Machine Learning approach to predict flight arrival delay.

Ryan Andrew Dmello
Parikshit Vijay Urs
Susheel Gounder
Robin Sharma

Problem Statement

United Airlines has direct flights into Syracuse (SYR) from four cities: Chicago (ORD), Denver (DEN), Newark (EWR), and Washington (IAD). There could be more than one flight from each of those cities into SYR each day. The goal is to predict 1-4 days in advance if each flight's arrival time into SYR would be early, on-time, delayed, or severely delayed.

Introduction

Flight delays can be unpredictable and often depend on a multitude of parameters and hence it becomes so difficult to predict it. Some of the parameters we think could affect the delay are –

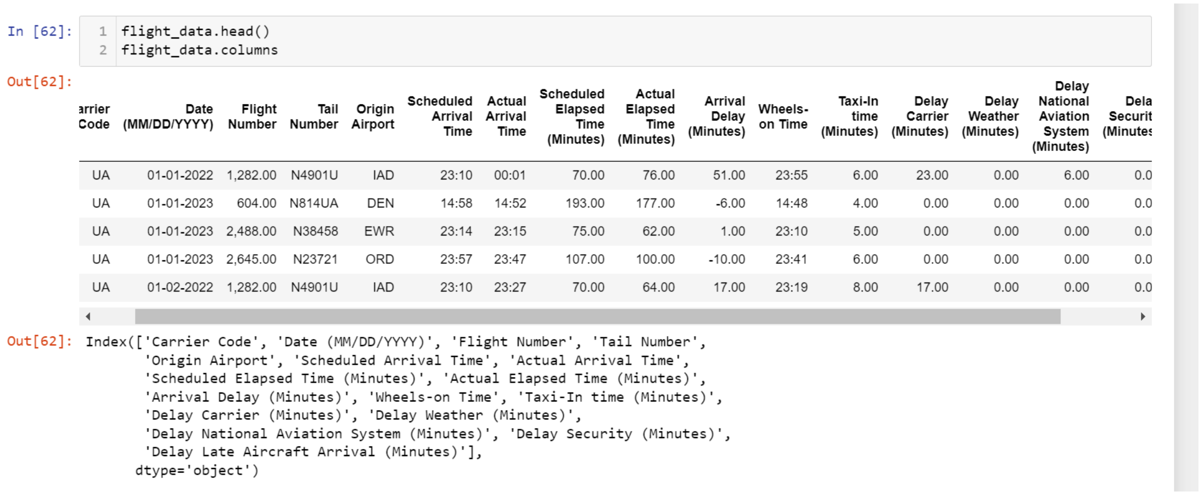
1. Weather conditions: Weather is a significant factor that can affect flight delays. Adverse weather conditions such as thunderstorms, snowstorms, or hurricanes can cause flight delays and cancellations.
2. Air traffic congestion: The number of flights taking off and landing at an airport can impact flight delays. Airports with high levels of congestion can experience delays due to runway and airspace capacity constraints.
3. Aircraft maintenance: Aircraft maintenance issues such as mechanical problems, maintenance checks, or repairs can cause flight delays.
4. Airline operations: Airlines may experience delays due to issues with crew scheduling, crew rest requirements, or aircraft availability.
5. Security issues: Security issues such as passenger screening or terrorism threats can cause flight delays.
6. Late-arriving aircraft: Delays caused by previous flights can cascade down to subsequent flights, causing delays.
7. Airline and airport policies: Airline and airport policies such as check-in procedures, baggage handling, and boarding processes can impact flight delays.
8. Airline scheduling: The airline's schedule, including the number of flights and their routes, can impact flight delays.
9. Air traffic control: Air traffic control issues such as traffic flow restrictions or airspace closures can cause flight delays.
10. Human factors: Human factors such as flight crew or ground crew errors, communication issues, or air traffic control errors can cause flight delays.

I know, these are a lot of parameters and some of them are such that one cannot benefit from historical data. Like Security Issues. They are often based on current events, security threats, or intelligence reports, which may not have any historical data to draw upon. As a result, predicting security-related flight delays using historical data alone may be challenging.

