Sales Prediction and Visualization Using R Shiny on Bank Dataset

Md Emon Sharkar

Alumni, Daffodil International University

emon15-3141@diu.edu.bd | github.com/emonsharkar

Abstract

This project presents a comprehensive sales analytics dashboard developed using R Shiny to analyze and predict sales patterns in a large-scale transactional banking dataset. The tool includes insightful visualizations, a linear regression model for predictive sales analysis, and a forecasting module using ARIMA. With real-time data interactivity, the dashboard enables users to explore trends and derive actionable insights from transaction values across multiple cities and domains. The predictive system responds dynamically to both user input and randomly selected data entries, showcasing its practical flexibility in real-world scenarios.

1. Introduction

1.1 Aim and Objectives

The aim of this project is to develop an interactive and intelligent dashboard to derive insights from banking transaction data. The specific objectives are as follows:

- Cleaning and structuring transaction datasets for analysis.
- Building dynamic and interpretable visualizations.
- Applying regression-based models to predict future values.
- Forecasting upcoming trends in transaction values.
- Enabling interactivity for business simulation use cases.

1.2 Dataset Description

The dataset has been taken from Kaggle which consists of 1,004,480 rows and 5 columns. The owner of the dataset indicated few questions which were the brain-driver to work with this dataset and answer all those questions via visualization besides doing necessary sale prediction using R. The presented results in the paper are tested initially on a subset (365 rows for rapid testing) but the model and the dashboard are designed to scale seamlessly with the full dataset.

The dataset comprises a large set of daily banking transactions with the following fields:

- **Date:** The date on which the transaction occurred.
- **Domain:** The category of transaction *i.e.*, E-commerce, Banking, Travel, Education.
- **City:** The city where the transaction took place.
- **Transaction value:** The monetary value of the transaction.
- **Transaction Count:** Number of transactions recorded.

2. Methodology

2.1 Tools and Technologies Used

- **R:** Core analytical and visualization language.
- **Shiny:** For web-based interactive dashboard development.
- **ggplot2** and **plotly:** Used to build visualizations and interactive plots.
- **forecast:** Used for time series modeling and forecasting.
- **Janitor** and **dplyr:** For data cleaning and manipulation

2.2 Data Cleaning and Preprocessing

Data was cleaned using the **janitor** package, converting field names to **snake_case**, and ensuring all categorical fields like city and domain were transformed into factors for modeling purposes**. Random sampling** was used to simulate a full-scale dataset.

2.3 Dashboard Design

The UI comprises three main sections:

- Dashboard: Includes faceted plots, average values, domain priority ranking, and city-wise summaries. The whole modular UI was designed using *shinydashboard*.

- Sales Prediction: Shows ARIMA-based 30-day forecast.

- Interactive Section: Includes real-time random predictions and user-input based forecasting.

2.4 Predictive Modeling

A multiple linear regression model was trained using transaction value as the dependent variable, and transaction count, domain, and city as independent predictors. This model is reused consistently across the app to ensure accurate predictions for both automated and user-driven inputs.

2.5 Forecasting

Time-series forecasting was implemented using the **Auto ARIMA** model from the ‘forecast’ package. A 30-day horizon was chosen to show near-future sales estimates with confidence intervals.

3. Results & Features

The final dashboard offers **dynamic charts** that includes average transaction values over time, domain, and city-wise breakdown, and top-performing cities.

