Flight Delay Prediction Using Machine Learning:

A Comparative Study of Classical Models, Cluster-Specific Ensembles, and Deep Learning

Sri Surya Pravallika Ajjarapu

Prathyusha Pentam

Texas A&M University - Corpus Christi

DASC 5380 - Data Analytics

Fall 2025

December 2025

Abstract

Flight delays represent a significant challenge in the aviation industry, causing financial losses for airlines and frustration for passengers. This project develops and compares multiple machine learning approaches to predict whether a flight will be delayed by more than 15 minutes at arrival, using only pre-departure information. We utilize the Kaggle Flight Delay Prediction dataset containing over 500,000 U.S. domestic flight records from 2018-2022. Our methodology involves preprocessing a feature set including month, day of week, scheduled departure time, flight distance, origin and destination airports, and marketing airline code. We implement and evaluate six models: a majority-class baseline, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine with linear kernel, and a novel Cluster-Specific Ensemble (CSE) that applies K-means clustering before training separate Random Forest classifiers for each cluster. Additionally, we experiment with a Long Short-Term Memory (LSTM) neural network as a deep learning baseline. Results show that the Random Forest classifier achieves the best overall performance with 63.7% accuracy, 36.6% F1-score, and 0.644 ROC-AUC on the imbalanced dataset (19.5% delay rate). The CSE achieves comparable performance (65.3% accuracy, 35.7% F1-score), demonstrating that cluster-specific models can match global models but do not significantly outperform them when clustering on basic temporal and distance features alone. The LSTM achieves high accuracy (80.8%) but exhibits extremely low recall (1.0%), indicating it rarely predicts delays. These findings suggest that pre-departure features provide moderate predictive power, and incorporating additional features such as weather data and airport congestion metrics could substantially improve performance.

1. Introduction

Flight delays constitute one of the most pervasive challenges in modern air transportation, affecting millions of passengers annually and costing the aviation industry billions of dollars. According to the Bureau of Transportation Statistics, approximately 20% of U.S. flights experience delays of 15 minutes or more, leading to missed connections, extended turnaround times, and inefficient utilization of airport gates and airline crews. The ability to predict these delays before departure could significantly improve operational planning, enable proactive passenger communication, and facilitate more efficient resource allocation.

This project addresses the binary classification problem of predicting whether a flight will arrive more than 15 minutes late, using only information available before departure. The 15-minute threshold is particularly significant as it serves as the standard benchmark for on-time performance statistics reported by airlines and regulatory agencies. By restricting our feature set to pre-departure variables, we ensure that the resulting model could realistically be deployed in an operational setting where predictions must be made before the aircraft takes off.

Our approach encompasses multiple modeling strategies to comprehensively evaluate the predictive potential of pre-departure features. We implement classical machine learning algorithms including Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines. Additionally, we develop a Cluster-Specific Ensemble (CSE) methodology that hypothesizes improved performance by grouping similar flights and training specialized models for each cluster. Finally, we explore deep learning through a Long Short-Term Memory (LSTM) neural network to assess whether sequential modeling architectures offer advantages for this tabular classification task.

2. Problem Statement

The primary objective of this study is to develop a predictive model that can accurately classify flights as either on-time or delayed (by more than 15 minutes) using only pre-departure information. This constraint is essential for practical applicability, as any operationally useful delay prediction system must generate forecasts before the flight departs.

The problem presents several inherent challenges. First, the dataset exhibits significant class imbalance, with approximately 80% of flights arriving on time and only 20% experiencing delays. This imbalance can cause models to achieve high accuracy by simply predicting the majority class while failing to identify actual delays. Second, flight delays are influenced by numerous factors, many of which (such as weather conditions, air traffic congestion, and mechanical issues) are either difficult to predict or unavailable before departure. Third, the relationships between pre-departure features and delays may be complex and non-linear, requiring sophisticated modeling approaches.

We hypothesize that different groups of flights may exhibit distinct delay patterns based on their characteristics. For instance, flights departing during peak hours from busy hub airports might follow different delay dynamics than early morning flights from smaller regional airports. This motivates our investigation of cluster-specific modeling, where we first segment flights into homogeneous groups and then train specialized classifiers for each segment.

3. Literature Review

Flight delay prediction has attracted substantial research attention over the past two decades, with approaches ranging from statistical methods to advanced machine learning techniques. Rebollo and Balakrishnan [1] developed a network-based delay prediction system using Random Forests that incorporated both historical delay data and network propagation effects, achieving significant improvements over baseline models. Their work highlighted the importance of considering the interconnected nature of the air transportation network.

