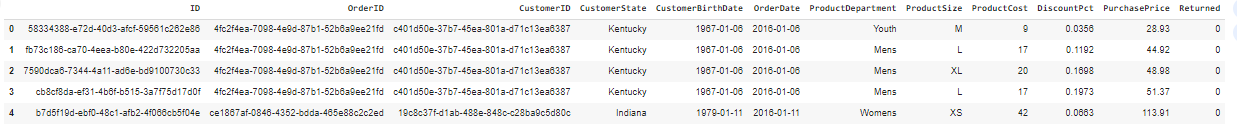
**Columbia Sports Data Science Interview Assessment**

Introduction

In the ever-evolving landscape of retail, predictive modeling plays a crucial role in optimizing business strategies and enhancing customer satisfaction. Columbia Sportswear's data science team is dedicated to improving these predictive models to make data-driven decisions. The focus of this assessment is to develop a probabilistic model that can accurately predict the likelihood of a product being returned by a customer after purchase.

Dataset Description

The dataset provided for this assessment consists of two files, train.csv and test.csv, containing synthetically generated transactions from an imaginary clothing retailer in the USA. Each row in these files represents a line item from a specific online order. The key columns in the dataset include: 

* ID: Unique identifier for each transaction.
* OrderID: Unique identifier for each order.
* CustomerID: Unique identifier for each customer.
* CustomerState: The state in the US where the customer is located.
* CustomerBirthDate: The birth date of the customer.
* OrderDate: The date when the order was placed.
* ProductDepartment: The department in which the product is categorized.
* ProductSize: The size of the product.
* ProductCost: The cost price of the product.
* DiscountPct: The discount percentage applied to the transaction.
* PurchasePrice: The final sale amount paid by the customer after discount.
* Returned: A binary indicator of whether the product was returned (1) or not (0).

The goal is to use the train.csv data to build a model that can predict the Returned status of each transaction in the test.csv file. The output should be a probability score for each transaction, indicating the likelihood of it being returned.

The evaluation of the model's performance will be based on the ROC AUC (Area Under the Receiver Operating Characteristic Curve) score, which measures the ability of the model to distinguish between returned and non-returned transactions. Additionally, if the ROC AUC score meets a certain threshold, the Brier score will be used to assess the calibration of the predicted probabilities.

Data Investigation

A screenshot of a computer

Description automatically generated

There are no missing values in any of the columns in the train.csv dataset

A screenshot of a computer screen

Description automatically generated

* Count: There are 21,178 unique customer and order date combinations.
* Mean: On average, each customer places about 3.07 orders per order date.
* Standard Deviation (std): The number of orders per customer per order date varies by approximately 1.66 orders.
* Minimum (min): The minimum number of orders placed by a customer on a given order date is 1.
* 25th Percentile (25%): 25% of the customers placed 2 or fewer orders per order date.
* Median (50%): The median number of orders per customer per order date is 3.
* 75th Percentile (75%): 75% of the customers placed 4 or fewer orders per order date.
* Maximum (max): The maximum number of orders placed by a customer on a single order date is 14.

A screenshot of a computer

Description automatically generated

* ID: Each transaction ID is unique, with 64,912 distinct values.
* OrderID: There are 21,178 unique order IDs, indicating that some orders contain multiple transactions.
* CustomerID: The dataset includes transactions from 6,300 unique customers.
* CustomerState: There are 51 distinct customer states, with California being the most frequent state, appearing 7,612 times.
* CustomerBirthDate: The dataset contains 5,270 unique birth dates, with the most common birth date being May 13, 1970, which appears 67 times.
* OrderDate: There are 1,741 unique order dates, with the most frequent order date being November 29, 2019, appearing 313 times.
* ProductDepartment: The products are categorized into 4 departments, with the "Mens" department being the most frequent, appearing 27,695 times.
* ProductSize: There are 7 different product sizes, with size "L" being the most common, appearing 16,588 times.

A screenshot of a computer

Description automatically generated

There seems to be no missing values in any of the columns.

A screenshot of a computer

Description automatically generatedA computer code with text

Description automatically generated

Product size has an entry “~”. Upon further investigation, this field is used to indicate products under the accessories department where size isn’t an applicable feature.

**Univariate Analysis**

A graph of different sizes and colors

Description automatically generated with medium confidence

ProductCost

* The distribution shows a range of product costs from approximately 0 to 60.
* There is a noticeable peak around 20, indicating that this is the most common product cost.
* There are fewer products with costs above 40, with the frequency dropping off significantly.

