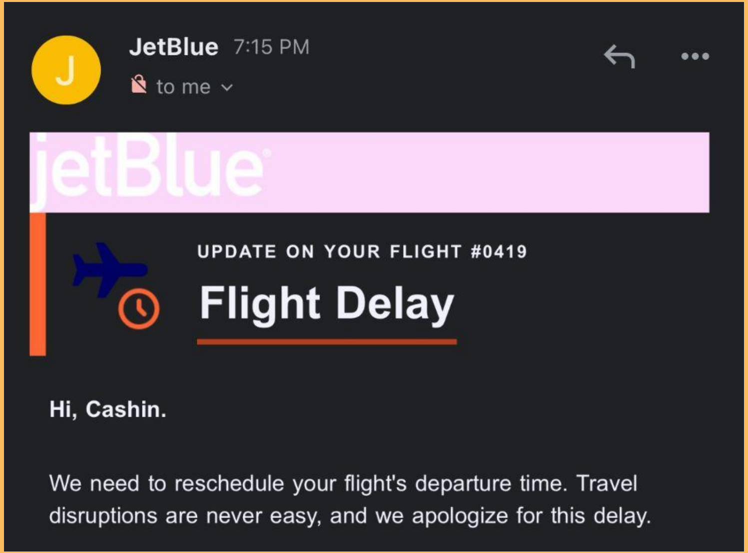
Optimized Airline Bookings

Cashin Woo, Dhruv Naheta, and Jordan Leslie

Motivation

- Travel plans can be **disrupted** and **ruined** by delayed and canceled flights
- Delays and cancellations are not only a huge inconvenience to travelers, but costly to airlines
- **What if we could predict whether a flight will be canceled or delayed when booking tickets?**



Our Dataset

- **Kaggle Flight Status Prediction Dataset**
- All U.S. domestic flights from 2018-2022
- 61 attributes
  - Airline, location, time, delay measurements
- 29.19 million samples
- **Original plan**
  - Sample 1 million rows from each year and drop rows with missing values → 4.8 million samples
  - Use 70/30 train test split → 3.4 million training, 1.4 million test

Existing Work

- Almost all of the existing work that we looked at used binary classification in predicting whether or not a flight will be cancelled.
- We use **multi-class classification** instead of binary classification to provide more useful information to consumers — However, it **provides more of a challenge** to create the model.

Delay	Time
On Time	N/A
Short	<30 mins
Med	30 - 1hr
Long	>1hr
Cancelled	N/A

6:15 AM – 7:29 AM Frontier	2 hr 14 min ATL–ORD	Nonstop	85 kg CO_2 -19% emissions	$26
5:57 PM – 6:54 PM Frontier	1 hr 57 min ATL–MDW	Nonstop	69 kg CO_2 -34% emissions	$26
3:35 PM – 8:15 PM Spirit	5 hr 40 min ATL–ORD	1 stop 51 min MCO	203 kg CO_2 +93% emissions	$48
7:50 AM – 9:02 AM Spirit	2 hr 12 min ATL–ORD	Nonstop	96 kg CO_2 -9% emissions	$68
7:00 AM – 8:05 AM United	2 hr 5 min ATL–ORD	Nonstop	105 kg CO_2 Avg emissions	$104

