National College of Ireland

Optimising Supply Chain Logistics with Predictive Analytics

MSc Research Practicum
MSc Data Analytics

Sidhesh Bhambid
Student ID: x23270110

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sidhesh Bhambid

Student ID: x23270110

Programme: MSCDAD_A **Year:** 2024-25

Module: MSc Research Practicum

Supervisor: Vladimir Milosavljevic

Submission Due Date: 11-08-2025

Project Title: Optimising Supply Chain Logistics with Predictive Analytics

Word Count: 8491 **Page Count**: 23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: SIDHESH BHAMBID

Date: 10-08-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimising Supply Logistics with Predictive Analytics

Sidhesh Bhambid
x23270110

Abstract

Modern supply chains operate within a highly volatile and complex global environment, where disruptions can lead to significant financial losses and damage to brand reputation. The ability to proactively identify and mitigate risks is a critical competitive advantage. This research project addresses the challenge of optimising supply chain logistics through the application of predictive analytics. The primary objective is to develop and evaluate a machine learning model capable of accurately classifying the risk level of logistics operations. The study utilises a comprehensive dataset encompassing over 32,000 logistical events, detailed with 26 features including real time vehicle tracking, fuel consumption, inventory levels, and environmental factors. Following a rigorous methodology of data preprocessing, feature engineering, and dimensionality reduction using Principal Component Analysis (PCA), a comparative analysis of five supervised learning algorithms was conducted: Decision Tree, AdaBoost, K Nearest Neighbors, Gaussian Naive Bayes, and Support Vector Machine (SVM). After careful hyperparameter tuning and using evaluation metrics including accuracy, precision, recall, F1-score, and ROC-AUC, the SVM model was the best predictor of the classes. The final retrained SVM model, after balancing classes and engineered features, had amazing accuracy of 97.83%, with an F1 score of 97.87% on test data. The results of this study highlight how effective predictive analytics can evolve logistics management from a reactive process to one that is proactive, providing a strong instrument to help inform decisions, reduce costs, and build the resilience of supply chains.

1 Introduction

The modern global marketplace is defined by elaborate and sprawling supply chain networks, which are the lifeblood of international commerce. These networks-while enabling greater quantities of fresh trade and conceptual efficiency-are also highly strained due to several factors that could easily topple their integrity. Among those numerous disruptors, geopolitical instability, events like extreme weather, traffic congestion, demand volatility, and unreliable suppliers can cumulatively lead to major disruptions; with time and money lost, lack of operational efficiencies, and customer dissatisfaction (Gardas and Narwane, 2024). The response to challenges in previous times took a reactive stance where, in essence, logistics managers responded to problems as they arose. However, in the time of big data--accompanied by the enormous power of computation it may be argued that such an approach is no longer adequate. With the changes in how it seeks things within the context of managing supply chains, there is not only the issue of timesaving but survival and growth itself is at stake.

More than ever, data analytics and, in particular, machine learning applied in real time are acting as the major sources of transformation within the domain of supply chain management (Aamer et al., 2020). Organizations are, however, enabled to tap into vast streams of data, such as several data sources from the Internet of Things (IoT) sensors, Global Positioning System (GPS) trackers, and enterprise resource planning (ERP) systems, to determine hidden relationships, transform behaviours projected on a graph of the future, and optimize operations in the complex with previously unseen precision. The present study seeks an exploration at the intersection of the above and focuses a great deal on the area of logistics risk management. The central focus is that business would be better off if they can accurately predict problems with efficiency and resilience.

The research question guiding this study is: **To what extent can supervised machine learning models, trained on a comprehensive set of logistical and environmental data, accurately classify the operational risk level of a supply chain shipment?**

To this end, the following objectives were set:

1. To carry out an exhaustive exploratory data analysis on a multi-featured logistics dataset for identifying key variable and relationship impacts on supply chain risk.
2. To develop and engineer relevant features from the raw data to increase the predictivity of machine learning models.
3. To apply and train a bunch of classification models (such as Decision Tree, AdaBoost, K-Neighbors, Naive Bayes, SVM) and compare them depending on predictive performance.
4. To examine the efficacy of the all-out models by performance metrics, validated in a robust way, and tuning parameters to the best predictive model.
5. To pick the best model and analyze it in terms of its utility in functional terms for proactive logistics management.

In scientific light, this study is an immense contribution by way of a strict, comparative qualitative assessment of various machine learning models potentially useful for operational risk classification in logistics. While there are studies mainly targeting single area such as predictive modelling in demand (Douaioui et al., 2024), many others in inventory management (Mugdha et al., 2024), the current study looks into integrating a multiplicative case of real time and historical data sets to construct an integrated risk predictive mechanism. The aim is to provide a highly effective and thoroughly validated predictive model useful for the articulation of the much-vaunted frontier of intelligent automation for risk mitigation in the supply chain.

The report organization is as follows. The next chapter (Chapter 2) will review the academic literature. The methodology for the study will be discussed in Chapter 3, explaining the systematic steps that were followed from data collection to model evaluation. The system design will be regard in Chapter 4, where a list of considerations including data and architecture will be made. Chapter 5 takes a look at the implementation aspects of data processing and modelling. Chapter 6 presents a detailed assessment of experimental results and concludes with a discussion to analyze the results. In turn, Chapter 7 will conclude the report, offering possible future research paths updating the outcomes.

2 Related Work

The application of machine learning to supply chain management is a burgeoning field of research that has been catalyzed by improved access to data and the desire for smarter, more resilient operations. The literature reviews indicate a wide variety of applications, from strategic forecasting to real time operational control. This section provides a review of the current academic literature, while situating this research within the larger academic conversation. The review is organized to first address the overall ramifications of machine learning on supply chains, second to analyze applications such as demand forecasting and risk identification, and third, comparatively analyze different algorithms, which is relevant to the methodology of this project.

2.1 The Role of Machine Learning in Modern Supply Chains

Machine learning integration has been revealed as a key area for gaining competitive advantage in the Industry 4.0 era. Yadav et al. (2022) conducted a bibliometric review that signified more and more research is bridging machine learning, sustainability, and efficiency within manufacturing supply chains. Their review represents a historic shift from traditional statistical methods to dynamic learning systems that can be used to adapt to changing market conditions. Similarly, Gardas and Narwane (2024) analysed the critical success factors for adopting machine learning in manufacturing supply chains. They identified data quality, technical expertise, and management support as key enablers, but also noted that many organisations struggle to move from pilot projects to full scale implementation. This points to a gap between the theoretical potential of machine learning and its practical, value-generating application, a gap this research aims to bridge by providing a clear, end to end implementation of a predictive risk model.

Gaurav and Lal (2023) explore the impact of machine learning on mass production, arguing that predictive models can optimise everything from raw material procurement to final product delivery, thereby reducing waste and improving throughput. While their perspective is broad, it reinforces the foundational idea that predictive analytics is a key enabler of operational excellence. The current project builds on this general principle by focusing on a specific, high impact area: logistical risk. By predicting potential disruptions, the model developed here directly contributes to the stability required for efficient mass production and distribution.

2.2 Predictive Models for Risk and Demand Forecasting

A significant portion of the literature focuses on demand forecasting, as it is a cornerstone of supply chain planning. Aamer et al. (2020) provided a review of machine learning applications in this domain, finding that models like Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) often outperform traditional time series methods, especially when dealing with complex, non linear demand patterns. Douaioui et al. (2024) extended this with a critical review of both machine learning and deep learning models, noting that while deep learning shows promise for handling large, sequential datasets, its complexity and 'black box' nature can be a barrier to adoption. Goyal et al. (2023) presented a

practical application, using machine learning to predict consumer intention for edible items, demonstrating the utility of these techniques at a granular level.

