Predicting Flight Delays at Scale

Kieffer Thomas, Ren Tu, Dan Ortiz, Dan Weitz

DataSci W261 - Summer 2021

Berkeley
UNIVERSITY OF CALIFORNIA

Agenda

- Question Formulation
- Join and EDA
- Feature Selection and Engineering
- Homebrew Model
- Scale Model

Berkeley
UNIVERSITY OF CALIFORNIA

Background & Question Formulation

Background

- Flight delays result in major costs to carriers and consumers
  - FAA estimate delay costs of $33 Billion; Includes lost time, operational expenditures, and externalities for non-airline sectors.
- Predicting flight delays can help carriers respond to delays and reduce associated costs
- Use open data to create a model to predict flight delays:
  - Delay of at least 15 minutes beyond scheduled departure time - binary variable
    - Weather Data
    - Flight Data

Metric Selection

- Metric: F1 Score - Harmonic mean of Precision & Recall
  - Precision (percentage of actual delays among all predicted delays)
  - Recall (percentage of all delays that were predicted by the model)

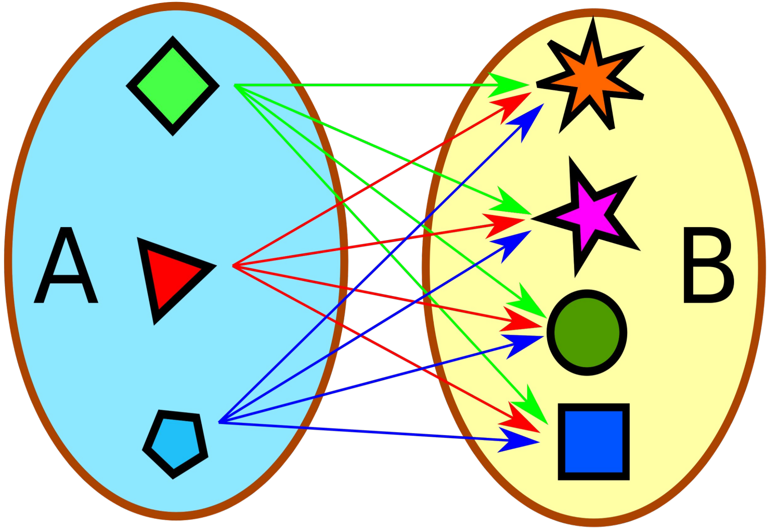
- Mistakenly predicting On-Time departures as delays and predicting delays as on-time departures both incur cost
- F1-Score penalizes both of these conditions in a single measure

Berkeley
UNIVERSITY OF CALIFORNIA

EDA and Data Join

Join - Smaller 3-Month Dataset

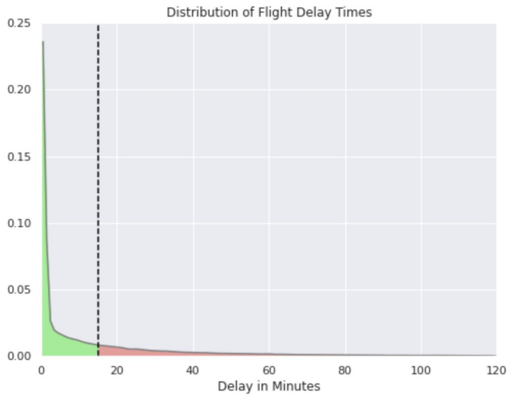
1. Unique airport codes
2. Join to stations using K-code/P-code/state
3. Get smaller weather table by joining on station ID
4. Full join to flights table at the departure and arrival airports
5. Find weather closest to 2 hours before departure



