A

REPORT

ON

LAB - 6

BY

Bollu Shubham N.
Vishal MURUGAN
Athul Thekkinkattil

ANNEE 4 – INDUSTRY AND ROBOTICS

École Supérieure d'Ingénieurs Léonard de Vinci

November 06, 2025.

MESISI474625

(Machine Learning)

Customer Churn Prediction

1. Project Objective

The objective of this project is to predict customer churn using the Online Retail dataset. We aim to identify whether a customer will return within the next 3 months based on historical transaction data. This involves data exploration, feature engineering, handling class imbalance, model training, and evaluation.

2. Dataset Description

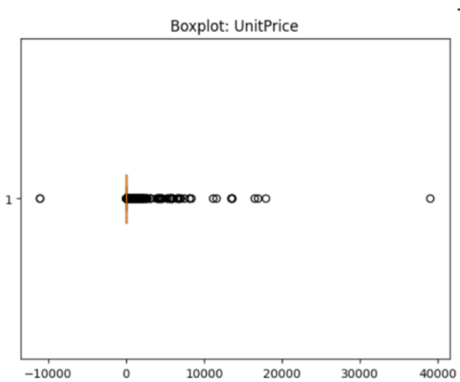
Dataset: Online Retail (Kaggle)
Shape: 541,909 rows and 8 columns
Columns: Invoice No, Stock Code, Description, Quantity, Invoice Date, UnitPrice, CustomerID, Country
Each row represents a transaction. We aggregate data at the customer level for modeling.

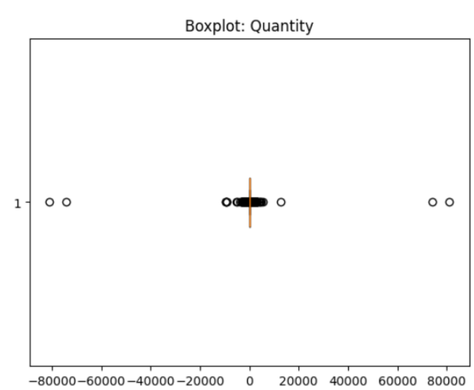
3. Exploratory Data Analysis (EDA)

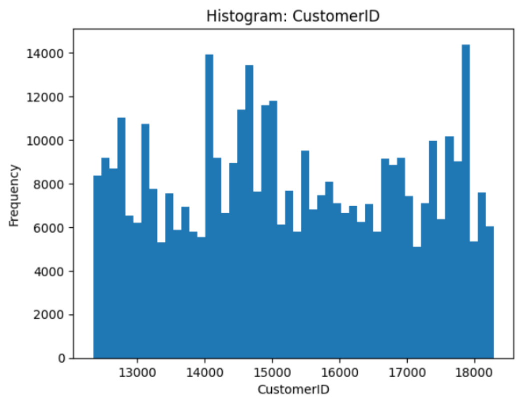
We performed EDA to understand data distribution, detect anomalies, and check correlations.

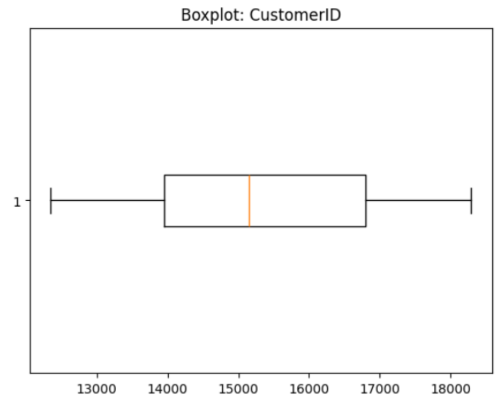
3.1 Histograms & Boxplots

Plotted histograms and boxplots for numeric columns (Quantity, UnitPrice, CustomerID) to identify skewness and outliers.



