Car Shipment Delay Prediction

Predicting shipment delays, delay duration, and financial loss using machine learning and business intelligence tools

**Prepared by:** Amneet Kaur

**Role:** Data Scientist

**Email:** amneet1224@gmail.com

**LinkedIn:** linkedin.com/in/amneetkaur24

**Date:** July 24, 2025

**Tools Used:** Python, Scikit-learn, Streamlit, Power BI, Git, Pandas, Random Forest, MAE, R²

**Company Use Case:** Automotive Supply Chain — Delay and Financial Risk Management

A isometric view of a car factory

AI-generated content may be incorrect.

Executive Summary

In the fast-moving automotive industry, even a small delay in receiving car parts can lead to big problems—like production slowdowns, extra costs, and missed delivery deadlines. That’s why it’s so important for companies to know in advance if there’s a risk of delay.

This project focuses on solving that exact problem.

We built a smart system that can **predict three key things—two days before a car is scheduled to be assembled**:

1. **Will the shipment be delayed or not?**
2. **If delayed, how many days will it be late?**
3. **How much financial loss could that delay cause?**

To do this, we used real shipment data and applied advanced machine learning techniques. The models we created—especially the Random Forest ones—performed really well, correctly predicting delays more than 85% of the time and estimating financial loss within $445 on average.

Along with the models, we also created a professional **Power BI dashboard**. This makes it easy for business teams to see where delays are happening, which suppliers are causing problems, and what the trends look like over time. With this, decision-makers can take quick action to fix issues before they become costly.

In short, this project helps supply chain teams make better decisions ahead of time—saving money, improving efficiency, and reducing the risk of unexpected delays.

# Table of Contents

1. Introduction

2. Business Problem

3. Dataset Overview

4. Data Preprocessing

5. Exploratory Data Analysis (EDA)

6. Feature Engineering

7. Modeling Approach

8. Model Performance

9. Deployment & App Demo

10. Power BI Dashboard

11. Business Impact

12. Future Work

13. Conclusion

14. References

15. Appendix

# Introduction

In automotive manufacturing, even a minor part delay can disrupt vehicle assembly and lead to substantial financial losses. With complex global supply chains and tightly scheduled production lines, predicting such delays before they occur is vital for proactive planning and cost control.

This project presents a machine learning solution that predicts **car shipment delays** ahead of time — specifically, **two days before a vehicle’s scheduled assembly date**. The objective is to help planners and supply chain teams act in advance by answering three key questions:

* **Will the shipment be delayed?**
* **If delayed, by how many days?**
* **What will be the financial impact of that delay?**

The solution is based on an AI-generated dataset of 1,010 historical car shipments, which includes part-level delivery records, supplier information, regional details, transport modes, holidays, weather flags, and more. To avoid **data leakage**, all logic and feature engineering are based on what would realistically be known **exactly two days before the car’s scheduled assembly**.

Custom rules were applied to define delay status: if a part had not yet arrived by **T−2 (two days before assembly)**, it was marked as **delayed** — regardless of whether it arrived later. Using this logic, several features were engineered: A **shipment-level delay flag**, whether **any critical parts** were delayed, the **sum of critical parts delayed.**

Three machine learning models were developed:

* A **classification model** to predict whether a car shipment will be delayed
* A **regression model** to estimate the number of delay days
* A **regression model** to calculate the financial cost of the delay

After thorough data cleaning, transformation, and feature selection, these models were trained using Random Forest algorithms and evaluated on F1-score, MAE, and R² metrics. The solution is deployed through a **Streamlit web application** that allows real-time prediction and interaction based on live inputs. A **Power BI dashboard** also supports leadership and planners in visualizing delay patterns, risk regions, and cost exposure across the supply chain.

This report outlines the entire pipeline — from business problem framing and data logic to model development, deployment, and business value delivery — with a strong focus on proactive decision-making in manufacturing logistics

# Business Problem

In car manufacturing, everything depends on timing. Each vehicle relies on dozens of parts arriving from different suppliers — on schedule — to keep the production line running smoothly. But delays happen all the time. A supplier might be running behind, a shipment could get stuck during a long weekend, or weather conditions might slow things down.

The challenge is that these delays do not just hold up one part — they can stop the entire car from being built. That means idle labor, disrupted production, and losses that add up quickly. Even worse, these problems are often discovered too late — when the car is already on the assembly floor and the missing part hasn’t arrived.

That’s the core business problem this project aims to solve: **how can we predict, ahead of time, whether a car’s shipment will be delayed?** And if we already suspect a delay is coming, **how long will it last, and what will it cost the company?**

The system developed in this project addresses those questions by analyzing information available **exactly two days before the scheduled car assembly date**. That includes part delivery records, supplier performance, holidays, regional data, and more. The goal is to give operations and planning teams a real window of time to act — whether that’s adjusting schedules, sourcing parts differently, or notifying stakeholders in advance.

In simple terms: instead of reacting to delays after they happen, this project helps prevent them from becoming costly disruptions in the first place.

# Dataset Overview

This project uses an AI-generated dataset that simulates approximately **1,010 historical car part shipments**, with each record representing the shipment status of a car just before its assembly.

Each data point includes part-level and shipment-level details available **two days before the scheduled car assembly date**. These include:

* **Shipment Metadata:** Car ID, scheduled assembly date, shipment month, route type, transport mode.
* **Part-Level Info:** Whether any critical part was delayed, number of delayed parts, part types (engine, brake, infotainment, etc.).
* **Supplier Details:** Supplier region, reliability scores, performance ratings by part type.
* **External Factors:** Traffic severity index, holiday proximity, weather on delivery, contract penalties.

