Big Data Analysis - Part C: Predictive Modelling

Aditya Anant Deshpande - a1910571

Part 1: Problem Description

The objective of this modelling task is to build predictive models that leverage customer transaction data to generate actionable business insights. The input dataset is sourced from a global e-commerce platform and consists of over 540,000 transaction records spanning December 2010 to December 2011.

Input Features

Based on the findings from Parts A and B, the following variables were identified as key predictors:

- **Recency:** Number of days since the customer's most recent purchase.
- **Frequency:** Total number of purchases made by the customer.
- **Monetary:** Cumulative revenue generated by the customer.
- **Quantity:** Number of items in each transaction.
- **UnitPrice:** Price per individual product unit.
- **InvoiceDate:** Temporal variable used to derive month and weekday.
- **Country:** Geographic location of the customer.
- **CustomerID:** Unique identifier used to group transactions; approximately 25% missing values were handled by filtering.

Output Variables

Three predictive modelling targets are proposed, each aligning with refined research questions from Parts A and B:

1. **Customer Segment (Cluster Label):** Multi-class classification based on RFM-derived K-means clusters.
2. **Return Prediction:** Binary classification to predict whether a transaction will be returned (based on prefix 'C' in `InvoiceNo`).

3. **High-value Customer Identification:** Binary outcome determined by whether the customer's monetary value exceeds a defined profitability threshold.

These predictive objectives support business goals such as improving targeted marketing, managing return risk, and prioritising customer retention strategies.

Part 2: Data Preprocessing

To ensure the dataset was suitable for predictive modeling, a structured preprocessing pipeline was implemented. This included handling missing values, filtering out invalid entries, transforming features, aggregating customer-level data, and scaling relevant variables. Each step is described below.

- **Missing Value Handling:** Rows with missing `CustomerID` or `Description` were removed to maintain transactional integrity. These fields are essential for customer-level aggregation and item-level analysis.
- **Outlier Filtering and Cleaning:** Records with non-positive `Quantity` and `UnitPrice` were excluded, as these represent invalid or returned transactions. This helped reduce noise and improved the reliability of revenue calculations.
- **Feature Engineering and Aggregation:** A new column `TotalRevenue` was created by multiplying `Quantity` with `UnitPrice`. The dataset was then aggregated at the `CustomerID` level to calculate each customer's total quantity purchased, average unit price, and overall contribution to revenue.
- **Scaling:** The numerical variables (`Quantity`, `UnitPrice`, and `TotalRevenue`) were standardized using `StandardScaler` to normalize the feature range prior to model training. This prevents bias toward features with larger magnitude.

