Project Report

Data Encoding and Feature Engineering

Prepared by:

PRIYAM PAL

DATA ENCODING

ENCODING - 1: Applying Target Encoding of aisle, department, product_name

What is Target Encoding:

Target encoding replaces categorical values with the mean of the target variable (e.g., reordered) for each category.

Features Being Encoded:

The features **aisle**, **department**, and **product_name** are encoded using the mean of the reordered column.

Why This Encoding is Needed:

It captures the relationship between categorical features and the target variable, improving model performance for numerical algorithms like Logistic Regression.

aisle_target_enc	department_target_enc	product_name_target_enc
0.547057	0.574195	0.585799
0.590163	0.574195	0.629367
0.671338	0.628133	0.632911
0.46946	0.346935	0.4375
0.493279	0.541577	0.509413
0.671338	0.628133	0.47093
0.706434	0.669969	0.736155
0.718275	0.649793	0.777074
0.718275	0.649793	0.843169
0.68665	0.669969	0.681264

ENCODING - 2: Applying One-hot Encoding on day_of_week

What is One-Hot Encoding:

One-hot encoding converts categorical values into binary columns, where each unique category becomes a separate column with values as 0 or 1.

Features Being Encoded:

The feature **order_dow** (day of the week) is encoded into new binary columns: **dow_0**, **dow_1**, …, **dow_6**.

Why This Encoding is Needed:

It allows the model to treat each category as independent and avoids assuming any ordinal relationship, which is essential for algorithms that require numerical input.

dow_0	dow_1	dow_2	dow_3	dow_4	dow_5	dow_6
0	1	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	1	0
0	0	1	0	0	0	0
1	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	0	0	1

ENCODING - 3: Applying Cyclic Encoding and Performing Sin-Cosine Transformation on order_hour_of_day

What is Cyclic Encoding:

Cyclic encoding transforms cyclic features (e.g., hours of the day) into sine and cosine components to preserve their cyclical nature.

Features Being Encoded:

The feature **order_hour_of_day** is encoded into two new columns: **order_hour_sin** and **order_hour_cos**.

Why This Encoding is Needed:

It helps the model understand the cyclical relationship (e.g., 23:00 is close to 00:00), which regular numerical encoding cannot capture.

order_hour_sin	order_hour_cos
2.58819E-01	-0.965926
8.66025E-01	-0.5
7.07107E-01	-0.707107
1.22465E-16	-1
-9.65926E-01	-0.258819
1.22465E-16	-1
7.07107E-01	-0.707107
-7.07107E-01	0.707107
-8.66025E-01	-0.5
5.00000E-01	-0.866025

ENCODING - 4: Applying Ordinal Encoding on add_to_card_order

What is Ordinal Encoding:

Ordinal encoding converts categorical or ordered data into integers while preserving their rank or order.

Features Being Encoded:

The feature **add_to_cart_order** is encoded into a new column **add_to_cart_order_encoded** with integer values representing its order.

Why This Encoding is Needed:

It converts ordered data into a numerical format suitable for machine learning models while retaining the inherent order information.

order_id	product_id	add_to_cart_order	reordered	add_to_cart_order_encoded
2722718	8619	9	0	8
2089674	13870	10	0	9
3024155	2029	1	0	0
2890872	16062	2	0	1
1798802	14335	3	0	2
1688514	40556	24	1	23
430688	14999	2	1	1
2022104	21137	5	1	4
456648	24852	1	1	0
579936	30442	2	1	1

ENCODING - 5: Apply Binning on day_since_prior_order

What is Binning:

Binning categorizes continuous values into discrete intervals (or bins), assigning each value to a specific category.

Features Being Encoded:

The feature **days_since_prior_order** is binned into categories: 0-7, 8-15, 16-23, 24-31, and Unknown.

Why This Encoding is Needed:

It simplifies continuous data, making it easier to interpret patterns and relationships, especially for models or analysis requiring categorical inputs.

order_id	product_id	days_since_prior_order_binned
2722718	8619	8-15
2089674	13870	24-31
3024155	2029	0-7
2890872	16062	0-7
1798802	14335	Unknown
1688514	40556	0-7
430688	14999	24-31
2022104	21137	0-7
456648	24852	0-7
579936	30442	8-15

FEATURE ENGINEERING

Feature I -- **average_days_between_purchases**: Average time between purchases by each product by each user. This gives insights into the typical frequency of repurchases of a specific product

Definition of the Feature:

average_days_between_purchases represents the average time interval (in days) between consecutive purchases of a specific product by a user.