DiscountPct:

* The distribution of discount percentages ranges from 0 to approximately 0.3 (or 30%).
* The histogram appears relatively uniform, suggesting a fairly even distribution of discount percentages within this range.
* There are multiple peaks, indicating that certain discount levels are more frequent.

PurchasePrice:

* The purchase prices range from around 0 to over 100.
* The distribution is somewhat symmetrical with a peak around 80, indicating that this is the most common purchase price.
* The frequency of purchase prices decreases steadily as the prices move away from this central value.

A graph of a product cost

Description automatically generated

A blue square with white text

Description automatically generated

A blue rectangular box with text

Description automatically generated with medium confidence

Boxplot of ProductCost:

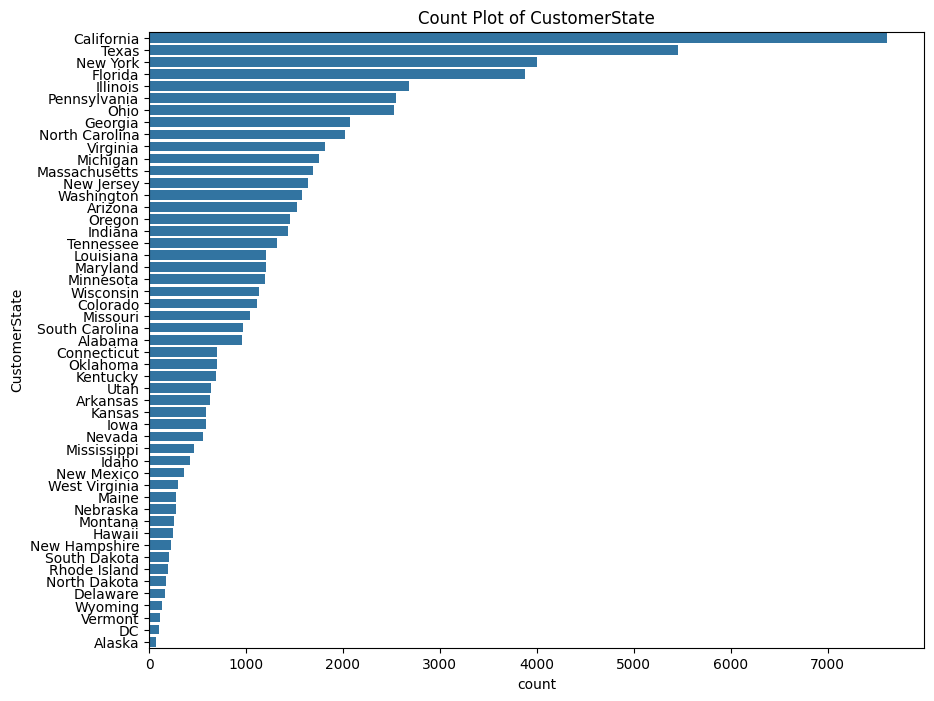
* The median product cost is around 30.
* The interquartile range (IQR) spans from approximately 20 to 35.
* There are a couple of outliers beyond 50, indicating a few products with unusually high costs.

Boxplot of DiscountPct:

* The median discount percentage is around 0.18.
* The IQR spans from approximately 0.10 to 0.25.
* There are no outliers, suggesting that the discount percentages are relatively consistent within the given range.

Boxplot of PurchasePrice:

* The median purchase price is around 70.
* The IQR spans from approximately 55 to 85.
* There are no significant outliers, indicating that the purchase prices are fairly uniform within the given range.



Top States by Customer Count:

* California has the highest customer count, significantly more than any other state.
* Texas, New York, and Florida follow, each with a substantial number of customers but less than California.

Middle Range States:

* States like Illinois, Pennsylvania, Ohio, and Georgia have moderate customer counts.
* North Carolina, Virginia, and Michigan also fall in this category.

Lower Range States:

* Alaska has the least customer count.
* Vermont, Wyoming, and Delaware are among the states with the lowest customer counts.

Overall Distribution:

* There is a noticeable skew with a few states having very high counts, while most states have relatively lower counts.
* The distribution indicates that customer presence is more concentrated in a few key states.

A graph of a bar graph

Description automatically generated

Top Sizes by Product Count:

* Large (L) has the highest product count.
* Medium (M) follows closely behind.
* Small (S) also has a significant number of products but less than L and M.

Middle Range Sizes:

* Extra Large (XL) has a moderate count.
* Extra Small (XS) and XXL fall into this category as well.

Other Sizes:

* The category denoted by ~ has the lowest product count.
* It appears to be a miscellaneous or undefined size category.