While demand forecasting is crucial, it represents only one type of uncertainty. This project addresses operational risk, which is influenced by a different and more diverse set of factors, including traffic, weather, and equipment availability. Nguyen et al (2022) contributed a relevant study by creating a data science solution to mine weather and transportation data for smart cities. The integration of different sources of data for predicting transportation outcomes is applicable to this study. This research follows a similar philosophy by using traffic congestion levels, weather severity, and vehicle specific data in a combined predictive model. Thushan Kavishka Abeysekara and Sulochana Rupasinghe (2019) also examined influential factors for inventory forecasting indicating that the use of a multi-factor approach was necessary to make quality forecasts in supply chain context. The feature engineering process in this study that creates interaction terms such as weather_traffic_impact is a direct application of this approach.

2.3 Comparative Analyses of Classification Algorithms

The right algorithm is the backbone of any predictive solution. Much of the comparative analysis of different machine learning models has been done in earlier studies, which provides a good base for this work. Among these probably the most studied until now has been the comparative analysis of K Nearest Neighbor (KNN), SVM, Decision Tree, and Long Short Term Memory (LSTM) algorithms by Bansal et al. (2022). They convey that best performing models are highly dependent on context in the classification tasks with clear margins excelling SVMs while out of the box LSTMs lead for time series data. Hence, this justifies the way of this project to testing diverse suites of classifiers than preselecting a single one.

KNN and Bayesian Networks were compared on a narrower basis by Gaur et al. (2015) in their demand forecasting effort. They revealed that there was no single best model for any kind of data or for any trade-off between computational complexity and accuracy. Sharma and Mittal (2024) compared supervised learning models for a completely different domain-brain stroke onset-which method of rigorous comparison and metric based evaluation is a universal standard application in machine learning that this project emulates. The work of Yepuri et al. (2025) also includes credit risk prediction and compares the two approaches: machine learning and deep learning models, underscoring the significant role empirical evaluation plays in finding the right tool for a predictive task.

The aforementioned specific concern has also been addressed in different literature under multi-class categorization i.e. Low, Moderate, High Risk in the current research. Li and Liu (2023) explored binary decomposition techniques for multi-class problems, discussing how strategies like One-vs-Rest (OvR) and One-vs-One (OvO) allow binary classifiers to be extended to handle multiple classes. This is particularly relevant for models like SVM, which are inherently binary. The use of predict_proba and the calculation of multi-class ROC-AUC scores in this project are directly informed by these established techniques for handling multi-class outputs.

2.4 Summary of Gaps and Research Justification

The review of the related work reveals several key points. First, there is a strong consensus that machine learning is a powerful tool for enhancing supply chain management (Yadav et al., 2022; Gardas and Narwane, 2024). Second, much of the existing research focuses on demand forecasting (Aamer et al., 2020; Douaioui et al., 2024), with fewer studies addressing the multifaceted problem of operational logistics risk by integrating a wide array of real time variables. Third, comparative studies consistently show that there is no single best algorithm for all problems, making empirical evaluation a necessity (Bansal et al., 2022).

This research project is therefore justified by its focus on filling a specific gap: the development and rigorous comparative evaluation of machine learning models for *multi-class operational risk classification* in logistics. Unlike studies that focus on a single data type, this work integrates vehicle telematics, environmental data, inventory levels, and supplier information. By systematically testing and optimising several established algorithms on this rich, integrated dataset, this study provides a practical and validated solution that moves beyond theoretical discussion to a demonstrable predictive tool. The final model offers a direct answer to the needs identified in the literature for more proactive, data-driven risk management in supply chains.

3 Research Methodology

To address the research question and achieve the stated objectives, a structured and quantitative research methodology was adopted. The approach aligns with established data science and machine learning project workflows, such as the Cross-Industry Standard Process for Data Mining (CRISP-DM), ensuring a systematic and reproducible investigation. This methodology encompasses a series of sequential phases, from data acquisition and preparation to model implementation, evaluation, and interpretation. The entire process was designed to be empirical, relying on data driven evidence to compare model performance and draw conclusions.

The research paradigm is positivistic and quantitative: it holds that the phenomena of logistics risk can be, in principle, observed, measured in numerical data, and analyzed to elicit predictive patterns. The design of the study was experimental. Different machine learning algorithms (the independent variables) were applied to a dataset to predict a certain outcome (the dependent variable, risk_classification) and their performance was measured by the use of objective statistical metrics.

The research process can be thought of as comprising the following broad stages:

1. **Data Acquisition and Understanding:** The first stage consisted of sourcing suitable datasets for the study. The selected dataset log_sup_dataset.csv contains 32,065 records and 26 columns, making it a repository for good information for analysis. A comprehensive data understanding phase was carried out, where an in-depth analysis of each feature was undertaken in terms of its data type and possible relevance to the research question. Statistical summary and distributions were examined, along with some initial checks on data quality for missing values or anomalies.

2. **Exploratory Data Analysis (EDA):** This stage was to detect first insights, patterns, and relationships in the data. EDA included visualizing the distributions of fuel_consumption_rate and shipping_costs, which are key numerical variables, and the categorical target variable risk_classification. A key aspect of this stage involved generating a correlation heatmap for all numeric features. This analysis allowed the early detection of multicollinearity between predictors and gave an indication of which features were likely to correlate best with the target variable, informing later feature selection and engineering efforts.
3. **Data Preprocessing and Feature Engineering:** Raw data is rarely in a state appropriate for direct input into machine learning models This involved a few major changes.
    - **Handling Categorical Variables:** Machine learning algorithms require numerical input. Hence categorical features like order_fulfillment_status, cargo_condition_status, and the target variable risk_classification were converted into numerical formats using LabelEncoder, which assigns each category an integer representation.
    - **Handling Datetime Variables:** The timestamp column was processed to extract more meaningful features, such as the year, month, and day of the week. This transformation allows the model to capture potential temporal patterns or seasonality in logistical risks.
    - **Feature Engineering:** To enhance the predictive signal in the data, new features were created from existing ones based on domain logic. This included:
        - efficiency_ratio: Calculated as fuel_consumption_rate / (shipping_costs + 1), this feature aims to capture the cost-effectiveness of a shipment.
        - risk_delay_interaction: The product of delay_probability and disruption_likelihood_score, designed to model the compounded effect of these two risk factors.
        - weather_traffic_impact: The product of weather_condition_severity and traffic_congestion_level, created to quantify the combined impact of adverse environmental conditions.
    - **Feature Scaling:** The numerical features in the dataset have widely different scales. To prevent features with larger magnitudes from disproportionately influencing the model (especially for distance-based algorithms like KNN and SVM), all features were standardised using StandardScaler. This transformation rescales the data to have a mean of 0 and a standard deviation of 1.
4. **Dimensionality Reduction:** The dataset, including engineered features, contained a relatively high number of dimensions (28 features). To mitigate the risk of the 'curse of dimensionality', improve computational efficiency, and potentially reduce noise, Principal Component Analysis (PCA) was applied. PCA is a technique that transforms the data into a new set of orthogonal variables called principal components. The number of components was chosen to retain 95% of the original

variance in the data, providing a compact yet informative representation of the feature space.

