Enhancing Flight Services: Predictive Modeling and Sequential Pattern Analysis

Analyzing 2013 flight data to predict and prevent delays, improving passenger experience and airline efficiency.

Supervisor: Soroush Sheikh

Student Names:

- Eric Traccitto (218835074),
- Nicholas Veronico (218567610),
- Marcus Sisouphanh (217544792),
- Evan Oni (218877670)

1. Project Objective

Analyze Flight Data

Identify key factors contributing to flight delays using 2013 data.

Predict Delay Duration

Develop regression models to estimate length of delays, not just classify them.

Optimize Scheduling

Generate improved scheduling strategies for flights predicted to experience delays.

2. Motivation

Improved Passenger Satisfaction

Better experiences through reduced delays

Lower Operating Expenses

More efficient resource allocation

Better Decision-Making

Data-driven insights for airlines

Understanding External Factors

Weather, traffic, operations impact

3. Related Work

Machine Learning Classifiers

R.G. et al. (2024) used SVM and Decision Trees to predict delays with high accuracy.

1

2

Algorithm Comparison

Hatıpoğlu and Tosun (2024) compared Random Forest and Gradient Boosting methods.

Sequential Pattern Mining

Zhang et al. (2020) analyzed how delays propagate through connecting flights.

3

4. Methodology Overview

Data Preprocessing

Clean and prepare flight data

Probabilistic Analysis

Estimate the likelihood of delays

Optimization

Refine for best performance

Exploratory Analysis

Identify patterns and trends

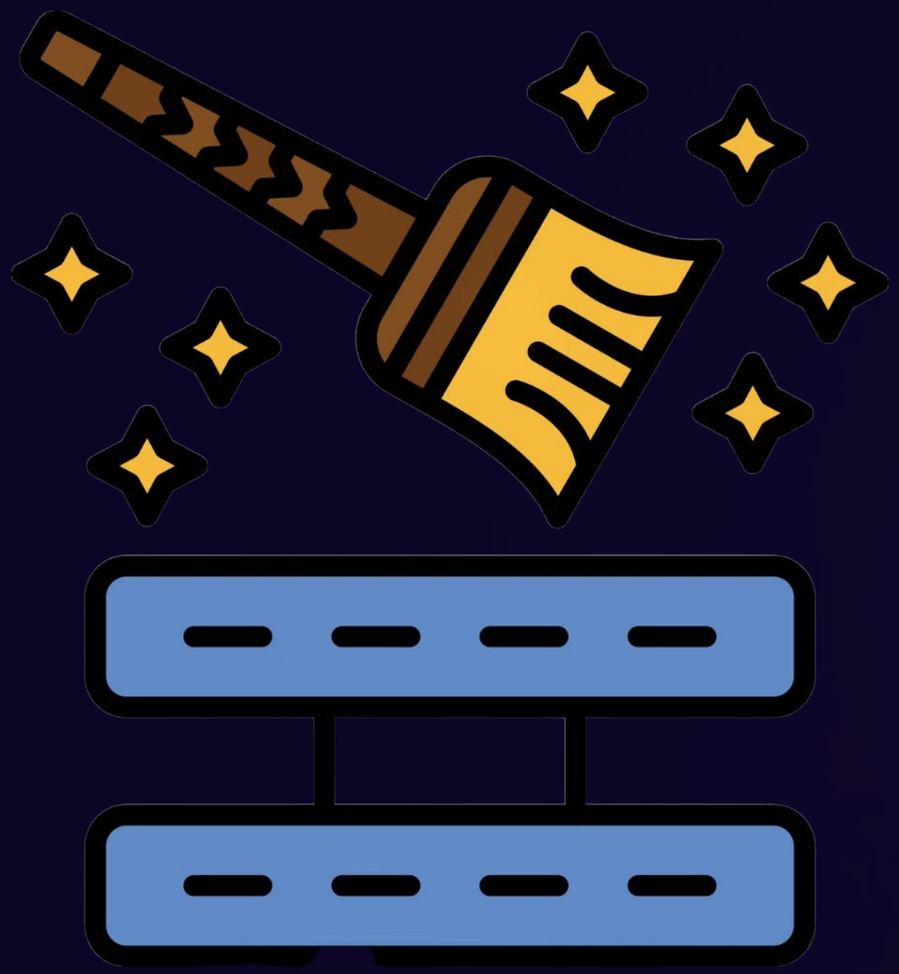
Model Selection

Choose appropriate algorithms

Training & Evaluation

Build and test models

https://png.pngtree.com/png-vector/20191021/ourmid/pngtree-working-tools-png-image_1795128.jpg



4.1 Data Preprocessing

Handle Missing Values

Replace missing categorical values and categorize flights by status.

