Applying Machine Learning to Solve Unexecuted Truckload Problem in e-Commerce Middle Mile Delivery

ABSTRACT

Problem definition: 17% of booked truckloads in the $940 billion U.S. trucking industry were cancelled during the execution stage, resulting in significant disruptions to trucking operations as well as massive recovery costs. However, little research has examined how to handle these cancelled truckloads, especially from a Make (i.e., using self-owned private fleet) VS Buy (i.e., using 3PL service) perspective. Therefore, our research aims to fill this void by collaborating with a U.S. e-Commerce company to offer solutions to this overlooked problem. ***Methodology/ results:*** We collect data of 3 million truckloads in 2020 from a leading U.S. e-commerce company and study 202,407 canceled truckloads in their middle mile delivery network where truckloads were transported from ports to warehouses and/or between warehouses. We employ three prevalent machine learning models (i.e., logistics regression, random forest, and XGBoost) on those canceled truckloads to predict the probability of each truckload being executed by its own fleet and by 3PL. We apply the best performing model *XGBoost + Sigmoid* to conduct a 6 week pilot run to determine the re-allocation of the cancelled truckloads between self-owned fleet and 3PL fleet, resulting in an average weekly cost savings of $0.5 million U.S. dollars. Our machine learning algorithm was subsequently implemented across the organization. ***Managerial implications:*** Our research contributes to transportation procurement literature by providing a machine learning approach to handle cancelled truckloads at the execution stage. We also contribute to e-commerce order fulfillment research by highlighting the largely omitted yet crucial leg of transportation – the middle mile delivery. For practitioners, our modeling approach can be readily applied or adapted to solve similar truckload cancelation problems for the 750,000 U.S. trucking companies to achieve significant cost savings.

1. INTRODUCTION

Covid-19 has fundamentally changed the way consumers shop, resulting in a significant increase in e-commerce orders (Gramling et al., 2021). For instance, since early 2021, 60% of the consumers indicated that they have visited brick-and-mortar stores less than before the pandemic, and 43% indicated that they shopped online more often for products that they would have previously bought in stores (Gramling et al. 2021). While this transition brings more sales for retailers, it also poses substantial supply chain challenges following the increase in online orders. In the wake of these supply chain challenges, on-time delivery stands out as the cornerstone of consumer experience (Gramling et al. 2021). Current consumers are increasingly demanding for on-time delivery more than ever such that more retailers (e.g., Amazon, Walmart, Target) have strengthened guaranteed delivery services for consumers' online orders (Barbee et al. 2021). However, surges and fluctuations in consumer demand, translated as surges and fluctuations in loads (Kamali and Wang 2021), have added additional complexity to truckload transportation, affecting retailer's ability to achieve guaranteed delivery services (Kamali and Wang 2021, Stalk and Mercier 2022).

Our current research, therefore, aims to solve truckload transportation issues to facilitate achieving guaranteed delivery services. The attention of achieving guaranteed delivery services in e-commerce has mainly been focused on last mile delivery (Lim et al. 2018, Mangiaracina et al. 2019), however, moving the right products to the right warehouses at the right time is the prerequisite for a successful last mile delivery. In our research, we primarily focus on how to efficiently transport products prior to the last mile delivery, such as from ports to warehouses and/or between warehouses. We term this leg of transportation middle mile delivery, which mainly involves truckload transportation. As John Kearney, CEO of Advanced Training Systems, commented: "In recent years, industry attention on e-commerce logistics has largely focused on last-mile delivery, middle-mile trucking, however, is emerging as perhaps the primary area of competition—and growth—in the U.S. consumer goods supply chain." (Insider 2020). Therefore, it is crucial to study this primary area of competition. The extant literature on middle mile delivery is predominantly focused on route optimization (Emadikhiav et al. 2020, de Vries et al. 2020, He et al. 2022) and assumes a perfect solution – all truckloads will be executed via route optimization algorithms (Acocella and Caplice 2023). However, we quickly realize the limitation of routing algorithms through collaboration with a prominent U.S. e-commerce company who transports a weekly average of 100,000 truckloads among their 200+ warehouses: no matter how advanced the routing algorithm iterates in this company, approximately about 6000 truckloads/week remain unexecuted due to various reasons. Although the percentage of unexecuted truckloads is small, the consequences are significant: Ali-Habib and Gonzalez (2018) found that the average cost to recover a cancelled load is $145 in the United States. Therefore, our

current study aims to solve a specific problem, faced by numerous retailers with guaranteed delivery services, yet draws little academic attention: *how to handle unexecuted truckloads in the middle mile?* More specifically, as almost all retailers use both third-party logistics (3PL) fleet and their own fleet to fulfill orders, we resolve the conventional *make-versus-buy predicament* to resolve two research questions: (1) As it is oftentimes inevitable for e-commerce firms to have unexecuted truckloads in their middle mile, how should the focal firm allocate those unexecuted truckloads between its in-house fleet (i.e., Make) and 3PL (i.e., Buy) to ensure maximum cost saving while still achieving guaranteed service levels? (2) What are the economic and operational benefits the focal firm can achieve by implementing such an allocation?

To answer the research questions, we collect truckload data from one of the leading e-commerce company for a 30-week period in 2020, during the period of which 3 million truckloads were transported by the focal company with 202,407 unexecuted at the first attempt. We employ three prevalent machine learning models (i.e., logistics regression, random forest, and XGBoost), train and calibrate them, and compare their performance. Subsequently, we utilize the best performing model to predict the probability of future truckloads being executed by the company's own fleet. We accordingly tender all model-predicted inexecutable truckloads to designated 3PLs for immediate execution while retaining those model-predicted executable truckloads in-house to be optimized and transported by the company's own fleet. During a six-week pilot run, our machine learning approach results in an average weekly cost savings of $455,084 (ranging from $199,814 to $918,731), compared to the existing practice of tendering all unexecuted truckloads to 3PLs.

Due to the significant cost savings, our machine learning solution was rolled out for a full-swing implementation in the focal company in early 2021 to continuously improve their operational efficiency as well as their bottom-line performance. Our analytical approach can also be readily applied for other companies in the retail industry. As firms in the retail industry are rushing to incorporate big data into their daily operations (Bean 2023), our study provides guidance regarding how e-commerce retailers can apply similar models to improve their operations efficiency. In addition, for the 750,000 U.S. trucking companies, truckload cancelation is an inevitable operations issue at a daily level (ATA 2023). Our modeling approach can also be readily applied for the whole trucking industry to tackle this problem to achieve cost savings for the trucking sector. Moreover, instead of designing custom-built algorithms due to strategic considerations (more in Section 10.1), we prove that off-the-shelf models can also yield significant cost savings. To this end, our off-the-shelf modeling approach may be a better or only solution for the 99.7% U.S. trucking companies that operate 100 or fewer trucks (ATA 2023) as these small and medium sized trucking companies may not afford the resources to develop custom-build algorithms at all. Accordingly,

we call for operation researchers to consider a wider range of generalizability and applicability in practice when conducting operations research.

2. CONTRIBUTION TO LITERATURE

Our research makes four major contributions to the related literature. First, we contribute to enrich the understanding of transpiration procurement literature. Transportation service procurement is considered more complicated than traditional physical product procurement due to its nonbinding contracts and the relatively weak legal enforcement for noncompliance (Acocella and Caplice 2023). Our research reveals a simultaneous two-tiered Make VS Buy decision in the focal company both at the strategic procurement stage (i.e., assigning tours to own fleet and 3PL fleet) and at the practical execution stage (i.e., assigning rejected loads to own fleet and 3PL fleet). In an in-depth literature review, Acocella and Caplice (2023) found that transportation service procurement is overwhelmingly focused on the strategic procurement stage while the actual execution stage has been largely ignored. Our research specifically tackles one key issue of rejected loads at the execution stage, contributing to the Make VS Buy literature in transportation procurement space at its execution stage from shipper's perspective, which has not been examined before.

Second, in transportation procurement literature where Make VS Buy decisions are studied, complete load acceptance by carriers is often assumed in previous studies to achieve feasible solutions (Emadikhiav et al. 2020, de Vries et al. 2020, He et al. 2022, Acocella and Caplice 2023). However, as is witnessed by the focal company, due to market demand uncertainty, service failure (i.e., rejected loads) can frequently occur and result in detrimental consequences to business, the issue of which is unfortunately often ignored in previous transportation procurement literature (Acocella and Caplice 2023). Notably, our study investigates this overlooked issue and applies machine learning to achieve a better business solution by both improving operations efficiency and yielding measurable cost savings. To this end, our study reminds researchers the importance of investigating service failures of transportation procurement at its execution stage.

Third, we contribute to the existing big data analytics literature in operations management. Although machine learning has been widely used to address big data analytics problems in this field, its application in the retail industry has been primarily limited to studying consumer welfare (Chan et al. 2016, Lau et al. 2018, Zhao 2019). Little attention has been given to how machine learning and big data analytics can be used to improve the operational welfare for retailers. Attempting to fill this gap, our research investigates how retailers can leverage machine learning to ensure their guaranteed delivery services in their end-to-end supply chain by focusing on one specific truckload planning activity in the middle mile

delivery. Our study highlights potential paths for future research to continue exploring the application of machine learning to improve retailer's operational welfare in other operations areas.

Fourth, our research makes incremental contributions to the e-commerce literature by reminding researchers the importance of middle mile delivery. There has been a significant amount of operations management research in the e-commerce industry. However, operations management research in the e-commerce industry either focuses on how e-commerce itself impacts operations and transportation management (Gunasekaran et al. 2002, Mokhtarian 2004, Rotem-Mindali and Weltevreden 2013, Lafkihi et al. 2019) or how retailers can improve last mile delivery (Lee and Whang 2001, Lim et al. 2018, Mangiaracina et al. 2019, Vakulenko et al. 2019). Last mile delivery research has indeed dominated e-commerce transportation research in the past two decades (c.f. the literature review of Mangiaracina et al. 2019). While last mile delivery is undoubtedly an important part of e-commerce operations, the primary competition has shifted to middle mile delivery (Insider 2020). Without the right products being transported to the corresponding warehouses at the right time, a successful last mile delivery cannot be guaranteed. Despite this critical role, research into middle mile delivery in the e-commerce industry is very limited, and no study has examined the middle mile truckload transportation problem in depth. Thus, our research addresses this gap by investigating the rejected load problem that frequently happens in the middle mile delivery yet receives little academic attention.

3. BACKGROUND INFORMATION

The focal company is a leading e-commerce company headquartered in the U.S. Over the years, the company has evolved into a global leading e-commerce platform and its online retail business has exploded during the Covid Pandemic, especially in the U.S. As a result, its U.S. delivery network has also expanded to such an unprecedented level that each week, there are about 100,000 53-foot-truck-loads being transported and delivered among its 200+ warehouses in the U.S. This part of transportation is what we term as middle mile delivery. Figure 1 below is a snapshot of a weekly shipment flow for the focal company in the U.S., showing a network of middle mile order fulfillment and replenishment in its Tier I warehouses (Note that all Figures and Tables in this manuscript are supplied as e-companion). The size of each node corresponds to the volume of shipments (inbound + outbound) in each warehouse. Currently, the focal company's middle mile network is divided into six different regions for the purpose of scheduling and route planning. Each region is represented by a different color. We see that apart from intra-region shipment flows, there are also tremendous inter-region shipment flows, indicating the dynamic and difficult-to-predict consumer demand.

[FIGURE 1]

How to plan these 100K truckloads efficiently so that they can be delivered to the right warehouses at the right time while achieving the minimum possible total cost is a great challenge involving complicated and large-scale resource planning and route optimization activities. The focal company rolled out its *weekly* automated-planning system in late 2019 in the U.S. Due to the sheer volume of its truckloads, the company utilizes both its own fleet and third-party fleets for its middle mile transportation. Software wise, the focal company manages everything in-house, i.e., all the planning and routing algorithms are written by its own software engineering and research teams to gain maximum flexibility to plan, schedule, and adjust all of the transportation activities. The overall weekly planning process for the 100K truckloads is a classic demand driven process illustrated in Figure 2. First, different demand scenarios (such as outbound shipments, inbound shipments, warehouse transfers, customer returns etc.) are generated from different databases. Then, each demand scenario is converted into potential total truckloads which will then be optimized into different routes using demand clusters (Figure 3). The last step is the execution stage where routes are either executed in-house or tendered to 3PLs. We next explain the three steps in detail.