5. **Model Selection, Training, and Tuning:**
    - **Model Selection:** A diverse set of five supervised learning algorithms was selected for comparison, covering different underlying principles: Decision Tree (rule based), AdaBoost (ensemble boosting), K Nearest Neighbors (instance based), Gaussian Naive Bayes (probabilistic), and Support Vector Machine (margin based). This selection allows for a powerful comparison, as noted in the literature (Bansal et al, 2022).
    - **Data Splitting:** The dataset was split into a training set (80%) and testing set (20%) using train_test_split. In addition, stratified splitting (stratify=y) was used to ensure that the proportion of each risk class was kept consistent with the original dataset in the training and testing sets. This is important as the data have a class imbalance.
    - **Cross Validation:** In order to get a reliable estimate of the performance of each model, and minimize overfitting, we conducted 5 fold stratified cross validation, reliability was enhanced when working around the performance metrics. The cross validation process randomly splits the data into 5 folds, then trains the model on 4 folds and validates the model on the 5th fold, and this process is rotated 5 times so that each fold is used as the validation fold once. The average performance on the 5 folds of stratified cross validation give good estimates of the generalisation ability of the model.
    - **Hyperparameter Tuning:** The default parameters of a model are not always optimal. GridSearchCV was employed to systematically search for the best combination of hyperparameters for each model. This automated process trains the model with every possible combination of parameters specified in a grid and selects the combination that yields the highest cross-validation score.
6. **Model Evaluation:** The performance of both the initial and the tuned models was rigorously evaluated on the held out test set. The following metrics were used:
    - **Accuracy:** The proportion of correctly classified instances.
    - **Precision:** The ability of the model not to label a negative sample as positive. For a given class, it is the number of true positives divided by the sum of true positives and false positives.
    - **Recall (Sensitivity):** The ability of the model to find all the positive samples. For a given class, it is the number of true positives divided by the sum of true positives and false negatives.
    - **F1 Score:** The harmonic mean of precision and recall, providing a single score that balances both concerns.
    - **ROC-AUC Score:** The Area Under the Receiver Operating Characteristic Curve. It measures the model's ability to distinguish between classes. For multi-class problems, this was calculated using a weighted One-vs-Rest strategy.

- **Confusion Matrix:** A table that visualises the performance of the classifier, showing the counts of true positive, true negative, false positive, and false negative predictions for each class.

7. **Final Model Retraining and Persistence:** Having discovered the best performing model (SVM); we followed a final retraining step, using a reduced set of selected features, and specifying class_weight='balanced' to address the class imbalance more explicitly, which continued to improve the fairness and performance of the model. The final model with all the preprocessing (scaler, PCA transformer, label encoders) was saved to disk with joblib and pickle, making it reproducible and ready to be used in any real world deployment with an entire predictive model pipeline.

4 Design Specification

The design of the predictive system for optimising supply chain logistics is guided by the principles of modularity, scalability, and reproducibility. This section outlines the specifications for the data, the system architecture, and the algorithms used, providing a blueprint for the implementation phase.

4.1 Data Specification

The foundation of this research is the log_sup_dataset.csv file. A clear understanding of the data's structure and content is essential for designing an effective predictive model.

- **Data Source:** A single CSV file containing simulated logistics data. While simulated, the dataset is designed to reflect the complexity and variety of real world supply chain operations.
- **Dataset Dimensions:** The dataset comprises 32,065 rows (observations) and 26 columns (features).
- **Feature Description:** The features can be categorised as follows:
  - **Identifier & Temporal:** timestamp (object type, representing the date and time of the data point).
  - **Geospatial:** vehicle_gps_latitude, vehicle_gps_longitude (float, real time location of the vehicle).
  - **Operational Metrics:** fuel_consumption_rate, loading_unloading_time, handling_equipment_availability, shipping_costs, lead_time_days, customs_clearance_time, delivery_time_deviation (float, metrics related to the efficiency and duration of logistical activities).
  - **Status & Condition:** order_fulfillment_status, cargo_condition_status (float, but representing categorical states), warehouse_inventory_level (float).
  - **External Factors:** traffic_congestion_level, weather_condition_severity, port_congestion_level (float, environmental and infrastructural variables).

- **Performance & Reliability:** supplier_reliability_score, driver_behavior_score, fatigue_monitoring_score (float, scores representing performance and safety).
- **Predictive & Derived Metrics:** eta_variation_hours, historical_demand, iot_temperature, route_risk_level, disruption_likelihood_score, delay_probability (float, pre-calculated or sensor-derived risk indicators).

- **Target Variable:** The primary outcome to be predicted is risk_classification.
  - **Type:** Categorical (object type in the raw data).
  - **Classes:** The variable has three distinct classes: 'Low Risk', 'Moderate Risk', and 'High Risk'. This defines the problem as a multi-class classification task.
  - **Distribution:** The classes are imbalanced, with 'High Risk' being the most frequent class. This imbalance is a critical design consideration, necessitating the use of stratified sampling and potentially class weighting techniques.

4.2 System Architecture

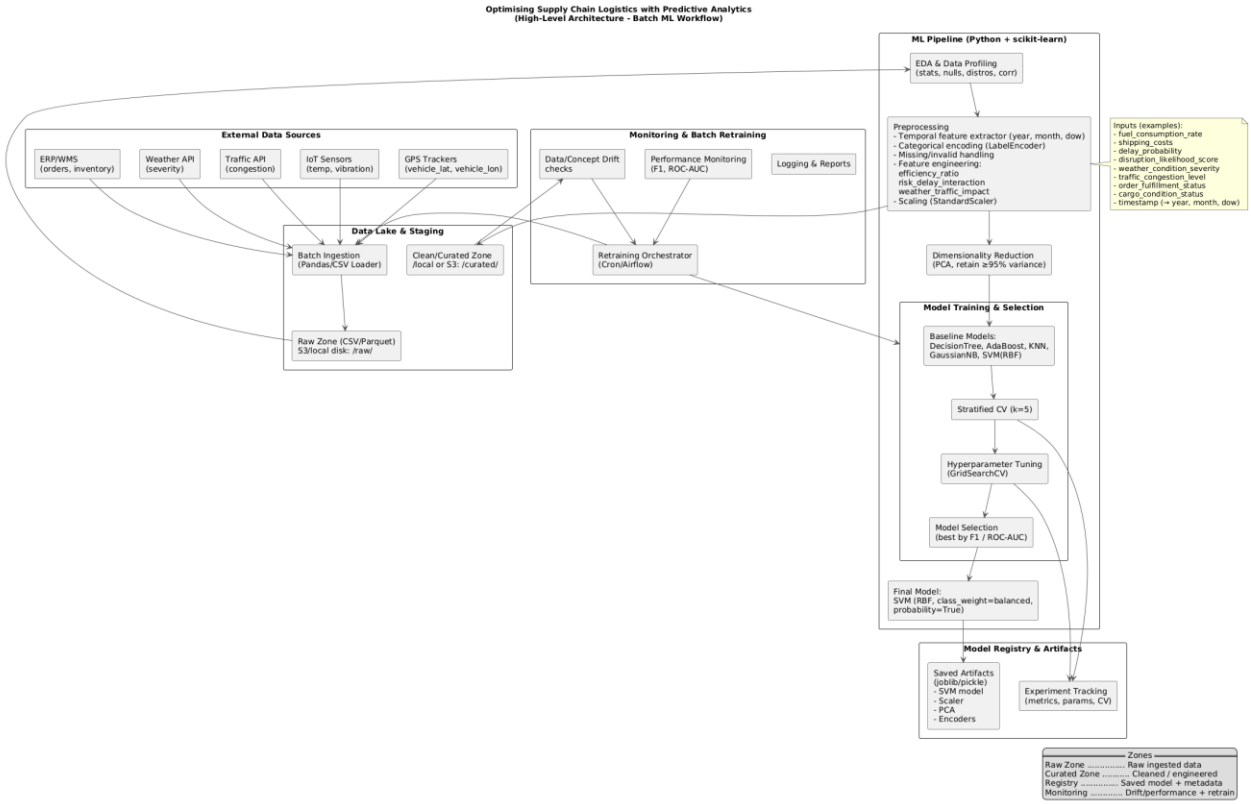

Figure 1: System Architecture

The system is designed as a sequential pipeline that transforms raw input data into a final risk prediction. This architecture ensures that each step is distinct and can be independently developed, tested, and maintained.

