SkyAlliance

Value Proposition:
Predict Flight Delays

Team 4-2

Safiya Alavi, Katya Aukamp, Ainsley Bock, Monica Martin, Clara Rhoades

Presentation Outline

- Introductions
  - SkyAlliance
  - Our Team
- Abstract
- Exploratory Data Analysis
- Feature Engineering
- Data Processing Pipeline
- Modeling
- Results
  - Top 10 Features
  - Final Model
  - Best Hyperparameters
- Next Steps

Our Team

Katya Aukamp
(Phase Leader)

Safiya Alavi

Monica Martin

Ainsley Bock

Clara Rhoades

What is SkyAlliance?

SkyAlliance

A New Value Proposition for Airlines

- Open to all US-based airlines
- Leverages Data Science and Machine Learning
- Provides unique value:
  - customer experience
  - staff satisfaction
  - airport logistics

Abstract

Fast Fact: flight delays cost. a lot.

- Airlines are estimated to lose $7.5-10 billion annually due to flight delays[1]
- In 2019, the total predicted cost experienced by travelers due to flight delays was $2.4 billion[2]

1. U.S. Passenger Carrier Delay Costs | Airlines For America
2. INVESTIGATING THE COSTS AND ECONOMIC IMPACT OF FLIGHT DELAYS IN THE AVIATION INDUSTRY AND THE POTENTIAL STRATEGIES FOR REDUCTION

Our Approach

SkyAlliance

Our general goal is to provide airlines information to get ahead of the problem: **predictive operations**

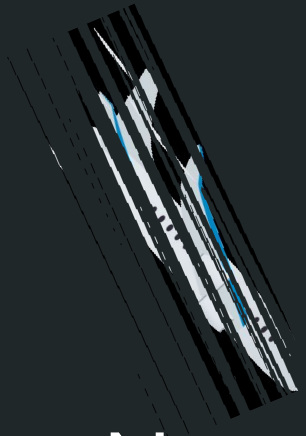
Based on experimentation done by our group we pivoted to predicting binary outcomes: **Delay or No Delay**

Our Results

We trained a variety of models to improve **Recall for Delayed Flights**

Logistic Regression with intricate feature engineering best aligns with SkyAlliance’s goal of reliably identifying delays.

SkyAlliance



Nosedive: Into the Data

Dataset Quick Facts

SkyAlliance

ORIGINAL DATASETS

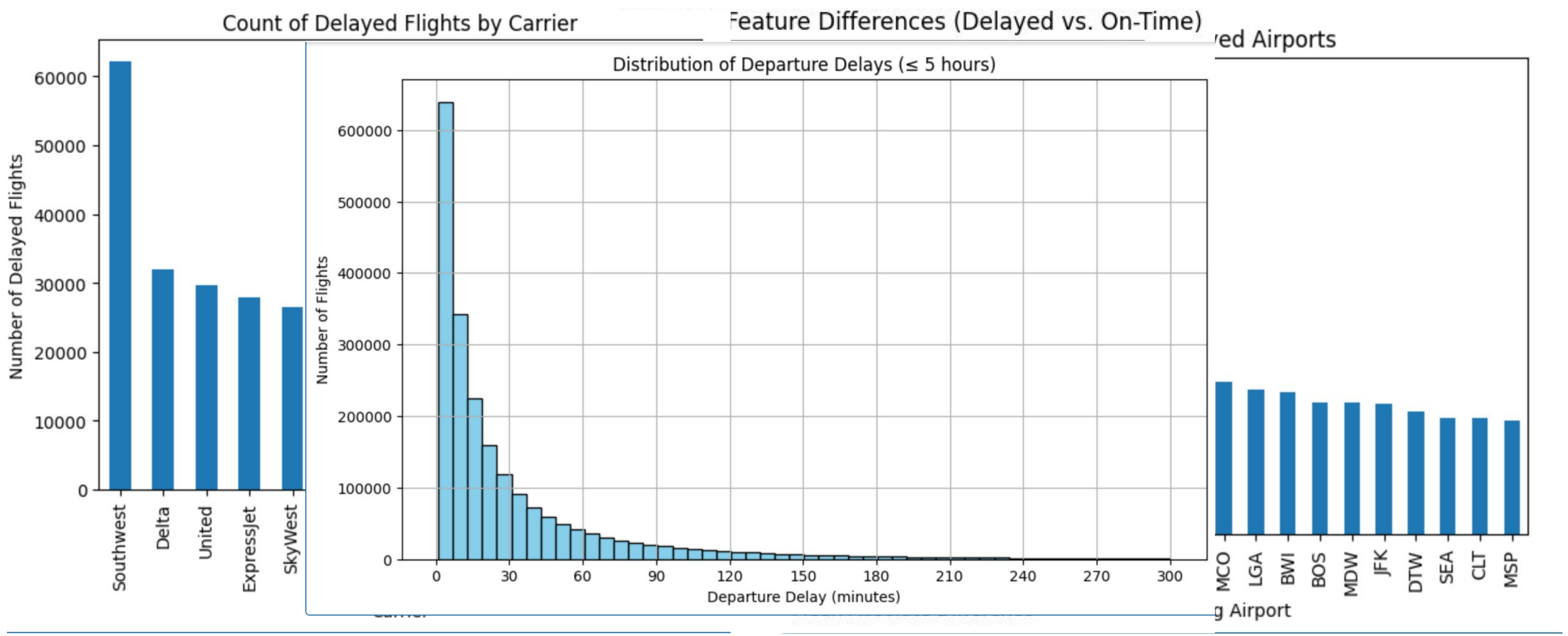
- Carrier On-Time Performance (OTP)
  - **Department of Transportation (DOT)**
- Quality Controlled Local Climatological Data (QCLCD) Publication
  - **National Oceanic and Atmospheric Administration (NOAA)**

