Flight Delay Prediction

Team 10:

- YuLing Chen
- Lesley Matheson
- Mitchell Li
- Ernesto Oropeza

Agenda

- **Problem Statement**
- **EDA**
- **Feature engineering**
- **Algorithm**
- **Implementation**
- **Conclusions**

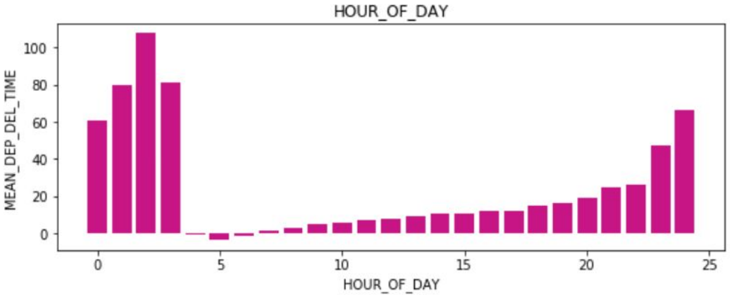
Problem Statement/Business Cases

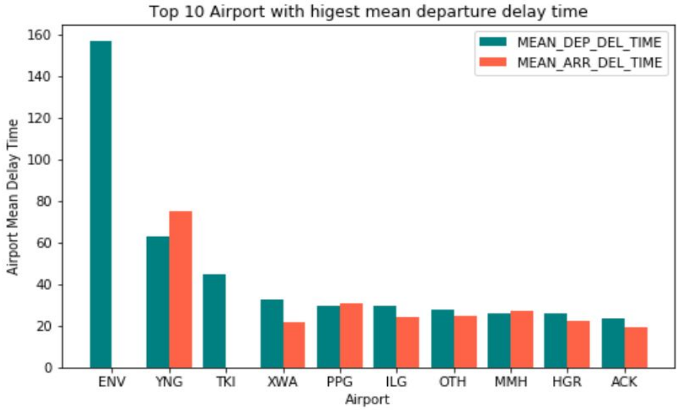
- Impact of Predicting Flight Delays:
    - Airline industry financial incentives
        - $250B domestic airline industry with upward $8B annually
    - Customer financial impacts and inconvenience
    - Operational efficiency for airports

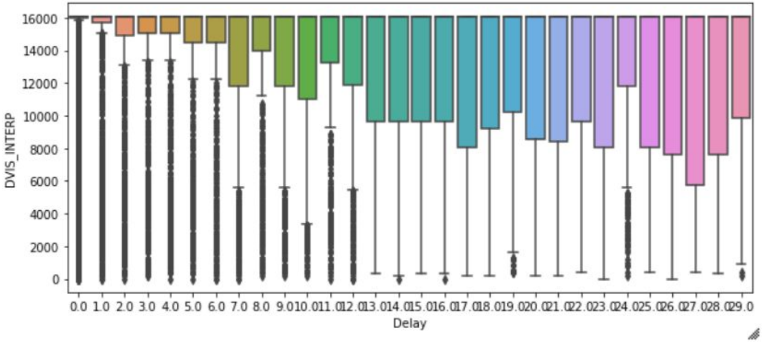
- Model objective:
    - Predict whether a given scheduled flight will be delayed by more than 15 minutes 2 hours in advance at a given departure time
    - Binary classification problem

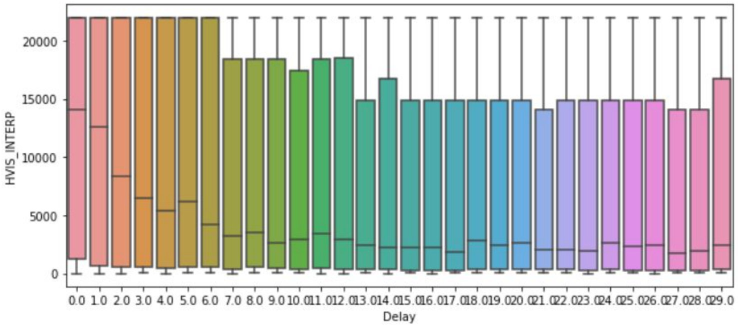
EDA - Summary

- Temporal impact on departure delay
- Airport and Airline impact on departure delay
- Weather impact on departure delay









