Transportation Research Part E 119 (2018) 149–164

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

Can machines learn how to forecast taxi-out time? A comparison of predictive models applied to the case of Seattle/Tacoma International Airport

Tony Diana

Federal Aviation Administration, United States

ARTICLE INFO

Keywords:
Machine learning
Supervised models
Predictive analytics
Penalized regression
Taxi-out operations

ABSTRACT

This study compares the performance of ensemble machine learning, ordinary least-squared and penalized algorithms to predict taxi-out time at two different periods of NextGen capability implementation. In the pre-sample, ordinary least-squared and ridge models performed better than other ensemble learning models. However, the gradient boosting model provided the lowest root mean squared errors in the post-sample. No algorithm fits data better in all cases. This paper recommends selecting the model that provides the best balance between bias and variance.

1. Introduction

A few years ago, taxi-out delays caught the attention of the press and Congress when passengers were stranded for seven hours on the tarmac.[1] Passenger complaints led the U.S. Department of Transportation (USDOT) to issue some rules to prevent less than three-hour tarmac times and fine airlines that would exceed that threshold.[2]

Taxi-out operations start when a plane leaves the gate and end when it takes off. They play a significant role in ensuring on-time departures and preventing delay propagation throughout the National Airspace System (NAS). An increase in taxi-out times often indicates when an airport's surface movement area is congested and when constrained departure capacity will affect maximum departure throughputs.

Predicting taxi-out times is of great interest to airport, airline, and regulatory analysts to ensure on-time departure performance. First, airlines are challenged to create more robust schedules, which needs to reflect the risk of longer taxi-out times, especially at

Abbreviations: AMS, Amsterdam Schipol Airport; ARN, Stockholm-Arlanda Airport; ASPM, Aviation System Performance Metrics; ASQP, Aviation Service Quality Performance; BCN, Barcelona El Prat; BRT, Bagging Regression Trees; CLT, Charlotte Douglas International Airport; DAL, Delta Air Lines; FAA, Federal Aviation Administration; AAL, American Airlines; ANN, Artificial Neural Network; ARIMA, Autoregressive Integrated Moving Average; ARINC, Aeronautical Radio Inc.; ASA, Alaska Airlines; MSE, Mean Squared Error; MTW, Maximum Takeoff Weight; NAS, National Airspace System; OLS, Ordinary Least Squared; OOOI, Out Off On In (data from ARINC); OPSNET, Operations Network; ORD, Chicago O'Hare International Airport; RMSE, Root Mean Squared Error; RVR, Low-visibility operations using lower runway visual range; SEA, Seattle/Tacoma International Airport; SID, Area Navigation Standard Instrument Departures; TPA, Tampa International Airport; TSK, Tagagi-Sugeno-Kang (model); SVM, Support Vector Machine; SVR, Support Vector Regression; TFMS, Traffic Flow Management System; UAL, United Airlines; USDOT, U.S. Department of Transportation; ZRH, Zurich Airport

E-mail address: tony.diana@aol.com.

[1] Susan Carey, "Seven Hours of Sitting and Waiting Leaves Northwest Passengers Near Breaking Point," Wall Street Journal, April 28, 1999.

[2] See 14 CFR Part 244 on reporting tarmac delays, retrieved at https://www.gpo.gov/fdsys/granule/CFR-2012-title14-vol4/CFR-2012-title14-vol4-part244/content-detail.html.

https://doi.org/10.1016/j.tre.2018.10.003
Received 22 May 2018; Received in revised form 6 October 2018; Accepted 8 October 2018
Available online 15 October 2018
1366-5545/ Published by Elsevier Ltd.

peak times. Given a set of parameters, analysts can determine an unimpeded taxi-out time that serves to compute taxi-out delays as the difference between actual and unimpeded taxi-out times. Ensemble learning methods make it possible for algorithms to learn quickly from experience and produce real-time data and predictions.

Second, it is important for airport operations and planning to identify choke points between gate-out and take-off to recommend and implement infrastructure improvements for optimal surface movement flows. Airport operators can estimate departure clearance time compliance.

Finally, government analysts have an interest in predicting taxi-out times to plan optimal capacity for maximum throughputs as well as to evaluate the impacts of regulations and implemented capabilities on traffic flows. Although a pilot may recover taxi-out delays during a non-stop flight, longer than expected taxi-out times can contribute to delay propagation, especially when a trip involves a series of legs. Research showed that point-to-point carriers find it more difficult to internalize delays and their derived costs because they do not operate a hub where they can coordinate connections between arrivals and departures (Mayer and Sinai, 2003).

This case study evaluates whether ensemble learning models are more likely to improve forecasts of taxi-out times than 'traditional' linear models. Nowadays, open-source software such as Python and R makes it easier for analysts to train, test, and validate models. Moreover, libraries such as Scikit-Learn enable Python programmers, for instance, to compare the validity and reliability of their models and select the best-performing one(s) using sophisticated diagnostic tools.

Based on a same set of variables, this study compares the outputs from ten algorithms including ordinary least-squared (OLS), regularized or penalized regression, Support Vector Regression (SVR), and ensemble learning models. Each model under investigation assumes that five independent operational variables are likely to explain variations in taxi-out times in the case of Seattle/Tacoma International Airport (SEA): departure demand, departure throughputs, percentage of total airport capacity utilized, approach conditions, and runway configurations. The theory of airport capacity predicts that an increase in departure demand will constrain available airport capacity, which may result in longer taxi-out times and, eventually, takeoff queues. As more planes wait in line to take off, the surface movement area gets congested, which delays gate departures (see DeNeufville and Odoni, 2003; Horonjeff et al., 2010).

This paper contributes to the literature of data analytics by evaluating the performance of ten regression algorithms using actual hourly data. Second, it utilizes Python, an open-source software, and selected libraries to determine whether ensemble learning models are best suited to learn complex relationships among variables and are more robust to outliers than 'traditional' linear models. Third, it compares the predictive capabilities of the selected algorithms before and after the implementation of NextGen technologies. NextGen or Next Generation Air Transportation System represents a portfolio of programs that the Federal Aviation Administration (FAA) has been spearheading to transition the NAS from a radar-based to a satellite-based navigation system.

2. Literature review

Taxi-out time often serves as an indicator of airport congestion as observed times from pushback to takeoff exceed a nominal time (Kistler and Gupta, 2009; Atkin et al., 2010). On the one hand, taxi-out time is an important input in determining the level of service and the overall airport strategy to meet departure demand. Queueing models often play a significant role in this perspective. Regulating taxi-out time represents one of the tools to manage an airport's runway service process with the goal of facilitating optimal departure flows (Shumsky, 1997). Pujet et al. (2000) estimated taxi-out time as the sum of travel time (from gate to entry into the departure queue) and wait time for takeoff. Simaiakis and Balakrishnan (2009) fragmented taxi-out time into three components: unimpeded taxi-out time, time in departure queue, and congestion delay.

On the other hand, the duration of taxi-out time can serve to predict the airport system's behavior and help airport operators and airlines manage arrival and departure capacity. In this perspective, analysts can resort to supervised models—which predicts taxi-out times as a function of response variables—or unsupervised models—which focus on the underlying structure or distribution in the data. In this article, OLS, penalized and support vector regressions, as well as ensemble learning methods are part of supervised learning problems, whereas clustering and associations belong to the unsupervised learning category. The models utilized in this study are designed to support short to medium-term forecasts. Some of the different approaches to the study of taxi-out time are summarized in Table 1.

The use of ensemble learning methods to forecast airport traffic activities is not yet widespread. However, it can be anticipated to increase as open-source software such R and Python tap on more sophisticated algorithm libraries.

Most of the studies referenced in Table 1 relied on airport cases to validate the outcomes of the predictive models. This allows analysts to assess whether each model was capable of learning from changes in observed and latent conditions at a given airport. Airport and airline operators who are familiar with an airport can improve the predictive power of each model by changing the parameters of the learning sample and validating the outcomes using cross-validation. Because each airport is different in its configuration, operations patterns, and traffic mix, it is difficult to compare the models' performance across a group of airports. Although the research and validation methodology can be applied to different cases, the model outcomes remain limited to a specific case. Few studies extended analysis of taxi time beyond two airports (Xu et al., 2008; Deshpande and Arıkan, 2012; Rebollo and Balakrishnan, 2012).