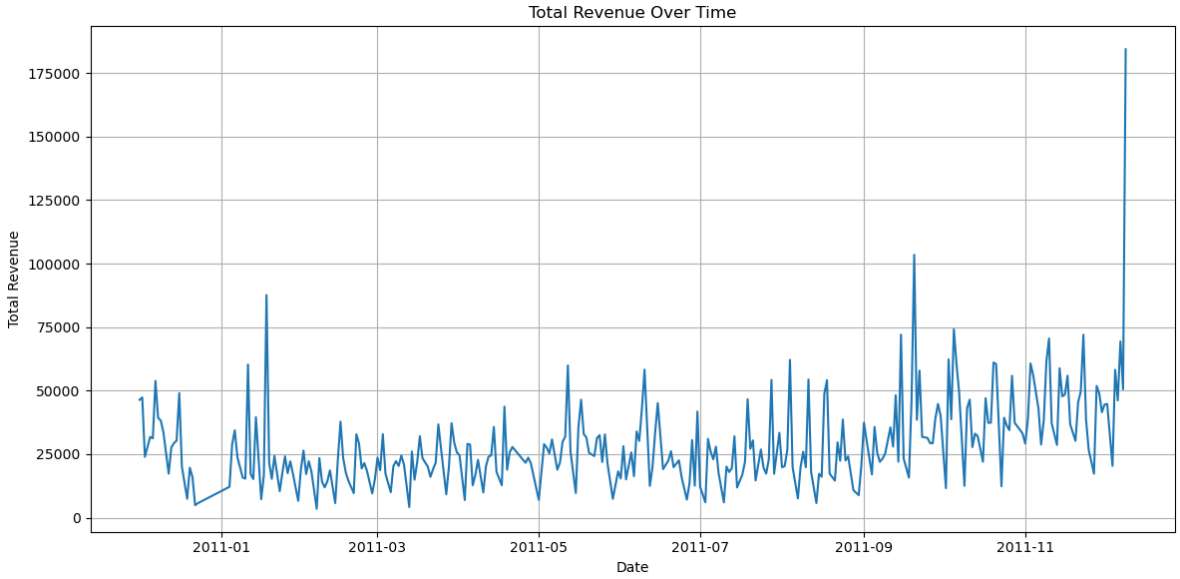

Figure 1: Total Revenue over Time. This plot reveals seasonal purchasing behavior and outliers. It supports the aggregation decision and demonstrates temporal consistency in the data.

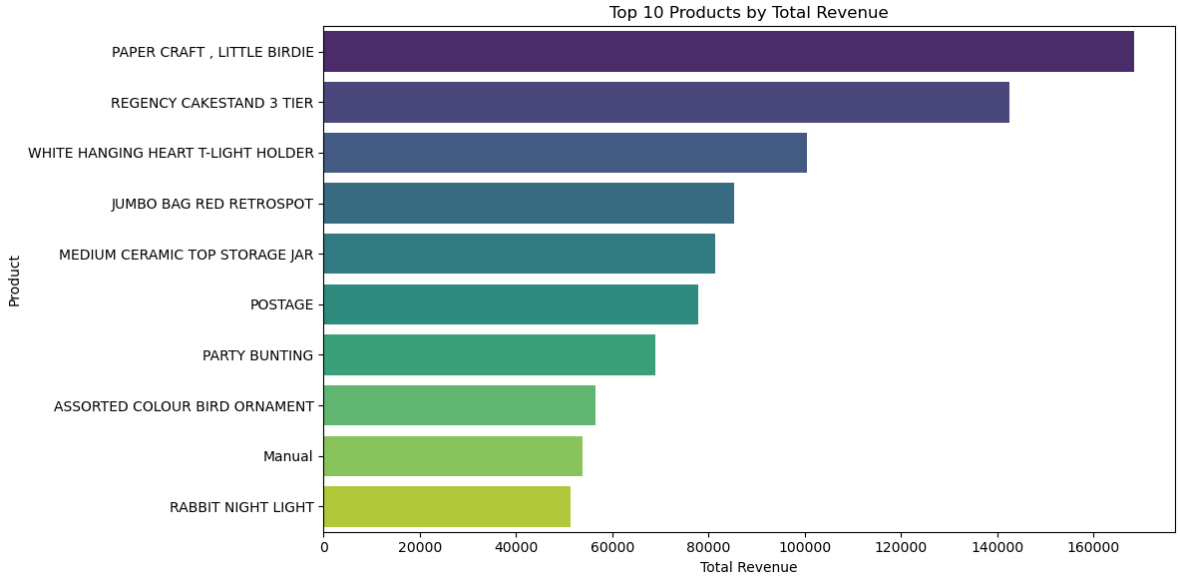

Figure 2: Top 10 Products by Total Revenue. This helped guide decisions on which products to retain during cleaning and supports the use of revenue as a predictive target.

