Project Documentation: Airline Delay Prediction System

1. Executive Summary

The **Airline Delay Prediction System** is a production-ready Machine Learning solution designed to forecast flight delays. It utilizes a **Hybrid Intelligence** architecture that synergizes a rigorously trained statistical model (Logistic Regression) with a rule-based expert system. This approach allows the system to balance historical data patterns with real-time dynamic factors—such as weather conditions—that are often absent from static datasets.

Key technical highlights include the use of **L1 (Lasso) Regularization** for automatic feature selection, **L2 (Ridge) Regularization** for handling multicollinearity, and a custom **Explainable AI (XAI)** module.

---

2. System Architecture & Components

The project is modularized into distinct components to separate data processing, model training, and user interaction logic.

2.1 Core Modules

- **airline_delay_prediction.py (Training Engine):** The primary script responsible for the end-to-end Machine Learning pipeline. It handles data loading, Exploratory Data Analysis (EDA), preprocessing, model training, and evaluation.
- **delay_prediction_gui.py (User Interface):** A modern graphical interface built with tkinter. It decouples inference from training, loading the pre-trained models to allow users to simulate flight scenarios.
- **prediction_utils.py (Shared Utility):** A helper module containing core logic for loading artifacts and calculating explainability metrics.
- **generate_data.py (Data Generator):** A script to generate synthetic flight data with realistic feature correlations.

---

3. Technical Deep Dive

3.1 Data Pipeline & Exploratory Analysis

The foundation of the model is a robust understanding of the data distribution. Initial analysis focused on identifying class imbalance and feature relationships.

