

Advancing a Major U.S. Airline's Practice in Flight-level Checked Baggage Prediction

Shijie Chen¹, Chiwoo Park¹, Qianwen Guo¹, Yanshuo Sun^{1*}

^{1*}Department of Industrial and Manufacturing Engineering, FAMU-FSU College of Engineering, Florida State University, 2525 Pottsdamer Street, Tallahassee, FL, 32312, United States.

^{2*}Department of Civil and Environmental Engineering, FAMU-FSU College of Engineering, Florida State University, 2525 Pottsdamer Street, Tallahassee, FL, 32312, United States.

*Corresponding author(s). E-mail(s): y.sun@eng.famu.fsu.edu;

Contributing authors: sc20hw@fsu.edu; chiwoo.park@eng.famu.fsu.edu;
qguo@eng.famu.fsu.edu;

Abstract

In this paper, we aim to address a relevant estimation problem that aviation professionals encounter in their daily operations. Specifically, aircraft load planners require information on the expected number of checked bags for a flight several hours prior to its scheduled departure to properly palletize and load the aircraft. However, the checked baggage prediction problem has not been sufficiently studied in the literature, particularly at the flight level. Existing prediction approaches have not properly accounted for the different impacts of overestimating and underestimating checked baggage volumes on airline operations. Therefore, we propose a custom loss function, in the form of a piecewise quadratic function, which aligns with airline operations practice and utilizes machine learning algorithms to optimize checked baggage predictions incorporating the new loss function. We consider multiple linear regression, LightGBM, and XGBoost, as supervised learning algorithms.

We apply our proposed methods to baggage data from a major airline and additional data from various U.S. government agencies. We compare the performance of the three customized supervised learning algorithms. We find that the two gradient boosting methods (i.e., LightGBM and XGBoost) yield higher accuracy than the multiple linear regression; XGBoost outperforms LightGBM while LightGBM requires much less training time than XGBoost. We also investigate the performance of XGBoost on samples from different categories and provide insights for selecting an appropriate prediction algorithm to improve baggage prediction practices. Our modeling framework can be adapted to address other prediction challenges in aviation, such as predicting the number of standby passengers or no-shows.

Keywords: Aviation, Airline operations, Checked baggage, Supervised learning, Custom loss function

1 Introduction

The air transportation system is a complex network of internal and external entities that interact with one another in a variety of ways. For example, passengers can choose from multiple flights operated by competing airlines, airlines plan and schedule their flights to minimize operating costs, and air traffic controllers ensure safety and minimize delays for flights sharing the airspace. The air transportation system is also impacted by external factors such as weather and socio-economic systems (Jardines, Soler, & García-Heras, 2021), which contribute to the generation of large and intricate datasets. By analyzing these datasets effectively, there is great potential for the aviation sector to undergo significant transformation. The Next Generation Air Transportation System (NextGen) is an initiative led by the National Aeronautics and Space Administration (NASA) and Federal Aviation Administration (FAA) to use advanced data mining tools to identify safety vulnerabilities across various types of data (NASA, 2007). Outside of air traffic management, airlines have also been known for improving revenues by tailoring their services and products to various customer segments based on analyses of their very rich passenger booking data (Sun et al., 2018). This paper aims to explore how advanced machine learning algorithms and large-volume operational data can revolutionize conventional practices in airline operations, with a particular focus on the flight-level checked baggage prediction problem, to be defined and analyzed next.

A few hours (e.g., three to four hours) before the departure of a passenger flight, load planners need to decide what shipments booked on this flight can be selected for loading or off-loading, how selected shipments can be packed into unit load devices (ULD) or palletized, and how ULDs or pallets can be assigned to different positions on the aircraft (Brandt & Nickel, 2019). Those loading decisions depend on how much checked baggage to anticipate, whose exact amount only becomes available shortly (such as 45 minutes) before the scheduled departure. Thus, the expected number of checked bags must be estimated in advance, to facilitate the aircraft load planning process where various weight and balance limits should be enforced. For instance, the maximum cargo (including checked baggage) to carry should be jointly determined with fuel load subject to the takeoff weight limit (FAA, 2016b). From the perspective of balance control, flight safety concerns may arise from baggage issues. A shift in the longitudinal center of gravity, caused by improperly loaded baggage, may result in unstable aircraft conditions or control difficulty (FAA, 2016a). In addition, loading checked bags can be time consuming and labor intensive. Having relatively accurate checked baggage information would allow an airline to allocate appropriate human power to load checked bags onto an airplane (BEONTRA, 2022). In case

of an unusually large number of bags, the baggage loading process would be prolonged, thus likely delaying a flight. While a good estimation of checked baggage is highly desirable, it is difficult to achieve such a prediction because the amount of checked baggage usually fluctuates substantially, due to the influence of various factors, such as passenger trip purposes and travel seasons.

A detailed literature review (to be presented in Section 2) indicates that only a few studies have focused on the checked baggage prediction problem and each of them has various shortcomings. For instance, non-time series data cannot be easily incorporated into a time series forecasting model for baggage prediction as in [Ma, Bi, Sai, and Li \(2021\)](#). The primary shortcoming of the existing literature is that the consequences of over-prediction and under-prediction are treated equally, contrary to aviation practice. Modeling the asymmetric effect is important in the context of baggage prediction, as the cost of not having sufficient space to accommodate checked baggage is much higher than having a partially filled checked baggage compartment.

In this study, we analyzed a checked baggage dataset consisting of nearly one million samples owned by a major U.S. airline. The primary dataset was enriched by including additional features, such as the population of a city and passenger enplanements of an airport. Next, we performed necessary data cleaning and feature processing to avoid the negative impact of data irregularities on the subsequent analyses and predictions. We introduced three prediction methods, namely linear regression, and two advanced supervised learning algorithms (i.e., LightGBM and XGBoost). We further customized them by designing and incorporating an asymmetric loss function. Some findings from the experiments are highlighted as follows: (1) LightGBM and XGBoost can achieve a much better predictive performance than linear regression regardless of whether the asymmetric loss function is considered or not, while XGBoost slightly outperforms LightGBM; (2) LightGBM is significantly more efficient than XGBoost; (3) the predictive performance of XGBoost is not uniform over destination airports or fleet types due to uneven training data coverage.

To the best knowledge of the authors, this is the first flight-level checked baggage prediction study considering an asymmetric loss function. This study also demonstrates the great potential of state-of-the-art machine learning algorithms in modernizing some conventional practices in airline operations with large-scale real-world data.

The rest of this paper is organized as follows. Section 2 briefly reviews some machine learning applications in air transportation. We next describe how exploratory analyses of the datasets are conducted along with data cleaning and feature engineering in Section 3. Then, in Section 4, multiple prediction methods of various complexities are introduced and the asymmetric loss function is presented. Experiments are conducted and major results are reported in Section 5. A discussion of how to choose an appropriate baggage prediction algorithm is presented in Section 6. Lastly, concluding remarks and future research directions are given in Section 7.

2 Literature Review

We divide related machine learning applications in air transportation into two broad categories: flight-related and payload-related. In the first group, we mainly review those predictive studies on aircraft trajectory and flight delay; in the second group, we focus on those studies on passenger, cargo, and checked baggage predictions.

2.1 Flight-related problems

2.1.1 Aircraft trajectory prediction

In traditional Air Traffic Management (ATM), controllers only need to know the current location of aircraft, while under the new Trajectory-Based Operations (TBO) paradigm, controllers require information on the future location of aircraft as well (Mondoloni & Rozen, 2020). This is where four-dimensional (4D) Trajectory Prediction (TP) is needed (Wang, Liang, & Delahaye, 2020; Wu, Yang, Chen, Hu, & Hu, 2022). 4D TP, which involves predicting an aircraft's longitude, latitude, altitude, and time, is crucial for improving air traffic safety and ATM efficiency under TBO. This is because an aircraft's actual trajectory can deviate from its planned route due to various factors such as congestion and weather conditions. Trajectories can be predicted either in the short term (in-flight) or long term (pre-flight). Two representative studies are reviewed next.

In a short-term predictive study, Gallego et al. (2019) proposed a measure of probabilistic interdependence for pairs of aircraft, which they used as a feature to predict the vertical profiles of aircraft trajectories during the descent phase of flight. Their study employed data from the Barcelona Air Traffic Control Center in Spain and demonstrated the usefulness of their neural network methods. The researchers concluded that incorporating the proposed interdependence measure enhanced the accuracy of trajectory prediction. In contrast, Wu et al. (2022) focused on predicting the 4D trajectory of an aircraft prior to takeoff using historical trajectory data. To achieve this, they converted historical time series data into images and used various types of generative adversarial networks (GANs) to generate new images, which were then transformed into time series data or flight trajectories. The study used data from flights between Beijing and Chengdu in China and compared different GAN variants based on training time, prediction time, and prediction accuracy. The researchers found that the one-dimensional convolution variant of GAN produced the best results.

2.1.2 Flight delay

The issue of flight delays has a significant impact on airline operations and customer satisfaction, leading many researchers to focus on predicting flight delays using machine learning. For example, Khan, Ma, Chung, and Wen (2021) observed that current flight delay classification systems use multiple threshold prediction classifiers running in parallel, which can lead to conflicting results and ambiguity. Instead, they proposed a sequential approach to predicting whether flight delays exceed certain thresholds by considering only part of the data. They demonstrated their prediction algorithms with a case study of a Hong Kong-based airline. In contrast, Rodríguez-Sanz et al. (2019) used a Bayesian Network approach to predict delays for the arrival systems of an airport. They also conducted a reliability analysis using a Markov chain approach to evaluate the system's reliability. Their model can adaptively capture the stochastic characteristics of arrival processes. As it is beyond the scope of this study to further review those flight-related machine learning studies, interested readers are directed to Chung, Ma, Hansen, and Choi (2020).

2.2 Payload-related problems

2.2.1 Passengers and air cargo

Passengers account for the majority of payload on commercial flights. Passenger demand predictions have been conducted at different levels, such as the national or airport level. [S. Xu, Chan, and Zhang \(2019\)](#) combined seasonal autoregressive integrated moving average (SARIMA) and support vector regression (SVR) for the prediction of several aviation metrics for China's aviation industry, such as domestic and international passenger miles. They used 133 observations between February 2005 to February 2016 to predict values from March 2016 to February 2018. While such predictions at a national level are useful for strategic planning purposes, other researchers have focused on operational problems at a lower level. For instance, to dynamically optimize the configurations of airport security screening lanes, [Hanumantha, Arici, Sefair, and Askin \(2020\)](#) proposed an ensemble forecasting model, which was tested in a case study of Phoenix Sky Harbor Airport.

It is also very common for passenger flights to carry air cargo, which is organized using pallets or unit load devices (ULDs) and loaded to the belly of the fuselage under the passenger compartment. Such cargo space is shared by air cargo and checked baggage, while the latter has priority over the former. This implies the available belly space for air cargo is uncertain. Therefore, [Tseremoglou, Bombelli, and Santos \(2022\)](#) used Long Short-Term Memory Networks (LSTMs) to predict available cargo space. They then solved a real-time booking acceptance problem with the predicted values as inputs. Their experiments indicated that their proposed method can significantly increase loaded volume while decreasing unplanned offloading. A comprehensive review of the application of machine learning in air cargo management can be found in [Barua, Zou, and Zhou \(2020\)](#).