OLS and penalized models are not as capable to capture the variability and granularity of change necessary to improve forecast as ensemble learning models. This explains why ensemble learning methods are often used to forecast the impact of weather on airline and airport operations (Wang, 2011). Yet, frailty analysis indicated that taxi-out times may not be significantly affected by either fixed or random effects (Diana, 2013).

The present analysis assumed that ensemble learning through the combination of several learners would improve forecasts of taxi-out times compared with a single learner in the OLS, penalized, and support vector models. Ensemble learning can be compared with asking the opinion of several experts to avoid bias and variance among advice. The combination of several learners would help decrease variance (bagging), bias (boosting), and improve predictions (stacking). In the ensemble learning method, either the base

Table 1
Highlights of taxi-out time studies.

Authors	Year	Methodology	Features	Sample/Case
Herbert and Dietz	1997	Queueing model	Departure modeled as non-homogeneous Poisson process. Service times were modeled as appropriate mixtures of exponential stages	LGA
Shumsky	1997	Explanatory aircraft-flow model to predict departure congestion and queueing model	Taxi-out times depended on airline, departure runway, and departure demand. The models produced accurate predictions of airfield congestion over 10 min to one-hour-forecast horizons	ATL and BOS
Idris et al.	2002	Queueing model	Fourteen-day average running model served as benchmark. Queueing model improved the Mean Absolute Error by 20% when compared with average running model	BOS
Tu et al.	2008	Spline, smoothing-based non parametric model	Prediction model used delay propagation effects and seasonal trends for forecasting flight departure delays	DEN
Xu et al.	2008	Estimation of positive and negative delays using multivariate adaptive regression spline models	Model enabled to detect non-linear relationships between response and predictors	34 most congested U.S. airports
Simaiakis and Balakrishnan	2009	Queueing model	The departure taxi (taxi-out) time of an aircraft was represented as a sum of three components, namely, the unimpeded taxi-out time, the time spent in the departure queue, and the congestion delay due to ramp and taxiway interactions. The model emphasized departure queue management to minimize fuel burn and emission	BOS
Balakrishnan et al.	2010	Non-parametric reinforcement learning model set in the probabilistic framework of dynamic programming	Mean predicted value of taxi-out time matched actual values with a standard error of 1.5 min	TPA
Srivastava	2011	Linear regression models to predict taxi-out and taxi-out delays	First model treated aircraft movement from starting location to the runway threshold uniformly. The second modeled aircraft time to get to the runway queues (nominal time), which was different from the wait time experienced by aircraft in the runway queue (queue time)	JFK
Deshpande and Arikan	2012	Determination of flight delay costs under newsvendor model	Model focused on the Impact of scheduled block time on on-time arrival performance	All domestic commercial flights operated by major carrier in 2007
Rebollo and Balakrishnan	2012	Random Forest model	Analysis of classification and regression performance. The average test error across these 100 origin and destination pairs was 19%	100 most delayed origin and destination pairs in U.S.
de Leege et al.	2013	Machine learning model to predict arrival trajectory. The model is based on a supervised learning regression problem	At a prediction horizon of 45 nautical miles, the model explained 63% of the observed variance in the arrival time. The mean absolute time error was 18 s. Aircraft types had little explanatory power. Aircraft trajectory and weather were used to train the model	AMS
Diana	2013	Survival and frailty analysis	Survival models explained how some operational factors explained changes in the duration of taxi-out times. Frailty analysis determined whether fixed or random effects explained differences between two samples	JFK
Ravizza et al.	2014	Multiple linear regression, Least Median Squared linear regression, Support Vector Regression, M5 model trees, and TSK and Mamdani fuzzy ruled-based systems	Tagagi-Sugeno-Kang (TSK) fuzzy-ruled based system increased taxi time predictions from historical data	ARN and ZRH
Kim	2016	Non-parametric additive models using splines	Short-term forecasting time. Linear and median regressions used as benchmark to validate non-parametric models	DEN
Lee et al.	2016	The authored compared various machine learning methods such as linear regression, support vector machines, k-nearest neighbors, random forest, and neural networks model	Linear regression and random forest techniques provided the most accurate prediction in terms of root-mean-square errors	CLT
Lordan et al.	2016	Log-linear regression to estimate taxi-out time	Model included route- and interaction-specific factors. Route-specific factors are useful to estimate taxi-times. The variability of taxi times depended on the combination of stand and rapid exit variables (landing) and runway (takeoff)	BCN

learners are generated sequentially to exploit dependence between them (i.e. AdaBoost) or in parallel to take advantage of independence between base learners (i.e. RandomForest).

Ensemble learning is still a fast-growing research area in aviation. Ensemble learning algorithms provide several benefits. A model can learn from historical data and be trained to provide the best prediction. Despite the recent success of ensemble learning methods, they present some disadvantages that may not make them appropriate in all situations. First, it is not always easy to interpret the results. Ensemble learning methods may look like 'black boxes.' For instance, one limitation of RandomForest is that it generates a forest consisting of many trees and rules, which makes interpretation challenging. Second, some algorithms such as Boosting require large training samples to be efficient. Obtaining a large volume of data can be expensive and time-consuming in processing. Third, the complexity of the algorithms and adopted decision rules to aggregate decisions do not necessarily provide the best predictions as measured by the Mean Absolute Error or Root Mean Squared Errors as Lee et al. (2016) pointed out. Finally, analysts do not always have control over model learning and predictive outcomes. It is, therefore, important to establish some robust criteria to validate and select the models that provide the best balance between bias and variance as explained later in this article.

3. Methodology

3.1. Data samples and variables

The data originated from the Aviation Systems Performance Metrics (ASPM) data warehouse[3], which includes information on operations and delays. ASPM merges data from several sources:

- ARINC (Aeronautical Radio, Inc.), i.e. Out-Off-On-In or OOOI times
- Federal Aviation Administration, i.e. Traffic Flow Management System (TFMS) and Operations Network (OPSNET)
- U.S Department of Transportation, i.e. Aviation Service Quality Performance (ASQP).

Data were organized by day and by hour in two samples during the peak travel season: June to August 2015 and June to August 2016. Each sample consisted of 1380 observations. Comparing the two samples allows analysts to evaluate how the models handled summer's stochastic weather events, airport congestion, fluctuations in departure demands, and the latent effects of NextGen capabilities implemented at SEA.

The samples included observations from 07:00 to 21:59 (local time), when traffic is most active. Data pertained to all days of the week including holidays. It is important to note that runway 16C|34C was closed from May 1 through October 2015. All departures used runway 16L|34R as departures are not permitted from runway 16R|34L.

The model, which includes six variables, features the lowest Akaike Information Critierion (AIC)[4] value among other considered OLS models. The same variables were used in the other models for comparison.

3.2. Response variable

Taxi-Out Time is the response or dependent variable. Taxi-out time includes push-back time, taxi operation, wait at hold points, and time spent in take-off queues. It represents the average number of minutes an aircraft spends between gate-out and take-off, or the difference between take-off and gate-out times. Taxi-out time is an indicator of airport congestion: the higher the volume of operations, the longer it will take for an aircraft to move from gate to take-off. As departure demand increases, available departure capacity declines. With more aircraft demanding to depart, conga lines form at take-off, taxiways get congested, and taxi-out speed declines consequently. Separation requirements further increase inter-departure times and reduce maximum throughputs. However, wake vortex re-categorization has made it possible for controllers to reduce inter-departure times between selected pairs of aircraft types. Such a NextGen capability had been implemented at SEA.

3.3. Explanatory variables

Departure refers to the counts of aircraft that left the gate to take off. A departure is determined by a 'DZ' departure message code sent from the enroute host computer to TFMS and/or an OOOI record (wheels-off time).

Departure Demand includes the aircraft that left the gate but have not yet taken off from SEA to their destination. Departure demand increases at peak times and contributes to increasing airport congestion and delays as airport capacity declines, especially in the case of adverse weather and wind conditions.

Available Airport Capacity Utilized is the ratio of total operations (arrivals plus departures) to total airport capacity (airport arrival plus airport departure rates). The percentage of available airport capacity utilized depends on weather conditions, peak traffic, and the selection of runway configurations.