**Target Variables:**

* Will\_be\_delayed – Binary classification (Yes/No)
* Delay\_Days – Regression target (numeric delay duration)
* Financial\_Impact\_USD – Regression target (estimated cost in dollars, log-transformed for modeling)

To ensure that the model only uses information available at the time of prediction, a custom logic was applied to define a part as **delayed** if it had not yet arrived by **T−2** (two days before assembly). Even if the part eventually arrived on time, it was considered **at risk** for modeling purposes.

Additionally, to prevent **data leakage**, features like Supplier\_Reliability (which could reflect post-outcome scores) were removed. Instead, the dataset uses **historical and observable indicators** such as Supplier\_Performance\_Score and real-time delay flags based on part arrival timestamps.

After cleaning and preprocessing, the final dataset was split and used to train three separate machine learning models — each addressing a different business question around delays, duration, and cost.

# Data Preprocessing

The preprocessing phase was a critical step in preparing the raw shipment dataset for analysis and machine learning. The original dataset contained 1,010 rows and over 60 features capturing car part deliveries, supplier information, regional factors, and potential external delays.

To ensure data quality and modeling reliability, the following preprocessing tasks were performed:

# 1 Data Type Conversion and Standardization

Several columns were misclassified at load time, especially date and categorical fields. These were corrected as follows:

* Date fields such as Scheduled\_Assembly\_Date and part delivery dates were converted to datetime objects for accurate time-based feature extraction.
* Delay reasons and supplier groups were encoded as categorical types to reduce memory usage and enable efficient encoding during modeling.
* Categorical string values were standardized by stripping spaces and converting to uppercase to prevent mismatches and duplication during grouping or analysis.

# 2 Missing Value Imputation

Missing data was found across delay reasons, supplier fields, transport modes, reliability scores, and financial impact. A domain-aware strategy was used for imputation:

* Delay reason columns were filled conditionally:
  + If the corresponding delay flag was 0 (not delayed), the reason was filled as “On-Time”.
  + If the delay flag was 1, the missing reason was set to “Unknown\_Delayed”.
* Numerical columns like Financial\_Impact\_USD and reliability scores were filled using the median to reduce the influence of outliers.
* Categorical fields such as supplier region and transport mode were imputed using the most frequent value (mode) in their respective columns.

This approach helped maintain both the integrity and meaning of the data.

# 3 Duplicate Removal

Duplicate rows were identified and removed to ensure that each car shipment represented a unique case, preventing biased model training.

# 4 Outlier Detection and Capping

Certain numeric fields, particularly Delay\_Days and Financial\_Impact\_USD, showed significant outliers:

* Negative values in Delay\_Days were corrected to 0, as negative shipping delays are logically invalid.
* Values in both Delay\_Days and Financial\_Impact\_USD above the 95th percentile were capped to reduce skew and limit the impact of extreme values on model predictions.

This ensured more stable learning during regression tasks.

# 5 Final Data Quality Checks

After preprocessing, the dataset was validated through:

* Missing value verification (all missing entries successfully imputed)
* Data type consistency check
* Duplicate re-check
* Export of the cleaned dataset for further use in feature engineering and modeling (Cleaned\_car\_automated\_dataset.csv)

# Exploratory Data Analysis (EDA)

To understand the dynamics of shipment delays and their financial implications, we conducted an in-depth exploratory data analysis across all key variables.

**Delay Patterns & Distribution**

* Over **60% of shipments** experienced delays.
* The **distribution of delay days is right-skewed**, indicating that while many shipments are delayed by only a few days, a smaller subset contributes to significantly longer disruptions.
* A **delay vs no-delay split** shows the widespread nature of the issue.

**Include Graphs**:

A graph showing a number of bars

AI-generated content may be incorrect.

A graph of a distribution of car shipment

AI-generated content may be incorrect.

**2. Criticality & Risk Exposure**

* **Critical parts** (such as engines, brakes, transmissions) consistently show **higher average delay days**.
* These parts are also linked to **higher financial losses** when delayed, underlining their role in production risk.

**Categorical Distributions**

* **The dataset includes various transport modes and Route types.**
* **These show diverse distributions that could influence shipment outcomes and were retained as features.**

A graph of a number of vehicles

AI-generated content may be incorrect.A graph of a traffic level

AI-generated content may be incorrect.

**Correlation Matrix**

* **Delay\_Days has a moderate positive correlation with Financial\_Impact\_USD, validating it as a predictive feature.**
* **Categorical binary features like criticality and supplier region have minimal direct correlation but may have interaction effects.**

A graph showing the heatmap

AI-generated content may be incorrect.

**Delay vs Financial Loss (Relationship)**

* **A clear positive trend is visible between delay days and financial loss: more delay = more financial damage.**
* **However, some delays cause high losses even with fewer days, indicating part criticality or supplier influence.**

A graph with numbers and points

AI-generated content may be incorrect.

**Financial Impact**

* **Financial loss** is positively correlated with **delay days** — shipments delayed by over 10 days contribute disproportionately to losses.
* Like delays, financial loss is also **right-skewed**, with a few high-loss cases standing out.

A graph with red squares

AI-generated content may be incorrect.

A graph with numbers and points

AI-generated content may be incorrect.

# Feature Engineering

# Feature Engineering (Classification Model)

The feature engineering process was designed to convert the cleaned raw shipment dataset into a model-ready format that accurately captures business context, real-world logistics constraints, and delay risk factors. This phase focused on generating informative and interpretable features for building a **classification model to predict whether a car shipment will be delayed or not**.