While predicting cargo and passenger demand seems similar, the information provided by cargo tracking and passenger service systems could be quite different, implying that models for passengers and air cargo are not immediately transferable. Gender, age, and travel purpose are examples of common passenger characteristics that do not apply to cargo predictions.

2.2.2 Checked Baggage

As the most relevant ones to this study, a few papers tried to predict the amount of checked baggage. Accurate prediction of checked baggage is fundamental for reasonable resource allocation to prevent the overloading of the baggage handling system ([Ma et al., 2021](#)) and allocate staff resources ([BEON-TRA, 2022](#)). A seasonal ARIMA model was adopted by [Ma et al. \(2021\)](#) for checked baggage prediction at the airport level. Based on the historical baggage volume covering eight weeks, the demand for the subsequent three days was predicted. They quantified the predictive ability with some metrics but no alternative models were involved for benchmarking purposes. [Mikram, Rhanoui, Yousf, and Briwa \(2020\)](#) also adopted the ARIMA model to forecast the baggage volume and found that the BOXJENKINS approach and exponential smoothing methods can improve the accuracy. One shortcoming of these time series models is that they rely on time-series data only, which implies a lot of available passenger-related characteristics cannot be incorporated. In addition, those predictions are intended to be at the airport level, rather than the flight level.

[Cheng, Gao, and Zhang \(2014\)](#) forecast the baggage volume for each flight from an international terminal using a back propagation neural network and multiple linear regression. Numerical experiments

on three datasets of various sizes were conducted. Only five features were involved, namely passenger count, flight date, flight type, departure time, and flight duration. This study had a quite limited sample size. In predicting the baggage amount for a single flight, only 29 samples were used, among which 21 were for training and 8 were for testing. The entire dataset covering all flights had 3,040 samples. Given the limited number of features and sample size, the resulting R^2 is also quite low, i.e., slightly over 0.5.

The identification of mishandled bags was studied by [van Leeuwen et al. \(2020\)](#) using a Gradient Boosting machine, which was not reviewed in detail due to its low relevance.

2.3 Summary

The large amount of data generated in the aviation industry has led to extensive developments of machine learning algorithms for aviation transportation management ([Y. Xu, Wandelt, & Sun, 2024](#)). A good variety of aviation problems have been studied with machine learning ([Chung et al., 2020](#)). However, only a few studies were for predicting the amount of checked baggage. A shortcoming in the time series models for checked baggage prediction is that the impact of many non-time series data on passenger baggage volume cannot be incorporated. More importantly, none of those studies have explored an asymmetric loss function in predictive analyses. This paper distinguishes it from those studies by presenting a flight-level checked baggage prediction method based on the customization of advanced machine learning algorithms. Another strength of the paper lies in the large-scale and multi-source datasets covering a whole country being analyzed in this study.

3 Data

3.1 Data Sources and Overview

The flight baggage dataset used in this study was obtained from a major U.S. airline, referred to as Airline λ for confidentiality purposes due to the Non-Disclosure Agreement. The dataset consisted of 926,395 domestic flights operated by the airline in 2019, pre-dating the COVID-19 pandemic. Each row in the dataset represented a flight on a specific day, with operational details and aggregated passenger counts being available. The dataset consisted of 22 features related to flights, including flight number, departure and arrival date and time, origin and destination cities, aircraft type, leg distance, and the total number of on-board passengers. Additionally, there were 27 other features related to passenger characteristics, such as payment type, presence of a child or infant, booking method, and cabin class. Airline λ also categorized their passengers based on their loyalty status and predicted travel purposes by other undisclosed algorithms. All those passenger-related features combined presented valuable insights into understanding who those customers were, why they traveled, and how they checked bags. The dataset contained a single target variable, which was the number of checked bags on each flight.

While the baggage dataset from Airline λ was essential, additional data were obtained to enhance the predictive performance of machine learning algorithms. It was because the baggage dataset did not have key information about airports, cities, and air travel trends over time. For instance, the baggage dataset itself did not provide any information on how the population of Houston, Texas compared with that of Omaha, Nebraska; it did not tell how the Reagan National Airport's enplanements differed from the Dulles International Airport serving the same metropolitan area; it did not properly reflect

the difference between Thanksgiving travels and New Year travels. Therefore, the following new features were added: city population from the U.S. Census Bureau (U.S. Census Bureau, 2021), annual enplanements by airports provided by the Federal Aviation Administration (FAA, 2022), and the TSA (Transportation Security Administration) checkpoint travel numbers (TSA, 2022). Figure 1 shows the ten busiest airports in 2021 by enplanement as well as the population of the city served by an airport. By enplanement, the Atlanta Airport was followed by the Dallas and Denver airports. As no enplanement data were found for 2019, and to avoid considering the disruptions of the COVID-19 pandemic in 2020, the 2021 data were used instead. For the TSA daily count, historical data were found for 2019 and thus used. Figure 2 shows the air travel peaks in summer months, especially July and August, as well as off-peaks in February and September.

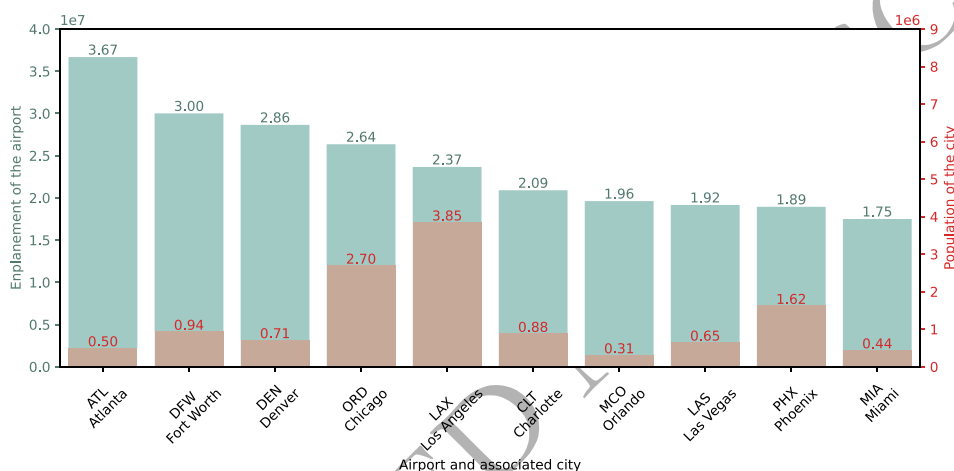


Fig. 1: Top 15 busiest airports in U.S. in 2021 by enplanement

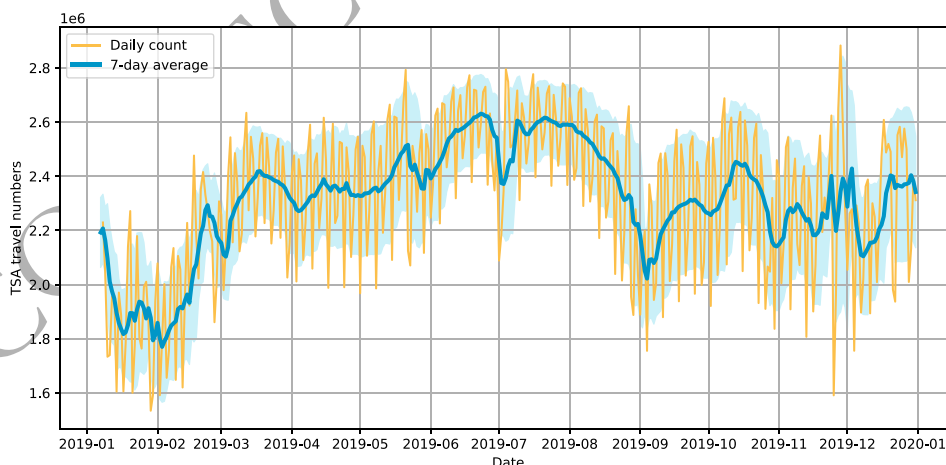


Fig. 2: TSA checkpoint travel numbers in 2019

3.2 Data cleaning

According to Airline λ , the baggage dataset was real and unaltered. Although we did not find any missing values or duplicate records, some irregularities in the data were identified. Therefore, we used the following procedure to clean the baggage dataset. First, any rows with negative passenger counts were dropped. Second, if the aggregated passenger count over different categories exceeded the total count, relevant data were dropped. For instance, if the number of local and connecting passengers exceeded the total number of onboard passengers, the corresponding sample was removed. Similarly, the number of passengers in the main cabin cannot be greater than the total passenger count. Third, if the passenger and bag counts were conflicting, relevant data were removed. For instance, certain flights had a few even zero passengers while carrying over a hundred checked bags. If either the passenger or bag count was lower than ten, we removed such flights as outliers. After data cleaning, 722,556 records remained.

3.3 Data explorations

We next conducted several exploratory analyses to illustrate some important relations in the baggage dataset, which would have been hidden otherwise. Figure 3 shows how the target variable varies with two passenger-related features, namely the number of paying passengers and the number of main cabin passengers, as well as the histograms for those two features. In Figure 3, the solid line represents the average bag count while the standard deviation of bag count is visualized by the buffer zone along the solid line. Those two solid lines seem very similar because of a high correlation between those two corresponding features. The majority of passengers pay for their travels, thus classified as paying passengers, and opt for the main cabin, thus classified as the main cabin passengers. While the ratio of revenue passengers to main cabin passengers varies, the average ratio is 0.9. It is also evident that as the number of passengers grows, the number of bags grows in general, especially before the number of revenue passengers hits 196. When there are more than 196 passengers on a flight, the relation between the bag and passenger counts is no longer smooth and the standard deviation also increases. The cutoff line represents the capacity of the largest narrow-body airplane used by the airline. Wide-body aircraft are capable of carrying more passengers than 196, while they are used for only 1.85% of all the flights in the dataset, as shown in Figure 4. The two histograms in Figure 3 also indicate very few observations for wide-body aircraft. Given the limited sample size for wide-body aircraft, the bag-passenger relation is not characterized adequately by the current data. It is also notable that when the number of paying passengers is within 10, the bag count is nearly 50, implying an unusually large bag-to-passenger ratio.

Figure 5 indicates that wide-body aircraft carry significantly more checked bags as expected. Figure 6 presents the distribution of load factor, i.e., the total number of passengers divided by the aircraft capacity, which shows that in most cases, aircraft are nearly fully loaded.

