Predictive Container Dwell Time Modeling for Optimized Port Yard Placement: A Machine Learning Approach

Fabio Andrés García Sánchez

School of Computer Science
The University of Auckland

Supervisor: Kaiqi Zhao

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Data Science, The University of Auckland, 2024.

Abstract

This research industry project presents a machine learning approach to predict container dwell times at the Ports of Auckland, aiming to optimize yard operations through data-driven decision support. The study analyzed over 1.5 million container movement records to develop predictive models that could effectively categorize container dwell times into three meaningful ranges: short-term (0-3 days), medium-term (4-11 days), and long-term (12+ days). Multiple machine learning algorithms were evaluated, including Random Forest, K-Nearest Neighbors, Logistic Regression, and Gradient Boosting. Logistic Regression was the most effective model, achieving F1 scores consistently above 73% in weekly predictions during the first quarter of 2024, with solid performance in short-term (86% F1 score) and medium-term (80% F1 score) predictions. K-Nearest Neighbors demonstrated complementary capabilities as a secondary validation model. The research established that container characteristics and temporal patterns were significant predictors of dwell time, while environmental factors such as wind conditions recorded by Ports of Auckland, provided for training, showed no meaningful impact on prediction accuracy. Models' strong performance in short- and medium-term scenarios, represented in most operational cases, indicated their readiness for practical implementation. Both models struggled with long-term predictions. This study also proposed integrating these predictive models into a Decision Support System (DSS) to automate container allocation decisions to reduce unnecessary container movements and improve space utilization. The findings provide bases for enhancing operational efficiency in port yard management, with clear pathways for future development through extended data collection, granular yard-level analysis, and real-time DSS implementation.

Acknowledgment

The journey of completing this dissertation has been transformative, made possible by the extraordinary support of many remarkable individuals.

I am profoundly grateful to the Port of Auckland, especially Manvi Madan and Valerie Wijaya, whose collaboration went far beyond professional courtesy. Their understanding during my health challenges and willingness to adapt to my circumstances made this research possible.

Two exceptional mentors guided my journey. To my supervisor, Dr. Kaiqi Zhao, thank you for your intellectual guidance, which consistently challenged me to reach higher standards while showing patience and understanding when I needed it most. Susanne Cooke, our Industry Engagement Manager, you were more than just a professional guide – your timely advice on managing health challenges alongside research responsibilities proved invaluable. Together, you shaped my research and understanding of proper academic mentorship.

In the moments when the path seemed most challenging, I found strength through the support of the university's student services team. Chanel Tafa from Campus Care and Emma Cooper-Williams, our Student Support Adviser, your compassionate guidance helped me navigate personal hurdles while keeping my academic dreams alive. You showed me that seeking support is not a sign of weakness but a step toward resilience.

To my beloved wife, your decision to build a new life with me in New Zealand speaks volumes about your love and commitment. Your strength was outstanding, holding everything in place as we learned about becoming parents while I did this dissertation. Your faith in me has been my guiding light through late nights of research and moments of doubt. This achievement is as much yours as it is mine, and sharing this journey of academic growth while preparing for parenthood has made it infinitely more meaningful.

This dissertation represents industry research and an example of human connection and support. Each person mentioned here has contributed to this scholarly work and my growth as a professional, a husband, and a soon-to-be father. For this, I am eternally grateful.

Contents

Chapter 1

Introduction

Imaging a place where thousands of boxes called containers flow through daily, carried by cranes or ships berth, orchestrating each movement to finally place them in a well-organized pile, set to be handled, rehandled, built, and retrieved when the time comes, to leave the port. For this logistical marvel, there is a critical challenge in this process: predicting, with precision, how long each box will stay to ensure they are moved only when necessary for their onward journey rather than to accommodate other cargo. This predictive capability can streamline port operations to reduce unnecessary movements while optimizing space utilization. This puzzle of time for improving movement and placements is the main point of this dissertation's inquiry.

In today's fast-paced world of global trade, container ports are like the beating hearts of our supply chains. However, here is the thing: these ports face a constant puzzle – how to manage their limited space most efficiently. This is not about squeezing in as many containers as possible but knowing which ones will leave soon or stick around for a while.

This research project was born from a collaboration with the Port of Auckland to find a way to use the data to optimize yard operations somehow. Following that mandate, using statistics, historical data, and the container's movements and features, this project proposed building predictive models to predict dwell times [19] , implementing machine learning techniques that can be involved inside the decision support system DSS (Figure 1.1) to help day-to-day operations by making decisions based on predicting plausible days of container stays.

That is where this research comes in to develop a machine learning model for doing something interesting: predict how long containers will stay in the port. However, why does this matter? What is the dissertation's primary goal? Predicting how long containers would hang around in the port (the trained predictive models) for being used as an input of a decision support system to optimize the yard operations.

Think of it this way: if port managers know Container A will be picked up in two days while Container B will hang around for a week, they can make smarter decisions about where to put them (stacking strategies). It is similar to a high-stakes Tetris game but with costly real-world consequences.

The proposed models are about more than just making educated guesses but are built for helping to solve challenges related to port operations:

1. Container Stacking Problem (CSP) [8]: The dwell time prediction helps to stack containers more effectively by reducing unnecessary movements. Those containers with shorter stays are placed on top, reducing relocations in future retrieval while improving operational efficiency.
2. Storage Space Allocation Problem (SSAP) [28]: By using dwell time predictions, the containers could be assigned to the yard based on how long they will stay, trying to leave short-term containers in easily accessible areas.
3. Container Retrieval Problem (CRP) [28] : Predicting dwell times helps organize containers efficiently for retrieval. Containers organized in a sequence leads to faster and smoother loading operations.
4. Container Relocation Problem (CRP/BRP) [16]: Dwell time predictions reduce unnecessary relocations by placing containers strategically with similar exit times together or in a strategic order, minimizing disruptions while lowering the costs of moving containers multiple times.

Here is the exciting part: All placement decisions should be made based on some rules considering the container's dwell time. This research aims to provide models for that by providing data-driven insights for reducing unnecessary container shuffling, saving time, cutting costs, and even reducing the environmental impact of port operations.

This dissertation will describe how these predictive models were built, the data used, and the machine learning techniques employed. It also shows how well it performs. This is about crafting a clever algorithm and improving how ports operate. This research aims to set additional tools to contribute to intelligent, efficient container handling by using the data and integrating an additional approach from data science theory to port operations.

1.1 System Architecture and Research Focus

Figure 1.1 illustrates a comprehensive two-tier system architecture that combines a Trained Predictive Model with a Decision Support System (DSS). This research focuses specifically on developing and optimizing the Trained Predictive Model component, which serves as the foundation for future yard optimization applications.

The Trained Predictive Model component is the core of this research. It integrates two fundamental data sources: Container Features, which include physical and operational characteristics from containers, and Time Features, which capture temporal patterns and timing-related data. Multiple classifiers process them to generate predicted dwell time ranges.

While the development of a complete DSS lies beyond the scope of this research, the architecture demonstrates how the predictive model could be integrated into a broader operational framework. The DSS reference is structured into two operational layers: Weekly Planning and Yard Operation. On the one hand, the Weekly Planning layer processes yard container data to generate predictions for current containers inside the yard, supporting medium-term operational planning. On the other hand, the Yard Operation layer handles real-time decisions by processing individual container features to predict dwell ranges for incoming containers, informing immediate stacking strategies and container allocation decisions. Both use the dwell times from predictions.

This dissertation focuses on predicting container dwell times across three ranges: short-term (0-3 days), medium-term (4-11 days), and long-term (12+ days), serving a practical purpose in yard operations: transforming raw predictions into actionable insights. By integrating ranges of dwell times into a Decision Support System (DSS), predictions could improve operations, enabling more informed container placement decisions that minimize unnecessary container movements and ultimately enhance the port's overall operational efficiency.