# 1.1 Temporal Feature Creation

To identify seasonal and cyclical trends in shipment delays, we extracted time-based features from the car’s scheduled assembly date:

* **Month\_Scheduled**: The calendar month when the car was scheduled for assembly.
* **Quarter\_Scheduled**: The corresponding quarter (Q1–Q4) to capture high-level seasonality.

These features help the model account for patterns such as peak demand seasons, holidays, and supplier performance cycles that often influence part delivery times.

# 1.2 Part Delay Flags Based on 2-Day Cutoff

In real-world automotive supply chains, parts must arrive at least **two days before the scheduled car assembly date** to allow for quality checks, staging, and processing. Based on this operational rule, we generated binary flags for each critical part:

* **Engine\_Delayed\_Flag**
* **Transmission\_Delayed\_Flag**
* **Brake\_System\_Delayed\_Flag**
* **Suspension\_Delayed\_Flag**
* **Infotainment\_Delayed\_Flag**

Each flag indicates whether the corresponding part arrived late relative to the 2-day cutoff. This forms the foundation for identifying whether shipment delays are caused by late or critical part arrivals.

# 1.3 Aggregated Delay Indicators

To summarize and quantify delay severity, the following composite features were created:

* **Total\_Parts\_Delayed**: Sum of all part delay flags per shipment. It reflects the number of components that arrived late and helps assess how widespread the delay issue is.
* **Any\_Critical\_Part\_Delayed**: A binary feature indicating whether at least one delayed part was marked as critical to the car’s functioning or completion.
* **Sum\_Critical\_Parts\_Delayed**: A count of critical parts that were delayed. This offers a granular measure of the severity of high-risk delays that may cause significant disruption.

These features provide the model with business-relevant signals to better classify shipments at risk.

# 1.4 Data Leakage Prevention

Several columns were removed to avoid leakage into the target variable. These include:

* Exact delivery dates of parts
* Delay days (regression target)
* Financial impact (only known after shipment)
* Internal flags such as Valid\_Delay, Inventory\_Buffer\_Used, and reason text fields

Removing these ensured the model only uses features that would be known **before the scheduled assembly date** — in alignment with the project objective to predict delays two days in advance.

# 1.5 Final Dataset Composition

After feature engineering:

* The dataset included **47 cleaned and derived features** suitable for classification modeling.
* All features were structured to reflect **real-world logistics behavior**, enabling the model to learn from both data patterns and business logic.
* The final engineered dataset was saved as df\_cleaned\_classification\_final.csv.

# Feature Engineering (Regression Model: Delay Days Prediction)

This section outlines the feature engineering strategy implemented for the **regression model** that predicts **how many days a car shipment will be delayed**, if any. The features were carefully crafted to reflect real-world business operations, maintain data integrity, and prevent information leakage during modeling.

# 2.1 Temporal Feature Extraction

To capture patterns associated with seasonality and operational cycles, two new features were extracted from the scheduled assembly date:

* **Month\_Scheduled**: Indicates the month (1–12) when the car is scheduled for production.
* **Quarter\_Scheduled**: Indicates the fiscal quarter (1–4) to detect broader seasonal effects.

These time-based variables help the model detect trends linked to holidays, supplier performance fluctuations, or demand surges.

# 2.2 Delay Flags Based on Business Rule (2-Day Cutoff)

In real-world manufacturing, parts must be delivered at least **two days prior to the scheduled car assembly date**. This allows for essential operations such as quality checks, documentation, sorting, and staging.

To encode this business constraint:

* All five part delivery columns were converted to datetime format.
* For each part, a **binary delay flag** was created:
  + 1 if the part arrived **after** the 2-day cutoff
  + 0 if it arrived on time or earlier

The following flags were created:

* Engine\_Delayed\_Flag, Transmission\_Delayed\_Flag, Brake\_System\_Delayed\_Flag, Suspension\_Delayed\_Flag, Infotainment\_Delayed\_Flag

# 2.3 Aggregated Delay Metrics

To capture the severity of delays for each shipment, multiple summary features were computed:

| **Feature Name** | **Description** |
| --- | --- |
| **Total\_Parts\_Delayed** | Count of parts that missed the 2-day cutoff |
| **Total\_Delayed\_Ratio** | Ratio of delayed parts out of total five |
| **Is\_Any\_Part\_Delayed** | Binary flag: at least one part delayed |
| **All\_Parts\_On\_Time** | Binary flag: all five parts arrived on time |

These variables give the model a sense of overall shipment readiness.

# 2.4 Criticality-Aware Delay Features

In vehicle assembly, not all parts are equally important. Delays in **critical parts** are more likely to impact final shipment dates. Therefore, the following engineered features were added:

| **Feature Name** | **Description** |
| --- | --- |
| **Any\_Critical\_Part\_Delayed** | Binary: at least one delayed part was marked critical |
| **Sum\_Critical\_Parts\_Delayed** | Count of delayed parts that were also marked critical |
| **Total\_Critical\_Parts** | Count of parts marked as critical for a given car |
| **Non\_Critical\_Parts\_Delayed\_Count** | Parts delayed that were not critical |
| **Critical\_Parts\_Ratio** | Ratio of critical delayed parts to total critical parts |
| **All\_Critical\_Parts\_On\_Time** | Binary: all critical parts arrived on time |

These indicators enable the regression model to weigh delays more accurately based on their actual production impact.

# 2.5 Leakage Prevention and Column Pruning

To ensure that the model only learns from data available **before the scheduled car assembly date**, several columns were dropped:

* Exact delivery dates and delay reasons for each part
* Post-event outcomes such as:
  + Car\_Delayed
  + Financial\_Impact\_USD
  + Valid\_Delay

Additionally, Car\_ID, Scheduled\_Assembly\_Date, and other reference fields were excluded to avoid overfitting or data leakage.