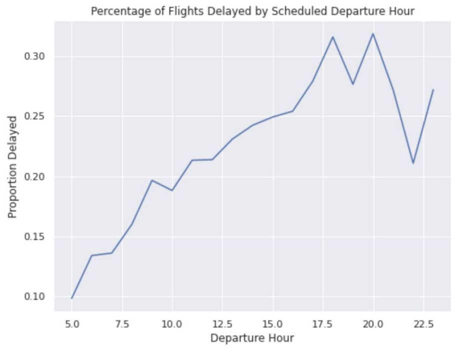
Berkeley
UNIVERSITY OF CALIFORNIA

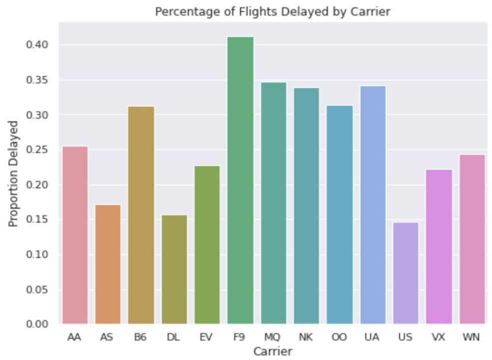
Exploratory Data Analysis

- 23% of flights delayed (excluding cancelled flights)
  - Imbalanced Class, long tail/logarithmic distribution in delay times



- Suggested strong relationships between a few categorical (e.g. flight carrier) and temporal variables (scheduled departure time) and outcome

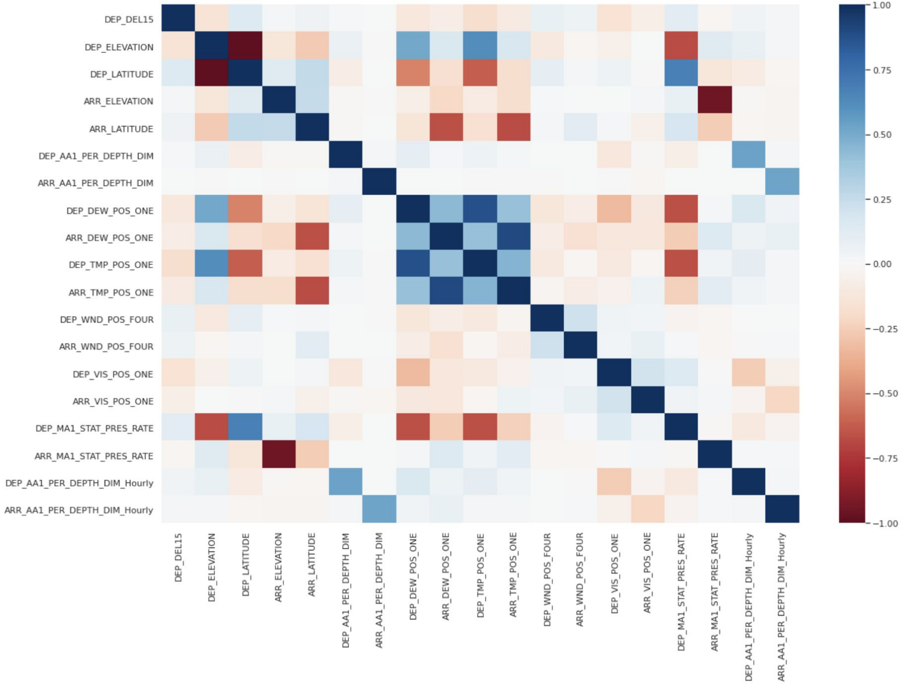




Berkeley
UNIVERSITY OF CALIFORNIA

Exploratory Data Analysis

- Correlations between weather variables and the target variable are generally weak, and many of the weather features appear somewhat collinear



Feature Engineering

- Delay Information from Flights Data:
  - Airport delay rate/probability (cumulative to previous day)
  - Rolling 3 hr and 6 hr window airport delay rates (up to 2 hrs before scheduled departure)
  - Airline delay rate (cumulative to previous day)
  - Tail number experienced delay previously on same day (binary)

- Airport PageRank:
  - Measure of airport hub “importance”, using number flights into an airport as weight for links to that airport

