Predicting Flight Delays Using Machine Learning

Javier Herbas

Presentation Outline

Introduction & Objective

Data Gathering & Preprocessing

Exploratory Data Analysis

Modeling

Model Evaluation

Summary and Recommendations

Way Forward

Introduction & Objectives

Flight delays has become a very important subject for air transportation all over the world because of the associated financial loses that the aviation industry is continuously going through

According to the Bureau of Transportation Statistics (BTS) of the US, over 20% of the US flights were delayed during 2018 ⇨ 41 billion US$

Delays caused inconvenience to airlines and passengers ⇨ Financial loses and increased stress

Is it possible to predict when a flight will be delayed even before it comes out in the departure board?

Objective: Design a model that predicts flight delays before they are announced on the departure boards

Data Gathering & Preprocessing

Full Dataset:

- Source: Kaggle
- Data from 10 years (2009 – 2018)
- 10 different files
- Average of 28 Categories (> 1 million rows)

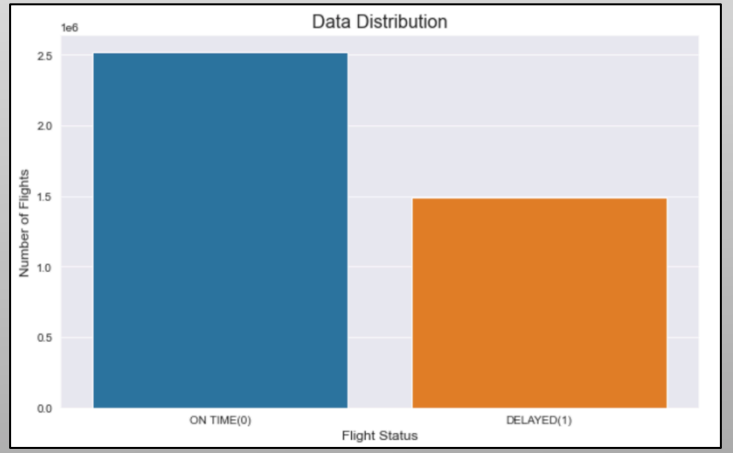
Selected Dataset:

- 1 year: 2018
- +7.2 million rows → 20 top destination (cities): +4.2 million rows

What differentiates my models from others:

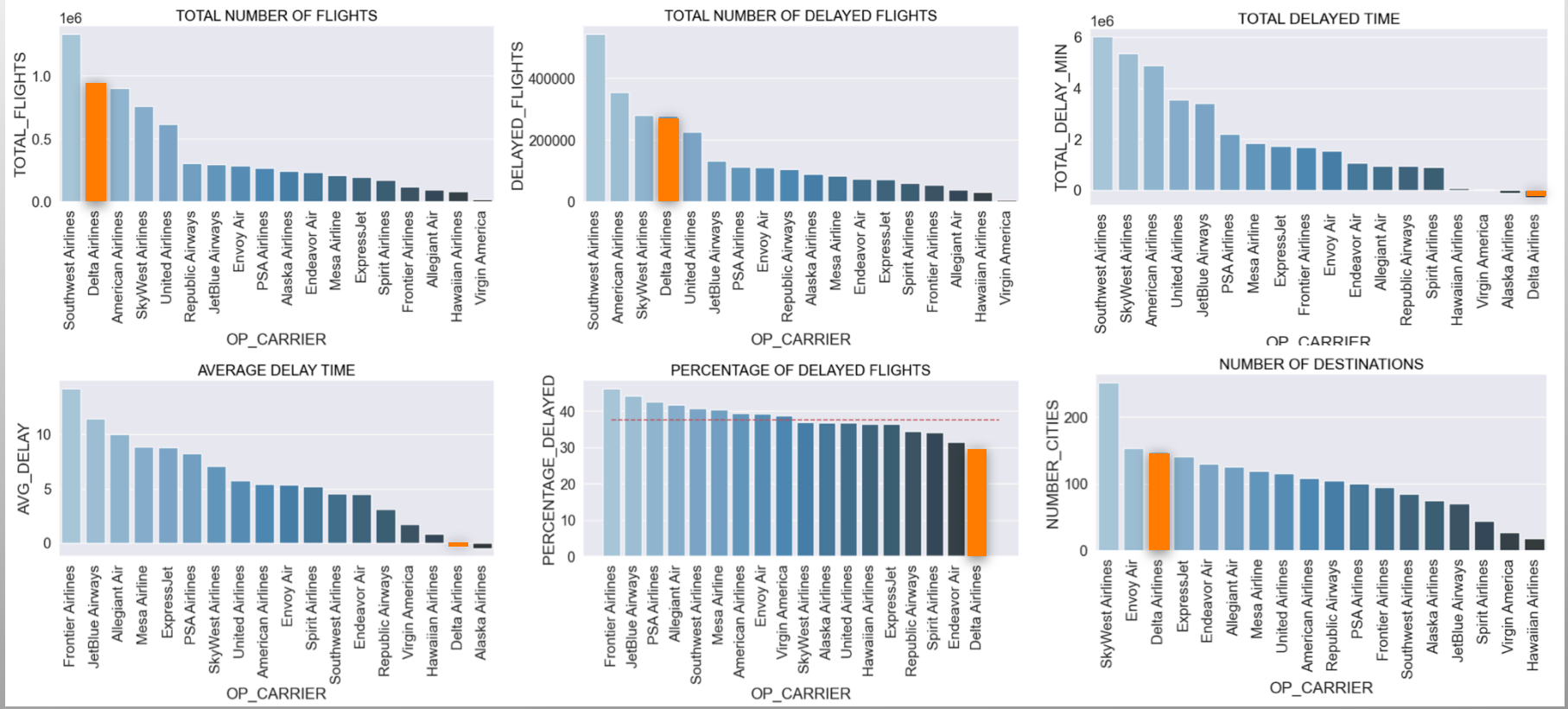
- No category that would imply passengers already in the plane or the delay announced on departure boards were considered