# 2.6 Final Dataset Output

* Final dataset size: **1,010 rows**, **45 columns**
* Dataset file saved as: **delay\_days\_regression\_final.csv**
* Features ready for use in regression models to predict delay durations

# Conclusion

This feature engineering pipeline builds a robust, business-aware dataset suitable for predicting **the number of delay days** per shipment. By incorporating temporal trends, part-level delays, and criticality context — while ensuring no data leakage — the engineered dataset balances predictive power with real-world relevance.

# Feature Engineering (Financial Impact Prediction)

This section details the feature engineering process for the **regression model** designed to predict the **financial impact (in USD)** of car shipment delays. The goal was to develop meaningful, delay-aware, and context-sensitive features that align with business reality and improve model robustness.

# 3.1 Temporal Features for Seasonality

To capture patterns linked to production cycles, seasonal variations, and holidays:

* **Month\_Scheduled** and **Quarter\_Scheduled** were extracted from the scheduled assembly date.
* These features help identify shipment delays related to peak seasons, factory workloads, or external disruptions.

# 3.2 Delay Flags Based on Business Cutoff

For a part to be considered “on time,” it must be delivered at least **2 days before the scheduled assembly date**.

* Delivery dates for five major parts were converted to datetime format.
* Delay flags were created:
  + 1 = delayed beyond the cutoff
  + 0 = delivered on time

These binary indicators were created for:

* Engine, Transmission, Brake System, Suspension, Infotainment

# 3.3 Aggregated Delay Indicators

To quantify the severity of shipment delays:

| **Feature** | **Description** |
| --- | --- |
| **Total\_Parts\_Delayed** | Total number of delayed parts |
| **Is\_Any\_Part\_Delayed** | Binary flag: at least one part was delayed |
| **All\_Parts\_On\_Time** | Binary flag: all parts were on time |

# 3.4 Critical Part Delay Metrics

Some parts are more essential than others. To measure impact severity:

| **Feature** | **Description** |
| --- | --- |
| **Any\_Critical\_Part\_Delayed** | Binary: at least one critical part was delayed |
| **Sum\_Critical\_Parts\_Delayed** | Count of critical delayed parts |
| **Total\_Critical\_Parts** | Total number of parts marked as critical |
| **All\_Critical\_Parts\_On\_Time** | Binary: all critical parts arrived on time |
| **Non\_Critical\_Parts\_Delayed\_Count** | Total delayed parts - critical delayed parts |
| **Critical\_Parts\_Ratio** | Proportion of critical parts delayed |

These features allow the model to prioritize shipments impacted by critical part failures.

# 3.5 Ratio and Interaction Features

We created advanced interaction features by combining urgency, part reliability, and environmental disruptions:

* **Delay\_Per\_Urgency**: Delay count normalized by order urgency level
* **Non\_Critical\_Parts\_Delayed\_Ratio**: Share of non-critical delayed parts
* **Reliability\_Gap\_X**: Differences in supplier reliability scores across parts
* **Reliability\_Score × Delay Flags**: Measures how unreliable parts affected outcomes
* **Criticality × Delay Flags**: Highlights severity due to delayed critical parts

# 3.6 External Risk Factors

To account for external drivers of shipment delays:

| **Feature** | **Description** |
| --- | --- |
| **Urgency\_x\_TotalDelayed** | High-urgency orders with more delays |
| **Holiday\_x\_CriticalDelay** | Holiday proximity combined with critical part delay |
| **Strike\_x\_TotalDelayed** | Supplier strike alerts and delay count |
| **Weather\_x\_TotalDelayed** | Adverse weather during delivery and delay intensity |

These features provide contextual depth that models often lack.

# 3.7 Target Transformation

The **Financial\_Impact\_USD** column showed strong right skew — with most shipments having low or no penalty, and few showing high costs.

* To normalize the distribution, a **log1p transformation** was applied.
* This stabilized model training and reduced variance.
* df['Financial\_Impact\_Log'] = np.log1p(df['Financial\_Impact\_USD'])

# 3.8 Data Leakage Control and Cleanup

To maintain model integrity, all columns that might leak target information were removed:

* Delay outcome flags (Delay\_Days, Car\_Delayed, etc.)
* Delivery dates (not available before prediction time)
* Reference-only fields (e.g., Car\_ID, part delay reasons)

# 3.9 Final Output

* Final dataset: 1,000 rows, 50+ engineered features
* File saved as: Financial\_Impact\_Regression\_Final\_Dataset.csv
* Target variable: Financial\_Impact\_Log (log-transformed USD impact)

# Conclusion

This feature engineering pipeline ensures that the model predicting financial impact is both **data-leakage-proof** and **business-aware**. It combines urgency, part-level delivery metrics, supplier reliability, and external factors into a powerful, real-world-ready dataset.

# Modeling Approach

# Model 1: Car Shipment Delay Classification (Binary Classification)

# 1.1 Business Objective

Predict whether a car shipment will be delayed or not (Will\_be\_delayed: Yes/No) using historical features related to supplier reliability, part-level delivery, traffic, weather, and shipment urgency.

# 1.2 Modeling Technique

A **Random Forest Classifier** was used for its robustness, interpretability, and ability to handle both categorical and numerical features without complex preprocessing.

# 1.3 Data Preparation Steps

* **Top 15 most predictive features** were selected based on prior EDA and domain knowledge.
* Applied **Label Encoding** on supplier group and environmental variables.
* Encoded Traffic\_Level using **Ordinal Encoding** (LOW < MODERATE < HIGH).
* Created the binary target Will\_be\_delayed from the original Car\_Delayed flag.
* Used **SMOTE** oversampling to balance the dataset, addressing class imbalance between delayed and non-delayed shipments.