EDA - DataSet

- 31,746,841 flights(rows)
- 109 flight features (columns)
- 371 airports
- 19 airlines

- 630,904,436 measurements (rows)
- 177 metrics per station(columns)
- 15,195 distinct stations

- 29,771 rows
- 11 columns
- 25,744 distinct stations

- Large dataset
- Should fully leverage Spark Parallel Processing Capabilities
  - Especially in joins

- For EDA scalability strategy:
  - Use sampling whenever it makes sense

Data Imbalance and Null handling

Data Imbalance

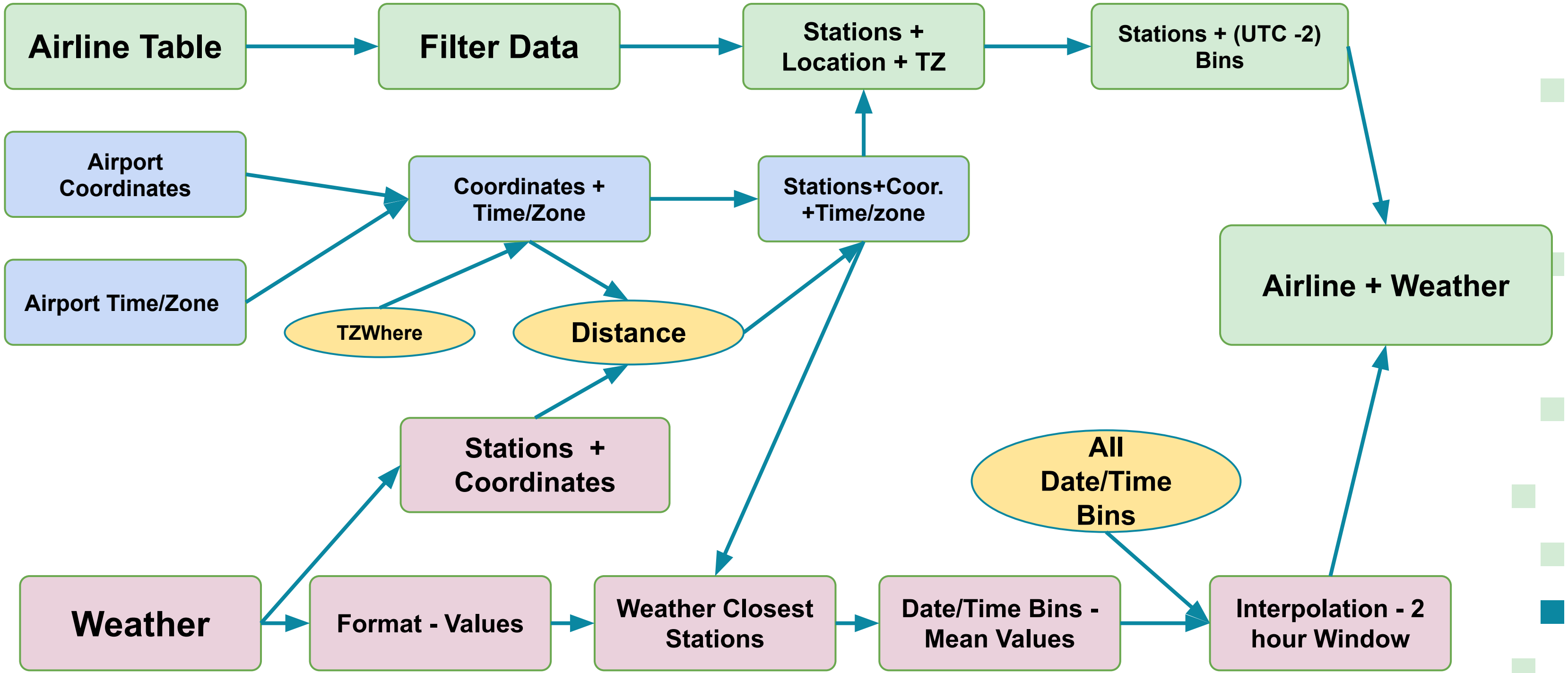
- Over sample the minority class (delays)
- Under sample the majority class (no delays)
- Class Weights

Null Handling

- Used the HyperParameter "HandleInvalid" in the ML Pipeline Vector Assembler.