SUPPLEMENTAL DATASETS

- Airport Data and Information (FAA-ADIP)
  - **Federal Aviation Administration (FAA)**
- Disaster Declarations Summaries - v2
  - **Federal Emergency Management Agency (FEMA)**
- Annual Airline On-Time Rankings 2003-2024
  - **DOT**
- US Holiday Dates
  - **Kaggle**
- Airport Timezones
  - **Timezone Boundary Builder + OpenStreetMap; Github**

Dataset	Source	Size	Samples	Features	Duplicates
Flights (OTP)	DOT	2.74 GB	74,177,433	109	31,746,844
Weather (QCLCD)	NOAA	32.64 GB	898,983,399	124	0
Airport Data	FAA	0.007077 GB	13,223	106	0
Runway Data	FAA	0.005486 GB	16,389	135	0
Disaster Declarations	FEMA	0.021011 GB	68,417	28	0
Annual Rankings	DOT	0.000002 GB	90	4	0
Holidays	Kaggle	0.000015 GB	342	6	0
Time Zones	GitHub	0.000371 GB	8,876	3	0
Custom Join	SA Team	4.307453 GB	41,557,594	72	0
Cross Validation	SA Team	0.815057 GB	7,999,380	4	0
Training Dataset	SA Team	0.611378 GB	8,007,608	2	0
Test Dataset	SA Team	0.348998 GB	6,861,207	2	0

Flight Delays - High Level 1 Year Dataset Review



Figures based on 12 month dataset

Data Preprocessing

SkyAlliance

- Merged airline carriers that were acquired by others
  - Including Virgin Atlantic and US Airlines
- UTC timestamp conversions for departure and arrival times
- Flagged if the flight typically uses the same aircraft (tail number).

More on Null Handling In Weather and Runway Data

Nulls & Data Cleaning in Weather Data

The weather data was also cleaned after the join to the flight data

Feature	Null Count	Dropped/Kept	Note
station	0	N/A	
date	0	N/A	
HourlyVisibility	32,400	Dropped	Smaller Stations don't always report
HourlyDewPointTemperature	33,161	Dropped	Smaller Stations don't always report
HourlyDryBulbTemperature	26,301	Dropped	Smaller Stations don't always report
HourlyWetBulbTemperature	121,017	Dropped	Smaller Stations don't always report
HourlyRelativeHumidity	33,469	Dropped	Smaller Stations don't always report
HourlyWindSpeed,	27,742	Dropped	Smaller Stations don't always report

Nulls & Data Cleaning in Runway Data

The features that are numeric they were all used for averages so the nulls didn't factor into the calculation

Feature	Null Count	Dropped/Kept	Note
Site_Id	0		
Loc_Id	0		
Runway_Id	0		
Length	0		
Width	0		
Base_Obstacle_Clearance_Slope	9009	Dropped	Not all runways have obstacles
Base LDA	14,983	Dropped	Some smaller airports don't report to this level
Base TORA	14,981	Dropped	Some smaller airports don't report to this level

Feature Importance

We leveraged logistic regression and a custom grid search on regularization hyperparameters to identify important features in the 1 year dataset.

- Elastic Net Tuning
  - 0 - 1 in increments of 0.1
  - 0 = Ridge (L2) Regularization
  - 1 = Lasso (L1) Regularization
- Regularization Strength Tuning
  - Lambda values: [0.001, 0.01, 0.1, 1, 10]

Best Model

- Elastic Net = 0.1
  - Much closer to Ridge than to Lasso
- Lambda = 0.001

Important Features	Coeff
Is there a prior flight?	0.0723
Hourly Visibility	-0.0578
Hourly Wind Speed	-0.0359
Hourly Dew Point Temperature	0.0348
Is the flight scheduled to arrive at night?	-0.0345
Hourly Dry Bulb Temperature	0.0215
Departure hour (cyclically encoded via sine)	0.0145
Avg delays at airport the prev week	-0.0126
Is the flight scheduled to depart at night?	0.0121
Number of runways at airport	0.0120
Unimportant Features:	
Day of week flight departs	-0.0012
repair_service	0.0012
avg_landing_distance_available	-0.0011
Previous year’s OTP percentage	0.001
Is flight scheduled to depart on weekend?	0.0008

