Machine Learning - Prediction of Flight's Arrival Status

Aditi Jadhav, Simran Tawar, Gurudev Ongole, Shubham Melvin Felix

Syracuse University, Department of ECS, Syracuse, USA

***Abstract*—The prediction of flight arrival times is an important task for the operations of airlines, especially airports that accommodate several airlines flying toward the same destination. For example, our dataset considers airports of the kind, as Chicago (ORD), New York (JFK), and Orlando (MCO) which all have several airlines flying into Syracuse (SYR). Our goal is to predict, one to four days in advance, the arrival status of flights into SYR, conditioned on the status of earlier flights. For this purpose, our dataset contains flight information by the Bureau of Transportation Statistics with additional information such as weather predictions from multiple sources. We will use logistic regression, decision trees, random forest, and gradient boosting to model the relationship of several factors with flight statuses. Our predictions are evaluated on the ground truth data obtained from flight-tracking websites. Our predictions substantially enhance airline scheduling and operations management.**

Keywords—Flight delay prediction, Machine learning, Classification, Random Forest Classification, Logistic Regression, Decision Tree.

I. Introduction

Air travel is an essential part of modern transportation that links distinct locations. However, reliability in flight schedules is often compromised by various factors that, in many cases, induce delays and disruptions. Specifically, there are plenty of airports with multiple airlines serving the same routes, which imposes a challenging task in predicting and managing flight arrivals. This will result in a huge economic loss to the airline companies and Airlines lose their reputation as well. So, proper monitoring and prediction of flight delays are very important. So, our study is mainly focused on the departure delay status and weather parameters to model the flight arrival delay status. The prior prediction helps airlines reduce their loss and also lessens the inconvenience faced by passengers. In this context, the ability to predict and estimate flight arrivals under various conditions such as weather conditions and the status of preceding flights plays a paramount role for airlines and airport authorities.

In this study, we work on predicting the arrival status of flights into Syracuse (SYR) from the airports Chicago (ORD), New York (JFK), and Orlando (MCO). Each of these origin airports has several airlines that fly into SYR. Thus, the research has to do with a detailed analysis of each source of flight data and the factors associated with those flights. We seek to provide very accurate predictions of flight arrival status a few days in advance considering the same-origin flight status.

To achieve this, we use the necessary data collected from the Bureau of Transportation Statistics and other data sources including airline information and weather predictions. We use the following machine learning algorithms: random forest, logistic regression, decision tree, and gradient boosting to model the complex interactions between other factors and flight statuses. We assess the effectiveness of our predictions against ground truth data collected from flight-tracking websites. The prediction model used in this paper consists of two phases. The first phase involves predicting if there is any delay in the arrival status of a flight, and the second phase involves predicting the arrival status of the latter flight based on the arrival status of the early flight. Here we will be stating arrival status as late if the delay time is more than 5 minutes, early if the flight arrives more than 5 minutes before, and in between as on-time. This research will enhance the effectiveness of flight arrival prediction to improve the efficiency and reliability of airline operations, mainly for the sake of passengers and other stakeholders within the aviation industry.

II. Data Collection

A. Flight Data Collection Module

Flight arrival and departure data were collected from the Bureau of Transportation Statistics website (https://www.transtats.bts.gov/ontime/) for the years 2020, 2021, 2022, and 2023. For departure flights, all statistics were chosen, specifying the origin airport as Orlando International (MCO), New York John F. Kennedy International (JFK), and Chicago O'Hare International (ORD) airports, including all months and days, for the years 2021 to 2023. Similarly, for arrival flights, all statistics were extracted, specifying the destination airport as Syracuse Hancock International Airport (SYR), including all months and days, for the years 2020 to 2023. The data includes information such as Carrier Code, Flight Number, Tail Number, Scheduled Departure Time, Actual Departure Time, Scheduled Elapsed Time, Actual Elapsed Time, Delay Carrier, Delay Weather, Delay National Aviation System, Delay Security, and Delay Late Aircraft Arrival.

B. Weather Data Collection Module

The Weather Data Collection module is a critical component of our machine learning project, designed to gather and process weather data which is instrumental in predicting flight timings. This module automates the retrieval of historical and forecast weather data for four major airports: Chicago, New York, Orlando, and Syracuse.

