ML_FINAL_PROJECT DOCUMENTATION

COURSE:DS230
STUDENT:ESRAA BANI SALMAN
DATE:JANUARY 2026

PROJECT INTRODUCTION

In this project, we used the Instacart dataset to analyze customer shopping behavior and build predictive machine learning models. The dataset contains information about users, orders, and products, which allows us to study purchasing patterns and reorder behavior. In this report, we focus on Task A, which is a classification problem that predicts whether a product will be reordered.

Importing Required Libraries

In this step, we imported the main Python libraries needed for data analysis, visualization, preprocessing, and building classification models. We also imported specific tools from scikit-learn to support the classification task, such as train-test split, scaling, PCA, different classification models, and evaluation metrics.

LOADING AND MERGING THE DATASET

After importing the libraries, we loaded the dataset files and merged them into one dataframe for analysis and modeling.

We used common IDs (order_id, product_id, aisle_id, department_id) to combine orders, products, aisles, and departments into one dataset.

EXPLORATORY DATA ANALYSIS (EDA)

We performed exploratory data analysis (EDA) to understand the dataset structure, check data quality, and identify useful patterns before training the classification models. The EDA included checking missing values and duplicates, analyzing the target distribution, and exploring user ordering behavior over time and across product categories.

- ***Checked the dataset shape and previewed the first rows***

We displayed the dataset size and used head() to understand what columns we have and how the data looks.

- **Explored column information and data types**

We used info() and dtypes to see each column type (numeric or text) and confirm the dataset structure.

- **Checked missing values (NaN)**

We used isnull().sum() to make sure there are no missing values in any column.

- **Checked duplicated rows**
  We used duplicated().sum() to verify that there are no duplicate records in the dataset.
- **Analyzed the target variable distribution(reordered)**

We used value_counts() to see how many products were reordered vs not reordered.

- **Explored most common aisles and departments**

We checked aisle and department value counts to understand which categories customers buy most from.

- **Explored ordering behavior by day of the week(order_dow)**

We checked the most common days when users place orders.

- **Explored ordering behavior by hour (order_hour_of_day)**

We checked the most common hours during the day when orders happen.

- **Reviewed numeric statistics using describe()**
  We summarized numeric features (mean, min, max, quartiles) to understand their distribution.

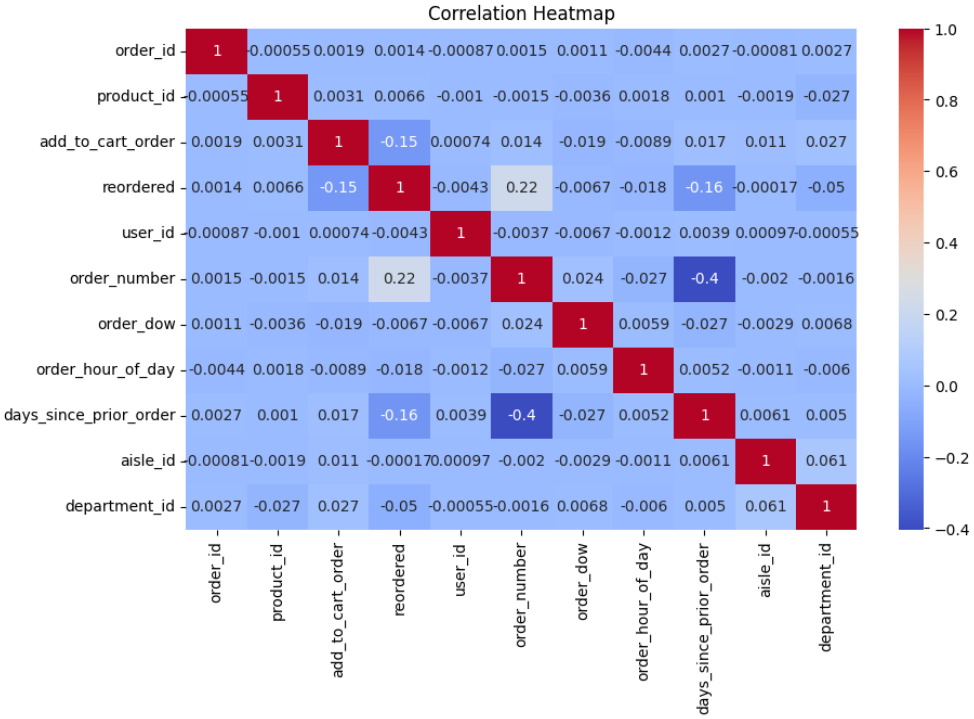
EDA Visualizations (Plots)

Then, we created several visualizations to support our exploratory data analysis (EDA). These plots helped us better understand the dataset, detect patterns in user behavior, and identify useful features for the classification task. The following figures summarize the main insights we observed:

Figure 1

Chart Type: Heatmap.