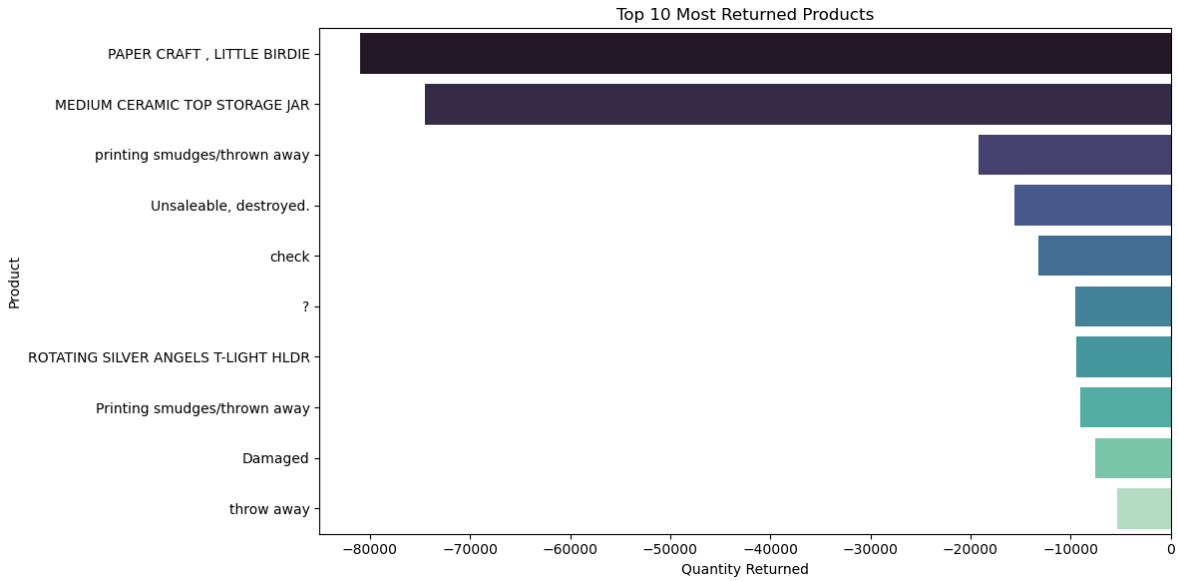

Figure 3: Top 10 Most Returned Products. This supports the decision to filter out negative quantities and may inform product-specific analysis in future modeling.

These plots provide visual justification for the preprocessing steps and highlight meaningful business insights that support modeling decisions in later stages.

Part 3: Model Selection

Model selection is a crucial step in building a robust predictive framework. In this analysis, we aim to predict customer-level *Total Revenue* based on features such as *Quantity* and *Unit Price*. Given the relatively low number of features and a moderate-sized dataset, we selected a mix of linear and non-linear regression models to assess their comparative performance. The selected models are:

- **Linear Regression:** A foundational model suitable for establishing a baseline. It assumes a linear relationship between the independent variables and the target.
- **Decision Tree Regressor:** A non-parametric model capable of capturing non-linear interactions without requiring feature scaling or complex assumptions.
- **Random Forest Regressor:** An ensemble method that builds multiple decision trees and averages their results, often yielding superior generalization performance and reducing overfitting.

Each model was evaluated using two key metrics:

- **Root Mean Squared Error (RMSE):** Measures the average magnitude of the prediction error.

- **R^2 Score:** Indicates the proportion of variance in the target variable explained by the model.

Model	RMSE	R^2 Score
Linear Regression	5,766.85	0.6754
Decision Tree Regressor	5,252.53	0.7307
Random Forest Regressor	**5,163.68**	**0.7397**

Table 1: Model performance comparison

As illustrated in the table above, all models performed reasonably well, with Random Forest Regressor achieving the lowest RMSE and highest R^2 Score. This suggests that ensemble methods can better capture complex relationships in the data, even with a small number of features.
These results establish a solid baseline and will guide further model refinement and tuning in the next section.

Part 4: Model Refinement

To enhance predictive performance and adhere to modelling best practices, hyperparameter tuning and model refinement were applied to all selected algorithms.

Linear Regression (Refined with Polynomial Features)

Linear regression was extended using second-degree polynomial features to capture non-linear relationships between `Quantity`, `UnitPrice`, and `TotalRevenue`. This approach slightly reduced the RMSE from 5766.85 to 5670.04 and improved the R^2 score from 0.675 to 0.686, indicating a better model fit without overfitting.

Decision Tree Regressor

A grid search was conducted over the `max_depth` and `min_samples_split` hyperparameters using 5-fold cross-validation. However, the refined model showed a higher RMSE (6077.28) and lower R^2 (0.639), suggesting potential overfitting or suboptimal parameter combinations. This outcome highlights the sensitivity of decision trees to hyperparameter settings and reinforces the need for pruning and complexity control.