Technical Overview:

- *Data Retrieval:* Utilizing the weatherbit API, the module fetches hourly weather data dating back to January 1, 2020, and also gathers a week's worth of forecast data.
- *API Rate Limitation Handling:* To circumvent the API's rate limitations, multiple tokens were generated and managed, ensuring a continuous and synchronized data collection process.
- *Data Storage:* The collected data is stored in CSV format, with the filename reflecting the location and type of data (historical or forecast).

Module Structure:

- *Initialization:* The WeatherDataMiner class initializes with environment variables defining base URL, API keys, mode

of operation, and other configurations.

- *Data Mining:* The mine_location method iterates through each location, fetching data within the specified date range and handling any API rate limitations encountered.
- *Error Handling:* The module includes robust error handling to manage HTTP errors and retry attempts, ensuring data integrity and continuity of the mining process.
- *Data Processing:* Upon retrieval, the data is flattened and saved to CSV files, with field names dynamically generated to match the data structure.

Efficiency and Reliability:

- *Checkpointing:* The module implements a checkpoint system, saving data at regular intervals to prevent data loss and allow for efficient resumption in case of interruptions.
- *Date Tracking:* A date tracking mechanism keeps a record of the last fetched date, optimizing subsequent data retrieval operations.

The Weather Data Collection module serves as a foundational element of our project, providing reliable and comprehensive weather data that is essential for the accurate prediction of flight timings. Its design reflects a balance between technical sophistication and operational simplicity, making it a robust yet user-friendly tool in our machine-learning arsenal. The module's code is structured to be both efficient and maintainable, ensuring that it can be easily adapted for future projects or enhancements.

Flight Data Preprocessing:

After downloading the data, the arrival and departure datasets were processed using Python's pandas library. Columns were renamed for clarity, the hour was extracted from 'Scheduled departure time', 'Date' was converted to DateTime, and the day of the week, hour, minutes, and year was extracted as integers. A 'dep_order' column with the value 'latter' was added, and irrelevant columns were dropped. To create a comprehensive dataset for analysis, the arrival and departure datasets were merged for the years 2020, 2021, 2022, and 2023. An inner join merge was performed based on common attributes such as Carrier Code, Flight Number, Tail Number, and Date, ensuring that each record corresponds to a unique flight.

Weather Data Preprocessing:

In the preprocessing stage, the weather data was processed to prepare it for analysis. This process involved converting the timestamp to a pandas DateTime object to enable time-based analysis. Additionally, the airport code was extracted from the file name to identify the location of the weather data. The hour component of the timestamp was extracted to facilitate hourly analysis of weather conditions.

Several columns that were not relevant to the analysis were dropped from the dataset, including 'app_temp', 'dhi', 'dni', 'ghi', 'pod', 'slp', 'solar_rad', 'datetime', 'timestamp_local', 'timestamp_utc', 'ts', 'uv', and 'weather.icon'. The remaining columns were renamed to indicate whether they corresponded to arrival or departure data, and the 'Date' column was standardized for consistency across datasets.

The processed weather data was then saved to new CSV files for further analysis. This preprocessing step was crucial for integrating weather information with flight data to investigate the impact of weather conditions on flight delays.

III. Methodology

1. *Combining Flight and Weather Data*

The research project involves preprocessing flight and weather datasets for analysis and predicting the arrival status of flights based on various features. The researchers first imported necessary libraries such as pandas, numpy, and scikit-learn. They then read the flight data from CSV files for flights between specific airports (e.g., MCO-SYR, JFK-SYR, ORD-SYR) and combined them into a single data frame, `main_data`, for analysis.

The preprocessing steps included dropping irrelevant columns and converting timestamps to appropriate datetime formats. The arrival and departure times were parsed to extract hour and minute components and included a day of the week column, which was then converted to categorical variables for analysis. Additionally, categorical variables like `arr_status`and 'dep_status', were created based on the arrival delay in minutes, categorizing delays as late if >5 minutes (2), on-time if -5 to 5 minutes (1), and early if < -5 minutes (0).