SkyAlliance

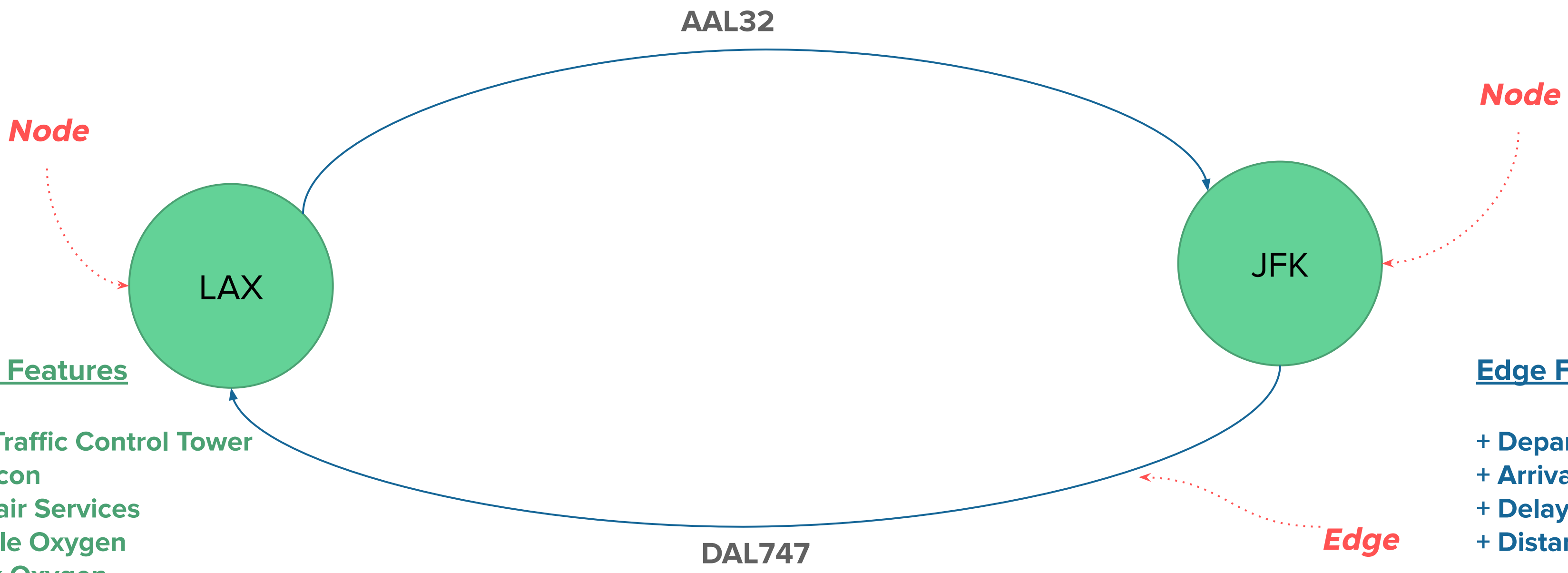
Graph Neural Net

- What we wanted to do was leverage the fact that flights do not operate in isolation

- While relationship could be modeled with specific features we wanted to leverage a Graph to model connections between airports, flights and carriers

- Behind the scenes we used a Graph Neural Network - which turns our flight network graph into information about patterns of delays. **We used this information to enhance our features, not for the purely predictive capabilities.**

Graph Example



Node Features

+ Air Traffic Control Tower
+ Beacon
+ Repair Services
+ Bottle Oxygen
+ Bulk Oxygen
+ Obstacle Clearance
+ Avg Runway Length
+ Avg Landing Distance
+ Avg Take off Distance
+ Number of Runways

Edge Features

+ Departure Time
+ Arrivate Time
+ Delay Bin
+ Distance

Graph Neural Net Usage

- Our models turn relationships into numbers to find patterns
- We used the GNN to look at the data and the network to create a digital profile of the relationships of flights from one airport to another in the form of a 128-dimensional vector
- These 128 numbers aren't random but the learned relationship from the graph, that just plain feature engineering can't do
- We then appended this enriched data to our feature set to run our downstream models on

The remaining features

Engineered:

- Does a prior flight exist
- Average delays at origin last 7 days
- Average arrivals at origin last 7 days
- Average flights from origin last 7 days
- Has there been a FEMA disaster in the state announced in last 5 days
- Is flight date a holiday

Standard Features

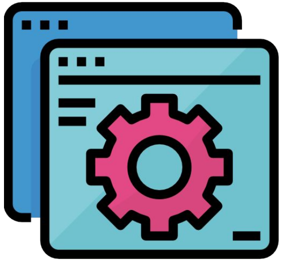
- Hourly Visibility
- Hourly Wind Speed
- Hourly Dew Point Temperature
- Hourly Bulb Temperature
- Hourly Web Bulb Temperature
- Hourly Relative Humidity
- Distance
- Elevation
- Scheduled flight duration
- Flight Date **