[FIGURE 2]

First, based on its demand scenarios, the company starts its weekly planning process with resource planning. Resource in the focal company refers to its own fleet and drivers as well as the fleet and drivers from 3PLs. Since the focal company has more than 200 Tier I warehouses and many other smaller Tier II warehouses across the states, it is challenging to plan all resources at a country level. Therefore, resource planning for the whole U.S. is broken down into smaller chunks based on the concept of "demand clusters" (Figure 3). For example, the demand from all Tier II warehouses within a radius of 100 miles to the nearest Tier I warehouse are pooled together for resource planning purposes. The graph on the left in Figure 3 shows all Tier I warehouses. Each blue dot on the map represents a Tier I warehouse. The graph on the right in Figure 3 shows how demands from both Tier II and Tier I warehouses were pooled together into demand clusters based on their geographical proximity.

[FIGURE 3]

After different demands are pulled together into clusters, optimization algorithms will run multiple times till a near-optimal resource plan is found, i.e., the optimal usage between self-managed resources and 3PL resources. The output from resource planning is "resource blocks". For example, 8am to 8pm on Monday for the Greater New York City cluster is one resource block. 3PLs have access to all resource blocks via a dedicated platform, where 3PLs can choose to bid on the different resource blocks. Based on its optimization results, the company decides which resource blocks to award to which 3PLs and which

resource blocks to be managed internally using its own fleet. If awarded, a 3PL is committed to providing the required resources (i.e., trucks and drivers) for the total duration of the awarded blocks. Once the required resources for the coming week are committed, even if the company does not utilize the committed 3PL resources, the company is still going to pay for them, which leads to the second step – route planning.

A resource block provides a "blocked" time window for both the focal company and 3PL as a generic reference. How many truckloads to be allocated in each specific block time window is determined by the route planning process, which also runs on a rolling weekly basis to plan for the coming week. Route planning utilizes all the committed resources, both internally and externally, in each cluster to maximize the utilization of the committed resources, especially 3PL resources. The assumption here is that any committed resources within each block at each cluster, whether self-owned or provided by a 3PL, are available to be deployed anytime for any demand scenario during the planning horizon. Route planning runs another set of route optimization algorithms on a weekly basis given the available resources, including own resources and 3PL resources. The output from route planning is to create as many closed-loop tours (i.e., A→ B → C → D → A) as possible for the whole coming week.

After the tours are created by the route planning algorithm, the tours using 3PL resources will be shared with 3PLs but the final execution of those tours may still change. Between the creation of the tours and the final execution of those tours, the demand in the transportation network may change significantly, such as spikes or plunges in consumer purchases, resulting in additional ad hoc truckloads or cancelled tours. In addition, due to the unbalanced demand in different regions, route planning will inevitably create some open-loop tours (i.e., trailers do not return to its starting warehouse) with less efficiency. All these lead to the third step in the planning process – final tour assignment or adjustment. Tour assignment will run its own optimization algorithm at fixed intervals on a *daily* basis by taking into account near-real time uncertainties and changes of demand in the network up until the minimum cutoff time for pickup, when the tours will finally be assigned to each committed resource for execution.

4. THE PROBLEM AND THE AS-IS BUSINESS SOLUTION

In an ideal state, resource planning, route optimization, and the final tour assignment follow a highly integrated process and will be decoupled seamlessly so that least amount of disruption and variability occurs in the final execution stage. However, due to constantly changing consumer demand, a hard-to-predict market situation, and uncontrollable 3PL resources, each week there are on average 6000 truckloads that are left unexecuted for two primary reasons. First, a small amount of truckloads (either executed by 3PLs or own fleets) are cancelled for various reasons, such as long-haul lanes that drivers are not willing to travel.

Second, a tiny portion of the truckloads are not captured by the company's three-step planning process. For example, dependencies in the planning system may not have been updated to include newly opened lanes between newly opened warehouses.

The unexecuted 6000 truckloads in each week are not static, i.e., each week may witness different unexecuted truckloads on different lanes. The 6000 unexecuted truckloads represent only about 5% of total weekly volume but the impact to the company's business is severe. First, as these truckloads are destinated to each warehouse to be further dispatched in the last mile delivery, if these truckloads were not executed (i.e., delivered to the required warehouses), the last mile delivery will be severely impacted, especially given the focal company's 2-day-delivery service of thousands of its products for its members. If products were not delivered on time, consumers will most likely cancel the purchase, buy from somewhere else, or return the product after a delayed delivery due to potential frustration. This negatively impacts customer satisfaction and customer repurchase behavior and decreases customer loyalty in the long run, in addition to incurring unnecessary return costs. Second, the impact to daily operations is also cumbersome. This is because the tour assignment algorithm maximizes the assignment of truckloads as closed-loop tours (A→ B → C → D → A). If one leg (B → C) in a closed-loop tour was cancelled, the remaining legs of the tour will be impacted, either resulting in cancellation of all the remaining legs or requiring excessive efforts to allocate new resources to execute the remaining legs. If the remaining legs of a tour are cancelled, this will further impact guaranteed delivery service. If new resources were allocated attempting to rescue the remaining legs, this will require extra work in operations that we discuss below.

The as-is solution to solve the problem is to assign a dedicated team to consolidate all the unexecuted truckloads on a weekly basis. Similar to the weekly automated-planning system, this dedicated team will run a manual planning process each week using a separate set of optimization algorithm to generates new tours for all unexecuted truckloads pooling available resources, such as number of drivers and available resource blocks. These newly generated tours can either be kept in-house using its own fleet to execute or tendered to 3PLs for immediate execution. Both approaches have been attempted by the company. If kept in-house, these new tours will become part of step 3 (tour assignment) in the three-step automated-planning process and will be fine-tuned for execution (if necessary) on a daily basis. The downside of this approach is that among these manually generated tours, some legs continue to get rejected or cancelled for various reasons. Those cancelled truckloads will again go through the next round of the manual planning process until the unexecuted truckloads can finally be executed, which may take days and sometimes weeks to complete. In addition, some urgent truckloads have to be dealt with daily at load-by-load basis, the daily fire-fighting process also exhausts the employees, resulting in inefficient operations

and low morale. The surge in online orders during Covid exacerbates this approach. The second approach is that after the manual planning process generates new tours for all the unexecuted truckloads, all the newly generated tours are posted immediately on a spot-market platform where a huge pool of 3PLs can bid and arrange their soonest possible resources for execution with minimum delay. This approach was heavily preferred during Covid before we started our project. Since this process falls out of the three-step automated-planning process where long term contract rates were signed, the tours coming out of the manual process will be treated as ad hoc shipments whereas no fixed contract rates are available between the company and 3PLs. As a result, the company has to pay a very high premium for 3PLs to execute these loads (sometimes 5 times higher than normal contract rates). The dramatic increase in cost to handle these unexecuted truckloads via 3PLs was further exacerbated during Covid when there was a nationwide shortage of truck drivers and trucks.

Neither of these two approaches is perfect and there exists a trade-off between the two. To solve both the operational efficiency issue (in-house) and the high cost issue (3PLs), the company requires a more scientific approach to answer the following questions: 1) Does the company need to keep all the 6000 truckloads in-house or does the company need to tender all 6000 truckloads on the spot market? 2) If the answer is no to either of the previous questions, then, how many truckloads should be kept in-house and how many should be tendered on the spot market? 3) Assume there are significant cost savings if not all 6000 truckloads are sent to the spot market, a) what are the cost savings and b) how can the company ensure that these kept-in-house truckloads can be executed successfully, without the risk of being cancelled again so that its guaranteed service level will not be impacted?

To answer these questions, especially the second question, we utilize three popular machine learning models (i.e., random forest, XGBoost, and logistics regression), compare their model performance, and select the best performing model to predict which truckloads to send to the spot market and which truckloads to remain in the company's automated-planning system. The rationale is very simple: if a truckload has a very high probability of not getting executed by the retailer's own fleet, then this truckload will be sent directly to the spot market; otherwise, it should remain in-house to be absorbed by its own route optimization process. Note that our research does not aim to solve any resource planning, scheduling, or route optimization problem as these problems have already been solved by the focal firm's in-house algorithms. We simply solve the classic Make VS Buy decision in the context of unexecuted truckload transportation/planning.

5. MODEL DESIGN

To achieve cost savings as well as to minimize operational disruption, the company requires an ex-ante solution to predict that among the weekly 6000 unexecuted truckloads, which truckloads have a higher probability of being executed by its own fleet, and which truckloads are not likely to be executed by its own fleet. Once identified, these truckloads can be handled more efficiently, i.e., if a truckload has a high probability of being executed by its own fleet, then the company will keep the truckload in its three-step automated-planning process (i.e., to be assigned in step 3 for execution). Otherwise, the company will send the truckload to the spot-market platform so a 3PL can execute it immediately. This way, the company both improves daily operational efficiency and avoids paying high spot-purchase premiums.

To predict which truckloads have a high probability of being executed by its own fleet, several key questions need to be further addressed. First, what are the variables or truckload-related features we can use to build the predictive models? Second, what are the available models we can use and which model performs the best? Third, once a probability is identified for each truckload, what is the threshold we should use to decide for self-execution and for sending to the spot market? To answer these detailed questions, we use the popular scikit-learn tool (Pedregosa et al. 2011) to build and train our models. A pseudo algorithm (to comply with the confidentiality agreement) for the modeling process flow is presented in Table 1.

[TABLE 1]

Data Description, Data Filtering, and Data Split

The focal company's automated-planning system was rolled out in late 2019 and stabilized in mid-2020 after a few rounds of fine-tuning processes. Therefore, we choose our data starting point to be week 20 of 2020 when the automated-planning system was stabilized with satisfying performance. This project was initiated in week 48 in 2020. Accordingly, we choose our data ending point to be week 49 in 2020. More specifically, our data starts from 2020-06-15 and ends at 2020-12-02. Altogether, we have 30 weeks' data consisting of about 3 million truckloads. All the data used in the current study was queried using SQL to pull from different data warehouses. Data cleaning and modeling in the current study were both conducted in Python version 3.9.

To predict if an unexecuted truckload has a high probability of being executed by its own fleet, we first need to look at the historic performance of executed versus unexecuted for all of the unexecuted truckloads in the entire 30 weeks. Among the 3 million truckloads in the 30 weeks, there are altogether 202,407 unexecuted truckloads that went through daily manual planning process and were retained internally for further execution. Since these 202,407 truckloads have been fed into one or multiple planning cycles, we were able to verify whether each truckload was eventually executed by its own fleet or not. We

assign 1 for those truckloads that have *not* been eventually executed by its own fleet and 0 otherwise. Following machine learning literature and practice, we split our data into train, validate, and test datasets. To match the focal company's planning system that runs on a weekly basis, we elect to keep approximately the last week's observations (2020-11-23 to 2020-12-02) as our test dataset. We build our model using data from 2020-06-15 to 2020-11-22, among which we split between train and validate datasets in an 80:20 ratio. We train our model on the training dataset and validate our model performance on the validation dataset and finally apply the model to the test dataset. Table 2 below summarizes the details in each dataset. Unexecuted (1) and executed (0) is also the outcome variable in our current study.

[TABLE 2]

Predictors/Features

To predict the probability of being unexecuted or executed, we use a set of predictors/features that are highly related to individual lane level as a tour is broken down at individual lane level. In a closed loop tour A→ B → C → D → A, A→ B is an individual lane, B → C is also an individual lane, and so on and so forth. We focus on individual lane level as the cancellations and operations disruptions occur only at individual lane level. Note that our unit of analysis is not at the closed-loop tour level. Rather, our unit of analysis is at the individual lane level. The outcome is to identify specific truckloads on specific lanes, either allocate to 3PLs for immediate execution or retain in-house to be optimized as tours with other truckloads.

Time to Departure (hours): This is the lead time to departure at the origin warehouse, i.e., how much time was allocated for drivers at each origin to handle necessary administrative tasks. If drivers were not assigned enough time to handle the necessary administrative tasks, there might be a delay at the start of the trip which might cause subsequent delays and potentially lead to disruption and cancellation.

Departure time at origin: this is the departure time assigned by the route planning algorithm for each lane. We delineate this variable into three levels: time of the day, day of the week, and week of the year to control for potential departure time bias. For example, late night departures might get a higher cancellation rate by drivers due to the unpreferable time.

