# Engineering Documentation: EDD Integration with ClickPost

## Overview

This document outlines the integration of the Estimated Delivery Date (EDD) prediction model into the ClickPost system, enabling consumption of EDD predictions on both individual order and bulk levels.

## Model Description

The EDD prediction model is a machine learning solution that estimates delivery times based on features like:  
- Courier partner ID  
- Account type  
- Order shipment details (e.g., shipping date)  
- Source and destination pin codes  
- Delivery SLA  
- Additional engineered features (e.g., historical data, account mode encoding).  
  
The model uses an XGBoost regressor trained on historical delivery data to predict the number of days required for an order to be delivered. Outputs are stored in the `submission.csv` file, which includes the predicted EDDs for orders.

## Integration Approach

### 1. System Requirements

- A REST API or webhook endpoint to receive order details and provide predicted EDDs.  
- Access to a database containing order, courier, and pin code details.  
- Bulk processing support for batch predictions.

### 2. Integration Architecture

1. Input Pipeline:  
 - Accept inputs via a REST API or file upload.  
 - Required input fields:  
 - `courier\_partner\_id`  
 - `order\_shipped\_date`  
 - `account\_type\_id`  
 - `drop\_pin\_code`  
 - `pickup\_pin\_code`  
 - `quantity`  
 - `account\_mode`  
  
2. Preprocessing:  
 - Validate input data (e.g., check pin code formats, courier IDs).  
 - Fetch latitude/longitude for `pickup\_pin\_code` and `drop\_pin\_code` if geodesic distance is added.  
 - Encode categorical fields (e.g., `account\_mode`).  
 - Standardize/normalize any numerical inputs if required by the model.  
  
3. Prediction Service:  
 - Load the trained XGBoost model (e.g., from a saved pickle or joblib file).  
 - Use the preprocessed data to generate EDD predictions.  
  
4. Output Pipeline:  
 - Return the predicted EDD as a response to the API call.  
 - For bulk requests, generate a downloadable CSV file containing `id` and `predicted\_exact\_sla`.

### 3. Workflow for Single Order EDD Prediction

1. Input: Order details (JSON payload via API).  
2. Processing:  
 - Validate data.  
 - Preprocess input features for compatibility with the model.  
 - Predict EDD using the model.  
3. Output: Return predicted EDD in JSON format.  
 Example:  
 {  
 "id": "428365149",  
 "predicted\_exact\_sla": 5  
 }

### 4. Workflow for Bulk EDD Prediction

1. Input: CSV file with order details.  
2. Processing:  
 - Validate and preprocess each row.  
 - Predict EDD for each order in the batch.  
3. Output: Generate a CSV file with `id` and `predicted\_exact\_sla` for each order.

### 5. Deployment Strategy

1. Infrastructure:  
 - Deploy the model as a microservice using Flask or FastAPI.  
 - Host the service on a scalable cloud platform (e.g., AWS Lambda, Azure Functions).  
2. Endpoints:  
 - Single Order Prediction: `/predict` (POST)  
 - Bulk Prediction: `/bulk\_predict` (POST with file upload)  
3. Monitoring:  
 - Implement logging for each prediction request.  
 - Use monitoring tools (e.g., Prometheus, Grafana) to track API usage and model performance.

### 6. Error Handling

- Invalid Inputs:  
 - Return HTTP 400 with error details.  
- Prediction Failures:  
 - Log errors and return HTTP 500 with a generic message.

### 7. Performance Optimization

- Batch process bulk predictions to reduce latency.  
- Cache static data like pin code details to reduce API/database calls.  
- Enable GPU acceleration for model inference if supported.

## Benefits of Integration

- Real-time and batch-level EDD predictions improve logistics planning.  
- Seamless integration with ClickPost enhances the platform’s capabilities.  
- Provides end-users with accurate delivery timelines, improving customer satisfaction.

## Example API Specifications

### Single Order Prediction

- Endpoint: `/predict`  
- Method: POST  
- Payload:  
 {  
 "id": "428365149",  
 "courier\_partner\_id": 3,  
 "order\_shipped\_date": "2022-09-02",  
 "account\_type\_id": 2520,  
 "drop\_pin\_code": 243001,  
 "pickup\_pin\_code": 122506,  
 "quantity": 1,  
 "account\_mode": "Air"  
 }  
- Response:  
 {  
 "id": "428365149",  
 "predicted\_exact\_sla": 5  
 }