# 1.4 Train/Test Split

* 80/20 stratified split to preserve class proportions.
* Applied encoding and SMOTE only on the training set to prevent data leakage.

# 1.5 Model Training and Evaluation

| **Metric** | **Value** |
| --- | --- |
| **Accuracy** | 85.15% |
| **Precision (Delay = Yes)** | 0.89 |
| **Recall (Delay = Yes)** | 0.87 |
| **F1 Score (Delay = Yes)** | 0.88 |
| **Precision (No Delay)** | 0.79 |
| **Recall (No Delay)** | 0.83 |

**Confusion Matrix:**

A blue squares with white text

AI-generated content may be incorrect.

* The model performs **better at identifying delays**, which aligns with business goals — catching a delay is more critical than false-alarming a non-delay.

# 1.6 Feature Importance

Top predictive features include:

* Supplier\_Performance\_Score
* Sum\_Critical\_Parts\_Delayed
* Traffic\_Severity\_Index
* Engine\_Delayed\_Flag
* Weather\_Condition\_On\_Delivery
* Transmission\_Reliability\_Score

A graph with blue and white bars

AI-generated content may be incorrect.

Feature importance was visualized using a horizontal bar chart (Random Forest. feature\_importances\_), supporting transparency and explainability.

# 1.7 Conclusion

This Random Forest classifier provides **accurate and interpretable predictions of shipment delays**, with high recall and precision for identifying delayed shipments. This allows the supply chain team to:

* Prioritize shipments at risk of delay
* Act in advance to avoid penalties or disruptions
* Improve customer satisfaction through early alerts

# Next Steps:

* Try more complex models (e.g., XGBoost, LightGBM)
* Perform hyperparameter tuning
* Add more recent data for retraining

# Model 2: Predicting Delay Days (Regression)

# 2.1 Business Objective

This model aims to **predict the number of days a car shipment will be delayed**, allowing supply chain planners to anticipate and mitigate disruptions ahead of time. Predictions are made **2 days before the scheduled car assembly date**, giving the team enough buffer to take corrective action.

# 2.2 Model Development & Comparison

We tested five regression algorithms on the cleaned dataset containing engineered features such as delivery flags, supplier scores, traffic and weather indicators:

| **Model** | **MAE (days)** | **RMSE (days)** | **R² Score** |
| --- | --- | --- | --- |
| **Random Forest** | **0.87** | **1.13** | **0.805** |
| CatBoost | 0.93 | 1.14 | 0.803 |
| XGBoost | 0.97 | 1.23 | 0.768 |
| Decision Tree | 1.04 | 1.52 | 0.650 |
| Linear Regression | 1.29 | 1.69 | 0.564 |

* **Best Model:** Random Forest
* **Average Delay Error:** Less than 1 day
* **Explained Variance:** ~81% of variation in shipment delay duration

# 2.3 Modeling Steps

* **Data Preparation**  
  Cleaned dataset was used, and top 15 predictive features were selected. One-hot encoding was applied to handle categorical columns.
* **Train-Test Split**  
  80/20 split using random seed 42 for reproducibility.
* **Model Comparison**  
  Five algorithms were trained: Linear Regression, Decision Tree, Random Forest, XGBoost, and CatBoost. Metrics were evaluated on the test set using MAE, RMSE, and R².
* **Model Selection**  
  Random Forest was chosen as the final model due to:
  + Highest R² (0.805)
  + Lowest MAE (0.87 days)
  + High resilience to outliers and noise

# 2.4 Feature Importance

The top features influencing shipment delay duration were:

* Supplier\_Performance\_Score
* Sum\_Critical\_Parts\_Delayed
* Critical\_Parts\_Ratio
* Contract\_Penalty\_Per\_Day
* Traffic\_Severity\_Index

A graph with a bar

AI-generated content may be incorrect.

These reflect real-world delay contributors such as low supplier reliability, delayed essential parts, or traffic congestion near delivery dates.

# 2.5 Conclusion

The Delay Days Regression model provides **highly actionable predictions**, with an average error under 1 day. This gives logistics teams a predictive advantage to:

* Flag critical orders at risk of multi-day delays
* Trigger early interventions with suppliers or transport partners
* Avoid contract penalties and maintain assembly line efficiency

# 2.6 Next Steps:

* Apply hyperparameter tuning (e.g., RandomizedSearchCV)
* Add new real-world data to improve generalization
* Integrate the model into the live shipment tracking system

# Model 3: Predicting Financial Impact of Part Delays

# 3.1 Business Objective

Delays in critical car parts can lead to costly consequences—ranging from production halts to contractual penalties. This model forecasts the **financial impact (in USD)** of part delivery delays, enabling operations and procurement teams to **quantify risk in advance**, optimize planning, and prioritize high-stakes shipments.