Runway characterizes the runway configuration used at a specific hour. In this study, we used the frequency of runway

[3] The ASPM website is https://aspm.faa.gov.

[4] The AIC value is defined as AIC $= 2k - 2\ln(\hat{L})$ with k, the number of estimated parameters and $\hat{L}$ the maximum value of the likelihood function of the model. See Akaike (1974).

Table 2
Change in selected operational variables.

Variables	June to August 2015	June to August 2016	Change
Total Operations	1,10,334	1,18,714	7.6%
Total Delays	1225	3441	180.9%
Weather	891	1881	111.1%
Volume	42	688	1538.1%
Equipment	0	37	–
Runway	269	529	96.7%
Other Causes	23	306	1230.4%
Operations in Instrument Approach Conditions (Percent)[1 2]	14.64	22.25	7.6%
Runway Configurations Use (Percent)[1]			
34L, 34R|34R	56.74	16.23	−40.5%
16L, 16R|16L	41.01	23.77	−17.2%
16L, 16R|16C, 16L	0.00	35.07	35.1%
34L, 34R|34C, 34R	0.07	22.97	22.9%
On-Time Gate Departures (Percent)[1 3]	78.25	81.18	2.9%
Average Taxi-Out Time (Minutes)[1]	16.27	17.17	0.90
Average Hourly Departure Demand[1]	35	39	11.4%
Average Hourly Departures[1]	33	35	6.1%
Total Available Capacity Utilized (Percent)[1]	84.74	83.13	−1.61%
Fleet Mix (Percent)			
Heavy	5.89	5.44	0.5%
Boeing 757	2.44	4.78	2.3%
Large Jet	55.50	57.45	2.0%
Commuter	35.24	31.45	−3.8%
Other	0.93	0.88	0.1%

Notes:
1. From 07:00 to 21:59. All other metrics are computed for the day.
2. Instrument approach conditions refer to ceilings less than 4000 ft and a visibility of less than 3 nautical miles.
3. On-time gate departures refer to the percentage of flights that arrived no later than 15 min at the gates compared with published schedules.
Source: ASPM.

configuration usage to label them in four categories as a dummy variable: '1' for '16L, 16R|16L', '2' for '16L, 16R|16C, 16L', '3' for '34L, 34R|34R', and '0' for 'other'. Appendix A, Section 3, provides a layout of the runways lat SEA.

IAC_1 stands for instrument approach conditions coded as '1′ and '0′ in case of visual approach conditions. Instrument approach conditions refer to circumstances when landing operations are conducted below minimum ceiling and visibility at a specific airport (less than 4000-foot ceiling and less than 3 nautical miles).

3.4. Key performance indicators of airport operations

Table 2 highlights changes in the selected variables and others not included in the models. This is important to understand the percentage of variations in taxi-out times not explained by departures, departure demand, the percent of total available capacity utilized, types of approach conditions, and the runway configuration used at a specific hour. Some metrics were available from 07:00 to 21:59, while others were computed for the entire day (as indicated in the table legend).

Table 2 summarizes differences in SEA's operational environment between the two samples. While operations increased 7.6 percent, total delays also went up in all categories tracked by the OPSNET database.[5] According to OPSNET data, all weather delays in 2016 occurred during the month of August.

As explained earlier, traffic mix affects inter-departure times to minimize wake-vortex turbulence and accidents. In this study, the takeoff weight determines the following categories:

- Any aircraft weighing more than 255,000 lbs. is classified as 'heavy' equipment (i.e. Boeing 777 or Airbus A 330)
- The label 'Boeing 757′ refers to Boeing 757–200 with a maximum takeoff weight (MTW) of 109,000 lbs., as well as Boeing 757–300 with an MTW of 272,500 lbs.
- 'Large' aircraft include aircraft with an MTW from 41,000 to less than 255,000 lbs. (i.e. Boeing 737 or Airbus A 320). This category includes larger commuter aircraft such as the Bombardier CRJ-700 with an MTW of about 75,000 lbs. and Embraer 175 with an MTW of 89,000 lbs. whose operations increased.

The percentage of Boeing 757s and large jet operations increased in the same proportion when comparing the two periods.

[5] See US Department of Transportation, Federal Aviation Administration, Air Traffic Organization Policy, Order JO 7210.55F, October 1, 2009 for a definition of the delay categories, retrieved at https://www.faa.gov/documentLibrary/media/Order/7210.55FBasic.pdf.

Table 3
NextGen capabilities.

Capabilities	Implementation
Airport Surface Detection Equipment (ASDE-X)	Feb-06 and Nov-08
Area Navigation (RNAV) Global Position System (GPS) Approach	Jan-07
Dependent Approaches to Parallel Runways JO 7110.308	Nov-08
Required Navigation Performance (RNP) Authorization Required (AR) Approaches	May-09
RNAV Standard Instrument Departures (SID)	May-09
Expanced Low-Visibility Operations Using Lower Runway Visual Range (RVR) Minima	May-11
RNP/AR Approaches	May-12
RNAV Standard Terminal Arrival Routes (STAR)	Mar-13
RNP/AR Approaches	Mar-13
Optimized Profile Descent	Mar-13
Expanded Low-Visibility Operations Using Lower Runway Visual Range (RVR) Minima	Mar-13
Deployment of Time-Based Flow Management (TBFM)	Aug-13
Advanced and Efficient RNP	Fiscal Year 2014
Qualifies for Dependent Runway Standards in Order 7110.65	Dec-15
Ground-Based Interval Management Spacing (GIM-S) adapted for Tower	May-16

Source: FAA, NextGen Performance Snapshots, http://www.faa.gov/nextgen/snapshots.

However, the percentage of commuter aircraft went down due to an increase in the number of larger commuter jet aircraft such as the Embraer 175 used by American (AAL), Alaska (ASA), and Delta (DAL), and United (UAL). While carriers such as Delta and American reduced their number of flights, they increased their available seats through larger aircraft.

The decline in the percentage of total available capacity utilized in the 2016 sample can be explained by poor weather conditions and resulting volume delays. Nonetheless, compared with published schedules, the percentage of on-time gate departures measured during the core operational hours slightly increased in the post-sample, from 78.25 to 81.18 percent.

It is also important to note that the predominant runway configurations changed in the 2016 sample. In the pre-sample, takeoff followed a South-to-North flow, while they were in the opposite directions in the 2016 sample.

3.5. NextGen capabilities implemented at SEA

NextGen capabilities refer to procedures and technologies designed to transition the NAS to a satellite-based navigation system. Table 3 provides details on the capabilities deployed at SEA: precision approaches (RNAV, RNP) to the runway, multiple runway operations, optimized profile descent (a smoother descent with fewer aircraft level-offs), and interval spacing before the 2016 sampled period. Capabilities are navigational improvements and changes in procedures that take advantage of the latest avionics technologies to improve flight efficiency, predictability, and improve airport capacity utilization.

It is difficult to compare two periods without referring to their latent effect on airport capacity. Table 3 can provide some insights on the percent of unexplained variations in taxi-out times. Nevertheless, the individual impact of each of the capacities listed is difficult to measure for several reasons. First, it takes time for controllers to learn and apply the new procedures. The same can be said for pilots. Second, the combined impact of NextGen capabilities may have some synergetic effects on taxi-out times only at specific times of the day (peak times) and under specific conditions (RVR). Such is the case, for instance, of departure metering and wake vortex re-categorization whose joint effects may improve capacity utilization and departure throughputs and reduce taxi-out times.

3.6. Model assumptions and algorithms

According to Bowles (2015), predictive modeling needs to balance three elements: (1) performance, (2) complexity, and (3) data quality and volume. Data analysts can choose among several types of model to predict a variable:

- Univariate versus multivariate
- Linear versus non-linear
- Parametric versus non-parametric.

In this study, we compared the performance of multivariate, linear, and parametric models (linear and regularized or penalized regressions) with a multivariate, non-linear, parametric model (Support Vector Regression) and multivariate, non-linear, non-parametric models (ensemble learning models). Time series, Petri Nets, survival/duration, and deep learning models (neural networks) represent a few alternatives to predictive models beyond the scope of this study (see Diana, 2010; Balakrishna et al., 2012; Diana, 2013). For a review of regression models with Python, see Massaron and Boschetti (2016).

