
31/12/2025

Assignment for Research Role at IIT Delhi

Overview

In the Indian e-commerce logistics ecosystem, **Return-to-Origin (RTO)** occurs when a shipment is sent back to the seller due to reasons such as customer refusal, failed delivery attempts, or incorrect or incomplete address details.

RTOs have a **direct and significant impact on business profitability**, leading to increased logistics costs, holding delays, and reduced customer lifetime value.

High RTO rates strain courier partnerships, disrupt supply chain planning, and materially affect margins, especially in cash-on-delivery heavy markets like India.

What Does an RTO (Return to Origin) Mean in Courier?

Objective: You are tasked with building a production-grade pipeline that ingests raw logistics data, predicts the probability of an order being returned (RTO), and deploys this logic as a scalable backend. This assignment tests your ability to handle messy real-world data, apply machine learning, and engineer a robust software solution suitable for deployment.

Datasets:

- Meesho order data
- DTDC Courier Dataset
- Delhivery Data
- Transport GIS Dataset | Open Government Data (OGD) Platform India

Key Deliverables (To-Do's)

Part 1 : **EDA** :

- Implement a data cleaning pipeline that takes care of duplicate records, inconsistent timestamps, missing values, and noisy categorical labels.
- Construct a unified, extensible dataset by joining and aligning the provided data sources (e.g., order-level data, courier data, and geographic or GIS data). Feel free to make assumptions.
- Use EDA to find potential sources of bias, distributional skews, and problems with data quality. Decisions about feature engineering and modeling should be influenced by key findings.
- Make sure the data processing code is testable, modular, and includes unit tests for important transformations.

Part 2: **Machine Learning:**

- Formulate the RTO prediction task as a supervised problem and create suitable baseline models.
- Use ethical strategies to address class disparity and provide justification for your chosen course of action.
- Create and assess features that are derived from temporal, geographic, and transactional signals, with a clear discussion of their predictive significance.
- Use a text-based approach (rule-based, NLP, or lightweight LLM-assisted) to evaluate the accuracy and comprehensiveness of address data.
- Discuss performance constraints and choose evaluation metrics suitable for unbalanced classification.

Part 3: **Interface/Results:**

- Create a dashboard-based UI to analyze and visualize model outputs.

- Provide comprehensible information, such as feature importance or attribution, geographic concentration of RTO risk
- Analyze results critically, emphasizing uncertainty, failure modes, and situations in which the model might not be trustworthy.

Part 4: **Deployment** (**DevOps+MLOps)**

- Organize the stages of data processing, model training, evaluation, and inference in the machine learning pipeline.
- By maintaining and versioning datasets, feature transformations, and trained model artifacts, you can guarantee reproducibility, use **MLFlow**.
- Use suitable testing techniques, such as consistency checks between training-time and inference-time data processing, to validate model behavior.
- Make deployment for inference as a backend service (e.g., using **FastAPI**)
- Create Test cases to evaluate your APIs and
- Handle the ML system's fundamental operational readiness, such as input validation, inference protections, or hooks for upcoming monitoring and retraining.

Evaluation Criteria

1. **Programming Skills :**
    - Efficient use of Python and ML libraries
    - Clean, well-structured code with appropriate abstractions
    - Proper handling of edge cases

2. **Software Engineering :**
    - Modular and logical project structure
    - Clear separation between data processing, modeling, and inference
    - Presence of unit tests for key components

3. **Exploratory Data Analysis (EDA):**
    - Analysis of data distributions, missing values, and outliers
    - Identification of potential biases (geography, courier, payment mode, time)
    - EDA insights reflected in feature engineering and modeling decisions

4. **Feature Engineering, Modeling & NLP:**
    - Meaningful feature engineering and handling of class imbalance

- Appropriate model selection and evaluation approach
- Use of NLP techniques for assessing address quality

5. **MLOps:**
    - Reproducible training and inference pipelines
    - Proper management of model artifacts
    - Input validation and inference safeguards

6. **Communication & Presentation**
    - Clear README explaining assumptions and design choices
    - Effective visualizations or dashboards