We next explore how the counts of bags and passengers vary over time at different granularity levels (i.e., month, day of the week, and hour of the day). The way how the bag count varies over months is very similar to the national air travel trend shown in Figure 2. From Figure 7, we observe more bags in summer months, such as June and July, while fewer bags are observed in September and October. In December, the bag count is the highest. This could be partially affected by school sessions. For example, there might be more family travels for recreational purposes over the summer and winter breaks, contributing to the bag surges. Figure 7 further shows that there are more checked bags on

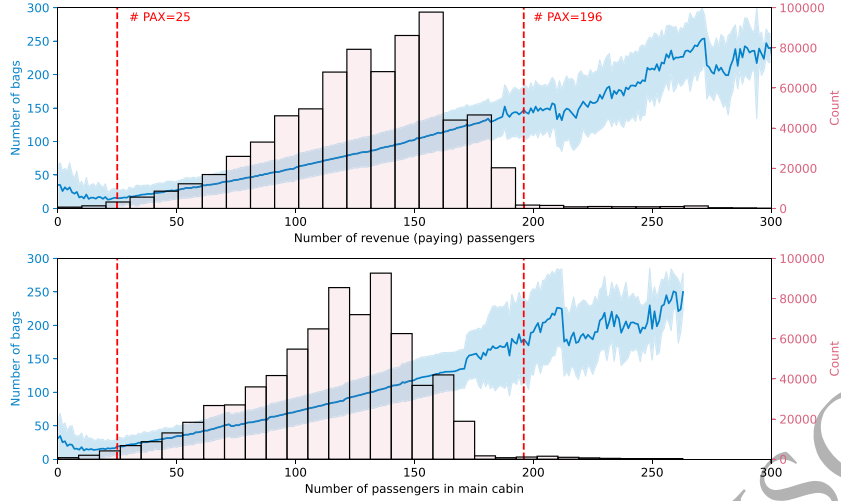


Fig. 3: Bag count vs passenger count

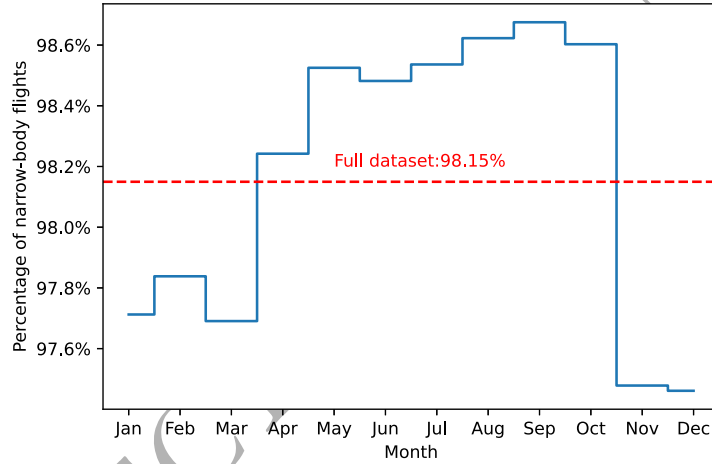


Fig. 4: Percentage of flights using narrow-body aircraft over months

Saturday and Sunday flights and fewer on Wednesday and Thursday. As Airline λ did not operate any flights around 3 and 4 am, there is a gap in how the bag count varies with the hour of the day in Figure 7. Flights carried fewer bags around noon and more bags close to midnight. This pattern may reflect a combination of operational scheduling by airlines and passenger preferences, with late-night flights possibly accommodating more long-haul or international travelers who tend to check more baggage. Additionally, the trend of increased baggage volumes close to midnight could be influenced by passengers choosing late-night flights for economic reasons or personal scheduling needs.

Figure 8 shows how passengers booked their tickets over time, for an example flight from a major hub (pseudonymized as Hub1) of Airline λ to New York City and a selected aircraft type Boeing 737. Approximately, 30% of passengers booked their tickets one month in advance, and 65% of passengers booked at least one week in advance. At the time of departure, 83.2% of tickets were sold, which was close to the average load factor shown in Figure 6.

Collinearity refers to a linear relationship between two or more predictor variables. This issue would become a concern in some regression models since the significance level of both variables will

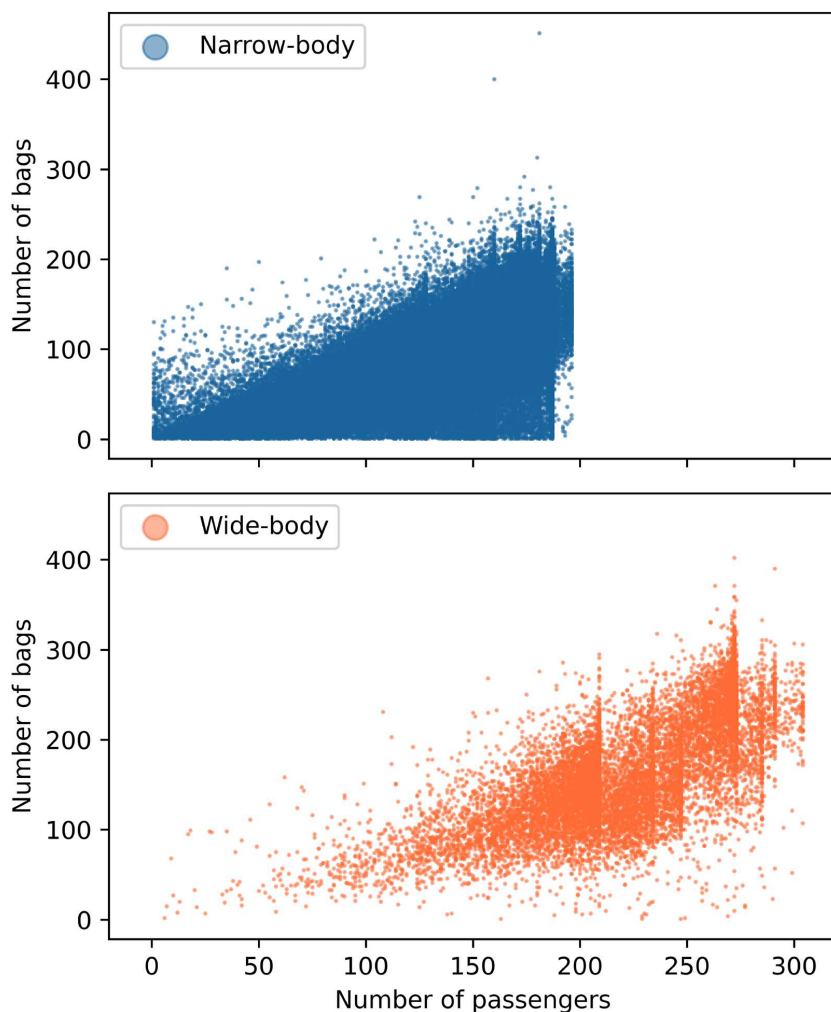


Fig. 5: Number of bags by aircraft type

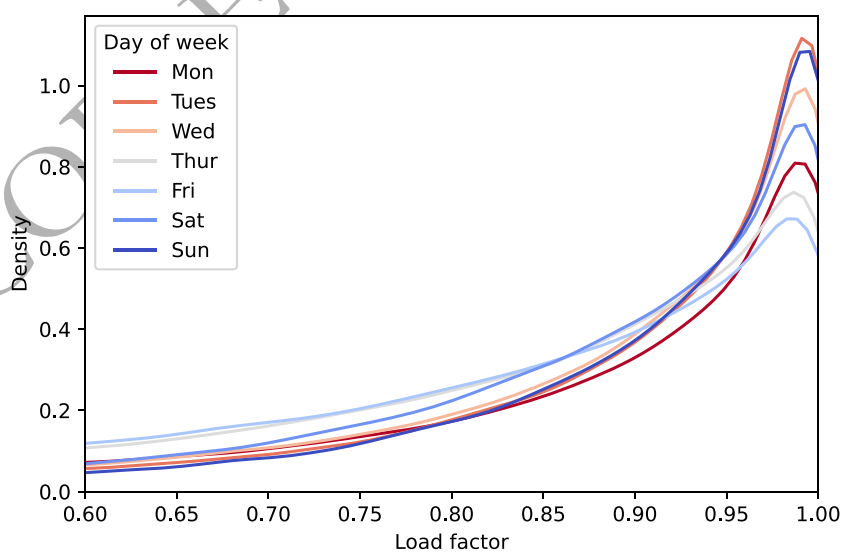
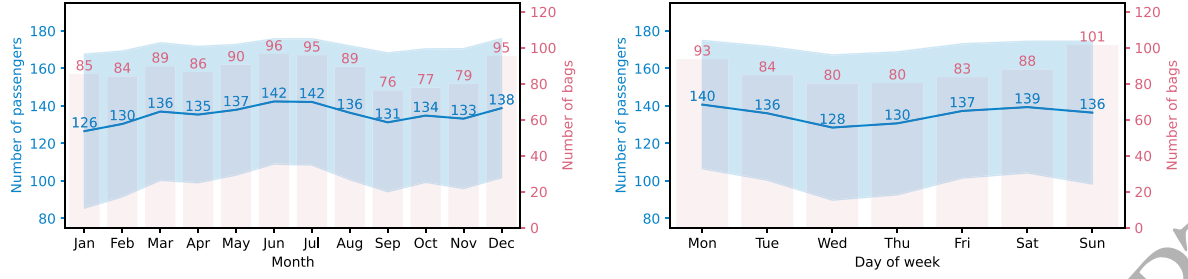
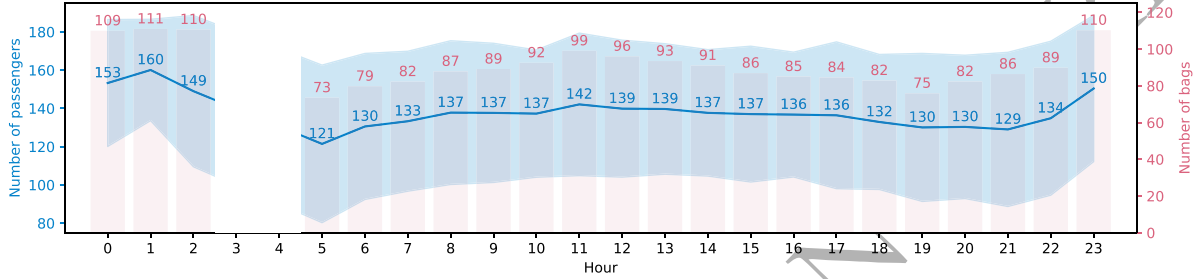


Fig. 6: Density distribution for load factor over days of the week



(a) Number of bags and passengers over months

(b) Number of bags and passengers over days



(c) Number of bags and passengers over hours

Fig. 7: Passenger and bag counts over time

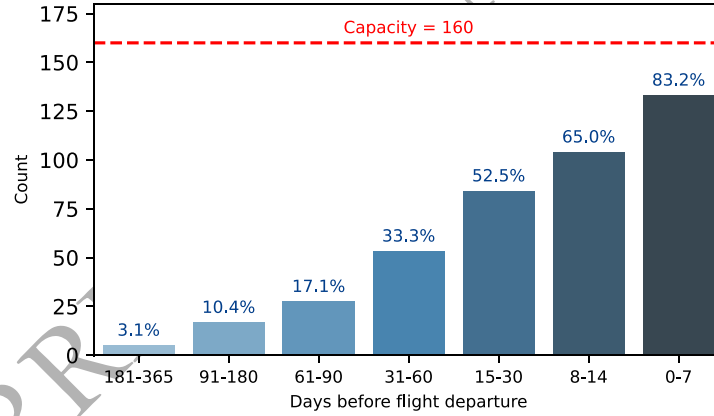


Fig. 8: Cumulative number of tickets sold before departure (Hub1 - New York)

dramatically decrease and the estimated coefficients of the predictors are unstable (Wold, Ruhe, Wold, & Dunn, 1984). This issue was detected in the baggage dataset because the number of leisure passengers was presented twice. The first estimate could be likely obtained from the booking data while the second estimate was obtained with an “improved” classification algorithm used by Airline λ . The two estimates were highly correlated with a Pearson’s correlation coefficient of 0.97. Therefore, one of the two variables was removed. Similarly, one variable related to business travelers was dropped.