Dataset Merge

Merge a airport weather dataset to our flights dataset.

Detect Outliers

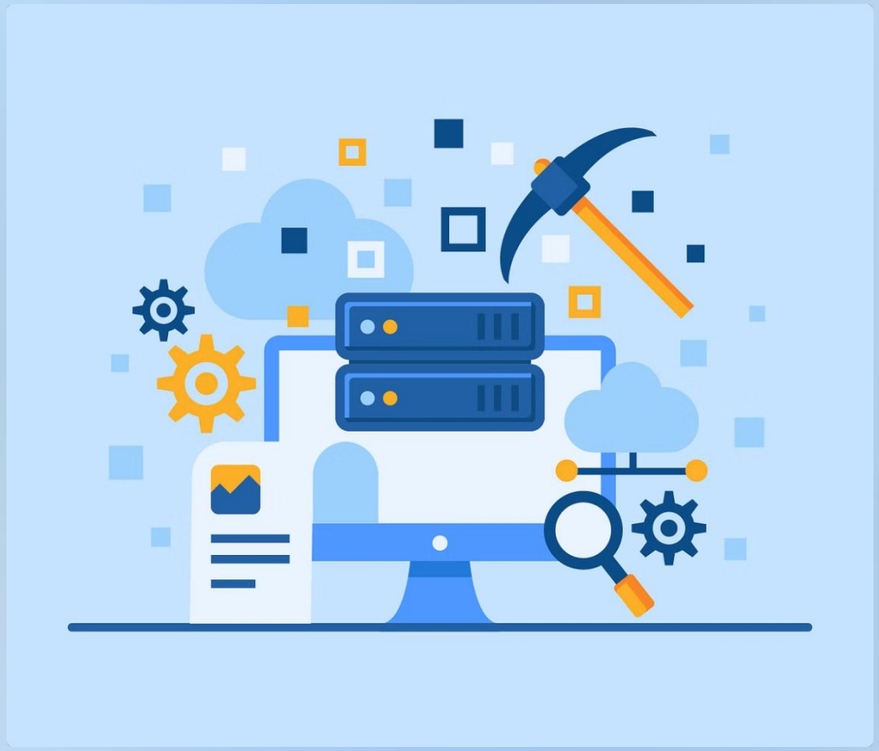
Use IQR method to identify extreme values in numerical features.

Encode Categories

Transform carrier, origin, and destination data with one-hot encoding.

Feature Engineering

Create time categories and apply cyclical encoding for temporal data.



Feature Engineering

Time Categories

Created departure and arrival peak time categories to capture patterns.

- Early Morning (0-600)
- Morning (600-1200)
- Afternoon (1200-1500)
- Evening Rush (1500-2000)

Cyclical Encoding

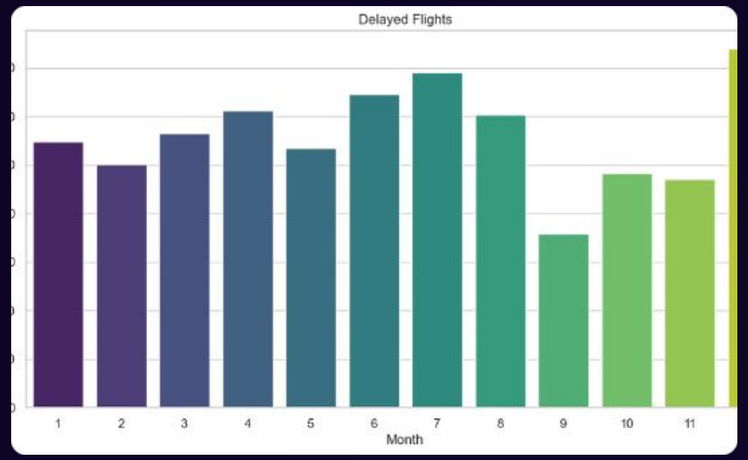
Applied sine/cosine transformations to capture cyclical time patterns.