Filtering the flight data to include only those with an arrival delay of less than 120 minutes helps focus the analysis on typical delays that are more likely due to common operational factors. After this filtering, the dataset was reduced from 8661 to 8405 flights.

Weather data for each airport (JFK, ORD, MCO, SYR) was processed similarly, converting timestamps and extracting relevant features. The weather data was then merged with the flight data based on the location, date, and timestamp to incorporate weather conditions into the analysis.

The combined dataset was then preprocessed, which included changing data types and removing missing values (NaNs). Unwanted columns were dropped to focus on relevant features.

2. *Plots*

Histograms were plotted for each column to analyze the distribution of data.

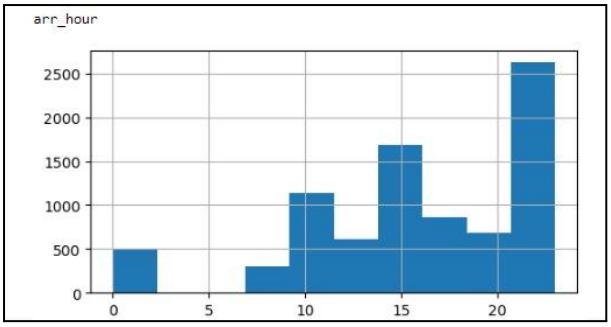

Fig. 1. Histogram showing the count of values across arr_hour

Additionally, scatter plots were created for each column against the arrival status to visualize any potential relationships. The columns dep_precip and arr_precip were modified by squaring the values, as the scatter plots suggested a quadratic (non-linear) relationship with the arrival status. However, no such pattern was observed with other columns.

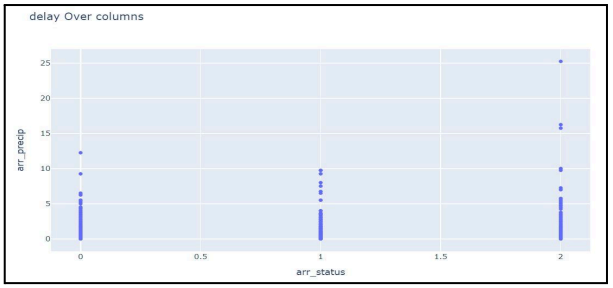

Fig. 2. Scatter plot for arr_precip against arr_status

A heatmap was generated to explore the correlation between all numeric values. It was found that the arrival status is highly correlated(0.65) with the departure status. However, since the departure status column won't be available for future data (as it depends on future events), it cannot be used in the model.

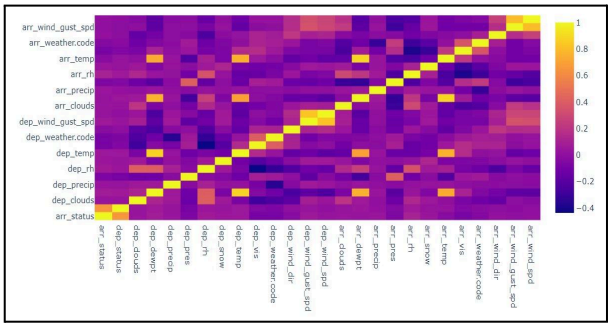

Fig. 3. Heatmap showing the correlation between numeric features

3. *Training Model*

The final columns selected for the arrival status model include dep_hour, dep_day, Origin_Airport, arr_hour, arr_day, dep_min, arr_min, dep_weather.code, arr_weather.code, along with various weather-related features such as dep_clouds, dep_dewpt, dep_precip, dep_temp, dep_vis, dep_wind_dir, dep_wind_gust_spd, dep_wind_spd, arr_clouds, arr_dewpt, arr_precip, and arr_temp, arr_vis. These features are crucial for predicting the arrival status of flights accurately.

The dataset was preprocessed by encoding categorical columns using the one-hot encoding (get_dummies) with drop_first=True. The data was then split into training and testing sets in an 80:20 ratio, with stratification based on the arrival status column (arr_status) to ensure a balanced distribution of classes in both sets.

The data was scaled using StandardScaler to standardize the features. Three models were trained and evaluated: logistic regression, random forest, and gradient boosting.