Role of the Feature:

It helps identify user buying behavior and product repurchase frequency, which is crucial for predicting reorder patterns and understanding product popularity.

order_id	product_id	add_to_cart_order_encoded	days_since_prior_order_binned	average_days_between_purchases
2722718	8619	8	8-15	15
2089674	13870	9	24-31	29
3024155	2029	0	0-7	4
2890872	16062	1	0-7	2
1798802	14335	2	Unknown	7.166667
1688514	40556	23	0-7	5.333333
430688	14999	1	24-31	11
2022104	21137	4	0-7	9.666667
456648	24852	0	0-7	6.375
579936	30442	1	8-15	21

Feature II -- **Product_purchase_frequency**: Count the total number of times each product has been purchased by a particular user.

Definition of the Feature:

product_purchase_frequency represents the total count of how many times a specific product has been purchased by a user.

Role of the Feature:

It highlights user preferences for specific products, aiding in understanding product loyalty and predicting future purchases.

order_id	product_id	product_purchase_frequency	total_purchases
2722718	8619	1	1
2089674	13870	1	1
3024155	2029	1	1
2890872	16062	1	1
1798802	14335	6	6
1688514	40556	3	3
430688	14999	4	4
2022104	21137	3	3
456648	24852	8	8
579936	30442	2	2

Feature III -- **total_purchases**

Definition of the Feature:

total_purchases represents the total number of purchases made for each user-product combination.

Role of the Feature:

It tracks the overall purchase behavior for a user-product pair, providing insights into user engagement with specific products.

Feature IV – **interval_std_dev**

Definition of the Feature:

interval_std_dev represents the standard deviation of the time intervals between consecutive purchases of a product by a user.

Role of the Feature:

It measures the variability in purchase frequency, helping identify inconsistent purchasing patterns or seasonal buying behavior.

order_id	product_id	product_purchase_frequency	total_purchases	interval_std_dev
2722718	8619	1	1	NaN
2089674	13870	1	1	NaN
3024155	2029	1	1	NaN
2890872	16062	1	1	NaN
1798802	14335	6	6	4.445972
1688514	40556	3	3	1.527525
430688	14999	4	4	11.518102
2022104	21137	3	3	4.932883
456648	24852	8	8	3.159453
579936	30442	2	2	14.142136

Feature V – **Product_reorder_rate:** Reorder rate for each product by dividing the number of times a product has been reordered by the total number of orders of product

Definition of the Feature:

product_reorder_rate is calculated by dividing the number of times a product has been reordered by its total number of orders.

Role of the Feature:

It measures how frequently a product is reordered, providing insights into product popularity and customer retention tendencies.

order_id	product_id	total_purchases	interval_std_dev	product_reorder_rate
2722718	8619	1	NaN	0.585799
2089674	13870	1	NaN	0.629367
3024155	2029	1	NaN	0.632911
2890872	16062	1	NaN	0.4375
1798802	14335	6	4.445972	0.509413
1688514	40556	3	1.527525	0.47093
430688	14999	4	11.518102	0.736155
2022104	21137	3	4.932883	0.777074
456648	24852	8	3.159453	0.843169
579936	30442	2	14.142136	0.681264

Feature VI – **Users_general_reorder_rate:** The ratio of reordered items to total items of each user which captures the user general tendency to reorder products.

Definition of the Feature:

users_general_reorder_rate is the ratio of items reordered to the total items purchased by a user, reflecting their overall tendency to reorder

Role of the Feature:

It helps in understanding user loyalty and predicting the likelihood of users reordering products in future purchases.

order_id	product_id	product_reorder_rate	users_general_reorder_rate
2722718	8619	0.585799	0.818182
2089674	13870	0.629367	0.47619
3024155	2029	0.632911	0.176471
2890872	16062	0.4375	0.27907
1798802	14335	0.509413	0.7125
1688514	40556	0.47093	0.78961
430688	14999	0.736155	0.675325
2022104	21137	0.777074	0.62037
456648	24852	0.843169	0.684932
579936	30442	0.681264	0.222222

Feature VII – **Avg_add_to_cart_order:** Avg. position of each product in the cart when it is purchased.

Definition of the Feature:

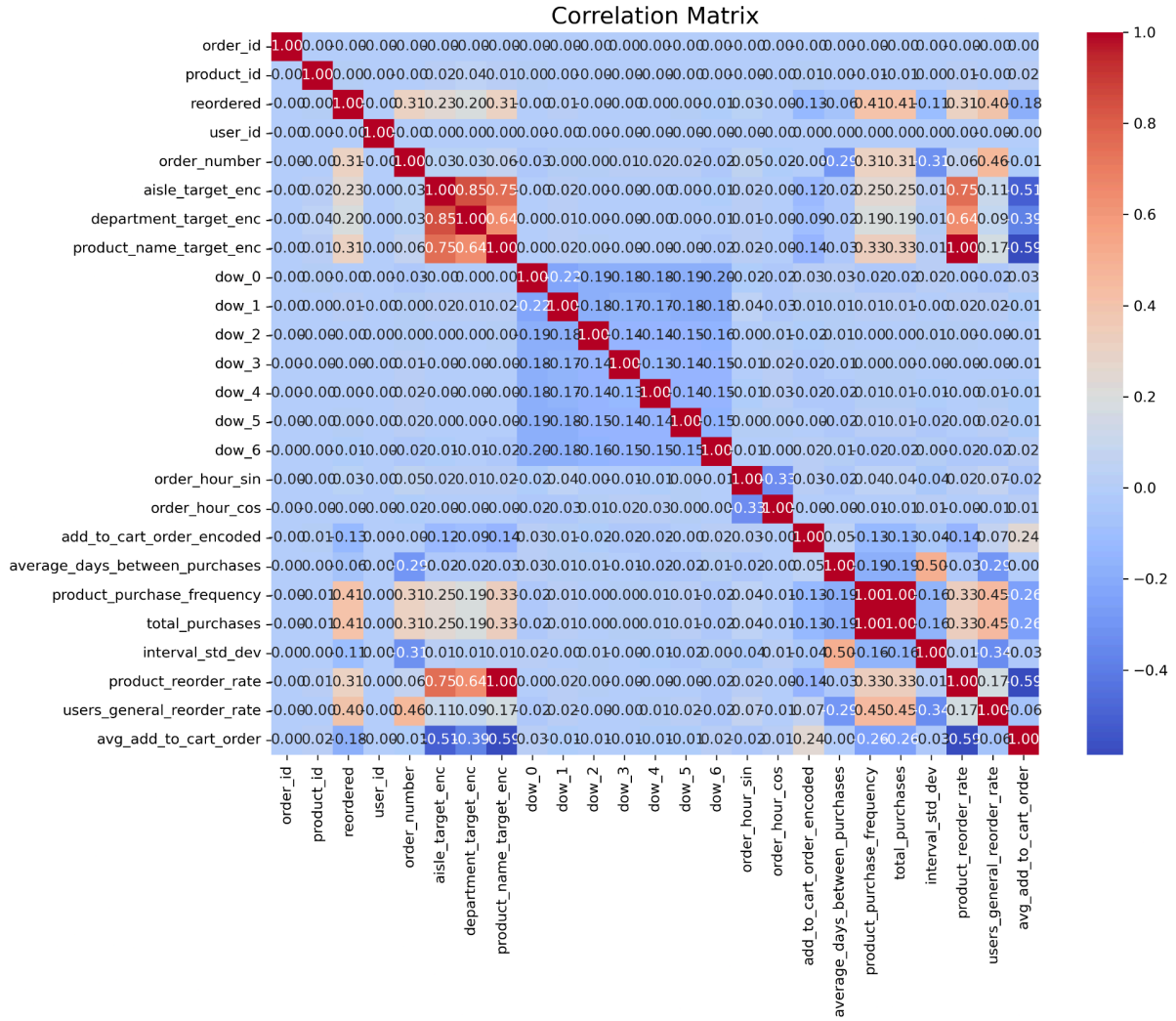
avg_add_to_cart_order represents the average position of a product in the shopping cart across all its purchases.

Role of the Feature:

It provides insights into user preferences, indicating whether a product is typically added early or late in the shopping process.

order_id	product_id	users_general_reorder_rate	avg_add_to_cart_order
2722718	8619	0.818182	10.91716
2089674	13870	0.47619	9.489091
3024155	2029	0.176471	9.468354
2890872	16062	0.27907	13.520833
1798802	14335	0.7125	8.58804
1688514	40556	0.78961	8.473837
430688	14999	0.675325	6.785266
2022104	21137	0.62037	7.251493
456648	24852	0.684932	4.889006
579936	30442	0.222222	7.992325

Correlation Matrix - HeatMap



Insights:

Strongly Correlated Features:

- aisle_target_enc, department_target_enc, and product_name_target_enc are highly correlated with product_reorder_rate, making them key predictors of reorder behavior.
- product_purchase_frequency is strongly correlated with total_purchases, indicating both represent user-product purchase activity.

Negative Correlations:

- avg_add_to_cart_order has a negative correlation with product_reorder_rate and users_general_reorder_rate, suggesting that earlier positions in the cart are linked to higher reorder likelihood.

Weak Correlations:

- Day-of-week features (dow_0, dow_1, etc.) show minimal correlation with other features, implying limited influence on purchase and reorder patterns.

Independent Features:

- Identifier columns like order_id, product_id, and user_id are uncorrelated with other features, as they serve only as unique identifiers.