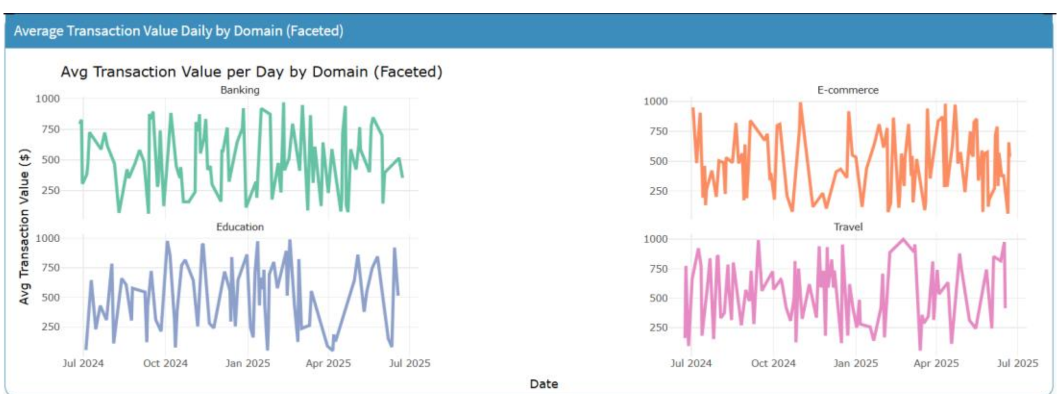


Figure: Average Transaction Value per Day by Domain

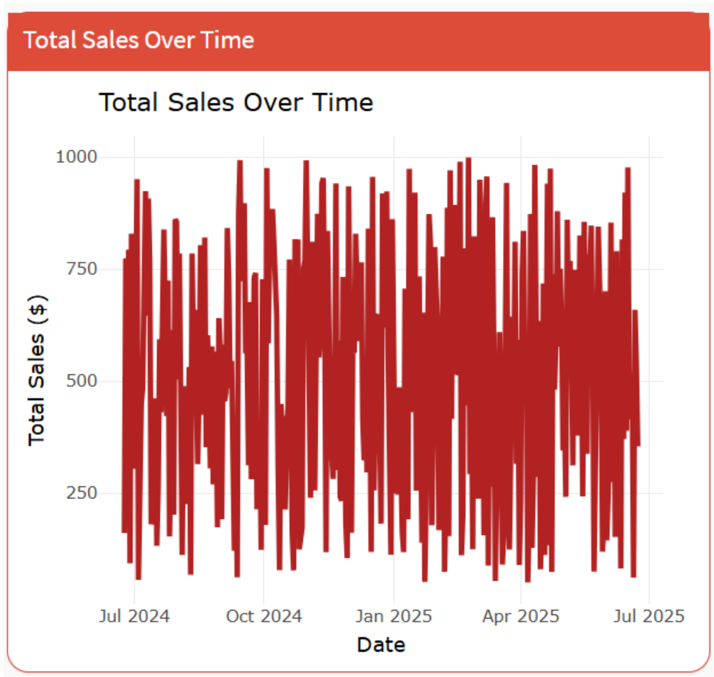


Figure: Total Sales over Time

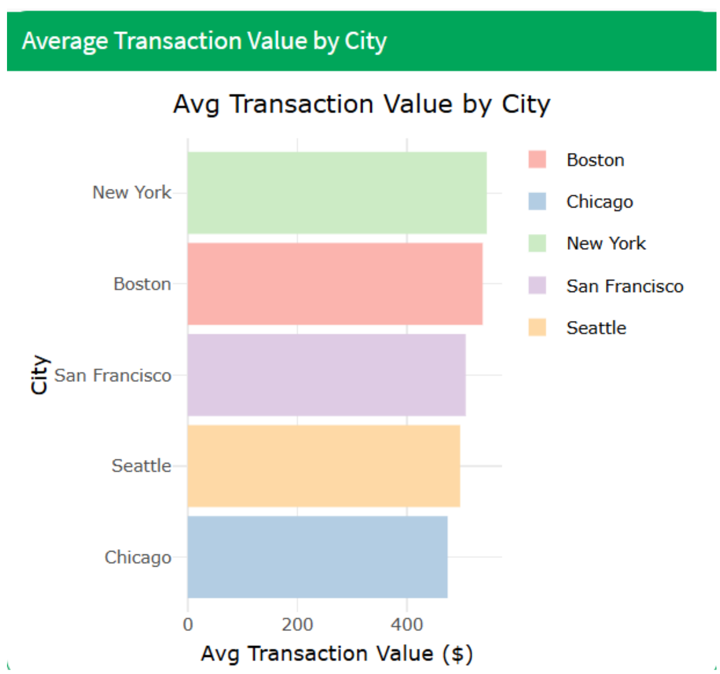


Figure: Average Transaction Value by City

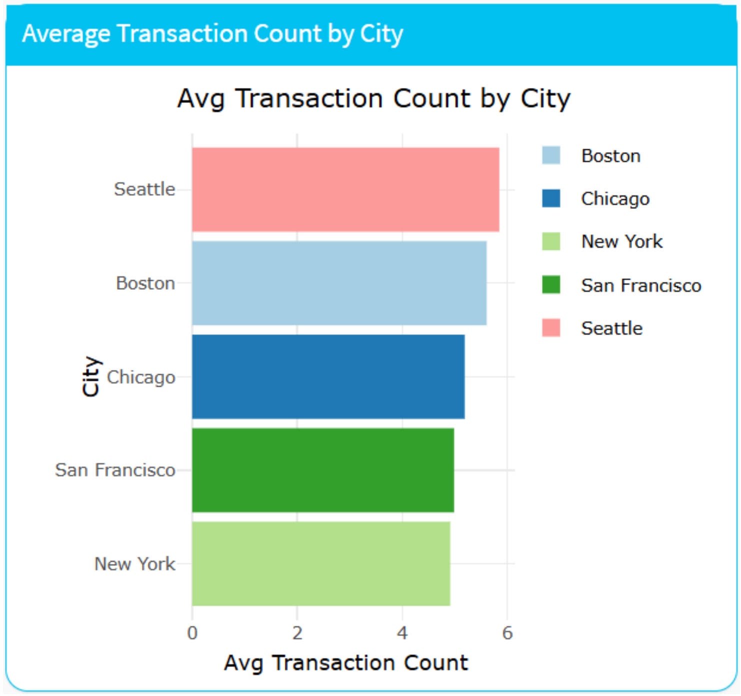


Figure: Average Transaction Count by City

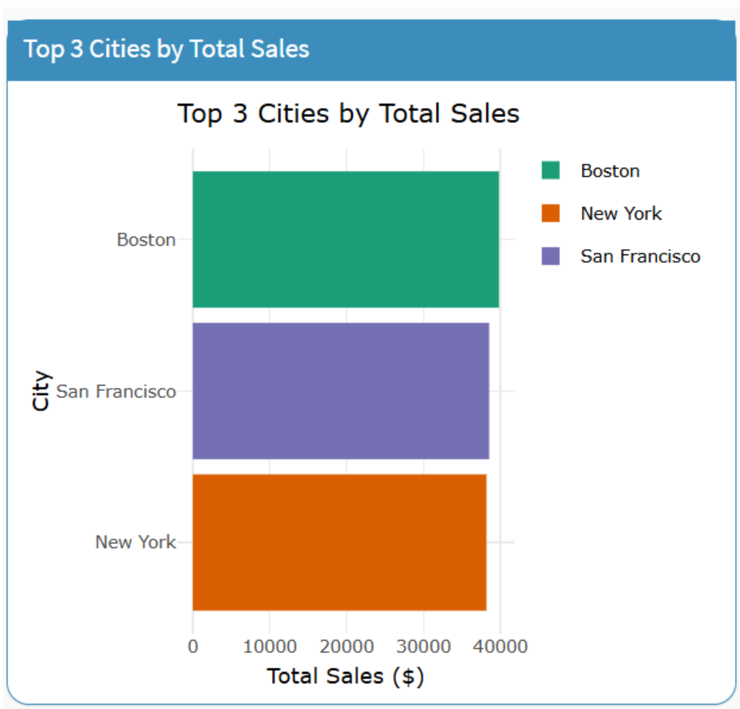