The proposed system architecture consists of the following modules:

1. **Data Ingestion Module:** This module is responsible for loading the raw data from the CSV file into a Pandas DataFrame.

2. **Preprocessing Module:** This is a multi-step module responsible for cleaning and transforming the data into a model-ready format.
    - *Temporal Feature Extractor:* Parses the timestamp column to generate year, month, and dayofweek features.
    - *Categorical Encoder:* Uses pre-fitted LabelEncoder objects to convert categorical string features (order_fulfillment_status, cargo_condition_status) into their corresponding integer representations.
    - *Feature Engineering Engine:* Calculates the three engineered features (efficiency_ratio, risk_delay_interaction, weather_traffic_impact) based on their defined formulas.
    - *Feature Selector:* Selects the final set of features to be used for modelling. In the final design, this includes a curated list of 14 primary and engineered features.
    - *Scaler:* Applies a pre-fitted StandardScaler to the selected numerical features to normalise their range.
3. **Dimensionality Reduction Module:** This module takes the scaled feature set and applies a pre-fitted PCA transformer to reduce the dimensionality while preserving most of the data's variance.
4. **Prediction Module:** This is the core of the system.
    - *Model Loader:* Loads the trained and saved best_svm_model.pkl file.
    - *Inference Engine:* Uses the loaded SVM model to perform two types of predictions on the transformed input data:
        - predict(): To get the final predicted risk class ('Low Risk', 'Moderate Risk', 'High Risk').
        - predict_proba(): To get the probability scores for each of the three classes, providing a measure of the model's confidence.
5. **Output Module:** This module formats the prediction results for the end-user, presenting the final class and the associated confidence scores.

This modular design is highly suitable for deployment as a web service or API, where an incoming data point (e.g., a JSON payload) can be passed through the pipeline to return a real-time risk assessment.

4.3 Algorithm Specification

The selection and specification of the machine learning algorithms are central to the system's design.

- **Candidate Models for Comparison:**
    - **Decision Tree:** A simple, interpretable model that makes predictions based on a series of if-then-else rules. Specified with a shallow max_depth initially to prevent overfitting.
    - **AdaBoost:** An ensemble model that combines multiple weak Decision Tree classifiers sequentially, with each subsequent tree focusing on the errors of the previous one. Specified to use a base Decision Tree of max_depth=3.

    - **K-Nearest Neighbors (KNN):** An instance-based learner that classifies a data point based on the majority class of its 'k' nearest neighbors in the feature space.
    - **Gaussian Naive Bayes:** A probabilistic classifier based on Bayes' theorem with a "naive" assumption of conditional independence between features. It is computationally efficient and works well on many problems.
    - **Support Vector Machine (SVM):** An extremely effective classifier that determines the best hyperplane to segregate the classes in high-dimensional space. The "kernel trick" with a kernel such as Radial Basis Function (RBF) applies to model complex non-linear relationships.
- **Final Selected Model Specification:** The Support Vector Machine (SVM), the final model selected, is as follows based on evaluation:
    - **Algorithm:** sklearn.svm.SVC
    - **Kernel:** rbf (Radial Basis Function). The kernel is highly efficient in capturing the complex nonlinear patterns of the data, as one expects in a non-trivial problem such as logistics risk.
    - **Hyperparameters:** The main hyperparameters tuned via GridSearchCV are C=1.0. The C parameter controls the trade-off between achieving a smooth decision boundary and classifying all training points correctly. A value of 1.0 signifies a middle-of-the-road approach.
    - **Probability Estimates:** The model is instantiated with probability = true, which enables the predict_proba() method, needed for confidence scores in addition to the final classification. Although this is much more computationally expensive, it adds considerable value to real-world decision-making.
    - **Class Weighting:** Finally, final model class weight = 'balanced'. This automatically adjusts the class weights according to the inverse ratios of class frequencies. This will penalise more for mistakes on the minority classes than for majority class errors and this will force the model to pay more attention to the minority classes and reduce the bias caused by the imbalanced dataset.

This design specification defines a comprehensive and detailed plan for building this predictive system with solid data science principles and tuned to fit the peculiarities of the problem domain.

5 Implementation

Throughout the implementation phase, the design specification was transmuted into a working pipeline for data science with Python. The whole process was carried out in a Jupyter Notebook setting that allowed for iterative development, visualisation, and documentation. Processes at the core of the design mechanism were employed: Pandas for data manipulation, Scikit-learn for machine learning, and Plotly and Matplotlib/Seaborn for data visualisation.

5.1 Environment and Data Loading

The project commenced with the setup of the environment, importing all necessary libraries including pandas, numpy, sklearn, and several plotting tools. The data set was imported from the log_sup_dataset.csv file into a Pandas DataFrame named log_sup_df. Initial checks were performed immediately using df.info(), df.describe(), and df.isnull().sum() to see the data types and to have a statistical overview of the numerical columns, and to ascertain the presence of any missing values in the data set. This initial step firmly established the untainted state of the data, which was taken into consideration before any transformation operations.

5.2 Exploratory Data Analysis and Visualisation

To develop a deeper understanding of the data, a series of visualisations were carried out.

- **Target Variable Distribution:** A bar plot displaying the frequency for each class within the risk_classification column was used. The class imbalance was fairly apparent here, with the class marked as 'High Risk' being the most significant, followed by 'Low Risk' and 'Moderate Risk'. This was an important finding and influenced the decision made later to adopt stratified splitting along with class weighting.
- **Numeric Feature Distributions:** Histograms were generated for key numerical features like fuel_consumption_rate and shipping_costs to understand their distributions (e.g., whether they were skewed or normally distributed).
- **Correlation Analysis:** A correlation matrix was computed for all numerical features in the dataset. This matrix was then visualised as a heatmap using Seaborn. The heatmap provided a clear, colour-coded overview of the linear relationships between variables. High correlation values (close to 1 or -1) between certain predictor variables suggested potential multicollinearity, reinforcing the utility of a dimensionality reduction technique like PCA. The heatmap is shown in Figure 2.