3.4 Feature encoding

The baggage dataset consisted of many categorical variables, because quite a few time- and location-related features were involved, such as month, hour, day of the week, and airport code. If one-hot

encoding was directly employed for all those categorical features, the number of additional dummy variables would be significant, especially when a categorical variable has many levels. For instance, when we examine the origin airport as a categorical variable, if there are 100 airports, the same number of dummy variables are needed with one variable corresponding to one airport. To avoid introducing too many additional features, we combined some levels for selected categorical features as shown in Table 1, mainly based on the data explorations described earlier. Hubs 1, 2, and 3 represent the three busiest hubs in the dataset, measured by the number of flights. Similarly, the encoding for the “hour” feature into specific groups is informed by our analysis of operational patterns and baggage volume trends at airports. Specifically, we observed a consistent increase in baggage volumes during the hours of 0, 1, 2, and 23 as shown in Figure 7(c). To protect the confidential and proprietary information of Airline λ , specific names of those hubs are not revealed, while this anonymization does not hurt the rigor and usefulness of this analysis.

Appendix A lists all features considered in the following predictive analyses. Specifically, 22 features are flight-related; 27 are related to passengers; five are supplementary.

Table 1: Grouping for categorical variables

Categorical feature	Level grouping
Origin hub	[Hub1], [Hub2], [Hub3], [Others]
Destination hub	[Hub1], [Hub2], [Hub3], [Others]
Aircraft type	[330, 767, 777, 787], [Others]
Day of the week	[Wednesday, Thursday], [Monday, Sunday], [Others]
Hour	[0, 1, 2, 23], [Others]
Month	[9, 10, 11], [6, 7], [12], [Others]

4 Methodology

In this section, we describe three machine learning algorithms to predict the checked baggage volume y given attributes $\mathbf{x}$, namely all features listed in Appendix A. Section 4.1 will describe the basic modeling propositions. Sections 4.2 to 4.4 will describe three predictive algorithms.

4.1 Basic Modeling Propositions

In machine learning, an unknown prediction function $f(\mathbf{x})$ that relates $\mathbf{x}$ and y is identified, using its N noisy observations,

$$y_i = f(\mathbf{x}_i) + \epsilon_i, i = 1, \dots, N, \quad (1)$$

where y_i and $\mathbf{x}_i$ are the dependent and explanatory variable values for the i th observation, respectively. ϵ_i is independent noise with a mean of zero and variance of σ^2 . In Eq. (1), N is also the sample size. For $f(\mathbf{x})$, we consider multiple linear regression as the baseline benchmark, which will be described in Section 4.2. We next consider two gradient boosting methods in Sections 4.3 and 4.4. We seek to find the best model $\hat{f}(\mathbf{x})$ that minimizes the training error:

$$J(\hat{f}) = \sum_{i=1}^N \ell(\hat{f}(\mathbf{x}_i), y_i), \quad (2)$$

where $\ell(\hat{f}(\mathbf{x}_i), y_i)$ is a loss function that quantifies the cost of making the prediction $\hat{f}(\mathbf{x}_i)$, which can be abbreviated as $\hat{y}_i$, when y_i is the actual value.

As for the loss function $\ell(\cdot)$, we consider the business costs related to the prediction inaccuracy. In airline operations, the business cost of underestimating checked bags is significantly higher than the cost of overestimation. In case of underestimation, insufficient human power is allocated for loading checked bags, thus resulting in possible departure delays. Or there might not be enough storage room for checked bags on the departing airplane, which means customers may spend additional time waiting for their checked bags to arrive with another flight. Delayed bags may need to be shipped to customer homes at a high cost. In case of overestimation, the cargo compartment is partially filled, which incurs some opportunity cost, because some cargo shipments should have been loaded if an accurate count of checked bags has been estimated. To properly consider this cost asymmetry, we design the following piecewise quadratic loss function:

$$\ell(\hat{f}(\mathbf{x}_i), y_i) = \frac{c_o}{2} (\hat{y}_i - y_i)_+^2 + \frac{c_u}{2} (y_i - \hat{y}_i)_+^2 \quad (3)$$

In Eq. (3), the subscript “+” in $(\hat{y}_i - y_i)_+$ means only the positive part of $(\hat{y}_i - y_i)$ is taken. In case $\hat{y}_i < y_i$, 0 is used. c_o and c_u are the overestimation and underestimation cost coefficients, respectively. When c_o and c_u are assigned different values, overestimation and underestimation are penalized differently. The c_u to c_o ratio depends on how likely the space or weight allowance for checked baggage is exceeded for a flight. When the likelihood is relatively low, the coefficient for underestimation c_u should be relatively low while still being greater than c_o . When the checked baggage allowance is frequently exceeded, c_u should be significantly larger than c_o . As this likelihood varies over time and with flight, an airline should carefully determine the c_u to c_o ratio to be consistent with their own operations.

4.2 Simple benchmark: multiple linear regression

We first consider a multiple linear regression (MLR) model that assumes a linear relation between features $\mathbf{x}$ and the main response y . As MLR has been studied rigorously for many decades, it has been applied in many practical contexts of business and engineering. The major reason for considering this is to explore the simplest possibility that gives the smallest risk of model overfitting. Another benefit of MLR lies in its interpretability. The estimated linear coefficients are straightforward to interpret as the main effects of exploratory variables. An MLR model is formally described as

$$f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + b, \quad (4)$$

where $\mathbf{a}$ and b are the model parameters.

In this study, the asymmetric loss function $J(\hat{f})$ for MLR model can be expressed as Eq (5).

$$\text{Minimize }_{\mathbf{a}, b} J(\hat{f}) = \sum_{i=1}^N \frac{c_o}{2} (\mathbf{a}^T \mathbf{x}_i + b - y_i)_+^2 + \frac{c_u}{2} (y_i - \mathbf{a}^T \mathbf{x}_i - b)_+^2. \quad (5)$$

To minimize this cost function, gradient descent is used. The gradient descent algorithm is an iterative optimization algorithm that works by starting with an initial set of regression coefficients, $\mathbf{a}$

and b , and iteratively updating them in the direction of the steepest descent of the cost function, which is given by the negative gradient of the cost function:

$$\begin{aligned}\mathbf{a} &= \mathbf{a} - \eta \cdot \nabla_{\mathbf{a}} J(\hat{f}) \\ b &= b - \eta \cdot \nabla_b J(\hat{f}),\end{aligned}\tag{6}$$

where η is the learning rate, which controls the step size of the update. Specifically, we can derive the partial derivatives of the loss function w.r.t. to the parameters $\mathbf{a}$ and b as follows:

$$\nabla_{\mathbf{a}} J(\hat{f}) = \frac{\partial J(\hat{f})}{\partial \mathbf{a}} = \frac{1}{N} \sum_{i=1}^N \left\{ c_o (\mathbf{a}^T \mathbf{x}_i + b - y_i)_+ - c_u (y_i - \mathbf{a}^T \mathbf{x}_i - b)_+ \right\} \mathbf{x}_i, \tag{7}$$

$$\nabla_b J(\hat{f}) = \frac{\partial J(\hat{f})}{\partial b} = \frac{1}{N} \sum_{i=1}^N c_o (\mathbf{a}^T \mathbf{x}_i + b - y_i)_+ - c_u (y_i - \mathbf{a}^T \mathbf{x}_i - b)_+. \tag{8}$$

The procedure continues until the cost function converges to a minimum, i.e., until the norm of the gradient becomes sufficiently small or until a maximum number of iterations is reached. As a result, a set of regression coefficients that best match the data is provided by the gradient descent process.

4.3 Proposed approach with extreme gradient boosting

We also consider Extreme Gradient Boosting (XGBoost). It is a variant of gradient boosted decision trees (GBDT) (Chen & Guestrin, 2016). Like GBDT, its underlying model is an ensemble of decision trees (DT),

$$f(\mathbf{x}) = \sum_{m=1}^M f_m(\mathbf{x}),$$

where $f_m(\mathbf{x})$ represents the m -th DT model, and M is the number of DTs in the ensemble. The M DTs are learned sequentially, and the m -th DT model f_m is learned at the m -th stage of sequential learning. For the first stage, the first decision tree f_1 is fitted to the original training data $\{(\mathbf{x}_i, y_i), i = 1, \dots, N\}$. For the m -th stage, the m -th DT f_m is fitted to the residual from the $(m-1)$ -th stage. Let $\hat{y}_i^{(m-1)} = \sum_{\ell=1}^{m-1} f_{\ell}(\mathbf{x}_i)$ represent the fit obtained up to the $(m-1)$ -th stage. DT f_m is fitted to $\{(\mathbf{x}_i, y_i - \hat{y}_i^{(m-1)})\}$. Such sequential stacking of multiple DTs can be easily overfitted. To avoid overfitting, XGBoost applies a model regularization. Specifically, the m -th iteration of XGBoost fits f_m to the residual $y_i - \hat{y}_i^{(m-1)}$ by minimizing the following objective function,

$$L^{(m)} = \sum_{i=1}^N \ell(y_i - \hat{y}_i^{(m-1)}, f_m(\mathbf{x}_i)) + \Omega(f_m), \tag{9}$$

where ℓ is a differentiable convex loss function, which is typically a negative log-likelihood for classification problems or the mean squared error for regression problems. The regularization term $\Omega(f_m)$ that penalizes the complexity of the m -th tree is defined as

$$\Omega(f_m) = \gamma J_m + \frac{1}{2} \lambda \sum_{j=1}^{J_m} w_j^2, \tag{10}$$

where J_m is the number of leaves in DT f_m , w_j is the complexity score of the j -th leave, and constants γ and λ determine weights on the two terms in the regularization term. For computational feasibility, [Chen and Guestrin \(2016\)](#) developed a second-order approximation to the objective function Eq. (9) as

$$L^{(m)} \simeq \sum_{i=1}^N \left[\ell(y_i, \hat{y}^{(m-1)}) + g_i f_m(\mathbf{x}_i) + \frac{1}{2} h_i f_m^2(\mathbf{x}_i) \right] + \Omega(f_m), \quad (11)$$

where $g_i = \partial_{\hat{y}^{(m-1)}} l(y_i, \hat{y}^{(m-1)})$ and $h_i = \partial_{\hat{y}^{(m-1)}}^2 l(y_i, \hat{y}^{(m-1)})$ are first and second order gradients of the loss function $l(y_i, \hat{y}^{(m-1)})$. We derive the derivatives for the asymmetric loss function:

$$g_i = \sum_{i=1}^N c_o (\hat{y}_i - y_i)_+ - c_u (y_i - \hat{y}_i)_+ \quad (12)$$

$$h_i = \sum_{i=1}^N c_o (\hat{y}_i - y_i)_+^0 - c_u (y_i - \hat{y}_i)_+^0 \quad (13)$$

After removing the constant terms in Eq. (11) and expanding $\Omega(f_m)$, the objective function is written as

$$\begin{aligned} \tilde{L}^{(m)} &= \sum_{i=1}^N \left[g_i f_m(\mathbf{x}_i) + \frac{1}{2} h_i f_m^2(\mathbf{x}_i) \right] + \gamma J_m + \frac{1}{2} \lambda \sum_{j=1}^{J_m} w_j^2 \\ &= \sum_{j=1}^{J_m} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma J_m, \end{aligned} \quad (14)$$

where $I_j = \{i \mid q(\mathbf{x}_i) = j\}$ is the data samples in leaf j . After creating the second-order approximation of the objective function, XGBoost proceeds to build the model by adding trees to the ensemble. At each iteration, the algorithm calculates the leaf weights that minimize the objective function for the newly added tree. This process of adding trees continues until a stopping criterion is satisfied, which is often determined by the validation error. This criterion prevents overfitting of the model and ensures that the algorithm can generalize well to unseen data.