Length of Check-in (minutes): the length of time allocated for drivers to check-in at the origin of each lane.

Length of hook trailers (minutes): the length of time allocated for drivers to hook trailers at the origin of each lane. For example, in a closed-loop tour A→ B → C → D → A, at the individual lane B → C, how much time was allocated for drivers to hook trailers at warehouse B. This variable matters as a busy

warehouse in New York City and a remote warehouse in Iowa may need totally different time to hook trailers.

Length of drop trailers (minutes): the length of time allocated for drivers to drop trailers at the end of each lane. The logic is the same as the length of hook trailers.

The logic to include the previous three variables is the same: if the allowed check-in time, trailer-drop, and trailer-hook time is not sufficient while drivers are measured by the stringent KPIs of this company, drivers may just cancel the job to avoid penalty on their personal KPIs knowing that they are not going to complete the job on-time.

Transit time (minutes): transit time between origin and destination for each lane. Transit time for each lane is assigned by the company using meta data as a reference to measure driver performance. Similar to other variables, if drivers see the transit time is challenging/unreasonable to achieve on certain lanes due to certain conditions, they may cancel the job to avoid penalty on personal KPIs.

Length of lane (miles): the length between origin and destination of each lane is generated by Google maps. Too short lanes might get canceled due to less economic gains while too long lanes might also get cancelled due to excessive driving hours. Therefore, we include this variable.

Number of resource blocks: the resource planning stage allocates all usable resources (i.e., available trucks and drivers) into different blocks of time windows in different clusters and different regions. With more resource blocks, the tour assignment algorithm will have more flexibility to assign lanes to different resource blocks, i.e., even if lanes get cancelled in some tours, the algorithm can still pick up those cancelled lanes and fit them into other tours using other resource blocks. However, if the number of available resource blocks are limited, the algorithm may not be able to fit those cancelled lanes into other tours because there are no resource blocks available. Hence, the more resource blocks available, the less probability of lanes get cancelled.

Total duration of resource blocks (minutes): the total duration of all resource blocks created by resource planning. “Number of resource blocks” controls for the amount of available resource blocks. “Total duration of resource blocks” further controls for the time length of resource blocks.

Origin zip code: the origin warehouse zip code where a lane starts.

Destination zip code: the destination warehouse zip code where a lane ends.

The origin and destination warehouse zip codes are included to control for potential location bias. For example, if a warehouse is located in a very remote area, drivers may cancel the job due to potential travelling issues.

Class Balancing

In machine learning, class balancing is the process of training a model on a dataset where the classes are not evenly represented. This can be done by oversampling the minority class or undersampling the majority class. The accuracy and performance of machine learning models tend to be better when the dataset has evenly distributed classes, i.e., balanced dataset. Most machine learning algorithms actually operate under the assumption of balanced data. However, among the total 153,814 truckloads in our train dataset, 99,331 (64.6%) truckloads were unexecuted (labeled as 1) and 54,483 (35.4%) truckloads were executed (labeled as 0). Therefore, the class of our outcome variable is unbalanced. Before we start building and training our models, we first need to deal with the imbalanced train data.

To balance the class distribution in our imbalanced data, we adopt a few popular resampling techniques following the extant machine learning literature where we examine both undersampling, oversampling, and a combination of undersampling and oversampling: SMOTE (Chawla et al. 2002, Han et al. 2005, Fernández et al. 2018), SMOTENN (Batista et al. 2004), SMOTETomek (Batista et al. 2003), and Near Miss (Winston 1970). SMOTE is a technique involving oversampling the minority class to make it more representative of the entire dataset. This can be done by either duplicating minority samples or by generating new samples using a Synthetic Minority Oversampling Technique. SMOTENN differs from SMOTE in that when resampling the data, the minority class is over-sampled while the majority class is under-sampled. SMOTETomek is a combination of the SMOTE and Tomek Link algorithms. In SMOTETomek, SMOTE creates synthetic data points that are similar to existing data points, while Tomek Link removes data points that are too close to each other. Near Miss basically adopts an oversampling technique but focuses on selecting samples from the minority class that are close to the boundary between the classes. Both undersampling and oversampling can be effective methods for dealing with imbalanced data in machine learning (Chawla et al. 2002, Liu et al. 2008).

Running the four sampling algorithms, we see that all four algorithms successfully balanced the class distribution in the outcome variable (Table 3). Figure 4 shows the class distribution before sampling and Figure 5 shows the class distribution after sampling using the four sampling techniques. We observe that SMOTE and SMOTETomek balanced the class distribution as exactly equal to each other (i.e., the sample of 0s and 1s are the same) while SMOTENN and Near Miss balanced the class distribution to a roughly equal split. We also observe that only SMOTE kept the 1s intact as 99,331 while all the other three algorithms ether slightly reduced or significantly reduced the 1s in our data. To maximize utilization of the observation of 1s, i.e., those unexecuted truckloads that were not able to be executed by the company's own fleet, we elect to use the balanced data generated by SMOTE for the next step model building.

[TABLE 3]

[FIGURE 4]

[FIGURE 5]

6. MODEL BUILDING

With the train data balanced, we start building models. As our outcome variable is a binary variable, we consider three popular models: random forest, XGBoost, and logistic regression. Random forest is a classic machine learning algorithm that is used for classification and regression tasks. The algorithm is an ensemble technique that is used to create a forest of trees, where each tree is a decision tree. The final prediction is made by taking the average of all the predictions from the individual trees. Random forecast has been applied in operations management to solve different tasks (Bertsimas et al. 2016, Merrick et al. 2022). However, random forecast models can be very computationally intensive, as they require large amounts of data to be processed. XGBoost is a newer model that uses gradient boosting to improve the performance of decision trees. XGBoost implements machine learning algorithms under the Gradient Boosting framework and provides a parallel tree boosting that solves problems in a fast and accurate way. In operations management, XGBoost has been shown to be effective in optimizing different types of operations management problems (Chuang et al. 2021, Merrick et al. 2022). Logistic regression is a classic statistical technique that has been widely used in operations management (Gopalakrishnan et al. 2022). Given the popularity of the three different models in the operations management field, we apply all the three models on our train data, compare their model performance, and select the best performing model for our prediction.

Hyperparameter Tuning and Grid Search

Hyperparameter tuning optimizes a machine learning model by tweaking its hyperparameters, the settings that control the behavior of a machine learning algorithm. Grid search is a hyperparameter tuning technique that involves systematically testing different combinations of hyperparameters to find the combination that results in the best performance for the model. To get the best performance of random forecast and XGBoost models, we first conduct hyperparameter tuning and grid search (the best grid search score for random forest is –0.13 and the best grid search score for XGBoost is –0.09). The best parameters derived from this process were then used to finalize the random forecast and XGBoost models for the next step model training. Logistic regression does not involve much hyperparameter tuning and grid search per se, we report the output from running logistic regression in Table 4 below.

[TABLE 4]

Model Performance Comparison

After tuning random forest and XGBoost models, we check the model performance for the tuned models on all the three datasets: train, validation, and test. Following machine learning practices, we adopt the following four measures to test our model performance: Accuracy, Brier score, Log-loss score, and AUC. We briefly summarize the definition and reason why we include these four measures in Table 5.

[TABLE 5]

We apply the four measures on our three datasets and report the model performance in Table 6. We also report Receiver Operating Characteristic curve and its associated AUC value for the three models on the three datasets in Figure 6. We see that XGBoost is the best performing model in terms of all the four measures in all the three datasets.

[TABLE 6]

[FIGURE 6]

Confusion matrix is another popular visualization tool in machine learning to quickly evaluate model performance. We also report the confusion matrices for the three models on the three datasets in Figure 7. A confusion matrix is a table that describes the performance of a classification model (or "classifier") on a set of data for which the true values are known. The matrix is N×N, where N is the number of classes. The first dimension is the actual class and the second dimension is the predicted class. We only have two classes in our data so our confusion matrix is a 2×2 matrix.

[FIGURE 7]

Model Performance on the Test Data

In this section, we specifically focus on elaborating the model performance on the test dataset of the three different models, as model performance on the test dataset more accurately reflects how the model will perform in predicting future unseen data. In addition to the measures we reported in the previous section, we also report Precision, Recall, and F1-Score on the test data for the three models in Table 7. From these additional three measures, we also see that XGBoost performs the best.

[TABLE 7]

Plotting the predicted probabilities of unexecuted and executed truckloads by the three models on the test dataset presents a clearer picture. Figure 8, Figure 9, and Figure 10 plot the predicted probabilities against actual data for Random Forest, XGBoost, and Logistic Regression respectively. We see from the plots that both Random Forest and XGBoost peaks at both 0 and 1 probability. However, Random Forest classifier also peaks in the range of 0.5 – 0.9 probability while XGBoost predicts significantly less probabilities out of 0 and 1. Logistic regression performs the worst among the three models with the peak probability prediction at 0.5 – 0.6 rather than at 0 and 1. Logistic regression normally demonstrates good

predictive accuracy in operations management (Gopalakrishnan et al. 2022). However, its predictive power can be affected by non-linear relationships between predictors and outcome variables. Logistic regression in our case performs the worst. To explore the potential reasons behind the difference in predictive probabilities among the three models in our test data, we next examine feature importance on the test data.

[FIGURE 8]

[FIGURE 9]

[FIGURE 10]

7. FEATURE IMPORTANCE ON TEST DATA

A Partial Dependence Plot (PDP) is a graphical tool to visualize the dependence of an outcome variable on predictors (Goldstein et al. 2015). PDPs reveal the marginal effect one or two predictors have on the predicted outcome of a machine learning model (Friedman 2001, Freitas 2014). PDPs can show if the relationship between predictors and the outcome variable is linear, monotonic, or more complex. PDPs are a useful tool for interpreting machine learning models and can help reveal if a model is relying on a single variable as well as to identify which variables are the most important predictors of the outcome variable (Friedman 2001, Hastie et al. 2009). If the PDP shows a U-shaped curve, this indicates a nonlinear relationship between the predictor and outcome variable. To identify which variables are the most important predictors for the outcome variable, we look at the magnitude of the outcome variable's changes as the predictor is varied. The larger the changes in the outcome variable, the more important the predictor is.

Figure 11 shows PDPs of all the predictors on test data. We see that some variables demonstrate equal or similar importance for each of the three models, such as Departure: Hour of Day – Sine, Departure: Hour of Day – Cosine, Departure Time: Day of Week, and Time to departure. We also observe that XGBoost captures a nonlinear relationship between some predictors and the outcome variable while logistic regression fails to capture the nonlinear relationship, such as Hook Trailer Time and Drop Trailer Time. While both XGBoost and Random Forest share a similar relationship between some predictors and the outcome variable, logistic regression shows a completely different relationship for the same predictors, such as Total Block Time and Number of Available Blocks. Lastly, XGBoost and Random Forest also demonstrate differences in capturing the relationships between predictors and the outcome variable, such as Departure: Hour of Day, Hook Trailer Time, and Drop Trailer Time. In sum, based on the PDPs, it seems that the reason that XGBoost performs the best while logistic regression performs the worst is that XGBoost was able to capture more nonlinear relationships that both random forest and logistic regression models failed to capture.

[FIGURE 11]

To further quantify the importance of each predictor, we use Permutation Feature Importance algorithm in scikit-learn on test data. Permutation feature importance is calculated by shuffling the values of a feature and comparing how much the model's predictions degrade. Figure 12, Figure 13, and Figure 14 report the feature importance of the three models on the test data to demonstrate how the three models handle each predictor differently. We observe that among the three models, only logistic regression has four negative features (Miles, Transit Time, Number of Available Blocks, and Destination Zip). The features with negative weights indicate that these four predictors do not contribute to the predicting power of logistic regression models and can be removed from the model. While the feature of Destination Zip shows a negative weight for logistic regression, it is ranked as the 6th most important feature for random forest and the 5th most important feature for XGBoost. Comparing random forest with XGBoost, we see that the top ranked features are similar but their rankings are different, such as Time to Departure, Check-in Time Window, and Total Block Time.

[FIGURE 12]

[FIGURE 13]

[FIGURE 14]

We check feature importance using different features both at the beginning stage of the project (i.e., to get the model started) and on a regular weekly basis thereafter (i.e., to ensure no important variables are missing with the network dynamics frequently changing). We do see that with time elapsing and the model retrained, the ranking of the importance of features changes over time, which also changes the best model we can select to predict future week's data.