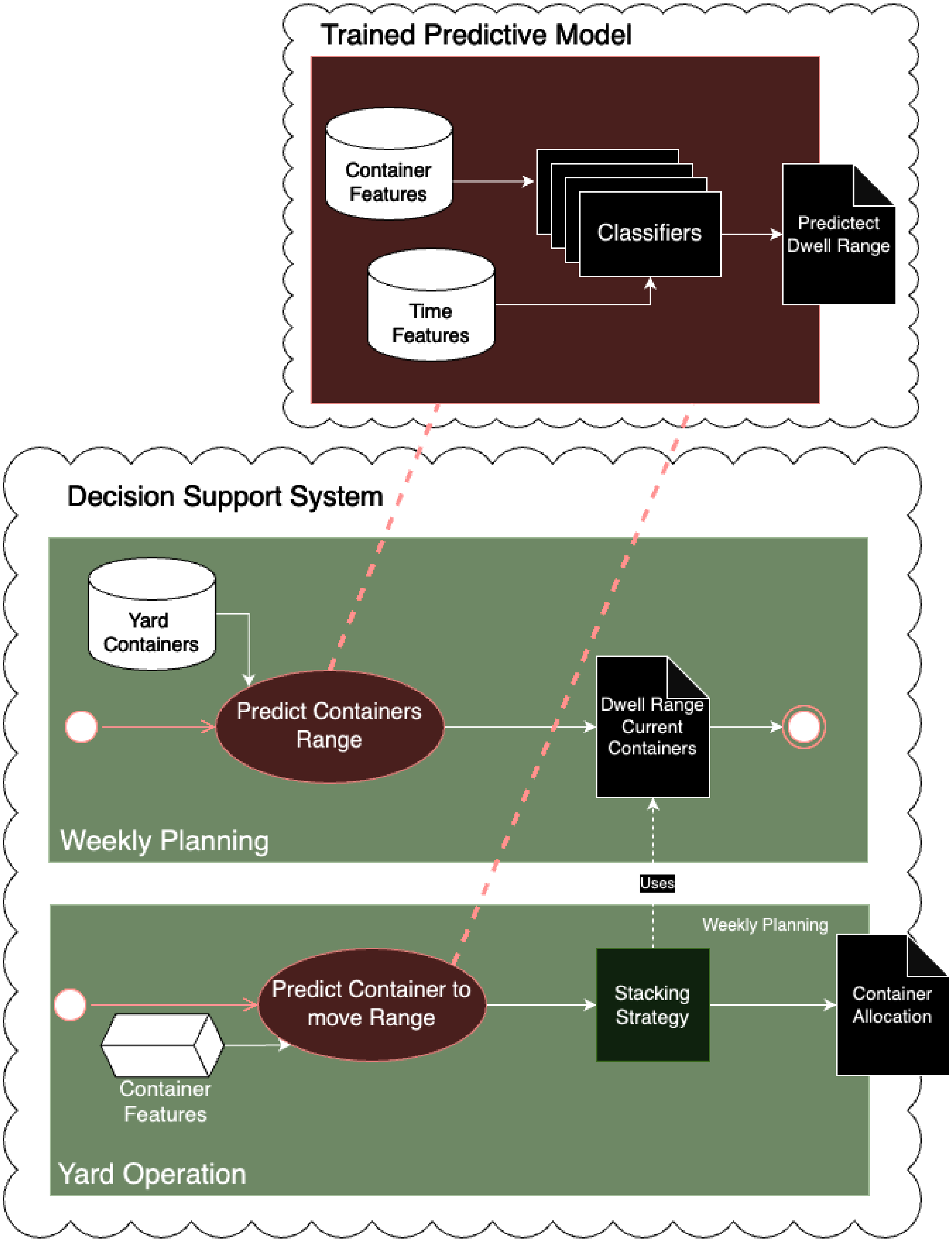


Figure 1.1: DSS architecture using machine learning models

Chapter 2

Literature Review / Related Works

Container stacking and decision support systems (DSS) have evolved significantly in port operations management.

While numerous studies have addressed container stacking problems [10, 24, 2, 13, 12, 22, 26, 23, 11], developing comprehensive DSS solutions has been relatively limited. Notable exceptions include the work of Murty et al. [21], Liu et al. [17], and Legato and Mazza [15], though these studies did not integrate predictive analytics or consider dwell times in their optimization models.

Even though dwell time performance is an important indicator, affected by multiple factor (Table 2.1), a gap exists in the literature regarding container dwell time prediction to solve optimization yard problems. Only some studies address this challenge; Moini et al. [20] analyzed U.S. ports, identifying crucial features such as port type, geographical location, and container traffic patterns. Based on this foundation, Kourounioti et al. [14] employed regression models and neural networks to predict dwell time.

The hierarchical approach to decision-making has emerged as a common framework in container terminal operations. As documented by several researchers [33, 5, 23], this two-stage process first considers block assignment at an aggregated level, followed by real-time decisions about specific container locations. Some insights about storage yard operations are provided by Carlo et al. [4].

Historical perspectives reveal the evolution of DSS applications for container handling. Starting with the early contribution from Van Hee and Wijbrands [30], the field has expanded to cover many operational aspects. Recent studies have addressed specific challenges such as real-time transportation planning [1, 31, 27], crane scheduling, berth allocation [32], and the integrated berth allocation and quay crane assignment Problem (BACAP) problem [29].

The effectiveness of category-based stacking policies in reducing rehandles has been well-documented. As a reference, Dekker et al. [6] demonstrated the benefits of category-based stacking. On the other hand, Borgman et al. [3] have highlighted the importance of departure time-based classification. Historical data can reduce operational costs, improving service levels for port users by leveraging and designing more effective stacking rules.

This research builds upon a study by Gaete et al. [7], which explored dwell time prediction using multi-class classification. Also, follows some of the ideas for a DSS architecture proposed by Maldonado et al. [18] which uses machine learning models for decision-making in port operations across a decision support system.

The current approach introduces a methodology for dwell time prediction by classifiers adjusted to The Ports of Auckland's operational context. It explains how these predictions could be integrated into DSS architecture that could be implemented in real-world port operations to optimize the yard finally.

Factor	**References**	**Type**
Vessel sailing schedule frequency	[19, 20]	Unique value
Container specifications*	[19, 20]	Nominal
Hinterland connections	[19, 20]	Unique value
Port governance structure	[19, 20]	Unique value
Terminal location & logistics	[19, 20]	Unique value
Terminal operations schedule	[19, 20, 25]	Unique value
Shippers and consignees	[20, 25]	Nominal
Regulatory procedures	[20]	Unique value
Transport corridors	[20]	Nominal
Maritime shipping details†	[20]	Nominal
Container flow balance	[20]	Nominal

*Including type (empty/full, dry/reefer), size (20/40 TEUs), contents

†Ocean carriers and empty container demurrage time

Table 2.1: Factors Influencing Container Dwell Time [7]

Chapter 3

Dataset

3.1 Introduction to container dataset

The dataset has records regarding container terminal operations. More over 1.5 million observations and 35 variables provide a detailed details container movements, including imports, exports, and container features.

3.2 Data Features Overview

The dataset encompasses a comprehensive set of features that characterize container operations at the port. These features can be categorized into five main groups. As shown in Table 3.1 , categorical features include container physical characteristics such as nominal length (20, 40, 45, or 10 feet) and location types. The operational categories, detailed in Table 3.2 , include import (IMPRT), export (EXPRT), storage (STRGE), transshipment (TRSHP), and through cargo (THRGH). Table 3.3 outlines the movement types captured through various categories like delivery (DLVR), loading (LOAD), yard moves (YARD), and others, while freight types are classified in Table 3.4 as either empty (MTY) or full container load (FCL).

Column Name	Description
nominal_length	Container size that could be 20, 40, 45, and 10
arrive_pos_loctype	Type of location arrival
last_pos_loctype	Last position's type where the container was fetched from, related to fm_pos _locid

Table 3.1: Container Characteristics Features

Category	**Description**
IMPRT	Import
EXPRT	Export
STRGE	Storage
TRSHP	When a box comes off a vessel and goes on another vessel
THRGH	When a box leaves on the same vessel it came on, it just went through

Table 3.2: Container Category Features

Category	**Description**
DLVR	Delivery
LOAD	Load
YARD	Yard Move
SHFT	Yard Shift
RECV	Receival
RLOD	Rail load
OTHR	Other
RDSC	Rail Discharge
DSCH	Discharge
SHOB	Shift O.B

Table 3.3: Move Kind Features

Category	**Description**
MTY	Empty
FCL	Full Container Load

Table 3.4: Freight Kind Features

Binary features, presented in Table 3.5 , track specific operational requirements such as power needs for temperature-sensitive cargo and twin operation capabilities (fetch, carry, and put).

Column Name	Description
requires_ power	Container designed to be transported as temperature sensitive cargo with precisely controlled temperatures
twin_fetch	Pairs of containers that were able be twin fetched
twin_carry	Pairs of containers that can able be twin carried
twin_put	Time when the whole chain of events related to a container_visit_ gkey is completed

Table 3.5: Binary Features

Temporal information is captured through datetime features, listed in Table 3.6 , that record crucial timestamps including port arrival (time_in), departure (time_out), and various handling operations (t_put, t_dispatch, t_fetch, t_discharge).

Column Name	Description
t_put	Time completed
t_dispatch	Time when a container is dispatched somewhere like on a vessel or truck
t_carry_dispatch	Requires verification
t_fetch	Datetime when the container was fetched
t_discharge	Datetime when container is left in position after being fetched
time_ in	Datetime of port arrival (for transships: can indicate category change from storage to export)
time_out	Time when the container leaves the port

Table 3.6: Datetime Features

Table 3.7 outlines numerical features that provide quantitative data about container handling, including rehandle counts, weight measurements (goods_anctr_wt_kg), and movement distances (dist_start, dist_carry).