Figure: Top 3 Cities by Total Sales

User prediction panel has been introduced to enter transaction count, city, and domain to receive predicted transaction value.

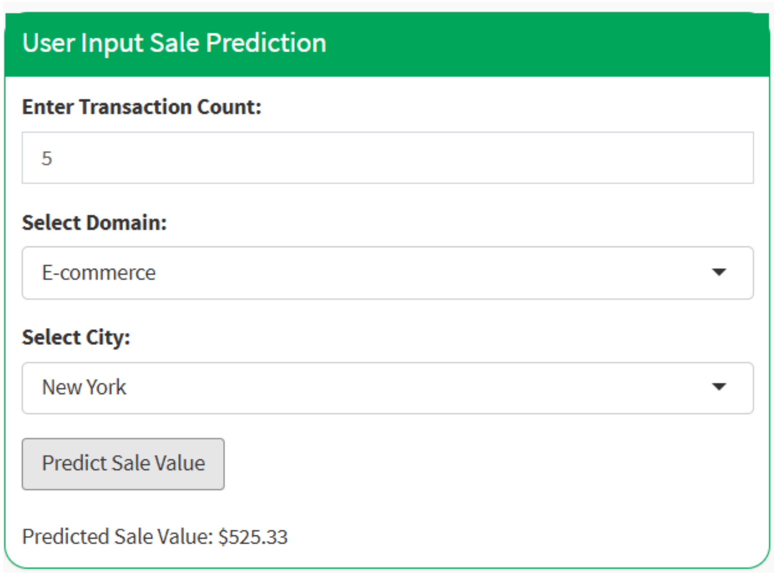


Figure: User Input Sale Prediction

A randomly selected data point updates every 5 seconds to simulate real-time prediction marked to be **Randomized Live Prediction.**