** Had to be formatted into cyclical features (sin/cos) to be used in our models

Total Features Used

27 Features that created 128 Pattern Features Generated by Graph

- These are used for pattern detection and don’t mean anything on their own

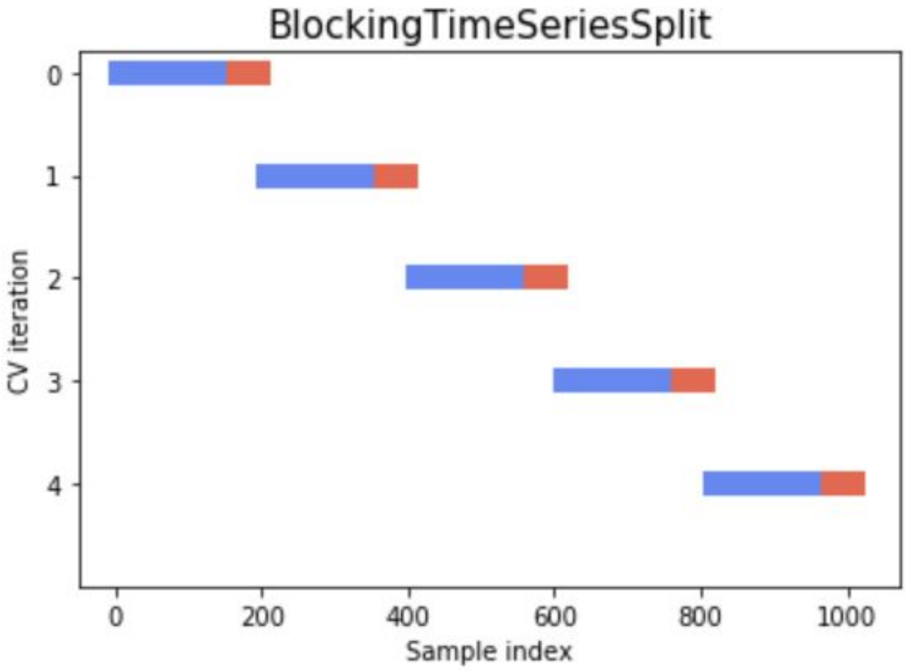


25 Features from datasets and those engineered

52 Input Features || 153 Total Features in Vector

Train/Test Split and Blocked Time Series Cross Validation

- Training Data: 2015-2018 ➔ 7,999,380 examples
- Test: 2019 ➔ 6,861,207 examples
  - 18% delay, 82% no delay
- Blocked Time Series Cross Validation
  - Train ➔ 5 folds (ordered by time)
  - Folds ➔ 80% Train, 20% Val, (ordered by time)
  - Independently scale and OHE each fold
  - Downsample each fold to balance classes
- Train and validate models as normal



Https://datascience.stackexchange.com/questions/116112/what-is-and-why-use-blocked-cross-validation

Preventing Data Leakage

Technique	What We Did	Why it Matters
Chronological Data Splitting	• Trained: four years of historical data • Tested: subsequent, unseen final year	• Simulates real-world forecasting • Prevents learning from future
Isolated Preprocessing	• Fit scalers & GNN on training data only	• Prevents test data contamination
Time-Aware Cross-Validation	• Blocking Time Series Split • 5 sequential folds	• Maintains temporal integrity during tuning

SkyAlliance

Algorithms

Algorithms We Experimented With

- We wanted to try a number of different models to see how they performed and scaled. Our primary goal was speed to result and performance

- Due to these reasons we dropped the Random Forest based models for predictive purposes but instead used it for Grid Search

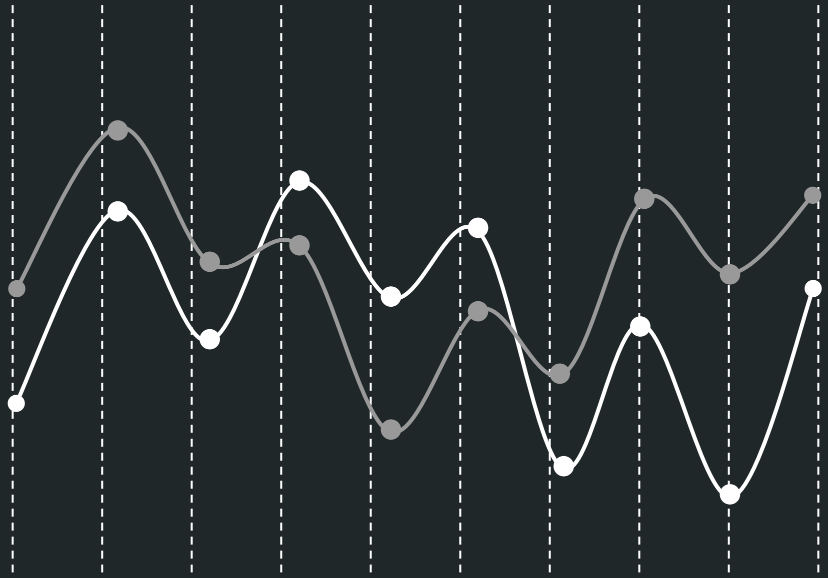
SkyAlliance

Evaluation Metric

WeightedRecall is the metric we will train our models to maximize on.

- Recall is prioritized over precision because the cost of missing a real delay (false negative) is higher than over-predicting one (false positive).
  - Missed delay ➜ insufficient staffing, customer dissatisfaction, and operational disruption.
  - False alarm ➜ some over-preparation but helps ensure readiness and prevents service breakdowns.
- Maximizing recall supports SkyAlliance’s ability to deliver proactive, cost-effective responses, ultimately improving both customer experience and operational resilience.