- Day of week (7-day cycle)
- Month (12-month cycle)

DateTime Extraction

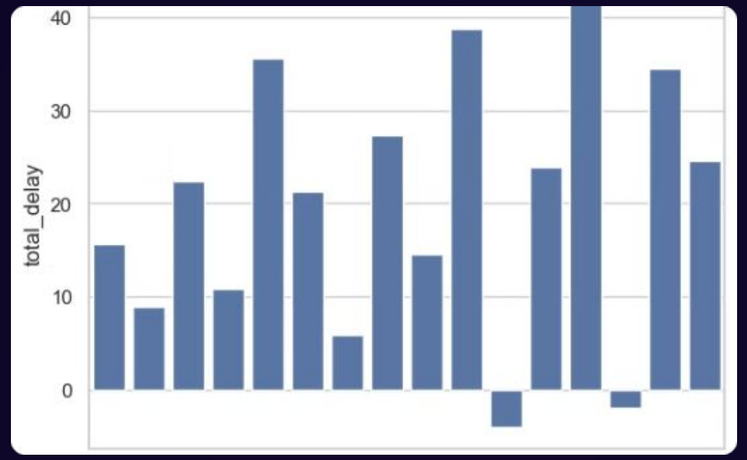
Pulled hour, day_of_week, and month from timestamps for analysis.

4.2 Exploratory Data Analysis



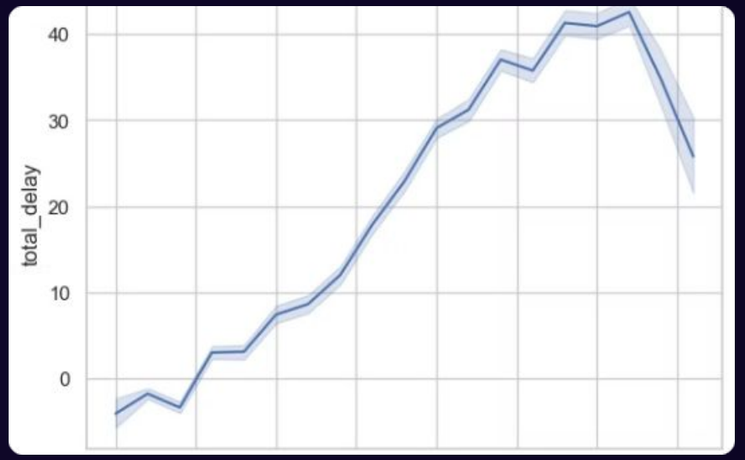
Seasonal Trends

Summer months show highest delays, followed by winter holiday season.



Carrier Performance

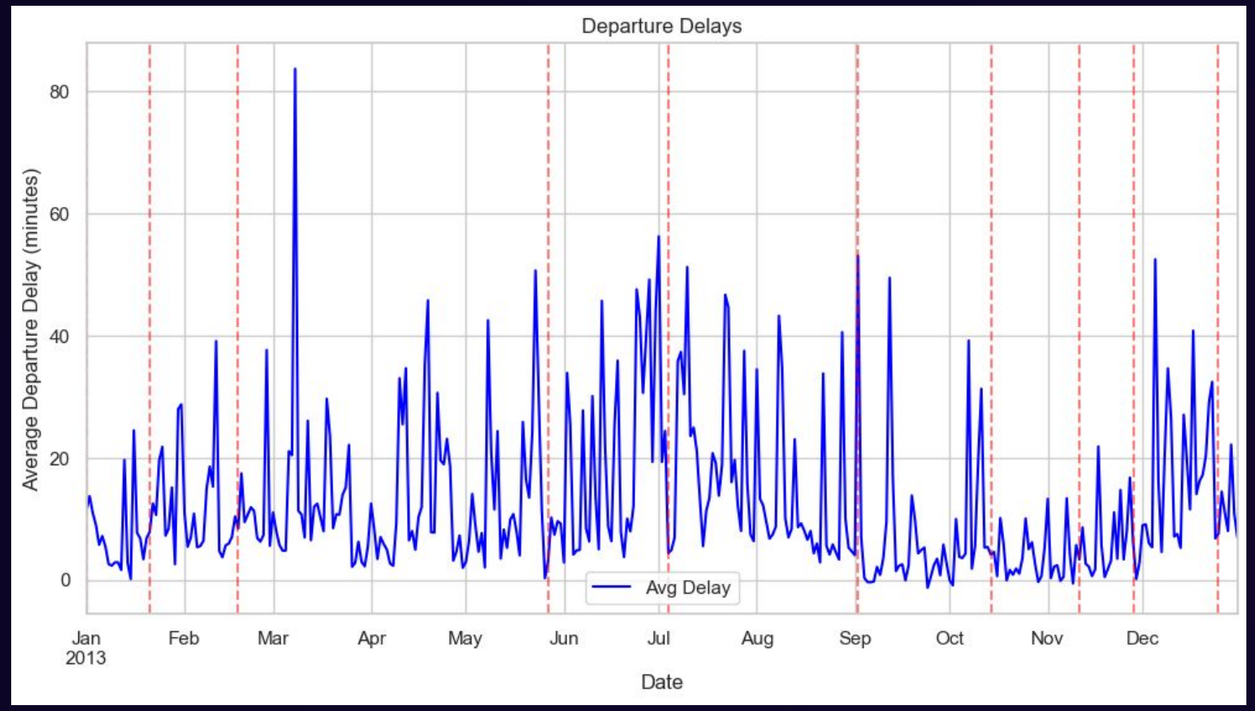
Some carriers consistently show higher delays than others.