About 20% delays and 80% on schedule

Feature Engineering – Workflow



Engineered Features

Airline + Weather

Pre Train/Test Data Split:

- Scheduled Flights per Aircraft/Day
- Minimum Layover per Aircraft/Day
- Number of Departures per Airport/Day
- Hour of the Day
- Prior Departure delays (2 hours)
- Prior Arrival Delays

Airline + Weather Training

- Post Train/Test Data Split:
- Average Airport Delay
- Percentage of Flights Delayed per Airport
- Average Carrier Delay
- Percentage of Flights Delayed per Carrier

Complete Feature Set

Flight Airline Table Features

- DAY_OF_WEEK
- MONTH
- QUARTER

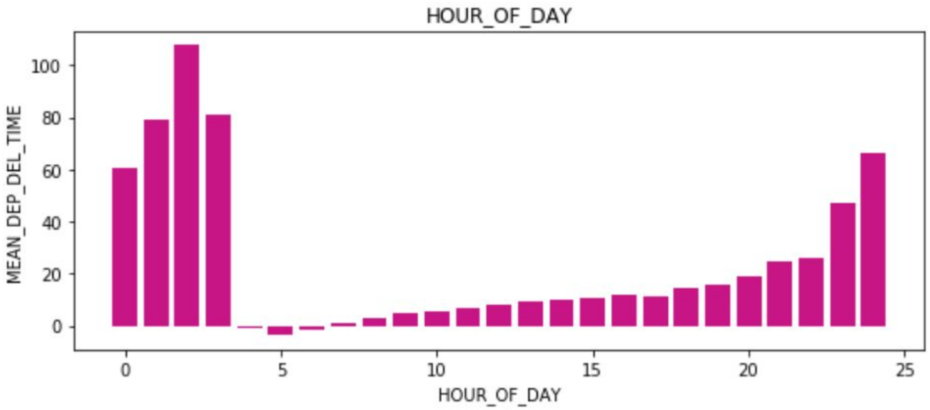
Weather Features

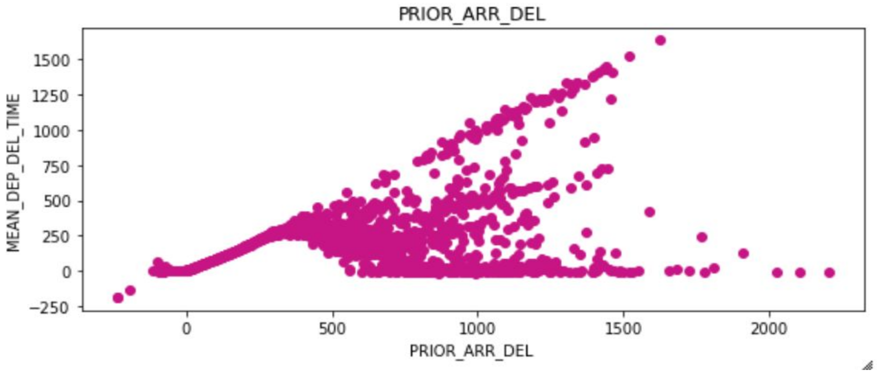
- TEMPERATURE_INTERP
- DVIS_INTERP
- HVIS_INTERP
- WVEL_INTERP
- WDIR_INTERP
- DEWPOINT_INTERP