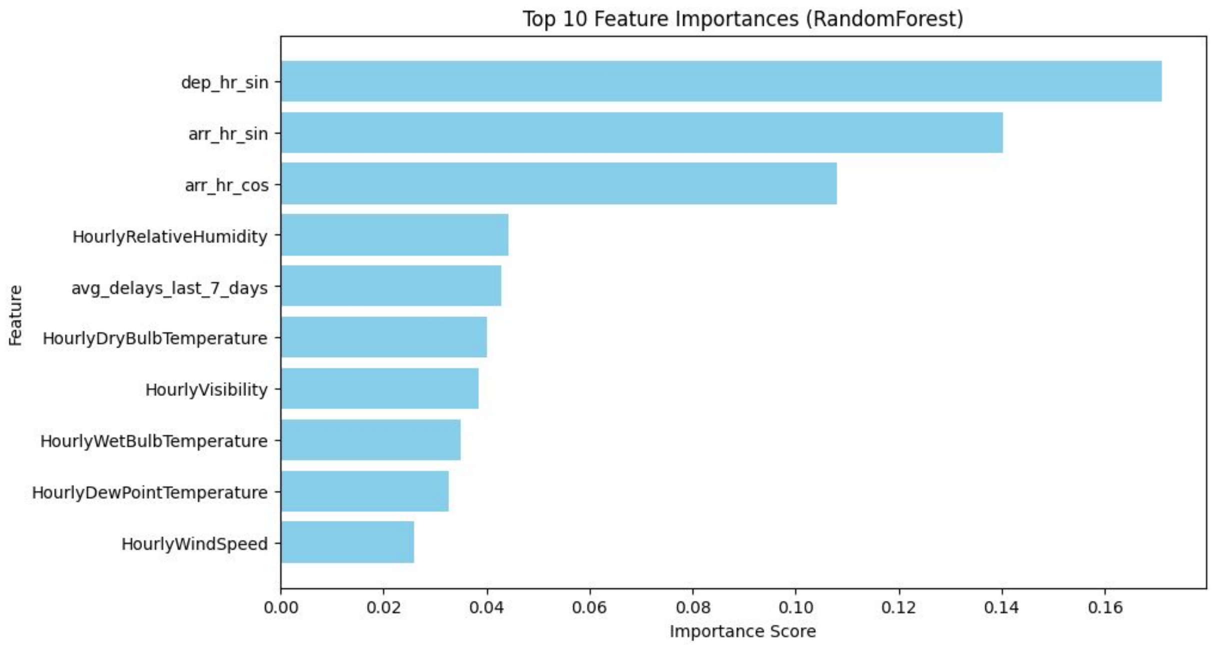
Results



SkyAlliance

Our Best Features

We identified our top 10 best features using a **Grid Search** on a Random Forest Classifier model



Top Performing Feature Families

1. Arrival and Departure Times
2. Weather
3. Airport Performance History

Model Performance Summary

- Delay Recall: Did the model predict every delay, even if some predictions were actually on time?
- Delay Precision: When we predicted a flight would be delayed, how often were we correct?
- Delay F1: How well does the model balance finding every delay without false alarms?

Model	Classification	Recall (train | val | test)	Precision (train | val | test)	F1 (train | val | test)
Logistic Regression (baseline)	No Delay	0.5849 | 0.6021 | 0.6117	0.6304 | 0.6094 | 0.8764	0.6068 | 0.6057 | 0.7205
	Delay	0.6570 | 0.6132 | 0.6125	0.6127 | 0.6059 | 0.2600	0.6341 | 0.6095 | 0.3650
XGBoost	No Delay	0.6307 | 0.6315 | 0.4553	0.6448 | 0.6316 | 0.8976	0.6377 | 0.6265 | 0.6041
	Delay	0.6532 | 0.6137 | 0.7668	0.6392 | 0.6227 | 0.2387	0.6461 | 0.6111 | 0.3640
Feed Forward Neural Net [154, 154, 2]	No Delay	0.1583 | 0.4648 | 0.1655	0.5191 | 0.5231 | 0.8336	0.2426 | 0.3840 | 0.2762
	Delay	0.8533 | 0.5415 | 0.8517	0.5033 | 0.4838 | 0.1852	0.6332 | 0.3683 | 0.3042

Best Model & Real World Interpretation

FFNN | *Trained to meet client priorities*

→ **achieved highest "delay" recall on the test set (85%)**

→ "delay" precision - 18.5% vs 'always guessing delayed' precision - 18.2%

→ poor generalizability

→ unstable deployment to future years

Logistic Regression | ***Recommended for deployment***

→ **61% delay recall, 26% precision:** captures most delayed flights with better precision than always predicting delay — aligned with SkyAlliance's priority to avoid missed delays.

→ **61% "no delay" recall and 88% "no delay" precision :** accurately identifies on-time flights, supporting confident operational planning.

→ **Consistent performance across all data splits**: generalizes well and offers a stable, interpretable solution for future deployment.

Next Steps / Potential Improvements

1. **Revisit Clustering methods to segment the flights:**
   - Looking at all flights in the same way, is too broad
   - Clustering may allow a more comprehensive analysis, and more robust results
2. **Edit Assumptions & Sources of Data**
   - Assess possibility of knowing about delays 1 hour ahead instead of 2 hours?
   - Utilize more advanced data sources with specialized, real-time information
     - Staffing and crew information
     - Airport congestion
3. **Limit the Scope by Route or Region**
   - Limiting to certain airports or regions ⇒ a more specialized model ⇒ likely better performance because the level of generalization is more limited
4. **Ensemble Modeling**
   - Combine predictions from LR, XGBoost, and FFNN to leverage complementary strengths

Thank You!

SkyAlliance

Feed Forward Neural Net

- Only guesses the majority class
  - Fails to outperform baseline
  - Attempted Input Layers: [5610, 32, 8, 4], [5610, 256, 32, 4], [5610, 16, 8, 4]
  - Long training times

Train

	Class	Precision	Recall	F1 Score
0	no delay	0.609765	1.0	0.757583
1	small delay	0.000000	0.0	0.000000
2	medium delay	0.000000	0.0	0.000000
3	large delay	0.000000	0.0	0.000000

Test

	Class	Precision	Recall	F1 Score
0	no delay	0.671471	1.0	0.803449
1	small delay	0.000000	0.0	0.000000
2	medium delay	0.000000	0.0	0.000000
3	large delay	0.000000	0.0	0.000000

References

- ChatGPT: SkyAlliance logos, quirky section titles, light editing
- US Passenger Carrier Delay Costs:
- INVESTIGATING THE COSTS AND ECONOMIC IMPACT OF FLIGHT INVESTIGATING THE COSTS AND ECONOMIC IMPACT OF FLIGHT DELAYS IN THE AVIATION INDUSTRY AND THE POTENTIAL DELAYS IN THE AVIATION INDUSTRY AND THE POTENTIAL STRATEGIES FOR REDUCTION STRATEGIES FOR REDUCTION
- BTS TranStats: Airline On-Time Statistics and Delay Causes
- Investigating the Costs and Economic Impact of Flight Delays

Backup Slides

Baseline Model

Our baseline model always predicts the most frequent class: no delay

Our baseline model’s unweighted recall is 0.25 because it has perfect recall for `no delay` and 0 recall for small, medium, and big delay.