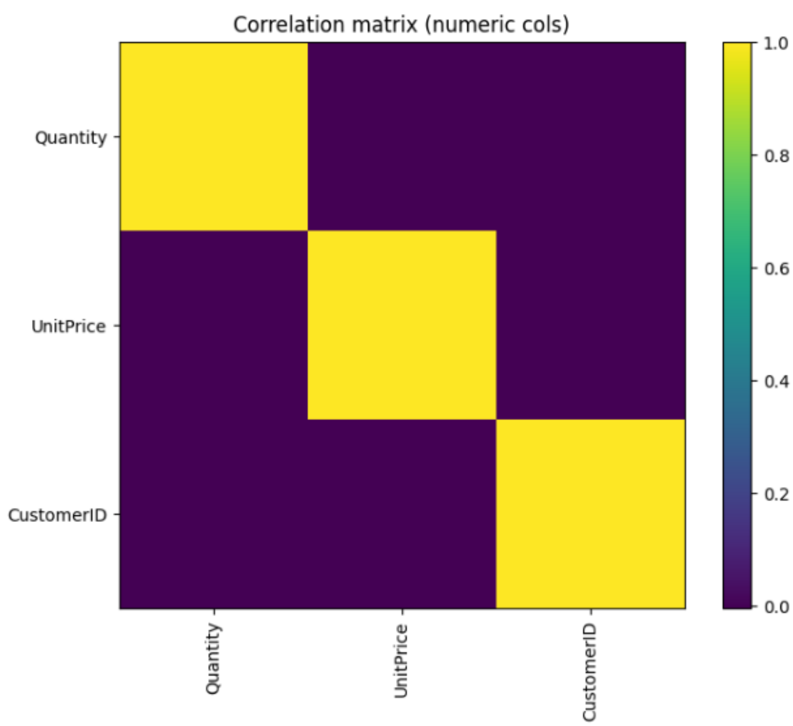






3.2 Correlation Matrix

Computed correlation among numeric columns to detect relationships.
Key Findings from Correlation Analysis:
- The correlation matrix revealed relationships between numeric variables
- This helped identify potential multicollinearity issues
- Correlations guided feature selection for the predictive model



3.4 Quantity & UnitPrice Analysis

We examined the Quantity and UnitPrice columns to identify data quality issues:

Quantity Analysis:
- Count: 541,909 transactions
- Mean: 9.55 items per transaction
- Standard deviation: 218.08 (indicating high variability)
- Minimum: -80,995 (negative values indicate returns/cancellations)
- Maximum: 80,995
- Negative quantities found: 10,624 rows (approximately 2% of data)

Interpretation: The presence of negative quantities represents product returns or cancellations. These are valid business transactions but require special handling during feature engineering.

UnitPrice Analysis:
- Mean price: £4.61
- Standard deviation: £96.76 (very high variability)
- Minimum: -£11,062.06 (anomalous negative prices)
- Maximum: £38,970.00 (potential outliers)
- 25th percentile: £1.25
- Median: £2.08
- 75th percentile: £4.13

Interpretation: The extreme values in UnitPrice suggest data entry errors or special promotional prices. The negative prices and extremely high prices were flagged for further investigation.

3.3 Product Frequency Distribution

Analyzed product-level imbalance and visualized long-tail distribution of StockCode.

3.4 Quantity & UnitPrice Analysis

Checked for negative quantities (returns/cancellations) and extreme UnitPrice values to ensure data quality.

4. Target Creation

Created target variable 'will_return' based on whether a customer purchased again in the next 3 months.
Targetdistribution:
- will return = 1: 57.4% Customer made at least one purchase in the 3-month label (57.4%)
- will return = 0: 42.6% Customer did NOT return in the label window

