On Predicting Flight Arrival Status with Tree-Based Machine Learning

Akib Jawad Nafis, Uddesh Shyam Kshirsagar, Rishitha Vanamala, and Arya Pathrikar

Syracuse University, New York, USA

***Abstract*—Despite having an advertised arrival time, a flights' actual arrival time can be far from the scheduled time. One metric, arrival status, can be used to measure the disparity between the scheduled arrival time and the actual arrival time of a flight. Arrival status of a flight can be early, on-time or delayed. In this project, our goal was to predict the arrival status of a flight 1-3 days earlier than the actual flight. Solving the general problem of predicting arrival status for any flight is too large to solve within the time-frame of one month. Hence, we attempted to solve a miniature version of the problem where we build a model to predict the arrival status of 6 flights arriving at Syracuse from 3 different origins (New York, Chicago, and Orlando). Given the miniature size of our problem, we did not employ any deep learning techniques to built the model. We built a tree-based based machine learning model from the historical data of flights that were operating between 2006 and 2024. In our model, we incorporated 3 categories of flight data: flight characteristics, airlines on-time arrival history, and airport's on-time departure history. Additionally, we incorporated historical weather data in our model. We built an ensemble of trees by using XGBoost algorithm to predict the arrival status of a flight given the flight date, flight no, origin, scheduled departure time, and scheduled arrival time. We evaluated our model based on its' prediction accuracy on the test dataset and based on its' accuracy in predicting arrival status of future flights operated in the period of 4 days (from April 19, 2024 to April 23, 2024). On our test dataset of 2818 samples, our model achieves 52.7% accuracy, while predicting arrival status of future flights, our model were correct on 12 out of 23 occasions.**

Keywords*—*Flight Arrival Status Prediction, XGBoost

I. Introduction and Problem Description

Aviation industry tries to follow pre-calculated schedule for each step of their operation. Airlines mention schedule departure time and scheduled arrival time of a flight while they are selling tickets for that flight. But while operating the flight, actual departure or arrival time might be different from the time scheduled earlier. In this project, we are more concerned about the arrival time of a flight.

Depending on the amount of difference between the scheduled arrival time and actual arrival time, we are classifying arrival status of a flight as "EARLY", "ON-TIME", or "LATE." If actual arrival time of a flight is falls between the 5 minutes offset (5 minutes earlier or 5 minutes later) of the scheduled arrival time, we are classifying the arrival status as "ON-TIME." If a flight arrives more than 5 minutes earlier than the scheduled time, we classify the arrival status of the flight as "EARLY." On the other hand, if it arrives more than 5 minutes later than the scheduled time, we classify the arrival status of the flight as "LATE."

Now, solving the general problem (predicting arrival status of all operating flights) would require way too much data and development of sophisticated machine learning algorithms. Given the one-month time period and the restriction on using neural network based algorithms, we decided to solve a relatively specific version of the problem.

In this version, instead of predicting arrival status of all flights, we decided to predict arrival status of 6 flights at one specific airport, Syracuse Hancock International Airport (SYR), from 3 different origin airports: John F. Kennedy International Airport (JFK), O'Hare International Airport (ORD), and Orlando International Airport (MCO). In particular, Six flights (3 pairs from each origin) that we choose for predicting arrival status are: United Airlines UA 1400 and American Airlines AA 3402 from ORD in Chicago, Jet Blue B6 116 and Delta Airlines DL 5182 from JFK in New York, and Jet Blue B6 656 and Southwest Airlines WN 5285 from MCO in Orlando. Flights in each pair from the same origin departs in sequence. For example, from ORD, UA 1400 departs at 18:52 and AA 3402 departs at 19:59. First, we need to predict arrival status (at SYR) of the earlier flight (UA 1400). After that, given the arrival status (EARLY, ON-TIME, LATE) of the earlier flight, we need to predict the arrival status of the later flight (AA 3402). For the first prediction, the prediction query will contain flight date, flight no, origin, scheduled departure time, and scheduled arrival time. For the second prediction, prediction will query will contain an additional information which is the arrival status of the earlier flight.

II. Proposed Solution

We built ensemble of tree based two models to predict arrival status of flights. One model is used for predicting arrival status of the earlier flight where previous flight status is not available in the prediction query. Remaining model is used for predicting the flight where previous flight arrival status is provided.

We tried to be methodical in solving this problem, we identified that arrival status of a flight can be affected by Four factors:

1) Characteristics of the flight coming to that destination
2) Weather properties on that day of flight
3) Reputation of the airlines in managing their flights
4) Reliability of the origin airport in assuring on-time departure of flights

We incorporated all those factors in building our models. In this section, we will describe our overall algorithm to build those two ensemble of tree models. We will describe our data

collection process in subsection II-A, data pre-processing in subsection II-B and finally model training in subsection II-C.

A. Data Collection

For our data collection we were largely dependent on Bureau of Transportation Statistics (BTS) [1].