Averaged over each class, this leads to a recall of 0.25.

```
model_name: baseline_model

recall: 0.25

precision: 0.15027762784747467

f1: 0.45135403750457836

accuracy: 0.6011105113898987
```

Logistic Regression - Test Dataset Findings

Model	Notes	Unweighted Recall	Unweighted Precision	F1 Score	Accuracy
Baseline		0.25	0.17	0.54	0.67
Vanilla Logistic Regression		0.258	0.33	0.55	0.67
L2 Log Regression	lambda = 1.0 100 epochs	0.25	0.23	0.54	0.67
L1 Log Regression	lambda = 1.0 100 epochs	0.25	0.17	0.54	0.67

Per Class Metrics for Vanilla Logistic Regression:

	Class	Precision	Recall	F1 Score
0	no delay	0.677310	0.980953	0.801332
1	small delay	0.324004	0.044811	0.078733
2	medium delay	0.222863	0.004322	0.008479
3	large delay	0.090487	0.000474	0.000943

SkyAlliance

XGBoost Classifier

Background:

XGBoost can naturally capture complex nonlinear relationships and feature interactions, leading to better performance on real-world data.

Model Hyperparameter Tuning on the 1 year 2015 dataset:

Below are the various hyperparameters tested and the optimal configuration was selected using the **average WeightedRecall across 5 cross validation folds spanning between Jan 1, 2015 and August 31, 2015**:

- Max_depth: [**8**] ← This was informed from previous Phase 2 experimentation
- Learning_rate: [**0.05**, 0.07] → Step size shrinkage
- Subsample: [0.75, **0.95**] → Fraction of rows sampled for each tree
- Gamma: [0.0, **2.5,** 5.0] → Minimum loss reduction to make a further split
- Reg_alpha: [0.15, 0.5, **0.95**] → L1 regularization term on weights
- Reg_lambda: [**2.5**, 5, 7.5] → L2 regularization term on weights

XGBoost Classifier

Optimal Model Configuration Trained and Tested on 5 year dataset:

- Trained using 2015-2018 data
- Tested on the 2019 data

Evaluation on the Training Set:

- **Yields Weighted Recall of 0.6419**
- **Yields Weighted Precision of 0.6420**
- **Yields Weighted F1 Score of 0.6420**

Label	Recall	Precision	F1
No Delay	0.6307	0.6448	0.6377
Delay	0.6532	0.6392	0.6461

Evaluation on the Testing Set:

- **Yields Weighted Recall of 0.6111**
- **Yields Weighted Precision of 0.5682**
- **Yields Weighted F1 Score of 0.4841**

Label	Recall	Precision	F1
No Delay	0.4553	0.8976	0.6041
Delay	0.7668	0.2387	0.3640

- Training set: **Balanced recall and precision** across both classes (~63–65%).
- Test set: Model **correctly identifies 77% of actual delays**, supporting SkyAlliance's goal of proactive disruption management.
- Low delay precision (24%): **Model often predicts delays that don't occur, leading to over-preparation**—but ensures SkyAlliance is **rarely underprepared**.
- No Delay recall drops to 45%, while precision rises to 90%: Model is **highly cautious when predicting on-time flights**, reducing the risk of false reassurance.

SkyAlliance

Random Forest Classifier

Key Insights (Generally):

- Reduces Overfitting
- Performance is consistent between Train and Test
- Insights into feature Importance

Performance Baseline Results:

- Biased towards “No-Delay” class
- Performance is consistent between Train and Test
- Low recall scores for the delay categories

Train:
* Default Parameters

	label	recall	precision	f1
1	no delay	0.7990844983298281	0.28744375763453667	0.4227995905427566
2	small delay	0.24816283579882945	0.2762805648475812	0.261467942447823
3	large delay	0.0118674654648052...	0.3434972822484423	0.0229422989228364...
4	medium delay	0.0475776037133739...	0.41391494552643754	0.0853451752655111

Test:
* Default Parameters

	label	recall	precision	f1
1	no delay	0.8012540099154273	0.23636018223463245	0.36503846079324176
2	small delay	0.27575938058368077	0.2612696690179056	0.26831904958263214
3	large delay	0.0131453397486651...	0.32993630573248406	0.0252833393531760...
4	medium delay	0.0182935043241641...	0.4615907545887152	0.03519228775785218

SkyAlliance

Final Selection

Logistic Regression

First model with multi-class logistic regression with L2 regularization

Grid Search on Random Forest

Systematic test of different combinations of model settings to find the combo that gives the best predictive performance

XGBoost

A high-performance gradient boosting algorithm that builds an ensemble of decision trees to improve predictive accuracy

Feed Forward Neural Net

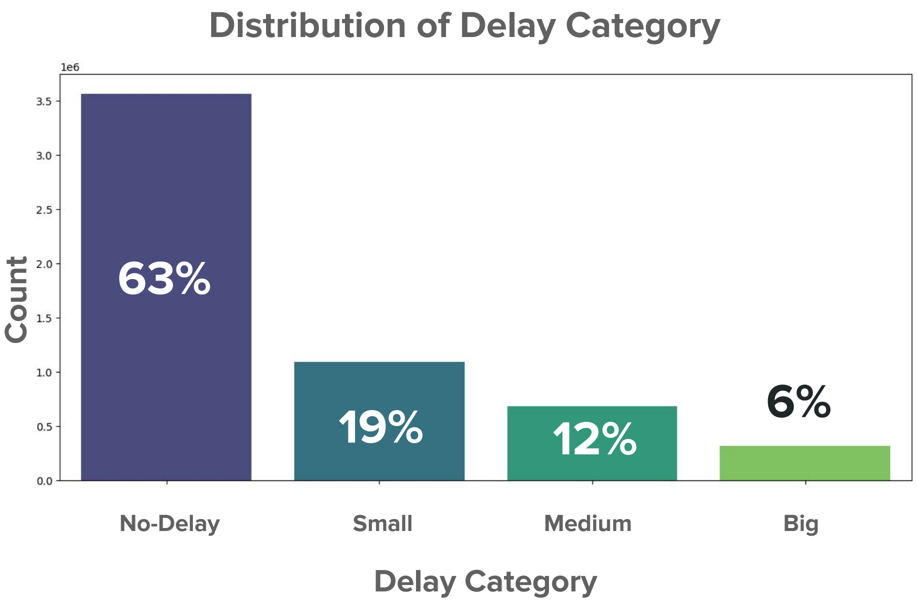
A deep-learning algorithm to look for non-linear behaviors