- Many Categoricals (days, months, origin, destination, times,…)
- Imbalance with almost a 2:1 ratio:



- Over 15 different features were engineered for Exploratory Data Analysis

Exploratory Data Analysis



Modeling

Binary Classification:

- 0 = Flight arrives on-time
- 1 = Delayed Flight

Algorithms tested:

- Bagged Tress
- Random Forest
- AdaBoost
- Gradient Boosted Trees
- XGBoost

- Deep Neural Networks

Over 70 different models tested

MLP Neural Networks - Best Model:

```
1  model_5 = Sequential()
2
3  model_5.add(Dense(50, activation='tanh', input_shape=(63,)))
4
5  model_5.add(Dense(30, activation='tanh'))
6
7  model_5.add(Dense(15, activation='tanh'))
8
9  model_5.add(Dense(5, activation='relu'))
10
11 model_5.add(Dense(1, activation='sigmoid'))
12
13 model_5.summary()
```

executed in 53ms, finished 01:23:42 2020-10-15

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_47 (Dense)             (None, 50)                3200
_________________________________________________________________
dense_48 (Dense)             (None, 30)                1530
_________________________________________________________________
dense_49 (Dense)             (None, 15)                465
_________________________________________________________________
dense_50 (Dense)             (None, 5)                 80
_________________________________________________________________
dense_51 (Dense)             (None, 1)                 6
=================================================================
Total params: 5,281
Trainable params: 5,281
Non-trainable params: 0
_________________________________________________________________
```

```
1  model_5.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

executed in 34ms, finished 01:23:53 2020-10-15

```
1  results5 = model_5.fit(X_train, y_train, epochs=25, batch_size=32, validation_split=0.1)
```

executed in 7h 19m 25s, finished 08:43:39 2020-10-15

Model Evaluation

Due to having an "Imbalance data", accuracy was not enough to measure the model's performance.