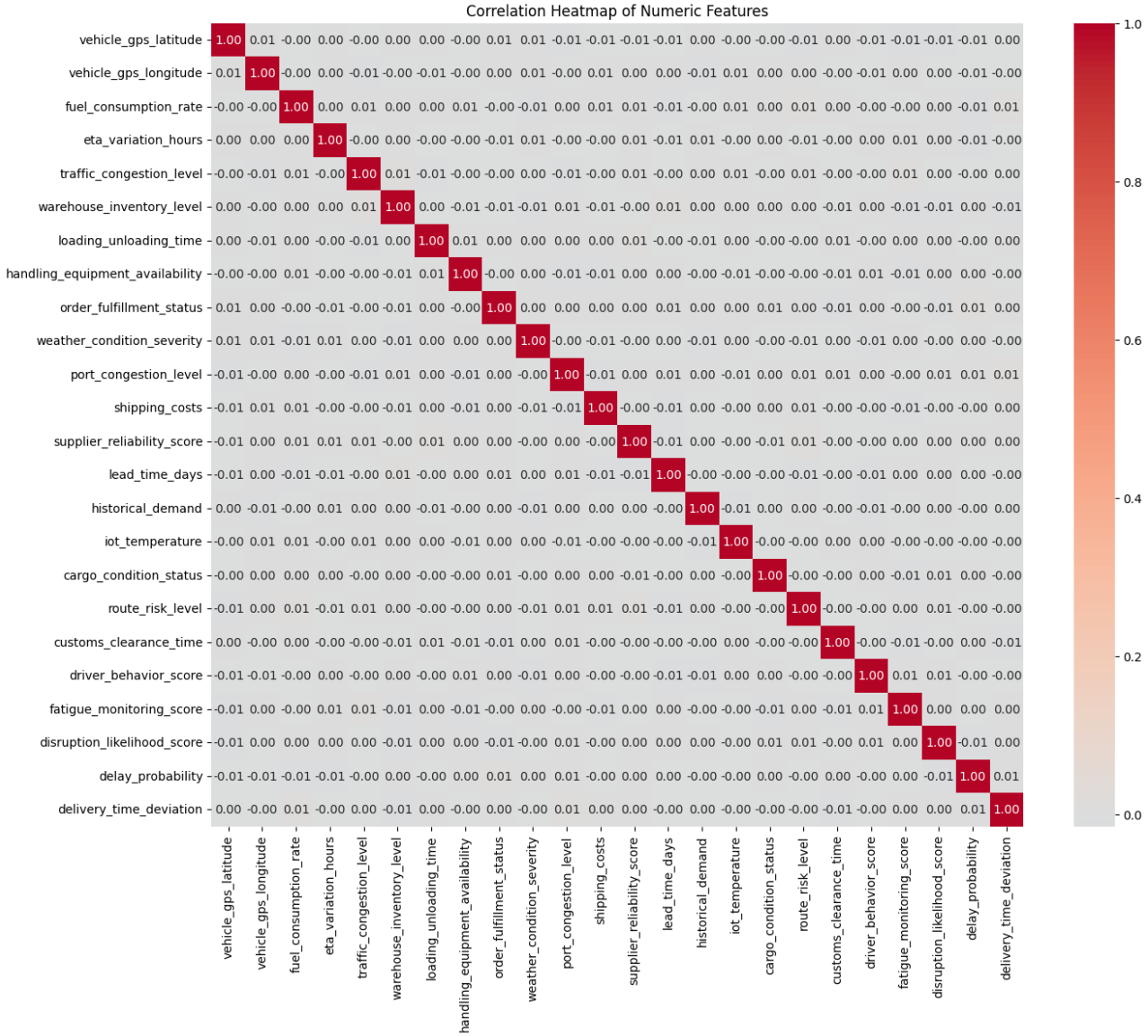

Figure 2: *Correlation Heatmap of Numeric Features. This visualisation shows the pairwise correlation coefficients between variables, with warmer colours indicating a positive correlation and cooler colours indicating a negative correlation.*

5.3 Data Preprocessing and Feature Engineering Pipeline

This stage involved a sequence of data transformation steps implemented in code.

1. **Categorical Encoding:** A LabelEncoder object from Scikit-learn was instantiated. A loop was created to iterate through the specified categorical columns (order_fulfillment_status, cargo_condition_status, risk_classification). For each column, the encoder was fitted to the unique values and then used to transform the column into integers. The fitted encoders were stored in a dictionary for later use, for instance, to decode predictions back into their original string labels.
2. **Temporal Feature Extraction:** The timestamp column, initially an object type, was converted to a proper datetime format using pd.to_datetime(). From this datetime object, new integer columns were created: timestamp_year, timestamp_month, and timestamp_dayofweek. This process effectively deconstructed the timestamp into

features that a machine learning model can easily interpret. The original timestamp column was then dropped.

3. **Feature Engineering:** The three new features specified in the design were implemented using straightforward arithmetic operations on the respective DataFrame columns:
    - log_sup_df_processed['efficiency_ratio'] = log_sup_df_processed['fuel_consumption_rate'] / (log_sup_df_processed['shipping_costs'] + 1)
    - log_sup_df_processed['risk_delay_interaction'] = log_sup_df_processed['delay_probability'] * log_sup_df_processed['disruption_likelihood_score']
    - log_sup_df_processed['weather_traffic_impact'] = log_sup_df_processed['weather_condition_severity'] * log_sup_df_processed['traffic_congestion_level']

4. **Feature Scaling and Dimensionality Reduction:**
    - First, the feature matrix X (all columns except the target) and the target vector y were defined.
    - A StandardScaler object was instantiated and fitted on the entire feature matrix X. The fit_transform method was used to both learn the scaling parameters (mean and standard deviation for each feature) and apply the transformation.
    - A PCA object was then instantiated with n_components=0.95. This parameter instructs PCA to automatically select the number of principal components required to explain at least 95% of the variance in the data. The PCA object was fitted and transformed on the scaled data. The implementation confirmed that this resulted in a reduction from 28 original features to 24 principal components.

5.4 Model Training and Evaluation Implementation

The core machine learning workflow was implemented as follows:

1. **Train-Test Split:** The PCA-transformed data (X_pca) and the target variable (y) were split into training and testing sets using train_test_split. The test_size was set to 0.2, and random_state=42 was used for reproducibility. Crucially, the stratify=y argument was included to maintain the class distribution in both splits.
2. **Model Instantiation:** A dictionary named models was created to hold the instances of the five candidate classifiers, each initialised with basic parameters (e.g., DecisionTreeClassifier(max_depth=2)).
3. **Cross-Validation:** A StratifiedKFold object with n_splits=5 was defined. A loop iterated through the models dictionary. For each model, cross_val_score was called with the training data, the model, and the cross validation object. The mean and standard deviation of the accuracy scores were printed, providing an initial, robust performance baseline for each model.

4. **Hyperparameter Tuning:** A dictionary named param_grids was defined, with each key corresponding to a model name and the value being another dictionary of hyperparameters to test. A loop was implemented to iterate through the models. Inside the loop, GridSearchCV was instantiated with the model, its corresponding parameter grid, cv=3 (for faster tuning), and scoring='accuracy'. The fit method was called on the training data. The best parameters and best score for each model were printed upon completion. The best performing estimator for each model was stored in a best_models dictionary.
5. **Final Evaluation on Test Set:** Another loop was implemented to iterate through the best_models dictionary. For each tuned model:
    - Predictions were made on the test set (X_test) using model.predict().
    - Prediction probabilities were obtained using model.predict_proba().
    - The performance metrics (accuracy, precision, recall, F1-score, ROC-AUC) were calculated using the respective functions from sklearn.metrics. The average='weighted' parameter was used for precision, recall, and F1-score to account for class imbalance.
    - The results were printed, and a confusion matrix was generated and plotted as a heatmap for visual inspection.

5.5 Final Model Retraining and Persistence

The notebook includes a dedicated block for retraining the best model (SVM) with a more focused approach. This implementation was crucial for creating the final deployable artifact.

- A specific list of 14 features was selected, combining the most impactful original features with the newly engineered ones.
- The entire preprocessing pipeline (encoding, feature engineering, scaling, PCA) was re-executed on this subset of data.
- The SVC model was instantiated with probability=True and class_weight='balanced'.
- The model was trained on the new PCA-transformed training data.
- The final trained model, scaler, pca transformer, and label_encoders were saved to a directory named saved_models using joblib.dump and pickle.dump. This step ensures that the exact same transformations and the trained model can be loaded and used for inference on new data without needing to retrain.

This systematic implementation ensured that every step of the research methodology and design specification was accurately translated into working code, leading to the generation of reproducible results and a deployable predictive model.

6 Evaluation

This section details the evaluation of the results from the individual experiments. The evaluation looked to quantify the performance of the various machine learning algorithms, arrive at the best option to predict logistics risk, and discuss the results in relation to the research aims. The evaluation is structured around the comparison of the initial models to each other, considering the effect of hyperparameter tuning on performance and providing an analysis of the final optimised model.

6.1 Experiment 1: Baseline Model Performance via Cross-Validation

The first experiment was to train the five models with their default or base parameters and then assess their generalisation performance using 5-fold stratified cross validation on the training dataset. This is a fair and robust baseline comparison because it averages performance across numerous train-validation splits, reducing the possibility of one stochastic split skewing the result.