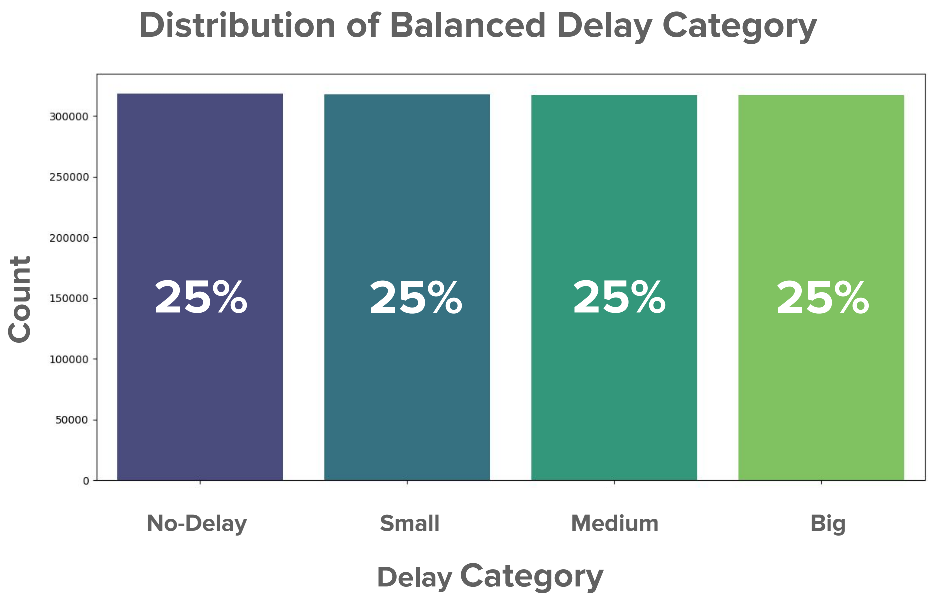
All models are evaluated using blocked cross validation as described previously.

Next Steps

Data Balance Experiments:

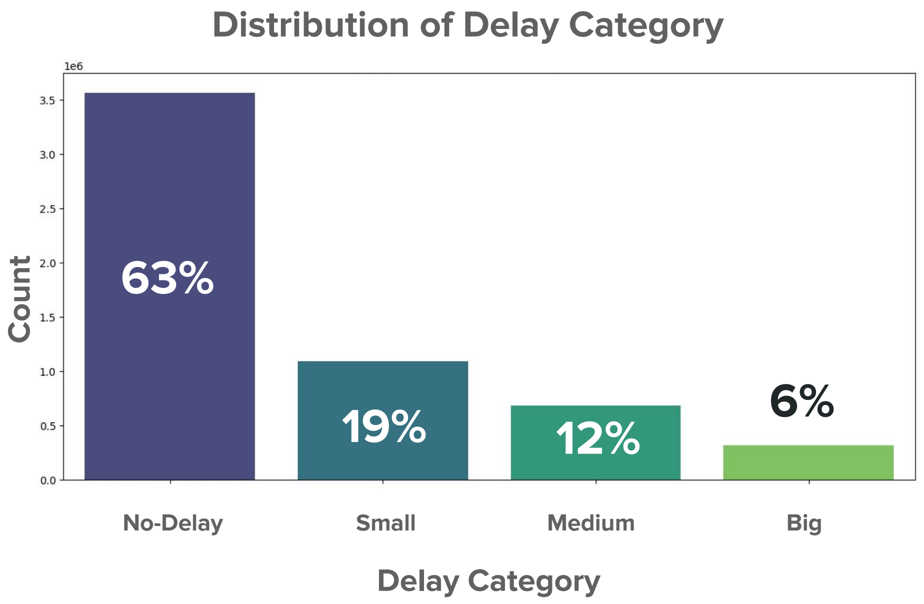
Method 1: DownSample **<u>all classes</u>** to **minority** class size