- Linear Regression

The objective of the linear model is to minimize the residual sum of squares between targets and responses predicted by linear

approximation:

$$\min_{w} ||Xw-y||_2^2 \tag{1}$$

where X is an independent variable; y, the response or target variable; w, a coefficient; and $||.||_2$, the Euclidean norm.[6]

In the present study, the ordinary least squares model or OLS can be summarized as follows:

$$\begin{aligned}\text{Taxi-Out Time} = \beta_0 + \beta_1 * \text{Departures} + \beta_2 * \text{Departure Demand} + \beta_3 * \text{Capacity Utilized} + \\ D_1 * \text{Runway Configuration} + D_2 * \text{Approach Condition} + \varepsilon\end{aligned} \tag{2}$$

where β_0 is the intercept; β, the regressor; D, a dummy variable, and ε, the error term. The basic premise of OLS is to predict the variation in taxi-out times explained by independent variables. OLS models have some limitations. First, OLS models may overfit the data. Second, they are constrained by many degrees of freedom, that is, not enough data for the number of degrees of freedom. This explains why Ridge, Lasso, and Elastic Net regressions were included in this study.

- Regularized or Penalized Regression

A regression model that uses L1 regularization is called Lasso regression, whereas a model using L2 is called Ridge regression. According to Bowles (2015), "penalized linear regression provides a way to systematically reduce degrees of freedom to match the amount of data available and the complexity of the underlying phenomena." This study presents the outcomes of three types of regularized linear regression.

- The *Ridge* model is expressed as

$$\min_{w} ||Xw-y||_2^2 + \alpha||w||_2^2 \tag{3}$$

where α is a tuning parameter that controls the strength of the penalty term $||w||_2^2$. The L2 regularization term ($||w||_2^2$) is a penalty equivalent to the square of the magnitude of the coefficients. When α = 1, then there is no difference with the linear regression model. In the present study, α = 0.88.

- The *Lasso* (Least Absolute Shrinkage and Selection Operator) model is defined as

$$\min_{w} \frac{1}{2n_{samples}} ||Xw-y||_2^2 + \alpha\rho||w||_1 \tag{4}$$

where $||w||_1$ is the L1 regularization as the penalty equivalent to the absolute value of the magnitude of the coefficients.

- The *Elastic Net* combines the L1 and L2 penalties of the Lasso and Ridge methods, respectively. The Elastic Net model is expressed as

$$\min_{w} \frac{1}{2n_{samples}} ||Xw-y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2 \tag{5}$$

w?>where 1 < α < 0. The hyper parameter α controls how much L2 and L1 penalization is used (0 for the Ridge and 1 for the Lasso model).

Regularized models fit a regression model and they penalize or shrink large coefficients. They can help with the bias/variance trade-off and model selection. Nevertheless, regularized models may present some limitations: they are demanding on large data sets and they do not perform as well as Random Forest and Boosting models.

- Support Vector Regression

The *Support Vector Regression* model is a non-parametric method that relies on kernel functions (especially polynomial and radial basis function). The goal of the model is to find a function f(x) that deviates from y by a value no greater than epsilon (ε) for each training point x. Epsilon was set to 1 in the SVR model. The kernel function transforms sampled data from non-linear space to linear space. The kernel function allows the SVR to find a fit and then data are mapped to the original space. The problem can be formulated as a convex optimization problem such that

$$\min \frac{1}{2} ||w||^2 \text{ subject to } y-w_1*w_i- b \leqslant \varepsilon \text{ and } w_1*x_i + b-y_i \leqslant \varepsilon \tag{6}$$

SVR models have been used in stock price prediction and optical character recognition.

- Ensemble Machine Learning

[6] The Euclidean norm or 2-norm, hence the 2 subscript, is defined by $\|v\| = \sqrt{\sum_{k=1}^{N} |v_k|^2}$ for a vector v with N elements.

Ensemble algorithms combine predictions from multiple separate models. Ensemble learning models allow analysts to take advantage of 'aggregated' rather than 'individual' answers. This study involves several ensemble algorithms. A detailed explanation of each model is beyond the scope of this article. Readers interested in ensemble algorithms are referred to Sutton (2005), Sammut and Webb (2011), Bühlmann (2012), Zhang and Ma (2012), and Raschka and Mirjalili (2017) for a comprehensive treatment and detailed specification of the algorithms.

In the *Random Forest* algorithm, the process of finding the root nodes and splitting the feature nodes happens randomly. Each tree is generated depending on a bootstrap sample based on a fixed probability distribution. Each random forest will predict different outcomes for the same test feature, which are aggregated by majority rule.

The *Extra Trees* (Extremely Randomized Trees) algorithm is a variant of the Random Forest algorithm. At each test node, the best split is chosen among k random splits, and each one is determined by a random selection of an input without replacement and a threshold.

Bagging and *Boosting* algorithms approach the problem from opposite directions. With Bagging, 'strong' learners are trained in parallel. The algorithm aims to reduce the possibility of overfitting complex models. Strong learners refer to models that are relatively unconstrained. Bagging combines all the strong learners to smooth their predictions.

In the case of *Boosting* models, the focus is on improving the predictive capability of simple models. 'Weak' learners are trained in sequence. The 'weak' learner represents a constrained model. Each 'weak' learner learns from the mistakes of the preceding one in sequence. Boosting combines all 'weak' learners into a single 'strong' learner.

3.7. Predictive performance and model selection criteria

Dangeti (2017) summarized the difference between statistical and machine learning modeling in these terms: "In statistical modeling, lots of tests are performed at the individual parameter level apart from aggregated metrics, whereas in machine learning we do not have visibility at the individual parameter level." This explains the choice of aggregated level metrics such as the coefficient of determination, learning curves, and cross-validation. This section provides the key performance indicators that guide analysts in the selection of a model.

- Coefficient of Determination

The 'r2_score' function in SciKit-Learn measures the variation in the dependent variable explained by the dependent variable(s). The coefficient can be positive or negative. A negative value occurs when a model tries to fit non-linear functions to sampled data. A score of 1.0 is the best possible value. The coefficient of determination can be expressed as

$$R^2 = 1 - \frac{Residual\ Sum\ of\ Squares}{Total\ Sum\ of\ Squares} \text{ or } R^2 = 1 - (\text{fraction of variance unexplained}) \tag{7}$$

- Root Mean Squared Error

One key metric in the evaluation of model performance is Root Mean Squared Error (RMSE), which can be characterized as

$$RMSE = \sqrt{\frac{sum\ of\ squared\ errors}{number\ of\ observations}} \tag{8}$$

- Cross-Validation

Cross-validation is used to identify the optimal value of a parameter. According to Sammut and Webb (2011), "cross-validation is a process for creating a distribution of pairs of training and test sets out of a single dataset." The 2015 and 2016 sampled data were divided into train, validation, and test sets. The training dataset represented 60 percent of the sampled data. Usually, the training dataset is used to fit the model. The test dataset serves to assess the generalization error of the selected model. The validation dataset enables estimate prediction errors for model selection.

Cross-validation is a valuable diagnostic tool to compare the robustness of the selected models. In ten-fold cross-validation, the 2015 and 2016 sampled data were separated into ten parts, subsequently trained on nine parts and validated on one part of the data. The process ran ten times to cover all points in the sampled data. The mean and the standard deviation were computed out of ten generated values.

Using the SciKit-Learn library, the 'cross_val_score' function generated the variance score for each for the ten folds. The mean score represents the average variance score over the ten folds. The standard deviation measures the spread to variance scores around the mean.

Error is defined as the difference between target and prediction. Errors can be decomposed into three components: bias, variance, and white noise. This can be summarized by the following equation:

$$E(y - \hat{f}(x))^2 = var(\hat{f}(x)) + [bias(\hat{f}(x))]^2 + var(\varepsilon) \tag{9}$$

Géron (2017) pointed out the trade-off involved in building models. When the model is biased, it will most likely underfit training data. A model characterized by high variance is more likely to overfit training data. As a model gets more complex, variance is likely to increase, while bias is likely to decrease, and vice versa. Overfitting occurs when the model shows low bias and high variance.