For logistic regression, the best parameters were found to be fit_intercept=True, solver='lbfgs', multi_class='ovr', penalty=None, and max_iter=1000. This model achieved an accuracy of 0.5886 on the training set and 0.5443 on the test set.

For the random forest model, the best parameters were random_state=50, min_samples_leaf=4, max_features='sqrt', and n_estimators=500. This model achieved an accuracy of 0.7940 on the training set and 0.5455 on the test set.

Lastly, for the gradient boosting model, the best parameters were random_state=50, min_samples_split=8, min_samples_leaf=4, and n_estimators=300. This model achieved an accuracy of 0.7125 on the training set and 0.5384 on the test set.

4. *Hybrid Model*

4.1. Predicting Departure Status

As the departure status was highly correlated with the arrival status, the hybrid model was constructed to predict departure status (`dep_status`) first, which was then included as a feature along with other relevant columns to predict arrival status (`arr_status`). The columns used for predicting departure status were:

`['dep_hour', 'dep_day', 'Origin_Airport', 'arr_hour', 'arr_day', 'dep_min', 'arr_min', 'dep_status', 'dep_clouds', 'dep_dewpt', 'dep_precip', 'dep_pres', 'dep_rh', 'dep_snow', 'dep_temp', 'dep_vis', 'dep_weather.code', 'dep_wind_dir', 'dep_wind_gust_spd', 'dep_wind_spd', 'arr_clouds', 'arr_dewpt', 'arr_precip', 'arr_pres', 'arr_rh', 'arr_snow', 'arr_temp', 'arr_vis', 'arr_weather.code', 'arr_wind_dir', 'arr_wind_gust_spd', 'arr_wind_spd']`

The hybrid model creation involved these key steps: converting categorical columns into dummy variables, splitting the dataset into 80% training and 20% testing with stratification, scaling the data, and training four models—logistic regression, decision tree, random forest, and gradient boosting—to predict departure status.

The logistic regression model, configured with `fit_intercept=True`, `solver='lbfgs'`, `multi_class='ovr'`, `penalty=None`, and `max_iter=1000`, achieved an accuracy of 0.5684 on the training set and 0.5235 on the test set.

The decision tree model, with `random_state=50` and `min_samples_leaf=4`, performed better with an accuracy of 0.8190 on the training set but slightly lower at 0.4194 on the test set.

The random forest model, utilizing `random_state=50`, `min_samples_leaf=4`, `max_features="sqrt"`, and `n_estimators=300`, had an accuracy of 0.7958 on the training set and 0.5068 on the test set.

Lastly, the gradient boosting model, with `random_state=50`, `min_samples_split=8`, `min_samples_leaf=4`, and `n_estimators=300`, achieved an accuracy of 0.7008 on the training set and 0.5092 on the test set.

The logistic regression model outperformed the other models in predicting the departure status.

4.2. Predicting Arrival Status based on Departure Status

For the task of predicting the arrival status (including the departure status column as ground truth), a logistic regression model was trained using various features. The features included are:

['dep_hour', 'dep_day', 'Origin_Airport', 'arr_hour', 'arr_day', 'dep_min', 'arr_min', 'arr_status', 'dep_status', 'dep_clouds', 'dep_dewpt', 'dep_precip', 'dep_pres', 'dep_rh', 'dep_snow', 'dep_temp', 'dep_vis', 'dep_weather.code', 'dep_wind_dir', 'dep_wind_gust_spd', 'dep_wind_spd', 'arr_clouds', 'arr_dewpt', 'arr_precip', 'arr_pres', 'arr_rh', 'arr_snow', 'arr_temp', 'arr_vis', 'arr_weather.code', 'arr_wind_dir', 'arr_wind_gust_spd', 'arr_wind_spd']

The dataset was preprocessed by converting categorical variables into dummy variables, splitting it into training and testing sets in an 80:20 ratio with stratification, and scaling the data.

The logistic regression model, configured with fit_intercept=True, solver='lbfgs', multi_class='ovr', penalty=None, and max_iter=1000, achieved an accuracy of 0.7424 on the training set and 0.7067 on the test set. This indicates that the model performed well and did not overfit the data.