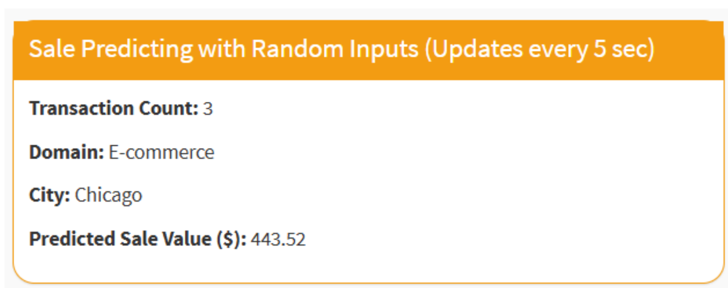
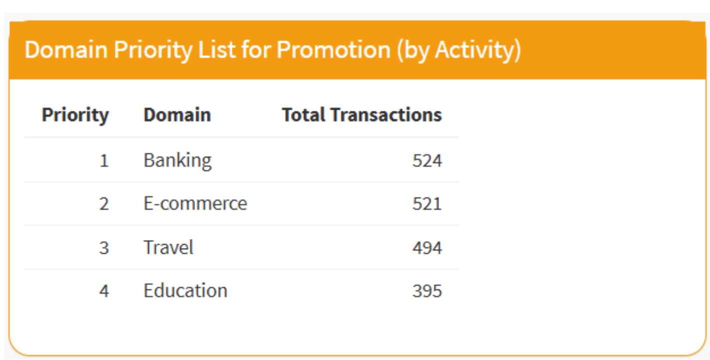


Figure: Sale Predicting with Random Inputs

Domain Priority Table ranks domain by transaction activity for business promotional targeting.



Domain Priority List for Promotion (by Activity)

Priority	Domain	Total Transactions
1	Banking	524
2	E-commerce	521
3	Travel	494
4	Education	395

Figure: Domain Priority List for Promotion

A 30-day prediction of total sales value is generated and plotted interactively as **Sales Forecasting**.

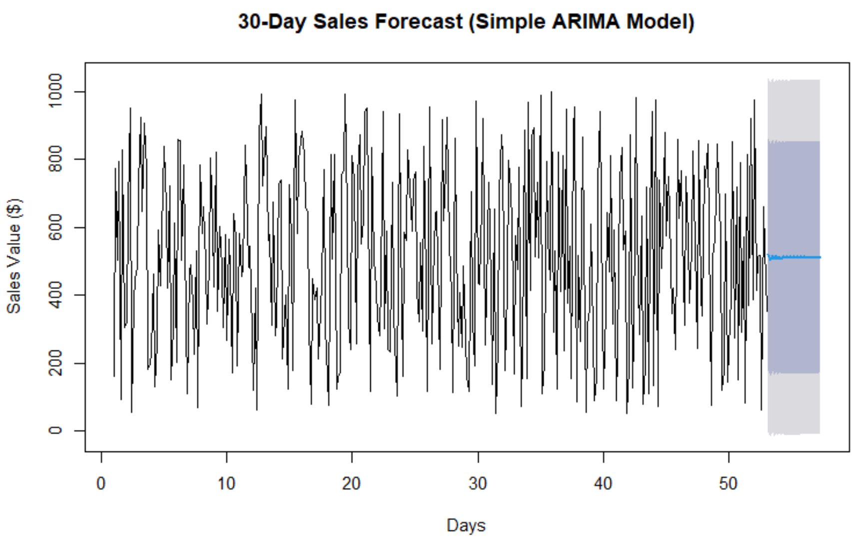


Figure: Sales Value Time Series & 30-Day Forecast

Showcasing Dataset on Live is another feature where the data from the dataset are shown selected random row but surely from the dataset.

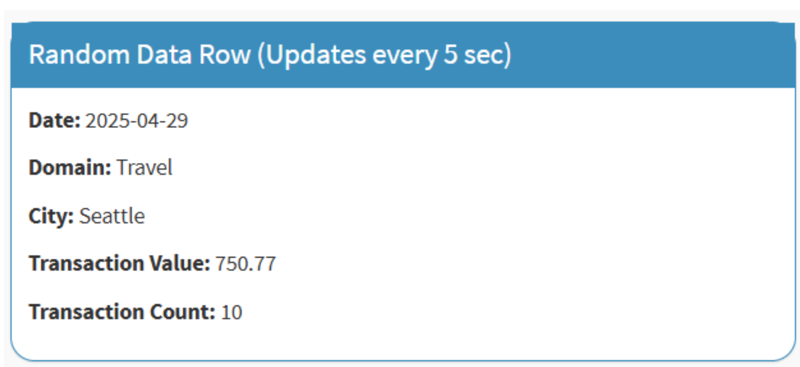


Figure: Random Data Row

Clean layout with faceted views, legends, and toggles, optimized for usability and comprehension. We focused on **Responsive Design** to make it usable on various pixels' devices.