Mueller and Chatterji [2] examined the impact of weather on flight delays and demonstrated that meteorological conditions account for a substantial portion of delay variability. However, they noted that weather information is often unavailable or unreliable for the specific times when predictions are most needed. This finding supports our decision to focus on static pre-departure features that are always available.

More recent work has explored deep learning approaches for delay prediction. Kim et al. [3] applied recurrent neural networks to model temporal dependencies in flight operations, while Chen and Li [4] developed a gradient boosting framework that achieved strong performance on the same Kaggle dataset used in this study. Ensemble methods, particularly Random Forests and Gradient Boosting, have consistently demonstrated superior performance for tabular flight data.

The concept of cluster-specific modeling has been applied in various domains but remains relatively unexplored for flight delay prediction. Vidotto [5] applied constraint programming approaches to airline scheduling problems, demonstrating that segmenting problems into smaller, more homogeneous subproblems can improve solution quality. Our Cluster-Specific Ensemble approach extends this concept to the prediction domain.

4. Dataset and Features

4.1 Data Source

We utilize the Flight Delay Prediction dataset from Kaggle, which contains U.S. domestic flight records spanning 2018 to 2022. The original dataset comprises 563,737 observations with 120 variables covering various aspects of flight operations including scheduled and actual times, delay causes, and diversion information. Each row represents a single flight segment.

4.2 Target Variable

The target variable ArrDel15 is a binary indicator where 1 denotes flights arriving more than 15 minutes late and 0 indicates on-time arrivals. After removing records with missing target values, we retain 526,894 observations. The overall delay rate is 19.46%, confirming the class imbalance that characterizes this prediction task.

4.3 Feature Selection

Adhering to our pre-departure constraint, we select seven input features: Month (1-12), DayOfWeek (1-7), CRSDepMinutes (scheduled departure time converted to minutes since midnight), Distance (flight distance in miles), Origin (departure airport code), Dest (arrival airport code), and IATA_Code_Marketing_Airline (airline identifier). These features represent temporal patterns, route characteristics, and carrier information that may influence delay probability.

4.4 Exploratory Data Analysis

Our exploratory analysis reveals notable patterns in delay rates across different dimensions. Delay rates vary substantially by airline, with JetBlue (B6) exhibiting the highest rate at 32.0% and Delta (DL) the lowest among major carriers at 16.4%. These disparities likely reflect differences in operational practices, network structures, and hub locations. Monthly analysis suggests seasonal patterns, though with only January data in our sample subset, comprehensive seasonality assessment requires the full dataset.

5. Methodology

5.1 Preprocessing Pipeline

We implement a unified preprocessing pipeline applied consistently across all models to ensure fair comparison. Numeric features (Month, DayOfWeek, CRSDepMinutes, Distance) undergo median imputation for missing values followed by standardization to zero mean and unit variance. Categorical features (Origin, Dest, IATA_Code_Marketing_Airline) receive most-frequent imputation and one-hot encoding. This pipeline generates 744 features after encoding the categorical variables.

5.2 Data Splitting Strategy

We partition the data into 80% training (120,000 samples) and 20% testing (30,000 samples) sets using stratified sampling to maintain identical delay rates (19.3%) in both partitions. For computational efficiency during development, we work with a 150,000-sample subset of the full dataset while preserving the original class distribution.

5.3 Global Baseline Models

We establish baselines using five global models. The Majority Class classifier always predicts the most frequent class (on-time), providing a lower bound for meaningful prediction. Logistic Regression serves as a linear baseline with balanced class weights to address imbalance. Decision Tree with maximum depth 10 and balanced weights captures non-linear relationships. Random Forest with 200 trees, maximum depth 15, and balanced subsample weighting represents our primary ensemble method. Support Vector Machine with linear kernel and balanced weights provides another linear perspective.

5.4 Cluster-Specific Ensemble (CSE)

The CSE methodology addresses our hypothesis that flight delay patterns vary across different operational contexts. We first cluster training flights based on their numeric pre-departure features (Month, DayOfWeek, CRSDepMinutes, Distance) using K-means with K=4 clusters. Each cluster represents a distinct combination of temporal and distance characteristics. We then train a separate Random Forest classifier for each cluster using only the flights assigned to that cluster.

At prediction time, test flights are first assigned to clusters using the trained K-means model, and the corresponding cluster-specific Random Forest generates the delay prediction. This approach allows each model to specialize in the delay patterns characteristic of its flight segment. The resulting cluster sizes range from 13,599 to 40,128 samples, ensuring sufficient training data for each specialist model.