Column Name	Description
rehandle_count	Number of times a container was rehandled
goods_and_ctr_wt_kg	Total weight of goods and container
dist_start	Distance where container started to be moved
dist_ carry	Distance container was carried during move (correlated to starting distance)

Table 3.7: Numerical Features

Finally, identifier features, detailed in Table 3.8 , maintain traceability through unique keys for moves (mve_gkey), locations (fm_pos_locid, to_pos_locid) , containers (inv_unit_id), and operators (bizunit_id).

Column Name	**Description**
mve_gkey	Unique identifier for each move
fm_pos_locid	FROM Position location id associated with fm_pos_name
fm_pos_name	Name of the location associated with fm_pos_locid
to_pos_locid	TO Position location associated with to_pos_name
to_pos_name	Name of the location associated with to_pos_locid
inv_unit_id	Container ID (persistent across port visits)
bizunit_id	Line operator's ID

Table 3.8: Identifier Features

3.3 Data Preprocessing

3.3.1 Data Augmentation

From existing features (table 3.6),new ones like the month, day, week of the month, day of the week, and even the time each container arrived were created based on the dates. Also, the number of days for each container, related to dwell time, was calculated using the dates in and out, creating the feature: "Days in the port category," a range of days.

3.3.2 Data Cleaning

The dataset was almost complete for the approach. However, some formats had to be set properly, from strings to dates. It had some columns with flags (like requires_ power) that had missing values. For these, an imputation with 0 was done.

3.3.3 Data Reduction

Our final dataset was reduced into 480,625 rows, each representing a unique container stay with ten chosen features split into two main chunks:

- 376,557 rows from 2023 for training and initial testing
- 104,068 rows from 2024 kept separate as our unseen data for final testing

The year 2024 was set up to test the model's performance on truly new data – the ultimate test of its predictive power.

Many columns containing IDs, session user information, dates, and flags outside the goal of predicting dwell time were removed.

After selecting the most important columns, the ones included were container's physical characteristics (nominal length), operational requirements (requires power status), and cargo attributes (freight kind and category). The temporal aspects were captured through multiple granularities: time of day (hour), day of week, month, business day status, week of year, and week of month.

3.4 Data analysis

3.4.1 Initial Exploratory Approach

The research began with an open-ended mandate from the Ports of Auckland to analyze their operational data to look for potential yard optimization insights. Rather than starting with a predefined hypothesis, this project follows an exploratory data science approach to discover patterns and opportunities from the provided data. The database structure was initially understood by examining column types and their values. Each finding pattern was evaluated with the stakeholders from Ports of Auckland.

3.4.2 Tool Selection: R

R was selected for the initial data exploration because of its robust statistical analysis and visualization tools. Packages such as ggplot2 enabled detailed visualizations, and data packages allowed efficient handling of large datasets. RStudios interactive environment provided quick visual feedback, making it easy to try different approaches rapidly. Rs built-in statistical functions and intuitive data manipulation also made it ideal for exploratory work in this phase.

R provided several clear benefits compared to Python, which was also a potential choice for data exploration. Its syntax allows data manipulation to be faster and easier. Also, its high-quality default plotting aesthetics minimized the need for extensive customization, allowing advanced customization when needed.

Rs built-in solid statistical analysis tools delivered immediate insights without extra library installations, and its efficient memory management was essential for effectively handling the sizeable operational dataset.

This data exploration is provided in [9].

3.4.3 Iterative Development Process

The methodology for this project follows an iterative approach that involves an in-depth data exploration using R, which exposed potential optimization areas. The insights generated in this phase were part of follow-up discussions with Ports of Auckland stakeholders to ensure they were aligned with real operational needs.

3.4.4 Data Exploration

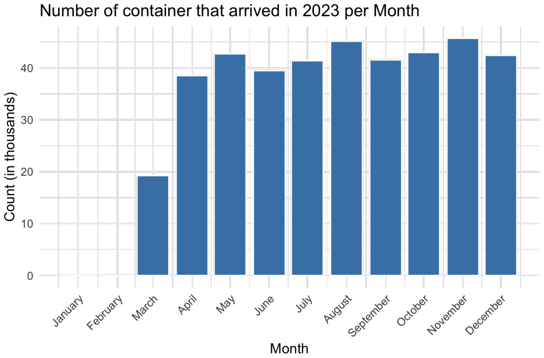


Figure 3.1: Container arrivals at a port 2023

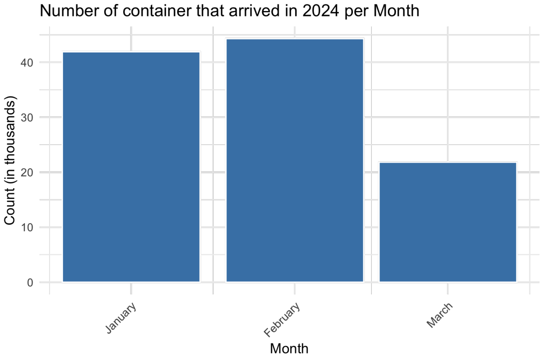


Figure 3.2: Container arrivals at a port 2024

Figure 3.1 illustrates the monthly container arrivals at a port, measured in thousands of units, spanning from January 2023 to March 2024, with the 2024 cycle needing to be completed as it only covers the first quarter. For the year 2023, there is no available data for January and February. It includes half the month of March with approximately 19,000 containers.

For April, the amount of containers reached about 38,000 containers. Throughout the remainder of 2023, the port maintained relatively stable activity levels from May through December, with monthly container arrivals fluctuating between 39,000 and 45,000 units. A peak activity was observed in September, with approximately 45,000 containers concluding in December, with approximately 42,000 container arrivals.

The available data for 2024 (Figure 3.2), which only spans the first quarter, shows that January began with approximately 42,000 container arrivals, followed by a slight increase to about 45,000 containers in February. However, March contains just the first two weeks of the month with approximately 22,000 containers.

Figure 3.3 shows, in an hourly analysis of container and their dwell times throughout 2023

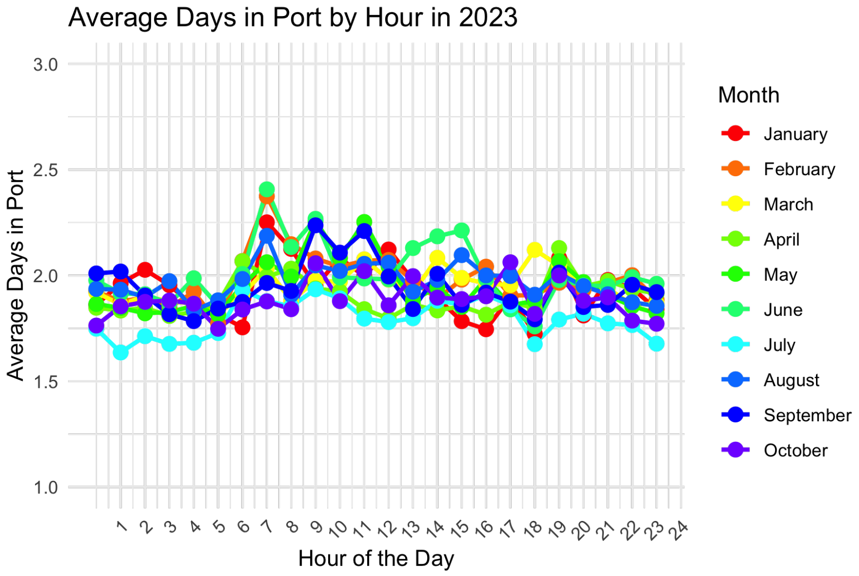


Figure 3.3: Average days in port by arrival hours per month

shows patterns that could be valuable for predictive modeling. The most notable feature is a consistent peak in dwell times during the morning (7:00-8:00 AM), where containers typically stay 2.2-2.4 days. This morning peak is particularly pronounced in June, which shows the highest dwell times during these hours. There is a cycle that starts with relatively stable dwell times of 1.8-2.0 days during early morning hours (1-5 AM), followed by the sharp morning peak (6-9 AM), then maintaining slightly elevated but more stable times during midday (10-15), before gradually declining through the afternoon and evening hours (16-24). These seasonal variations are evident, with summer months (June-August) having more volatility in dwell times, while July consistently shows lower overall dwell times than other months. Winter months (January-February) exhibit more consistent patterns with less variation. These interactions between arrival hour and month seem to be a possible factor in determining dwell times, as evidenced by the varying patterns across different months at specific hours. This suggests that both the hour of arrival and the month and their interaction term should be considered features for the predictive model.

