Title:

Enhancing Customer Lifetime Value Predictions with Advanced Machine Learning and Deep Learning Models

Full Name: Shashwat Dhayade

Report Submission Date: 11/01/2024

1. Introduction

Over the last two weeks, I focused on refining the dataset, creating foundational models for Customer Lifetime Value (CLV) prediction, and conducting extensive feature engineering. This report outlines the approach taken for data cleaning, model building based on the Recency-Frequency-Monetary (RFM) strategy, and experiments with different feature selections to address skewed data distributions and model accuracy challenges. Additionally, it highlights the challenges encountered with negative or low R-squared values, which informed me of the next steps in my modeling process.

---

2. Summary of Work Done

My first priority was completing data cleaning to understand the impact of stock codes like C2, Next Day Carriage, POSTAGE, DOTCOM POSTAGE, and duplicate entries on model predictions. Initially, I tested the dataset that included these stock codes and duplicates: it contained 11,665 duplicates, 80 entries for Next Day Carriage, 254 for C2, 16 for DOTCOM POSTAGE, and 1,983 for POSTAGE. After this exploratory analysis, I created a new column, *Revenue*, calculated as the product of Price and Quantity, to capture the revenue generated by each purchase. There were 808240 total purchases.

To implement the RFM strategy, I created a calculate_rfm_features function that extracted *Recency*, *Frequency*, and *Monetary Sum* values for each customer for the whole dataset of 2 years.

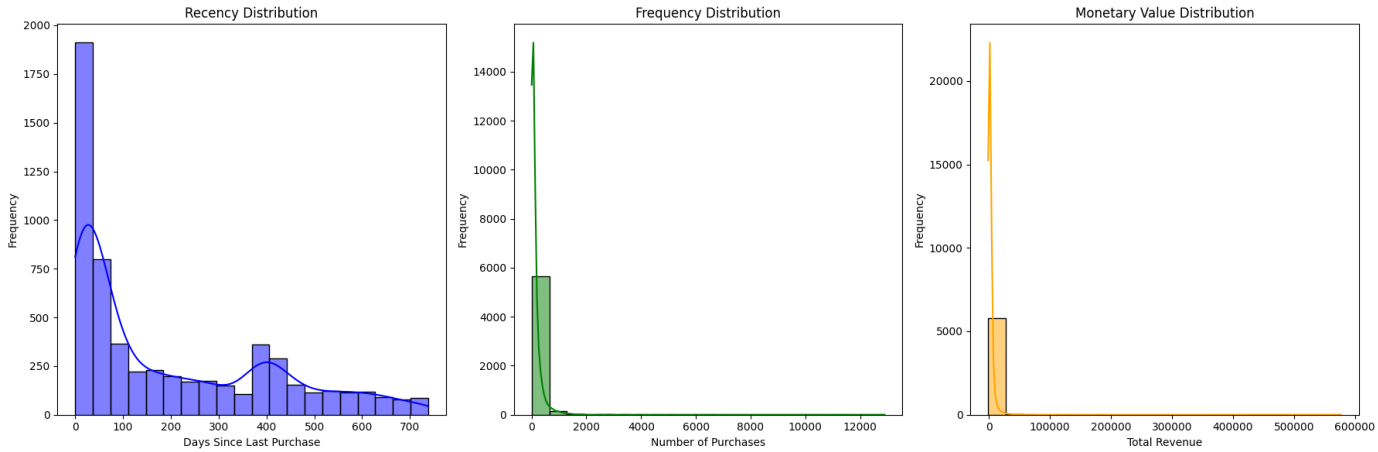

Figure 1: Distribution of Features

The features are extremely skewed. Using these metrics, I segmented the dataset to create training, validation, and test sets (70%, 20%, and 10%, respectively). The X_train was feature dataset of training data, y_train was the Monetary_Sum of validation data, X_test was feature dataset of validation data, and y_test was Monetary_Sum of test data with the intention of model being able to predict Monetary_Sum for the next time period and not the same.