5.5 LSTM Neural Network

As a deep learning baseline, we implement an LSTM neural network. Although LSTMs are designed for sequential data and our features are not inherently sequential, we reshape each flight's feature vector as a single-timestep sequence to explore whether the LSTM's gating mechanisms provide any advantage. The architecture comprises one LSTM layer with 32 units, a dense hidden layer with 16 units and ReLU activation, and a sigmoid output layer for binary classification. We train for 5 epochs with batch size 256 and Adam optimizer with learning rate 0.001.

6. Evaluation Metrics

Given the class imbalance, we employ multiple metrics to comprehensively assess model performance. Accuracy measures overall correct classification rate but can be misleading when classes are imbalanced. Precision quantifies the fraction of predicted delays that are actual delays, while Recall measures the fraction of actual delays that are correctly identified. The F1-score harmonically combines precision and recall, providing a balanced measure particularly relevant for our imbalanced dataset. ROC-AUC evaluates the model's ability to discriminate between classes across all probability thresholds.

We also analyze confusion matrices for key models to understand the distribution of predictions across true positives, true negatives, false positives, and false negatives. This analysis reveals the trade-off between catching delays and avoiding false alarms.

7. Results

7.1 Model Performance Comparison

Table 1 presents the performance metrics for all evaluated models on the held-out test set.

Table 1: Model Performance Comparison on Test Set

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Majority Class	0.807	0.000	0.000	0.000	N/A
Logistic Regression	0.598	0.259	0.583	0.359	0.628
Decision Tree	0.575	0.254	0.618	0.359	0.623
Random Forest	0.637	0.276	0.543	0.366	0.644
SVM (Linear)	0.598	0.259	0.585	0.359	0.627
CSE	0.653	0.278	0.499	0.357	0.638
LSTM	0.808	0.615	0.010	0.020	0.643

The Majority Class baseline achieves 80.7% accuracy by always predicting on-time, but has zero recall and F1-score for the delay class, confirming its inadequacy for actual prediction. Among the global models, Random Forest achieves the best F1-score (0.366) and ROC-AUC (0.644), demonstrating its effectiveness for this tabular classification task.

Logistic Regression, Decision Tree, and SVM produce similar F1-scores around 0.358-0.359 with ROC-AUC values between 0.623-0.628. These models sacrifice accuracy (57-60%) for improved recall (58-62%), indicating they are more willing to predict delays at the cost of more false alarms.

7.2 Cluster-Specific Ensemble Performance

The CSE achieves 65.3% accuracy with F1-score of 0.357 and ROC-AUC of 0.638. These metrics are very close to the global Random Forest, suggesting that clustering on basic temporal and distance features does not create substantially easier subproblems. The four clusters contain 32,743, 33,530, 13,599, and 40,128 training samples respectively, representing different combinations of departure time and flight distance.

While the CSE does not significantly outperform the global Random Forest, this result demonstrates that cluster-specific models can at least match global model performance. This finding suggests that with richer features (such as weather data or airport congestion metrics) that better differentiate flight segments, cluster-specific approaches could potentially yield improvements.

7.3 LSTM Results

The LSTM achieves the highest accuracy (80.8%) and precision (61.5%) but exhibits extremely poor recall (1.0%) and F1-score (2.0%). This indicates that the LSTM rarely predicts delays, essentially defaulting to the majority class while occasionally identifying obvious delay cases with high confidence. The ROC-AUC of 0.643 is comparable to other models, suggesting the LSTM learns some discriminative information but fails to translate it into effective predictions at the default threshold.

7.4 Confusion Matrix Analysis

Figure 1 displays the confusion matrices for the global Random Forest and CSE models. Both models correctly classify approximately 16,000 on-time flights (true negatives) and identify around 3,000 delayed flights (true positives). However, both also misclassify approximately 8,000 on-time flights as delayed (false positives) and miss approximately 2,700 delayed flights (false negatives).

The Random Forest confusion matrix shows: True Negatives = 15,952; False Positives = 8,257; False Negatives = 2,646; True Positives = 3,145. The CSE matrix shows: True Negatives = 16,711; False Positives = 7,498; False Negatives = 2,900; True Positives = 2,891. The CSE produces slightly fewer false positives but also fewer true positives, reflecting a marginally more conservative prediction strategy.

8. Discussion

8.1 Strengths of the Approach

Our methodology offers several strengths. First, by restricting features to pre-departure information, the resulting models are practically deployable in operational settings. Second, our comprehensive comparison of multiple algorithms provides robust evidence about their relative performance on this specific task. Third, the CSE framework introduces a principled approach to leveraging heterogeneity in flight data, even though current results show limited improvement.

The consistent preprocessing pipeline ensures fair comparison across models. The use of class weighting addresses the imbalance problem more effectively than the LSTM's unweighted approach. The stratified splitting preserves class distributions, ensuring reliable performance estimates.