To collect characteristics of flights coming to the destination airport, SYR, we collected data of all flights arriving at SYR from the BTS On-Time Arrival Statistics [2]. The dataset contains data of flights operated by 11 carriers during the time period starting from 2006 to 2023. To avoid outliers in our dataset, we skipped flight data for some specific years 2008, 2020, and 2021. We assume due to recession in 2008 and COVID pandemic in 2020 and 2021, flight data has drastically different data pattern. Dataset contains 17 features in total. All the features of the data can be visible on our code and in our github repo. Some of the important features along with other data properties are included in the Table I.

TABLE I: Description fo the On-Time Arrival Data for all flights ariving at SYR.

Data Property	Description
Time Period	Month: March, April, and May Year: 2006, 2007, 2009-2019, 2022-2023
Airlines	American, Delta, United, SouthWest, JetBlue, Endeavor, Envoy, Mesa, PSA, Republic, and Skywest.
Notable Features	Carrier Code, Date (MM/DD/YYYY), Flight Number, Origin Airport, Scheduled Arrival Time, Scheduled Elapsed Time (Minutes), Arrival Delay (Minutes), Delay Carrier (Minutes), Delay Weather (Minutes).

We collected historical weather data for training purpose and future weather predictions for testing purpose (to include weather data in the future test samples) from the source Visual Crossing [3]. While our data source can provide hourly weather statistics, we collected daily statistics to reduce the amount of data. Properties of the weather data is provided in Table II

TABLE II: Description fo the Weather Data

Data Property	Description
Time Period	From January 1, 2006 to May 15 2024
Notable Features	DATE, tempmin, feelslikemin, dew, humidity, precip, precipcover, snow, snowdepth, windgust, windspeedmax, sealevelpressure, cloudcover, visibility, severerisk, weather_code (cloudy, partly_cloudy_day, rain, snow, wind)

We assume on-time arrival status of a flight heavily depends on how airline crew manages their flight. For this reason, we collected a ranking of airlines (airline ranking data) based on the percentage of flights operated by a certain airline arrived on time. This ranking in published by BTS [4] and data is available for the years from 2003 to 2023.

We also assume on-time arrival status of a flight depends on how the origin airport ensures that the flights depart on-time. For this reason, we collected a ranking of airports (airport ranking data) based on the percentage of flights originating from that airport departs on-time. This ranking of airports is also published by BTS [5] and ranking data is available for the years from 2003 to 2023.

B. Data Pre-Processing

As we mentioned in II-A, we collected 4 categories of data to build our model. We combined all the collected data to create two datasets that is required for building the two models we proposed at the beginning of this section. The sequence of steps we took to create those two datasets are:

1) Combine flight data of all 11 carriers to create a single dataset containing all flights arriving at SYR. We then filtered to keep only those flights that departed from either of the three airports (ORD, JFK, and MCO). At this stage we have data for 14087 flights where we have 17 features for each flights.
2) We cannot actually utilize all 17 features in our combined flight data, because during real time testing (predicting arrival status of future flights from April 19 to April 23) we cannot provide data for all the features. Example of such feature can be Carrier Delay, Weather Delay, Taxi-In Time, Wheel-on Time etc. Since we are predicting arrival status of flights that will take in future we do not know how much Carrier Delay or Weather Delay that specific flight is going to observe.
3) Hence, we drop those extra features and converted some of the features to match features of our test data-frame. For example we computed scheduled departure time by subtracting scheduled elapsed time from the scheduled arrival time.
4) We created our target variable ARRIVAL STATUS, based on the feature "arrival delay" available in the dataset. For example, If "arrival delay" is less than -5 (flight arrived more than 5 minutes early) we set the "ARRIVAL STATUS" to 'EARLY'. In a similar manner we classified all the flights to "EARLY", "ON-TIME", and "LATE".
5) At this stage, we need to process our airline on-time arrival percentage data. Collected dataset from BTS [4], contains a percentage of flights that arrived on-time out of all the flights operated by a certain airline on a specific year. But the problem is all airlines did not submit that percentage data to BTS for all the years. In those cases, we filled those percentage data with minimum percentage reported by that specific airline. Our assumption is if an airline did not submit their on-time arrival percentage to BTS, they must performed worse than years they reported to BTS. We created a new feature "AIRLINE YEARLY ON-TIME ARR PERCENTAGE" from this data.
6) In a similar manner, we also created another feature "AIRPORT YEARLY ON-TIME DEP PERCENTAGE" from our airport ranking data.
7) After processing both airline ranking data and airport ranking data, we included weather data for all the flights. We simply selected dates for each flight, collected weather data for that specific date, and then added

weather related features for that flight. To reduce column count, we did not include all the weather features.

8) At this stage, dataset for our first model is should be ready. Features in this dataset are: DATE, DAY, FLIGHT NUMBER, ORIGIN, DEPARTURE TIME, ARRIVAL TIME, AIRPORT YEARLY ON-TIME DEP PERCENTAGE, AIRLINE YEARLY ON-TIME ARR PERCENTAGE, ARRIVAL STATUS, weather_code, tempmin, precipcover, snowdepth, windspeedmax, visibility, cloudcover.
9) We created dataset-2 based on the dataset-1. We selected all the flight that arrives from the same origin on a same day. Then for each flight, we created one extra feature "previous_flight_status" based on the arrival status of the flight that arrived earlier from the same origin on a same day. Hence, features of dataset-2 includes all the features of dataset-1 and one extra feature named "previous_flight_status".