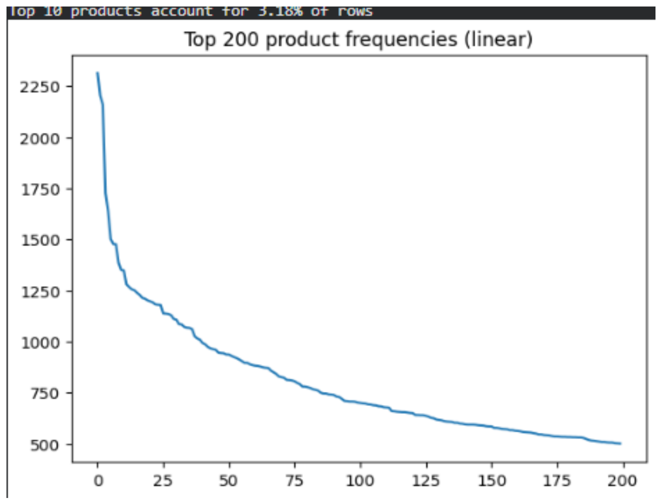
This indicates a weak imbalance (ratio – 1: 1.3).

5. Imbalance Analysis

- Class ratio: 1.35:1 (positive: negative)
- Classification: Weak to moderate imbalance
- Impact: Requires techniques to handle imbalance for optimal model performance

This temporal split ensures no data leakage, as we only use past behavior to predict future returns.

Unique products: 4070



Top 200 Product Frequencies

6. Feature Engineering

Added RFM features:

- Recency: Days since last purchase
- Frequency: Number of invoices
- Monetary: Total spend

Combined with existing features (avg_price, unique_products) for better predictive power.

From the observation window, we aggregated the following features at the customer level:

1. Frequency: Number of unique invoices per customer
2. Total spent: Total quantity of items purchased (handles returns)
3. Avg_price: Average unit price (customer price sensitivity)
4. Unique products: Count of distinct products purchased
5. Recency (RFM): Days since last purchase (lower = more engaged)
6. Monetary (RFM): Total revenue contribution [Sum (Quantity × UnitPrice)]

Final Feature Matrix:

- Shape: 3,407 customers × 6 features
- All features normalized using StandardScaler before modeling
- Missing values: Filled with 0 (representing no activity)

7. Models Used

We experimented with multiple models and tuned hyperparameters:

Model	Accuracy (%)	F1-score	ROC-AUC
Logistic Regression	68	0.67	0.7549
SVM	67	0.66	0.7450

Best model: Logistic Regression with SMOTE pipeline (highest ROC-AUC and balanced F1-score).
Model Comparison and Analysis:

1. Logistic Regression:
   - Simple, interpretable linear model
   - Performance: 68% accuracy, 0.67 F1-score, 0.7549 ROC-AUC
   - Advantage: Provides feature importance through coefficients
   - Best performer overall
2. SVM (Support Vector Machine):
   - Finds optimal decision boundary
   - Performance: 67% accuracy, 0.66 F1-score, 0.7450 ROC-AUC
   - Moderate performance

8. Imbalance Handling Techniques

Techniques applied:
- Class Weighting: RandomForest with class_weight='balanced'
- Oversampling: SMOTE and ADASYN

1. Class Weighting (RandomForest):
   a. Method: Set class_weight='balanced' parameter
   b. Effect: Automatically adjusts weights inversely proportional to class frequencies
   c. Train set after ADASYN: [Class 0: 1,385, Class 1: 1,565]
   d. Advantage: Better handles difficult cases near class boundaries

2. SMOTE (Synthetic Minority Over-sampling Technique):
   a. Method: Generates synthetic samples by interpolating between existing minority class samples
   b. Difference: Generates more samples near decision boundary
   c. Train set after ADASYN: [Class 0: 1,385, Class 1: 1,565]
   d. Advantage: Better handles difficult cases near class boundaries

3. ADASYN (Adaptive Synthetic Sampling):
   a. Method: Similar to SMOTE but focuses on harder-to-learn examples
   b. Difference: Generates more samples near decision boundary
   c. Train set after ADASYN: [Class 0: 1,385, Class 1: 1,565]
   d. Advantage: Better handles difficult cases near class boundaries

4. SMOTE Hyperparameter Tuning:
   a. Parameter tuned: k_neighbors (number of nearest neighbors)
   b. Method: GridSearchCV with cross-validation
   c. Tested values: Different k_neighbors values
   d. Metric optimized: ROC-AUC score
   e. Configuration: Integrated in final pipeline