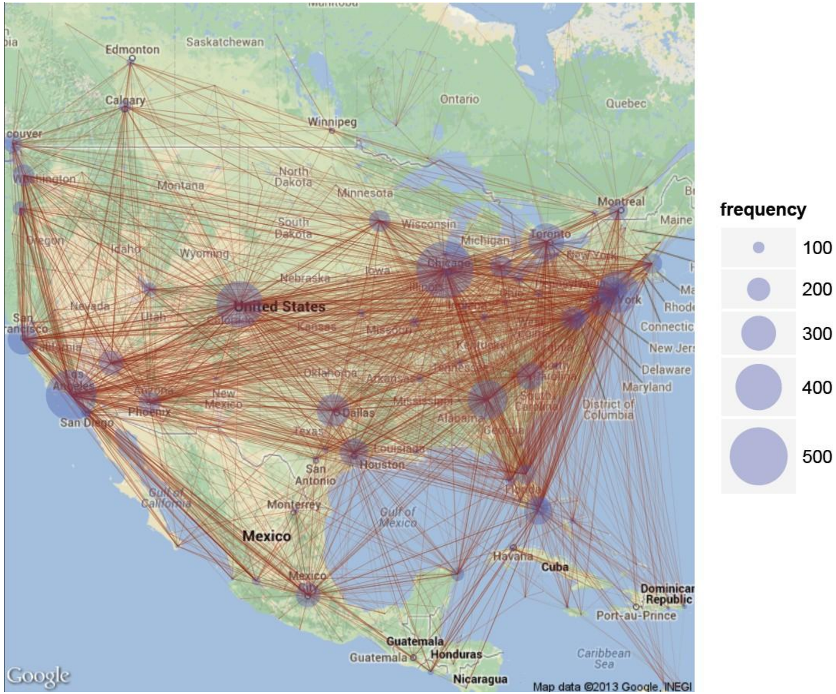
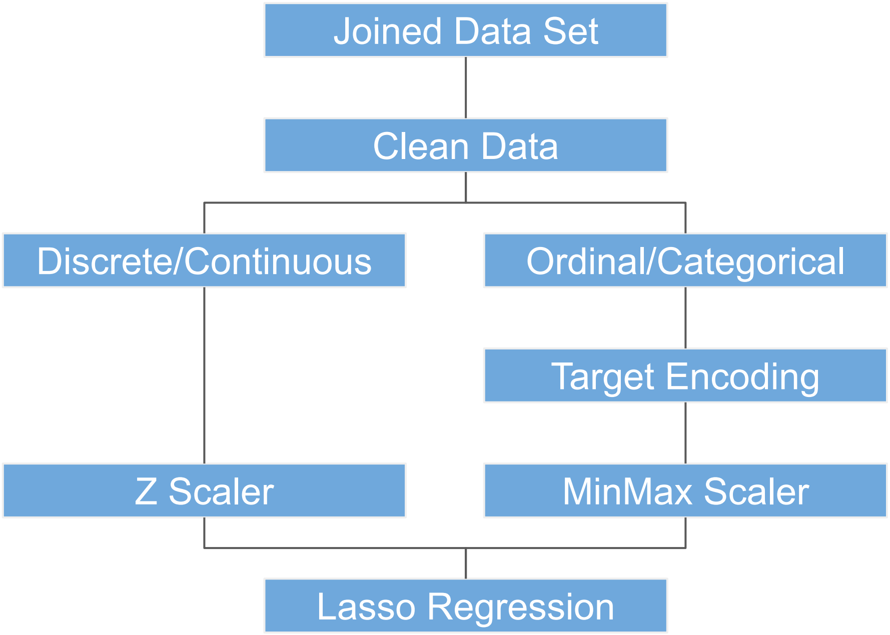


Image: http://proc-x.com/2013/06/the-us-airports-with-most-flight-routes/

Feature Selection

Preprocessing

- Clean data
  - Eliminate attributes missing more than 80% of its data
  - Eliminate missing rows
  - Result 112819/141285 rows remain



Feature Selection with Lasso Regression

```
#Source
https://medium.com/@sabarirajan.kumarappan/fe
ature-selection-by-lasso-and-ridge-regression
-python-code-examples-1e8ab451b94b

sel_ =
SelectFromModel(LogisticRegression(C=1,
penalty='l1', solver='liblinear'))

sel_.fit(train_x,
np.ravel(train_y,order='C'))

sel_.get_support()

train_x = pd.DataFrame(train_x)
```