8. MODEL CALIBRATION

Based on the analysis we conducted so far, XGBoost is the best performing model among the three. Before concluding that XGBoost is the best model to use to predict future week's data, we calibrate the three models using train data and apply the calibrated models on test data to verify XGBoost is indeed the best model we can use even after model calibration. We adopt two popular approaches to calibrate our models: Sigmoid and Isotonic.

Sigmoid model calibration is a technique used in machine learning to effectively improve the accuracy of model predictions by accounting for errors in the data which the model was trained on. The goal of sigmoid model calibration is to make the model's predictions more accurate on new or unseen data. Though initially popular in support vector machine models (Platt 1999), Sigmoid has been extended to

other various models. Similar to Sigmoid method, Isotonic method (Niculescu-Mizil and Caruana 2005) can also be used to improve the prediction performance of a machine learning model on new or unseen data. Isotonic method is based on the assumption that the model is a function of its input data, and that unseen values can be more accurately predicted by calibrating the model with the known data.

We again utilize Scikit-learn package to calibrate the three models. Models were first calibrated on the train data. Then, the calibrated models were applied on the test data to compare model performance as test data will be a more accurate reflection of how the calibrated models perform for future week's data. Figure 15, Figure 16, and Figure 17 report the probability calibration curves on train data for random forest, XGBoost, and logistic regression respectively. Table 8 reports calibrated model performance on the test data. Looking at the four different measures in Table 8, we observe that both Sigmoid and Isotonic calibration significantly improved the model performance for random forest measured by Log-loss score. Both Sigmoid and Isotonic calibration also significantly improved the model performance for XGBoost measured by Brier and Log-loss score. However, neither Sigmoid nor Isotonic calibration are able to improve the model performance of logistic regression. Based on the calibrated model performance comparison, we can finally conclude that *XGBoost + Sigmoid* is the best performing model and is accordingly selected in the final step to predict future week's data.

[FIGURE 15]

[FIGURE 16]

[FIGURE 17]

[TABLE 8]

9. MODEL EVALUATION

Since our research is a practice-based research engaging with a real operation problem (Gallien et al. 2016), we evaluation our model from two key perspectives following the advice from Gallien et al. (2016): generalizability and validity. Generalizability refers to "the extent to which the research question considered is of interest to a large number of practitioners" while validity is "the extent to which research results and predictions are well-founded and apply effectively to real-world operations" (Gallien et al. 2016, p.7).

9.1 Generalizability

For a practice-based research, generalizability mainly refers to how a larger number of practitioners will be interested in the topic. We discuss generalizability from the following three perspectives. First, our study only has access to truckload data from a single leading e-commerce player. The whole trucking industry, on the other hand, is a $940 billion dollar business in the U.S. (ATA 2023). Load cancellation, though

overlooked in the literature, represents significant costs to the trucking industry. Ali-Habib and Gonzalez (2018) found that 17% of booked loads were cancelled and the average cost to recover a cancelled load is $145. Given the $940 billion industry size and 17% cancel rate, we reckon that our approach to handle cancelled loads can be readily applied to the whole trucking industry to reduce the percentage of cancelled loads and, therefore, help yield significant cost savings for practitioners in this industry. Second, our predictive modeling approach can also be extended to solve transportation procurement problems and other operations problems beyond just cancelled loads. Specifically, our modeling approach can be extended to the strategic Make VS Buy decision stage to decide how many loads to award to which suppliers by building demand uncertainty into the model in contrast to the traditional optimization approach where demand uncertainty is assumed away (Acocella and Caplice 2023). Third, for the 99.7% small and medium sized enterprises (SMEs) in trucking industry who operate less than 100 trucks and may not have the required resources to develop custom-built algorithms to improve operational efficiency, our step-by-step modeling approach of using off-the-shelf models can be readily applied or adapted by these SMEs to solve daily operations problems. As is witnessed from our focal company, the expected cost savings from off-the-shelf models are significant without necessitating the extra time and resources to engage in developing custom-built algorithms, the result of which may not even justify the extra time and resources (more in Section 10.1). To this end, we call for operations researchers to consider the ease of application of custom-built algorithms so that a larger number of companies and practitioners can benefit from our research.

9.2 Validity

Gallien et al. (2016) proposed several approaches to test the validity of practice-based research, such as testing the model's predictive accuracy, reporting of implementation, and reporting of measurable benefits from the implementation. We discuss these perspectives in this section.

Implementation – Testing the Statistical Significance

Before we roll out the implementation, we first test the statistical significance of our proposed approach. Working with our counterpart in the focal company and using propensity score matching, we identified 13 warehouses to be used to test the statistical significance of our model (i.e., the treatment group) and 13 warehouses as the control group. Names of the warehouses are not reported for confidentiality concerns. As our focal company competes on speed and did not allow extra time for implementation, we tested our machine learning approach in a 24-hour window in the 13 treatment warehouses on December 8th, 2020. To draw causal inference, we adopt regression discontinuity design (RDD), a popular design frequently used to draw causal inference for treatment effects (Coviello et al. 2018, Calvo et al. 2019).

The statistical inference for RDD is as follows: among the 26 warehouses, we assume that each site has two potential outcomes in term of cancelled loads upon implementation, Yi(1) and Yi(0). If site *i* receives the treatment (impacted by implementation), we will observe Yi(1), and Yi(0) will remain unobserved. Similarly, if site *i* receives the control condition (unimpacted by the implementation), we will observe Yi(0) but not Yi(1). This logic forms the fundamental logic of causal inference in RDD. The observed outcome, therefore, is:

$$Y_i = (1 - T_i)Y_i(0) + T_iY_i(1) = \begin{cases} Yi(0) & if\ Xi < c \\ Yi(1) & if\ Xi \geq c \end{cases}$$

The effect upon implementation is thus formally defined as:

$$T \equiv \mathbb{E}[\, Y_i(1) - Y_i(0) | X_i = c]$$

From the mathematical formula, we see that the effect *T* recovered by the RDD is the average effect upon implementation for the 13 sites "local" to the cutoff *c* on 2020-12-08, i.e., for those sites with score values Xi = c. It answers the following question: what would be the average outcome change (average number of cancelled loads) for these 13 sites with Xi = c (on 2020-12-08) if we switched their status from control (before the implementation) to treated (after the implementation)?

Using an optimal and data-driven bandwidth selection method (robust bias correction), Figure 18 plots the graph of the estimation result ($T = -71.40$, $p = 0.064$), meaning that upon the implementation, for these 13 sties in the treatment group, the average number of cancelled loads was reduced by 71 with statistical significance.

[Figure 18]

Implementation – Testing Model Predictive Accuracy in Real Life

After confirming the statistical significance of our approach, next we roll out a three week's pilot run to test the model's predictive accuracy in real life operations. Based on the calibrated model performance on test data, we select XGBoost + Sigmoid for pilot run. Note that XGBoost + Sigmoid is not always the best model for every week after project implementation as the model training process is conducted every week using a moving time window to prevent the model from becoming stale. As a result, the best performing model also changes from week to week depending on the market situation and truckload-specific features.

Our initial data cutoff point is week 49 in 2020. Therefore, we first predict the planning activity for week 50. In week 50, to avoid potential disruptions to the network, we pilot run the model using a small portion of the unexecuted truckloads. Our pilot run consists of 1058 truckloads that are supposed to be sent to 3PLs. Applying the calibrated model (XGBoost + Sigmoid) on these 1058 truckloads, the model predicts 812 truckloads as 1 and 246 truckloads as 0, with 1 being unexecuted and 0 being executed by the

company's own fleet. Recall the decision logic is to send all model-predicted unexecuted truckloads to 3PLs for immediate execution. For those 812 truckloads sent to spot market for 3PLs to execute, we are confident that 3PLs will strive to implement them successfully and we can expect almost a 100% execution rate as this is why spot-purchase exists in the first place. But for the 246 truckloads that are predicted as executed by the company's own fleet, another two questions arise: 1) Will all these 246 truckloads actually be executed by the company's own fleet? 2) Can we withhold more truckloads for the company's own fleet to save costs as 3PLs charge higher spot rates for these ad hoc shipments? Less truckloads to 3PLs means more truckloads to be absorbed by its own fleet. If we increase the truckloads to the company's automated-planning system and most of these truckloads can actually be executed by its own fleet, this will be beneficial to the business. However, if those truckloads switched from 3PL cannot be executed by the company's own fleet, those unexecuted truckloads will have to go through another manual process again, creating the same daily operations chaos. This will be against the initial purpose of using predictive modeling to ease the daily workload.

To answer these two questions, a trial was conducted for three weeks. In addition to predicting a binary outcome of 1 and 0, the XGB model also predicts the probability of each truckload being executed and unexecuted. We therefore use the predicted probability of being unexecuted to test the above-mentioned trade-off issue. For week 50, the decision is to send all model predicted unexecuted truckloads (812 labeled with 1) to 3PLs and let the remaining 246 be absorbed by the daily-planning cycle. This is equivalent to sending all truckloads with an unexecuted probability of higher than 50% to 3PLs (i.e., all truckloads with an unexecuted probability higher than 50% are labeled as 1 by XGB model). When the weekly planning cycle was completed for the entire week 50, we were able to verify how many of those 246 truckloads in week 50 left to be planned by the automated-planning system were actually executed by the company's own fleet. When predicting for week 51, we increased the cutoff threshold to 60%, i.e., all truckloads predicted to have an unexecuted probability higher than 60% will be sent to 3PLs and the rest retained for the daily-planning cycle. We then verify the actual executed rate for week 51 when the week 51 planning cycle was completed. For week 52, we further increased the cutoff threshold to 70%, i.e., all truckloads predicted to have an unexecuted probability higher than 70% will be sent to 3PLs and the rest retained for the daily-planning process. We then check the actual executed rate for week 52 when its planning cycle was completed. Table 9 below summarizes the cut-off threshold analysis.

[Table 9]

From Table 9, we see that setting the probability cut off at 50% maximizes the actual executed rate (65.85%). Higher cutoff threshold (i.e., more truckloads retained for the self-planning system) actually

lowers the execution rate. We attribute this to the fact that the in-house planning system's algorithm is not ready to absorb those cancelled truckloads yet because these cancelled truckloads have very different characteristics compared with other normal truckloads the planning algorithm handles. Based on the three week's trial, 50% cutoff was used since week 53. The cutoff threshold fine-tuning practice repeats itself to avoid becoming stale. We do see that the cutoff threshold changes when the network situation has changed dramatically, such as increase of shipment volume during peak seasons or decrease of shipment volume in non-peak seasons. We did not test a cutoff threshold less than 50% in our initial threshold analysis but we tested less-than-50% threshold in the subsequent weeks and the execution rate is not significantly better than the cutoff of 50%.

Implementation – Measurable Benefits

In addition to easing operational disruption, the most important business goal for implementing the predictive modeling for truckload planning is to save costs, as using 3PLs to handle the ad hoc shipments are much more expensive than handling them in-house by the company's automated-planning system. To verify the cost savings, we use the external rates provided by 3PLs and the internal rates provided by the finance department to compare the costs for those withheld truckloads following the three weeks' trial. Table 10 shows the pilot run cost savings for the first 6 weeks in 2021 (indicated by the row of "Net Gain"). Depending on different truckload characteristics in each week, we see that the cost savings per week range from $200K up to $850K. Due to the significant amount of cost savings in the pilot run period, the predictive modeling solution was rolled out in February 2021 in full swing to be implemented across different departments which accordingly updated their respective dependencies to support the rollout.

[TABLE 10]

10. KEY LEARNINGS AND FUTURE RESEARCH

10.1 Key Learnings – the Law of Diminishing Returns

The focal company is a leading e-Commerce company that attracts the best talents to work for it. Despite having a large team of research scientists and data scientists in its operations department, the majority of the operations problems are solved using off-the-shelf models. In collaborating with this department to solve the cancelled truckload problem, we initially proposed a full-scale custom-built algorithms to handle cancelled truckload. However, this is what our counterpart responded:

"Custom-built algorithms for us are only intended for fundamental infrastructural problems that lay the foundation of our business operations as we all know it takes a long time to build, finetune, and implement custom-built algorithms...As the most fast-paced online retailer in this industry, our business cannot afford to do everything custom-built...Besides, there was a time when we were doing A/B test

between custom-built algorithms and off-the-shelf models...and unfortunately, the extra benefit from custom-built algorithms are almost negligible...especially when you consider how much more time and resources we have to dedicate to custom-built algorithms....we just cannot afford wasting the extra time and resources for the tiny gains...In addition, we are not looking for a perfect solution to everything...we are looking for continuous improvement and quick wins... this is why if you come to work for us for a while, you might say that most of our processes are like glued together by duck tapes...oh, yes, we compete by speed and by moving fast, not by perfection..."