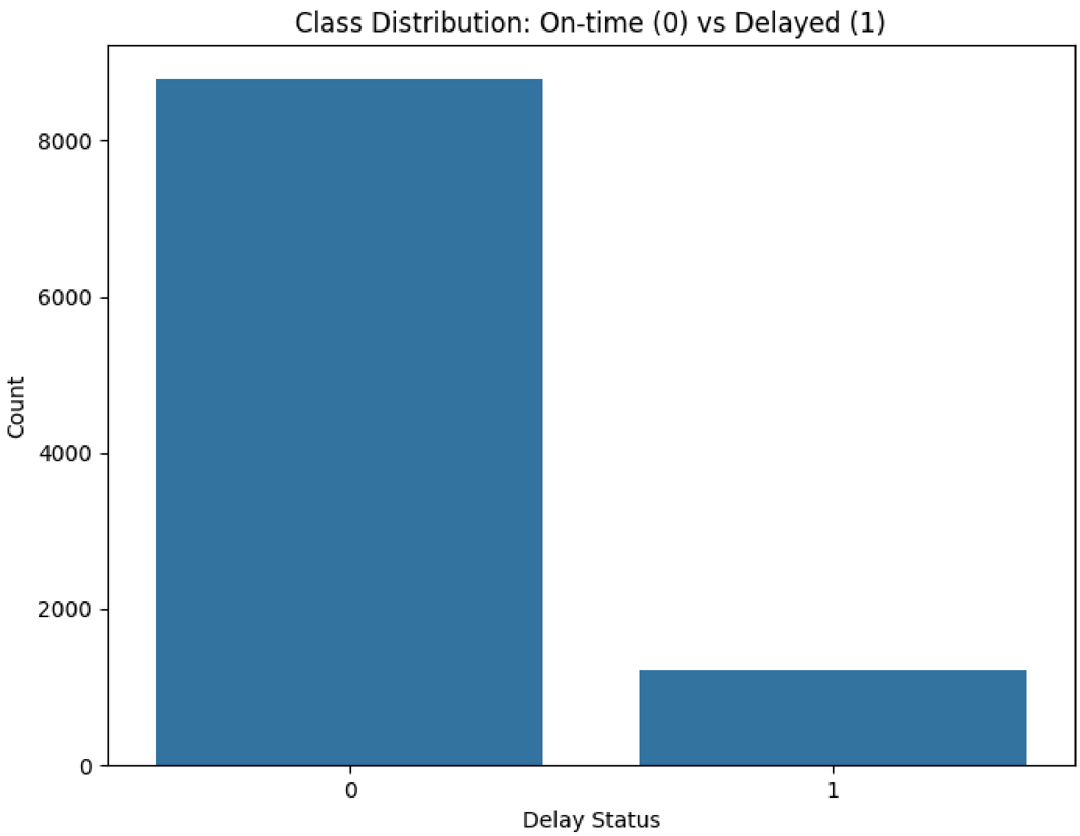

Figure 1: Distribution of flight delays. This visualization highlights the class imbalance between on-time and delayed flights, necessitating the use of balanced class weights during training.

The system implements a robust preprocessing pipeline:

- **Feature Engineering:** Utilization of temporal features (Month, Day) and flight metadata.
- **Correlation Analysis:** To ensure model stability, we analyzed feature correlations to identify multicollinearity (e.g., between DISTANCE and AIR_TIME).