- Confusion Matrices
- Precision
- Recall
- F1

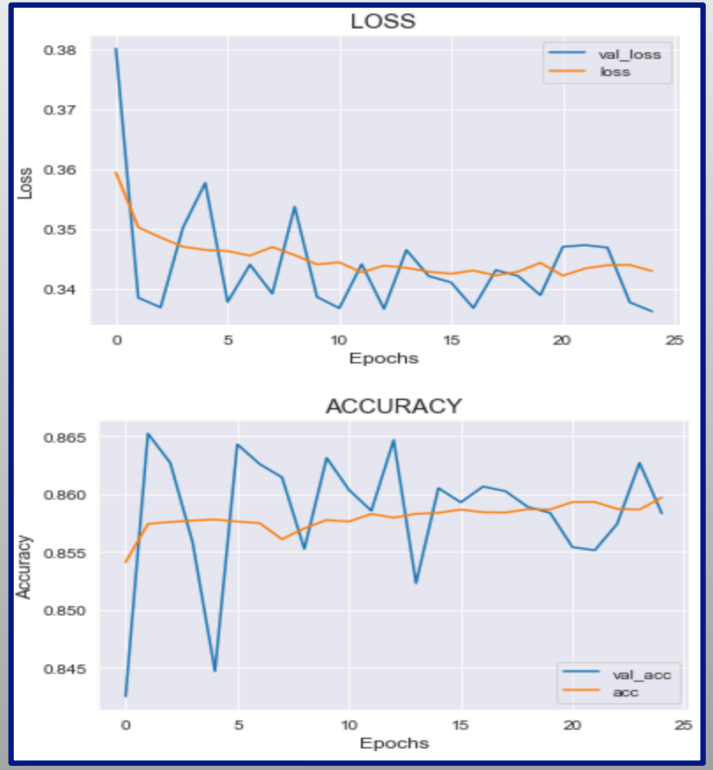
MLP Neural Network

		Inbalanced data			
Notebook	Models	Precision	Recall	Accuracy	F1
MLP_NN_Part_I	Model 1	79	54	66	47
	Model 2	79	54	66	47
	Model 3	68	62	69	62
	Model 4	70	62	69	61
	Model 5	79	54	66	47
	Model 6	68	63	69	63
	Model 7	69	63	70	63
MLP_NN_Part_II	Model 1	89	83	87	85
	Model 2	88	84	87	85
	Model 3	87	80	84	82
	Model 4	87	80	84	82
	Model 5	86	84	86	84
	Model 6	87	80	84	82
	Model 7				
MLP_NN_Part_III	Model 1				
	Model 5	69	63	70	63
	Model 6	69	62	70	62
	Model 7	71	61	61	70

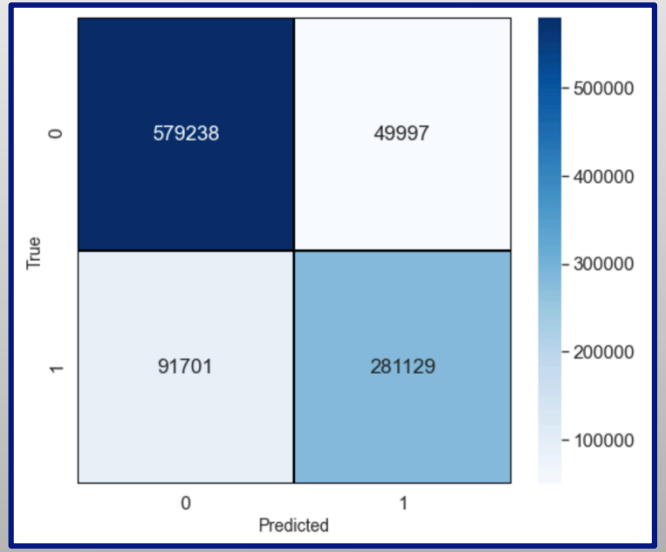
		Inbalanced data			balanced data		
	Algorithm	Precision	Recall	Accuracy	Precision	Recall	Accuracy
1	Baseline Tree (without DEP_DELAY)	64.00	51.00	63.25	57.00	57.00	57.76
2	Bagged Tress (without DEP_DEPAY(	66.00	51.00	63.33	61.00	52.00	63.68
3	Random Forest (without DEP_DELAY)	74.00	50.00	62.89	57.00	57.00	56.76
4	Random with Bootstrat Weighting	57.00	57.00	56.67	57.00	57.00	57.25
5	AdaBoost_V1 (with DEP_DELAY)	84.00	82.00	81.72			
6	AdaBoost_V2 (without DEP_DELAY)	65.00	54.00	64.64			
7	Gradient Boosted Trees (with DEP_DELAY)	85.00	79.00	83.09			
8	Gradient Boosted Trees (without DEP_DELAY)	70.00	57.00	66.84			
9	XGBoost (with DEP_DELAY)	88.00	82.00	86.00	87.00	83.00	85.65
10	XGBoost (without DEP_DELAY)	71.00	61.00	69.37	69.00	63.00	69.68

Model Evaluation – Best Model

MLP Neural Networks – Model_5 performance evaluation:



Loss and Accuracy plot (25 epochs)



	precision	recall	f1-score	support
0.0	0.86	0.92	0.89	629235
1.0	0.85	0.75	0.80	372830
accuracy			0.86	1002065
macro avg	0.86	0.84	0.84	1002065
weighted avg	0.86	0.86	0.86	1002065

Confusion Matrix & Classification Report

```
Accuracy: 85.86 %
Precision score: 84.9 %
Recall score: 75.4 %
F1 score: 79.87 %
```

Final Performance Metrics

Summary & Recommendations

- From the EDA done it seems as Delta Airlines and Alaska Airlines are two of the most reliable airlines in terms of flights arriving on time

- It is quite hard to create a ML model to predict flight delays without giving them any features that could affect the models by biasing them, whereas Deep Neural Networks were more time consuming but easier to tune them for better metrics and therefore predictions

- The best model ended up with an accuracy of over 85%, however a series of categories believed to be key could not be taken into account due to a shortage of data. Adding these could increase the accuracy and other metrics.

Way Forward

- Engineer a “Time of the day” category to understand if there are any time windows during the day more prone than others to have delays
- Re-do the EDA but with the 10 year historic data instead of only the 2018
- Scrape weather information such as temperature, humidity, wind and precipitation and add it to the current dataset as it is known that these are major factors that affect the flight delays
- Re-run the ML and Neural Networks models selected as the best performers with one departure city and top 20 destinations chosen by the number of arriving flights


Thank You