Several insights about freight characteristics and their potential impact on dwell times could be valuable features for predictive modeling. Looking at figure 3.4 , the data shows two main types of freight: FCL (Full Container Load) and MTY (Empty) containers, with FCL having higher counts in both 2023 and 2024. From figure 3.5 , when examining dwell times across dif-

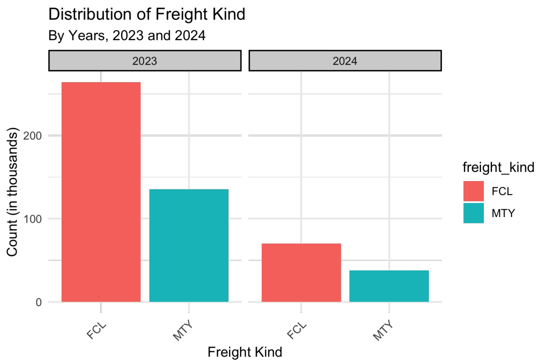
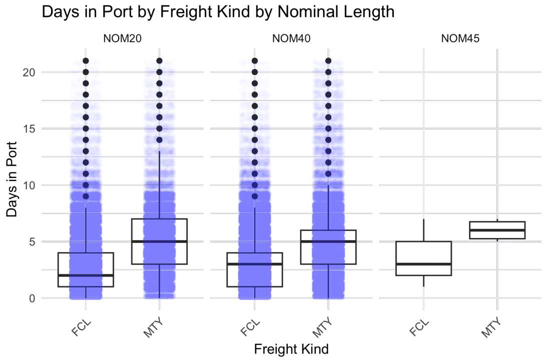


Figure 3.4: Distribution of container by Freight Kind



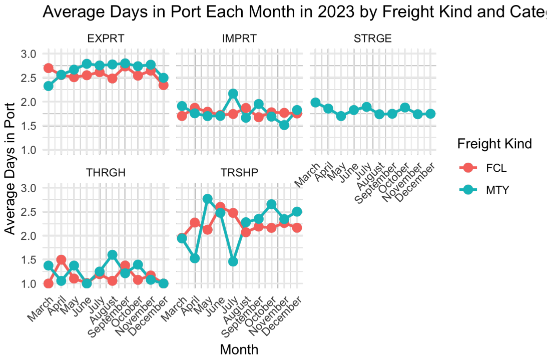


Figure 3.5: Freight kind vs size

Figure 3.6: Month vs freight kind

ferent nominal lengths (NOM20, NOM40, NOM45), MTY containers consistently show longer average dwell times than FCL containers, with the median dwell time for MTY containers being approximately five days compared to 2-3 days for FCL across all container sizes, suggesting that container type (FCL vs MTY) and nominal length are important predictive features. Figure 3.6 breaks down the operational types (EXPRT, IMPRT, STRGE, THRGH, TRSHP) by month, revealing distinct patterns: EXPRT containers show consistently higher dwell times (around 2.5-3 days) with slight monthly variation, while THRGH (through) containers have the lowest dwell times (around 1-1.5 days). TRSHP (transshipment) containers show the highest variability in dwell times, suggesting that category is a crucial predictor. The interaction between freight kind and category type also appears significant, as shown in figure 3.6, where FCL and MTY containers show different patterns within each operational category. For a predictive model, key features should include freight kind (FCL/MTY) as shown in figure 3.4, container size (nominal length) from figure 3.5, category (EXPRT/IMPRT/STRGE/THRGH/TRSHP) from figure 3.6, and their possible interactions.

Additionally, for categories, figure 3.8 shows patterns where export (EXPRT) containers consistently show higher dwell times (around 4 days) compared to import (IMPRT) containers (around 2 days) across all days of the week and the year. There is slight variation across weekdays, with Friday and Saturday showing marginally higher dwell times for both categories. Figure 3.7 shows the volume distribution across different operational categories, with IMPRT having the highest volume, followed by EXPRT in 2023 and 2024, though both show significant decreases in 2024. STRGE, THRGH, and TRSHP have notably lower volumes. figure 3.9 demonstrates how container size impacts dwell times across different operational categories -

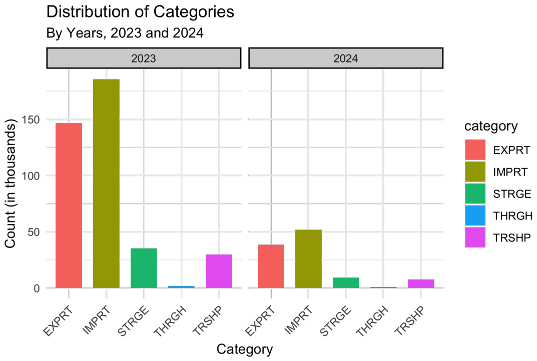


Figure 3.7: Distribution of categories

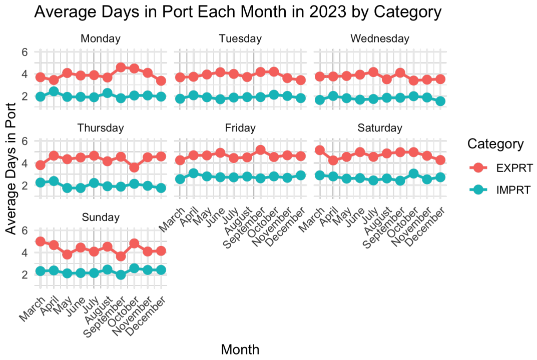


Figure 3.8: Month, day, and category

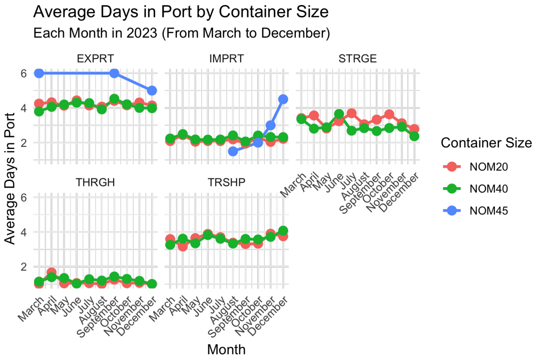


Figure 3.9: Month, category, and size

NOM45 containers show distinctly higher dwell times in EXPRT operations (around 6 days) than NOM20 and NOM40 (around 4 days). In IMPRT operations, all container sizes show similar dwell times (around 2 days), while TRSHP shows more variability across container sizes. For predictive modeling, key features should include the day of the week (shown important in figure 3.8), operational category (EXPRT/IMPRT/STRGE/THRGH/TRSHP as shown in Figure 3.7), container size (NOM20/40/45 from figure 3.9). The temporal patterns across days of the week (figure 3.8) suggest that day-of-week and weekend/weekday indicators would be helpful to features.

This analysis uncovered some patterns that influence how long containers stay in port. The timing of container arrivals matters significantly - there is a clear morning peak around 7-8 AM when containers tend to stay longer, and this varies across different days of the week and seasons. The category type plays a significant role: export containers consistently stay about twice as long as imports (4 days versus 2 days), regardless of when they arrive. We have also seen empty containers generally stay longer than full ones, and larger containers (especially 45-footers) tend to need more time in port, particularly for exports. Given these findings, a predictive model should focus on three main aspects: when containers arrive (time of day, day of week, and season), what kind of operation they are part of (export, import, storage, etc.), and their physical characteristics (size and whether they are full or empty). Building models upon those found features could make it possible to develop solid predictors for how long containers will stay in port.

Chapter 4

Methodology

4.1 Problem Definition

The main challenge for optimizing container yard operations at the Ports of Auckland was predicting dwell times to support placement decisions.

Predicting the exact number of days for each container was the first approach. Using classifier models with the selected features, predicting how many days each container would stay in the port was set as a target. The metrics in the models, especially accuracy, were low.

The variance in the predicted days was tremendous. Some similar-character containers were taken out daily while others stuck around for weeks. There needed to be a clear pattern for the classifier to generate acceptable predictions.

Rather than predicting an exact number of days, a range of days was a better approach improving the results. Having more than one range of days allows port managers to have options to make decisions, balancing prediction accuracy with operational utility.

4.2 Dataset Strategy and Model Evaluation

The experimental approach employs a temporal split strategy to ensure robust model development and realistic performance evaluation. As detailed in Table 4.1 , 2023 dataset will be used for the development phase, allocating 80% for training and 20% for cross-validation. This split enables thorough model tuning and hyperparameter optimization while maintaining statistical validity. For final evaluation, 2024 dataset will be used as an unseen test set, providing a true assessment of how the models perform on future data. This temporal separation is crucial as it mirrors real-world conditions where models must predict outcomes for future periods.

Time Period	Data Split	Purpose
2023	80%	Training Set: Model development and fitting
	20%	Validation Set: Cross-validation and tuning
2024	100%	Testing Set: Final evaluation on unseen data

Table 4.1: Dataset Split Strategy for Model Development and Evaluation

The development process begins with initial model training on the 2023 training set. Cross-validation techniques are employed during this phase to optimize model hyperparameters and assess performance stability. Once the bet models are selected and tuned, those final models will be evaluated over the 2024 dataset. This approach provides a rigorous test of model generalization, as the evaluation data represents genuinely unseen future observations.