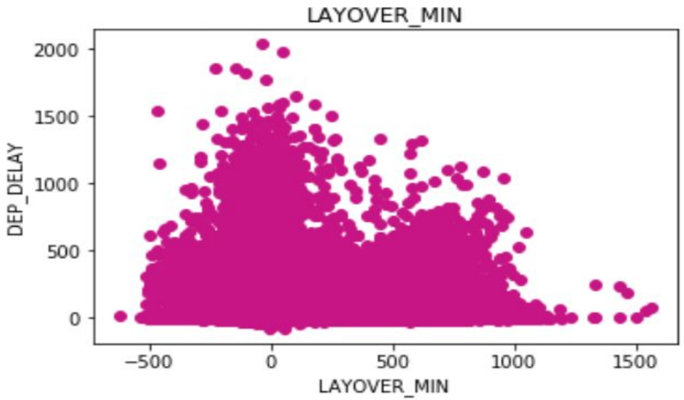
Engineered Features:

- AIRPORT_AVERAGE_DELAY_MINS
- AIRPORT_PERCENTAGE_DELAY
- CARRIER_AVERAGE_DELAY_MINS
- CARRIER_PERCENTAGE_DELAY
- TOTAL_FLY_DAY
- LAYOVER_MIN
- FLY_AIRPORT_DAY
- TOTAL_FLIGHTS
- HOUR_OF_DAY
- PRIOR_ARR_DEL
- PRIOR_DEP_DEL