Time of Day Impact

Delays increase throughout the day, peaking at 19:00-20:00.

Exploratory Data Analysis

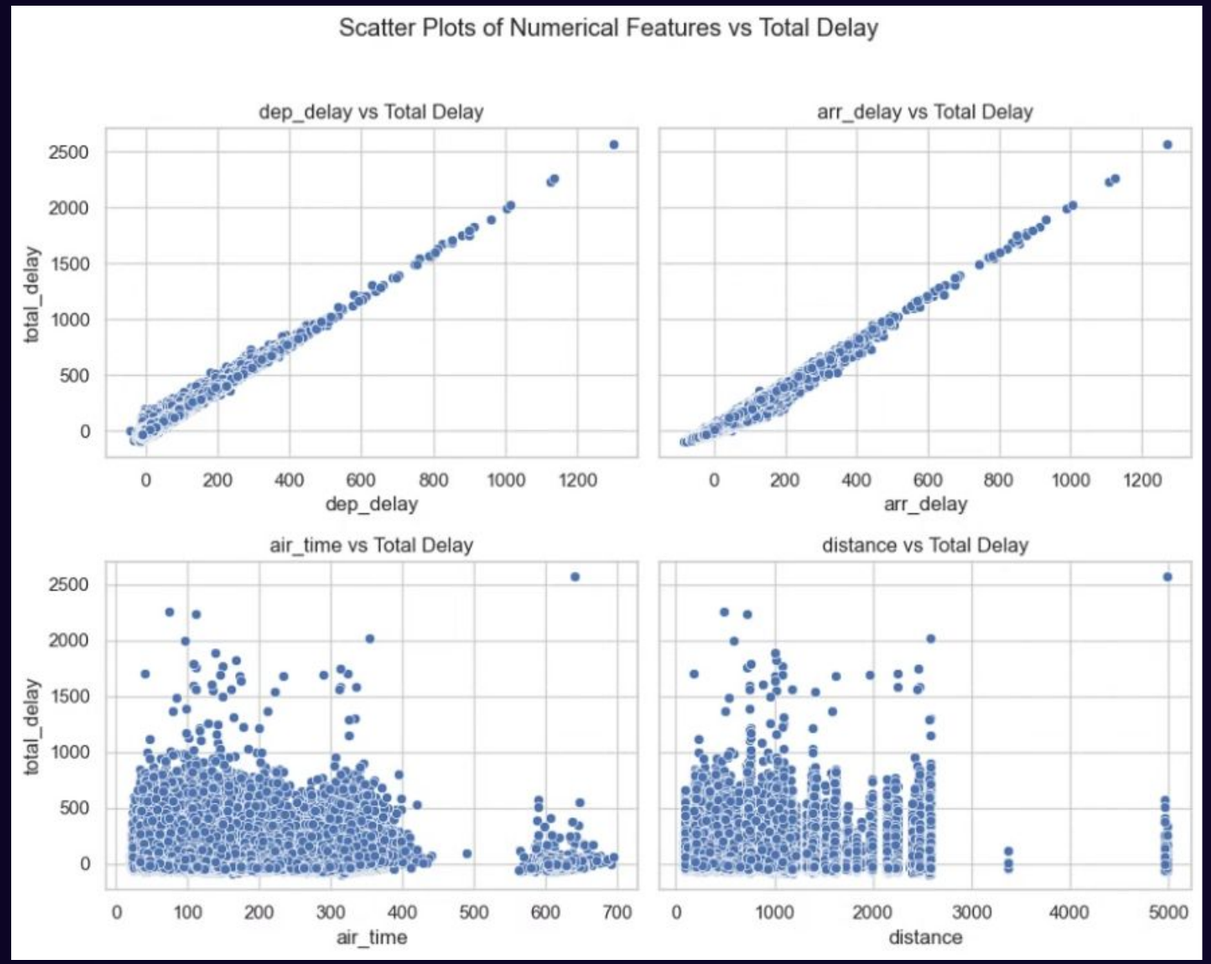


Holiday and Date Analysis

Holidays: Delay spikes on Holidays

Non-Holidays: Delay spikes due to other factors

Exploratory Data Analysis

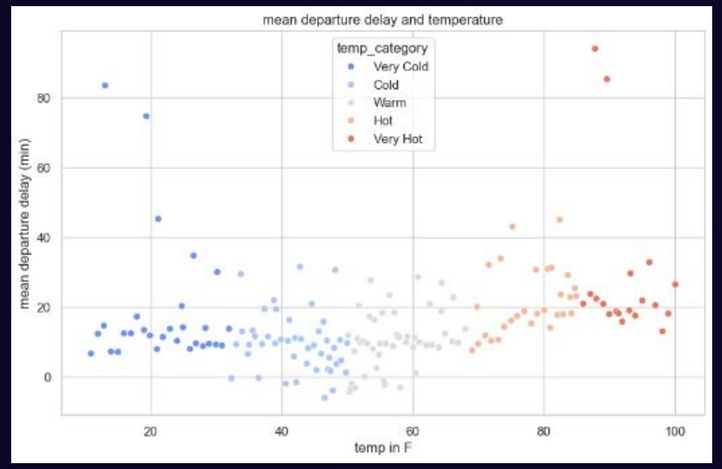


Correlation Analysis