4.3 Selecting Time Range Categories and Classifiers

This research took a data-driven approach to identify an optimal set of ranges of days to predict dwell time. Iteratively, multiple time range configurations were tested to find groupings that allowed operational needs to be achieved and predictions with solid performance generated.

The search for optimal time ranges involved systematically evaluating different bin configurations. Each configuration was tested across multiple machine learning algorithms to ensure the selected ranges would be robust across different modeling approaches. The goal was to find time ranges reflecting meaningful operational distinctions in container handling, provide sufficient data in each category for reliable model training, and enable predictive solid performance across different algorithms.

Multiple models were fit, using the training set (table 4.1), to be evaluated based on their metrics in finding ranges of days and classifiers. The models and ranges with the best performance, evaluated by F1 Score, were chosen to follow the next step: prediction evaluation.

Five classifiers were fitted initially: Random Forest, K-Nearest Neighbors, Logistic Regression, Gradient Boosting, and Naive Bayes (table 4.2).

Random Forest was a top performer (Table 4.3) with an impressive 87.0% F1 Score, showing remarkable consistency with its similar accuracy of 86.9% , having the best overall balance between precision and recall. KNN's performance had the second place with 86.0% F1 Score close to Random Forest in both metrics. More powerful classifier than initially expected. Gradient Boosting and Logistic Regression performed adequately but notably below the top two. Gradient Boosting achieved 77.0% F1 Score while Logistic Regression close behind at 76.0% . Naïve Bayes underperformed with just a 5.0% F1 Score, clearly not suitable for this specific

Model Type	**Models Fitted**
Gradient Boosting	64
KNN	64
Logistic Regression	64
Naïve Bayes	52
Random Forest	64
Grand Total	**308**

Table 4.2: Number of Models Fitted by Type

prediction task.

Model Type	**Max F1 Score**	**Max Test Accuracy**
Random Forest	87.0%	86.9%
KNN	86.0%	86.3%
Gradient Boosting	77.0%	78.7%
Logistic Regression	76.0%	76.7%
Naïve Bayes	5.0%	9.6%

Table 4.3: Model Type Performance Metrics

The models selected with appropriate metrics were Random Forest, KNN, Gradient Boosting and Logistic Regression.

After being sure that there were models that performed as expected, the next step was selecting the group of range to be used in the model. Those models were fitted with multiple groups of ranges to allow a posterior analysis based on F1-Score values, making sure that at least any range was significant.

4.3.1 Finding Ranges and Defining data scaling strategy

Some groups of ranges were over 80% (table 4.4) of accuracy and F1-Score. As there were appropriate thresholds of metrics overtaken, the analysis continued over classifiers to check which ranges were shared among them. The best ones were kept for posterior analysis.

Additionally, each model and range was also trained using a combination of techniques, including scaling via SMOTE (Synthetic Minority Over-sampling Technique) and stratification based on weekly and monthly. The goal was to explore whether scaling or stratification would improve the model's ability to handle imbalanced data effectively.

After multiple iterations and testing (table 4.5), regarding data scaling and stratification, the metrics in each approach did not lead to any significant improvements and, in some cases, resulted in worse performance. Consequently, it was determined that neither scaling nor stratification was beneficial for this particular dataset, and these techniques were not applied in the

Bins Ranges	**Models Fitted**	**Max Test Accuracy**	**Max F1 Score**
[0, 3, 10, 21]	20	79%	79.0%
[0, 3, 11, 21]	20	80%	80.0%
[0, 3, 12, 21]	20	80%	80.0%
[0, 3, 9, 21]	20	78%	77.0%
[0, 4, 10, 21]	16	83%	82.0%
[0, 4, 11, 21]	16	83%	83.0%
[0, 4, 12, 21]	20	84%	83.0%
[0, 4, 9, 21]	16	81%	81.0%
[0, 5, 10, 21]	20	85%	85.0%
[0, 5, 11, 21]	20	85%	85.0%
[0, 5, 12, 21]	20	86%	85.0%
[0, 5, 9, 21]	20	84%	84.0%
[0, 6, 10, 21]	20	86%	86.0%
[0, 6, 11, 21]	20	87%	86.0%
[0, 6, 12, 21]	20	87%	87.0%
[0, 6, 9, 21]	20	86%	85.0%
Total	**308**		

Table 4.4: Ranges and Max possible F1-Scores and accuracy by Fitted Models

final model.

Given the presence of attractive ranges and classifiers and the fact that no stratification or scaling was required, a group of ranges was selected. The optimal group, identified as the best across all the configurations, consisted of 0-3 days, 4-11 days, and 12+ days ([0, 4, 12, 21]). These groups of ranges provided solid performance, an acceptable data distribution (fig. 4.1) while allowing the defined ranges to be good enough to establish stacking strategies for the port.

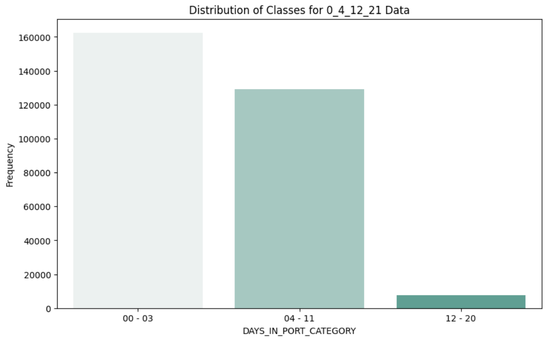


Figure 4.1: Ranges Distribution Labels

Range	F1 Score	Precision	Recall	Scaled	Stratified
[0, 4, 10, 21]					
	0.81	0.81	0.81	False	False
	0.81	0.81	0.81	False	True
	0.81	0.81	0.81	True	False
	0.81	0.81	0.81	True	True
	0.81	0.82	0.81	True	False
	0.82	0.82	0.83	False	False
	0.82	0.83	0.83	False	True
[0, 4, 11, 21]					
	0.81	0.82	0.82	True	True
	0.82	0.82	0.82	False	False
	0.82	0.82	0.82	False	True
	0.82	0.82	0.82	True	False
	0.82	0.83	0.82	True	False
	0.83	0.83	0.83	False	False
	0.83	0.83	0.83	False	True
[0, 4, 12, 21]					
	0.81	0.82	0.80	True	True
	0.82	0.82	0.82	False	False
	0.82	0.82	0.82	False	True
	0.82	0.82	0.82	True	False
	0.82	0.82	0.82	True	True
	0.83	0.83	0.82	True	False
	0.83	0.84	0.84	False	False
	0.83	0.84	0.84	False	True

Table 4.5: Performance Metrics by Range Under Different Conditions

4.3.2 Class Imbalance Analysis

The distribution of container dwell times are imbalanced across the three categories (fig. 4.1). The largest group, the short-term stays (0-3 days), has around 160,000 containers. For Medium-term stays (4-11 days), there are about 130,000 containers. For the smallest group, long-term stays (12-20 days), there are significantly fewer containers, with only 10,000 units representing a small portion of the overall data.

4.3.3 Selecting the Metric: F1 Score

Addressing the Imbalance:

Ranges evaluated were more common than others (Figure 4.1), which led to an imbalanced evaluation, generating a wrong sense of good predictions. The accuracy metric misled these models' results by predicting the most frequent classes. F1-Score corrected this, providing a more appropriate understanding of the real performance.

Due to this imbalanced distribution, the F1 Score provides a handy evaluation metric. Since accuracy could be misleading, with short and medium-term stays making up more than 95% of cases, the F1 Score offers a more balanced assessment by considering precision and recall. Addressing imbalance with a proper metric is essential because, in port operations, incorrectly predicting a longer stay (false positives) or failing to identify a long-term stay (false negatives) can result in inefficient container placement and higher rehandling costs. The F1 Score's sensitivity to both error types helps prevent models from appearing highly accurate by predicting the most common stay categories. Long-term stays are less frequent but impact yard management and resource allocation for the longer term; this is why the F1 Score is particularly valuable. The balance provided by the F1 Score fits well with the practical needs of port operations, providing accurate prediction evaluation across all dwell time categories, which is essential for efficient yard optimization.

Balancing Precision and Recall (Trade-off):

The F1 score is widely recognized for combining precision (proportion of true positive predictions among all positive predictions) and recall (proportion of true positive predictions among all actual positives).

Identifying the correct range and minimizing the misclassifications was essential because the cost of misclassification was high. F1-score evaluates false positives (incorrectly assigning a range) and false negatives (failing to detect the correct range).

4.4 Model Refinement and Optimization

After selecting the classifiers, a metric, and defining their parameter ranges, they were fine-tuned to enhance performance. GridSearchCV was employed with 5-fold cross-validation to identify the optimal hyperparameters for each model. This process utilized 80% of the 2023 data for training, with the remaining 20% reserved for evaluation.

4.4.1 The Scoring Metric: Weighted F1 Score

Weighted F1 score was used for optimization metric during the tuning process, a choice driven by the imbalanced nature of our three classes.