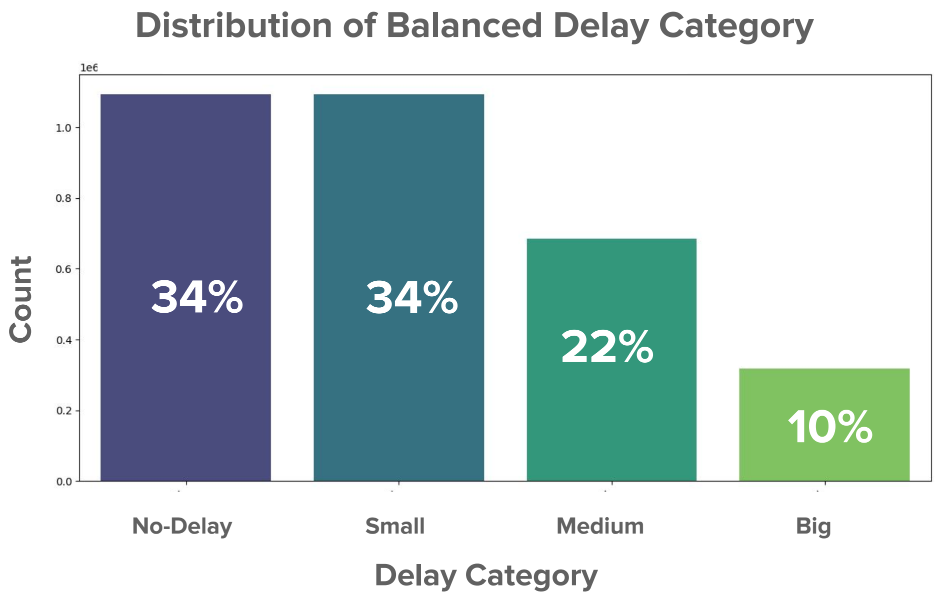




Data Balance Experiments:

Method 2: Down Sample **majority only** class to the **2nd largest** class size





Results : 4 Year 2015-2018 Training Set

Below is a table of the unweighted recall, precision and F1 across the no delay and delay classes.

The model with the best results on the training set in regard to recall for the delay class is the Feed Forward Neural Network. Although, the XGBoost model has a more balanced recall across the delay and no delay classes.

Baseline model

Model	Classification	Recall	Precision	F1
Logistic Regression	No Delay	0.5849	0.6304	0.6068
	Delay	0.6570	0.6127	0.6341
XGBoost	No Delay	0.6307	0.6448	0.6377
	Delay	0.6532	0.6392	0.6461
Feed Forward Neural Net [154, 154, 2]	No Delay	0.1583	0.5191	0.2426
	Delay	0.8533	0.5033	0.6332

Results : Cross Validation Results using 5-fold CV Made With 4 Year 2015-2018 Training Set

The model with the best results on the cross validation folds is the XGBoost, which indicated that this may have been the best candidate for the testing set, as it seemingly generalized well to the unseen data.

Baseline model

Model	Classification	Recall	Precision	F1
Logistic Regression	No Delay	0.6021	0.6094	0.6057
	Delay	0.6132	0.6059	0.6095
XGBoost	No Delay	0.6315	0.6316	0.6265
	Delay	0.6137	0.6227	0.6111
Feed Forward Neural Net [154, 154, 2]	No Delay	0.4648	0.5231	0.3840
	Delay	0.5415	0.4838	0.3683

Results : 1 Year 2019 **Test Set**

Feedforward Neural Network (FFNN) achieved the highest recall on the Delay class, correctly identifying 85% of actual delays. This performance suggests that SkyAlliance will rarely be underprepared when a delay is truly expected → supporting the goal of proactive operational readiness.

Model is highly conservative when predicting No Delay, doing so only 17% of the time, but with high precision. → When the model does predict a flight will depart on time, SkyAlliance can trust that prediction and avoid allocating unnecessary resources.

Baseline model

Model	Classification	Recall	Precision	F1
Logistic Regression	No Delay	0.6117	0.8764	0.7205
	Delay	0.6125	0.2600	0.3650
XGBoost	No Delay	0.4553	0.8976	0.6041
	Delay	0.7668	0.2387	0.3640
Feed Forward Neural Net [154, 154, 2]	No Delay	0.1655	0.8336	0.2762
	Delay	0.8517	0.1852	0.3042

Model Name

Background:

Opt to include one sentence background on the model

Initial Model Parameter Tuning:

Below are the various hyperparameters tested and the optimal configuration was selected using the **average WeightedRecall across 5 cross validation folds spanning between Jan 1, 2015 and August 31, 2015**:

- [Fill in the different hyperparemeters tested]

Model Name

Optimal Model Configuration Trained and Tested on 5 year dataset:

- Trained using 2015-2018 data
- Tested on the 2019 data

Evaluation on the Training Set:

- **Yields Weighted Recall of 0.__**
- **Yields Weighted Precision of 0.__**
- **Yields Weighted F1 Score of 0.__**

Label	Recall	Precision	F1
No Delay			
Delay			

- Discuss results here

Evaluation on the Testing Set:

- **Yields Weighted Recall of 0.__**
- **Yields Weighted Precision of 0.__**
- **Yields Weighted F1 Score of 0.__**

Label	Recall	Precision	F1
No Delay			
Delay			

Feature Selection & Data Sources

SkyAlliance

- To enhance the basic information we already had we included new sources
- FAA data to get facilities and maintenance information on airports
- Timezone data to standardize everything
- FEMA data to see if major disruptions happened
- Holiday schedule for higher travel days

Best Model & Real World Interpretation

FFNN – trained per client's request

➔ **achieved highest "delay" recall on the test set (85%)**
➔ "delay" precision - 18.5% vs 'always guessing delayed' precision - 18.2%
➔ poor generalizability
➔ unstable deployment to future years 😢

Logistic Regression – overall best model for SkyAlliance to deploy at current stage

➔ 61% "delay" recall and 26% "delay" precision
- **flags most delayed flights**, at a higher precision than just always guessing a delay
- acceptable for SkyAlliance, where **not catching a delay is more costly than a false alarm**

➔ 61% "no delay" recall and 88% "no delay" precision
- **confidently and correctly flags most on-time flights**
- enables reliable operational planning

➔ consistent recall across the training, validation and test set
- generalizes well to future data
- **deployment to future years will be stable**
- model is highly interpretable to stakeholders