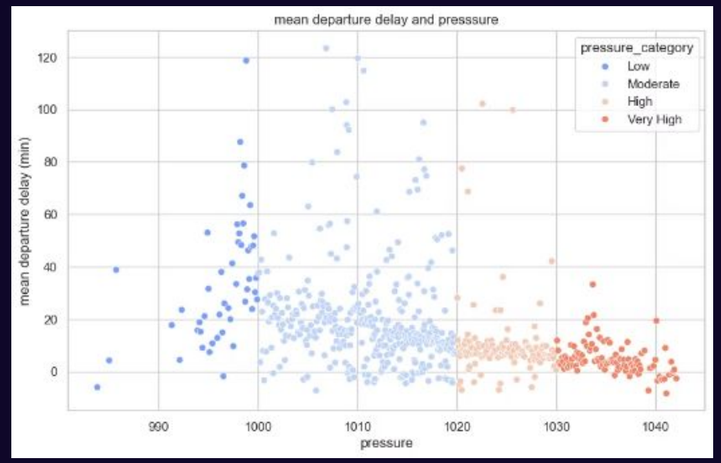
Strong Correlation: Departure Delay, Arrival Delay and Total Delay

Weak Correlation: Air Time and Distance

Weather Impact Analysis



Temperature extremes (both hot and cold) correlate with increased delays.

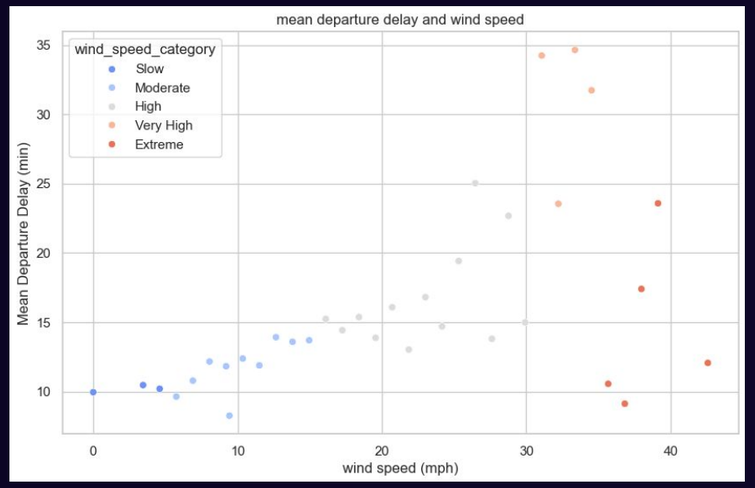


Lower pressure values tend to cause higher delays in most cases.