Unlike a simple F1 score, the weighted F1 score calculates metrics for each label and finds their average, weighted by the number of true instances for each label. This ensures that the performance of smaller classes is not overshadowed by larger classes.

The three classes were imbalanced. The weighted F1 score offers an interpretable single metric that summarizes performance across all of them while respecting their different sizes. This approach led to more robust and reliable models that can effectively handle the class distribution in our dataset.

4.4.2 The hyperparameters

Each model was tuned with a set of hyperparameters to optimize its performance. The hyperparameters, as shown in Appendix A.1.1 , were selected based on the models' characteristics and the nature of the dataset. The tuning process aimed to identify the best combination of hyperparameters that would maximize the models' predictive power. Many models were evaluated based on each algorithm's complexity during the hyperparameter tuning process. Random Forest required fitting 864 models while tuning six hyperparameters, including parameters like n_estimators, max_ depth, and bootstrap options. Gradient Boosting, being the most complex regarding tunable parameters, involved fitting 972 models across seven hyperparameters, such as learning rate, number of estimators, and tree-specific parameters. Logistic Regression was less complex, requiring 360 models to be fitted while tuning five hyperparameters, including regularization strength and solver options. The simplest model in terms of tuning was K-Nearest Neighbors, which needed only 60 models to be fitted while optimizing three hyperparameters: the number of neighbors, weight function, and distance metric.

4.4.3 The Best Parameters

The best parameters were selected based on the weighted F1 score as shown in Appendix A.1.2.

For Random Forest the best parameters were max_depth (20) and min_samples_ leaf (2) limit the tree size while reducing overfitting and maintaining robust learning. Using max_ features as "sqrt" and 500 estimators ensures better generalization by creating diverse trees.

Nearest Neighbors selected parameters were Manhattan distance makes the model less sensitive to outliers, while 23 neighbors smooth the predictions. Distance-weighted voting emphasizes closer data points, improving local prediction accuracy.

The best parameters for Gradient Boosting where a moderate learning rate (0.1) combined with a controlled max_ depth (5) helps to prevent overfitting. The choice of 500 estimators allows the model to build a strong ensemble and subsampling features (max_features: "sqrt") enhance generalization.

Regarding Logistic Regression, the best parameters were C=1 and l2 penalty for regularization and newton-cg solver.

4.4.4 Analysis of Model Tuning Results

Model	Precision	Recall	F1-Score
Random Forest	0.8397	0.8395	0.8368
K-Nearest Neighbors	0.8349	0.8337	0.8308
Gradient Boosting	0.8061	0.8073	0.8035
Logistic Regression	0.7521	0.7666	0.7581

Table 4.6: Model Performance Metrics (Ordered by F1-Score)

Random Forest was the best among the four models. It achieved a high F1-score of 0.8368, with precision (0.8397) and recall (0.8395) showing consistency across all classes.

K-Nearest Neighbors performed impressively, taking second place with an F1-score of 0.8308. The model's success suggests that the dataset's features naturally cluster, meaning similar instances tend to group in the feature space. KNN leveraged these local patterns well, leading to high precision (0.8349) and recall (0.8337), making its performance nearly comparable to Random Forest.

Gradient Boosting delivered solid metrics, with an F1-score of 0.8035. Its precision (0.8061) and recall (0.8073) were balanced.

Logistic Regression was the worst performer, with an F1-score of 0.7581, precision of 0.7521, and recall of 0.7666. Even though it did not match the performance of the other models, it is still a solid baseline when simplicity and interpretability are priorities.

4.5 Code Implementation Details

4.5.1 Overview

The implementation represents a comprehensive machine-learning pipeline explicitly designed for container dwell time prediction. The system incorporates multiple sophisticated components working together to create, evaluate, and deploy predictive models while maintaining high code organization and reproducibility standards. This implementation is provided in [9].

4.5.2 Core Components

The ScoringMetrics enumeration interface provides a systematic approach to managing model evaluation, allowing multiplele metrics, including accuracy, various F1 score implementations (micro, macro, weighted), precision, recall, and ROC AUC scores.

The ModelType enumeration interface defines the supported machine learning algorithms: Random Forest, Logistic Regression, Gradient Boosting, Support Vector Machine (SVM), Naive Bayes, and K-Nearest Neighbors (KNN).

Feature selection data structure focuses on vital predictive elements combining container characteristics (such as power requirements, length, and freight type), temporal aspects (month, hour, weekday), and operational indicators (business day flags and week information). These features were selected based on their potential predictive value for dwell time estimation.

4.5.3 Key Classes

The DataPreprocessor class coordinates all processes related to data preparation, creating categorical labels for dwell time ranges and managing data cleaning and transformation. It allows different binning strategy definitions to categorize different dwell time ranges, ensuring consistent data processing across training and prediction phases.

The ModelTrainer class serves as the core engine for model development. It is responsible for implementing comprehensive hyperparameter tuning through grid search with cross-validation. It manages the entire model lifecycle, from data splitting and preprocessing to training and validation while incorporating stratification options for temporal consistency. The class also handles performance evaluation and metrics recording, ensuring thorough documentation of model behavior.

The ReportGenerator class manages the analysis and presentation of results, generating model reports for comparison in multiple formats, such as Excel and CSV files, with detailed classification performance analyses and temporal performance evaluations. This component ensures that model performance can be effectively communicated and analyzed across different timeframes and metrics while persisting results from expensive training and tuning processes.

4.5.4 Tuning, Features and Metrics

The workflow allows model-tuning functions implementing a grid search approach and 5-fold cross-validation. It supports hyperparameter optimization across all the model types in the same way, accommodating both scaled and unscaled data and providing flexibility in data preprocessing approaches while maintaining consistent evaluation frameworks.

The evaluation framework assesses model performance with a different range of metrics, generating detailed classification reports, confusion matrices, and feature importance analysis, giving a complete picture of the model's performance, including all critical aspects of the prediction task.

It implements strategic data-splitting approaches that consider the temporal features in the data, ensuring reliable model evaluation and prediction capabilities by weekly ranges.

4.5.5 Technical Implementation

The object-oriented programming principle is the paradigm used in this project to create a modular and maintainable system. For each class, a clear responsibility is defined. The implementation includes popular Python libraries such as scikit-learn for machine learning algorithms, pandas for data manipulation, and various visualization tools for result analysis.

This implementation provides a solid base for developing and evaluating multiple machine-learning models to predict container dwell time. The best practices in software engineering combined with machine learning libraries create a reliable, easy-to-follow, extendible, and maintainable system.

Chapter 5

Results and Discussion / Model Evaluation

5.1 Comparative Performance Analysis of Models

5.1.1 Temporal Performance Analysis

Month	**Week**	**Max Acc**	**Max Prec**	**Max Recall**	**Max F1 Score**	**Predictions**
1	1	76.8%	75.3%	76.8%	76.0 %	4
1	2	78.4%	78.0%	78.4%	78.0 %	4
1	3	79.2%	78.2%	79.2%	78.0 %	4
1	4	69.5%	67.8%	69.5%	69.0 %	4
2	1	74.5%	74.2%	74.5%	74.0 %	4
2	2	79.3%	77.9%	79.3%	79.0 %	4
2	3	81.1%	81.9%	81.1%	81.0 %	4
2	4	79.3%	78.7%	79.3%	79.0 %	4
3	1	83.1%	82.2%	83.1%	83.0 %	4
3	2	77.0%	79.7%	77.0%	77.0 %	4
Total						**40**

Table 5.1: Overal Models' Performance Metrics by Week and Month

The performance metrics shown in Table 5.1 reveal significant insights about the models' predictive capabilities across the first quarter of 2024. The models demonstrate strong overall performance, with accuracy consistently ranging between 69.5% and 83.1% . The highest performance was achieved in the first week of March, where the model reached its peak accuracy of 83.1%, accompanied by strong precision (82.2%) and F1 score (83.0%). Conversely, the final week of January showed the lowest performance across all metrics, with accuracy dropping to 69.5% . An interesting trend emerges in February, where we observe steady improvement from the first to the third week, culminating in a robust 81.1% accuracy and 81.0% F1 score in the third week, before slightly declining in the final week.

The consistency between accuracy and F1 scores throughout the dataset indicates well-balanced models with good harmony between precision and recall, suggesting reliable and stable predictions. With a total of 40 predictions across the quarter, spanning from January through early March, the sample size provides sufficient confidence in these performance metrics.

5.1.2 Model-Specific Performance

Model	Max Acc	Max Prec	Max Recall	Max F1 Score	Min F1 Score
Log. Regression	83.1%	82.2%	83.1%	83.0%	69.0%
KNN	77.5%	75.6%	77.5%	77.0%	64.0%
Random Forest	57.4%	71.7%	57.4%	57.0%	34.0%
Grad. Boosting	22.6%	74.3%	22.6%	18.0%	0.0%

Table 5.2: Model Performance Comparison Week and Month

The comparative analysis of models, shown in Table 5.2 , highlights significant differences in predictive effectiveness across various algorithms.