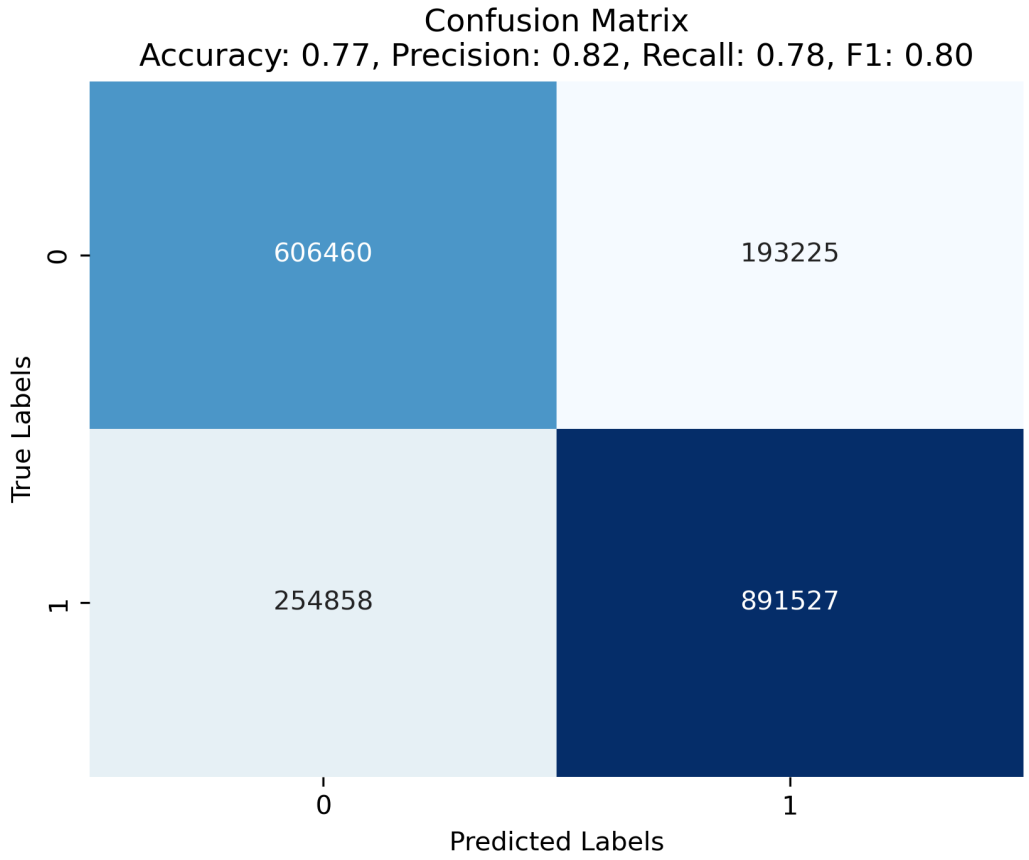
Temporal Insights:

- order_hour_sin and order_hour_cos have weak correlations with reorder metrics, indicating time-of-day has a slight influence on reorder behavior.

Reorder Behavior:

- Features like users_general_reorder_rate and product_reorder_rate show significant relationships with metrics like frequency and interval, making them central to understanding reorder tendencies.

Confusion Matrix



- Accuracy: 0.77 - Indicates that 77% of the predictions were correct.
- Precision (Positive Predictive Value): 0.82 - Out of all predicted positive instances, 82% were actual positives.
- Recall (Sensitivity/True Positive Rate): 0.78 - Out of all actual positives, 78% were correctly identified as positive.
- F1 Score: 0.80 - Harmonic mean of precision and recall. This shows a balance between precision and recall, valued at 0.80.

Classification Matrix:

Top Left (606,460):

True Negatives (TN): The number of cases where the actual class was 0, and the model correctly predicted 0.

Top Right (193,225):

False Positives (FP): The number of cases where the actual class was 0, but the model incorrectly predicted 1.

Bottom Left (254,858):

False Negatives (FN): The number of cases where the actual class was 1, but the model incorrectly predicted 0.

Bottom Right (891,527):

True Positives (TP): The number of cases where the actual class was 1, and the model correctly predicted 1.

Conclusion of Feature Engineering

Feature engineering and encoding have played a significant role in enhancing the model's performance. The most important features, such as *total_purchases* and *product_purchase_frequency*, contributed notably to improving accuracy. The model achieved an accuracy of **0.77**, a precision of **0.82**, a recall of **0.78**, and an F1 score of **0.80**, indicating a solid balance between precision and recall.

The classification matrix reveals that the model correctly predicted 606,460 true negatives and 891,527 true positives. However, it also incorrectly predicted 193,225 false positives and missed 254,858 false negatives.