ShipmentSure: Predicting On-Time Delivery Using Supplier Data

1. Objective

The objective of this project is to develop a machine learning model that predicts whether an order will reach the customer on time based on various supplier-related and order-related factors. This will help manufacturing or logistics firms evaluate operational reliability and plan corrective actions in procurement and delivery systems.

2. Dataset

Source: Kaggle – Supply Chain Logistics Dataset

Link: https://www.kaggle.com/datasets/prachi13/customer-analytics

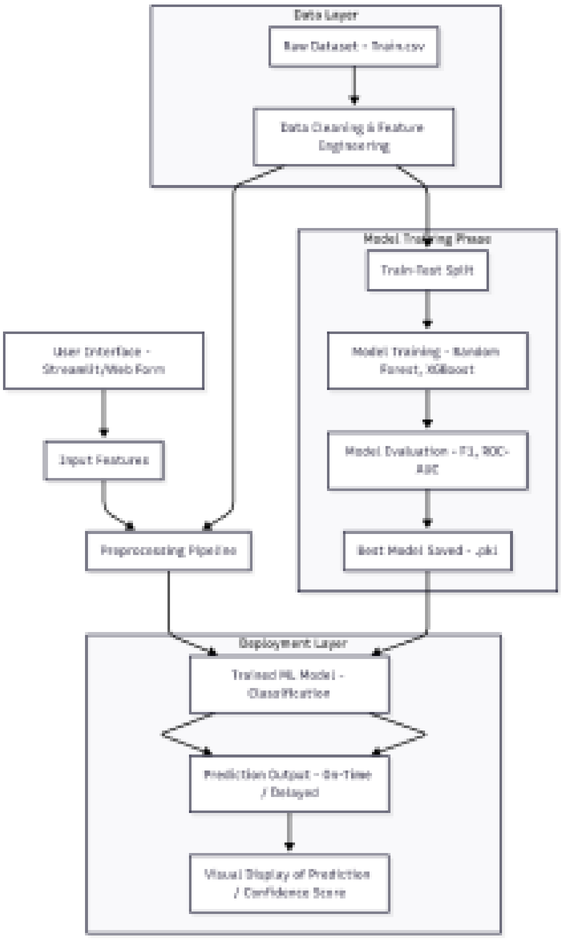
3. Project Outcome

- A classification model that predicts whether a shipment will be delayed or arrive on time.
- Insights into which features most influence on-time delivery.
- A dashboard or user interface for real-time prediction.

4. Project Workflow

1. Data Collection and Understanding
2. Exploratory Data Analysis (EDA)
3. Data Preprocessing
4. Feature Engineering
5. Model Building
6. Model Evaluation
7. Deployment and Documentation

5. System Architecture


Data Layer
Raw Dataset - Train.csv
Data Cleaning & Feature Engineering
Model Training Phase
Train-Test Split
Model Training - Random Forest, XGBoost
Model Evaluation - F1, ROC-AUC
Best Model Saved - .pkl
User Interface - Streamlit/Web Form
Input Features
Preprocessing Pipeline
Deployment Layer
Trained ML Model - Classification
Prediction Output - On-Time / Delayed
Visual Display of Prediction / Confidence Score

6. Week-wise Module Implementation and Milestones

Milestone 1: Week 1–2

Module: Data Understanding and Exploration

- Understand dataset schema and data types
- Perform univariate and bivariate analysis
- Visualize feature distributions and relationships
- Examine class imbalance in `Reached.on.Time_Y.N`

Deliverables:

- Annotated Jupyter notebook for EDA
- Initial insights and EDA visualizations

Milestone 2: Week 3–4

Module: Data Preprocessing and Feature Engineering

• Handle missing values (if any)
• Encode categorical variables (`Mode_of_Shipment`, `Product_importance`, `Gender`) • Normalize numerical features
• Engineer features like cost-to-weight ratio if needed
• Split data into train-test sets

Deliverables:

• Cleaned dataset with documented pipeline
• Feature correlation heatmap
• Final feature set ready for modeling

Milestone 3: Week 5–6

Module: Model Building and Evaluation

- Train multiple models:
  - Logistic Regression
  - Random Forest
  - XGBoost
- Use GridSearchCV for hyperparameter tuning
- Evaluate using metrics:
  - Accuracy
  - Precision
  - Recall
  - F1-Score
  - Confusion Matrix
  - ROC-AUC

Deliverables:

- Model performance comparison table
- Confusion matrix and ROC-AUC plots
- Final model selection

Milestone 4: Week 7–8

Module: Deployment and Documentation

- Build a Streamlit or Flask-based web app
  - Input: order-level feature data
  - Output: probability of on-time delivery
- Prepare project report and presentation

- Upload code and documentation to GitHub

Deliverables:

- Working ML application interface
- Final PDF report
- GitHub repository with code and documentation

7. Evaluation Criteria

Milestone Evaluation Metrics Week 2 EDA completeness, variable understanding Week 4 Quality of feature engineering and preprocessing Week 6 Model accuracy and evaluation scores Week 8 Successful deployment and presentation clarity

8. Tech Stack

Component	**Tools/Libraries**
Programming	Python
Data Handling	Pandas, NumPy
Visualization	Seaborn, Matplotlib
Modeling	scikit-learn, XGBoost
Deployment	Streamlit or Flask