Medium delay

Long delay

Short delay

Cancelled

On time

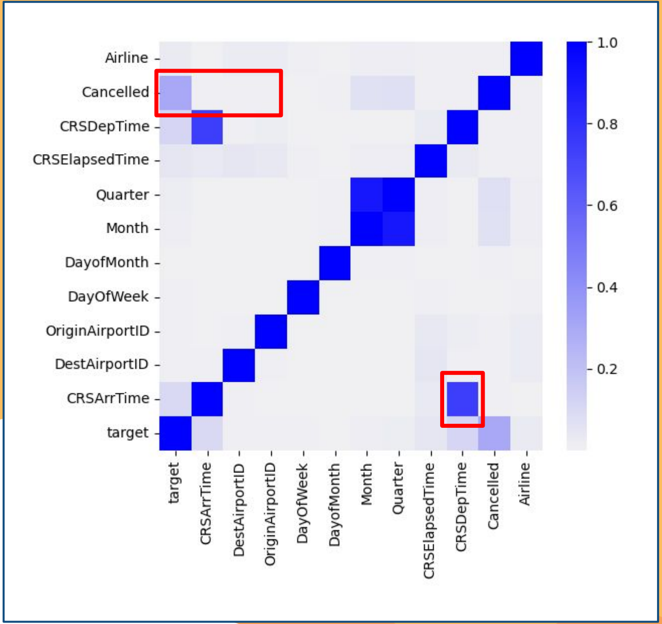
Procedure

- ✈️ Preprocessing
- ✈️ Feature Selection
- ✈️ Model Training and Tuning
- ✈️ Model Assessment and Selection

Preprocessing

1. Derive **multi-class label**
   - [on-time, short-delay, med-delay, long-delay, cancelled]
2. Drop features which are **unknown to consumers** at the time of booking
   - I.e. Departure delay, plane ID number
3. One-hot encoding and standardization

Feature Selection



- 61 attributes → 11 features
- Do we need more dimension reduction to avoid overfitting?
- **Kendall's Tau correlation**
  - Rank-order correlation metric which works for both categorical and continuous data
  - Values between [-1,1], 0 = independent
  - Used absolute values for heatmap
- **Dropped** 'Cancelled' feature → 10 features

```
(boat) ~/Desktop/ML/airplane (preprocessing) $ python preprocessing.py

Elapsed time: 28856.58
```

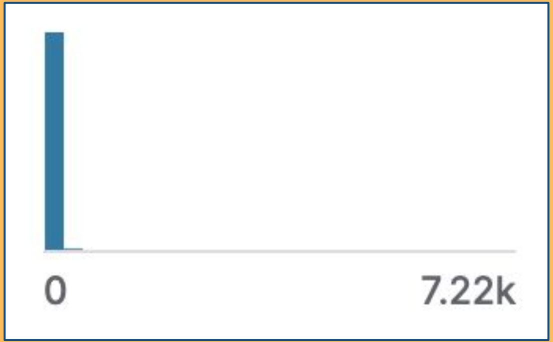
28856 seconds = 8.01 hours

So 4.8 million is probably too many samples
for our computers

New Approach

- If preprocessing takes 8 hours, MLP is going to take **forever**
- 1 million from each year → **10,000** from each year
- Increase # samples if performance is bad

Model Assessment



Delay time (minutes)

- Data is imbalanced (only ~3% are cancelled, delay distribution is heavily skewed)
- **Balanced accuracy and micro-average precision**
- **Prediction Time:** consumers need to make quick predictions while booking

Model Selection

- ✈️ Decision Tree, Naive Bayes, Logistic Regression
- ✈️ Random Forest
- ✈️ Multilayer Perceptron
- ✈️ No KNN (prediction takes too long)

Model Selection

- ✈️ Decision Tree, ~~Naive Bayes~~, Logistic Regression
- ✈️ Random Forest
- ✈️ Multilayer Perceptron
- ✈️ No KNN (prediction takes too long)

- No Naive Bayes classifier for both categorical and continuous features
- Might discretize continuous features and train NB if we have time

Hyperparameter Tuning

Using GridSearch

Decision Tree Best Parameter:
{'criterion': 'gini',
'max_depth': 30,
min_samples_leaf': 10}