Initial Performance of Engineered Features



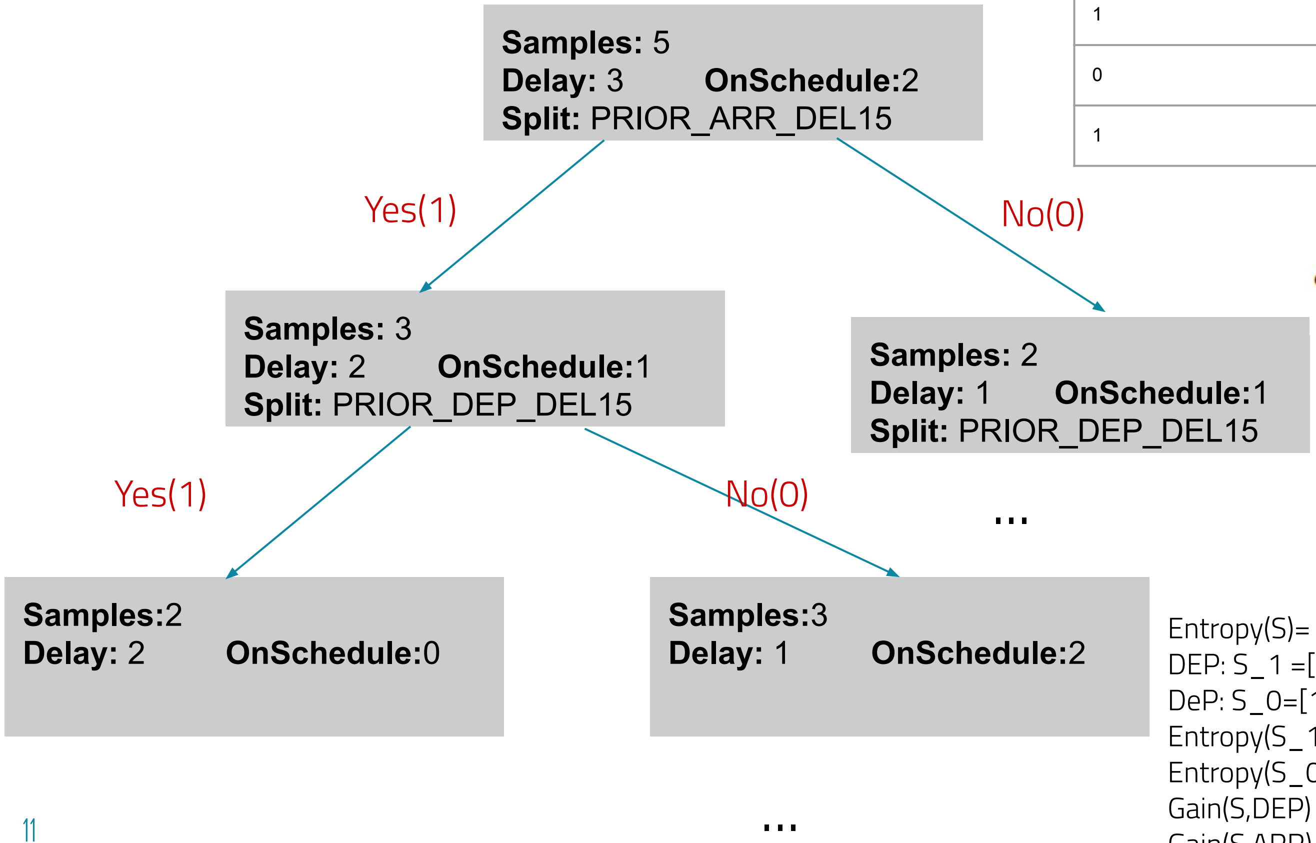




Toy Example

Two features:
Prior_dep_del15
Prior_arr_del15

PRIOR_ARR_DEL15	PRIOR_DEP_DEL15	DEP_DEL15
0	0	1
1	1	1
1	0	0
0	0	0
1	1	1



$$Entropy(S) = -(P_{\oplus} log_2 P_{\oplus} + P_{\ominus} log_2 P_{\ominus})$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Entropy(S)= -((2/5)log(2/5)+(3/5) log(3/5)) =0.97
DEP: S_1 =[1+, 1-]
DeP: S_0=[1+,2-]
Entropy(S_1) = -((1/2)log(1/2) +(1/2) log(1/2))=1
Entropy(S_0)=-((1/2)log(1/2)+(2/2)log(2/2))=0.93
Gain(S,DEP) = 0.97-((2/5)*1+(3/5)*0.93))=**0.012**
Gain(S,ARR) = **0.054**

Algorithms Tried and Lessons Learned

Models Tried

- Logistic Regression
- Decision Tree
- Random Forest
- Gradient Boost Tree

Lessons Learned

- A correctly engineered feature greatly boosts model performance.
- HyperParameter tuning using Cross Validation but challenged by resources.
- Tree type models are superior compared with Logistic Regression for nonlinear problems

Evaluation

$$\textbf{Precision} = \frac{TP}{(TP + FP)}$$

$$\textbf{Recall} = \frac{TP}{(TP + FN)}$$

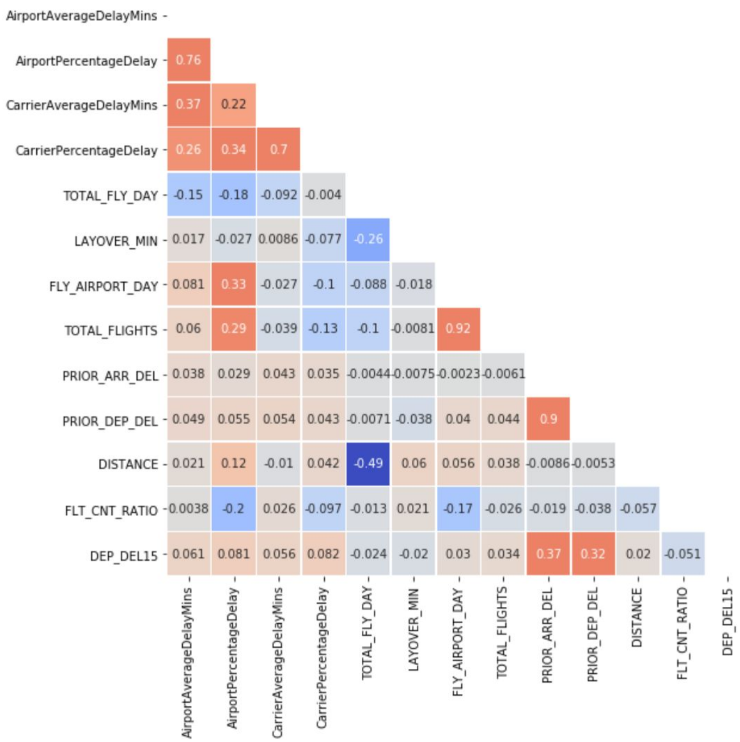
$$\textbf{F1-Score} = \frac{TP}{(TP + \frac{1}{2}(FN + TP))}$$