4.4 Proposed approach with light gradient boosting machine

Another approach we consider is the Light Gradient Boosting Machine (LightGBM), an efficient and scalable implementation of gradient boosting framework ([Ke et al., 2017](#)). LightGBM is known for its high efficiency, low memory usage, and its ability to handle large-scale data. The core algorithm is similar to that of XGBoost, but with some distinct features that enhance performance, particularly on large datasets.

LightGBM builds the model in the form of an ensemble of decision trees, similar to XGBoost:

$$f(\mathbf{x}) = \sum_{m=1}^M f_m(\mathbf{x}), \quad (15)$$

where $f_m(\mathbf{x})$ represents the m -th decision tree model, and M is the total number of trees.

Two unique features of LightGBM are its Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which reduce the amount of data and number of features without significant

loss of accuracy. The GOSS method focuses on instances with larger gradients, as they are considered more informative, while EFB effectively reduces the number of features by bundling mutually exclusive features.

The training process involves minimizing a similar objective function as XGBoost, with a loss function and a regularization term. However, LightGBM uses histogram-based algorithms for computing the gradients, which significantly speeds up the learning process. The objective function is defined as:

$$L = \sum_{i=1}^N \ell(y_i, \hat{y}_i) + \Omega(f), \quad (16)$$

where $\ell(y_i, \hat{y}_i)$ is the loss function, $\hat{y}_i$ is the predicted value, and $\Omega(f)$ is the regularization term.

The loss function can be tailored to our specific problem, considering the asymmetric cost of underestimation and overestimation as previously described. The regularization term helps to control the complexity of the model and prevent overfitting. The algorithm iteratively builds trees, each one focusing on correcting the errors of the previous ensemble.

To optimize the LightGBM model, we employ a similar gradient descent strategy as described for the MLR model, with the objective of minimizing the asymmetric loss function. The LightGBM framework offers several hyperparameters such as the number of leaves, the learning rate, and the maximum depth of trees, which we can tune to achieve the best model performance.

In summary, LightGBM offers an efficient and effective approach for predictive modeling in large-scale data scenarios. Its ability to handle large datasets with a high speed and lower memory consumption, while providing high accuracy, makes it a suitable choice for our predictive analysis of checked baggage volume.

5 Results

5.1 Comparison of MLR and LightGBM under symmetric loss

We first compare MLR and LightGBM without considering an asymmetric loss function. In other words, we assume $c_u = c_o = 1$ in Section 5.1. A standard hold-out testing method is adopted where 75% of the samples (541,917 samples) are in the training dataset and the remaining 180,639 samples are in the testing dataset. Figure 9 compares the residual distributions for MLR and LightGBM. In Figure 9, the mean residuals for LightGBM and MLR are virtually the same and close to 0. The standard deviation of MLR is 17.6, which is greater than 14.9 for LightGBM. This is consistent with the finding that the R^2 value achieved with MLR (0.79) is clearly lower than that of LightGBM (0.85). The scatter plot in Figure 9 shows that the residuals for MLR are more dispersed than LightGBM, a further sign of weaker prediction performance. While a considerable number of residuals are outside of the range $[-30, 30]$ for MLR, significantly more LightGBM residuals are within the range. For MLR, when the true bag count is below 100, residuals tend to be negative (i.e., underprediction); when the true bag count is over 100, residuals tend to be positive (i.e., overprediction). For LightGBM, the correlation of residuals and true values is lower. Note that the residuals from only 1% of all the testing samples are shown in Figure 9 to avoid overcrowding.

With 0.5% of testing samples drawn, the Quantile-Quantile plot in Figures 10 indicates that the residuals of LightGBM and MLR largely follow a normal distribution, as most of the points fall approximately on the 45-degree reference line. This further indicates that MLR captures the underlying relation between various features and the target variable quite well.

The above comparisons suggest that the advanced LightGBM outperforms the standard linear regression model. LightGBM has multiple advantages over MLR. First, LightGBM, being a tree-based method, can naturally model these non-linear relationships, making it more effective for complex datasets. Second, LightGBM automatically captures interactions between features through its hierarchical tree structure. Lastly, LightGBM is more robust to outliers as decision trees split the data based on certain conditions, reducing the influence of extreme values.

We further note that for both methods, if the true value is smaller than 50 or larger than 200, it is quite challenging to predict the true value accurately, as shown in Figure 9. The primary reason is that there are no adequate training data in those bag count ranges to understand the relation between the bag count and other features as stated earlier. For instance, wide-body aircraft, which can accommodate more than 200 passengers, are severely underrepresented (accounting for only 1.85% of samples), as shown in Figure 4.

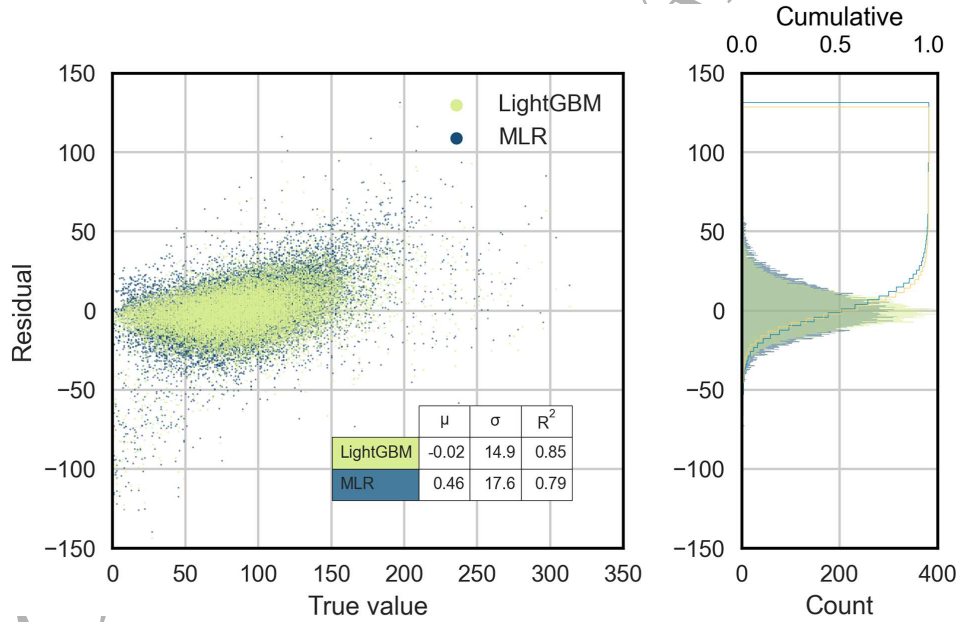


Fig. 9: Histograms for residuals under symmetric loss

5.2 Comparison of three methods under asymmetric loss

We next compare the three methods when an asymmetric loss function is considered. We use the same hold-out testing method described in Section 5.1. To model the degree of asymmetry, we set c_o to be 1 while considering four values for c_u , namely 1, 5, 10, and 15. When $c_u = c_o$, the asymmetric loss function reduces to a symmetric one. As the value of c_u increases, the penalty for underestimation grows. The XGBoost hyperparameters are tuned by Bayesian Optimization (Snoek, Larochelle, & Adams,

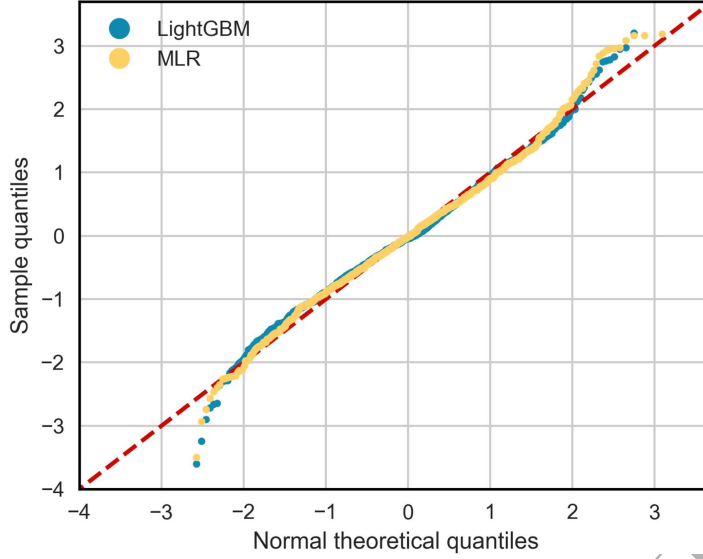


Fig. 10: Quantiles of residuals of MLR and LightGBM

2012). The hyperparameter tuning process has 100 iterations. A hold-out strategy is used aiming to minimize the sum of asymmetric loss function values over testing samples. After hyperparameter tuning, values of some key hyperparameters of XGBoost are: learning rate = 0.1, maximum tree depth = 6, minimum sum of instance weight (hessian) needed in a child = 19, and subsample ratio = 0.8. While using more estimators leads to better performances, the number of estimators is configured as 600 because no significant improvements are observed by increasing this value. As the XGBoost API supports multi-threading, the number of threads is configured as the maximum value, namely 16. Similar to XGBoost, the hyperparameters for LightGBM are also tuned using Bayesian Optimization. Key LightGBM hyperparameters include a learning rate of 0.1, a maximum depth of 6, and 31 leaves per tree, which is a default yet effective choice for LightGBM. The number of estimators is kept the same as XGBoost at 600. The subsample ratio is set at 0.8, aligning with the XGBoost setup. The LightGBM model is configured to use all available cores to ensure efficient computation.