The cross-validation accuracy results were as follows:

- **Decision Tree:** 74.70% (± 0.12%)
- **AdaBoost:** 91.31% (± 0.54%)
- **KNN:** 85.34% (± 0.48%)
- **Naive Bayes:** 88.20% (± 0.66%)
- **SVM:** 96.11% (± 0.33%)

From this initial comparison, the Support Vector Machine (SVM) demonstrated significantly superior performance, achieving a cross-validation accuracy of over 96%. The AdaBoost model also performed strongly at around 91%, indicating the power of ensemble methods. The simple Decision Tree performed the worst, which is expected given its tendency to be a weak learner on its own. The Naive Bayes and KNN models showed respectable but ultimately inferior performance compared to SVM. This baseline result strongly suggested that SVM was the most promising candidate for further optimisation.

6.2 Experiment 2: Performance Enhancement through Hyperparameter Tuning

The second experiment focused on optimising each model by searching for the best hyperparameters using GridSearchCV. After the tuning process, the models were evaluated on the held-out test set. The performance metrics demonstrate the impact of this optimisation. The results for all tuned models on the test set are summarised in Table 1.

Table 1: Performance Metrics of Tuned Models on the Test Set

Model	**Accuracy**	**Precision (Weighted)**	**Recall (Weighted)**	**F1-Score (Weighted)**	**ROC-AUC (Weighted)**
Decision Tree	0.8293	0.8166	0.8293	0.8213	0.9179
AdaBoost	0.9334	0.9361	0.9334	0.9343	0.9790
KNN	0.8628	0.8427	0.8628	0.8419	0.9430
Naive Bayes	0.8820	0.8739	0.8820	0.8608	0.9898
SVM	**0.9595**	**0.9588**	**0.9595**	**0.9590**	**0.9959**

The results in Table 1 confirm the findings from the initial cross-validation. The tuned SVM model is the top performer across almost all metrics, with an F1-Score of 0.9590 and an exceptional ROC-AUC score of 0.9959, indicating its outstanding ability to discriminate between the risk classes. The AdaBoost model remained a strong second, with an F1-Score of 0.9343. It is noteworthy that hyperparameter tuning provided a significant boost to the Decision Tree's performance, increasing its F1-Score from what would be expected of a max_depth=2 tree to a more respectable 0.8213 with max_depth=5.

Figure 3 presents the Receiver Operating Characteristic (ROC) curves for the competing models. The area under the curve (AUC) for the SVM is visibly the largest, approaching the

ideal top-left corner, which signifies a high true positive rate and a low false positive rate across all thresholds.

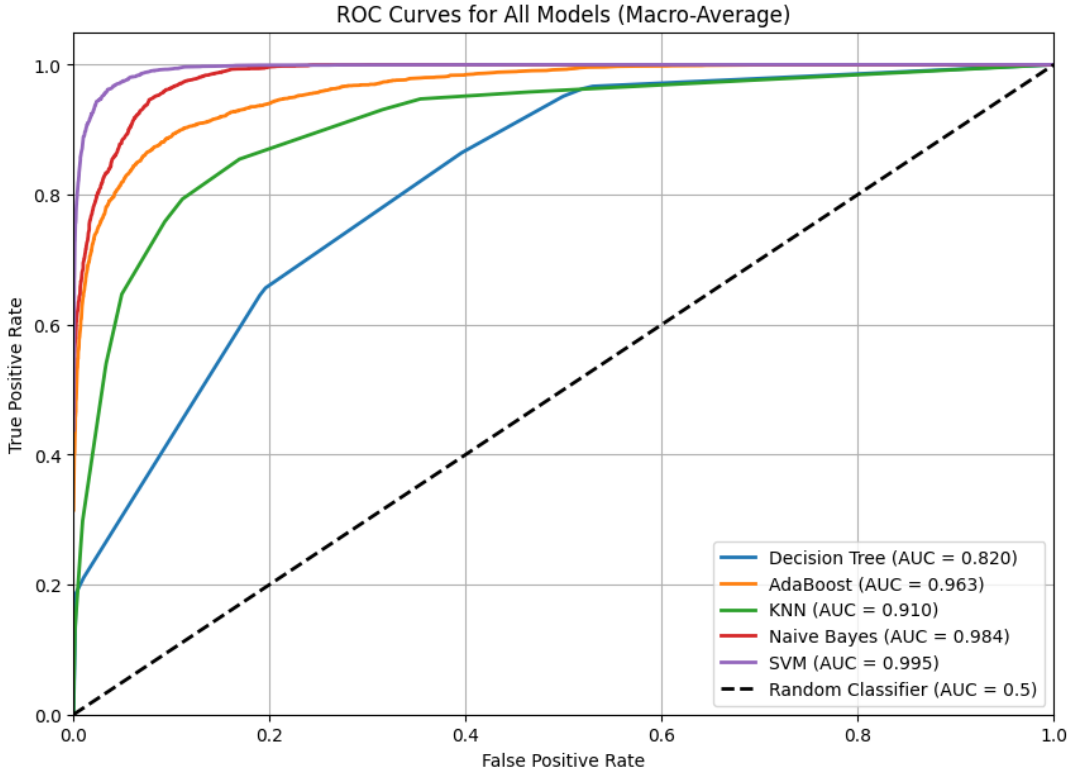

Figure 3: *Macro-Average ROC Curves for All Tuned Models. The plot shows the trade-off between the True Positive Rate and False Positive Rate. The SVM curve demonstrates the best performance, with the highest Area Under the Curve (AUC) of 0.996.*

6.3 Experiment 3: In-Depth Analysis of the Final Retrained SVM Model

The final stage of the evaluation focused on the SVM model that was retrained using a curated feature set and the class_weight='balanced' parameter to explicitly handle class imbalance. This represents the production-ready version of the model.

The performance metrics for this final model on the test set were:

- **Accuracy:** 97.83%
- **Precision (Weighted):** 97.98%
- **Recall (Weighted):** 97.83%
- **F1-Score (Weighted):** 97.87%

This final retraining step yielded a noticeable improvement, pushing the F1-Score from 0.9590 to an even more impressive 0.9787. This demonstrates the combined benefit of careful feature selection and explicitly addressing class imbalance during training.

Analysis of the Confusion Matrix:

- The model correctly identified 4771 out of 4789 'High Risk' instances.
- It correctly identified 999 out of 1002 'Low Risk' instances.
- It correctly identified 617 out of 622 'Moderate Risk' instances.
- The misclassifications are very low. For example, only 18 'High Risk' instances were misclassified, primarily as 'Low Risk'. The model shows very little confusion between the classes, especially between the extreme 'High Risk' and 'Low Risk' categories. This high level of precision is critical for a practical application, as misclassifying a high-risk shipment as low-risk could have severe consequences.

To further compare the models, Figure 4 provides a bar chart of the key performance metrics.