	Features	Impact
4	TAIL_NUM	4.663927
0	OP_CARRIER_FL_NUM	4.619650
29	TODAY_PREV_DELAY_SAME_TAIL	4.228209
59	DEP_WND_POS_TWO	3.146258
3	OP_CARRIER	2.880740
60	DEP_WND_POS_THREE	2.611976
86	FLIGHT_DAY_OF_YEAR	2.457012
2	OP_UNIQUE_CARRIER	2.454673
5	ORIGIN	1.264836
65	ARR_CIG_POS_THREE	1.215989
39	DEP_AA1_PER_QUALITYCODE	1.089051
19	DEP_QUALITY_CONTROL	1.083902
30	ORIGIN_DELAY_RATE_3H	0.845951
53	DEP_TMP_POS_TWO	0.797524
38	DEP_AA1_PER_CONDITION_CODE	0.778778
56	DEP_VIS_POS_THREE	0.766304
66	ARR_CIG_POS_FOUR	0.721816
21	ARR_STATION_ID	0.715776
1	DAY_OF_WEEK	0.605790

Feature Selection Implementation

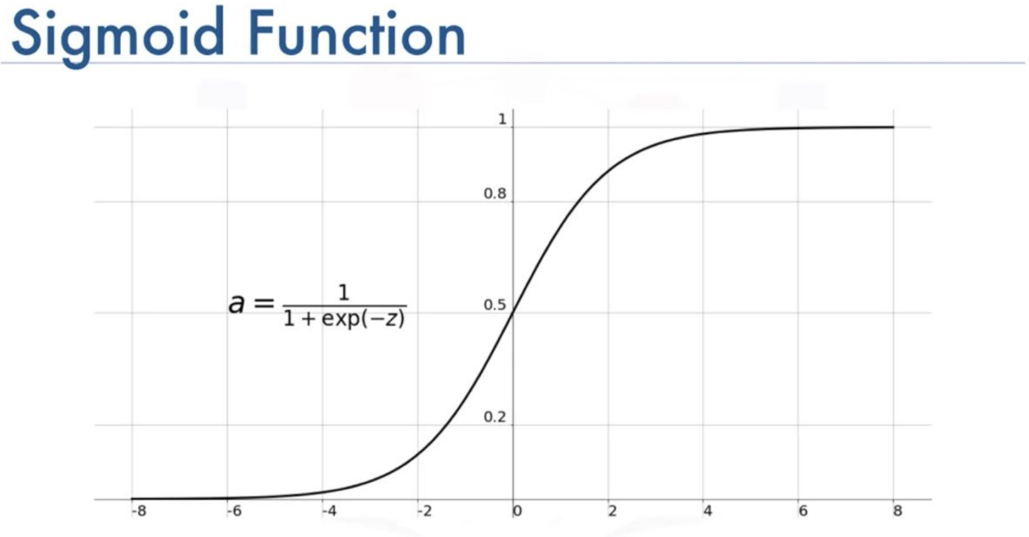
- Selected top 30 attributes from the down selected 67
  - Used to reduce the cost of further operations with the most potential power
- Selected features used to pare down join for large data set and at scale models

Berkeley
UNIVERSITY OF CALIFORNIA

Homebrew Model

Logistic Regression

- Used to predict something is True or False (1 or 0)
- Fits an S shaped logistic function (Sigmoid)
- Can be used as a classifier
- Uses Maximum Likelihood to select model



Source: Medium

Key Functions

- Sigmoid Function (I.E. Logistic Function)

$$z = \left(\sum_{i=1}^{n} w_i x_i \right) + b$$

$$y = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+\exp(-z)}$$

Source: Speech and Language Processing. Daniel Jurafsky & James H. Martin

Key Functions

- Conditional Maximum Likelihood Function (Cost Function)

$$L_{\mathrm{CE}}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))]$$

Source: Speech and Language Processing. Daniel Jurafsky & James H. Martin

Key Functions

- Gradient Descent Function

$$\frac{\partial L_{\mathrm{CE}}(\hat{y},y)}{\partial w_j} = [\sigma(w\cdot x+b)-y]x_j$$