The quote from our counterpart reveals two important lessons. First, industry moves much faster than academia (Gallien et al. 2016). While academia can afford multiple rounds of reviews to publish custom-built algorithms, industry normally apply whatever models are available to achieve fast wins. Speed matters for our focal company. In addition, custom-built algorithms are doable for our focal company who attracts PhDs from top universities. For other small and medium sized businesses that do not have access to top talents, custom-built algorithms may not be an option at all, in which case off-the-shelf models simply serve a better business purpose. To this end, we reiterate the importance of applying off-the-shelf models in industry as this might be the best available option for a vast majority of companies and practitioners. Second, the focal company's choice reflects the law of diminishing returns in operations management (Schmenner and Swink 1998). The focal company chooses to trade off the potential extra savings brought by custom-built algorithms with off-the-shelf models as the focal company does not believe the extra time and effort is worthy. In other words, the extra savings requires much more company resources whereas these extra resources only yield less significant savings. To this end, we echo Gallien et al. (2016) in that as a field, we need to rethink the concept of highly valuable operations research, i.e., if, in practice, custom-built algorithms cannot help yield more savings/benefits for practitioners, how can we really justify the importance and significance of our research?

10.2 Future Research

First, due to the limited timeframe given to implement this project, we were not able to design custom-built algorithms and compare the cost savings between custom-built algorithms and off-the-shelf models. We, therefore, call for researchers who conduct practice-based research to compare the cost-savings between custom-built algorithms and off-the-shelf models, if time and resources allow. As is evident from the focal company's quote, custom-built algorithms failed to yield significant savings compared with off-the-shelf models for the focal company. To this end, it is both interesting and important to conduct this comparison to reveal the dynamics that affect the performance of custom-built algorithms and a further analysis of the trade-offs between custom-built algorithms and off-the-shelf models will provide deeper insights to

operations issues to better improve operations efficiency as well as the bottom-line performance. Second, an efficient middle mile plays a critical role in the overall order fulfillment and delivery performance. To maintain competitive pricing and healthy margins, cost-saving in the middle mile delivery represents a sizable opportunity. Poor middle mile performance not only disrupts the whole end-to-end delivery cycle but also ultimately affect consumer shopping experience. Therefore, we also call for more middle-mile research to continue knowledge accumulation in this space.

In sum, this study addresses a real business problem by gathering data to develop a better solution and subsequently achieves significant measurable cost savings as well as improved operational efficiency. This study provides insights into the underlying dynamics of cancelled truckloads and the methodology used can be readily applied or adapted to a wide range of practice problems to help practitioners improve operational efficiency.

REFERENCES

Abrahams AS, Fan W, Wang GA, Zhang Z, Jiao J (2015) An integrated text analytic framework for product defect discovery. *Production Oper. Management*. 24(6):975-990.

Acocella A, Caplice C (2023) Research on truckload transportation procurement: A review, framework, and future research agenda. *J. Business Logist*. 44:228–256.

Al-Habib A, Gonzalez FN (2018) Predicting carrier load cancellation. MIT Dissertation.

ATA American Trucking Associations (2023) https://www.trucking.org/economics-and-industry-data.

Bansal P, Gualandris J, Kim N (2020) Theorizing supply chains with qualitative big data and topic modeling. *J. Supply Chain Manag*. 56(2):7-18.

Barbee J, Dubeauclard R, Jensen K, Spielvogel J (2021) Creating a competitive edge in omnichannel grocery fulfillment. *McKinsey*. https://www.mckinsey.com/industries/retail/our-insights/creating-a-competitive-edge-in-omnichannel-grocery-fulfillment.

Batista GE, Bazzan AL, Monard MC (2003) Balancing training data for automated annotation of keywords: A case study. In WOB:10-18.

Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*. 6(1):20-29.

Bean R (2023) Has progress on data, analytics, and AI stalled at your company? *Harv. Bus. Rev*. https://hbr.org/2023/01/has-progress-on-data-analytics-and-ai-stalled-at-your-company.

Bertsimas D, Kallus N, Hussain A (2016) Inventory management in the era of big data. *Production Oper. Management*. 25(12):2006-2009.

Boone T, Ganeshan R, Hicks RL, Sanders NR (2018) Can Google trends improve your sales forecast? *Production Oper. Management*. 27(10):1770-1774.

Calvo E, Cui R, Serpa JC (2019) Oversight and efficiency in public projects: A regression discontinuity analysis. *Management Sci*. 65(12):5651-5675.

Chan HK, Wang X, Lacka E, Zhang M (2016) A mixed-method approach to extracting the value of social media data. *Production Oper. Management*. 25(3):568-583.

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res*. 16:321-357.

Choi TM, Wallace SW, Wang Y (2018) Big data analytics in operations management. *Production Oper. Management*. 27(10):1868-1883.

Chuang HHC, Chou YC, Oliva R (2021) Cross-item learning for volatile demand forecasting: An intervention with predictive analytics. *J. Oper. Management*. 67(7):828-852.

Corbett CJ (2018) How sustainable is big data?. *Production Oper. Management*. 27 (9):1685-1695.

Coviello D, Guglielmo A, Spagnolo G (2018) The effect of discretion on procurement performance. *Management Sci.* 64(2):715-738.

De Vries H, Van Wassenhove LN (2020) Do optimization models for humanitarian operations need a paradigm shift?. *Production Oper. Management*. 29(1):55-61.

Emadikhiav M, Bergman D, Day R (2020) Consistent routing and scheduling with simultaneous pickups and deliveries. *Production Oper. Management*. 29(8):1937-1955.

Fernández A, García S, Galar M, Prati RC, Krawczyk B, Herrera F (2018) *Learning from imbalanced data sets*. Cham: Springer.

Fisher M, Raman A (2018) Using data and big data in retailing. *Production Oper. Management*. 27(9):1665-1669.

Foster K, Penninti P, Shang J, Kekre S, Hegde GG, Venkat A (2018) Leveraging big data to balance new key performance indicators in emergency physician management networks. *Production Oper. Management*. 27(10):1795-1815.

Freitas AA (2014) Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*. 15(1):1-10.

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann. Stat*. 1189-1232.

Gallien J, Graves SC, Scheller-Wolf A (2016) OM Forum—Practice-based research in operations management: What it is, why do it, related challenges, and how to overcome them. *Manufacturing Service Oper. Management*. 18(1):5-14.

Goes PB (2014) Editor's comments: Big data and IS research. *MIS Q*. 38(3):iii–viii.

Goldstein A, Kapelner A, Bleich J, Pitkin E (2015) Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat*. 24(1):44-65.

Gopalakrishnan M, Zhang H, Zhang Z (2022) Multiproduct pricing under the multinomial logit model with local network effects. *Decision Sci*. 1–20.

Gramling K, Orschell J, Chernoff J (2021) How e-commerce fits into retail's post-pandemic future. *Harv. Bus. Rev.* https://hbr.org/2021/05/how-e-commerce-fits-into-retails-post-pandemic-future.

Guha S, Kumar S (2018) Emergence of big data research in operations management, information systems, and healthcare: Past contributions and future roadmap. *Production Oper. Management*. 27(9):1724-1735.

Gunasekaran A, Marri HB, McGaughey RE, Nebhwani MD (2002) E-commerce and its impact on operations management. *Int. J. Prod. Econ*. 75(1-2):185-197.

Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In Advances in Intelligent Computing: International Conference on Intelligent Computing, Proceedings, Part I 1: 878-887. Springer Berlin Heidelberg.

Hastie T, Tibshirani R, Friedman JH (2009) *The elements of statistical learning: data mining, inference, and prediction.* 2:1-758. New York: Springer.

He L, Liu S, Shen ZJM (2022) Smart urban transport and logistics: A business analytics perspective. *Production Oper. Management*. 31(10):3771-3787.

Hopkins J, Hawking P (2018) Big data analytics and IoT in logistics: A case study. *Int. J. Logist. Manag*. 29(2):575-591.

Insider 2020. Middle-Mile Logistics Play Evolving Role in U.S. Consumer Supply Chain. https://markets.businessinsider.com/news/stocks/middle-mile-logistics-play-evolving-role-in-u-s-consumer-supply-chain-1029329166.

Kamali P, Wang A (2021) Longer Delivery Times Reflect Supply Chain Disruptions. *International Monetary Fund*. https://www.imf.org/en/Blogs/Articles/2021/10/25/longer-delivery-times-reflect-supply-chain-disruptions.

Ko DG, Mai F, Shan Z, Zhang D (2019) Operational efficiency and patient-centered health care: A view from online physician reviews. *J. Oper. Management*. 65(4):353-379.

Lafkihi M, Pan S, Ballot E (2019) Freight transportation service procurement: A literature review and future research opportunities in omnichannel E-commerce. *Transp. Res. E Logist. Transp. Rev*. 125:348-365.

Lam HK, Yeung AC, Cheng TE (2016) The impact of firms' social media initiatives on operational efficiency and innovativeness. *J. Oper. Management*. 47:28-43.

Lau RYK, Zhang W, Xu W (2018) Parallel aspect-oriented sentiment analysis for sales forecasting with big data. *Production Oper. Management*. 27(10):1775-1794.

Lee HL, Whang S (2001) Winning the last mile of e-commerce. *MIT Sloan Manag. Rev.*

Lim S, Yim D, Khuntia J, Tanniru M (2020) A Continuous-Time Markov Chain Model–Based Business Analytics Approach for Estimating Patient Transition States in Online Health Infomediary. *Decision Sci*. 51(1):181-208.

Lim SFW, Jin X, Srai JS (2018) Consumer-driven e-commerce: A literature review, design framework, and research agenda on last-mile logistics models. *Int. J. Phys. Distrib. Logist. Manag*. 48(3):308-332.

Liu XY, Wu J, Zhou ZH (2008) Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst.. Man Cybern. B Cybern*. 39(2):539-550.

Mangiaracina R, Perego A, Seghezzi A, Tumino A (2019) Innovative solutions to increase last-mile delivery efficiency in B2C e-commerce: a literature review. *Int. J. Phys. Distrib. Logist. Manag.* 49(9):901-920

Merrick JR, Dorsey CA, Wang B, Grabowski M, Harrald JR (2022) Measuring prediction accuracy in a maritime accident warning system. *Production Oper. Management*. 31(2):819-827.

Mokhtarian PL (2004) A conceptual analysis of the transportation impacts of B2C e-commerce. *Transportation*. 31:257-284.

Nenova Z, Shang J (2022_a) Chronic disease progression prediction: Leveraging case-based reasoning and big data analytics. *Production Oper. Management*. 31(1):259-280.

Nenova Z. Shang J (2022_b) Personalized Chronic Disease Follow-Up Appointments: Risk-Stratified Care Through Big Data. *Production Oper. Management*. 31(2):583-606.

Niculescu-Mizil A, Caruana R (2005) Predicting good probabilities with supervised learning. In Proceedings of the 22nd international conference on Machine learning. 625-632.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O., ... Duchesnay E (2011) Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res*. 12:2825-2830.

Platt J (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classifiers*. 10(3):61-74.

Rotem-Mindali O, Weltevreden JW (2013) Transport effects of e-commerce: What can be learned after years of research?. *Transportation*. 40:867-885.

Sanders NR, Ganeshan R (2018) Big data in supply chain management. *Production Oper. Management*. 27(10):1745-1748.

Schmenner RW, Swink ML (1998) On theory in operations management. *J. Oper. Management*. 17(1):97-113.

Smyth KB, Croxton KL, Franklin R, Knemeyer AM (2018) Thirsty in an ocean of data? Pitfalls and practical strategies when partnering with industry on big data supply chain research. *J. Bus. Logist*. 39(3):203-219.

Stalk G, Mercier P (2022) Today's supply-chain fluctuations require systemic solutions. *Harv. Bus. Rev*. https://hbr.org/2022/10/todays-supply-chain-fluctuations-require-systemic-solutions.

