Work

Gap	Why We Skipped	Impact on Current Model
Holiday Calendar	No integrated BC holiday data was available.	May mispredict transaction volume on statutory holidays (e.g., Family Day).
Hyperparameter Tuning	Time constraints.	Expected marginal improvement; not a priority for a robust baseline model.
ARIMA Comparison	Weekly seasonality (s=7) adds complexity.	Worth trying, but tree models are often well-suited for this type of data (as noted by Jeff).
COVID Analysis	2020 data is used for validation.	Adds a degree of uncertainty, as 2020 may be atypical.

6.2 What-If Scenarios

- **What if weekends aren't zero in 2021?** The model will under-predict any weekend activity, but 7 years of consistent data supports the zero-weekend assumption.
- **What if we used ARIMA?** ARIMA is better for pure trend/autocorrelation but less clean for incorporating strong categorical features like *day-of-week* or *is-weekend*.

6.3 Individual Reflections

- **Himanish:** *"Kept the code simple on purpose; the 4 calendar features capture the main patterns without risking overfitting."*
- **Ho Ngoc An:** *"Data analysis confirmed the weekend pattern immediately it's the most obvious factor to focus on."*
- **Hasan:** *"Adding validation and explicit flags (IsWeekend) gave us the ability to measure RMSE and gain confidence before submission."*

7. Conclusion

What We Built: A Random Forest regression model with 6 calendar-based features that forecasts daily transaction volumes for Q1 2021.

Why It Works:

1. **Data Understanding:** We prioritized the key pattern: weekends = zero.
2. **Solid Foundation:** We enhanced a working baseline model.

3. **Validation:** We tested on 2020 data to prove the model's performance (14.0% improvement over mean baseline).
4. **Interpretability:** Feature importance clearly shows the factors driving the predictions.