Title: Correlation Heatmap.



Result / Observation:

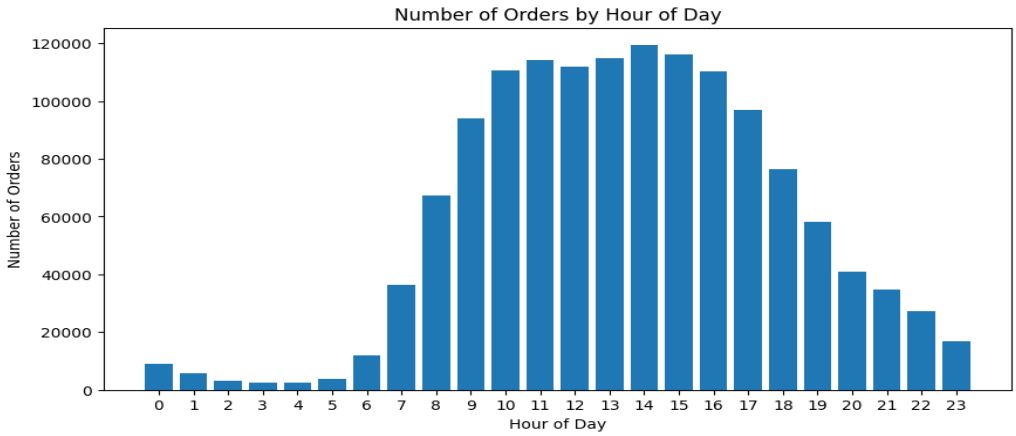
Most correlations are weak, which suggests that the features are mostly independent.

This means we can use these features without strong multicollinearity issues.

Figure 2

Chart Type: Bar Chart

Title: Number Of Order By Hour Of Day



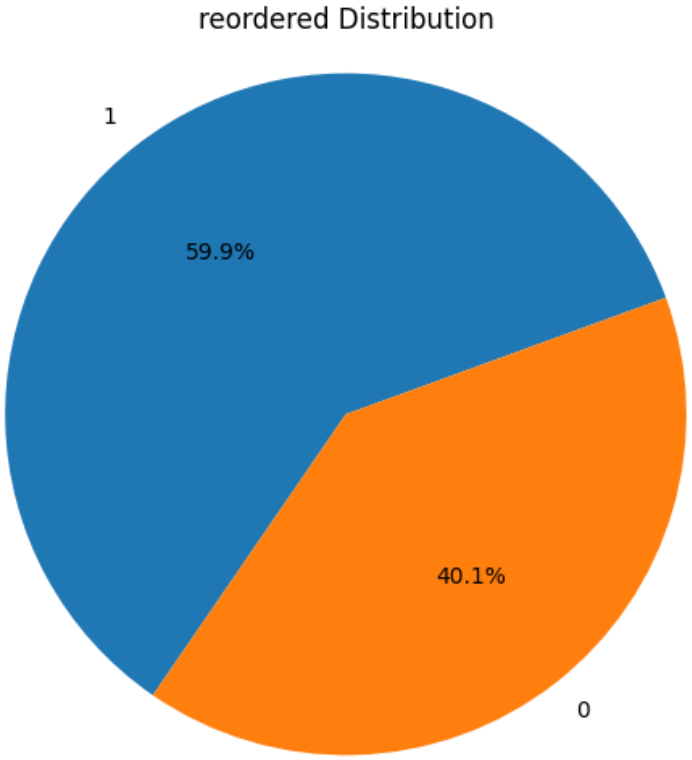
Result / Observation:

Orders increase during the day and peak between 10 AM and 4 PM, indicating that most users place orders during daytime hours.