Our approach to solve this problem was to build a model iteratively. Add features one by one and see how it performs.

Data Collection

We used https://www.transtats.bts.gov/ontime/ to get the airline data. Below is an example of the data –



```
In [62]: flight_data.head()
         flight_data.columns
```

Out[62]:

arrier Code	Date (MM/DD/YYYY)	Flight Number	Tail Number	Origin Airport	Scheduled Arrival Time	Actual Arrival Time	Scheduled Elapsed Time (Minutes)	Actual Elapsed Time (Minutes)	Arrival Delay (Minutes)	Wheels-on Time	Taxi-In time (Minutes)	Delay Carrier (Minutes)	Delay Weather (Minutes)	Delay National Aviation System (Minutes)	Dela Securit (Minutes
UA	01-01-2022	1,282.00	N4901U	IAD	23:10	00:01	70.00	76.00	51.00	23:55	6.00	23.00	0.00	6.00	0.0
UA	01-01-2023	604.00	N814UA	DEN	14:58	14:52	193.00	177.00	-6.00	14:48	4.00	0.00	0.00	0.00	0.0
UA	01-01-2023	2,488.00	N38458	EWR	23:14	23:15	75.00	62.00	1.00	23:10	5.00	0.00	0.00	0.00	0.0
UA	01-01-2023	2,645.00	N23721	ORD	23:57	23:47	107.00	100.00	-10.00	23:41	6.00	0.00	0.00	0.00	0.0
UA	01-02-2022	1,282.00	N4901U	IAD	23:10	23:27	70.00	64.00	17.00	23:19	8.00	17.00	0.00	0.00	0.0

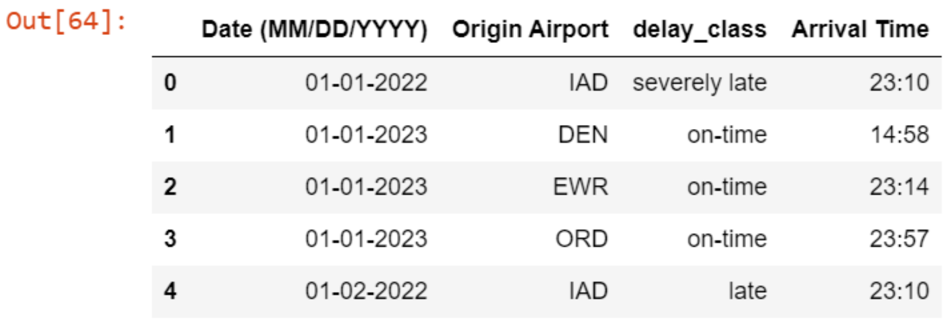
```
Out[62]: Index(['Carrier Code', 'Date (MM/DD/YYYY)', 'Flight Number', 'Tail Number',
                'Origin Airport', 'Scheduled Arrival Time', 'Actual Arrival Time',
                'Scheduled Elapsed Time (Minutes)', 'Actual Elapsed Time (Minutes)',
                'Arrival Delay (Minutes)', 'Wheels-on Time', 'Taxi-In time (Minutes)',
                'Delay Carrier (Minutes)', 'Delay Weather (Minutes)',
                'Delay National Aviation System (Minutes)', 'Delay Security (Minutes)',
                'Delay Late Aircraft Arrival (Minutes)'],
               dtype='object')
```

The data provided is very informative but most of the columns are not useful to predict the delay as they are real time information. Like 'Actual Arrival time', 'Actual Elapsed Time', 'Delay Carrier', etc. would only be available during and after the course of the flight. So, most of these columns would be useless to us. Of course, we did perform some analysis to get the most influential parameter. 'Delay Carrier', 'Delay National Aviation System' were some parameters which had high weights.

Let's talk about the data that we used. Flight delays are somewhat related to the Airlines too. Some Airlines have lower average delays than others. And in our case the problem statement is to predict only for UA. So there goes our reason to only use data of UA flights. To make it more simplified we reduced the data even further to only use flight data pertinent to Syracuse Airport.