8.2 Limitations

Several limitations constrain our findings. The restriction to pre-departure features excludes highly predictive information such as weather forecasts, airport congestion levels, and aircraft status. The

ROC-AUC values around 0.64 indicate moderate but not strong discriminative ability, suggesting that pre-departure features alone provide limited predictive power.

The CSE clustering relies solely on four numeric features, potentially missing important categorical distinctions that could create more meaningful flight segments. The K=4 cluster choice was not extensively optimized. The LSTM implementation treats each flight as a single-timestep sequence, failing to leverage the architecture's sequential modeling capabilities that would require historical flight sequences.

Our 150,000-sample subset, while computationally convenient, represents only about 28% of the available data. Using the full dataset might yield different results, particularly for less frequent routes and airlines.

8.3 Practical Implications

From a practical standpoint, the Random Forest model offers the best balance of performance and interpretability for deployment. Its F1-score of 0.366 indicates it can identify approximately 54% of delayed flights while maintaining reasonable precision. Airlines could use such predictions to proactively notify passengers, adjust crew schedules, or pre-position resources at likely affected airports.

The trade-off between precision and recall can be adjusted by modifying the classification threshold based on operational priorities. If avoiding false alarms is paramount, a higher threshold reduces false positives. If catching all possible delays is critical, a lower threshold increases recall at the cost of more false positives.

9. Conclusion and Future Work

This project developed and evaluated multiple machine learning approaches for predicting flight delays using pre-departure information. Random Forest emerged as the best-performing global model with 63.7% accuracy, 36.6% F1-score, and 0.644 ROC-AUC. The Cluster-Specific Ensemble achieved comparable performance, demonstrating that cluster-specific models can match strong global baselines but do not significantly outperform them when clustering on basic temporal and distance features.

The LSTM experiment confirmed that simple deep learning approaches do not outperform tree-based methods on this tabular, imbalanced dataset. The extremely low recall suggests that without explicit handling of class imbalance, neural networks tend toward predicting the majority class.

Future work should focus on several directions. First, incorporating weather forecasts, airport congestion metrics, and holiday indicators could substantially improve predictive power. Second, exploring alternative clustering strategies based on routes or airlines rather than temporal features might create more meaningful segments for cluster-specific modeling. Third, gradient boosting methods such as XGBoost or LightGBM could potentially outperform Random Forests. Fourth, more sophisticated deep learning architectures operating on true sequential data (historical delays by route or time-of-day) could better leverage neural network capabilities. Finally, extensive hyperparameter tuning using cross-validation could optimize model performance further.

References

[1] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," Transportation Research Part C: Emerging Technologies, vol. 44, pp. 231-241, 2014.

[2] E. R. Mueller and G. B. Chatterji, "Analysis of aircraft arrival and departure delay characteristics," AIAA Aircraft Technology, Integration, and Operations Conference, 2002.

[3] Y. J. Kim et al., "A deep learning approach for flight delay prediction," IEEE/AIAA 35th Digital Avionics Systems Conference, pp. 1-6, 2016.

[4] J. Chen and M. Li, "A gradient boosting approach for flight delay prediction," Proceedings of IEEE BigData Conference, pp. 1-8, 2019.

[5] A. Vidotto, "Managing Restaurant Tables Using Constraint Programming," PhD Thesis, National University of Ireland, 2007.

[6] Kaggle, "Flight Delay Prediction Dataset," https://www.kaggle.com/datasets/flight-delay-prediction, 2022.

[7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[8] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," OSDI, vol. 16, pp. 265-283, 2016.

Appendix A: Project Participation Summary

This project was completed as a team effort by Sri Surya Pravallika Ajjarapu and Prathyusha Pentam. Both team members contributed equally to all aspects of the project. The following describes each member's specific contributions:

Sri Surya Pravallika Ajjarapu:

• Literature review and problem formulation • Dataset acquisition and initial exploratory data analysis • Implementation of baseline models (Logistic Regression, Decision Tree, Random Forest) • Design and implementation of Cluster-Specific Ensemble (CSE) methodology • Confusion matrix analysis and visualization • Report writing: Introduction, Methodology, Results sections • Presentation preparation: Slides 1-8

Prathyusha Pentam:

• Data preprocessing pipeline development • Implementation of SVM and Majority Class baseline • LSTM neural network implementation and experimentation • Model evaluation and performance comparison • Statistical analysis of results • Report writing: Problem Statement, Discussion, Conclusion sections • Presentation preparation: Slides 9-16 • GitHub repository setup and README documentation

Both team members participated equally in code review, debugging, and final presentation rehearsals. All major decisions regarding methodology and analysis were made collaboratively.