Usually, if a model does better on the training set than on the test set, then it is most likely overfitting. We can resort to a training set to train and tune the model using cross-validation. Overfitting implies that the train model fits too closely the training dataset. Although it may be accurate on the training data, it may not be accurate on untrained or new data. Therefore, an overfitting model cannot be trusted to generalize the results and infer from new data. When overfitting occurs, the model has learned to describe the noise in the model instead of the actual relationship between the response and independent variables.

Underfitting occurs when a model exhibits low variance but high bias, especially when a model is over-simplified. Both overfitting and underfitting lead to poor predictions on new datasets. When a model is underfitted, it does not fit the training data and it misses trends such as increases in operations and demand at peak-times. This explains why train/test split and cross-validation are important to remedy fitting issues.

3.8. Software and libraries

Python (version 3.6), an open-source software, generated all model outputs. The software utilized the Scikit-Learn library,[7] which includes linear, regularized, ensemble, and Support Vector Regression models, among many others. Interested readers are referred to Géron (2017) for an overview of the capabilities of SciKit-Learn with Python. The 'model selection' module enabled to train, test, and validate samples as well as to determine cross-validation predictions. The 'metrics' library generated score metrics and root mean squared error.

4. Comparison of model outcomes

This section presents several methods to compare and select (a) model(s). Some models (i.e. OLS and regularized) are parametric, whereas all the others are not. For non-parametric models, it is not possible to drill down to the variable estimates and evaluate changes in the magnitude and sign of the estimates.

4.1. The coefficients of determination

Table 4 includes the coefficients of determination (referred to as 'scores' in Scikit-Learn) for the ten models using 2015 and 2016 sampled data. The magnitude of the coefficients increased overall in the 2016 sample, as weather events and frequent runways configuration changes were more likely to impact taxi-out times. Usually, a higher R^2 value means that the independent variables explained a larger proportion of the variations in taxi-out times. However, high coefficients of determination may also suggest some overfitting issues.

The non-parametric SVR model applies a classification algorithm to predict real values. The Radial Basis Function (RBF) served as a kernel function as the default. The SVR model generated a negative value in the 2015 sample, meaning that the model did not fit variations in the data. Although the R^2 value coefficient turned positive in the 2016 sample, the independent variables could explain only 11 percent of the variations in taxi-out times. Therefore, the SVR algorithm should not be considered for further consideration in the present case.

The higher coefficients of determination in the 2016 model imply that the fraction of unexplained variation declined as most of the independent variables accounted for a greater effect on taxi-out operations. At first sight, the coefficients suggest a better fit in the post-sample. However, analysts should be cautious when using the coefficient of determination to measure and compare the performance of models, especially when data depart from normal assumptions due to outliers. While kurtosis measures the peak of the distribution, skewness determines whether the distribution is symmetrical around the mean. The kurtosis of the normal distribution is 3. In the case of taxi-out time, it was 2.35 and 2.99, respectively for the 2015 and 2016 sample. The skewness of a normal distribution is zero. However, the skewness coefficient of taxi-out time was 1.12 and 1.44 respectively, thus implying a longer right tail in the 2016 sample than in the 2015 one.

In the 2015 and 2016 samples, the 'imc_1' variable in the OLS models was not significant at a 95 percent level: the p-value was respectively 0.127 and 0.121 (p-value $>$ 0.05). Also, the intercepts for both sample models were not significant at a 95 percent level. This means that the response variable would be very close to zero if all factors were zero. Finally, the Durbin-Watson coefficients suggested the presence of some positive autocorrelation in the residuals (1.63 and 1.74 respectively for the 2015 and 2016 samples). Compared with OLS and penalized models, the non-parametric SVR and ensemble learning models do not provide any test on the significance of the factors. This may limit sensitivity analysis and estimation of the potential impact of selected variables on taxi-out time in the pre- and post-sample.

4.2. Cross-Validation

Table 5 provides the RMSE and standard deviations for the train and test datasets.

First, we determine the fit the algorithms on the train dataset. Then, we make predictions on the test dataset.

The OLS and Bagging models appear to provide the best balance of bias compared with variance with 2015 data. However, ensemble learning models such as Random Forest, Bagging, and Gradient Boosting appeared more balanced as there was less difference between the train and test RMSE values. The Support Vector Regression model was the worst performing model in both sampled periods. To short-list the models, we turn to the RMSE values generated with the validation dataset.

[7] The Scikit-Learn library documentation is available at http://www.scikit-learn.org.

Table 4
Coefficients of determination (June-August).

Model	2015	2016
Linear Regression	0.56	0.74
Ridge Regression	0.56	0.74
Lasso Regression	0.37	0.70
ElasticNet Regression	0.51	0.71
Support Vector Regression	−0.22	0.11
Random Forest Regression	0.90	0.95
AdaBoost Regression	0.45	0.65
Bagging Regression	0.90	0.95
Extra Trees Regression	0.98	0.99
Gradient Boosting Regression	0.67	0.82

Table 6 features the same methodology as the one used to generate the mean, standard deviation of the root-mean-squared errors (RMSE) from the ten-fold process (Table 5). As a heuristic, the model with the lowest RMSE is preferred.

Table 6 indicates that the OLS and Ridge regression models had a similar mean and standard deviation for the RMSE in the 2015 validation dataset. The preferred model is the one featuring the lowest RMSE—highlighted by the gray-shaded areas. The OLS and Ridge regressions performed better with the 2015 validation dataset and the Gradient Boosting regression with the 2016 validation dataset.

5. Learning curves

In Section 1 and 2 of the Appendix A, the learning curves show how the scores of the training and cross-validation models for each algorithm varied as a function of the training set size. Cross-validation used 1000 iterations to get smoother mean test and train score curves, each with 30 percent data randomly selected as a validation set.

On the one hand, high bias may exist when training and testing scores converge and are high. Such is the case, for instance, of the AdaBoost algorithm with 2016 data. On the other hand, there may be high variance in a model when a large gap separates the training and cross-validation scores. The training scores get closer to the cross-validation ones as the performance of the algorithm on the training and cross-validation datasets become similar. Such is the case of the Random Forest and Bagging algorithms with 2015 data. Note that, in case of the linear model, the cross-validation curve using 2015 data converges as a vertical line toward the training one at the maximum number of observations. The variance associated with the linear model increased with 2016 data because of the increased complexity of the model. Adding more training samples may not necessarily increase generalization.

5.1. Comparison of predictions among models

Figs. 1 and 2 show the predicted average minutes of taxi-out time for 20 h using the ten algorithms with 2015 and 2016 sampled data.

A comparison of both graphs suggested closer prediction values in terms of magnitude among the ten algorithms with 2016 data. However, the predictions varied over a greater range in the 2016 sample (14 to 28 min) compared with the 2015 sample (14 to 23 min).

5.2. Mean and variance tests of predictions

Assuming the data were not normally distributed, we reject the null hypothesis in the case of the 2015 sample that all sampled

Table 5
Cross-validation scores. (Train and test datasets, root mean squared error, 10-fold cross-validation).

	Mean				Standard Deviation			
	2015		2016		2015		2016	
Model	Train	Test	Train	Test	Train	Test	Train	Test
Linear Regression	2.2303	2.3079	3.5444	3.4089	0.2963	0.3128	0.3896	0.4613
Ridge Regression	2.2303	2.3079	3.5443	3.4088	0.2963	0.3128	0.3896	0.4613
Lasso Regression	2.2276	2.2965	3.5249	3.3928	0.3000	0.2933	0.3894	0.4626
ElasticNet Regression	2.2302	2.3012	3.5279	3.3929	0.2985	0.2983	0.3904	0.4615
Support Vector Regression	2.8280	3.2845	4.1223	4.6844	1.0535	0.9126	0.6273	1.1886
Random Forest Regression	2.5712	2.6482	3.9260	3.9554	0.2798	0.2661	0.3905	0.6010
AdaBoost Regression	2.4594	2.5340	3.9215	3.8446	0.2350	0.3110	0.3703	0.4179
Bagging Regression	2.6267	2.6582	3.8755	3.8949	0.3287	0.3149	0.4065	0.6056
Extra Trees Regression	2.7077	2.8100	4.1089	4.0472	0.3467	0.2560	0.5228	0.6113
Gradient Boosting Regression	2.3144	2.4968	3.6319	3.6415	0.2953	0.3690	0.4025	0.6113

Table 6
Cross-validation scores. (Validation dataset, root-mean-squared error, 10-fold cross-validation).