Next, we had an option to decide the time frame of the data to consider. We know that airline operations took a hit globally due to the pandemic of 2020-2021. This means the data during that wouldn't be useful to us and may result in wrong fitting of the model. So, we only considered flight data from Jan 2022 to April 2023 (1088 records).

We ended up using just the below columns –

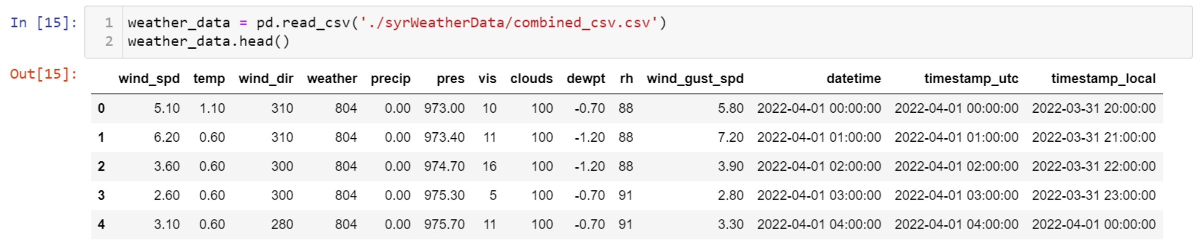


Out[64]:

	Date (MM/DD/YYYY)	Origin Airport	delay_class	Arrival Time
0	01-01-2022	IAD	severely late	23:10
1	01-01-2023	DEN	on-time	14:58
2	01-01-2023	EWR	on-time	23:14
3	01-01-2023	ORD	on-time	23:57
4	01-02-2022	IAD	late	23:10

I know this does not look at all right, but we still decided to go with this just for our initial prediction. We got a training accuracy of 40% and test accuracy of 29% with this data for the best model (will talk about later).

The model was not really performing well, and we felt the need to add more parameters. We decided to add weather data. For weather data we used https://www.weatherbit.io/ . They offer a free tier for students and researchers. We got the hourly weather data of Syracuse (location of arrival) from Jan 2022 to April 2023. We decided to use the below parameters for our model –



```
In [15]:
weather_data = pd.read_csv('./syrWeatherData/combined_csv.csv')
weather_data.head()
```

```
Out[15]:
   wind_spd  temp  wind_dir  weather  precip    pres  vis  clouds  dewpt  rh  wind_gust_spd             datetime        timestamp_utc      timestamp_local
0      5.10  1.10       310      804    0.00  973.00   10     100  -0.70  88           5.80  2022-04-01 00:00:00  2022-04-01 00:00:00  2022-03-31 20:00:00
1      6.20  0.60       310      804    0.00  973.40   11     100  -1.20  88           7.20  2022-04-01 01:00:00  2022-04-01 01:00:00  2022-03-31 21:00:00
2      3.60  0.60       300      804    0.00  974.70   16     100  -1.20  88           3.90  2022-04-01 02:00:00  2022-04-01 02:00:00  2022-03-31 22:00:00
3      2.60  0.60       300      804    0.00  975.30    5     100  -0.70  91           2.80  2022-04-01 03:00:00  2022-04-01 03:00:00  2022-03-31 23:00:00
4      3.10  0.60       280      804    0.00  975.70   11     100  -0.70  91           3.30  2022-04-01 04:00:00  2022-04-01 04:00:00  2022-04-01 00:00:00
```

wind_spd	Wind Speed
temp	Temperature
wind_dir	Wind Direction
precip	Precipitation
pres	Pressure
vis	Visibility
clouds	Cloud Cover
dewpt	Dew Point
rh	Humidity
wind_gust_spd	Wind Gust

We didn't consider the weather data at departure. We merged the weather data with the flight data on timestamp_local x Arrival Time. We took average of the weather for 1 hour before and after the Arrival hour. For example, if the flight is to arrive at 10:30 AM, then we took the average of weather conditions at 9 AM, 10 AM and 11 AM.