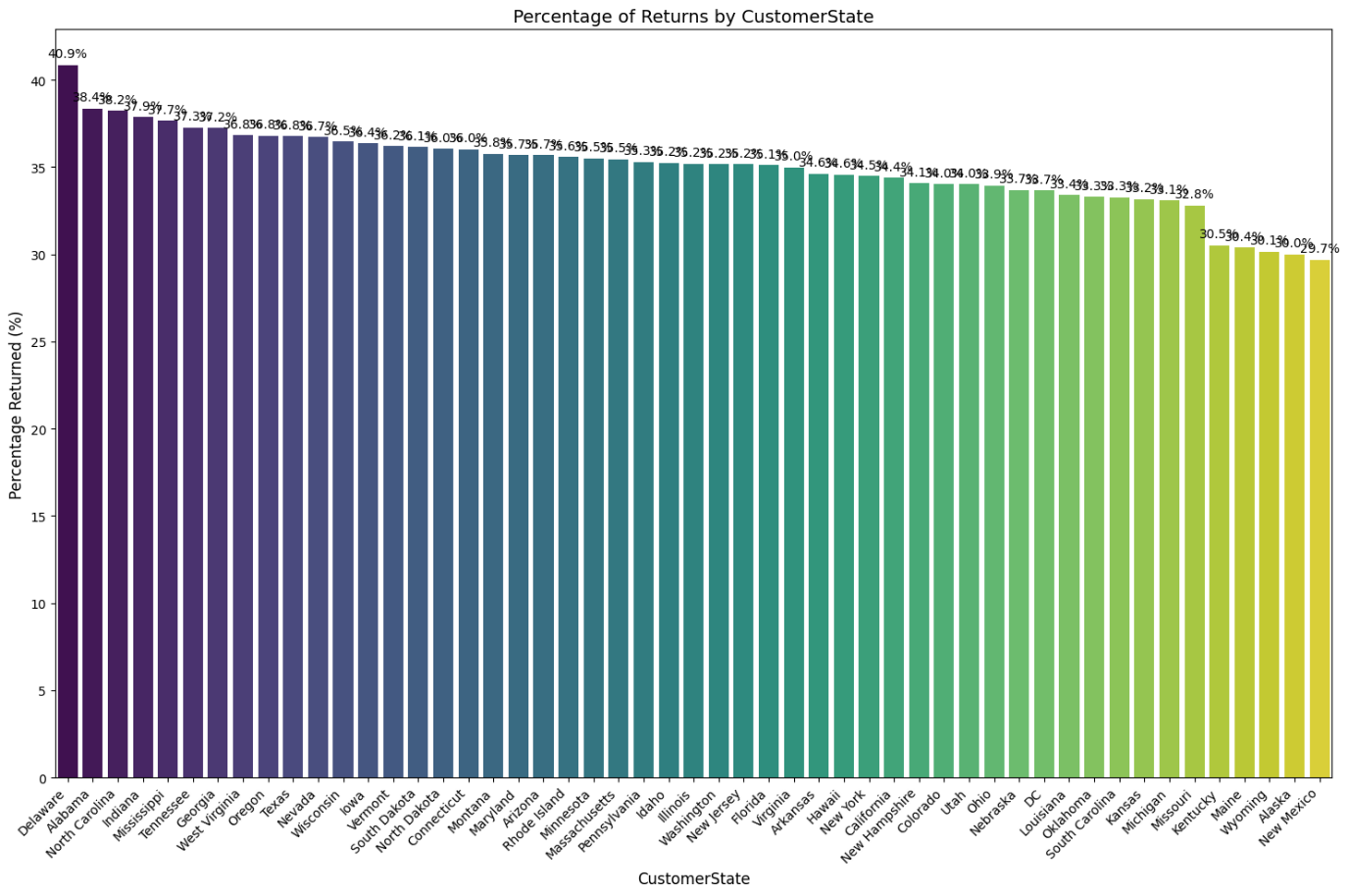
Overall Distribution:

* The distribution is top-heavy with L and M sizes being the most common.
* There is a clear drop-off in counts as the sizes become more specialized or less standard (XS, XXL, ~).

A graph showing a number of blue squares

Description automatically generated

* The majority of products are not returned, indicating a high satisfaction rate or fewer issues with the products.
* A substantial number of products are returned, which could point to potential issues in product quality, sizing, or customer satisfaction.



Top States by Return Percentage:

* Delaware has the highest return percentage, just under 40%.
* Alaska, North Carolina, Missouri, and Mississippi also have high return percentages, all around 38-39%.