Figures 11, 12, and 13 compare the residual distributions of MLR, LightGBM, and XGBoost, respectively. Clearly, the distributions of XGBoost and LightGBM are tighter than MLR. The standard deviations of the residuals of XGBoost and LightGBM are around 15 while MLR yields a larger standard deviation of 17.6 when $c_u = 1$. This means XGBoost and LightGBM outperform MLR when a symmetric loss function is considered. As c_u increases from 1 to 10, the distribution of residuals shifts to the left and the standard deviation of residuals increases, for all three methods. As c_u increases further to 15, the change in the mean and standard deviation of residuals becomes relatively small. This means the impact of a growing c_u on the residual distribution diminishes. While c_u increases from 1 to 15, XGBoost and LightGBM have a consistently better predictive performance than MLR judged by the distribution of residuals. When c_u increases from 1 to 15, the R^2 of MLR decreases from 0.79 to 0.56; LightGBM is capable of achieving a high R^2 consistently, although it decreases from 0.85 to 0.71; XGBoost performs similar to LightGBM, with R^2 of 0.85 to 0.72.

We briefly show why as coefficient c_u in the asymmetric loss function increases, R^2 decreases. The coefficient of determination, i.e., R^2 , is defined as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (17)$$

R^2 can be rewritten as:

$$R^2 = 1 - \frac{\text{MSE}}{\text{Var}(Y)} \quad (18)$$

by noting the definitions of mean squared error (MSE) and variance $\text{MSE} = \sum_i (y_i - \hat{y}_i)^2$ and $\text{Var}(Y) = \sum_i (y_i - \bar{y})^2$.

Since the variance of the target variable is a constant, it is quite clear that R^2 is a standardized or rescaled MSE. In particular, a larger MSE leads to a smaller R^2 . As an asymmetric loss function is adopted in training the learning algorithm, the asymmetric loss is the optimization objective, rather than MSE (a symmetric metric). In the special case where $c_u = 1$, asymmetric loss reduces to symmetric loss, which means minimizing an asymmetric loss gives the minimal MSE, thus the highest R^2 . As c_u increases, underestimation is more heavily penalized than overestimation, which means the predictive algorithm will overestimate, thus directly hurting MSE (becoming larger) and R^2 (becoming smaller). Figures 11, 12, and 13 also show how R^2 varies with c_u for the three methods.

Figure 14 compares the loss function values of three methods, which shows that MLR consistently has the highest loss for all values of c_u , and XGBoost performs with the lowest loss, indicating that XGBoost outperforms LightGBM and MLR. Furthermore, as c_u increases from 1 to 15, the loss values for all methods increase, with MLR exhibiting a percentage increase in loss of about 45% compared to XGBoost. Overall, XGBoost outperforms LightGBM very slightly and MLR by a notable margin when an asymmetric loss function is considered, regardless of the value of c_u .

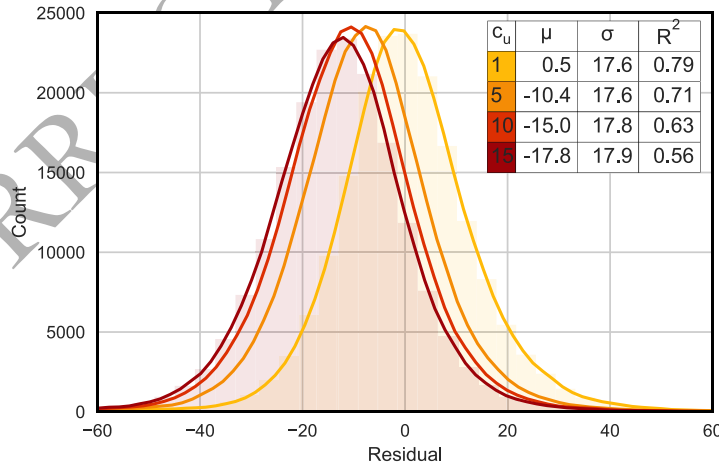


Fig. 11: Effect of c_u on MLR residual distribution

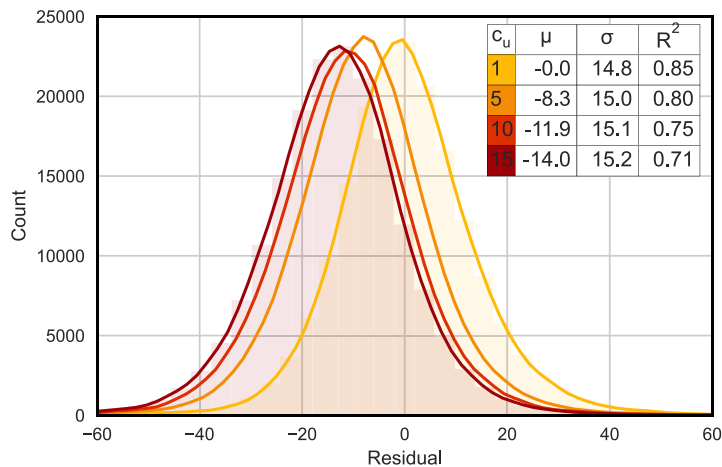


Fig. 12: Effect of c_u on LightGBM residual distribution

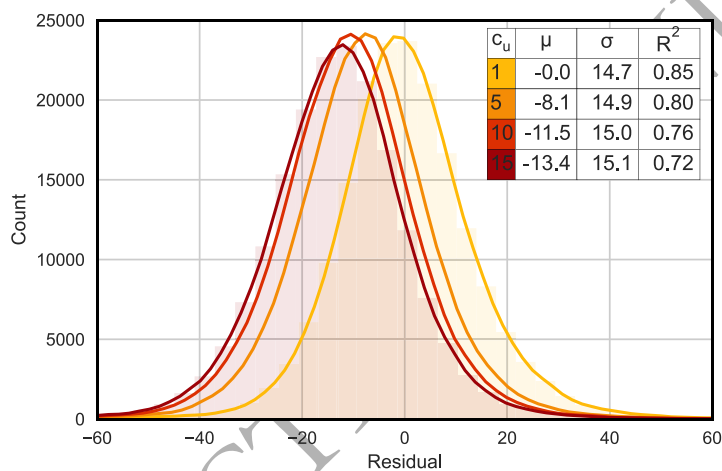


Fig. 13: Effect of c_u on XGBoost residual distribution

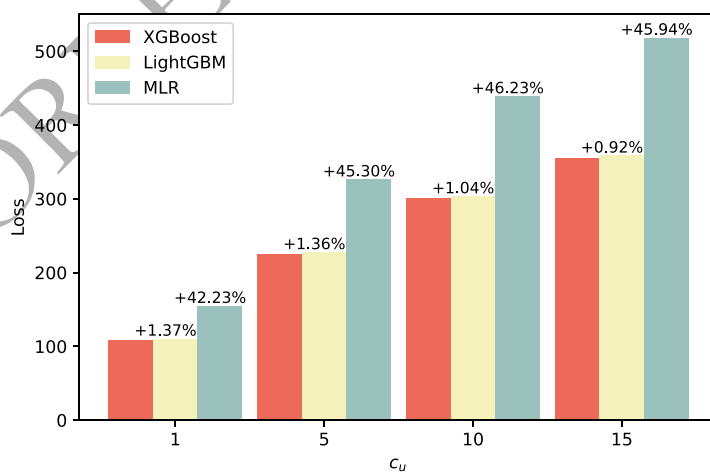


Fig. 14: Comparison of loss function values under different c_u

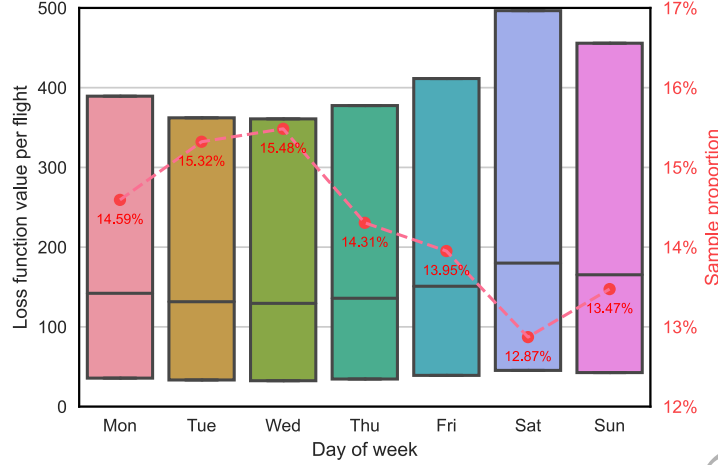


Fig. 15: Loss function value per flight of XGBoost by day of the week

5.3 Exploration on residuals of XGBoost

Here we explore the predictive performance of XGBoost across different categories when $c_o = 1$ and $c_u = 5$. For instance, we seek to explore whether XGBoost would achieve similar performance for flights on different days of the week. Figure 15 thus shows the 25th percentile, median, and 75th percentile of the loss function value. No significant differences are observed over different days of the week except for Saturday with a higher loss function value. This means the performance of XGBoost is overall consistent over the days.

We next define a criterion for misprediction. We state a value is mispredicted, thus constituting a *misprediction*, if the absolute residual is larger than 25 (i.e., $|y - \hat{y}| > 25$) and relative deviation is larger than 20% (i.e., $|y - \hat{y}| / y > 20\%$). Then for each destination city, we can compute the proportion for mispredicted samples. Figure 16 shows the top destination cities with the highest misprediction proportions. The two notable destinations are Anchorage, Alaska and Hayden, Colorado. Figures 18 show that the residual distribution for Atlanta, Georgia is clearly tighter than those for the two identified airports with significant mispredictions.

One possible reason for baggage mispredictions for Anchorage-bound flights is that 75 out of 268 flights (28.0%) arriving in Anchorage are operated by wide-body aircraft. The number of bags per flight for Anchorage is 178 on average while the average bag count for the whole dataset is 92. The latter city Hayden, Colorado is one of the smallest cities getting air services in the U.S. because its population was only around two thousand in 2020. The whole dataset has only 126 samples for Hayden covering a four-month winter period from December to March. In other words, those are seasonable flights. It is very likely that most of those passengers are taking ski vacations in Hayden during winter months. It is thus understandable that those passenger characteristics are quite different. For instance, on average 108 passengers are classified as leisure travelers and 8 passengers are classified as business travelers by Airline λ , i.e., the percentage for business travelers is 0.7%. In contrast, a much higher proportion of passengers, i.e., 33%, travel for business instead of leisure, as classified by Airline λ . In addition, a higher percentage of passengers travel with a child or infant on Hayden-bound flights than the same percentage for the entire dataset. Clearly, the underlying patterns for the identified cities in

Figure 16 are different from most other flights in the dataset, which mean those associated flights can be considered “outliers.” Therefore, substantial mispredictions occur.

Figure 17 shows the misprediction proportion for each fleet type. Four leading fleet types with a large misprediction proportion are all wide-body aircraft, which suffer from the same underrepresentation issue. Figure 19 also shows that the residual distribution for Airbus A319 is much tighter than the distributions for Boeing 777 and 787.