Logistic Regression is the best classifier, peaking at an accuracy of 83.1% and an F1 score of 83.0% . Its consistent performance across diverse scenarios—evident from a minimum F1 score of 69.0% —demonstrates robust and reliable predictive power.

K-Nearest Neighbors (KNN) also performs well, ranking as the second-best model. It maintains stable results, with a maximum accuracy of 77.5% and an F1 score reaching 77.0%, though its minimum F1 score dips to 64.0 %.

Random Forest achieves moderate results, with a maximum accuracy of 57.4% , though it demonstrates solid precision at 71.7% , indicating reliable positive predictions when the model is more optimistic.

The Gradient Boosting model struggles in this context, with a maximum accuracy of only 22.6% despite a decent precision of 74.3%. Its F1 score, ranging from 18.0% to as low as 0.0% , points to high instability and limited reliability for this specific task.

These findings suggest that linear models like Logistic Regression and instance-based learning methods like KNN are better suited to this predictive task than more complex ensemble methods. The results indicate that Logistic Regression should be used for core predictions and KNN as a validation tool. The consistently strong performance (above 70%) for the top models suggests they are well-suited for integration into the operational decision-making processes at the Ports of Auckland.

5.2 Logistic Regression

5.2.1 Model Performance Analysis

Month	Week	Accuracy	Precision	Recall	F1 Score	ROC AUC
1	1	76.8%	75.3%	76.8%	76.0%	81.7%
1	2	78.4%	78.0%	78.4%	78.0%	84.8%
1	3	79.2%	78.2%	79.2%	78.0%	78.1%
1	4	69.5%	67.8%	69.5%	69.0%	79.6%
2	1	74.5%	74.2%	74.5%	74.0%	74.6%
2	2	79.3%	77.9%	79.3%	79.0%	83.8%
2	3	81.1%	81.9%	81.1%	81.0%	85.5%
2	4	79.3%	78.7%	79.3%	79.0%	83.9%
3	1	83.1%	82.2%	83.1%	83.0%	83.6%
3	2	77.0%	79.7%	77.0%	77.0%	86.8%

Table 5.3: Logistic Regression - Weekly and Monthly Model Performance Metrics

Table 5.3 presents the weekly and monthly performance metrics for the Logistic Regression model in predicting container dwell times.This classifier performed well during the evaluated period. Its F1 scores were between 69.0% and 83.0%, often above 74% . Accuracy metrics supported its performance, which also fell within that range, peaking at 83.1% . Its ability to differentiate between dwell time categories was also impressive. ROC AUC values between 74.6% and 86.8% consistently surpassed 83% , demonstrating its reliability in identifying different dwell time classes under various conditions.

Logistic regression was the best classifier for predicting container dwell times. Its stable F1 scores above 74 % in typical operational settings and high ROC AUC values, frequently over 83% , make it a strong choice for primary deployment in dwell time prediction. This model is particularly effective in applications where clear separation between dwell time categories is essential, adding real value in port operations where accurate dwell time predictions are critical for resource planning and allocation. The Logistic Regression model is an ideal baseline for dwell time prediction and sets a standard for evaluating other approaches.

5.2.2 Class-Specific Performance Analysis

Class	Precision	Recall	F1 Score
From 00 to 03 days	88.0%	85.0%	86.0%
From 04 to 11 days	77.0%	83.0%	80.0%
From 12 to 20 days	0.0%	0.0%	0.0%

Table 5.4: Logistic Regression - Performance Metrics by Class March Week 1

Table 5.4 looks at how well the Logistic Regression model performs across different dwell time categories in the first week of March 2024, corresponding to its training timeframe from March 2023. This period allows evaluations of the model's effectiveness in similar seasonal periods in different years.

The model metrics showed predictive capabilities for shorter dwell times: containers in the 0-3 day range were identified with an F1 score of 86.0%, supported by 88.0% precision and 85.0% recall. The model maintained robust performance for containers within the 4-11 day range with an 80.0% F1 score, demonstrating 77.0% precision and 83.0% recall. However, this model did not identify containers with extended stays of 12-20 days, recording 0% across all performance metrics for this category.

The metrics reveal insights about the model's operational capabilities and limitations. Under 11 days, the model performed well when predicting, making it a reliable tool for short- to medium-term operational planning. High precision in the 0-3 day range is valuable for identifying containers requiring immediate attention. However, the model could have performed better when predicting extended dwell times (12-20 days), indicating a substantial limitation. While the model effectively handles typical container dwell times, these findings suggest enhancements to address the identification and management of extended-stay containers.

5.2.3 Feature Importance Analysis

As shown in Table A.6 , the logistic regression model makes explicit patterns that affect how long containers stay in port. The most critical factor is the container's purpose or category. Through containers (marked as 'THRGH') strongly suggest shorter stays, with a high positive value of 2.688. On the flip side, containers marked for transshipment ('TRSHP'), storage ('STRGE'), or export ('EXPRT') typically stay longer, shown by their negative values around

-1.1. Import containers ('IMPRT') tend toward shorter stays but not as dramatically as through containers, with a moderate positive value of 0.686.

The physical features of containers also matter, but their purpose is different. Regarding container sizes, 20-foot and 40-foot containers show similar mild positive effects (0.197 and 0.189), suggesting slightly shorter stays. However, 45-foot containers tend toward longer stays, indicated by a negative value of -0.385. Whether a container is full or empty also makes a difference: full containers (FCL) tend toward shorter stays (0.147), while empty ones (MTY) lean toward longer stays (-0.147).

Time-related factors have more subtle effects on how long containers stay. The day of the week matters most among these (-0.264), showing that container movements follow weekly patterns. Other time factors, like whether it is a business day (-0.059), the week of the year (0.040), and the time of day (0.020), have minor effects. Interestingly, which month it is (-0.010) or which week of the month (-0.011) barely matters at all. Whether a container needs power also has a negligible effect (-0.041), suggesting powered containers might stay slightly longer. These findings help port operators understand what drives container stay times and could help them make better planning decisions.

5.3 K-Nearest Neighbors

5.3.1 Model Performance Analysis

Month	Week	Accuracy	Precision	Recall	F1 Score	ROC AUC
1	1	65.1%	65.6%	65.1%	65.0%	71.4%
1	2	73.9%	74.8%	73.9%	74.0%	78.0%
1	3	65.3%	64.6%	65.3%	65.0%	66.2%
1	4	67.3%	65.3%	67.3%	66.0%	70.1%
2	1	64.3%	64.7%	64.3%	64.0%	66.6%
2	2	70.8%	72.7%	70.8%	71.0%	77.5%
2	3	67.3%	69.9%	67.3%	67.0%	71.3%
2	4	77.5%	75.6%	77.5%	77.0%	71.5%
3	1	71.9%	74.5%	71.9%	73.0%	74.7%
3	2	64.2%	74.7%	64.2%	67.0%	78.6%

Table 5.5: K-Nearest Neighbors - Weekly and Monthly Model Performance Metrics

Table 5.5 presents the performance metrics for the K-Nearest Neighbors (KNN) model across different weeks and months. This model performed consistently across the period. Its F1 scores went from 64.0% to 77.0% , with scores above 65.0% in most measurements. Additionally, ranging from 64.2% to 77.5% , its accuracy demonstrated balanced precision and recall characteristics supporting the F1 scores. Also, its ROC AUC values from 66.2% to 78.6%, with several periods exceeding 70% , indicating a reliable ability to distinguish between different dwell time classes, albeit at a lower level than the Logistic Regression model.

This model was the second most effective classifier, becoming a valuable complementary tool to the primary Logistic Regression model in container dwell time prediction. This model generally trailed Logistic Regression by 5-8 percentage points but still performed solidly, with F1 scores above 65% under normal conditions and ROC AUC values often exceeding 70% . These results make it a dependable secondary tool for validating predictions. This model is precious in providing independent verification of logistic regression predictions while offering an alternative.

5.3.2 Class-Specific Performance Analysis

Class	Precision	Recall	F1 Score
From 00 to 03 days	80.0%	71.0%	76.0%
From 04 to 11 days	68.0%	74.0%	71.0%
From 12 to 20 days	5.0%	15.0%	7.0%

Table 5.6: K-Nearest Neighbors - Performance Metrics by Class Week 1

Table 5.6 presents the class-specific performance metrics for the K-Nearest Neighbors (KNN) during the first week of March 2024 that covers the same evaluation period corresponding to its training data from March 2023. The model showed good capability in identifying shorter dwell times, achieving a 76.0% F1 score for containers in the 0-3 day range, supported by 80.0% precision and 71.0% recall. The model maintained moderate effectiveness for containers within the 4-11 day range with a 71.0 % F1 score, exhibiting 68.0% precision and 74.0% recall. Unlike the Logistic Regression model, the KNN algorithm demonstrated some ability to identify extended stays of 12-20 days, albeit with limited success, achieving a 7.0% F1 score with 5.0% precision and 15.0% recall.