Figure 3

Chart Type: Pie Chart

Title: Reordered Distribution



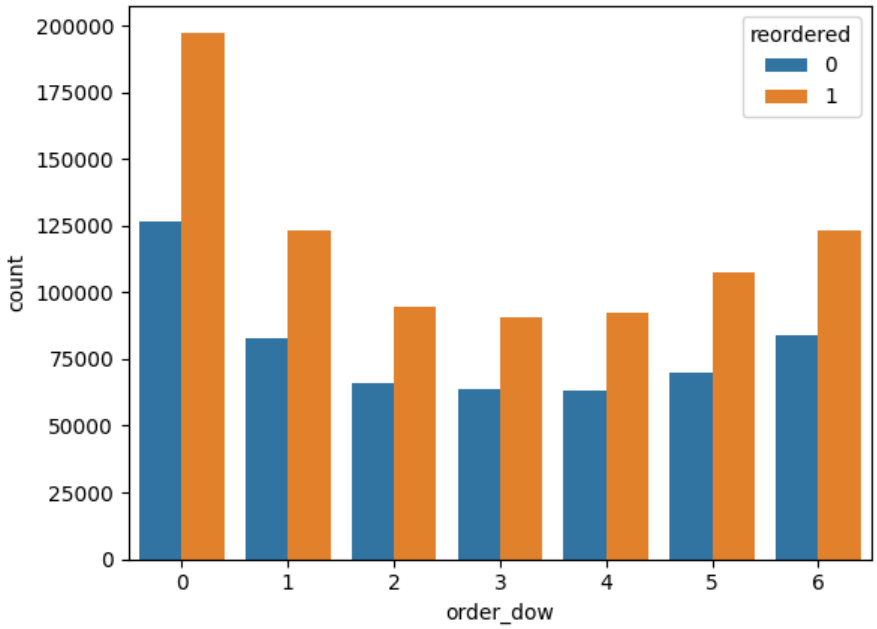
Result / Observation:

The dataset shows that 59.9% of products were reordered (label 1), while 40.1% were not reordered (label 0).

Figure 4

Chart Type: Count Plot / Bar Chart (Grouped)

Title: Orders by Day of Week (order_dow) with Reorder Status**.**



The highest orders occur on day 0 (~198K reordered vs ~127K not reordered).

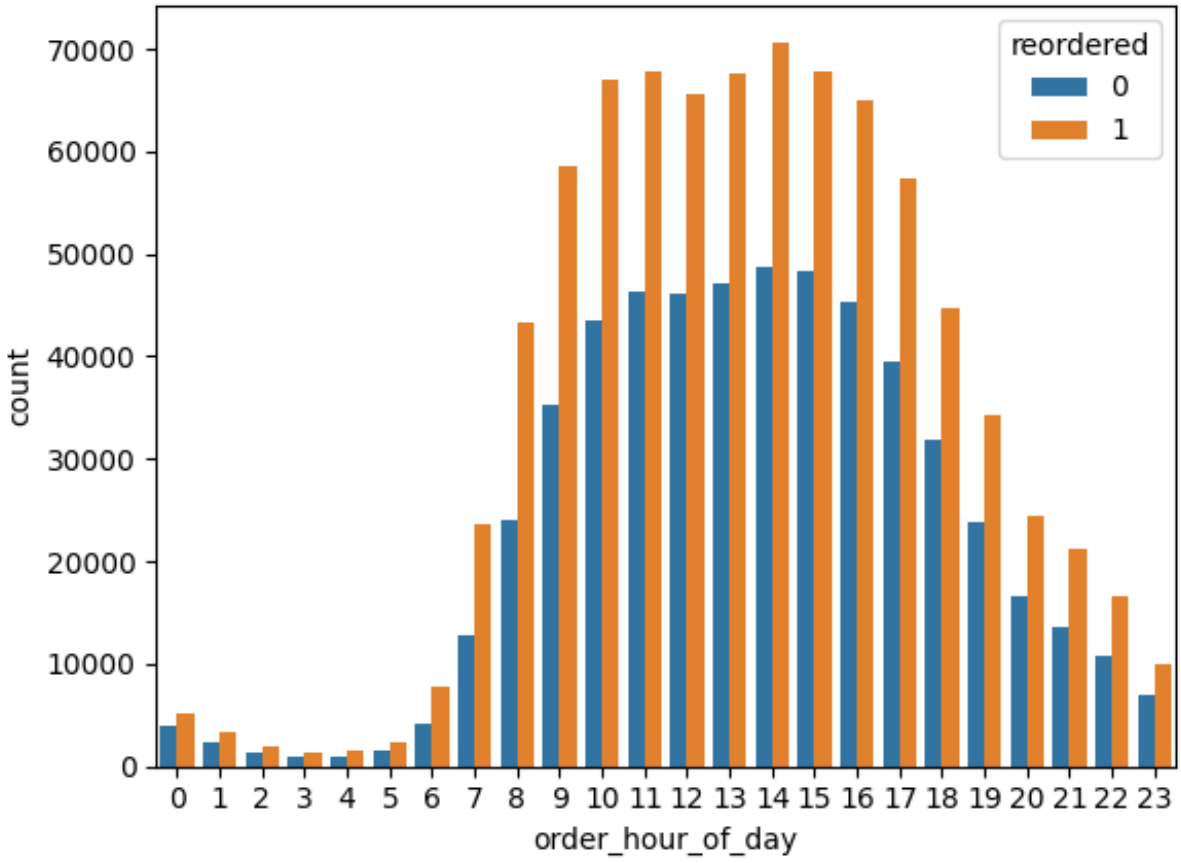
Orders stay relatively high on day 1 & day 6 (~124K reordered each).