Additionally, we introduced day of the week features i.e Sunday, Monday, etc. The idea behind this is that the day of the week feature can help identify patterns and seasonality that are specific to certain days of the week. Example: flights on Mondays may have a higher likelihood of being delayed due to congestion from weekend travelers or issues with airport staffing.

With this we were ready to start training and experimenting with our models.

Model Training and outcome

We tried below models with different hyper parameters –

1. **Logistic Regression**:
    - Training Accuracy: 0.496
    - Testing Accuracy: 0.431
    - Precision: 0.322
    - Recall: 0.431
    - F1-Score: 0.364

The Logistic Regression model has the lowest accuracy, precision, and recall among all the models. It struggles to distinguish between the four classes of train arrival times, especially the 'late' and 'severely late' categories. This model is underfitting the data, and it is not the best choice for predicting train arrival times.

2. **Gradient Boosting Classifier**:
    - Training Accuracy: 0.836
    - Testing Accuracy: 0.459
    - Precision: 0.390
    - Recall: 0.459
    - F1-Score: 0.376

The Gradient Boosting Classifier model is much better than the Logistic Regression model, with higher accuracy, precision, and recall. This model performs well in predicting the 'on-time' train arrival category, but it struggles with the other categories.

3. **Random Forest Classifier**:
    - Training Accuracy: 1.0
    - Testing Accuracy: 0.422
    - Precision: 0.343
    - Recall: 0.422
    - F1-Score: 0.355

The Random Forest Classifier model is overfitting the data, with a training accuracy of 1.0 and a testing accuracy of 0.422. This model performs well in predicting the 'on-time' train arrival category, but it struggles with the other categories. This model has slightly lower accuracy, precision, and recall than the Gradient Boosting Classifier model.

4. **Decision Tree Classifier**:
    - Training Accuracy: 1.0
    - Testing Accuracy: 0.349
    - Precision: 0.355
    - Recall: 0.349
    - F1-Score: 0.352

The Decision Tree Classifier model has the lowest accuracy among all the models, with an accuracy of 0.349. This model performs well in predicting the 'early' and 'on-time' train arrival categories, but it struggles with the other categories. This model is overfitting the data, with a training accuracy of 1.0 and a testing accuracy of 0.349.

5. **K-Nearest Neighbors Classifier**:
    - Training Accuracy: 0.578
    - Testing Accuracy: 0.422
    - Precision: 0.312
    - Recall: 0.422
    - F1-Score: 0.360

The K-Nearest Neighbors Classifier model performs similarly to the Random Forest Classifier, with an accuracy of 0.422. This model performs well in predicting the 'on-time' train arrival category, but it struggles with the other categories. This model is underfitting the data, with a training accuracy of 0.578 and a testing accuracy of 0.422.

In summary, among all the models, the Gradient Boosting Classifier model has the highest accuracy, precision, and recall. However, it is overfitting the data. The Random Forest Classifier model performs similarly to the Gradient Boosting Classifier model, but it is also overfitting the data. From all the above models we got the best accuracy with Gradient Boosting Classifier (max_features=3,max_depth=8,n_estimators=10). Training accuracy of 83% and testing accuracy of 44%. This testing accuracy was almost like the final assessment result (42%). Later we realized that we should have tried XGBoost as well but never really tried it.

Conclusion

In conclusion, the machine learning approach to predict flight arrival delays has shown somewhat average results. By analyzing historical data on various parameters affecting flight delays, we were able to develop a model that predicts the probability of flight delays. The model's performance was evaluated using metrics such as accuracy, precision, recall, and F1 score.

Overall, the machine learning approach to predict flight delays has the potential to provide significant benefits to the aviation industry by reducing the number of flight delays and cancellations, improving customer satisfaction, and increasing revenue. As the availability of data continues to increase, further research and development in this area could lead to even more accurate and reliable models for predicting flight delays.

"A machine learning model is only as good as the data it learns from."

- Yann LeCun