Middle Range States:

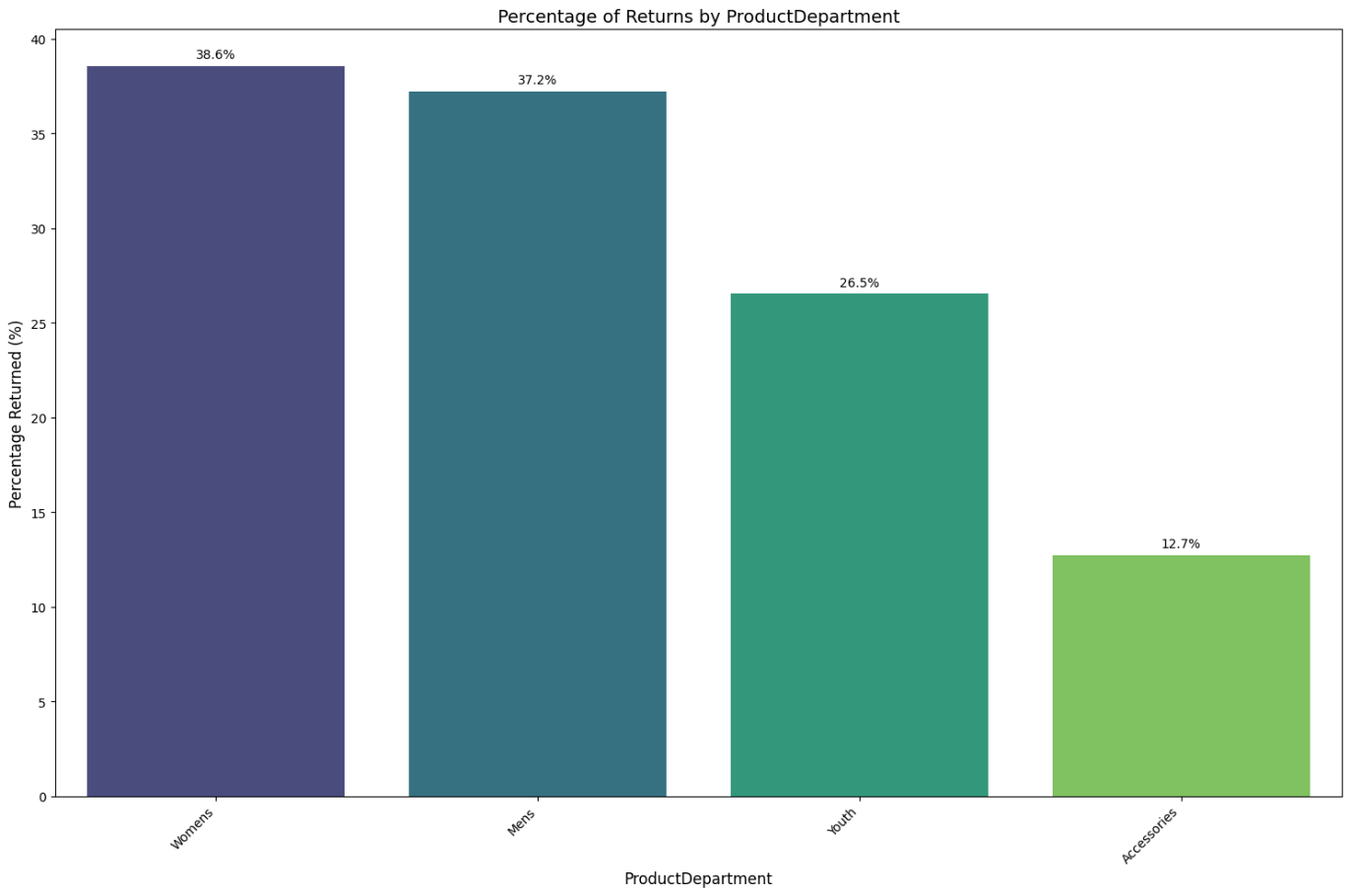
* States like Tennessee, Georgia, West Virginia, Oregon, and Texas have return percentages ranging from approximately 35% to 37%.
* A large group of states falls into the range of 30% to 35% return rates, including Nebraska, Montana, Arkansas, Florida, Alabama, and others.