Overall, reordered (1) is higher than not reordered (0) across all days.

Figure 5

Chart Type: Count Plot / Bar Chart (Grouped)

Title: Orders by Hour of Day (order_hour_of_day) with Reorder Status



Result / Observation (Short + Numbers)

The highest ordering activity happens between 10 AM and 4 PM.

Peak hours are around 2 PM – 3 PM, where:

Reordered (1) reaches about 70,000 orders

Not Reordered (0) reaches about 48,000–49,000 orders

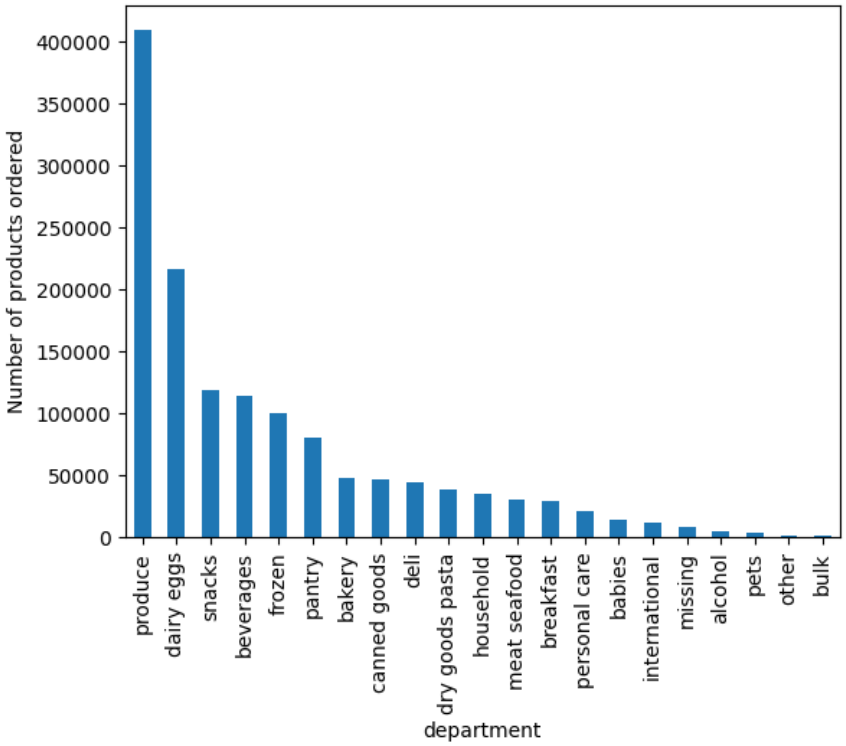
During late night and early morning (0–6 AM), orders are very low (mostly below 5,000–10,000).

Reordered (1) is consistently higher than Not Reordered (0) across almost all hours.

Figure 6

Chart Type: Bar Chart

Title: Most Ordered Departments



Result / Observation:

Produce is the most ordered department with ≈ 410,000 orders.

Dairy & Eggs comes second with ≈ 220,000 orders.