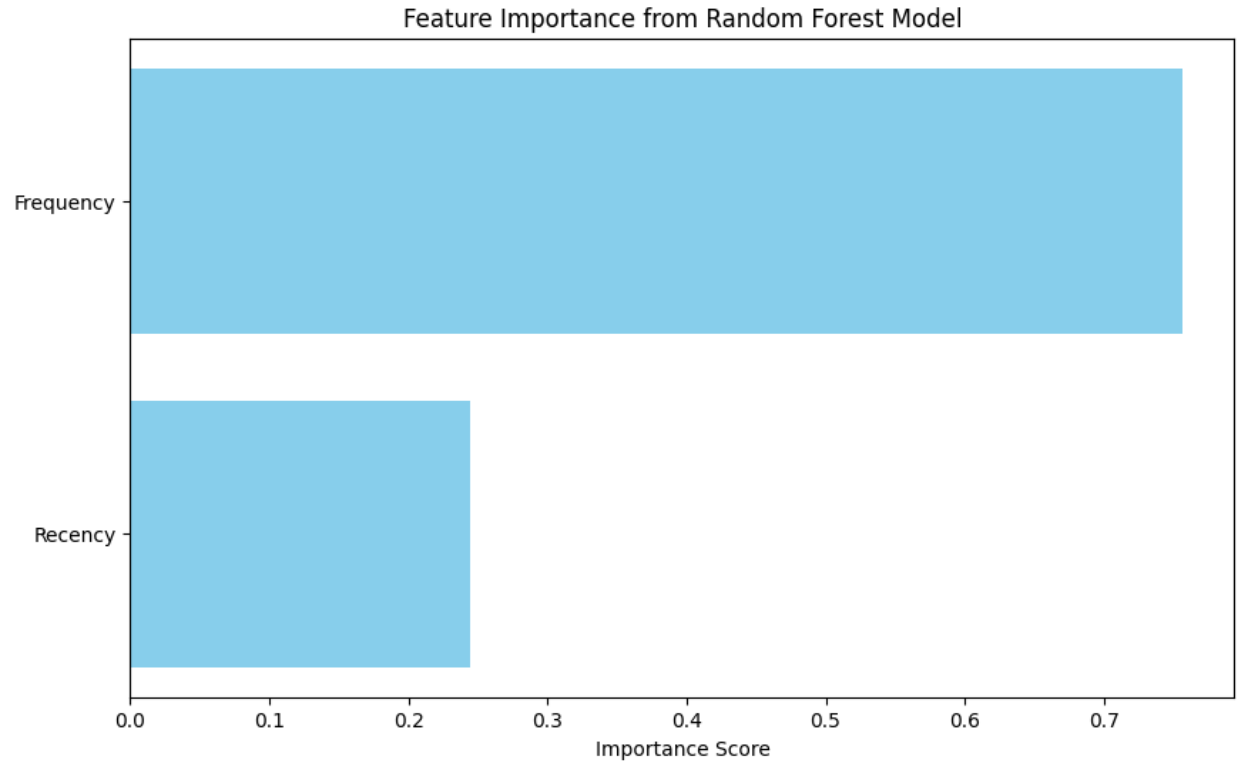

Figure 2: Feature Importance Plot for RF

The Random Forest Regressor model was then applied to predict Monetary_Sum, where *Frequency* was identified as the most critical feature (with a feature importance score of over 70%), followed by *Recency*. This model produced a Mean Absolute Error (MAE) of 505, Root Mean Squared Error (RMSE) of 37.27, and a negative R-squared value of -0.329, indicating issues with model accuracy.

To test the model's sensitivity to stock codes and duplicates, I created another model excluding C2, Next Day Carriage, and DOTCOM POSTAGE, while keeping POSTAGE due to its high purchase volume, which might represent a significant product purchase. This revised dataset contained 796,225 rows, and RFM features were recalculated. Upon training the Random Forest Regressor again, the MAE increased to 517.45, RMSE to 37.57, and R-squared remained negative at -0.46. Observing this increase in error, I decided to reintroduce duplicates, hypothesizing they might represent valid double orders.