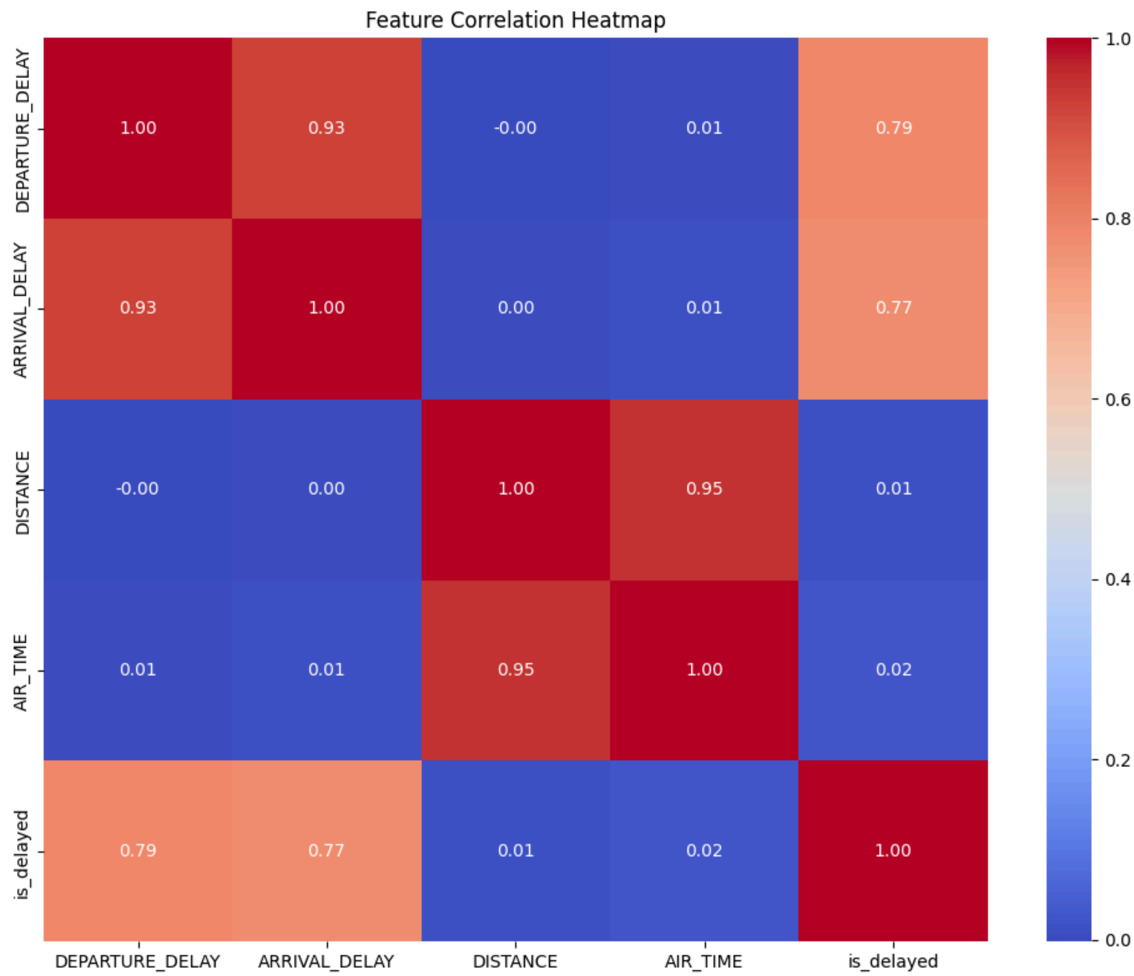

Figure 2: Feature Correlation Heatmap. This analysis helps in selecting features and confirms the need for L2 regularization to handle highly correlated predictors.

3.2 Machine Learning Models

The system trains and compares two variations of **Logistic Regression**:

1. **L1 Regularization (Lasso):** Serves as a feature selector by forcing coefficients of irrelevant features to zero.
2. **L2 Regularization (Ridge):** Used to penalize large weights, effectively handling multicollinearity.