# 3.2 Model Development & Performance

We developed a Random Forest Regressor using **log-transformed financial impact values** to handle data skewness. The model was tested using multiple feature sets to identify the best trade-off between accuracy and simplicity.

| **Model Version** | **MAE (USD)** | **RMSE (USD)** | **R² Score** |
| --- | --- | --- | --- |
| All Features | 448.79 | 675.10 | 0.7222 |
| Top 10 Features | 452.94 | 675.38 | 0.7220 |
| **Top 5 Features (Best)** | **445.52** | **664.21** | **0.7311** |
| Top 3 Features | 462.39 | 686.96 | 0.7124 |

* **Best Model:** Random Forest with Top 5 Features
* **Accuracy:** On average, model predicts financial losses within **$445** of actual values
* **Explained Variance:** 73% of financial outcome variation captured

# 3.3 Modeling Pipeline

* **Target Variable:**  
  Financial\_Impact\_Log (log-transformed to reduce skew from extreme values)
* **Feature Engineering:**  
  Selected impactful features based on domain logic and importance metrics, including:
  + Supplier\_Performance\_Score
  + Distance\_KM
  + Brake\_System\_Supplier\_Reliability
  + Sum\_Critical\_Parts\_Delayed
  + Engine\_Supplier\_Reliability
* **Data Preprocessing:**
  + Handled missing values using domain-specific logic
  + One-hot encoded categorical features
  + Standardized numerical features
  + Aligned test/train columns post-encoding
* **Modeling & Evaluation:**
  + Applied Random Forest with 100 trees
  + Transformed predictions back from log-scale to USD
  + Evaluated with MAE, RMSE, R² on test data

**3.4 Feature Importance**

The most influential drivers of cost impact are:

1. **Supplier\_Performance\_Score**
2. **Distance\_KM**
3. **Brake\_System\_Supplier\_Reliability**
4. **Sum\_Critical\_Parts\_Delayed**
5. **Engine\_Supplier\_Reliability**A graph with blue squares

   AI-generated content may be incorrect.

These align with practical business realities—poor supplier performance and long transportation routes increase risks of costly disruptions.

**3.5 Conclusion**

The financial impact regression model equips the supply chain team with **quantitative risk predictions**, two days in advance of the assembly date. With just **5 features**, the model offers:

* High predictive accuracy
* Lightweight design suitable for integration into tracking systems
* Scalability to other product lines or plants

**3.6 Next Steps:**

* Use SHAP or LIME for interpretability in live systems
* Add contextual features (e.g., supplier location, strike data)
* Monitor real-world performance post-deployment

# Model Performance

This section presents a consolidated view of the performance of all three machine learning models developed to address the delay risk in car shipments—whether a delay will occur, how many days it may last, and its financial impact. Each model was rigorously trained and evaluated using appropriate metrics based on the type of problem (classification or regression).

**Model 1: Shipment Delay Classification**

**Goal:** Predict whether a car shipment will be delayed or not (Yes/No)

**Best Model:** Random Forest Classifier

**Evaluation Metrics:**

| **Metric** | **Value** |
| --- | --- |
| Accuracy | 85.15% |
| Precision (Delayed = Yes) | 89% |
| Recall (Delayed = Yes) | 87% |
| F1 Score (Delayed = Yes) | 88% |

**Confusion Matrix:**

|  | **Predicted No** | **Predicted Yes** |
| --- | --- | --- |
| Actual No | 63 | 13 |
| Actual Yes | 17 | 109 |

**Conclusion:**  
The classifier performs exceptionally well, particularly in identifying **delayed shipments**, which is critical to mitigate production risks. With an **F1-score of 88%**, it maintains a strong balance between precision and recall.

**Model 2: Delay Days Regression**

**Goal:** Estimate how many days a delayed shipment will be late

**Best Model:** Random Forest Regressor

**Evaluation Metrics:**

| **Metric** | **Value** |
| --- | --- |
| MAE | 0.86 days |
| RMSE | 1.11 days |
| R² Score | 0.81 |

**Conclusion:** The model accurately predicts delay durations with an **average error of less than one day**. The high R² score (0.81) demonstrates strong explanatory power, making this model ideal for fine-grained delay forecasting and production rescheduling.

A graph of a bar chart

AI-generated content may be incorrect.

**Model 3: Financial Impact Regression**

**Goal:** Predict the financial loss (in USD) caused by shipment delays

**Best Model:** Random Forest Regressor using Top 5 Features

**Evaluation Metrics:**

| **Metric** | **Value** |
| --- | --- |
| MAE | $445 |
| RMSE | $664 |
| R² Score | 0.731 |

**Conclusion:** With **$445 average error** and over **73% variance explained**, this model enables financial planning and risk prioritization. It's lightweight, efficient, and accurate enough for integration into dashboards or real-time alerts.

**Model Comparison Summary**

| **Model** | **MAE** | **RMSE** | **R² Score** | **Accuracy** | **F1 Score** |
| --- | --- | --- | --- | --- | --- |
| Delay Classification | — | — | — | 85.15% | 88% |
| Delay Days Regression | 0.86 days | 1.11 days | 0.81 | — | — |
| Financial Impact Regression | $445 | $664 | 0.731 | — | — |

# Deployment & App Demo

To ensure real-world usability, the machine learning models were deployed as part of an end-to-end interactive web application using **Streamlit**, enabling supply chain managers and stakeholders to make instant predictions on car shipment delays, expected delay duration, and associated financial impact—**two days before the scheduled assembly date**.

# Deployment Pipeline Overview

**Model Serialization:**

All three models were saved using joblib for production deployment:

* + - random\_forest\_model\_classification.pkl
    - best\_model\_rf.pkl (for delay days)
    - random\_forest\_top5\_cost\_model.pkl (for financial cost)

**Web Framework:**

Developed using **Streamlit**, an open-source Python library ideal for quick and interactive app creation.

**User Input Handling:**

Users can enter real-time data for:

* + - Supplier performance
    - Number of critical parts delayed
    - Traffic index, distance, reliability scores
    - Transport/weather conditions

Internally, the app applies the same **data preprocessing logic** used during model training to prevent data leakage and ensure consistency.

**Model Inference:**

* + Once data is entered, the app:
    - Predicts if a delay will occur.
    - If delayed, predicts the **number of days**.
    - Calculates **estimated financial loss in USD**.

**Deployment Hosting:**

* + The application is hosted on **Streamlit Cloud**, accessible via a public URL.