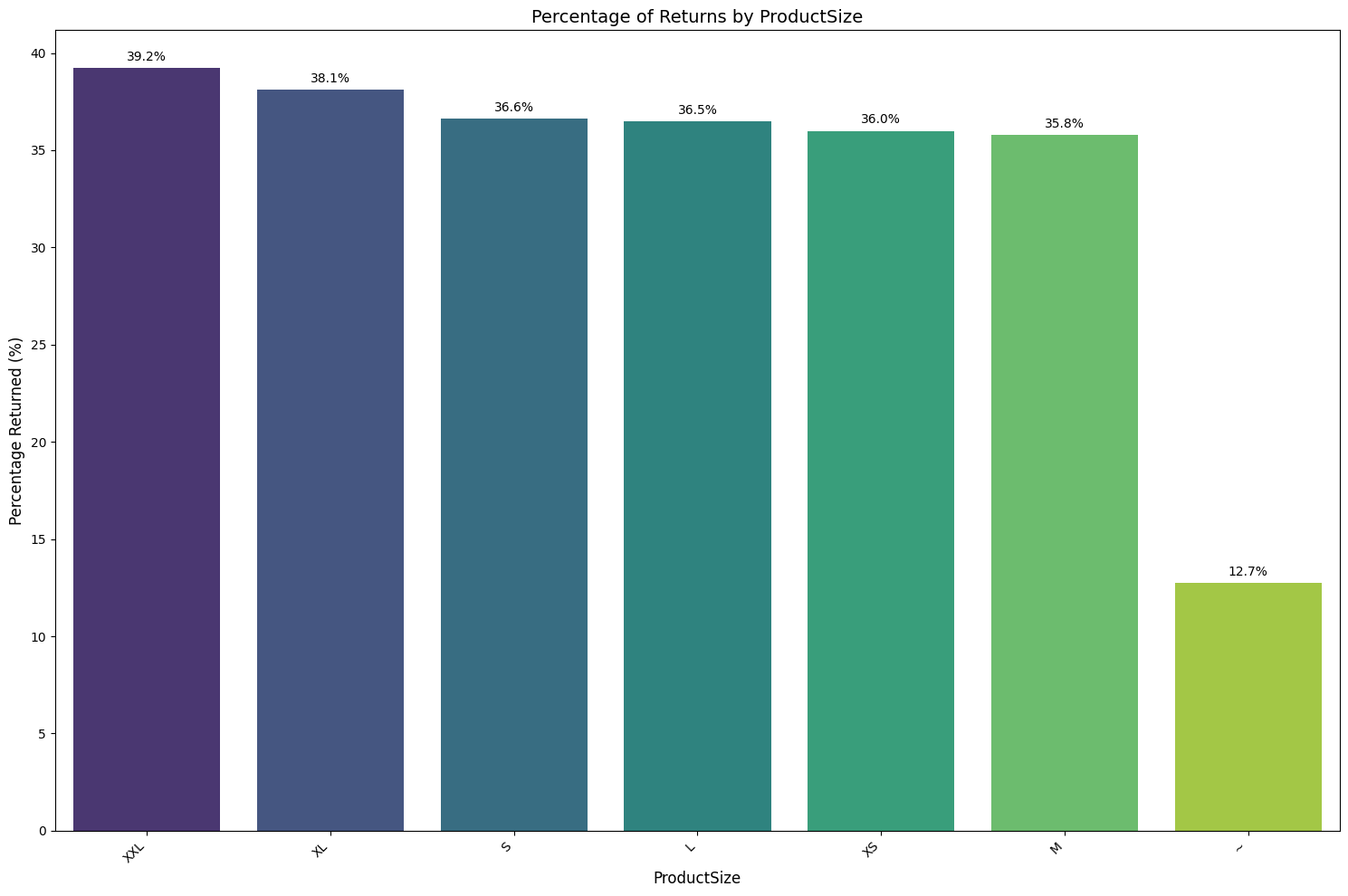
Lower Range States:

* Wyoming, Maine, and New Mexico have the lowest return percentages, with New Mexico being the lowest at around 20%.

Overall Distribution:

* The distribution shows a relatively broad range of return percentages across states.
* The highest return percentages are approaching 40%, while the lowest are around 20%.





Departments with Highest Return Rates:

* Women's department has the highest return rate at 38.6%.
* Men's department follows closely with a return rate of 37.2%.

Moderate Return Rate:

* Youth department has a return rate of 26.5%.

Lowest Return Rate:

* Accessories department has the lowest return rate at 12.7%.

Sizes with Highest Return Rates

* XXL has the highest return rate at 39.2%.
* XL follows with a return rate of 38.1%.

Moderate Return Rates:

* S, L, XS, and M all have return rates ranging from approximately 35.8% to 36.6%.

Lowest Return Rate:

* The category denoted by ~ has the lowest return rate at 12.7%.

Insights

* High Return Rates in Women's and Men's Departments: These categories might be facing issues related to product fit, quality, or customer expectations, leading to higher return rates.
* Lower Return Rate in Accessories: This could indicate higher customer satisfaction or fewer issues with product expectations in the Accessories category.
* Size-Specific Issues: Larger sizes such as XXL and XL have higher return rates, which may indicate fit or quality issues specific to these sizes.