- L1 Regularization

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)}) - \alpha R(\theta)$$

- La

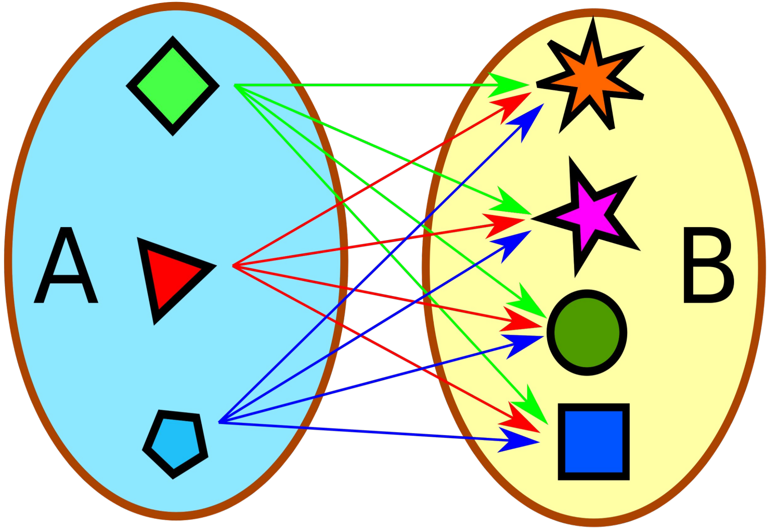
Source: Speech and Language Processing. Daniel Jurafsky & James H. Martin

Berkeley
UNIVERSITY OF CALIFORNIA

Scale Model

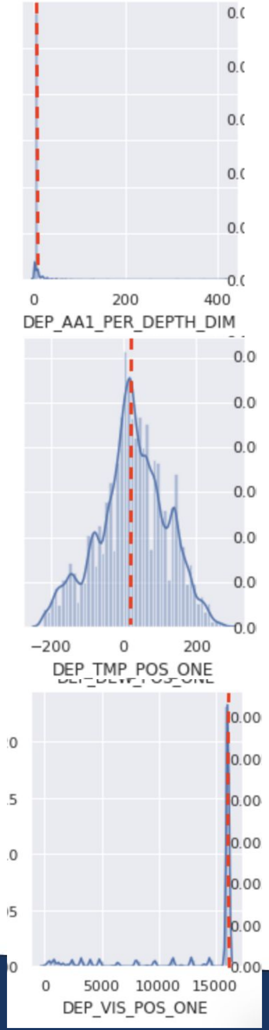
Join - Full 5-Year Dataset

- Same join logic as 3-Month join
- Make join more scalable:
  - Split into 5 annual joins, then union together everything at the end
  - Only used ~30 features in the join from previous feature selection
- Full join completed in ~90 minutes



Preprocessing

- “Clear Sky” Imputation for Data:
  - Compute mean/mode of variable in training data depending on interpretation (e.g. mode for visibility), impute for missing train and test data values
- Target Encoding of Categorical Variables:
  - Utilizes Bayesian/smoothed approach to avoid overfitting for under-represented categories
  - Encoded on training data, mapped to test data
- Z-Score Scaling of Data:
  - StandardScalar fit on training data, transform test data
- Sampling for Class Balance:
  - SMOTE upsampling on 3 month dataset, downsampling majority class for full flights data



Random Forest - Introduction

- Group of random decision trees to make classification predictions
- Tuned model on 5 key hyperparameters:
  - Number of trees, max tree depth
  - Minimum information gain
  - Minimum instances per node
  - Subsampling rate
- Must balance scalability, overfit avoidance and performance considerations
- Inherent cross validation
- Downsampled majority class to balance classes

Random Forest - Results

- Surprisingly, model only chose to use **3 features**:
    - **Same tail with previous delay on same day** - 0.62 feature importance
    - **Delay % at origin 2-8 hours before departure** - 0.28 feature importance
    - **Departure hour** - 0.10 feature importance
- Best model hyperparameters tended to limit overfitting:
    - 50 trees, 8 depth, 0.01 information gain
    - 100 instances per node, 0.5 subsampling rate
- **Performance on test set:**
    - 0.78 precision, 0.48 recall
    - **0.59 F1-Score** (Gradient Boosting - 0.56, Logistic Regression - 0.56)
    - 3 minute run time

Berkeley
UNIVERSITY OF CALIFORNIA

Modeling Conclusions

- Random Forest model shows promise in reducing business costs of delays
  - Predictions capture ~50% of delays at 78% precision
- Very few features drove all of the predictive power
- None of the weather features made a significant contribution to the models
  - Hypothesis: The best features such as delays by the same tail and origin airport delay rate in prior 6 hours have strong embedded weather signals
- If we had more time:
  - More detailed feature engineering on weather data
  - More hyperparameter tuning to find incremental improvements
  - Find data tied to delays caused by non-weather factors