**Live App Demo**

🔗 **App Link:**  
🔗 Click here to access the live demo (<https://car-shipment-delay-prediction-8gxekgz52yqxspeqtu8svv.streamlit.app/>)

**User Interface Screenshots**

A screenshot of a computer

AI-generated content may be incorrect.

# Power BI Dashboard

The **Car\_Shipment\_Power\_BI Dashboard** provides a comprehensive visualization of supply chain bottlenecks, delay patterns, and their financial consequences. It consolidates data from various shipment and supplier systems into a unified interface to support real-time decision-making and operational forecasting.

**Pages in the Dashboard:**

The dashboard contains **10 interactive pages**, each focusing on a critical dimension of delay analysis:

1. **Parts & Supplier Issues**
2. **Trends & Seasonality**
3. **Delay Impact**
4. **Mitigation & Risk Factors**
5. **Transport & Lead Time**
6. **Route Type & Delay Drivers**
7. **Supplier & Traffic Performance**
8. **Strategic Recommendations**
9. **Suggested Next Steps**
10. **Model Summary 1 & 2**

**Key KPIs Displayed:**

| **Metric** | **Value** |
| --- | --- |
| Total Cars Scheduled | 1,010 |
| Delayed Shipments | 631 |
| % Delayed Cars | **62.48%** |
| Total Financial Impact | **$380 Million** |
| Average Delay (Days) | **2.38 days** |

**Slicers/Filters Available:**

* **Shipment Status** (On-time / Delayed)
* **Delay Reason** (Weather, Transport, Part, etc.)
* **Month of Shipment**
* **Part Type** (Critical / Non-Critical)
* **Supplier Region**

**Key Visual Insights:**

* **Delay Days by Supplier Region:**

A graph of blue rectangular shapes

AI-generated content may be incorrect.

**West** region contributes the **highest delay days**, while **South** reports the lowest.

* **Delayed Parts by Region & Part Type:**

A graph of different colored bars

AI-generated content may be incorrect.

In **West**, **Infotainment** is the most delayed part type.

**East** sees lowest delays, especially for **Engine** parts.

In **South**, **Suspension** is the lowest delayed.

* **Sum of Car Delayed by Order Urgency:**

A graph with text and numbers

AI-generated content may be incorrect.

**Medium urgency orders** face the most delays, while **High urgency** orders show the fewest critical car delays.

* **Holiday Proximity vs Delay:**

A blue and red pie chart

AI-generated content may be incorrect.

Cars delayed near holidays: **154 (15.25%)**

On-time near holidays: **856 (84.75%)**

* **Weather Condition on Delivery:**

A graph of a number of different colored squares

AI-generated content may be incorrect.

Majority of delays occurred under **clear** weather, followed by **rainy** days.

* **Unified Transport Mode & Delays:**

A screen shot of a number

AI-generated content may be incorrect.

**Air** transport shows the **highest delay count**, **Road** transport the lowest.

* **Delay Trends by Month & Transport Mode:**

A graph of a line graph

AI-generated content may be incorrect.

A **notable drop** in delays via **Sea** transport over the months, indicating improvement.

* **Delay Days vs Contract Penalty per Day:**

A graph with blue squares

AI-generated content may be incorrect.

Highest delay days are for penalties between **$0–175**, suggesting a weaker penalty incentive.

* **Traffic Severity Index vs Delay Days:**

A graph with a line

AI-generated content may be incorrect.

At a **TSI of 1.2**, the sum of delay days peaks at **1,111 days**, highlighting severe congestion effects.

* **Traffic Level vs Delay:**

A graph with red and blue bars

AI-generated content may be incorrect.

**Moderate traffic** is most associated with delays, surpassing both high and low levels.

# Business Impact

The insights and predictive models developed in this project bring tangible value across multiple aspects of the automotive supply chain — from operations and finance to strategic planning.

**Reduced Shipment Uncertainty**

With over 62% of shipments experiencing delays, the predictive models serve as a data-driven early warning system. High-risk deliveries are flagged in advance, allowing teams to take proactive steps such as rerouting, escalating to suppliers, or reallocating resources — minimizing disruptions before they escalate.

**Cost Savings through Financial Forecasting**

The financial impact model estimates potential losses with a margin of + $445 USD, equipping finance and procurement teams to:

* Quantify risks early in the cycle
* Prioritize high-impact shipments
* Negotiate smarter contracts with performance clauses
* Avoid surprises in quarterly financial reports

This predictive capability supports better financial planning and risk mitigation.

**Smarter Logistics & Routing Decisions**

By analyzing transport modes, route types, and traffic/weather disruptions, the models help logistics teams:

* Optimize shipment paths
* Shift to more reliable transport methods during critical periods

**Strategic Supplier & Part Risk Management**

With part-level granularity (e.g., transmission delays in the South region), the solution enables:

* Supplier re-evaluation and performance benchmarking
* Better inventory planning with just-in-time buffer strategies
* Targeted interventions for high-risk components and suppliers

**Integrated Decision-Making Dashboard**

The Power BI dashboard consolidates all insights into a dynamic executive tool. Managers can:

* Filter data by urgency, part type, supplier region, and more
* Track delay patterns and financial impacts in real time
* Support cross-functional collaboration through shared visibility

**Future-Ready Operations**

Deployed through an app, live dashboards, and automated models, this solution transforms operations by:

* Providing a predictive edge in managing supply chain disruptions
* Building data literacy across teams
* Laying the groundwork for real-time alerts and automated mitigation systems

# Future Work

While the current project offers robust predictive capabilities and business insights, several opportunities exist to enhance its scope, accuracy, and operational value.