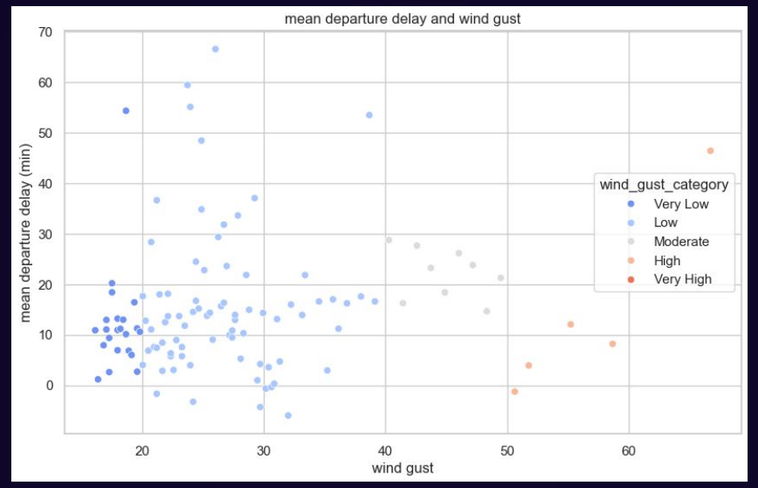
Key Weather Findings

- Higher precipitation links to increased delays
- Low visibility (0-2) strongly contributes to delays
- Wind speed shows complex relationship with delays
- Pressure anomalies correlate with delay patterns

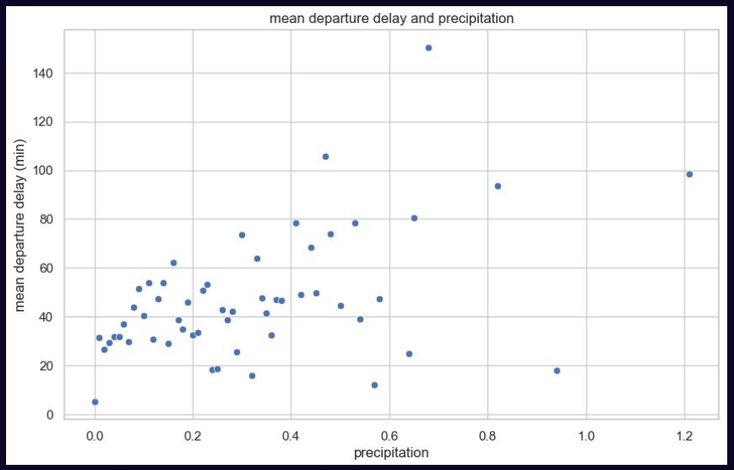
Weather Impact Analysis



High wind tends to cause higher delays in most cases



Higher wind gust tends to increase delays in most cases



Higher precipitation tends to cause higher delays in most cases

4.3 Model Selection

Linear Regression

Baseline model to assess dataset linearity and provide interpretability.

Random Forest

Ensemble method to handle nonlinear relationships and provide feature importance.

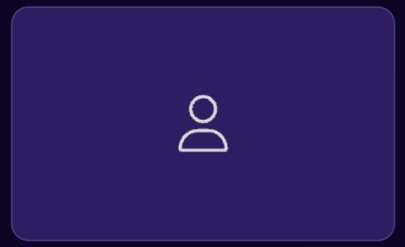
XGBoost

Gradient boosting optimized for structured data with high predictive accuracy.

Support Vector Regressor

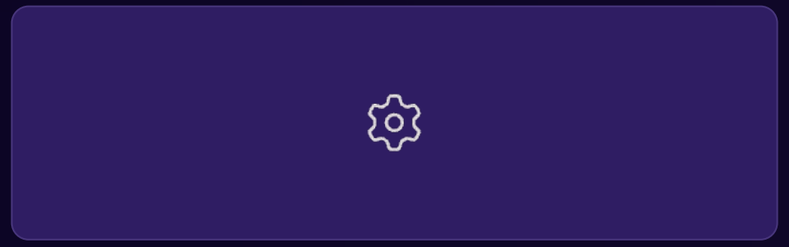
Effective in high-dimensional spaces for distinguishing delay patterns.

4.4 Model Training and Evaluation



Data Splitting

70% training, 30% testing to ensure robust model performance



Model Training

Applied Linear SVR, SVM with RBF kernel, Random Forest, and XGBoost

Performance Metrics

Evaluated using RMSE and 5-fold cross-validation

4.5 Probabilistic Analysis

Poisson Regression

Implemented with XGBoost to model delays to be assumed count-based distribution.

Delay Time

Delay times are represented in minutes, and not a count value representing an event. Poisson could not be used for this situation.