In a subsequent model, duplicates were treated as double orders, leading to a dataset size of 807,890. The RFM features remained right skewed, and upon training, *Frequency* retained its position as the most critical feature. The error metrics improved, with an MAE of 434.21, RMSE of 36.67, and an R-squared value of -0.32. The decrease in error suggested that duplicates could improve model performance, possibly due to representing repeat purchases.

Finally, I prepared a refined dataset that excluded stock codes C2, Next Day Carriage, and DOTCOM POSTAGE while retaining duplicates, resulting in 807,890 rows with 5,854 unique customers.

Building on this refined dataset, I conducted additional feature extraction to enhance the model's predictive power. Key features added were:

- **RFM Metrics**: *Recency*, *Frequency*, *Monetary_Mean*, *Monetary_Sum*, *Monetary_Min*, and *Monetary_Max*.
- **Purchase Behavior**: *Unique_Items*, *UnitPrice_Mean*, *UnitPrice_Min*, *UnitPrice_Max*, *UnitPrice_Var*.
- **Time-Based Purchases**: *Frequency_First_Two_Weeks*, *Monetary_First_Two_Weeks*, *Frequency_Last_Two_Weeks*, *Monetary_Last_Two_Weeks*.
- **Weekday vs. Weekend Analysis**: *Frequency_Weekdays*, *Monetary_Weekdays*, *Frequency_Weekends*, *Monetary_Weekends*.
- **Order Intervals**: *Avg_Frequency*, *Median_Frequency_Per_Month*, *Std_Frequency_Per_Month*, *Mean_Days_Between_Orders*, *Std_Days_Between_Orders*, *Max_Days_Between_Orders*.

- **Repurchase Metrics**: *Repurchase_Absolute* and *Repurchase_Ratio*.