(Validation Dataset, Root-Mean-Squared Error, 10-Fold Cross-Validation)

	2015		2016	
Model	Mean	Stdev	Mean	Stdev
Linear Regression	1.5065	0.2249	1.7921	0.1736
Ridge Regression	1.5065	0.2249	1.7921	0.1737
Lasso Regression	1.8098	0.3351	1.9242	0.1904
ElasticNet Regression	1.5939	0.2350	1.8710	0.1861
Support Vector Regression	1.7466	0.4609	2.1699	0.8588
Random Forest Regression	1.6827	0.2156	1.9126	0.1654
AdaBoost Regression	1.7996	0.2128	2.2131	0.1420
Bagging Regression	1.6400	0.2159	1.9134	0.1617
Extra Trees Regression	1.7563	0.2187	2.0088	0.1655
Gradient Boosting Regression	1.5416	0.2254	1.7681	0.1547

Best bias/variance balance

predictions came from populations with identical medians (the p-value of the non-parametric Kruskal-Wallis H test was 0.0046). However, for the 2016 sample, we accept the null hypothesis that all sampled predictions came from populations with different medians (the p-value of the non-parametric Kruskal-Wallis H test was 0.2202).

The Bartlett test determines whether the ten samples had equal variance. Based on 2015 sampled data, we conclude at a 95 percent level that there was variance in the predicted values of taxi-out times generated by the ten algorithms (statistic = 14.2345 and p-value = 0.1142). We also conclude that there was variance in the predicted values of taxi-out times based on 2016 sampled data (statistics = 3.3868, p-value = 0.9469).

6. Final remarks

Taxi-out time is difficult to predict because it depends on many latent variables such as airlines' taxiing procedures, airport layout, peak traffic hours, the choice of runway configurations, implemented NextGen capabilities (i.e. the implementation of departure metering, wake vortex re-categorization), and weather events, among many other factors. Taxi-out time represents a key component in building robust schedules and predicting airport congestion. Given the complexity of taxi-out operations, this paper proposed to evaluate and compare the predictive power of several popular regression and ensemble machine learning models. This research is timely as there is some increasing interest among aviation practitioners for ensemble learning models to help predict complex, non-linear relationships.

The popularity of ensemble learning models has increased over the last decade thanks to open-source software such as Python and

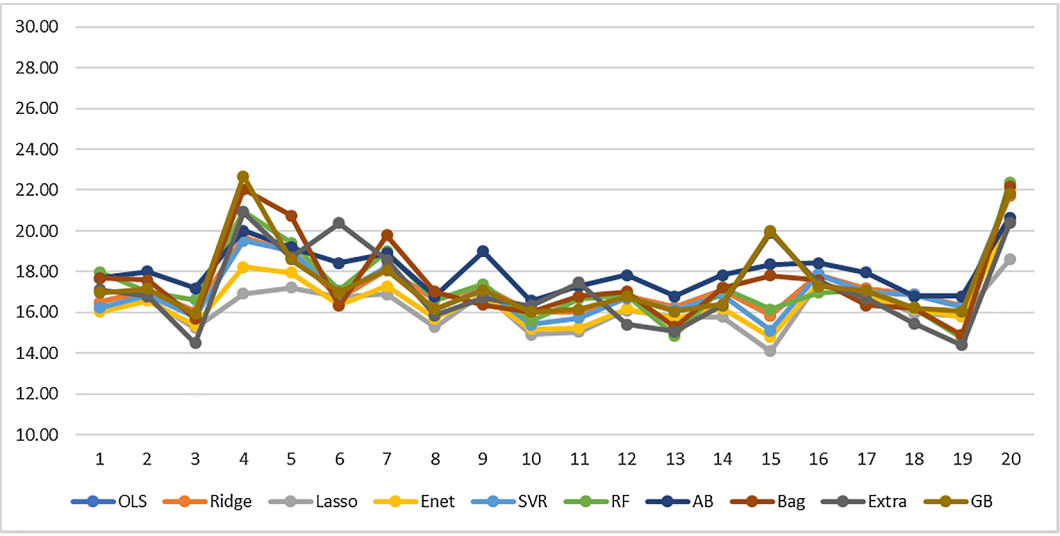


Fig. 1. Predicted average taxi-out time in minutes (2015 sample).

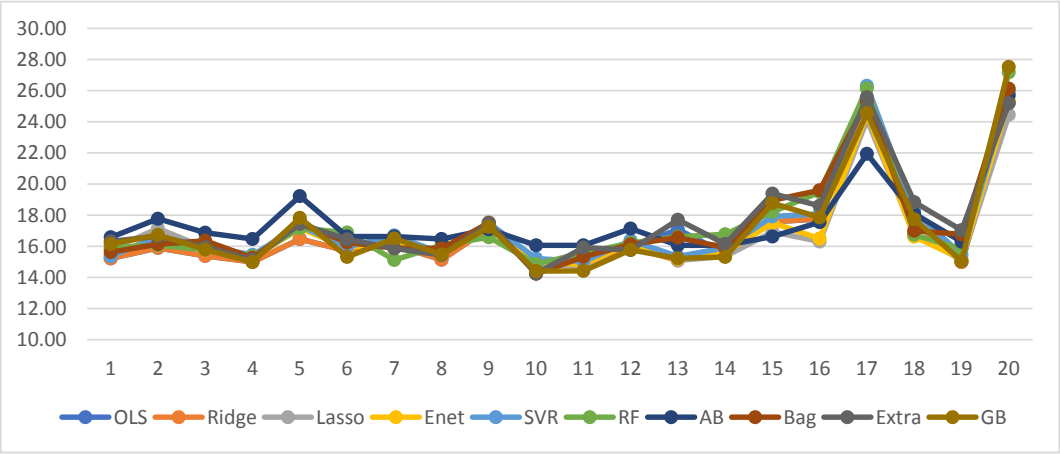


Fig. 2. Predicted average taxi-out time in minutes (2016 sample).

related libraries used in this case study. Such algorithms have made it possible to teach machines how to improve predictions through training, testing, and validating data. Machine learning enables models to learn based on a given sample of data and generate a short-term forecast that airline and airport operators can use to anticipate surface congestion. However, analysts need to remain alert to issues of trade-off between variance and bias when selecting models. Moreover, ensemble learning models do not always offer the possibility to delve to the variable level as in the case of linear models, for instance. Linear and penalized models make it easier to explain the individual impacts of explanatory variables on the variations of taxi-out times, whereas non-parametric ensemble learning and Support Vector Regression models do not. Scikit-Learn can generate the intercepts and coefficients in the cases of the OLS and regularized models, but it cannot for the SVR and ensemble models. While the OLS and regularized models allow to test the significance of estimates, non-parametric ensemble learning models cannot. This may prevent aviation practitioners from conducting sensitivity analysis and determining how the impact of some explanatory variables may have changed in a period-to-period comparison.

This study compared the predictive power of some selected supervised machine learning models including regression (OLS and regularized/penalized), Support Vector Regression, and ensemble machine learning (Random Forest, Bagging, Adaptive Boosting, Extra Trees, and Gradient Boosting). In the case of SEA, the analysis showed that algorithms fit data differently when comparing two samples. When a model is too complex for a given training dataset, it tends to overfit the training data. As a result, it cannot generalize well to unseen or new data.

One issue associated with machine learning systems is that they are difficult to interpret. Besides, the behavior of ensemble learning model is unpredictable since the model performance depends on training. Several tools were used to measure the performance of each model: the coefficients of determination (R2), root mean squared error, cross-validation, and learning curves. Among them, the R^2 coefficient was the least reliable to single out the best model. Although the learning curves identified the models likely to be overfitting and underfitting, the RMSE and cross-validation represented a better assessment of predictive accuracy.

The model outcomes suggested that the OLS and Ridge models performed better in terms of lower RMSE when there were fewer instrument approach conditions and more predictable use of runway configurations such as in the case of 2015 data. The Gradient Boosting algorithm handled uncertainty and the latent effects of NextGen capability implementation better, although the OLS and Ridge models came a close second best in the case of the 2016 sample. Among all the algorithms, SVR was the least efficient.