Others: in progress

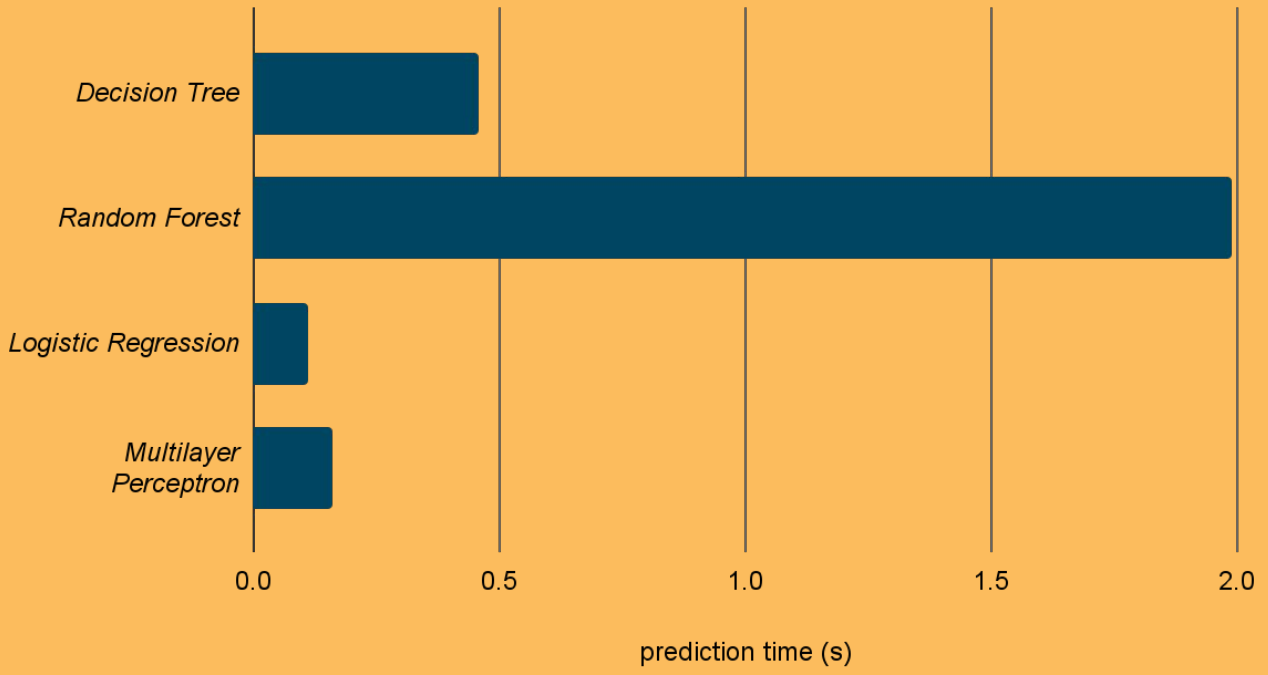
```
parameters = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [5, 10, 20, 30],
    'min_samples_leaf': [10, 50, 100, 500, 1000]
}
```

```
parameters = {'activation': ['relu'],
              'solver': ['adam', 'sgd'],
              'alpha': [0.0001, 0.001, 0.01],
              'batch_size': [100, 200, 500, 1000],
              'learning_rate': ['adaptive'],
              'max_iter': [100, 200, 300],
              'early_stopping': [True, False]}
```

Preliminary Results


Model performance
Decision Tree
Random Forest
Logistic Regression
Multilayer Perceptron
0.00
0.25
0.50
0.75
1.00
Balanced Accuracy
Micro precision

Prediction time for 15000 samples



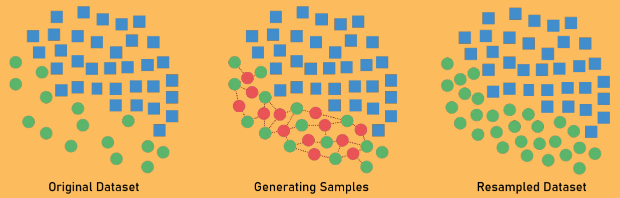
Discussion

- **What do the results signify?**
  - All of the models performed subpar compared to previous and existing work
  - Imbalanced **multi classification is much more difficult than binary classification.**
  - *But something about our approach is probably wrong, if all models are performing poorly
- The observed performance, trailing established models, highlights the **complexities of multi-classification.**
  - Despite initial setbacks, the project shows promise as a good starting point for future improvements.

Next Steps

- Imbalanced classification is known to be a difficult problem- research more techniques
- **Oversample** infrequent classes
  - Synthetic minority oversampling technique **(SMOTE)**
- **Imbalanced learn** python library
- **Use more data** (if our computers can handle it)

- Create UI for consumers to input flight details and receive a prediction!



Thank you!

Questions?