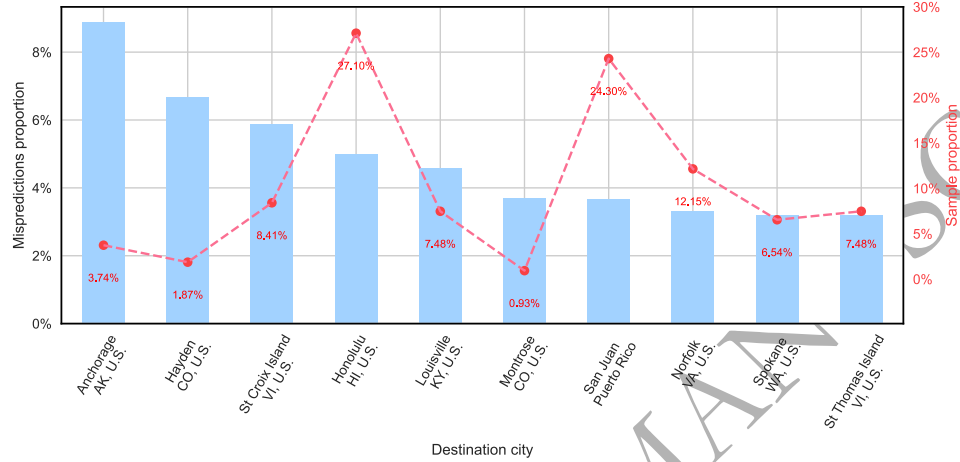


Fig. 16: Distribution of top 10 destination cities with significant mispredictions

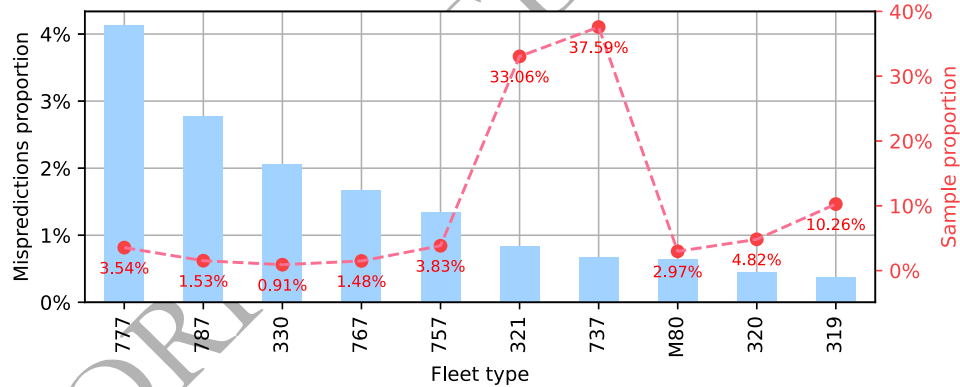


Fig. 17: Distribution of fleet types with significant mispredictions

Even though XGBoost achieves a very high prediction accuracy overall, its performance may not be satisfactory for certain flights with insufficient data support. For instance, mispredictions are significant for those flights associated with the identified destination cities and fleet types. Additional efforts need to be made to improve the predictive performance of those outlier flights. For instance, more historical data covering multiple years rather than one year is necessary to achieve an acceptable prediction performance.

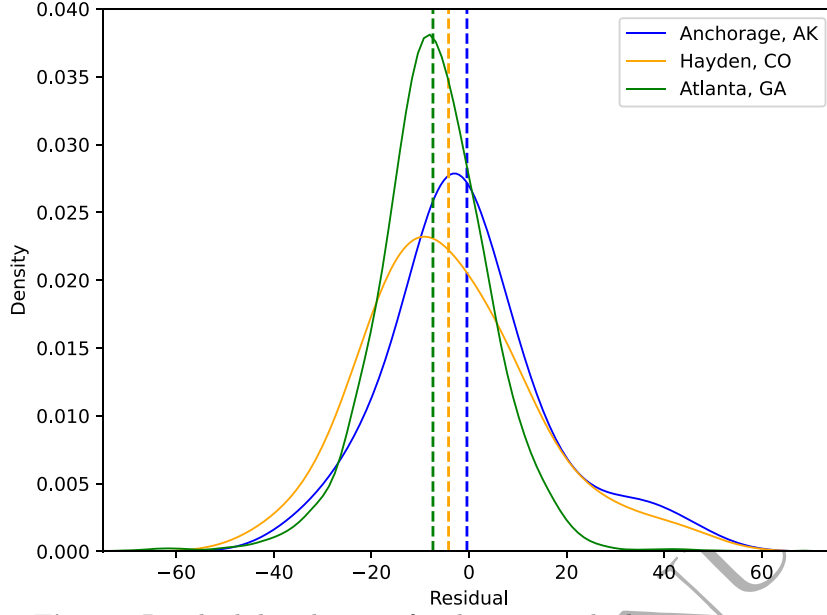


Fig. 18: Residual distributions for three example destination cities

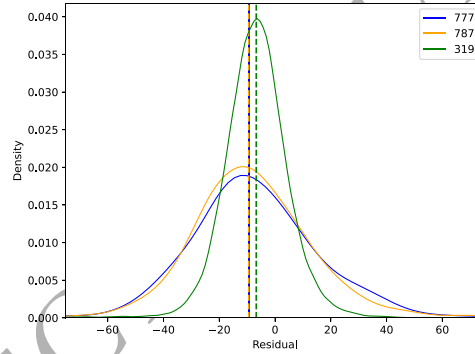


Fig. 19: Residual distributions for three example fleet types

6 Discussions

Selecting an appropriate prediction algorithm is never an easy task as a trade-off between a learning algorithm's complexity and its performance needs to be made. For checked baggage prediction, the historical average approach is used in practice primarily due to its intuitiveness and simplicity. However, even a very typical linear regression algorithm can outperform the state-of-the-practice approach by a good margin; an advanced machine learning algorithm, such as LightGBM or XGBoost, is more effective in achieving high prediction accuracy, which is expected. Despite the superior performance of LightGBM and XGBoost, especially measured by accuracy, both gradient boosting methods, as an ensemble model, are not interpretable (Luo et al., 2022; Shi, Jin, Lu, & Shi, 2023). It is less transparent than the linear regression or historical average approach for analysts to understand how a value is predicted. From a pragmatic perspective, low interpretability means a lower chance of acceptance by analysts. In addition, the complexity of XGBoost is high, as it requires more hyperparameter tuning than multiple linear regression. Figure 20 clearly indicates that the MLR training time is almost

negligible in relative to the training time for XGBoost even though 16 threads are used in parallel to train XGBoost. LightGBM, while it is also a gradient boosting method, takes 11.4 and 28.2 seconds for training with 100,000 and 700,000 samples, respectively, which are dramatically shorter than that of XGBoost.

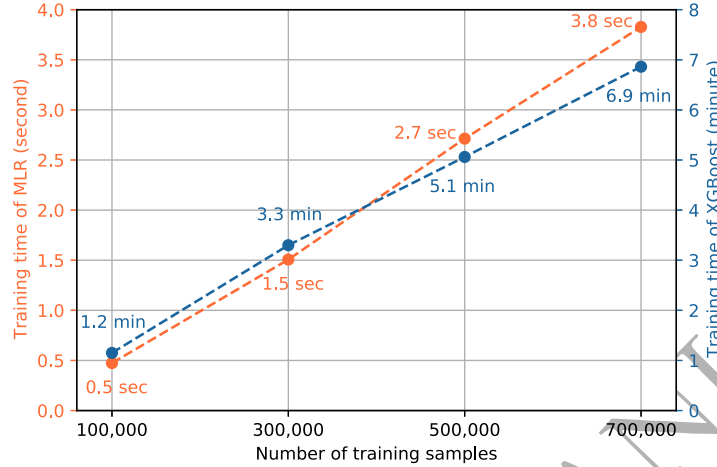


Fig. 20: Training time as a function of training sample size

For this specific prediction task, the relation between the target variable (i.e., the checked bag count) and a few key features (e.g., the revenue passenger count) is found to be largely linear, although other factors (e.g., destination city characteristics and travel seasons) also play a role. The clear linear relation explains why a linear regression algorithm can achieve quite satisfactory performance. Linear regression results are highly interpretable. For instance, the coefficient for the number of main cabin passengers measures how many bags a main cabin passenger carries. Although this coefficient varies over time and across regions, values of this coefficient can be tracked by airlines to understand the trends in checked baggage. Linear regression is thus recommended for replacing the historical average approach. XGBoost or other complex learning algorithms are recommended for those cases where the linear relation is unclear or not well characterized by available data.

A more detailed baggage prediction model can be developed at the individual passenger level, as the current analysis is conducted at the flight level. In the individual prediction model, historical baggage data for an individual can be leveraged to understand how likely she/he may check bags. Then, individual bag counts can be aggregated to obtain the flight-level bag count.

7 Conclusions

With a very large-scale multi-source dataset covering the entire U.S., we customized three machine learning algorithms of various complexities to predict the number of checked bags on a flight and evaluated the performance improvements of those three machine learning algorithms. Our primary contribution is to systematically consider the different impacts of overprediction and underprediction on airline operations and customize three machine learning algorithms with a piecewise quadratic loss function. Our numerical experiments indicate that (1) the two gradient boosting methods can significantly outperform standard multiple linear regression; (2) XGBoost can further improve the

predictive performance of LightGBM, regardless of whether the asymmetric loss function is considered or not; (3) LightGBM is more efficient than XGBoost in terms of training process. We also find that even though XGBoost has remarkably high accuracy in predicting the bag count at the flight level, the performance is not uniform, as checked bags in certain categories cannot be predicted satisfactorily. For instance, the misprediction portion tends to be high for flights to Anchorage, Alaska, or flights operated by wide-body aircraft. One major shortcoming of XGBoost is its low interpretability. Therefore, it is noted that an ideal prediction algorithm for adoption in practice should strike a reasonable balance between accuracy and interpretability.

This present study can be improved in the following ways. First, the entire dataset can be partitioned so that certain categories of samples (e.g., those involving wide-body aircraft) can be predicted by dedicated machine learning algorithms with high complexity while the rest of the samples can be predicted with a low-complexity algorithm. Second, the checked baggage weight can be further predicted when such data are available. Third, additional passenger-related features, such as gender, age, and travel duration (time difference between departure and arrival times), may be added to improve the predictive performance. Additionally, efforts are needed to investigate how such a prediction tool can be incorporated into airline revenue management. Finally, in this study, the baggage prediction is conducted only once, several hours prior to departure. The predicted baggage volume, when available, initializes the aircraft load planning process. As the availability of information varies over time, it is interesting and relevant to update prediction results when new information (e.g., trip cancellations or itinerary changes) becomes available as a flights departure nears. The current static prediction can be extended to a dynamic prediction in a future study.

Acknowledgments

The corresponding author is partially supported by the National Science Foundation (Grant Nos. 2055347, 2100745, and 2332161). The data support from Airline λ is appreciated.

Conflict of Interest statement

The corresponding author is an Editorial Board Member for Intelligent Transportation Infrastructure and is blinded from reviewing or making decisions for the manuscript. The authors do not have any competing financial interests to declare.

Authors' Contributions

Shijie Chen: Methodology, Investigation, Software, Writing - original draft, Writing review & editing. Chiwoo Park: Methodology, Validation, Writing - original draft. Qianwen Guo: Investigation, Validation, Writing - original draft. Yanshuo Sun: Conceptualization, Supervision, Validation, Writing - original draft, Writing review & editing. All authors reviewed the manuscript.

Appendix A List of considered features

Table A1 lists the name, description, and variable type of each feature that is considered in the case study.