Additionally, decision tree, random forest, and gradient

boosting models were trained for comparison. The decision tree model, with random_state=50 and min_samples_leaf=3, had a high accuracy of 0.9007 on the training set but lower at 0.6169 on the test set, suggesting some overfitting.

The random forest model, utilizing random_state=50, min_samples_leaf=4, max_features="sqrt", and n_estimators=500, achieved an accuracy of 0.7985 on the training set and 0.7079 on the test set.

Out[62]:

	0
dep_status_2	0.29
dep_status_1	0.06
dep_dewpt	0.03
dep_temp	0.03
arr_rh	0.03

Fig. 4. Feature importance for random forest model

The gradient boosting model, with random_state=50, min_samples_split=8, min_samples_leaf=4, and n_estimators=500, achieved an accuracy of 0.8551 on the training set and 0.6978 on the test set.

Overall, the random forest model outperformed the other models, demonstrating high accuracy on the test data without overfitting.

With the hybrid model, the score improved from 0.5455 to 0.5622 on the test data split. And for the entire dataset, the score is 0.6136.

5. *Predicting Arrival Status based on Departure status and Arrival status of the earlier flight*

Latter Flight Prediction involved preprocessing steps similar to Early Flight Prediction. The data was divided into latter and early flights based on location and carrier code, and a left merge was performed on the latter data with the early data by date and location. Columns were filtered where the latter flight time was next and within 3 hours of the early flight.

For the latter flight, all columns from Early Flight Predictions were considered, along with the arrival status of the early flight. This resulted in a dataset with 893 rows. Weather data was then fetched and merged based on the date, hour, and location of the latter flight. The preprocessed dataset included the following columns:

['dep_hour', 'dep_day', 'Origin_Airport', 'arr_hour', 'arr_day', 'dep_min', 'arr_min', 'dep_status', 'arr_status_x', 'arr_status_y', 'arr_clouds', 'arr_dewpt', 'arr_precip', 'arr_pres', 'arr_rh', 'arr_snow', 'arr_temp', 'arr_vis', 'arr_weather.code', 'arr_wind_dir', 'arr_wind_gust_spd', 'arr_wind_spd', 'dep_clouds', 'dep_dewpt', 'dep_precip', 'dep_pres', 'dep_rh', 'dep_snow', 'dep_temp', 'dep_vis', 'dep_weather.code', 'dep_wind_dir', 'dep_wind_gust_spd', 'dep_wind_spd'] where arr_status_x is for the latter flight and arr_status_y is for the early flight.

A hybrid approach similar to Early Flight Predictions was followed, where the model first predicted the departure status and then used it along with the early flight arrival status to predict the arrival status of the latter flight.

So we first trained the model to predict departure status by converting categorical variables to dummy variables, split the data into training and testing datasets, and scaled the data.

Logistic regression was used to predict the departure status, while random forest was chosen to predict the arrival status.

The logistic regression model for predicting departure status, with parameters `fit_intercept=True, solver='lbfgs', multi_class='ovr', penalty=None, max_iter=1000`, achieved a training accuracy was 0.5683, and testing accuracy was 0.5222.

For predicting the arrival status, the logistic regression model, including departure status and early flight arrival status, with parameters `fit_intercept=True, solver='lbfgs', multi_class='multinomial', penalty=None, max_iter=1000`, achieved a training accuracy of 0.7924 and a test accuracy of 0.6045.

A decision tree model with parameters `random_state=50, min_samples_leaf=3` had a training accuracy of 0.8941 and a test accuracy of 0.5254.

The random forest model, with parameters `random_state=50, min_samples_leaf=4, max_features="sqrt", n_estimators=300`, achieved a training accuracy of 0.8432 and a test accuracy of 0.6780.

IV. Results And Discussion

A. *Results of Predicting Arrival Status for Earlier Flights*

The addition of the departure status column significantly improved the performance of the models. For the logistic regression model, the accuracy on the test set increased from 0.5443 to 0.7067, for random forest from 0.5455 to 0.7079, and for gradient boosting from 0.5384 to 0.6978. This indicates that incorporating the departure status as a predictor enhanced the models' predictive power.