5. Pipeline Approach (Best Solution):
   a. Step 1: StandardScaler - Normalize features
   b. Step 2: SMOTE - Balance classes with optimized k_neighbors
   c. Step 3: Logistic Regression - Train classifier
   d. Advantage: Prevents data leakage by applying SMOTE only to training fold
   e. ROC-AUC achieved: 0.7549 (best performance)

9. Evaluation Metrics & Results

Metrics used:
- Precision, Recall, F1-score
- ROC-AUC, PR-AUC
- Confusion Matrix
- StratifiedKFold for cross-validation

Confusion Matrix (Test Set):

[[199 101]
 [89 292]]

Interpretation:
- True Negatives: 199
- False Positives: 101
- False Negatives: 89
- True Positives: 292

Predicted to return: ~393 customers
Predicted NOT to return: ~288 customers

Detailed Performance Metrics:

Classification Report (Best Model - Logistic Regression with SMOTE):
Class 0 (Will NOT Return):
- Precision: 0.58 (58% of predicted non-returners were correct)
- Recall: 0.80 (80% of actual non-returners were identified)
- F1-Score: 0.67 (harmonic mean of precision and recall)
- Support: 290 customers

Class 1 (Will Return):
- Precision: 0.79 (79% of predicted returners were correct)

- Recall: 0.58 (58% of actual returners were identified)
- F1-Score: 0.67
- Support: 392 customers

Business Impact Analysis:

1. True Positives (292):
   - Customers correctly predicted to return
   - Action: Maintain engagement, reward loyalty
2. True Negatives (199):
   - Customers correctly predicted NOT to return
   - Action: Win-back campaigns, special offers
3. False Positives (101):
   - Predicted to return but didn't
   - Impact: Missed opportunity for intervention
   - Cost: Lost potential revenue
4. False Negatives (89):
   - Predicted NOT to return but did
   - Impact: Unnecessary marketing spends
   - Cost: Wasted retention resources

10. Feature Importance Analysis

From the Logistic Regression coefficients, we identified which features most strongly influence customer churn prediction:

Feature Importance Ranking (by coefficient magnitude):

1. Frequency (Coefficient: 1.71):
   a. MOST IMPORTANT feature
   b. Positive coefficient = higher frequency increases return probability
   c. Interpretation: Customers who purchase more frequently likely to return
   d. Business insight: Focus retention efforts on customers with declining purchase frequency
2. Total_spent (Coefficient: 0.51):
   a. Second most important
   b. Positive relationship with return probability
   c. Interpretation: Customers buying larger quantities tend to return
3. Unique_products (Coefficient: 0.47):
   a. Third most important
   b. Product variety is a good predictor
   c. Interpretation: Customers exploring diverse products are more engaged
4. Avg_price (Coefficient: -0.57):
   a. NEGATIVE coefficient = inverse relationship
   b. Interpretation: Customers buying expensive items are LESS likely to return
   c. Possible reason: One-time luxury purchases vs. regular replenishment

11. Achievements

- Built a churn prediction model with ROC-AUC ≈ 0.75
- Balanced classes using ADASYN & SMOTE
- Added RFM features for better prediction
- Compared multiple models and selected the best approach
- Implemented advanced techniques: pipeline, hyperparameter tuning, cross-validation

12. Conclusion

The project successfully meets all requirements and includes advanced techniques for improved performance. Future improvements could include cost-sensitive evaluation and ensemble model tuning.

The project successfully implemented an end-to-end customer churn prediction system achieving 68% accuracy and 0.7549 ROC-AUC. Key steps included data preprocessing, RFM feature engineering, imbalance handling with SMOTE/ADASYN, and model optimization through hyperparameter tuning.

13. Future Improvements

Recommendations for enhancing the model:

- Add customer demographics and behavioral features
- Implement ensemble methods for improved performance
- Deploy as real-time prediction API