4.6 Optimization Strategies

Hyperparameter Tuning

Used Grid Search to find the best model parameters for model

- Learning rate optimization
- Tree depth calibration
- Regularization parameter adjustment

Computational Optimization

Implemented parallel processing to handle large dataset efficiently.

- Reduced training time
- Enabled more extensive testing
- Supported complex model architectures

5. Deliverables

```python
from sklearn.ensemble import RandomForestRegressor

# Random Forest Regressor
rf_model = RandomForestRegressor(
    n_estimators=10,
    max_depth=10,
    random_state=42,
    n_jobs=-1
)

# Train
rf_model.fit(X_train, y_train)

# Predictions
y_pred_rf = rf_model.predict(X_test)

# RMSE evaluate
rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred_rf))
print("Random Forest RMSE:", rmse_rf)
```

```python
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import numpy as np

# Data split
X_train, X_test, y_train, y_test = train_test_split(X_preprocessed, y, test_size=0.3, random_state=42)

# Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# RMSE Evaluation
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error (RMSE):", rmse)
```

Codebase

Well-documented repository with implementation of models and analysis scripts.

Trained Models

Optimized predictive models saved as artifacts for future use.

Technical Report & Presentation

Comprehensive documentation of methodology, findings, and implications.

6. Experimental Setup

Hardware Resources

Personal computer or cloud-based environment with sufficient computational power.

- Google Colab for cloud-based model training and experimentation Git & GitHub for version control and collaborative development

Software Tools

Development environment and libraries for machine learning implementation.

- Python 3.x
- Jupyter Notebook/VS Code
- Scikit-learn, Pandas, NumPy, Statsmodels, PyTorch
- GitHub

Document Preparation & Reporting

- LaTeX or Markdown for structured documentation
- PowerPoint or Google Slides for presentation preparation

Data Access

Historical flight datasets and weather information via API and dataset.

- Flight schedules and delays
- Weather conditions
- Airline operational data

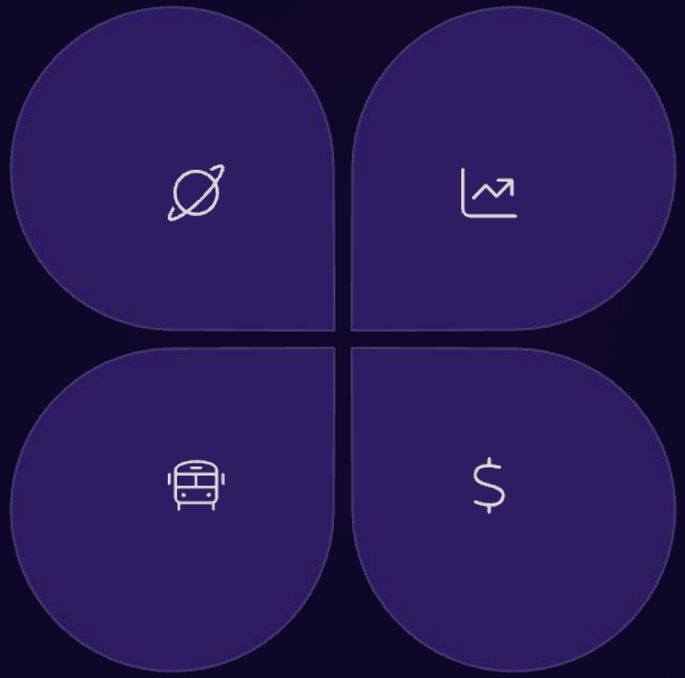
7. Project Impact

Optimized Airline Scheduling

Proactively manage congestion and adjust departure times to minimize disruptions.

Cross-Sector Applications

Methodology adaptable to train schedules and public transportation optimization.



Data-Driven Decision Making

Provide actionable insights for fuel planning, crew scheduling, and operations.

Economic Benefits

Reduce costs from crew overtime, additional fuel, and passenger compensation.

8. Results: Feature Engineering

Cyclical Encoding

Applied to temporal features like day of week and month to capture periodic patterns.

- Transformed into sine/cosine components
- Better captured cyclical relationships

Categorical Processing

One-hot encoding applied to carrier, origin, destination, and tail number.

- Created high-dimensional feature space
- 4,180 columns in final dataset

Model Selection

XGBoost regressor emerged as best performer with optimized parameters.

- Learning rate: 0.1
- Max depth: 5
- Estimators: 200

Results: Optimization Approach

Basic Time Shifting

Initially tested shifting departures earlier in 1-hour increments.