A graph of numbers and a number of returns

Description automatically generated

Most Common Return Frequencies:

* The highest number of customers have 1 return.
* Following this, customers with 2 or 3 returns are also common.
* The frequency decreases progressively with an increasing number of returns.

Moderate Return Frequencies:

* There are a moderate number of customers with 4 to 6 returns.
* The number of customers significantly drops after 7 returns.

Rare Return Frequencies:

* Very few customers have 10 or more returns.
* The distribution tail tapers off with a maximum of 19 returns.

Insights

* Majority with Few Returns: Most customers tend to have 1 to 3 returns, which might indicate occasional dissatisfaction or specific issues with a few products.
* High Return Customers: A small number of customers have a high frequency of returns. These customers might have higher expectations or consistently face issues with the products.

**Feature Engineering**

Msrp:

* Formula: df['msrp'] = df['PurchasePrice'] \* (1 - df['DiscountPct'])
* MSRP (Manufacturer's Suggested Retail Price) is calculated by dividing the purchase price by (1 minus the discount percentage). It represents the original price of the product before any discount is applied.

RepeatReturnFlag:

repeat\_returns = df.groupby('CustomerID')['Returned'].sum()

df['RepeatReturnFlag'] = df['CustomerID'].map(repeat\_returns > 1).astype(int)

* This flag indicates whether a customer has made more than one return. It is set to 1 if the customer has returned more than one product, otherwise 0.

RecentReturnRate:

sorted\_train = df.sort\_values(by=['CustomerID', 'OrderDate'])

df['RecentReturnRate'] = sorted\_train.groupby('CustomerID', group\_keys=False)['Returned'].apply(recent\_return\_rate)

* This feature calculates the recent return rate for each customer. It is computed using a custom function recent\_return\_rate applied to the 'Returned' column, sorted by customer ID and order date.

ProductReturnRate:

product\_returns = df.groupby('ProductDepartment')['Returned'].mean()

df['ProductReturnRate'] = df['ProductDepartment'].map(product\_returns)

* This feature represents the average return rate of products within each product department. It is computed as the mean return rate grouped by product department.

PriceSensitivity:

df['PriceSensitivity'] = df['DiscountPct'] / df['msrp']

* This feature measures the customer's sensitivity to price changes. It is calculated as the discount percentage divided by the MSRP.

MultiItemOrder:

multi\_item\_orders = df.groupby('OrderID').size()

df['MultiItemOrder'] = df['OrderID'].map(multi\_item\_orders > 1).astype(int)

* This flag indicates whether an order contains multiple items. It is set to 1 if the order includes more than one item, otherwise 0.

Season:

df['Season'] = df['OrderDate'].apply(get\_season)

* This feature categorizes the order date into a season (e.g., Winter, Spring, Summer, Fall) using the get\_season function.

CustomerAge:

df['CustomerAge'] = df.apply(lambda row: calculate\_age(row['CustomerBirthDate'], row['OrderDate']), axis=1)

* This feature calculates the age of the customer at the time of order. It is computed using the customer's birth date and the order date.

Holiday:

df['Holiday'] = df['OrderDate'].isin(holidays()).astype(int)

* This flag indicates whether the order date falls on a holiday. It is set to 1 if the order date is a holiday, otherwise 0.

DaysSinceFirstOrder:

df['DaysSinceFirstOrder'] = (df['OrderDate'] - df.groupby('CustomerID')['OrderDate'].transform('min')).dt.days

* This feature calculates the number of days since the customer's first order.

CustomerLifetimeValue:

df['CustomerLifetimeValue'] = df.groupby('CustomerID')['PurchasePrice'].transform('sum')

* This feature represents the total amount spent by the customer over their lifetime.

OrderFrequency:

df.groupby('CustomerID')['CustomerID'].transform('size')

* This feature counts the total number of orders made by each customer.

DayOfWeek:

df['DayOfWeek'] = df['OrderDate'].dt.day\_name()

* This feature represents the day of the week on which the order was placed.

DaysBetweenOrders

df['DaysBetweenOrders'] = df.groupby('CustomerID')['OrderDate'].diff().dt.days.fillna(0)

* This feature calculates the number of days between consecutive orders for each customer.

AvgDaysBetweenOrders

df.groupby('CustomerID')['DaysBetweenOrders'].transform('mean')

* This feature calculates the average number of days between orders for each customer.

A graph with blue and orange bars

Description automatically generated

* Higher Return Rate in Multi-Item Orders: The higher number of returns in multi-item orders suggests that customers who purchase multiple items are more likely to return at least one item.
* Lower Returns in Single-Item Orders: Single-item orders have fewer returns, which might indicate higher satisfaction with single-item purchases or fewer issues encountered.