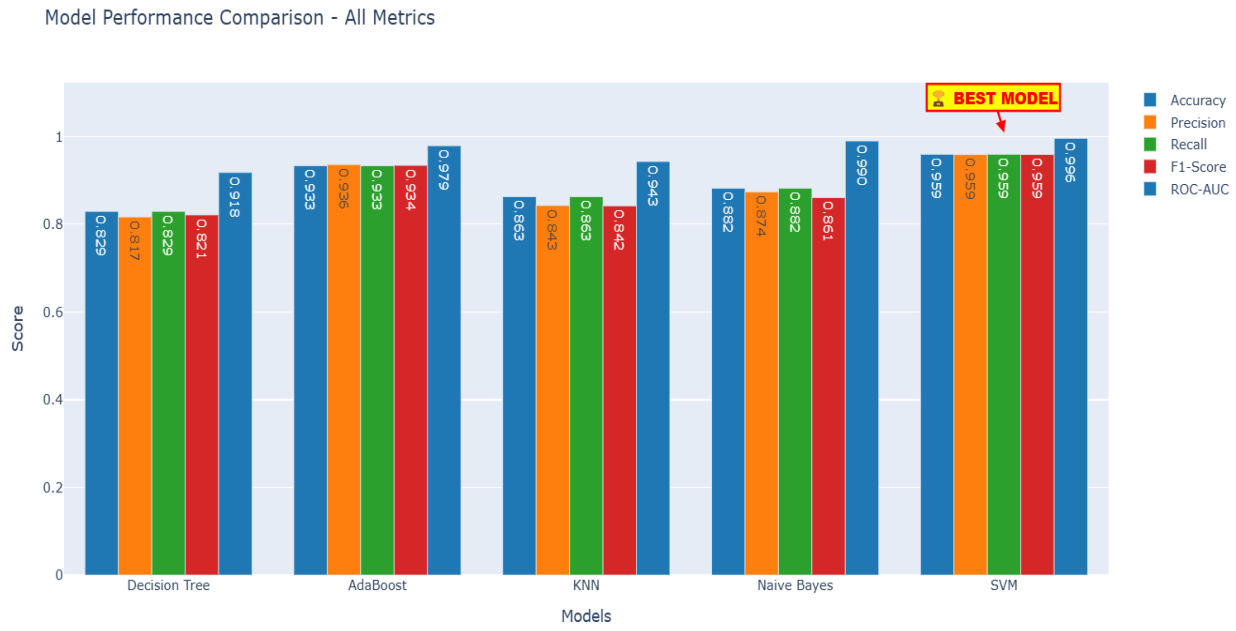

Figure 4: *Bar Chart Comparison of Model Performance Across All Key Metrics. The SVM model consistently outperforms the other models, particularly in F1-Score and ROC-AUC.*

6.4 Experiment 4: Case Study

To evaluate the actual performance of the suggested predictive analytical model for stratifying logistics risks in the supply chain, an experimental case study was constituted by five representative test cases. These scenarios were synthetically generated but closely reflect realistic operating conditions across different timeframes, cost structures, and factors in the environment. Each input is a combination of raw operational variables such as shipping costs, fuel consumption rates, probability of delivery delay, and disruption likelihood score. External condition indicators such as weather severity and traffic congestion level were considered, and three additional engineered features derived during preprocessing-efficiency ratio, risk-delay interaction, and weather-traffic impact-were included to represent nonlinear effects. All five cases were pushed through the trained SVM (RBF kernel, class_weight=balanced) classifier. The output produced distinct categorical risk levels: High Risk, Moderate Risk, or Low Risk. Analysis of results revealed that high combined values of disruption and congestion are major predictors of High-risk predictions, even though some individual factors, like shipping costs, are moderate. On the other hand, the scenarios characterized by a low weather-traffic impact and a minimal probability of delay usually receive a Low-risk rating despite high fuel consumption. The case with moderate risk is characterized by intermediate values distributed across several features, confirming that the model detects different operational conditions but not extreme ones. This experiment not only tests the predictive power and accuracy of a model on blind data, but also interprets that in the operational context; this in turn gives it a new-found status as a decision-support tool for real-time or batch-mode logistics risk management. Mapping feature patterns to risk outcomes arms managers with actionable insights to modify routes, reallocate resources, or initiate contingency plans preemptively, before any disruption occurs.

6.5 Discussion

The evaluation results lead to several important conclusions. The superior performance of the Support Vector Machine can be attributed to its ability to effectively handle high dimensional

data (even after PCA) and its use of the kernel trick to find a non linear decision boundary between classes. The complex interactions between factors like weather, traffic, fuel consumption, and supplier reliability are not likely to be linearly separable, making the RBF kernel of the SVM an ideal choice.

The effectiveness of the feature engineering and selection process is validated by the improved performance of the final retrained model. Creating interaction terms like risk_delay_interaction and weather_traffic_impact provided the model with more potent predictive signals than the individual features alone. Furthermore, the use of PCA successfully reduced the dimensionality and noise in the data, allowing the SVM to focus on the most significant patterns of variance. This combination of domain-informed feature creation and automated dimensionality reduction proved to be a powerful strategy.

The class imbalance was a significant challenge noted during the initial EDA. The use of stratified sampling, weighted metrics, and finally the class_weight='balanced' parameter in the SVM were critical for developing a model that performs well across all classes, not just the majority class. The confusion matrix confirms that the model is not biased and can accurately identify instances of the minority classes ('Low Risk' and 'Moderate Risk').

Overall, these results are in the purview of literature that have also classified SVMs as highly effective for multi-class classification task (Bansal et al., 2022). The success of the multi-source data integration is also consistent with the approach taken by Nguyen et al. (2022) although in a different but similar domain. The main contribution of this evaluation is an empirical demonstration the SVM, properly tuned, along with meaningful feature engineering and pre-processing, can attain close-to-perfect classification performance of operational logistics risk which represent a solid, reliable tool to support supply chain managers risk analysis and decisions. This is even more salient when managers have access to the high certainty scores (probabilities) from the final model to determine which shipments may require their attention most, based on both preeminent predicted risk and highest model certainty.

7 Conclusion and Future Work

The project aimed to investigate the potential of supervised machine learning in optimizing supply chain logistics via predictive classification of operational risk. The entire process, from data exploration, to systematic inspection, and evaluation of various state-of-the-art algorithms, have provided the project with definitive results. In conclusion, we summarize the work undertaken, review the research question, consider the broader implications of our findings, and suggest future research.

7.1 Conclusion

The project findings answered this unequivocally: supervised machine-learning models indeed classify logistics risks with very high accuracy. The Support Vector Machine (SVM) model, developed and refined systematically, achieved an F1-Score of 97.87% and an accuracy of 97.83% on unseen test data, showcasing exemplary performance in distinguishing 'Low', 'Moderate', and 'High' risk scenarios and hence achieving the primary research goal.

Accomplishing this goal required the fulfillment of several sub-objectives. Exploratory data analysis demonstrated the complicated nature of the data and the major problem surrounding class imbalance. A powerful preprocessing pipeline was built, comprising feature engineering of interaction terms, categorical encoding, and standardisation. Discrimination of data points using Principal Component Analysis (PCA) was found to be effective in reducing

dimensionality while retaining 95% of variance of the dataset. While evaluating five different algorithms—Decision Tree, AdaBoost, K Nearest Neighbors, Naive Bayes, and SVM—contributed to identifying SVM as the top contender, hyper-parameter tuning and final retraining proceeded with class balancing to confirm superior performance.

The major finding of this research is that a data-driven approach must be holistic in nature. Model success was not just a function of algorithm choice but the interplay of strong feature engineering, sound preprocessing strategies to combat challenges like class imbalance and inconsistent scaling of features, and systematic model optimization. This allowed for the development of a model that does not merely classify but gives a probability estimate with confidence. In real-life applications, this variance boosts decision-making and strengthens it consciously toward risk.

In relation, this study has wide-ranging effects on the logistics and supply chain sector. The developed model offers a workable solution for the paradigm shift of logistics management from reactiveness to proactiveness. Instead of reacting to delays and disruptions after the fact, managers can now flag high-risk shipments and pre-emptively redirect vehicles, reallocate resources, inform customers of potential delays, or source alternative transportation. Such improvements would translate to lower operating costs, enhanced customer satisfaction, and the creation of an agile, competitive supply chain.

7.2 Limitations

Despite the strong results, it is important to acknowledge the limitations of this study.

1. **Data Source:** The dataset used was simulated. While comprehensive, it may not capture all the nuances, noise, and unpredictability of real-world operational data. A model's true robustness can only be fully validated against live, historical data from a specific company.
2. **Feature Scope:** The model is based on the 26 features provided in the dataset. There may be other external factors not included, such as social or political events, labour strikes, or sudden regulatory changes, that can significantly impact logistics risk.
3. **Static Model:** The current model is static; it is trained on a batch of historical data. In a real-world scenario, supply chain dynamics change over time ('concept drift'). The model would require periodic retraining with new data to maintain its accuracy.