Model Performance Evaluation Output

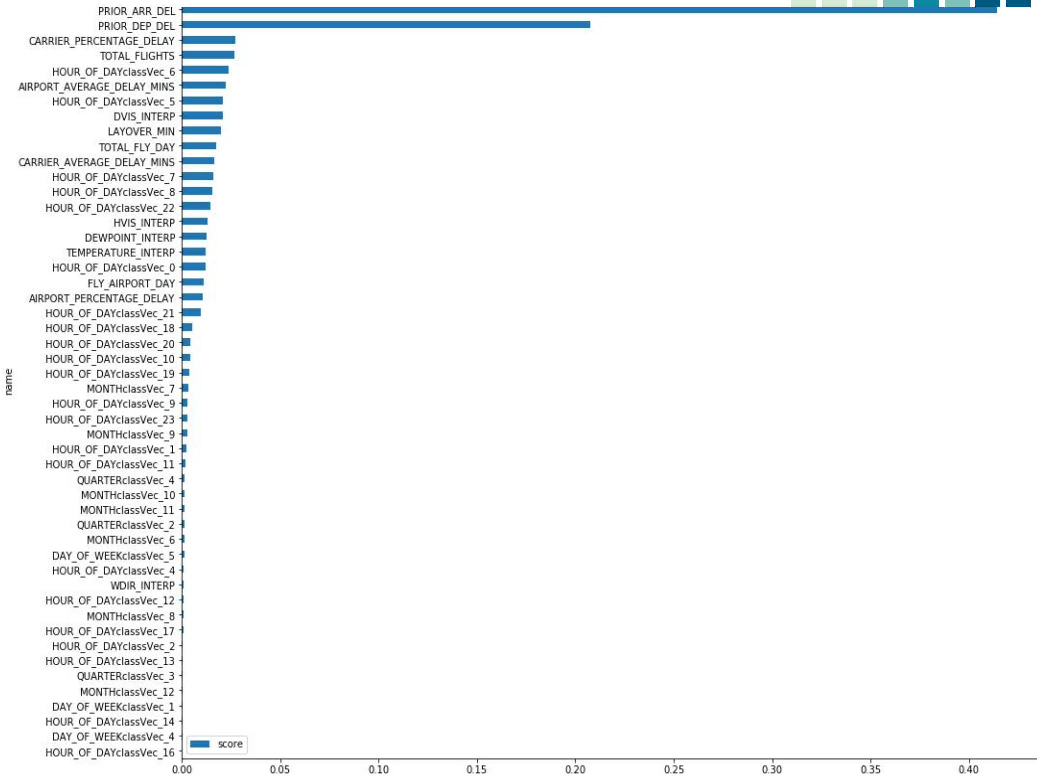
	precision	recall	f1-score	support
0.0	0.88	0.96	0.91	5575213
1.0	0.68	0.41	0.52	1292542
accuracy			0.85	6867755
macro avg	0.78	0.68	0.71	6867755
weighted avg	0.84	0.85	0.84	6867755

Feature Correlation and Importance

Feature Correlation



Feature Importance



Performance and Scalability

- Table Joins
  - Avoid looping table joins
  - Apply broadcast join to boost the performance
  - Write tables containing precomputable values and join incrementally
  - Utilize caching

- Model phase
  - Pyspark native ML models were used in training and testing the models

Conclusions and Future Work

- Tree type models are good fit in predicting flight delays
- Feature engineering is key to achieve good model performance
- Future efforts should be spent in the following areas:
  - Creative feature engineering
  - Data enrichment
    - Integrating better avionic radar based weather data, such as Storm Events DB by NOAA
  - Deep learning models including Neural Nets and LSTM

THANK YOU!

Any questions?