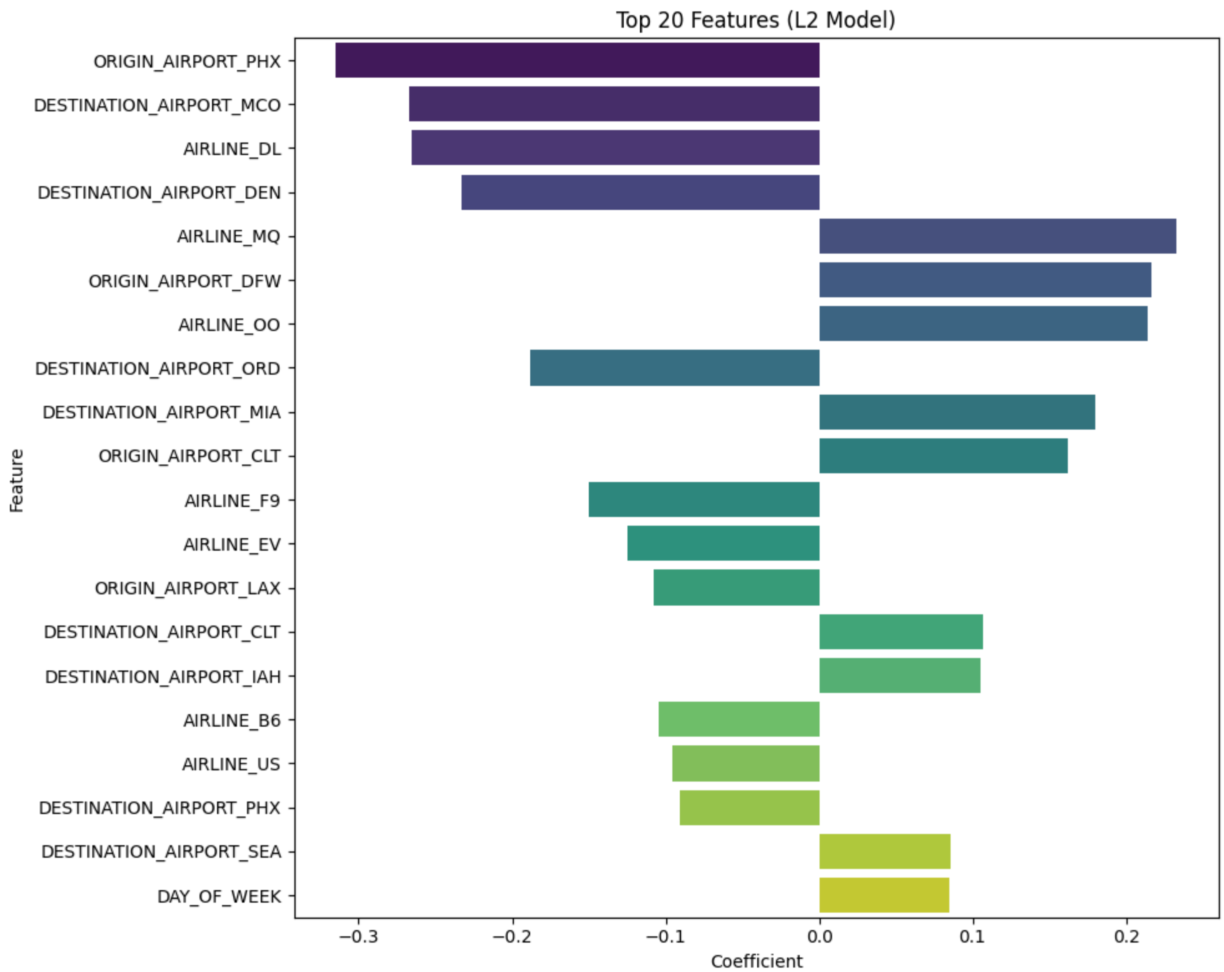

Figure 3: Top 20 Features (L2 Model). This chart visualizes the coefficients of the Ridge model, showing which factors (e.g., specific airlines or airports) contribute most positively or negatively to the delay risk.

3.3 Hybrid Intelligence Layer

Since the training dataset lacks real-time weather data, the system applies expert rules during inference to adjust risk probabilities:

- **Snow:** +30% Delay Risk
- **Rain:** +15% Delay Risk
- **Storm:** +45% Delay Risk
- **Fog:** +10% Delay Risk

3.4 Explainable AI (XAI)

The system calculates the **Marginal Probability Contribution** for each feature. This approximates the change in probability caused by a specific factor, allowing users to see exactly *why* a delay is predicted.

---

4. Evaluation & Metrics

The project prioritizes robust metrics over simple accuracy due to the imbalanced nature of flight delay data.

4.1 Model Performance

We evaluated the models using ROC-AUC to measure the ability to distinguish between classes across all thresholds.