Next, the combined hybrid model was tested on the test data split for arrival status prediction. The process involved getting dummies and scaling the data, predicting the departure status using the logistic regression model, adding the predicted column to the original data, converting the categorical variables into dummy variables and scaling the data, and then predicting the arrival status using the random forest model. The hybrid model achieved a score of 0.5622, which is an improvement from 0.5455 without the hybrid model.

The hybrid model achieved an overall test score of 0.6136 when evaluated on the entire dataset. For predicting actual results, the process involved retrieving data from a CSV file, preprocessing it, merging it with weather forecast data, applying the logistic regression departure status model to add the departure status column, and then applying the random forest arrival status model to predict the arrival status. Out of 12 predictions, 8 were correct, resulting in a prediction accuracy of 66.67%. This indicates that while the hybrid model improved the overall prediction score, there is still room for further improvement in prediction accuracy.

B. *Results of Predicting Arrival Status for Latter Flights*

Moving forward with the random forest model, the next step was to predict the results. The latter flight data was retrieved, and a logistic regression model was applied to predict the departure status. The predicted departure status column was then added to the original data. This augmented dataset was merged with three possible values of the early flight arrival status. Subsequently, the random forest model was used to predict the arrival status. The final prediction on the test dataset resulted in 5 out of 11 correct predictions for the arrival status, yielding an accuracy of 45.45%. Overall, Combined with the results from Early Flight Predictions,

the overall prediction accuracy was 56.52%, with 13 correct predictions out of 23.

V. Future Work

The project has certainly revealed numerous opportunities for further exploration and improvement. The main focus would be on enhancing the predictive accuracy of the model. This could be achieved by integrating more features into the model, such as comprehensive historical flight data, airport metadata like capacity and the number of runways, and real-time airport traffic conditions.

We have trained a model including the air traffic as a calculated column by getting a count of flights arriving and taking off in a particular hour by grouping on date and hour at each airport. It did not increase the accuracy. Maybe we need more relevant real-time air traffic data including all the flights.

Expanding the model to include a broader range of airports is another promising direction for future work. Currently, the model is limited to flights to Syracuse from Chicago, New York, and Orlando. Incorporating flights from additional cities could enhance the model's robustness and applicability.

Moreover, refining the model to predict not just the flight status (early, on time, or delayed) but also the extent of the delay could provide passengers with more nuanced and actionable information.

In terms of machine learning techniques, there is potential to explore more advanced methodologies, such as deep learning techniques. These are particularly effective for prediction tasks involving sequential data, an essential aspect of our project. This exploration promises to bring us closer to our objective of creating a highly accurate and robust flight delay prediction model. We eagerly anticipate the journey ahead.

VI. Conclusion

To sum up, this project has been a notable attempt to apply machine learning methods to a real-world issue. The team built a model that can accurately forecast flight statuses. Advanced machine learning algorithms like Logistic Regression, Random Forest Classifier, and Gradient Boosting Classifier were utilized to construct the model.

The project used flight and weather datasets and compared them with given inputs, validating them using classification and regression concepts of Machine Learning. The team conducted feature extraction, addressed missing values with suitable methods, and employed sampling techniques for imbalanced data. Moreover, adjusting the hyperparameters resulted in improved accuracy of the model.

This project has highlighted the capacity of machine learning to bring about significant change in addressing problems. We managed to create a model that offers valuable insights by utilizing data from different sources. This improves our comprehension of the reasons behind flight delays and lays the groundwork for later creating advanced prediction models.

Although there is room for improvement in the model's performance, it is crucial to keep in mind that machine learning is a continuous process. Every cycle takes us nearer to our objective. We believe that we can improve the model's effectiveness by making further improvements and expanding. This project has been a stepping stone leading us closer to that objective, and we eagerly look forward to the upcoming journey. Exploring more advanced machine learning techniques like deep learning shows potential in developing a highly accurate and robust model for predicting flight delays, bringing us closer to our goal. This project has served as a milestone in reaching that objective, and we are excited for the path ahead.

VII. Acknowledgments

We would like to express our sincere gratitude to Professor Natarajan Gautam, for his invaluable guidance, continuous support, and expertise throughout this project.