The Kruskal-Wallis H test determined that the population medians on taxi-out times generated by the ten algorithms were different in both samples. Moreover, there was variance in the predicted values of taxi-out times generated by the ten algorithms in both samples. More challenging approach conditions may explain the variance.

As part of the lessons learned from this case study, it is important to stress that the model outcomes are specific to SEA, under the specific conditions underlying both samples. What can be generalized to other cases is the methodology to select among competing models. Second, high coefficients of determination may imply overfitting as the cases of ensemble learning models illustrated. Third, it is difficult to determine the impact of NextGen capabilities with ensemble learning models because analysts do not have the flexibility of identifying the magnitude and direction of change in selected variables. Fourth, the models discussed in this study may be more appropriate for short-term forecasts.

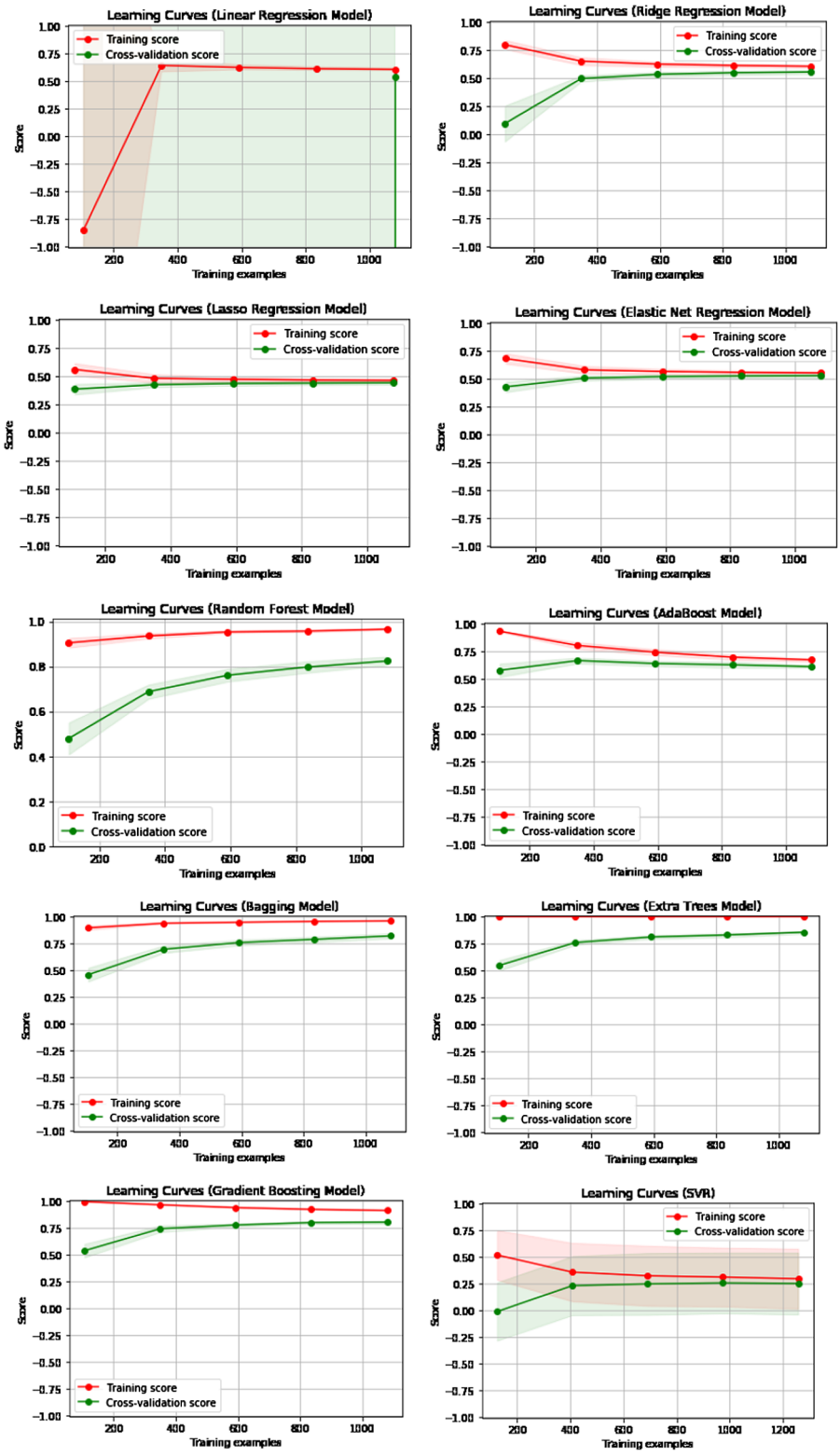
Machines can learn how to forecast taxi-out times. However, this case study suggested that despite the sophistication of ensemble learning models, analysts may still get a better predictive performance from more ‘traditional’ regression models based on the recommended selection criteria.

7. Disclaimer

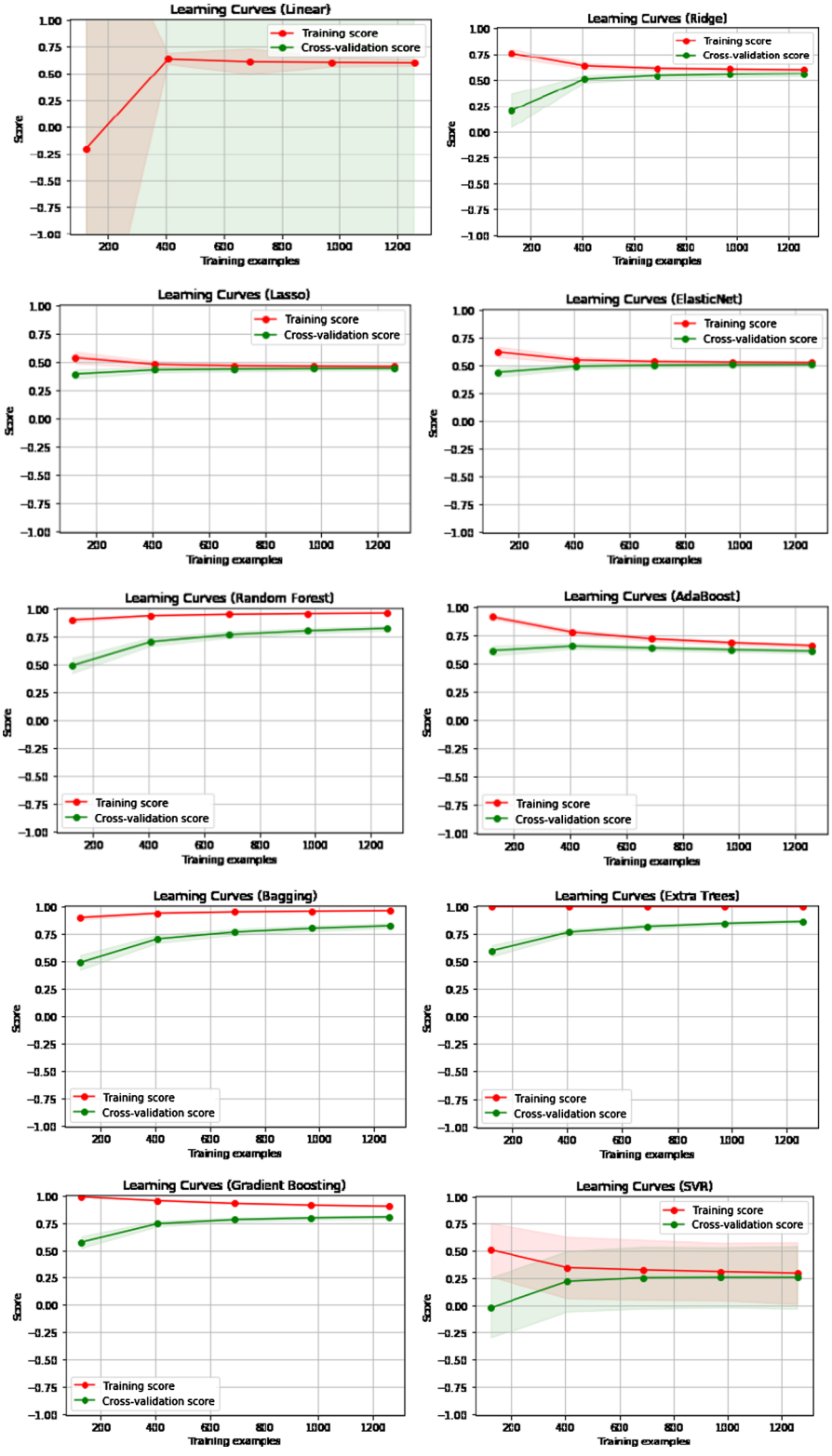
This paper does not represent the official opinion of the U.S. Federal Aviation Administration.