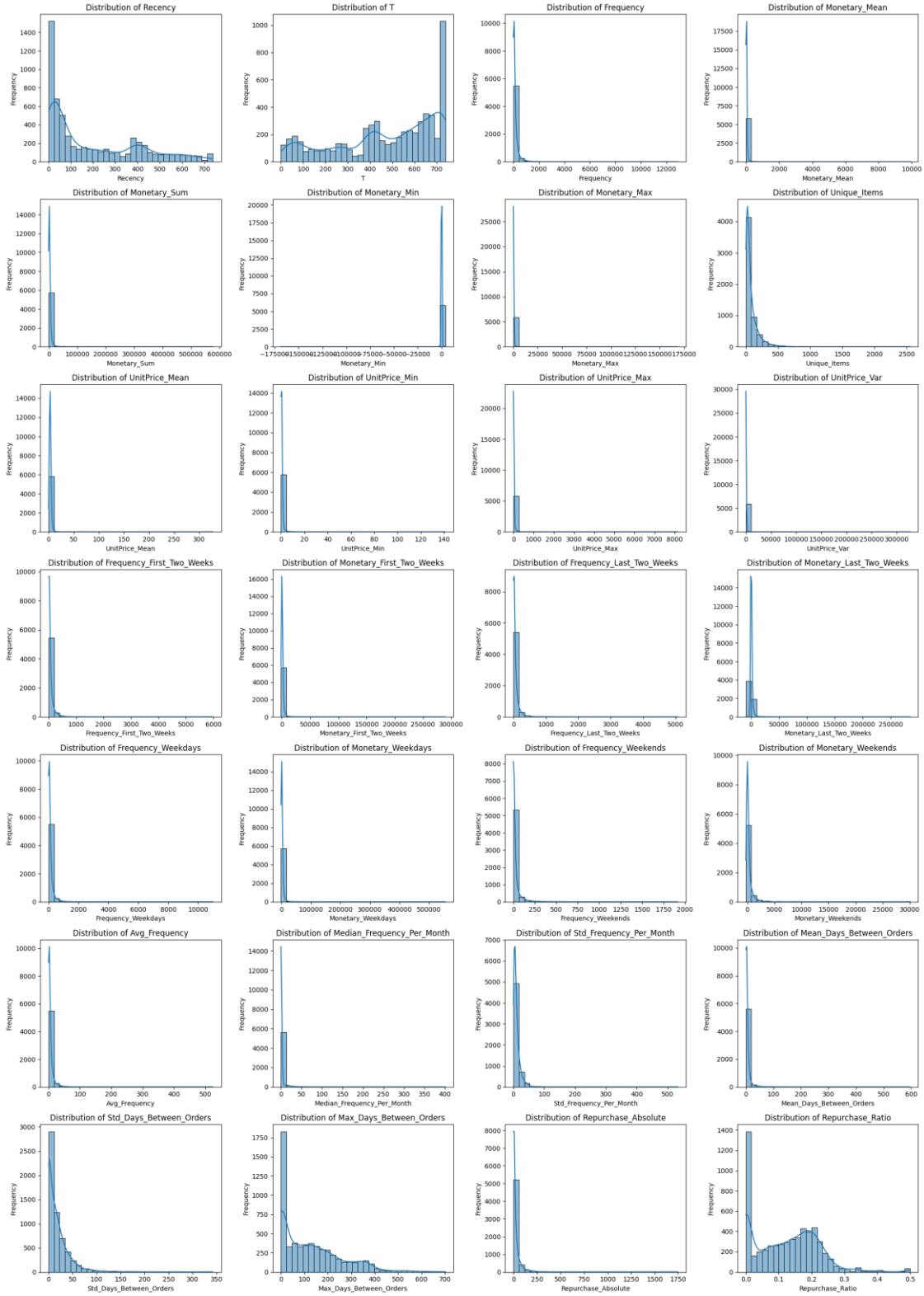

Figure 3: Visualization of the new features extracted

These features were plotted and found to be highly skewed (either right or left).

