Business Request

Data Science Team

Business Context

- You are a Data Scientist at a growing e-commerce company. Your manager is concerned about **high return rates** — they affect profit margins and logistic costs. You've been asked to develop a **machine learning model** that predicts whether a product will be returned, based on features known at purchase time.
- Before jumping into deep learning, your manager expects:
- A **thorough exploration** of the dataset to understand the return patterns
- A solid **baseline model**
- A progressively more **complex neural network** with the right **regularization** techniques to improve generalization
- Jupyter notebook summarizing your approach, findings, and model performance.

Dataset

- You will use a **synthetic e-commerce dataset** containing features like:
- Customer ID, Purchase Amount, Product Category
- Shipping Method, Delivery Time, Customer Review Score
- Product Size, Discount Applied, Order Date
- Return Status (target variable)

Field Name	Type	Description
Customer_ID	Categorical	Unique identifier for each customer. Useful for analyzing repeat behavior (e.g., return patterns).
Purchase_Amount	Numeric	Value (in dollars) of the transaction. High-value vs. low-value return behavior.
Product_Category	Categorical	Product type: Electronics, Clothing, Home, Beauty, Sports. Assess category-wise return risks.
Shipping_Method	Categorical	Options: Standard, Express, Two-Day, Overnight. Analyze impact of delivery type.
Delivery_Time_Days	Numeric	Days it took to deliver. Time-sensitive satisfaction factor.
Customer_Review_Score	Numeric	Rating given by customer (1.0 to 5.0). Proxy for satisfaction.
Product_Size	Categorical	Size: Small, Medium, Large. Important for clothing and large items.
Discount_Applied	Binary	1 = Yes, 0 = No. Promo items might have higher returns.
Order_Date	Date	Date of purchase. Explore seasonality or sales periods.
Return_Status	Binary	Target variable. 1 = Returned, 0 = Not Returned.

EDA: Understand the Problem

- **Tasks**: Simulate a "morning task" to analyze what might be driving returns.
  - Load the dataset and provide summary statistics
  - Visualize class imbalance (Returned vs Not Returned)
  - Plot correlations or feature importance heatmaps
  - Identify features with missing data and propose fixes
  - Identify suspicious patterns or business insights (e.g., return rate by category)
- **Deliverable**:
  - Write-up with visuals and key insights.

Baseline Model

- **Tasks**: Build a **Multilayer Perceptron (MLP)** to predict returns.
  - Normalize features and split into train/val/test sets
  - Create a simple model with 1–2 dense layers
  - Use binary cross-entropy and accuracy
  - Plot training and validation loss curves
- **Deliverable**:
  - Baseline performance metrics
  - Training curve visualization

Overfitting & Regularization

- **Tasks**: Use class techniques to improve generalization.
  - Apply:
    - **L2 Regularization**
    - **Dropout Layers**
    - **EarlyStopping**
  - Compare new performance to baseline
  - **Deliverable**:
    - Table comparing before/after regularization
    - Explanation of each technique’s impact

CNN Variant

- **Tasks**: Use Fashion MNIST or CIFAR-10 to practice image-based classification.
  - Train a CNN with:
    - Conv2D + MaxPooling
    - Flatten + Dense
    - Dropout and L2
  - Discuss when CNNs are better than MLPs
  - **Deliverable**:
  - CNN summary and comparative notes

Final Deliverable

- **Jupyter Notebook** with:
  - Clean code, comments, explanations
  - Visuals and performance reports
- **1-slide summary** (PPT file):
  - Problem, approach, insights, value

Evaluation Criteria

Section	Points	Criteria
EDA & Insights	1	Relevant visuals, patterns, and business logic
Baseline Model	1	Code quality, architecture logic, baseline accuracy
Regularization Strategy	2	Techniques applied, reasoning, performance improvement
CNN Challenge (Optional)	2	Correct logic, architecture discussion
Final Summary	2	Business clarity and synthesis of results
Code Style & Documentation	2	Commented, readable, cleanly structured