Swaminathan JM (2018) Big data analytics for rapid, impactful, sustained, and efficient (RISE) humanitarian operations. *Production Oper. Management*. 27(9):1696-1700.

Vakulenko Y, Shams P, Hellström D, Hjort K (2019) Service innovation in e-commerce last mile delivery: Mapping the e-customer journey. *J. Bus. Res*. 101:461-468.

Winston PH (1970) Learning structural descriptions from examples. Ph.D. Thesis

Zhao S (2021) Thumb Up or Down? A Text-Mining Approach of Understanding Consumers through Reviews. *Decision Sci*. 52(3):699-719.

Zhu X, Ninh A, Zhao H, Liu Z (2021) Demand forecasting with supply-chain information and machine learning: Evidence in the pharmaceutical industry. *Production Oper. Management*. 30(9):3231-3252.

E-Companion Tables and Figures

Table 1 Pseudo Algorithm for Solving Unexecuted Truckloads

Step 1: Begin
Step 2: Update the starting and ending dates of training data using a sliding time window to avoid stale model performance
Step 3: Pre-processing test data
1. Start with the assumption of non-missing features and non-missing transit times in test data. Both conditions set to False. This step yields how many lanes have missing features and/or missing transit times.
2. Query for missing features from the planning system using truckload ID, change the condition for missing features as True, rerun 1.
3. Change condition for missing transit times as True, read in a static transit-time table, rerun 1.
4. Check how many loads are still missing features and transit times, exclude from modeling process.

Step 4: Split data into train, validation, and test datasets for modeling
Step 5: Select between different sampling algorithms
Run SMOTENN
Run Near Miss
Run SMOTETomek
Run SMOTE
Step 6: Parameter tuning for different models to find the best parameters for each model
Tune XGB parameters
Tune random forecast parameters
Run Logistics regression
Step 7: Define and collect model measures for the outputs from Step 6
Step 8: Model selection based on model measures from Step 7
Step 9: Model prediction – predict probability on the test data
Step 10: End

Table 2 Data Summary

	Train (80% of data from 2020-06-15 to 2020-11-22)	Validate (20% of data from 2020-06-15 to 2020-11-22)	Test (2020-11-23 and 2020-12-02)	Total
No. of Observations	153,814	38,454	10,139	202,407
0 in y	54,483	24,955	5,144	84,582
1 in y	99,331	13,499	2,995	115,825

Table 3 Under Sampling and Over Sampling for Train Data

	Train Data Before Sampling	SMOTE	SMOTENN	SMOTETomek	Near Miss
No. of Observations	153,814	198,662	110,001	182,774	108,240
0 in y	54,483	99,331	58,906	91,387	54,483
1 in y	99,331	99,331	51,095	91,387	53,757

Table 4 Logistic Regression Output

	coefficient	s.e.	z	P>|z|
Departure: Hour of Day-Sin	-0.073	0.019	-3.85	0.000
Departure: Hour of Day-Cos	0.220	0.021	10.45	0.000
Departure: Hour of Day	0.026	0.001	34.41	0.000
Departure: Day of Week	0.121	0.003	45.69	0.000
Departure: Week of Year	-0.017	0.000	-33.78	0.000
Hook Trailer Time	0.001	0.000	1.74	0.081
Drop Trailer Time	0.010	0.000	27.36	0.000
Transit Time	0.002	0.000	25.18	0.000
Miles	-0.001	0.000	-14.97	0.000
Check-in Time Window	0.000	0.000	6.04	0.000
Total Block Time	0.000	0.000	17.50	0.000
Number of Available Blocks	-0.015	0.000	-48.86	0.000
Origin Zip	0.001	0.000	37.76	0.000
Destination Zip	0.000	0.000	-15.41	0.000
Time to departure	-0.006	0.000	-88.63	0.000

Table 5 Four Model Performance Measures

Model Performance Measures Used	Definition	Rationale of using this performance measure
Accuracy	Measure of how well a model performs on a dataset, defines as the proportion of correct predictions made by the model out of all the predictions made.	Accuracy is appropriate for balanced data. We have a balanced data by using the SMOTE, accuracy is a proper performance measure given our data structure.
Brier score	Measures the mean squared error between the predicted probabilities and the true outcomes. The lower the Brier score, the better the predictions.	The Brier score has several appealing properties. First, it is a proper scoring rule, meaning that it is always optimal to make predictions that maximize the Brier score. Second, the Brier score is closely related to the log-loss, another popular metric for evaluating probabilistic predictions. In fact, the Brier score is simply the mean squared error of the log-loss. Third, the Brier score can be decomposed into a component that measures the calibration of the predictions and a component that measures the sharpness of the predictions. This decomposition is useful for understanding the sources of error in probabilistic predictions.
Log-loss	Log-loss is a measure of the accuracy of classifier predictions. The log-loss measure penalizes false negatives more heavily than false positives, thus, is commonly used in binary classification problems where the goal is to identify the positive class. A lower log-loss value means better predictions.	The Log-loss has several advantages. First, it is differentiable, which makes it easier to optimize. Second, the log-loss function is convex, which means that there is only one global minimum. The log-loss function is also scale-invariant, meaning that it is not affected by changes in the scale of the data, especially for logistic regression where data needs to be scaled sometimes. Lastly, the log-loss function is robust to outliers, meaning that it is not affected by a few extreme values. Given these advantages of log-loss measure, we include it as our third measure.
Area under the curve (AUC)	AUC represents the probability that a model will correctly classify a positive instance as positive and a negative instance as negative. The AUC ranges from 0 to 1, with a higher AUC indicating a better model.	The AUC has several desirable properties. First, the AUC is scale-invariant, meaning that it is not affected by changes in the distribution of the data. Second, the AUC is insensitive to the class imbalance, meaning that it is not affected by changes in the proportion of positive and negative instances. Third, the AUC is invariant to the specific choice of threshold, meaning that it can be used to compare models with different threshold values. Therefore, we include AUC as our fourth measure.

Table 6 Model Performance Comparison

	Random Forest	XGB Classifier	Logistic Regression
Training Data			
Accuracy Score	0.892	0.930	0.645
Brier Score	0.089	0.054	0.215
Log-loss Score	0.301	0.186	0.616
AUC	0.962	0.983	0.711
Validation Data			
Accuracy Score	0.827	0.869	0.643
Brier Score	0.125	0.095	0.217
Log-loss Score	0.394	0.311	0.622
AUC	0.901	0.936	0.711
Test Data			
Accuracy Score	0.799	0.837	0.605
Brier Score	0.143	0.116	0.227
Log-loss Score	0.439	0.375	0.638
AUC	0.883	0.915	0.679

Table 7 Precision, Recall, and F-1 Score for the Test Data

	Precision			Recall			F1-Score			Total Observation
	Random Forest	XGB	Logistic Regression	Random Forest	XGB	Logistic Regression	Random Forest	XGB	Logistic Regression	
1	0.82	0.86	0.66	0.81	0.84	0.55	0.81	0.85	0.60	1144
0	0.78	0.82	0.56	0.79	0.84	0.67	0.79	0.83	0.61	995
Weighted Average	0.80	0.84	0.61	0.80	0.84	0.60	0.80	0.84	0.60	2139

Table 8 Calibrated Model Performance on Test Data

	Random Forest	XGB Classifier	Logistic Regression
Test Data			
Accuracy Score	0.799	0.837	0.605
Brier Score	0.143	0.116	0.227
Log-loss Score	0.439	0.375	0.638
AUC	0.883	0.915	0.679
Model Calibration on Test Data			
Sigmoid			
Accuracy Score	0.793	0.835	0.613
Brier Score	0.142	0.124	0.225
Log-loss Score	0.447	0.418	0.634
AUC	0.883	0.915	0.679
Isotonic			
Accuracy Score	0.800	0.837	0.598
Brier Score	0.141	0.120	0.224
Log-loss Score	0.570	0.469	0.631
AUC	0.883	0.914	0.679

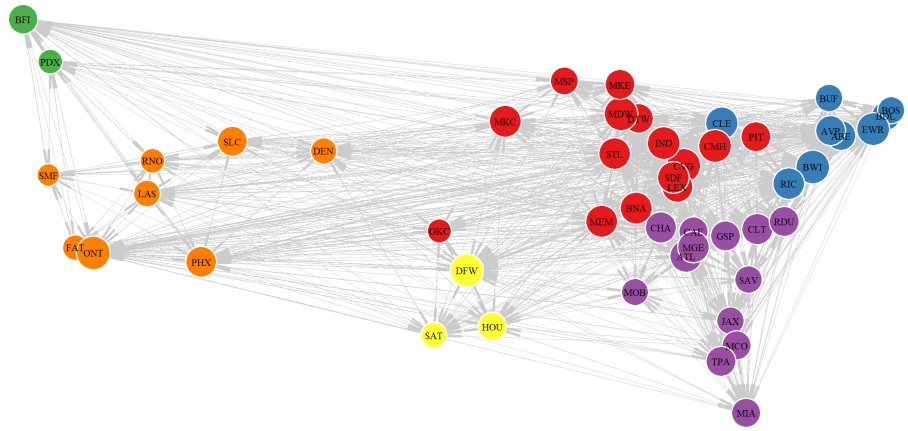
Table 9 Model Predictive Accuracy – A Cut-off Analysis in 2020

	Week 50 (50% cutoff)	Week 51(60% cutoff)	Week 52(70% cutoff)
Truckloads Withheld for Self-planning System	246	334	421
Actually being Executed by Self-own Fleet	162	139	207
Execution Rate	65.85%	58.38%	50.83%

Table 10 Cost Savings from Pilot Run Weeks in 2021

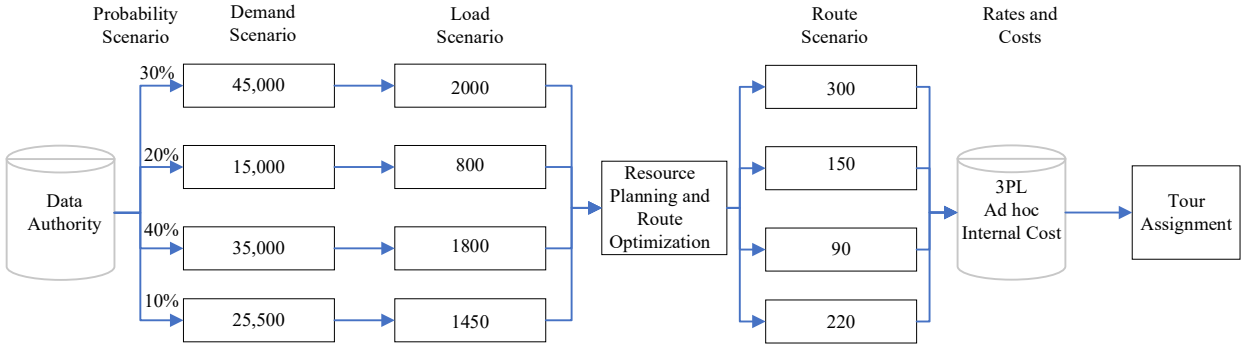
	W1	W2	W3	W4	W5	W6
# Truckloads Piloted	1,158	2,944	2,838	1,842	3,750	1,993
Cut-off Threshold	50%	50%	50%	30%	40%	40%
# Truckloads Retained	407	724	942	699	1321	551
# Truckloads Executed	195	267	655	293	688	202
# Truckloads Unexecuted	212	457	287	406	633	349
Mean Distance of Truckloads Executed	762	831	818	801	849	860
Mean Distance of Truckloads Unexecuted	790	883	792	729	778	842
Premium paid to 3PL for Unexecuted	$ 28,185	$ 169,614	$ 60,159	$ 94,623	$ 140,113	$ 53,839
Savings from Executed Truckloads	$ 247,807	$ 369,428	$ 978,890	$ 389,677	$ 986,973	$ 304,258
Net Gain	$ 219,622	$ 199,814	$ 918,731	$ 295,055	$ 846,861	$ 250,419

Figure 1 Shipment Flow of Order fulfillment and Replenishment



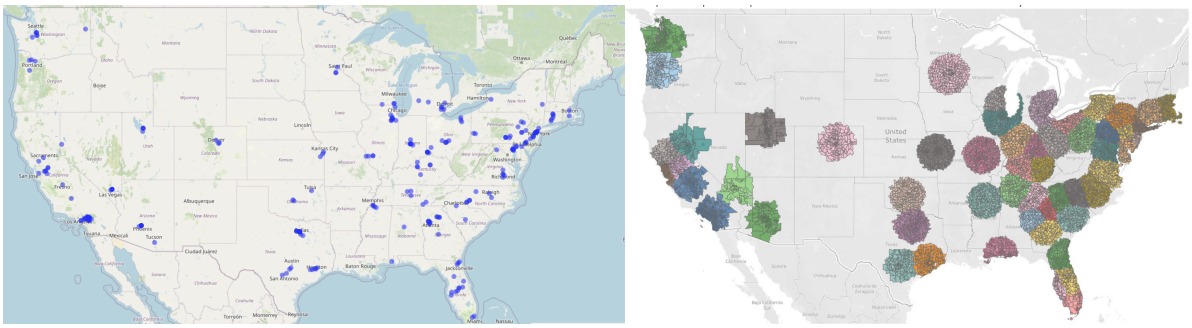
Note: Figure 1 presents an average weekly truckload flow among different warehouses of the focal company. Each warehouse is represented by a single node. The size of each node corresponds to the volume of shipments (inbound + outbound) in each warehouse. Different color represents different regions. Nodes with the same color belong to the same region. There are altogether six regions for the focal company.