7.3 Future Work

The success of this project opens up several exciting avenues for future research and development.

1. **Real-World Deployment and A/B Testing:** The most critical next step is to deploy the model in a real-world logistics environment. This would involve integrating the saved model into a company's Transportation Management System (TMS) or a dedicated dashboard. An A/B test could be conducted, where one group of shipments is managed using the model's predictions and a control group is managed traditionally, to quantify the real-world impact on cost, delays, and efficiency.
2. **Explainable AI (XAI):** While the SVM model is highly accurate, it is often considered a 'black box'. Future work could involve implementing XAI techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations). This would provide insights into *why* the model classifies a particular shipment as high-risk, highlighting the specific features (e.g., "high traffic congestion" and "poor weather") that contributed most to the prediction. This

explainability would build trust and provide more actionable intelligence to logistics managers.

3. **Dynamic Retraining and Online Learning:** To address the limitation of a static model, a dynamic retraining pipeline could be developed. This system would automatically ingest new data as it becomes available, monitor the model's performance for drift, and trigger retraining when performance degrades below a certain threshold. Exploring online learning algorithms, which can update their parameters incrementally with each new data point, would be a further advancement.
4. **Integration of Unstructured Data:** Future research could explore the integration of unstructured data sources. For example, using Natural Language Processing (NLP) to analyse news feeds, social media, or weather reports to detect disruptive events in real time could provide an even earlier warning system, further enhancing the model's predictive power.
5. **Exploration of Deep Learning Models:** For supply chains with vast amounts of sequential, time-series data, deep learning models like Long Short Term Memory (LSTM) or Transformer networks could be explored. These models are specifically designed to capture temporal dependencies and might offer superior performance in predicting risk trajectories over time.

In conclusion, this research has successfully demonstrated the immense potential of predictive analytics to revolutionise logistics risk management. The high-performing SVM model developed herein serves as a robust proof-of-concept and a solid foundation upon which more advanced and integrated intelligent supply chain systems can be built.

References

Aamer, A., Eka Yani, L.P. and Alan Priyatna, I.M. (2020). Data Analytics in the Supply Chain Management: Review of Machine Learning Applications in Demand Forecasting. Operations and Supply Chain Management: An International Journal, [online] 14(1), pp.1–13. doi:https://doi.org/10.31387/oscm0440281.

Bansal, M., Goyal, A. and Choudhary, A. (2022). A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. Decision Analytics Journal, [online] 3(100071), p.100071. doi:https://doi.org/10.1016/j.dajour.2022.100071.

Douaioui, K., Oucheikh, R., Benmoussa, O. and Mabrouki, C. (2024). Machine Learning and Deep Learning Models for Demand Forecasting in Supply Chain Management: A Critical Review. Applied System Innovation, 7(5), pp.93–93. doi:https://doi.org/10.3390/asi7050093.

Fairee, S., Khompatraporn, C., Sirinaovakul, B. and Prom-On, S. (2020). Trim Loss Optimization in Paper Production Using Reinforcement Artificial Bee Colony. IEEE Access, 8, pp.130647–130660. doi:https://doi.org/10.1109/access.2020.3008922.

Gardas, R. and Narwane, S. (2024). An analysis of critical factors for adopting machine learning in manufacturing supply chains. Decision Analytics Journal, [online] 10, p.100377. doi:https://doi.org/10.1016/j.dajour.2023.100377.

Gaur, M., Goel, S. and Jain, E. (2015). Comparison between Nearest Neighbours and Bayesian Network for demand forecasting in supply chain management. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/7100485.

Govindaraju, P., Achter, S., Ponsignon, T., Ehm, H. and Meyer, M. (2018). Comparison of two clustering approaches to find demand patterns in semiconductor supply chain planning. 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), [online] pp.148–151. doi:https://doi.org/10.1109/coase.2018.8560535.

Goyal, H., Ashna Sukhija, Bhatia, K., Kalpna Guleria and Sharma, S. (2023). Demand Prediction of Consumer Intention to Buy Edible Items Using Machine Learning Techniques. [online] pp.1–6. doi:https://doi.org/10.1109/smartgencon60755.2023.10442766.

Li, P. and Liu, H. (2023). Binary Decomposition for Multi-Class Classification Problems: Development and Applications. 2023 International Conference on Machine Learning and Cybernetics (ICMLC), [online] pp.452–457. doi:https://doi.org/10.1109/icmlc58545.2023.10327973.

Mallala, B., Azeez Khan, P., Pattepu, B. and Eega, P.R. (2024). Integrated Energy Management and Load Forecasting Using Machine Learning. 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), [online] pp.1004–1009. doi:https://doi.org/10.1109/icscss60660.2024.10625623.

Nguyen, B., Shinnie, M.J.D., Leung, C.K., Kuzie, M.R., Kokilev, N.N. and Kaur, S. (2022). A Data Science Solution for Mining Weather Data and Transportation Data for Smart Cities. 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), [online] pp.1672–1677. doi:https://doi.org/10.1109/compsac54236.2022.00266.

None Gaurav and Lal, R. (2023). The Impact of Machine Learning on Supply Chain Management in Relation with Mass Production for Society. doi:https://doi.org/10.1109/icaeeci58247.2023.10370785.

None Mugdha, Sharma, D. and Jose, D. (2024). Transforming Inventory Management Through Machine Learning based Demand Forecasting for a Circular Economy. [online] pp.1–6. doi:https://doi.org/10.1109/ccis63231.2024.10932031.

Pankaj Kumar Dalela, Bansal, P., Yadav, A., Majumdar, S., Yadav, A. and Tyagi, V. (2018). C4.5 Decision Tree Machine Learning Algorithm Based GIS Route Identification. [online] doi:https://doi.org/10.1109/icufn.2018.8436994.

Sharma, A. and Mittal, S. (2024). Prospecting Brain Stroke Onset: A Comparative Analysis Of Supervised Learning Models. [online] pp.1–6. doi:https://doi.org/10.1109/icspcre62303.2024.10675244.

Singh, R., Mandal, P. and Nayak, S. (2024). Reinforcement Learning Approach in Supply Chain Management. pp.271–302. doi:https://doi.org/10.1002/9781394214167.ch17.

Thushan Kavishka Abeysekara and Sulochana Rupasinghe (2019). Analysis of Influential Factors for Inventory Forecasting Systems. doi:https://doi.org/10.1109/i2ct45611.2019.9033725.

Yadav, A., Rajiv Kumar Garg and Anish Kumar Sachdeva (2022). Application of Machine Learning for Sustainability in Manufacturing Supply Chain Industry 4.0 Perspective: A Bibliometric Based Review for Future Research. 2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). doi:https://doi.org/10.1109/ieem55944.2022.9989858.

Yepuri, S.D., Gajjala, P.R., Uppala, Y., Gogulamudi, A.R. and Srinivas, M. (2025). Comparative Analysis of Machine Learning, Deep Learning, Statistical Models on Credit Risk Prediction. 2025 International Conference on Artificial Intelligence and Data Engineering (AIDE), [online] pp.301–306. doi:https://doi.org/10.1109/aide64228.2025.10987509.

Zare Oskouei, M., Zeinal-Kheiri, S., Mohammadi-Ivatloo, B., Abapour, M. and Mehrjerdi, H. (2022). Optimal Scheduling of Demand Response Aggregators in Industrial Parks Based on Load Disaggregation Algorithm. IEEE Systems Journal, 16(1), pp.945–953. doi:https://doi.org/10.1109/jsyst.2021.3074308.