Appendix A

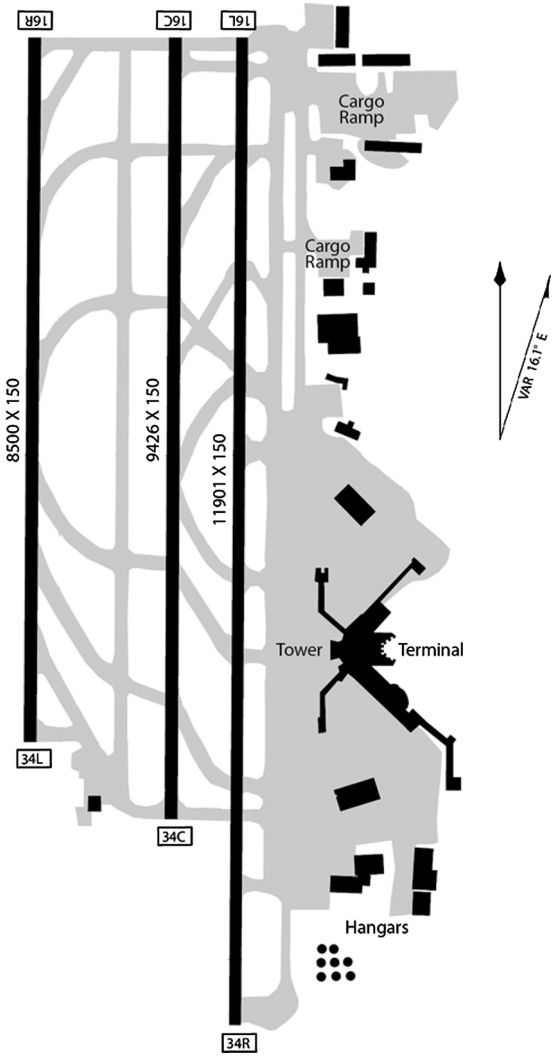
Section 1. Learning Curves (2015 Validation Dataset)



Section 2. Learning Curves (2016 Validation Dataset)



Section 3. Runway configurations at Seattle/Tacoma International Airport



Source: Federal Aviation Administration, NextGen Performance Snapshots website (http://www.faa.gov/nextgen/snapshots).

Appendix B. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.tre.2018.10.003.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Autom. Control 19 (6), 716–723.
Atkin, J.A., Burke, E.K., Ravizza, S., 2010. The airport ground movement problem: Past and current research and future directions. In: Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary. pp. 131–138.
Balakrishna, P., Diana, T., Kondo, A., 2012. Mapping the surface movement area operations: An application of Petri Nets to the case of New York JFK airport. J. Airport Manage. 6 (3), 260–273.
Balakrishnan, P., Ganesan, R., Sherry, L., 2010. Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures. Transport. Res. Part C: Emerging Technol. 18 (6), 950–962.
Bowles, M., 2015. Machine Learning in Python: Essential Techniques for Predictive Analysis. John Wiley and Sons, Indianapolis, IN.
Bühlmann, P., 2012. Bagging, boosting and ensemble methods. In: In Handbook of Computational Statistics. Springer, Berlin, Heidelberg, pp. 985–1022.
Dangeti, P., 2017. Statistics for Machine Learning. Packt Publishing, Birmingham, UK.
de Leege, A., van Paassen, M., Mulder, M., 2013. A machine learning approach to trajectory prediction. In: AIAA Guidance, Navigation, and Control (GNC) Conference. pp. 4782.
DeNeufville, R., Odoni, A., 2003. Airport Systems Planning, Design, and Management. McGraw Hill, New York, NY.
Deshpande, V., Arıkan, M., 2012. The impact of airline flight schedules on flight delays. Manuf. Serv. Oper. Manage. 14 (3), 423–440.
Diana, T., 2013. An application of survival and frailty analysis to the study of taxi-out time: A case of New York Kennedy Airport. J. Air Transport Manage. 26, 40–43.

Diana, T., 2010. Can delay propagation be predicted? A case study of the three largest New York area airports using neural networks. J. Airport Manage. 4 (2), 170–177.
Géron, A., 2017. Hands-On Machine Learning with SciKit-Learn & TensorFlow. O'Reilly, Sebastopol, CA.
Hebert, J.E., Dietz, D.C., 1997. Modeling and analysis of an airport departure process. J. Aircraft 34 (1), 43–47.
Horonjeff, R., McKelvey, F.X., Sproule, W.J., Young, S.B., 2010. Planning and Design of Airports. McGraw Hill, New York, NY.
Idris, H., Clarke, J.P., Bhuva, R., Kang, L., 2002. Queuing model for taxi-out time estimation. Air Traffic Control Quarterly 10 (1), 1–22.
Kim, M.S., 2016. Analysis of short-term forecasting for flight arrival time. J. Air Transport Manage. 52, 35–41.
Kistler, M., Gupta, G., 2009. Relationship between airport efficiency and surface traffic. In: 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS). p. 7078.
Lee, H., Malik, W., Jung, Y.C., 2016. Taxi-out time prediction for departures at Charlotte airport using machine learning techniques. In: 16th AIAA Aviation Technology, Integration, and Operations Conference. p. 3910.
Lordan, O., Sallan, J.M., Valenzuela-Arroyo, M., 2016. Forecasting of taxi times: The case of Barcelona-El Prat airport. J. Air Transport Manage. 56, 118–122.
Massaron, L., Boschetti, A., 2016. Regression Analysis with Python. Packt Publishing, Birmingham, UK.
Mayer, C., Sinai, T., 2003. Network effects, congestion externalities, and air traffic delays: Or why not all delays are evil. Am. Econ. Rev. 93 (4), 1194–1215.
Pujet, N., Delcaire, B., Feron, E., 2000. Input-output modeling and control of the departure process of busy airports. Air Traffic Control Quart. 8 (1), 1–32.
Raschka, S., Mirjalili, V., 2017. Python Machine Learning, second ed. Packt Publishing, Birmingham, UK.
Ravizza, S., Chen, J., Atkin, J.A., Stewart, P., Burke, E.K., 2014. Aircraft taxi time prediction: comparisons and insights. Appl. Soft Comput. 14, 397–406.
Rebollo, J.J., Balakrishnan, H., 2012. A network-based model for predicting air traffic delays. In: 5th International Conference on Research in Air Transportation. pp. 22–25.
Sammut, C., Webb, G.I. (Eds.), 2011. Encyclopedia of machine learning. Springer Science & Business Media.
Shumsky, R.A., 1997. Real-time forecasts of aircraft departure queues. Air Traffic Control Quart. 5 (4), 281–308.
Simaiakis, I., Balakrishnan, H., 2009. Queuing models of airport departure processes for emissions reduction. In: AIAA Guidance, Navigation, and Control Conference. p. 5650.
Srivastava, A., 2011. Improving departure taxi time predictions using ASDE-X surveillance data. In: Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th, IEEE. pp. 2B5-1.
Sutton, C.D., 2005. 11-classification and regression trees, bagging, and boosting. Handbook Statist. 24, 303–329.
Tu, Y., Ball, M.O., Jank, W.S., 2008. Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern. J. Am. Stat. Assoc. 103 (481), 112–125.
U.S. Department of Transportation, Federal Aviation Administration, Aviation System Performance Metrics, https://aspm.faa.gov.
Wang, Y.X., 2011. Prediction of weather impacted airport capacity using ensemble learning. NASA, retrieved at the following website on July 14, 2018. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140008304.pdf.
Xu, N., Sherry, L., Laskey, K.B., 2008. Multifactor model for predicting delays at US airports. Transp. Res. Rec. 2052 (1), 62–71.
Zhang, C., Ma, Y., 2012. Ensemble Machine Learning: Methods and Applications. Springer, New York, NY.