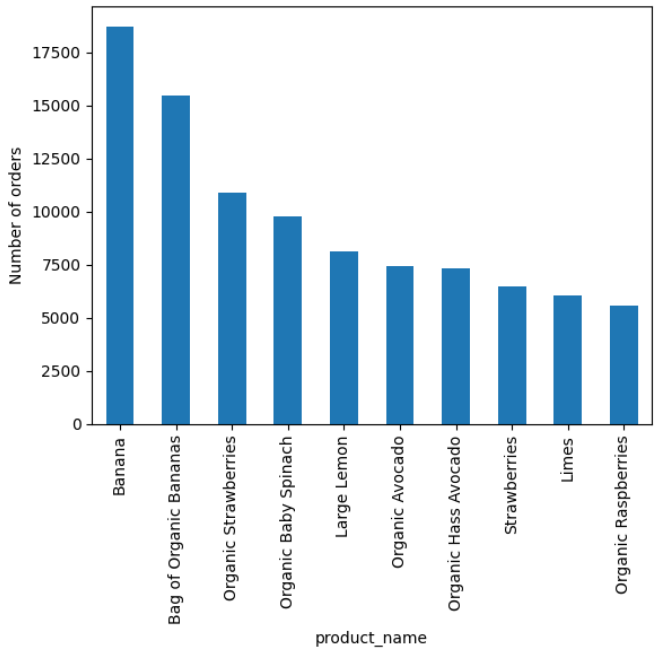
The next most ordered departments are Snacks, Beverages, and Frozen, each with around ≈ 100,000 – 120,000 orders.

Departments like Pets, Bulk, and Other have the lowest number of orders.

Figure 7

Chart Type: Bar Chart

Title: Top 10 Most Ordered Products



Result / Observation:

Banana is the most ordered product with ≈ 18,726 orders.

Bag of Organic Bananas comes second with ≈ 15,480 orders.

Organic Strawberries ranks third with ≈ 10,894 orders.

Other popular products include Organic Baby Spinach (≈ 9,784) and Large Lemon (≈ 8,135).

Overall, the top ordered products are mostly fresh fruits and organic items, showing a strong preference for healthy categories.

Feature Engineering

After completing the EDA, we performed feature engineering to create new features that help the model understand user behavior better. These features summarize each user's ordering habits and provide additional information that can improve the classification results.

user_total_orders

We calculated the total number of orders made by each user using the maximum value of order_number.

Purpose: to represent how active each user is.

user_reorder_rate

We calculated the average value of reordered for each user to measure how often they reorder products.

Purpose: to capture the user's tendency to reorder.

user_avg_days_between_orders

We calculated the average number of days between orders for each user using days_since_prior_order.

Purpose: to represent how frequently a user shops.

- These user-level features helped improve the dataset by adding behavioral patterns that support the classification model.

DATA PREPROCESSING AND PREPARING THE MODEL INPUT

After feature engineering, we prepared the dataset for the classification models. We selected the target variable (reordered) and defined the input features (X). Since machine learning models cannot work with text categories, we applied encoding to convert categorical features into numeric form.

Defined the target variable (y):

We used reordered as the target label, where 1 means reordered and 0 means not reordered.

Selected the input features (X):

We selected relevant columns such as order information, user-level features, and product category features.

Applied one-hot encoding:

We used one-hot encoding to convert categorical columns like aisle and department into numeric dummy variables.

Final check:

We reviewed the final dataset shape to confirm that all features are numeric and ready for training.

MODEL TRAINING AND EVALUATION (Training the Models)

After preparing the dataset and converting all features into numeric form, we moved to the machine learning stage. In this step, we split the data into training and testing sets, trained multiple classification models, and evaluated their performance using different evaluation metrics to compare results and select the best model.

Data Splitting (Train/Test Split)

To ensure fair evaluation, we divided the dataset into two parts

Training Set: used to train the models.

Testing Set: used to test the model on unseen data.

We used train_test_split from sklearn with a common split ratio:

Training 80%

Testing 20%

This helps us measure how well the models generalize and avoid overfitting.

Training Classification Models (Report Section)

In this step, we trained several classification models to predict whether a product will be reordered (reordered = 1) or not (reordered = 0).

We used multiple models to compare their performance and choose the best one

The models used were:

Logistic Regression

Support Vector Machine (SVM)

Decision Tree Classifier

Random Forest Classifier

To ensure a fair comparison, all models were trained using the same training dataset (X_train, y_train) and evaluated using the same testing dataset (X_test, y_test)

MODEL	Accuracy	F1(Class 1)
Logistic Regression	0.65	0.69
Logistic Regression+PCA	0.65	0.68
Decision Tree	0.61	0.67
Random Forest	0.66	0.70

Model Evaluation & Final Conclusion

After training and evaluating multiple classification models (Logistic Regression, Logistic Regression + PCA, Decision Tree, and Random Forest), we compared their performance using Accuracy and F1-score (Class 1).

The results showed that Random Forest achieved the best overall performance with the highest accuracy (0.66) and the highest F1-score for class 1 (0.70). This indicates that Random Forest was more effective in identifying reordered products compared .to the other models

Therefore, Random Forest was selected as the best model for this classification task.