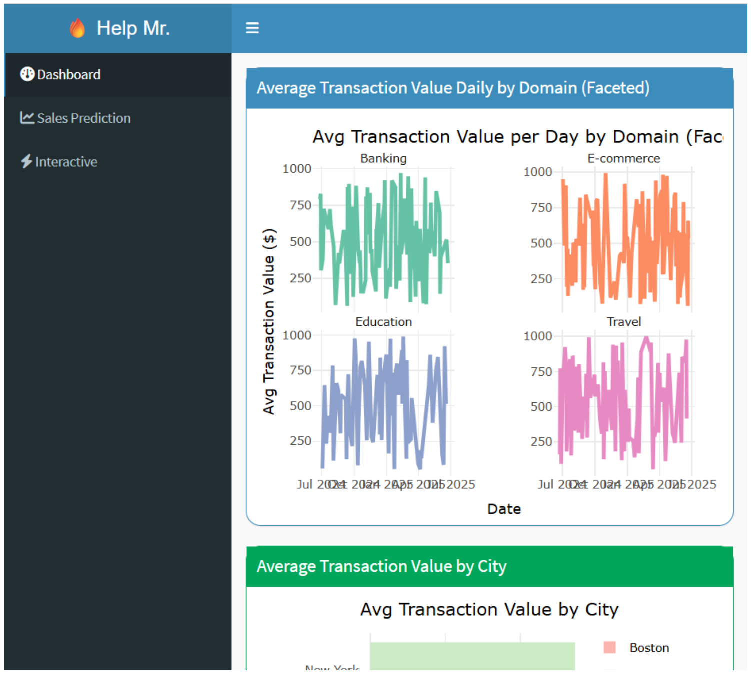


Figure: View on an odd pixel size

The results were validated using subsets of the large datasets and are scalable to the full dataset without performance degradation.

4. How to Run the Project

Software Requirements:

- **R** (version 4.1 or higher recommended)
- **RStudio IDE** (Latest stable release)

Required R Packages:

install.packages(c("shiny", "shinydashboard", "ggplot2", "dplyr", "janitor", "plotly", "forecast"))

Execution Steps:

1. Clone or download the project folder from GitHub.
2. Open the main R script in RStudio.
3. Click **Run App** (top-right corner in RStudio IDE) or use the console: *shiny:: runApp()*
4. Interact with the tabs and explore visualizations and predictions.

5. Discussion

5.1 Limitations

- The applied linear model assumes a linear relationship, which may not hold in all real-world cases.
- Categorical encoding does not reflect deeper semantic relationships.
- Forecasting may not account for unexpected economic changes or anomalies.

5.2 Expected Future Work

- Integrate more advanced models such as Random Forest, Gradient Boosting, and others.
- Deploy application on public cloud or Shinyapps.io.
- Add anomaly detection, alert triggers, and user login system.
- Expand dataset to real transaction APIs or continuous data ingestion pipelines.

6. Conclusion

This R Shiny project demonstrates the power of combining statistical models, visualization, and user interactivity in banking and financial analytics. The combination of statistical modeling, interactive visuals, and real-time prediction provides a robust foundation for data-driven strategies in finance and e-commerce. The work is expandable for industry-scale applications and serves as a solid framework for further academic research or product development.

7. References

1. Hyndman, R.J., & Athanasopoulos, G. (2018). Forecasting: principles and practice.
2. Sharkar, M. E., Hosen, M. J., Abdullah, M., Islam, S., Rana, S., & Sultana, N. (2024, June). Sentiment Analysis of Israel-Palestine Conflict Comments Using Sentiment Intensity Analyzer and TextBlob. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.
3. Sharkar, M. E. (2025). Unveiling Emotional Landscape: A Sentiment Expedition into Instagram Play Store and App Store Reviews using TextBlob. أطراس ,1(6), 260-274.
4. Wickham, H., et al. (2019). dplyr: A Grammar of Data Manipulation.
5. Chang, W. (2021). shiny: Web Application Framework for R.
6. K S ABISHEK. (2023). Massive Bank dataset (1 Million+ rows) [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/5038425