A graph with blue and orange squares

Description automatically generated

* Higher Return Rates Among Repeat Return Customers: Customers who have previously made returns are more likely to return products again, indicating a pattern of dissatisfaction or recurring issues.
* Lower Return Rates Among Non-Repeat Return Customers: Customers who do not typically return products maintain a higher rate of keeping their purchases, which might indicate higher satisfaction.

A graph of a graph

Description automatically generated

* Both returned and non-returned products follow a similar distribution pattern.
* The density peaks around the ProductCost range of 10 to 20 for both returned and non-returned products.
* For lower ProductCost values (below 10), the density of returned products is slightly lower than that of non-returned products.
* For ProductCost values between 10 and 30, the returned products show slightly higher density peaks compared to non-returned products, indicating a higher return rate for products in this cost range.
* Beyond a ProductCost of 30, the density of both returned and non-returned products gradually declines, with a similar pattern for both categories.

A graph of a function

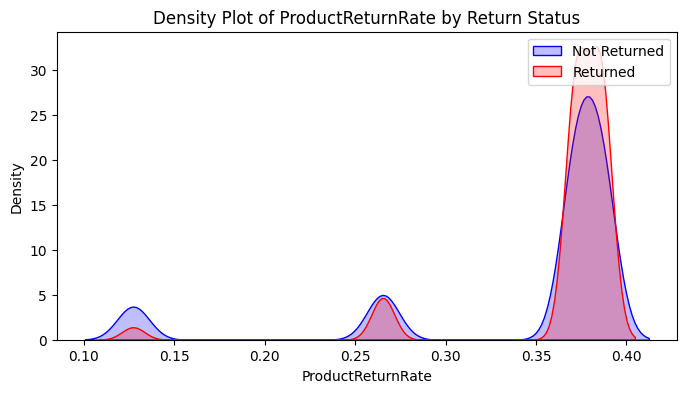
Description automatically generated with medium confidence

* Both returned and non-returned products follow a similar distribution pattern, with multiple peaks indicating the frequency of different discount percentages.
* The density peaks are observed around 0.05, 0.10, 0.15, 0.20, and 0.25 for both returned and non-returned products.
* For lower DiscountPct values (around 0 to 0.05), the density of returned products is slightly higher than that of non-returned products.
* For DiscountPct values around 0.10 and 0.15, the density of returned products is slightly higher compared to non-returned products.
* For higher DiscountPct values (around 0.25), the density of non-returned products is higher than that of returned products.

A graph of a price

Description automatically generated with medium confidence

* Both returned and non-returned products follow a similar distribution pattern, with several peaks indicating the frequency of different purchase prices.
* The density peaks are observed around 40, 60, and 80 for both returned and non-returned products.
* For lower PurchasePrice values (below 40), the density of non-returned products is higher than that of returned products.
* For PurchasePrice values between 40 and 60, the density of returned products is slightly higher compared to non-returned products.
* For higher PurchasePrice values (above 60), the density of both returned and non-returned products follows a similar pattern, with returned products showing slightly higher peaks at certain points.



* There are three prominent peaks in the density plot at approximately 0.125, 0.25, and 0.375.
* Both returned and non-returned products follow a similar distribution pattern, with slightly higher densities for returned products at each peak.
* At the first peak around 0.125, the density of non-returned products is slightly higher than that of returned products.
* At the second peak around 0.25, the densities of returned and non-returned products are similar, with returned products having a slightly higher peak.
* At the third peak around 0.375, the density of returned products is higher than that of non-returned products.

A graph of a graph

Description automatically generated with medium confidence

* There are distinct peaks in the density plot at different RecentReturnRate values.
* The density for non-returned products is highest near 0, indicating that most non-returned products have a low recent return rate.
* The density for returned products is more spread out, with notable peaks at higher RecentReturnRate values.

A screenshot of a computer

Description automatically generated

Highly Significant Results (p-value < 0.001):

* RepeatReturnFlag: Chi-square test shows a very strong association.
* ProductReturnRate, RecentReturnRate: T-tests indicate significant differences in means.
* ProductDepartment: Chi-square test reveals significant association with categories.
* ProductSize: Significant categorical differences detected by Chi-square test.
* PurchasePrice, msrp, PriceSensitivity, ProductCost: T-tests show significant mean differences.
* MultiItemOrder: Chi-square test indicates significant categorical associations.
* CustomerAge, DayOfWeek, AvgPurchaseCost, Season: Both continuous and categorical variables show significant differences or associations.