**Model Enhancements & Feature Engineering**

* **Include real-time data** feeds such as weather APIs, live traffic, or supplier portal updates for more dynamic predictions.
* Explore **deep learning architectures** (LSTM, GRU) to capture time-sequenced supplier behaviors or cumulative delay patterns.
* **AutoML pipelines** can be incorporated to automate hyperparameter tuning and improve model generalization.

**Integration with Supply Chain Systems**

* Integrate predictions into **ERP platforms (like SAP, Oracle)** or **supply chain control towers** for seamless operations.
* Use model outputs to **trigger alerts and automated actions** (e.g., advance part reorders, priority shipping).

**Mobile & Email Notifications**

* Develop a mobile-friendly interface and **email alert system** for operations teams.
* High-delay risk shipments can trigger proactive messaging to key stakeholders.

**Explainable AI & SHAP-Based Dashboards**

* Incorporate **SHAP (SHapley Additive exPlanations)** into the dashboard to provide feature-level insights for each prediction.
* Help managers understand **why a shipment is at risk**, not just that it is.

**Feedback Loop for Continuous Learning**

* Deploy **retraining loops** where model learns from actual delivery outcomes.
* This improves performance over time and adapts to changes in supplier behavior, routes, or external events.

# Conclusion

This project successfully demonstrates how machine learning and data-driven insights can be applied to **predict car shipment delays and their financial impact** within the automotive supply chain. By analyzing supplier performance, part criticality, traffic severity, weather conditions, and other key factors, we were able to build robust models that:

* **Classify shipments as delayed or on-time** with an accuracy of ~85%.
* **Predict the number of delay days** with an average error of less than 1 day.
* **Estimate financial loss due to delays** with high reliability, achieving a mean absolute error of ~$445.

Our models were carefully designed to avoid data leakage by incorporating realistic cutoff logic (e.g., 2-day prior delay checks) and selecting top features that are truly actionable for business stakeholders. The deployment-ready Streamlit app and the interactive Power BI dashboard provide practical tools for operations and supply chain managers to make proactive decisions in real time.

Overall, this end-to-end solution can help companies like Daikin **anticipate disruptions**, **minimize losses**, and **improve planning**—ultimately enhancing operational efficiency and customer satisfaction.

# References

1. IBM Machine Learning with Python (Coursera)
2. Scikit-learn Documentation: <https://scikit-learn.org/>
3. CatBoost Documentation: <https://catboost.ai/>
4. XGBoost Documentation: <https://xgboost.readthedocs.io/>
5. SHAP Library for Explainable AI: <https://github.com/slundberg/shap>
6. Power BI Official Documentation: <https://learn.microsoft.com/en-us/power-bi/>

# Appendix

**This section contains supporting materials, additional resources, and technical references used throughout the project.**

**A. List of Key Features Used in Each Model**

**1. Delay Classification Model (Random Forest):**

* **Supplier\_Performance\_Score**
* **Sum\_Critical\_Parts\_Delayed**
* **Any\_Critical\_Part\_Delayed**
* **Traffic\_Severity\_Index**
* **Traffic\_Level**
* **Total\_Parts\_Delayed**
* **Distance\_KM**
* **Engine\_Delayed\_Flag**
* **Weather\_Condition\_On\_Delivery**
* **Transmission\_Reliability\_Score**
* **Suspension\_Reliability\_Score**
* **Brake\_System\_Supplier\_Group**
* **Transmission\_Supplier\_Group**
* **Infotainment\_Supplier\_Group**
* **Engine\_Reliability\_Score**

**2. Delay Days Regression Model (Random Forest):**

* **Supplier\_Performance\_Score**
* **Critical\_Parts\_Ratio**
* **Distance\_KM**
* **Sum\_Critical\_Parts\_Delayed**
* **Contract\_Penalty\_Per\_Day**
* **Total\_Critical\_Parts**
* **Any\_Critical\_Part\_Delayed**
* **Transmission/Suspension/Brake System/Engine/Infotainment Reliability Scores**

**3. Financial Impact Regression Model (Random Forest - Top 5):**

* **Supplier\_Performance\_Score**
* **Distance\_KM**
* **Brake\_System\_Supplier\_Reliability**
* **Sum\_Critical\_Parts\_Delayed**
* **Engine\_Supplier\_Reliability**

**B. Model Performance Overview**

| **Model** | **MAE** | **RMSE** | **R² Score** |
| --- | --- | --- | --- |
| Classification (Accuracy) | — | — | **85.15%** |
| Delay Days (Random Forest) | 0.86 days | 1.13 days | 0.81 |
| Financial Impact (Top 5 features) | $445 | $664 | **0.7311** |

**C. Tools & Technologies Used**

* **Python Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, CatBoost, XGBoost, joblib**
* **Modeling Techniques: Random Forest, CatBoost, Linear Regression, SMOTE**
* **Visualization & BI: Power BI, Matplotlib, Seaborn**
* **App Deployment: Streamlit**
* **Version Control: GitHub**

**D. Links to Project Artifacts**

* **📂 GitHub Repository: [**[**Amneetkaur24/car-shipment-delay-prediction**](https://github.com/Amneetkaur24/car-shipment-delay-prediction/tree/main)**]**
* **📊 Power BI Dashboard: Embedded in PDF/ppt or available on request**
* **🌐 Live Streamlit App Demo: [**[**https://car-shipment-delay-prediction-8gxekgz52yqxspeqtu8svv.streamlit.app/**](https://car-shipment-delay-prediction-8gxekgz52yqxspeqtu8svv.streamlit.app/)**]**
* **📄 Full Presentation PDF: Attached with the submission**