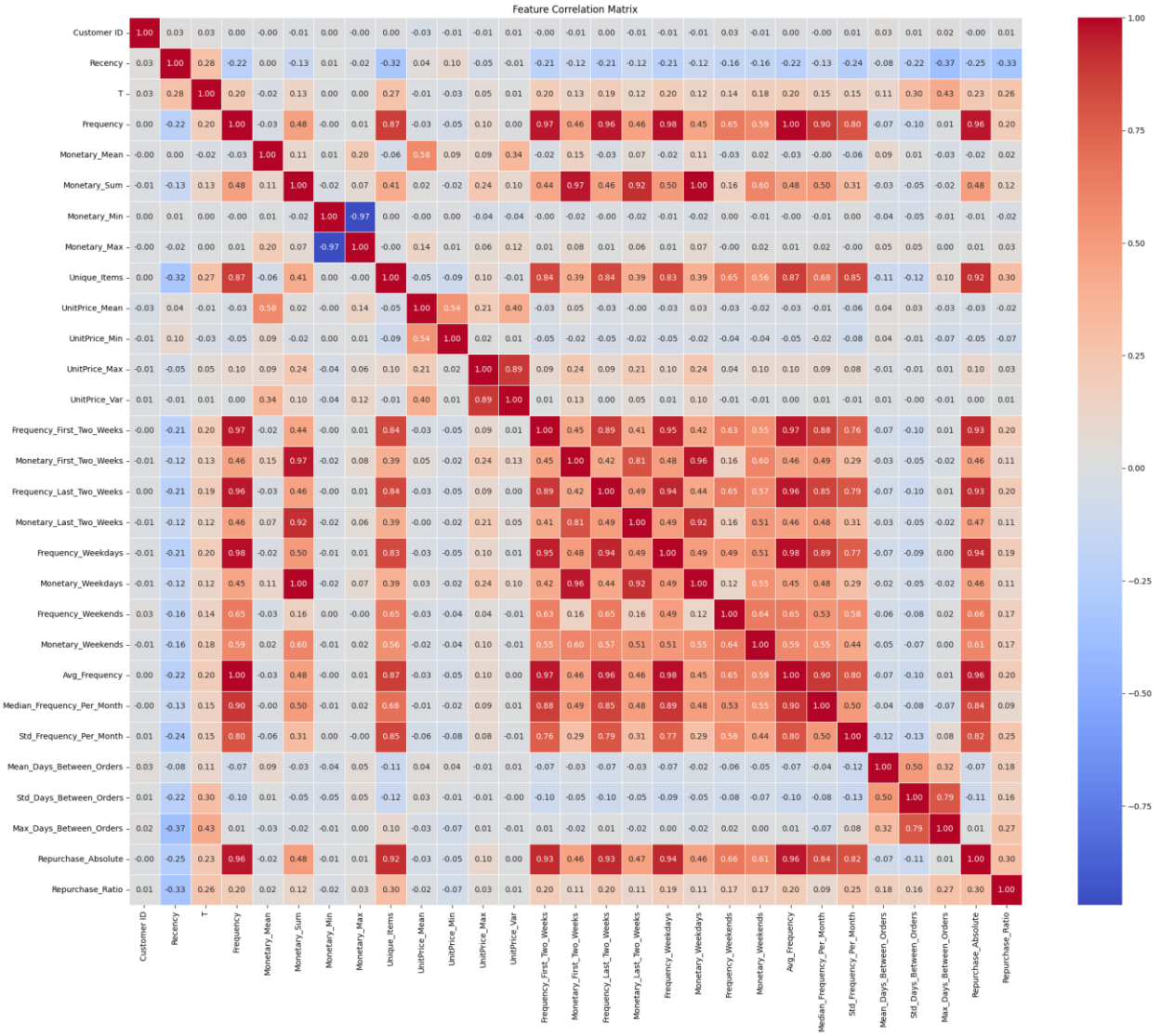

Figure 4: Correlation matrix of the features

A correlation matrix revealed strong multicollinearity, particularly for features related to frequency and monetary sums. Therefore, I selected a final feature set that excluded highly correlated variables. The excluded features are Unique_Items, UnitPrice_Var, Monetary_First_Two_Weeks, Monetary_Last_Two_Weeks, Repurchase_Absolute, Avg_Frequency, Std_Frequency_Per_Month, Frequency_First_Two_Weeks, Frequency_Last_Two_Weeks, Median_Frequency_Per_Month ,Frequency_Weekdays.