Figure 2 Demand Driven Planning Process (For Illustration Purpose)



Note: Figure 2 demonstrates how demand flows through the resource planning process. First, different demand scenarios (such as outbound, inbound, warehouse transfer, etc.) are generated from database. Then, each demand scenario is converted into possible truckloads. Third, these truckloads will be optimized into routing using demand clusters in Figure 2. The last step is the tendering process where set of routes are tendered to 3PLs for execution.

Figure 3 Resource Planning Based on Demand Clusters



Note: Figure 3 demonstrates the concept of "Demand Clusters". The graph on the left shows all Tier I warehouses. Each blue dot is a Tier I warehouse. The graph on the right shows how demands from both Tier II and Tier I warehouses were clustered together based on their geographical proximity.

Figure 4 Class Distribution Before Sampling for Train Data

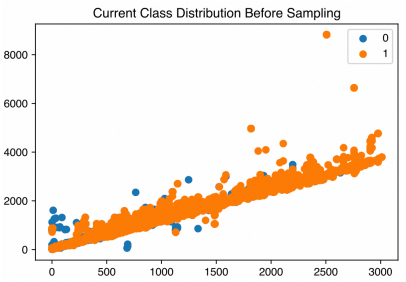


Figure 5 Class Distribution After Sampling for Train Data

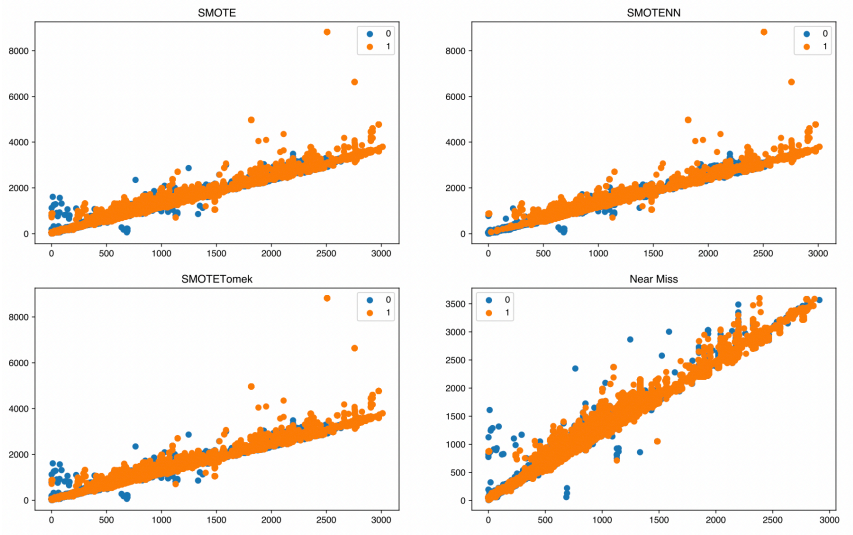


Figure 6 ROC Curve of the Three Models

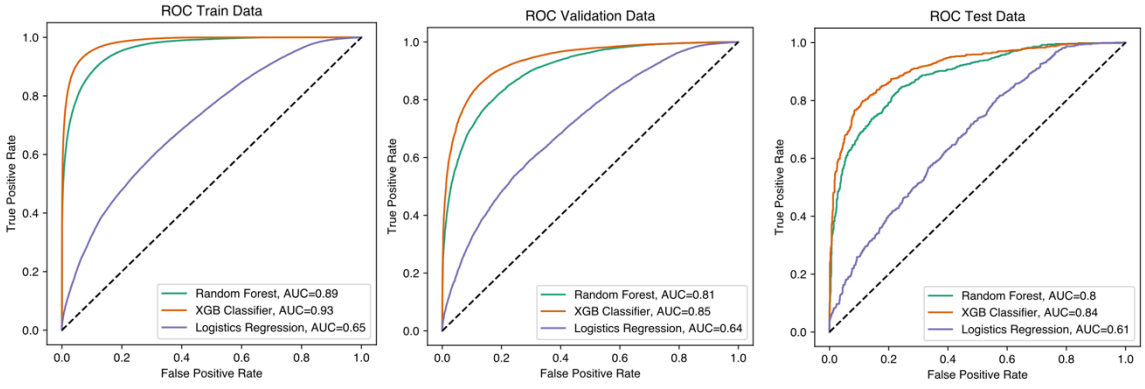


Figure 7 Confusion Matrix of the Three Models

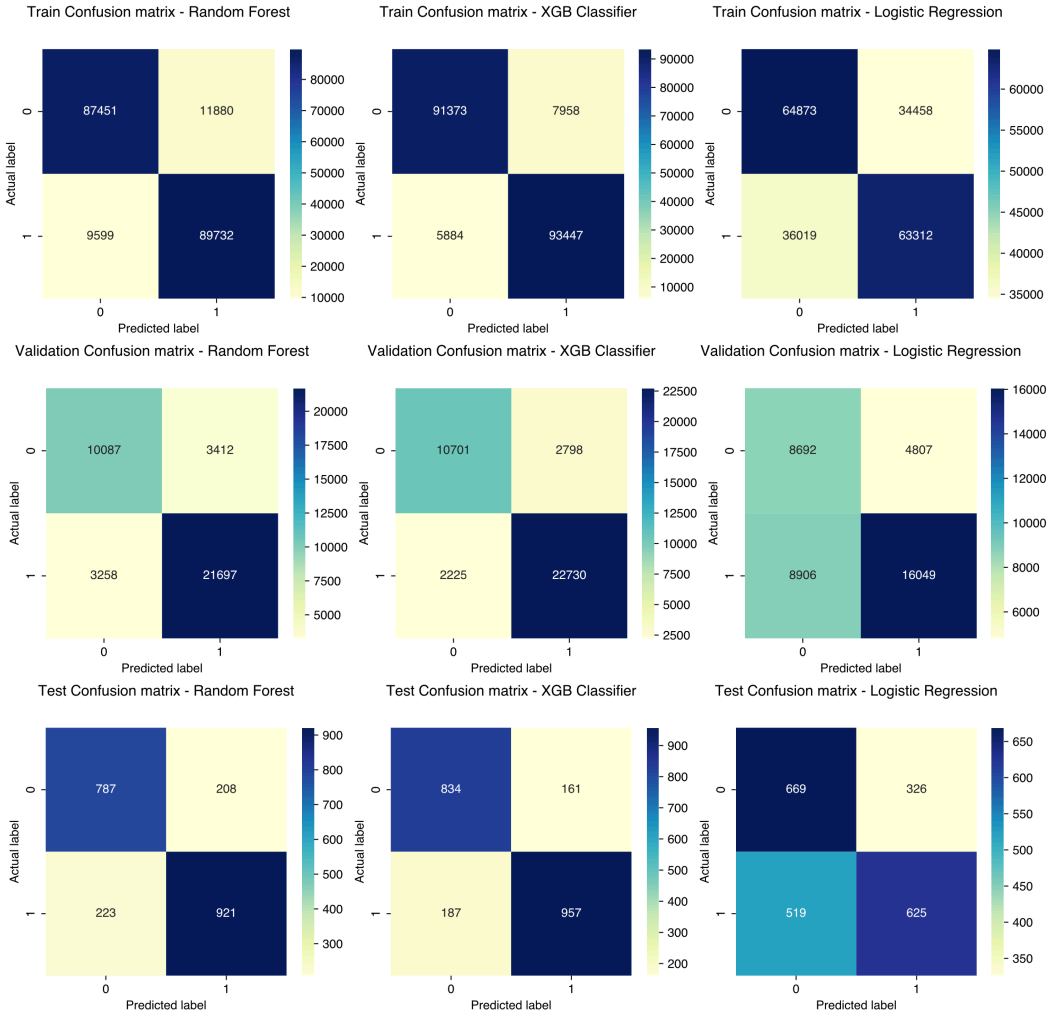


Figure 8 Random Forest Predicted Probability for Actual Executed VS Actual Unexecuted

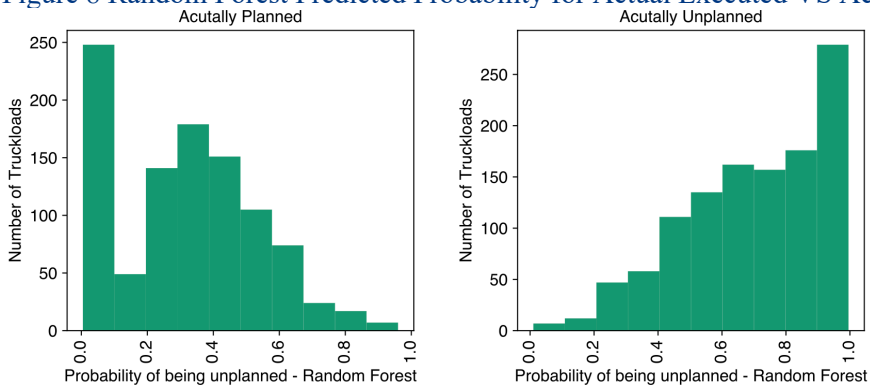


Figure 9 XGBoost Predicted Probability for Actual Executed VS Actual Unexecuted

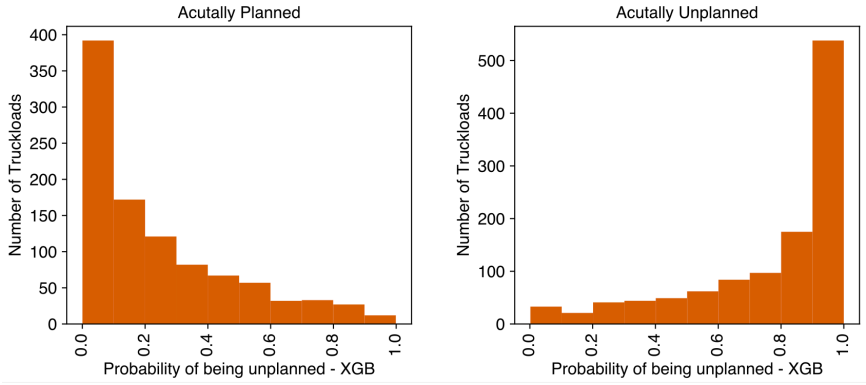


Figure 10 Logistic Regression Predicted Probability for Actual Executed VS Actual Unexecuted