Table A1: Descriptions and variable types of features

Feature name	Description [# features if categorical]	Variable type
Flight-related		
FLIGHT_DEP_M	Scheduled departure date (month only) [12]	Categorical
FLIGHT_ARVL_M	Scheduled arrival date (month only) [12]	Categorical
FLIGHT_DEP_DW	Scheduled departure date (day of the week) [7]	Categorical
FLIGHT_ARVL_DW	Scheduled arrival date (day of the week) [7]	Categorical
FLIGHT_DEP_HOUR	Scheduled departure time (hour only) [24]	Categorical
FLIGHT_ARVL_HOUR	Scheduled arrival time (hour only) [24]	Categorical
LEG_DISTANCE	Number of miles of this leg	Continuous
ORIG_CITY_NM	Origin city name [112]	Categorical
DEST_CITY_NM	Destination city name [116]	Categorical
AIRCRAFT_TYPE	Type of aircraft (wide or narrow body) [2]	Categorical
FLEET_TYPE	Fleet classification code [11]	Categorical
TTL_SEATS	Total number of seats	Continuous
TTL_PAX	Total number of passengers	Continuous
LF	Load factor = number of passengers/number of seats	Continuous
Passenger-related		
PAX_REV	Number of revenue passengers on the flight	Continuous
PAX_NONREV	Number of non-revenue leisure passengers	Continuous
PAX_POS_SPC	Number of non-revenue business passengers	Continuous
PAX_CONNECT	Number of passengers on the flight who have a connection	Continuous
PAX_LOCAL	Number of passengers who start their flight at the departure airport	Continuous
PAX_BUSINES	Number of passengers classified as business passengers	Continuous
PAX_LEISURE	Number of passengers classified as leisure passengers	Continuous
PAX_CHILD_WINF	Number of passengers with a child or infant	Continuous
PAX_ASSIST	Number of passengers who require assistance	Continuous
PAX_ANIMAL	Count of all the service/emotional support animals on the flight	Continuous
LYLTY_X	Number of passengers with a loyalty status of X	Continuous
NUM_GROUP_Y	Number of groups that contain Y passengers	Continuous
CABIN_FIRST	Number of passengers in first class	Continuous
CABIN_BUSINESS	Number of passengers in business class	Continuous
CABIN_PREM	Number of passengers in premium economy	Continuous
CABIN_MAIN	Number of passengers in main cabin	Continuous
BOOK_DAYS2DEP_Z	Number of passengers booked within Z days prior to departure	Continuous
BOTH_CARD	Number of passengers who hold credit cards A and B	Continuous
CARD1	Number of passengers who hold credit card A only	Continuous
CARD2	Number of passengers who hold credit card B only	Continuous
SALES_BUSINESS	Number of passengers booking via a business sales channel	Continuous
SALES_DIRECT	Number of passengers booking via official websites.	Continuous
SALES_LEISURE	Number of passengers booking via a leisure sales channel	Continuous
SALES_OTA	Number of passengers booking via an Online Travel Agency	Continuous
SALES_TMC	Number of passengers booking via a Travel Management Company	Continuous
Supplemental		
CK_POINT	TSA checkpoint travel numbers	Continuous
ENP_ORIG	Enplanement of origin airport	Continuous
ENP_DEST	Enplanement of destination airport	Continuous
POP_ORIG	Population of origin city	Continuous
POP_DEST	Population of destination city	Continuous

References

- Barua, L., Zou, B., Zhou, Y. (2020). Machine learning for international freight transportation management: a comprehensive review. *Research in Transportation Business & Management*, 34, 100453, <https://doi.org/https://doi.org/10.1016/j.rtbm.2020.100453>
- BEONTRA (2022). *Machine learning helps improve baggage handling planning*. <https://www.beontra.com/machine-learning-helps-improve-baggage-handling-planning/>. ([Online; accessed July 23, 2023])

- Brandt, F., & Nickel, S. (2019). The air cargo load planning problem-a consolidated problem definition and literature review on related problems. *European Journal of Operational Research*, 275(2), 399–410, <https://doi.org/https://doi.org/10.1016/j.ejor.2018.07.013>
- Chen, T., & Guestrin, C. (2016). XGboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 785–794).
- Cheng, S., Gao, Q., Zhang, Y. (2014). Comparative study on forecasting method of departure flight baggage demand. *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference* (pp. 1600–1605).
- Chung, S.-H., Ma, H.-L., Hansen, M., Choi, T.-M. (2020). Data science and analytics in aviation. *Transportation Research Part E: Logistics and Transportation Review*, 134, 101837, <https://doi.org/https://doi.org/10.1016/j.tre.2020.101837>
- FAA (2016a). *Pilots Handbook of Aeronautical Knowledge*. https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/phak/media/12_phak_ch10.pdf. ([Online; accessed July 23, 2023])
- FAA (2016b). *Weight & Balance Handbook*. https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/media/faa-h-8083-1.pdf. ([Online; accessed July 23, 2023])
- FAA (2022). *Passenger Boarding (Enplanement) and All-Cargo Data for U.S. Airports*. <https://www.tsa.gov/coronavirus/passenger-throughput>. ([Online; accessed July 23, 2023])
- Gallego, C.E.V., Comendador, V.F.G., Carmona, M.A.A., Valdés, R.M.A., Nieto, F.J.S., Martínez, M.G. (2019). A machine learning approach to air traffic interdependency modelling and its application to trajectory prediction. *Transportation Research Part C: Emerging Technologies*, 107, 356–386, <https://doi.org/https://doi.org/10.1016/j.trc.2019.08.015>
- Hanumantha, G.J., Arici, B.T., Sefair, J.A., Askin, R. (2020). Demand prediction and dynamic workforce allocation to improve airport screening operations. *IIEE Transactions*, 52(12), 1324–1342, <https://doi.org/https://doi.org/10.1080/24725854.2020.1749765>
- Jardines, A., Soler, M., García-Heras, J. (2021). Estimating entry counts and ATFM regulations during adverse weather conditions using machine learning. *Journal of Air Transport Management*, 95, 102109, <https://doi.org/https://doi.org/10.1016/j.jairtraman.2021.102109>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, ,

- Khan, W.A., Ma, H.-L., Chung, S.-H., Wen, X. (2021). Hierarchical integrated machine learning model for predicting flight departure delays and duration in series. *Transportation Research Part C: Emerging Technologies*, 129, 103225, <https://doi.org/https://doi.org/10.1016/j.trc.2021.103225>
- Luo, Z., Zhan, Y., Liu, Y., Zhang, A., Lin, X., Zhang, Y. (2022). Research on influencing factors of asphalt pavement International Roughness Index (IRI) based on ensemble learning. *Intelligent Transportation Infrastructure*, 1, liac014, <https://doi.org/https://doi.org/10.1093/iti/liac014>
- Ma, Q., Bi, J., Sai, Q., Li, Z. (2021). Research on prediction of checked-baggage departed from airport terminal based on time series analysis. *2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC)* (pp. 264–269).
- Mikram, M., Rhanoui, M., Yousf, S., Briwa, H. (2020). Unit load devices (ULD) demand forecasting in the air cargo for optimal cost management. *Journal of Automation Mobile Robotics and Intelligent Systems*, 14, , <https://doi.org/https://doi.org/10.14313/JAMRIS/32020/37>
- Mondoloni, S., & Rozen, N. (2020). Aircraft trajectory prediction and synchronization for air traffic management applications. *Progress in Aerospace Sciences*, 119, 100640, <https://doi.org/https://doi.org/10.1016/j.paerosci.2020.100640>
- NASA (2007). *NASA & the next generation air transportation system (NEXTGEN)*. https://www.nasa.gov/sites/default/files/atoms/files/nextgen.whitepaper_06_26_07.pdf. ([Online; accessed July 23, 2023])
- Rodríguez-Sanz, Á., Comendador, F.G., Valdés, R.A., Pérez-Castán, J., Montes, R.B., Serrano, S.C. (2019). Assessment of airport arrival congestion and delay: Prediction and reliability. *Transportation Research Part C: Emerging Technologies*, 98, 255–283, <https://doi.org/https://doi.org/10.1016/j.trc.2018.11.015>
- Shi, C., Jin, Y., Lu, H., Shi, J. (2023). A BIM-based framework for automatic numerical modelling and geotechnical analysis of a large-scale deep excavation for transportation infrastructures. *Intelligent Transportation Infrastructure*, 2, liad012, <https://doi.org/https://doi.org/10.1093/iti/liad012>
- Snoek, J., Larochelle, H., Adams, R.P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2951–2959,
- Sun, Y., Jiang, Z., Gu, J., Zhou, M., Li, Y., Zhang, L. (2018). Analyzing high speed rail passengers train choices based on new online booking data in China. *Transportation Research Part C: Emerging*

Technologies, 97, 96–113, <https://doi.org/https://doi.org/10.1016/j.trc.2018.10.015>

- TSA (2022). *TSA checkpoint travel numbers*. <https://www.tsa.gov/coronavirus/passenger-throughput>. ([Online; accessed July 23, 2023])
- Tseremoglou, I., Bombelli, A., Santos, B.F. (2022). A combined forecasting and packing model for air cargo loading: A risk-averse framework. *Transportation Research Part E: Logistics and Transportation Review*, 158, 102579, <https://doi.org/https://doi.org/10.1016/j.tre.2021.102579>
- U.S. Census Bureau (2021). *City and town population totals: 2010-2020*. <https://www.census.gov/programs-surveys/popest/technical-documentation/research/evaluation-estimates/2020-evaluation-estimates/2010s-cities-and-towns-total.html>. ([Online; accessed July 23, 2023])
- van Leeuwen, H., Zhang, Y., Zervanou, K., Mullick, S., Kaymak, U., de Ruijter, T. (2020). Lost and Found: Predicting Airline Baggage At-risk of Being Mishandled. *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART* (pp. 172–181).
- Wang, Z., Liang, M., Delahaye, D. (2020). Automated data-driven prediction on aircraft Estimated Time of Arrival. *Journal of Air Transport Management*, 88, 101840, <https://doi.org/https://doi.org/10.1016/j.jairtraman.2020.101840>
- Wold, S., Ruhe, A., Wold, H., Dunn, W., Iii. (1984). The collinearity problem in linear regression. the partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3), 735–743, <https://doi.org/https://doi.org/10.1137/090505>
- Wu, X., Yang, H., Chen, H., Hu, Q., Hu, H. (2022). Long-term 4D trajectory prediction using generative adversarial networks. *Transportation Research Part C: Emerging Technologies*, 136, 103554, <https://doi.org/https://doi.org/10.1016/j.trc.2022.103554>
- Xu, S., Chan, H.K., Zhang, T. (2019). Forecasting the demand of the aviation industry using hybrid time series SARIMA-SVR approach. *Transportation Research Part E: Logistics and Transportation Review*, 122, 169–180, <https://doi.org/https://doi.org/10.1016/j.tre.2018.12.005>
- Xu, Y., Wandelt, S., Sun, X. (2024). Airline Scheduling Optimization: Literature Review and a Discussion of Modeling Methodologies. *Intelligent Transportation Infrastructure*, 3, liad026, <https://doi.org/https://doi.org/10.1093/iti/liad026>