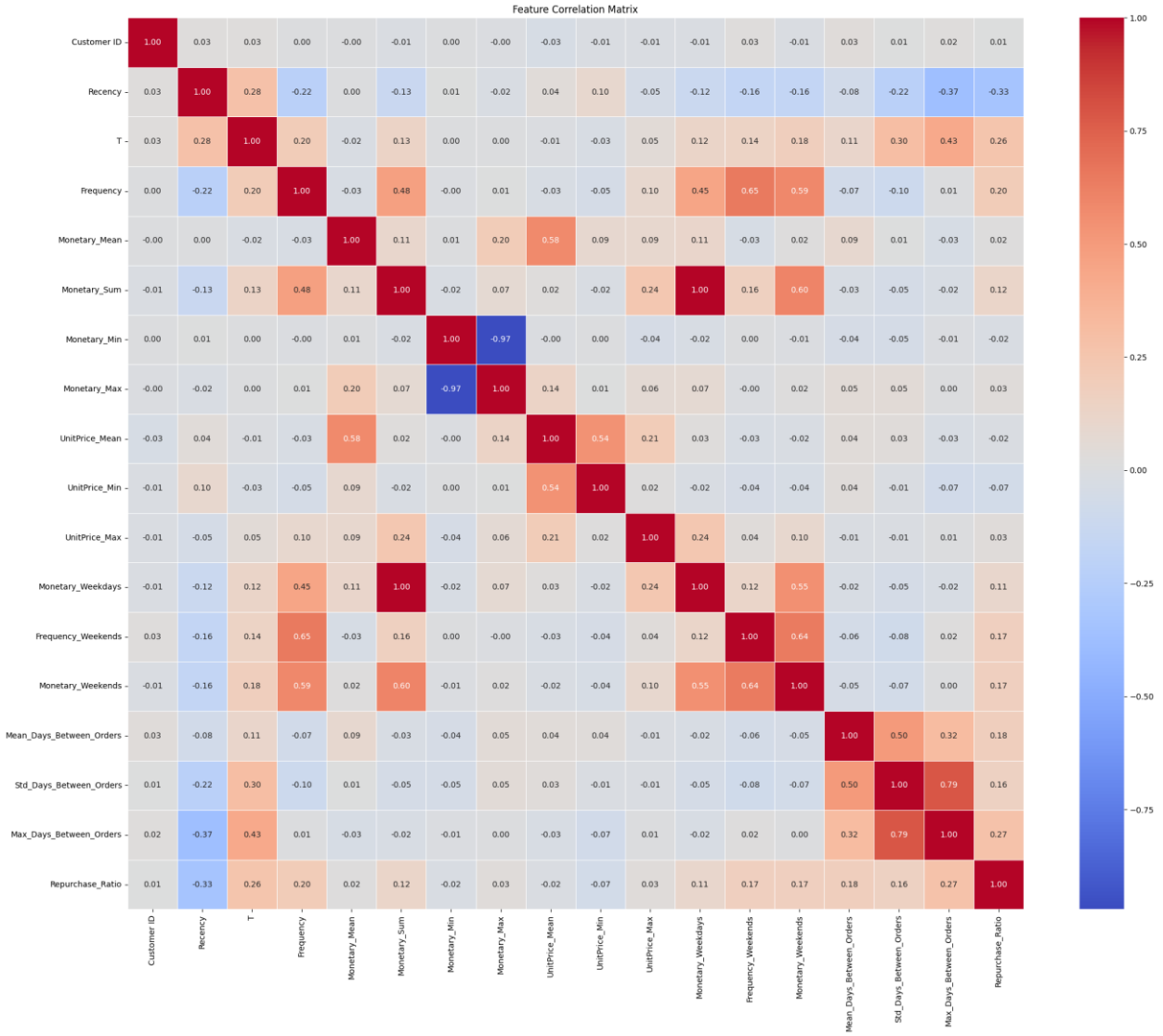

Figure 5: Correlation matrix after excluding highly correlated features

The correlation matrix shows a good heat map with Monetary_Weekdays being highly correlated with our target variable Monetary_Sum. Using this optimized feature set, I created training, validation, and test sets.

