

# Intel® AI for Manufacturing Certificate Course

## Week 6: Streamlit Deployment Report

**Name:** Paresh Patil

**Submission Date:** 12 – 06 – 2025

### 1. Introduction

The final objective of the Timelytics project is to deploy the machine learning model that predicts **Order to Delivery Time**. To make this accessible to end-users, we developed a **Streamlit-based web application**. The app allows users to input order-related details like product category, customer location, and shipping method to receive an estimated delivery time.

---

### 2. Model Preparation

Before deployment, the following steps were completed to prepare the model:

- **Data Cleaning and Preprocessing:** Handled missing values, outliers, and encoded categorical variables.
  - **Feature Engineering:** Created features such as:
    - `distance` between seller and customer
    - `wait_time` (difference between order date and delivery date)
    - `purchase_day_of_week` and `purchase_month`
  - **Model Training:** Trained a regression model (e.g., Random Forest Regressor) to predict delivery time using training data.
  - **Model Evaluation:** Evaluated using metrics like MAE, MSE, RMSE, and  $R^2$  Score to ensure acceptable performance.
- 

### 3. Streamlit App Development

The Streamlit app was structured into the following components:

#### 3.1 User Interface Design

Using `st.sidebar`, a simple and user-friendly form was created to allow user input:

- **Product Category** (Dropdown)
- **Customer State** (Dropdown)
- **Seller State** (Dropdown)
- **Shipping Method** (Dropdown)

- **Distance** (Number input)
- **Order Purchase Date** (Date input)

### 3.2 Backend Logic

Upon user input, the backend code performs:

- **Feature Transformation:** Inputs are encoded/processed to match the model's expected format.
- **Model Loading:** The trained model is loaded using `joblib` or `pickle`.
- **Prediction:** The processed input is passed to the model, and the predicted delivery time is returned.

### 3.3 Sample Code Snippet (Backend Logic)

```
import streamlit as st
import pandas as pd
import joblib

# Load model
model = joblib.load('final_model.pkl')

# Get input from user
product = st.sidebar.selectbox("Product Category", categories)
customer_state = st.sidebar.selectbox("Customer State", states)
shipping_method = st.sidebar.selectbox("Shipping Method", methods)
distance = st.sidebar.number_input("Distance (in km)")
purchase_date = st.sidebar.date_input("Order Purchase Date")

# Process input (e.g., label encoding, feature extraction)
# ...

# Make prediction
input_data = pd.DataFrame({...}) # Constructed from processed user input
prediction = model.predict(input_data)

st.write(f"Predicted Delivery Time: {round(prediction[0], 2)} days")
```

---

## 4. Interface Features

- The user interface was designed to be intuitive and requires no technical knowledge.
  - All inputs are controlled via dropdowns or simple fields, ensuring valid data entry.
  - Once inputs are provided, users can view the predicted delivery time directly on the app screen.
- 

## 5. Conclusion

This assignment successfully demonstrates the integration of a machine learning model into a real-time web application using Streamlit. The interface is designed for simplicity and ease of use. It allows businesses and logistics teams to estimate delivery timelines effectively using AI predictions based on historical order data.