Pattern Discovery

Found that earlier departures sometimes unexpectedly increased delays.

Net Savings Calculation

Developed formula: Net Savings = (Hours Shifted Earlier × 60) - (New Delay - Original Delay)

Targeted Optimization

Discovered 6 AM to 5 AM shift produced most significant improvements.

Smart Constraints

Blocked shifting of evening flights due to poor performance.

Results: Key Findings

170+

Minutes Saved

Maximum net time savings for optimized 6 AM flights

5 AM

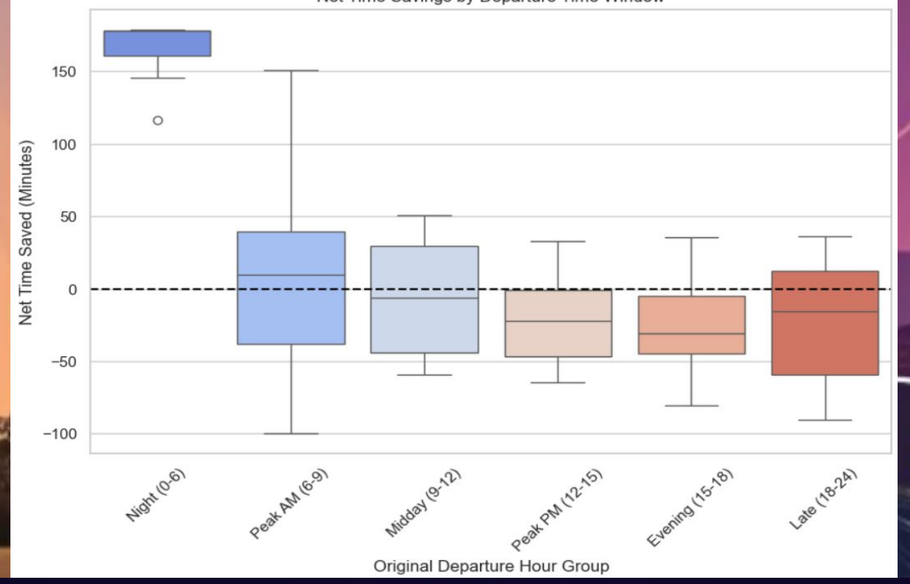
Optimal Time

Best departure time for previously 6 AM flights

2 PM+

No Benefit Zone

Flights after this time showed no improvement when shifted



9. Conclusion: Time-of-Day Impact

Early Morning Success

5 AM departures benefit from quieter airspace and smoother operations.

Mid-Day Mixed Results

Variable benefits depending on specific airport and carrier conditions.

Afternoon/Evening Challenges

Shifting earlier often worsens delays due to increased congestion.

Conclusion: Strategic Recommendations

Time-Segmented Approach

Adopt different strategies based on time of day rather than universal rescheduling.

- Early morning: Consider shifting earlier
- Afternoon/evening: Maintain or delay

Data-Driven Decisions

Use predictive models to guide specific flight adjustments rather than blanket policies.

- Evaluate each route individually
- Consider seasonal variations

Targeted Improvements

Focus on high-impact changes that show consistent benefits in historical data.

- Prioritize 6 AM to 5 AM shifts
- Monitor results and adjust strategy

10. Limitations: Optimization Challenges

Time-Dependent Effectiveness

Rescheduling benefits vary greatly by time of day, limiting universal application.

Inconsistent Patterns

Shifting later flights earlier often increased delays rather than reducing them.

Congestion Factors

Early morning flights benefit from less traffic, while later flights face more constraints.

Weather Complexity

Weather impacts vary by season, location, and time of day in ways hard to model.

Limitations: Model Constraints

Predictive Modeling Gaps

XGBoost Regression needs further tuning to better handle complex relationships.

- Peak time congestions
- External airport delay factors

Missing Real-Time Data

Model lacks current congestion levels.

- Air traffic density
- Gate availability
- Ground operations status

External Variables

Several important factors not included in current model:

- Airline regulations
- Fuel constraints
- Airplane status
- Crew scheduling
- Security clearance times
- Many more

Future Work

Real-Time Data Integration

Incorporate live air traffic and weather information

Operational Constraints

Add maintenance requirements and crew scheduling factors

Decision Support System

Develop end-to-end framework for airline scheduling decisions

Advanced AI Models

Explore deep learning approaches for complex pattern recognition

Thank You

Questions? Contact us for more information about implementing these predictive models to enhance your flight operations and reduce delays.