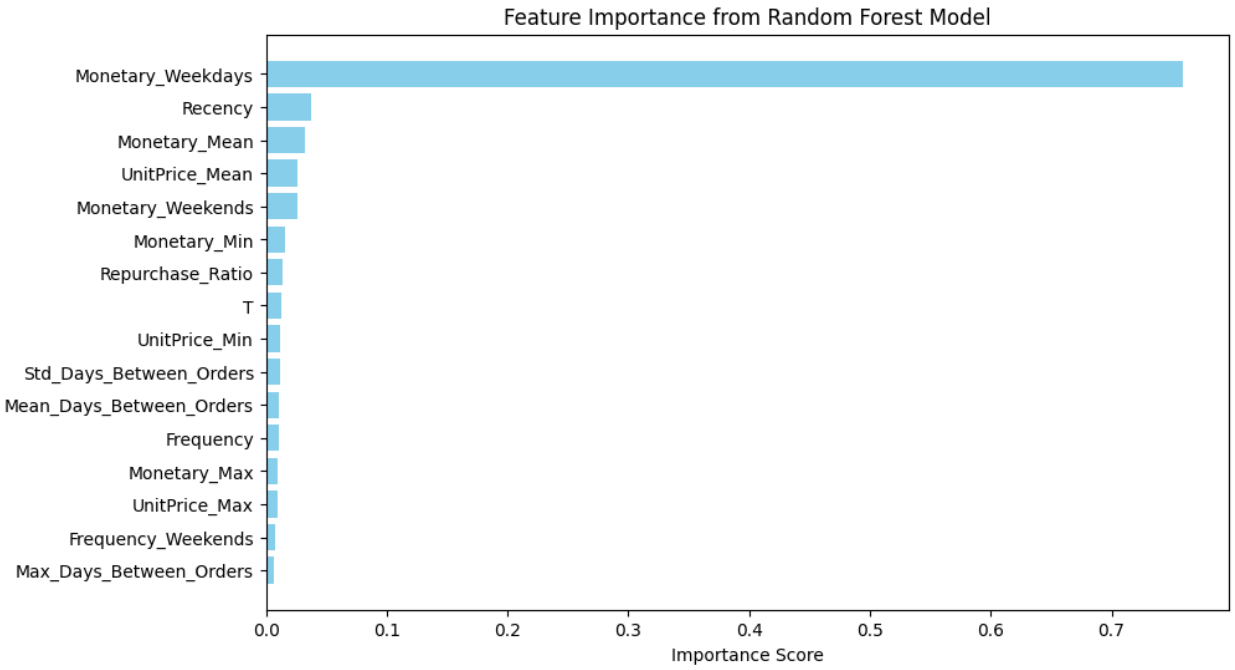

Figure 6: Feature importance plot

In this model, *Monetary_Weekdays* emerged as the most important feature (over 70% importance), followed by *Recency*, *Monetary_Mean*, *UnitPrice_Mean*, and *Frequency* was not as important compared to the previous base model. This model yielded an MAE of 490.65, RMSE of 33.67, and an R-squared value of 0.06. While the R-squared value was slightly positive, it was still very low.

To address the skewness issue, I used a Standard Scaler to normalize the features and then created a new Random Forest Regressor model.