C. Model Training

We trained two models from our two datasets that we created in the data pre-processing stage mentioned in subsection II-B. Both of our models are XGBoost model. Our model training stages are:

1) Both of our datasets contains two categorical features: ORIGIN and weather_code. We converted the categorical features using get_dummies.
2) Split the dataset of 14087 flights into two sets called train and test. Train dataset contains 80% of the data containing 11269 flights while test dateset contains 2818 flights.
3) Since features in two datasets are different, we used two different scalers.
4) After scaling, our goal was find proper parameters for XGBoost model. We employed two different techniques to find the parameters. We employed grid search with cross-validation to find the parameter. Additionally we also manually tried out different combinations of parameters to find the configuration with highest test accuracy.
5) Our parameter space can be defined by the code block mentioned in the listing 1.
6) For model-1 we achieved highest test accuracy with the configuration provided in listing 2
7) For model-2 we achieved highest test accuracy with the configuration provided in listing 3
8) After finding the best hyper parameters for both models, we used best parameter configurations to train another set models with all the available data.

```
n_estimators: [100, 150, 200],
tree_methods = ['exact', 'hist', 'approx'],
tree_depth = [4, 6, 7, 8, 10, 15, 20],
learning_rates = [0.05, 0.1, 0.5, 0.9]
gamma: [0.03, 0.5, 1],
booster: ['gbtree', 'dart'],
objective: ['multi:softmax']
```

Listing 1: Parameter Space for XGBoost hyper-parameter tuning

```
{
  objective: 'multi:softmax',
  tree_method: 'exact',
  max_depth: 20,
  earning_rate: 0.2,
  n_estimators: 200
}
```

Listing 2: Best Parameters for model-1

```
{
  objective: 'multi:softmax',
  tree_method: 'approx',
  booster: 'gbtree',
  gamma: 0.5,
  learning_rate: 0.2,
  n_estimators: 200
  max_depth: 15
}
```

Listing 3: Best Parameters for model-2

III. Evaluation

We evaluated our 2-model solution based two criteria:

1) Accuracy on Test Data: Accuracy of the model's classification on our test data
2) Real World Prediction Accuracy: Accuracy of the model in predicting arrival status flights 1-3 days early

A. Result: Accuracy on Test Data

Both of our model have two different test-suite consisting of 2818 samples of flight data. Our model-1 achieves 52.7% accuracy in predicting ARRIVAL STATUS of flights when previous flight status information is not provided. On the other hand, when previous flight status is provided, our model-2 achieves 51.63% accuracy on the test-data.

B. Result: Real World Prediction Accuracy

We also measured accuracy of our model based on it's capability in predicting arrival status of future flights. We created a query loop, where a user can provide flight date, flight no, origin, scheduled departure time, and scheduled arrival time to get a prediction of the "ARRIVAL STATUS" of that flight. If a user provides previous_fight_status in the query along with other features, query loop will generate the prediction of "ARRIVAL STATUS" using model-2. On both of the case, query loop will include airline ranking data, airport ranking data and weather data in the query to get the correct prediction from the model.

We employed this query loop to get predictions for the 23 flights arriving at Syracuse from 3 origins (ORD, JFK, and MCO) during the time period starting from April 19, 2024 to April 23, 2024. We got the predictions from our model on April 18, 2024 and matched the prediction of model with actual data that was observed on April 25, 2023. Out of 23 predictions, our model was able to predict "ARRIVAL STATUS" of 12 flights correctly. If we count test accuracy on this real world dataset, it would be 12/23, which is 52.17%. Surprisingly this value is very close to the test-accuracy we observed earlier in subsection III-A.

IV. Discussions

If we consider test accuracy, our model is clearly underfitting. Meaning it did not capture much of the varriance in the training data. Which is understandable because the test-data we are using has very low number of meaningful features. Because much of the features that are available on our training data, cannot be determined if we want to predict status of a flight that will take place in future. At least our model is better than baseline random guessing . Random guessing provide 33% accuracy where this model provides 52% correct predictions.

V. Conclusion

In this project, we tried build an ensemble of tree based machine learning model to predict arrival status of flights. We employed our model to get predictions for real world flights which was a fun exercise. By doing this exercise we learned to collect proper features and incorporating different types features in the test data for better prediction accuracy. Our model did perform accuracy around 52% of the time, which is better than baseline random guessing.

References

[1] Bureau of transportaion statistics. [Online]. Available: https://www.bts.gov/

[2] On-time statistics, bureau of transportaion statistics. [Online]. Available: https://www.transtats.bts.gov/ontime/Arrivals.aspx

[3] Visual crossing. [Online]. Available: https://www.visualcrossing.com/weather/weather-data-services

[4] On-time arrival percetage for airlines. [Online]. Available: https://www.bts.gov/topics/airlines-and-airports/annual-airline-time-rankings-2003-2023

[5] On-time departure percetage for airports. [Online]. Available: https://www.bts.dot.gov/annual-time-departure-performance