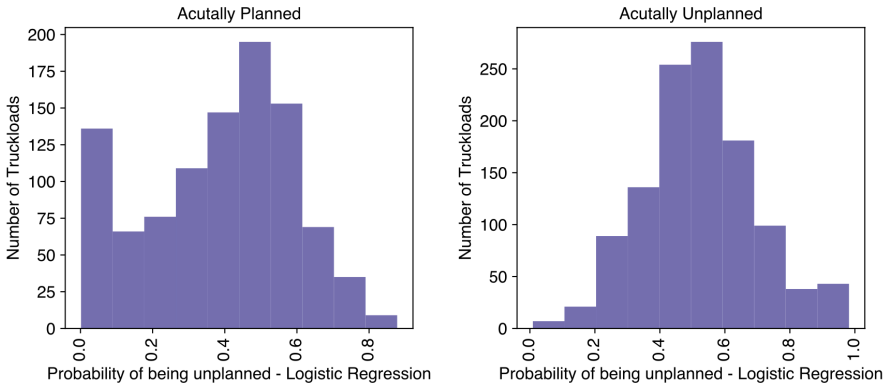


Figure 11 Partial Dependence Plot for Test Data

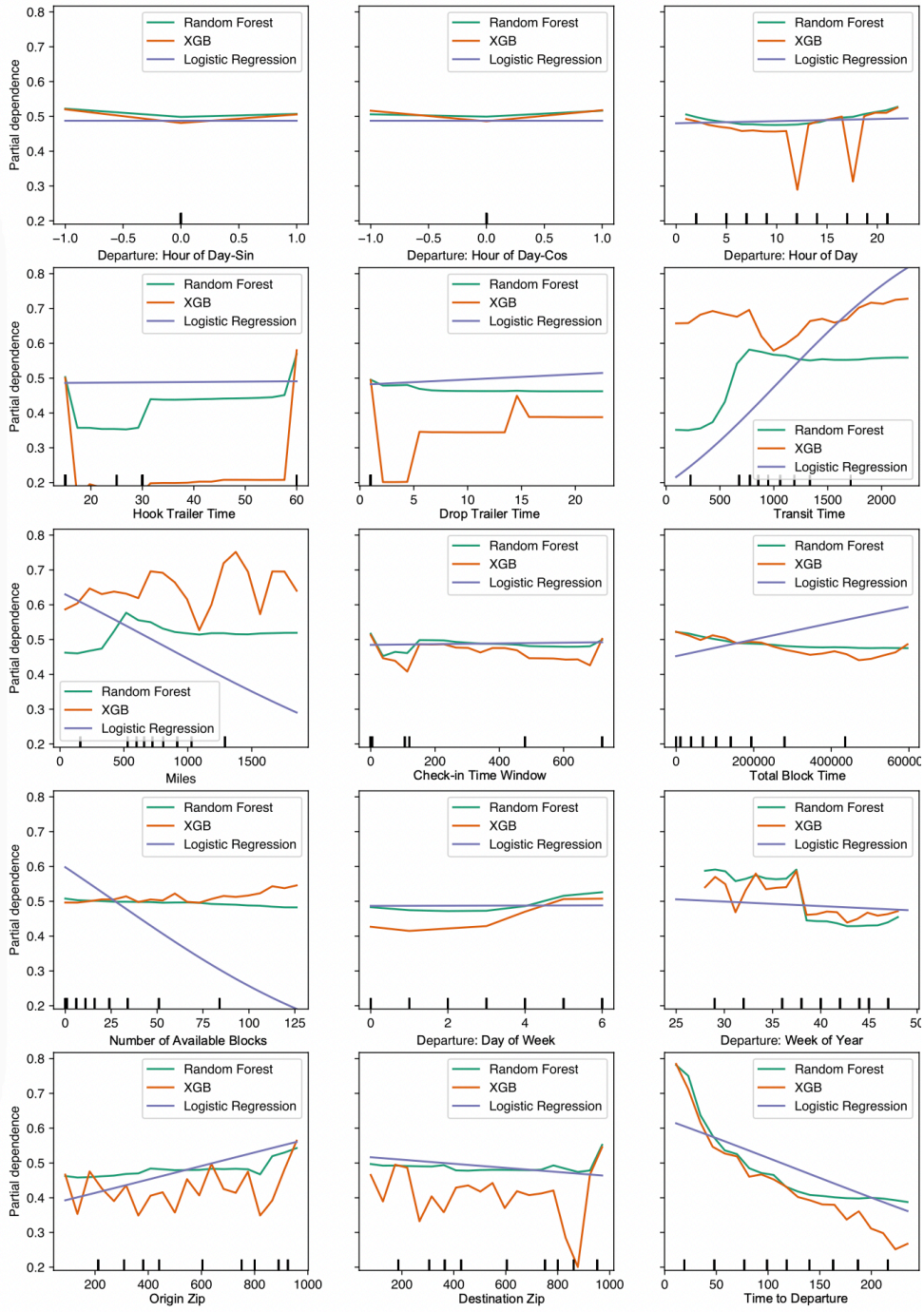


Figure 12 Permutation Feature Importance – Test Data – Random Forest

Weight	Feature
0.1126 ± 0.0138	Time to Departure
0.0827 ± 0.0083	Check-in Time Window
0.0488 ± 0.0073	Departure: Hour of Day-Sin
0.0410 ± 0.0106	Total Block Time
0.0379 ± 0.0034	Origin Zip
0.0251 ± 0.0070	Destination Zip
0.0225 ± 0.0038	Departure: Hour of Day-Cos
0.0149 ± 0.0032	Departure: Hour of Day
0.0115 ± 0.0093	Number of Available Blocks
0.0066 ± 0.0058	Transit Time
0.0044 ± 0.0063	Departure: Day of Week
0.0020 ± 0.0015	Miles
0.0014 ± 0.0054	Departure: Week of Year
0 ± 0.0000	Drop Trailer Time
0 ± 0.0000	Hook Trailer Time

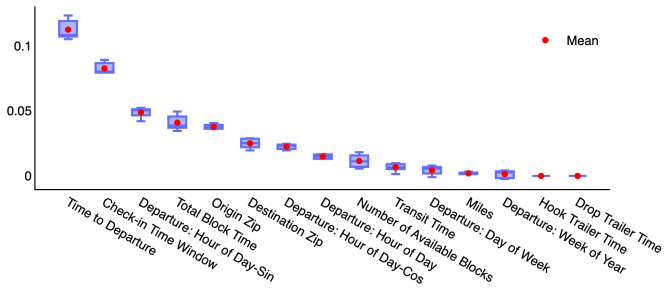


Figure 13 Permutation Feature Importance – Test Data – XGB

Weight	Feature
0.1796 ± 0.0138	Total Block Time
0.1193 ± 0.0030	Check-in Time Window
0.1075 ± 0.0054	Time to Departure
0.0555 ± 0.0123	Origin Zip
0.0410 ± 0.0101	Destination Zip
0.0354 ± 0.0083	Departure: Hour of Day-Sin
0.0341 ± 0.0082	Departure: Hour of Day-Cos
0.0263 ± 0.0107	Departure: Hour of Day
0.0197 ± 0.0022	Number of Available Blocks
0.0132 ± 0.0053	Departure: Day of Week
0.0112 ± 0.0046	Departure: Week of Year
0.0041 ± 0.0052	Transit Time
0.0009 ± 0.0017	Miles
0 ± 0.0000	Drop Trailer Time
0 ± 0.0000	Hook Trailer Time

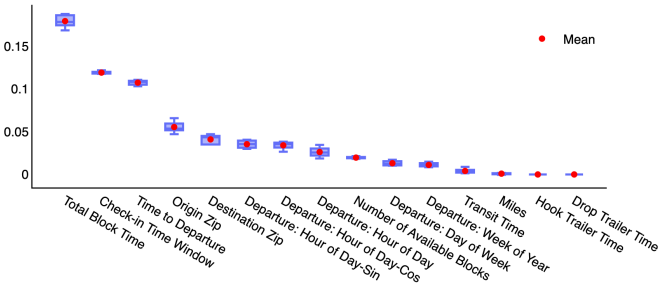


Figure 14 Permutation Feature Importance – Test Data – Logistic Regression

Weight	Feature
0.0554 ± 0.0132	Departure: Week of Year
0.0456 ± 0.0122	Time to Departure
0.0362 ± 0.0147	Check-in Time Window
0.0220 ± 0.0111	Total Block Time
0.0148 ± 0.0149	Origin Zip
0.0112 ± 0.0120	Departure: Day of Week
0.0003 ± 0.0023	Departure: Hour of Day
0.0002 ± 0.0005	Departure: Hour of Day-Cos
0 ± 0.0000	Drop Trailer Time
0 ± 0.0000	Hook Trailer Time
0 ± 0.0000	Departure: Hour of Day-Sin
-0.0004 ± 0.0016	Miles
-0.0007 ± 0.0015	Transit Time
-0.0020 ± 0.0023	Number of Available Blocks
-0.0038 ± 0.0075	Destination Zip

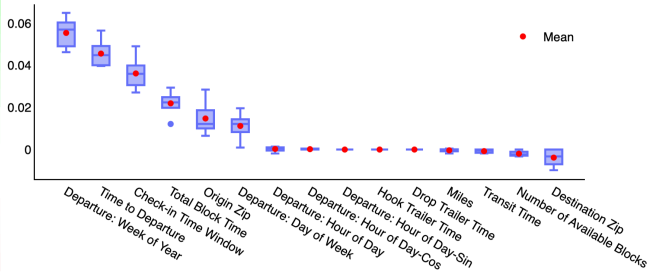


Figure 15 Probability Calibration Curve on Train Data – Random Forest

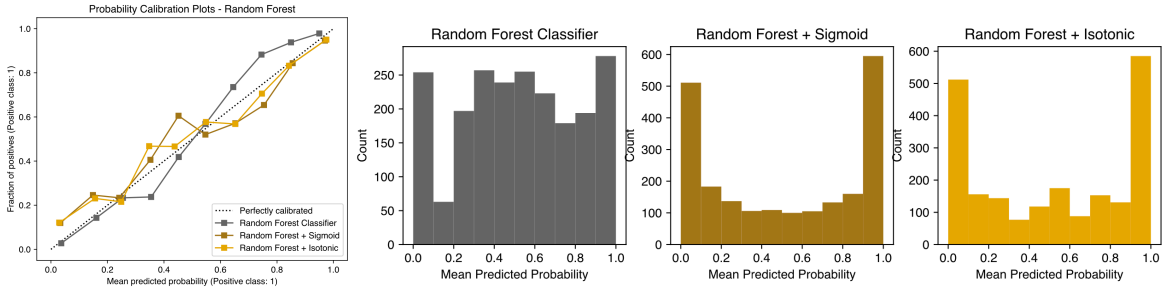


Figure 16 Probability Calibration Curve on Train Data – XGB

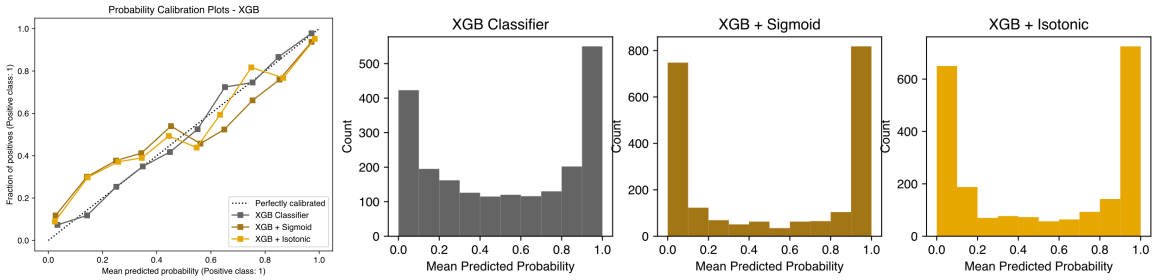


Figure 17 Probability Calibration Curve on Train Data – Logistic Regression

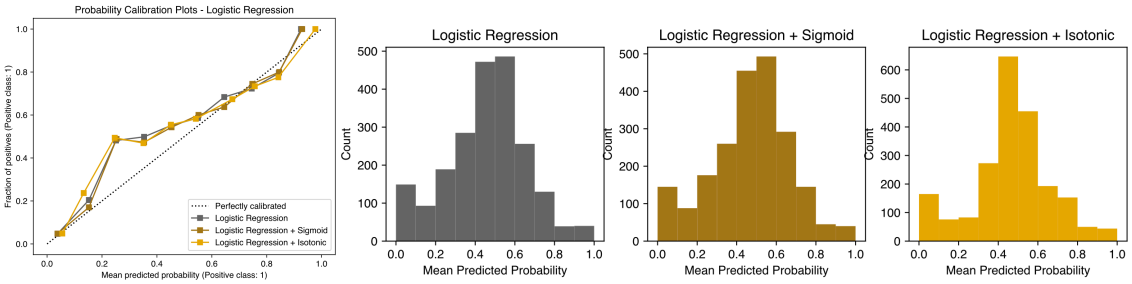


Figure 18 Implementation – Testing the Statistical Significance