Moderately Significant Results (p-value < 0.05 but ≥ 0.001):

* AvgDaysBetweenOrders, CustomerLifetimeValue, DaysBetweenOrders: T-tests suggest moderate differences in means.
* CustomerState, DiscountPct: Chi-square and T-test indicate some level of association or difference.

Non-Significant Results (p-value ≥ 0.05):

* Holiday: Chi-square test suggests no significant association.
* DaysSinceFirstOrder: T-test shows no significant difference in means.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Model Fit Statistics:

* Dep. Variable: Returned
* No. Observations: 64,732
* Model: Logit
* Method: MLE (Maximum Likelihood Estimation)
* Log-Likelihood: -29,961
* LL-Null: -42,022
* LLR p-value: 0.000
* Pseudo R-squared: 0.287
* Converged: Yes
* Iterations: 7

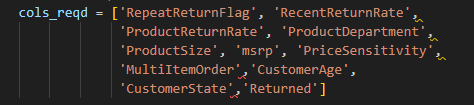
Significant Predictors (p < 0.05):

* CustomerState: Only a few states like Indiana, Washington, Wisconsin show significant effects.
* DayOfWeek: Monday, Sunday, Tuesday, Wednesday are significant, indicating specific days of the week may affect the likelihood of a product being returned.
* MultiItemOrder: Positive coefficient (0.7576), suggesting that orders with multiple items are more likely to be returned.
* RepeatReturnFlag: Positive coefficient (0.3037), indicating that customers who have returned items before are more likely to return again.
* AvgPurchaseCost: Negative coefficient (-0.0073), indicating that higher average purchase costs are associated with a lower likelihood of return.
* RecentReturnRate: Very strong positive coefficient (5.7339), showing a high likelihood of return as the recent return rate increases.
* DaysSinceFirstOrder: Slight positive effect (0.00009557), suggesting a gradual increase in the likelihood of return over time since the first order.

Notable Non-Significant Predictors (p >= 0.05):

* Season: Spring, Summer, Winter show no significant effect compared to Fall.
* ProductDepartment: No significant difference between departments like Mens, Womens, and Youth.
* Holiday: No significant effect, indicating holidays do not significantly alter the return behavior.
* PriceSensitivity: Despite a substantial coefficient, it's not statistically significant.
* CustomerLifetimeValue, DiscountPct, PurchasePrice, msrp: These financial metrics show no significant impact on the likelihood of a product being returned under the conditions of this model.

From the above analysis, the features to prioritize for your model include:



Model Fitting and Evaluation

Steps and Processes:

Encoding and Scaling:

* The script successfully completes one-hot encoding of categorical variables.
* An existing scaler is loaded to standardize the feature values, followed by the completion of the scaling process.

Model Evaluation using Cross-Validation:

* The models evaluated include Random Forest, Gradient Boosting, Extra Trees, XGBoost, and LightGBM.
* Each model is evaluated using a 5-fold cross-validation method, focusing on metrics such as ROC AUC, accuracy, precision, recall, and F1 score.

Detailed Training Logs for LightGBM:

* Multiple entries related to LightGBM training show its automated decision-making regarding data splitting (row-wise vs. column-wise), alongside continuous monitoring and adjustments of internal parameters like initscore based on the proportion of positive to negative samples.

Selection of the Best Model:

* Based on the average ROC AUC score across the models, Gradient Boosting is determined to be the best, achieving a score slightly higher than LightGBM.

Hyperparameter Tuning:

* Extensive hyperparameter tuning is performed for the Gradient Boosting model using RandomizedSearchCV.
* Ultimately, the best parameters for the Gradient Boosting model are found, including learning rate, tree depth, the minimum number of samples per leaf and split, the number of estimators, and the subsample ratio.

Model Saving and Reporting:

* Metrics for the best model configuration are saved.
* The model itself is saved to the specified path

Key Outcomes

Performance Metrics for Models:

* Random Forest: ROC AUC = 0.820, Accuracy = 0.748, Precision = 0.672, Recall = 0.554, F1 = 0.608
* Gradient Boosting: ROC AUC = 0.837, Accuracy = 0.759, Precision = 0.689, Recall = 0.575, F1 = 0.627
* Extra Trees: ROC AUC = 0.799, Accuracy = 0.734, Precision = 0.644, Recall = 0.550, F1 = 0.592
* XGBoost: ROC AUC = 0.832, Accuracy = 0.757, Precision = 0.691, Recall = 0.560, F1 = 0.618
* LightGBM: ROC AUC = 0.836, Accuracy = 0.759, Precision = 0.695, Recall = 0.563, F1 = 0.622