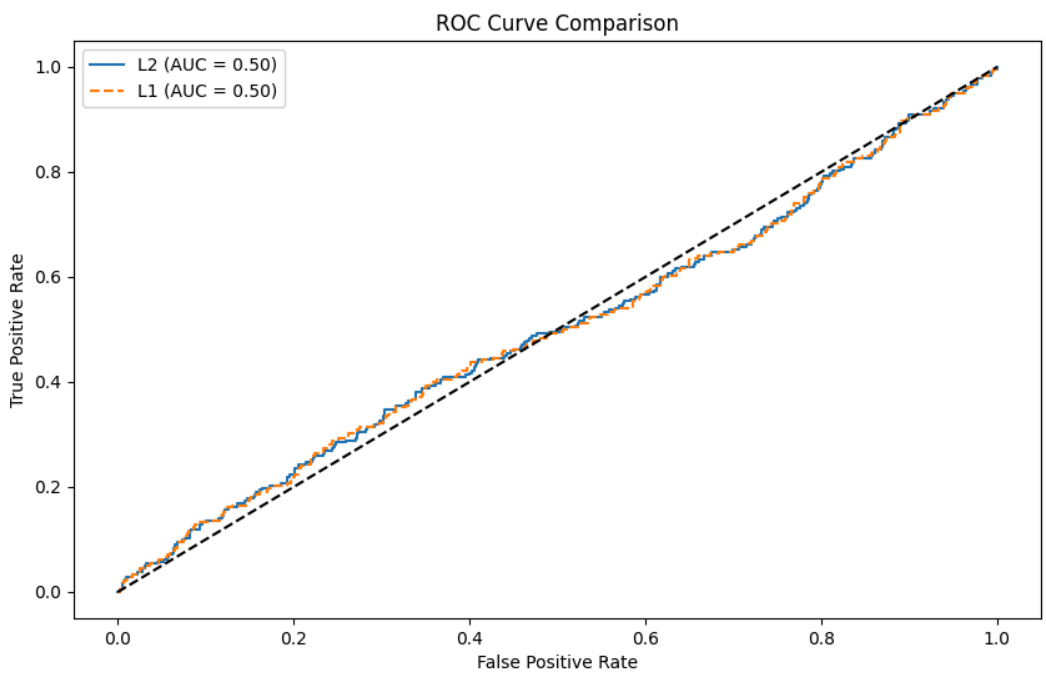

Figure 4: ROC Curve Comparison. This graph compares the True Positive Rate vs. False Positive Rate for both L1 and L2 models, demonstrating the model's discrimination capability.

4.2 Confusion Matrices

To understand the trade-offs between Precision (minimizing false alarms) and Recall (catching actual delays), we analyzed the confusion matrices.

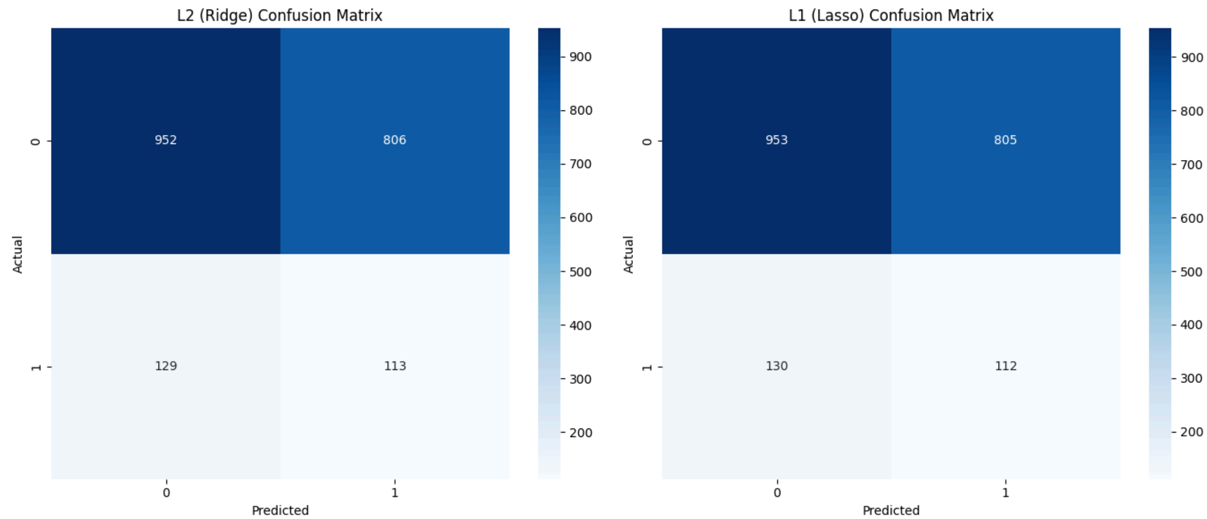

Figure 5: Confusion Matrices. Detailed breakdown of True Positives, False Positives, True Negatives, and False Negatives, validating the effectiveness of the class_weight='balanced' parameter.

---

5. Installation & Usage Guide

5.1 Prerequisites

- Python 3.8+
- Dependencies: pandas, numpy, scikit-learn, matplotlib, seaborn, joblib.

5.2 Setup Instructions

1. **Environment Setup:**
   ```bash
   python -m venv venv
   source venv/bin/activate
   pip install -r requirements.txt
   ```
2. **Data Generation & Training:**
   ```bash
   python generate_data.py
   python airline_delay_prediction.py
   ```
3. **Run Application:**
   ```bash
   python delay_prediction_gui.py
   ```

6. Conclusion

This project demonstrates a sophisticated application of AI by combining statistical rigor with practical utility. The inclusion of comprehensive EDA , feature importance analysis , and rigorous evaluation metrics ensures a transparent and reliable tool for aviation risk assessment.