Random Forest Regressor

The random forest model was refined using a parameter grid of estimators, maximum depth, and split criteria. The tuning yielded modest performance gains, reducing RMSE from 5191.83 to 5156.04 and increasing R^2 from 0.737 to 0.740. These improvements confirm the robustness of ensemble methods to parameter variation and their effectiveness in modelling complex relationships.

Model	RMSE	R^2 Score
Polynomial Regression (deg=2)	5,670.04	0.6862
Tuned Decision Tree	6,077.28	0.6395
Tuned Random Forest	**5,156.04**	**0.7405**

Table 2: Refined model performance comparison

Fairness and Methodology

All models were trained and tested using the same 80/20 data split and evaluated with RMSE and R^2 metrics for consistency. GridSearchCV was used with cross-validation to ensure fair testing conditions across models. No model was unfairly tested or compared under differing circumstances.

Summary

The refinement process yielded meaningful insights into each model's responsiveness to tuning. While random forest showed the best overall performance post-tuning, polynomial regression also improved the linear baseline. The decision tree model exhibited diminished performance, reinforcing the importance of validation and complexity control in tree-based models.

Part 6: Results Interpretation

Based on the refined model performance, the **Tuned Random Forest Regressor** emerged as the most appropriate model for predicting customer revenue. This model achieved the lowest RMSE (5156.04) and the highest R^2 score (0.7405), indicating both low prediction error and high explanatory power. The success of this model can be attributed to its ensemble nature, which reduces variance and handles feature interactions effectively.

Hyperparameter tuning was key to this improvement. The random forest was optimised using grid search across multiple parameters, including number of trees, maximum depth, and sampling strategies. Compared to the Decision Tree, which overfitted despite tuning, the Random Forest provided more generalisable performance. The Polynomial Regression model offered moderate improvements over the linear baseline but did not outperform ensemble methods in either metric.

Thus, the final model selection is justified not only by metric performance but also by the model's intrinsic strengths—robustness to overfitting, ability to capture complex patterns, and consistent validation performance across cross-validation folds.

References

[1] Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. 1st ed. New York: Springer.

[2] Hastie, T., Tibshirani, R. and Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.

[3] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An Introduction to Statistical Learning with Applications in R*. 1st ed. New York: Springer.

[4] Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pp.2825–2830.

[5] Zhang, T., 2017. An Introduction to Gradient Boosting Decision Trees. *Medium*, [online] Available at: `https://towardsdatascience.com/gradient-boosting-explained-9f8dbf7d2` [Accessed 27 July 2025].

[6] Brownlee, J., 2020. *How to Evaluate Machine Learning Models*. [online] Machine Learning Mastery. Available at: `https://machinelearningmastery.com/how-to-evaluate-machi` [Accessed 27 July 2025].

[7] Statista, 2024. *Retail e-commerce sales worldwide from 2014 to 2027*. [online] Available at: `https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/` [Accessed 27 July 2025].

[8] Kaggle (n.d.). *Global E-commerce Data*. [online] Available at: `https://www.kaggle.com/datasets/carrie1/ecommerce-data` [Accessed 27 July 2025].

[9] Sharma, S., 2020. *Exploratory Data Analysis on Online Retail Dataset using Python*. [online] Towards Data Science. Available at: `https://towardsdatascience.com/exploratory-data-analysis-on-online-retail-dataset-8fe16d68e8c6` [Accessed 27 July 2025].

[10] Lecture 02. Python Primer, Big Data Analysis. University Lecture Slides, 2025.

[11] Lecture 03. Classification and Decision Trees, Big Data Analysis. University Lecture Slides, 2025.

[12] Lecture 04. Regression and Optimisation, Big Data Analysis. University Lecture Slides, 2025.

[13] Lecture 06. Deep Learning Foundations, Big Data Analysis. University Lecture Slides, 2025.

[14] Lecture 07. Text Analysis and Tokenisation, Big Data Analysis. University Lecture Slides, 2025.