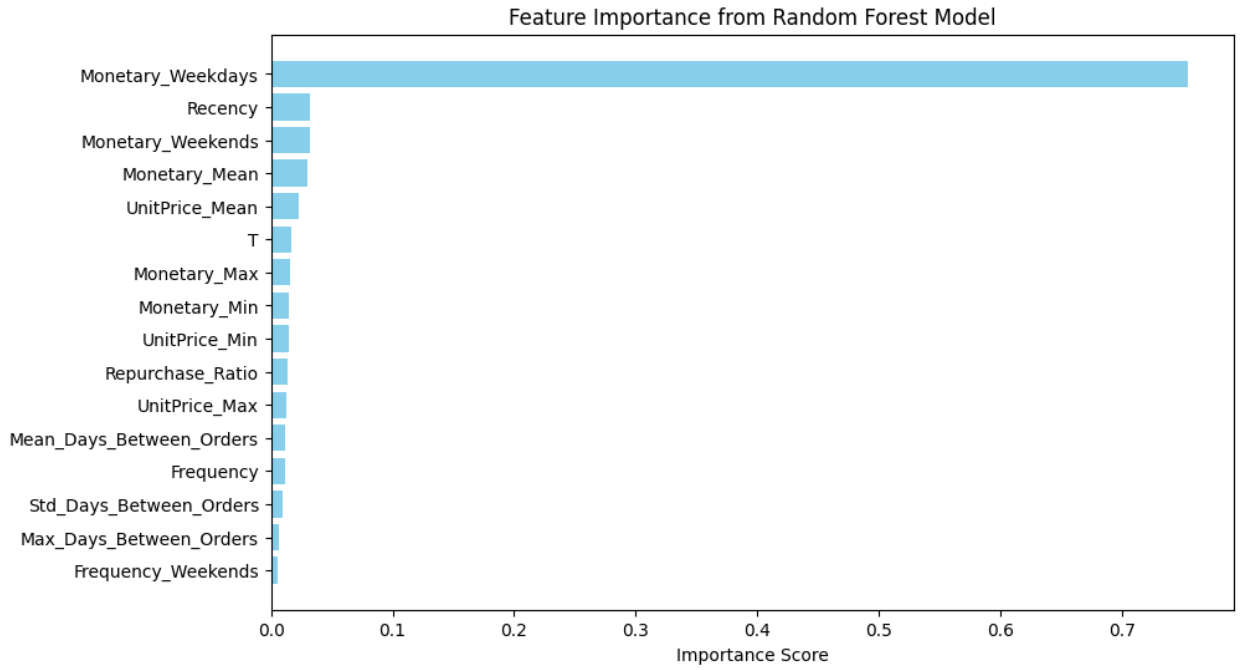

Figure 7: Feature Importance Plot

The feature importance order remained almost the same, with *Monetary_Weekdays* as the top predictor, but followed by *Recency*, *Monetary_Weekends*, and *Monetary _Mean*. This scaled model yielded a Mean Absolute Error (MAE) of 0.049, Root Mean Squared Error (RMSE) of 0.35, and an R-squared value of -0.077, indicating that the model still struggled to achieve high predictive accuracy despite scaling.

---

3. Progress and Milestones

Completed Tasks:

- Completed data cleaning, including the decision to retain POSTAGE and reintroduce duplicates.
- Created a new column for *Revenue* and implemented an RFM_features function to extract key metrics.
- Developed base models with different dataset versions, testing with and without specific stock codes and duplicates to observe their effects.
- Conducted advanced feature engineering and optimized the feature set by removing highly correlated variables, resulting in an improved correlation matrix.

- Built models using the optimized feature set and assessed feature importance and model performance metrics.

Pending Tasks:

- Explore PCA and additional dimensionality reduction techniques to address feature multicollinearity.

---

4. Problem-Solving and Challenges

A significant challenge was the consistently low or negative R-squared values across different models, indicating poor predictive accuracy. This issue persisted despite scaling the features using a Standard Scaler. The scaled model maintained the same feature importance order, with *Monetary_Weekdays* as the top predictor, yet it produced a negative R-squared value of -0.077, MAE of 0.049, and RMSE of 0.35.

Additionally, the strong skewness in features seemed to negatively impact the model's accuracy. Addressing this involved testing with scaled and unscaled data, as well as optimizing feature selection by removing multicollinear features. But still the problem was not mitigated.

---

5. Technical Depth and Accuracy

The current Random Forest model, despite thorough feature engineering, did not achieve expected predictive accuracy. The R-squared values, consistently low or negative, suggest that the features extracted may not capture all elements of customer purchasing behavior needed for accurate CLV prediction. Although *Frequency* initially showed high importance, later models revealed *Monetary_Weekdays* as more predictive, which reshaped the feature engineering strategy.

---

6. Future Plans and Goals

Moving forward, I will take the following steps to improve model performance by the guidance of Dr. Rostami and TA Utkarsh Pant:

1. **Drop Duplicates**: Completely remove duplicates and re-evaluate the model's baseline performance.

2. **Neural Network Modeling**: Experiment with neural networks on raw features (including StockCode, Quantity, Price, Revenue, Country, Year, Month, Day, Week, Hour, and potentially Description) to assess non-linear relationships.
3. **Raw Data Features in Random Forest**: Feed the raw dataset (including StockCode, Quantity, Price, Revenue, Country, Year, Month, Day, Week, Hour, and potentially Description) into the Random Forest model to evaluate any improvement in prediction.
4. **Classification of Monetary Value**: Categorize customers into high, normal, and low CLV segments to explore classification modeling.
5. **PCA Analysis**: Perform Principal Component Analysis (PCA) to reduce feature dimensionality and potentially capture better underlying structures in the data.
6. **Refund and Fraud Research**: Investigate chargeback or fraud-related features, incorporating knowledge on refunds and fraudulent transactions to improve CLV predictions.

The planned model refinements, coupled with new exploratory analyses on feature importance and potential classification approaches, will guide my efforts to improve model performance and address the R-squared challenges identified.