Despite its lower overall performance compared to Logistic Regression, this model is a valuable complement, particularly in its capacity to detect extended dwell times. Its accuracy falls approximately 10% points below the Logistic Regression model for shorter stays, but it performs better identifying some extended dwell times (12-20 days), providing additional operational value.

The KNN model works well as a supporting tool, especially for validating predictions on standard-length stays. This year-over-year analysis for the first week of March highlights its usefulness for a possible dual-model framework implementation, where its strengths can complement the Logistic Regression's weakness to improve the overall system's reliability.

5.4 Model Comparison and Integration Strategy

A comparative analysis of the Logistic Regression and K-Nearest Neighbors models reveals distinct performance characteristics and complementary strengths in container dwell time prediction. The Logistic Regression model has a superior overall performance, mainly in shorter dwell time ranges (0-3 days), with an F1 score of 86.0% compared to KNN's 76.0% . The same is true for the medium-range predictions (4-11 days), where Logistic Regression maintains an F1 score of 80.0% versus KNN's 71.0% . However, for extended dwell time prediction (12-20 days), Logistic Regression did not have detection capabilities, while KNN, despite its modest performance, maintains some predictive ability with a 7.0% F1 score.

Complementing both models with the best of their characteristics could provide a potential enhancement through ensemble methods. For instance, a stacking approach could be efficient by leveraging for shorter stays to the high precision of Logistic Regression and incorporating KNN's ability to detect extended dwell times. Such an ensemble could implement a weighted voting system, giving greater weight to Logistic Regression predictions for containers likely to stay under 11 days while engaging KNN's predictions more heavily when extended stays are suspected.

Looking ahead, these models could become the foundation for a practical decision support system (Figure 1.1) in ports. Such a system could work with live container tracking data to provide different levels of alerts and recommendations. When Logistic Regression shows high confidence in a short-stay prediction, the system could automatically suggest resource allocation plans. Meanwhile, if KNN flags a container as potentially staying longer, it could trigger early attention from port operators.

Chapter 6

Limitations

Considering its findings and possible applications, some limitations that should be acknowledged are related to this dissertation.

6.1 Stakeholder Information Limitations

The model training could be improved by adding more features related to operators or companies handling containers. Information regarding container origins and destinations or details about transportation companies were unavailable. This lack of stakeholder-specific data restricts finding patterns related to operators and company-specific operational tendencies.

6.2 Cargo Content Information

The research did not use cargo information, presenting another notable limitation. With no access to data related to container contents and cargo priority, the training process could not learn from that information. This limitation affects the understanding of content-specific dwell time patterns, prevents analysis of differences between high-priority and standard cargo handling, and limits the identification of content-based operational requirements that might influence container dwell times.

6.3 Environmental Data Integration

Integrating external official datasets to capture weather effects in port operations was challenging. The weather dataset for 2023 was not available at the moment this project was performed, making it hard to see how weather conditions might influence container movements. While the port provided some weather information, like the recorded wind data, this did not significantly affect how well the models worked. Adding additional features related to weather conditions could allow the models to predict container dwell times better. Future research could look into how heavy rain or extreme temperatures affect dwell times by using official data to make predictions even more accurate.

6.4 Temporal Data Constraints

The training data was limited to nine months, from the middle of March to December 2023, while testing data covered only January through March 2024. Without a complete annual cycle in the training data, this restriction could impact the model's capabilities to identify annual patterns in container movements for long-term trends in port operations to capture seasonal variations that might occur throughout the year.

6.4.1 Time Series Analysis

Having just one year of data (2023-2024) limits the use of powerful prediction tools that work with patterns over time. These time series models need several years of data to spot reliable patterns like seasonal changes or yearly trends in how containers move through the port. For example, the training and analysis could not see how summer peaks compare to winter lows across different years or how holiday seasons affect container stay year after year.

While the current models work well for range predictions, having more years of historical data would let us use specialized tools with different approaches to generate insights for longer-term patterns. A time series analysis could help the Ports of Auckland better understand busy seasons and quiet periods and create new features for training. It is something worth looking into once more historical data becomes available.

6.5 Scheduling Information Gaps

The absence of scheduled pickup information presented another significant constraint in this research. It is worth mentioning that was why predictions were necessary.

Chapter 7

Conclusion

This research revealed significant findings regarding container dwell time prediction at the Ports of Auckland. It identified temporal patterns and specific characteristics related to the containers that influence dwell times. None of the weather-related variables, such as wind conditions, recorded by the port demonstrate significant predictive gain for the implemented models.

The study established three meaningful dwell time categories that proved effective for predictive modeling:

- Short-term (0-3 days)
- Medium-term (4-11 days)
- Long-term (12+ days or more)

The proposed predictive models had a robust performance, with F1 Scores exceeding 73% for most weeks in the analysis period 2024. Only one week showed a lower performance of 69% in the F1 score, which is still a good metric for its predictive power. Among the machine learning algorithms evaluated, Logistic Regression emerged as the most effective approach, followed closely by K-Nearest Neighbors.

These models can become critical components to be integrated into a Decision Support System (DSS) workflow to know how long a container will stay, which is essential in automating container allocation decisions. The Ports of Auckland could significantly enhance its operational efficiency by integrating them to address yard optimization challenges with data-driven solutions, which aligns with the primary goals of this project.

These models are highly reliable for operation planning, given that their performance suggests their predictive power is significant. The following valuable steps are implementing them in a real-time DSS environment and evaluating their performance in terms of time, costs, and rehandles.

7.1 Future Work

This research has established a foundation for predicting container dwell times that could help improve yard optimization at the Ports of Auckland. From data movements, promising avenues for future research have been identified that could improve the effectiveness and applicability of this work.

7.1.1 Granular Yard-Level Analysis

The current models operate at a port-wide level. However, implementing a more granular yard-level analysis provides new research opportunities through the development of location-specific predictive models. These models would account for individual yard section characteristics, specific operational constraints of different areas, and local traffic patterns and accessibility. At this level, which is more granular, it could provide better decision-making and practical and specific yard optimization strategies.

7.1.2 Extended Training Data Implementation

The current models were trained with nine months of 2023 data. Expanding the training dataset could improve metrics by including additional months (January, February, and March). By adding these months, 2023 will be complete for container movements, capturing missing seasonal patterns at the beginning of the year and strengthening the model's understanding of monthly trends. This temporal expansion could improve the model and its performance for the testing dataset and its months of January, February, and March.

7.1.3 Weather Data Integration

Data augmentation through meteorological data could enhance prediction metrics by integrating key weather features, including precipitation patterns and visibility data. This would enable a comprehensive analysis of weather-related impacts on container handling efficiency and dwell time variations. These new features could provide valuable insights into environmental factors affecting port operations.

7.1.4 Model Simulation Framework

The evaluation of model-driven decisions in container yard operations presents significant challenges due to the complex nature of container movements. Testing the effectiveness of predictive models in reducing unnecessary movements is more complex than comparing simple before-and-after scenarios. Each container's movement can affect multiple others, creating a ripple effect throughout the yard that's difficult to track and quantify. For instance, moving one container might require shifting several others, and these interdependencies multiply rapidly across thousands of containers.

Current data allows tracking actual movements, but simulating alternative scenarios - what could have happened if containers were placed differently based on predicted dwell times - remains challenging. This testing environment would need to copy actual port behavior: how putting a container in one spot limits where others can go, how knowing a container's expected stay time affects where it should be placed, and how all these choices add up over days and weeks of operation. Simulations would allow ports to experiment with different container management strategies, helping to prove whether these prediction models make ports run more smoothly and efficiently.

7.1.5 DSS Implementation

A practical next step would be building a user-friendly Decision Support System (DSS) that puts these prediction models to work in day-to-day port operations. This system would take real-time data about containers, run it through our models (mainly using Logistic Regression backed up by KNN), and give yard operators straightforward suggestions about where to place containers. The idea is simple but powerful - if it is known how long a container will likely stay, making decisions can be more thoughtful about where we put it. A container leaving soon should be easily accessible, while one staying longer can go to a less convenient spot. By tracking how many containers move, what we avoid, and how much faster we can get containers in and out, metrics could show exactly how much time and money these predictions save. This real-world testing would be the best way to prove whether these models can make ports run more efficiently.

7.1.6 More sofisticates classifiers

Adding new prediction models - like Neural Networks - could help spot patterns in container movements that simpler methods might miss. Adding a third model to work alongside the current two could create a voting system where the models work together to make better predictions.

Appendix A

Appendices, List of Figures, List of Tables

A.1 Model Optimization

A.1.1 Hyperparameter Tuning

Parameter	**Description**	**Values**
n_estimators	Number of trees in the forest	100, 200, 500, 1000
max_depth	Maximum depth of the trees	None, 10, 20, 30
min_samples_split	Minimum samples required to split an internal node	2, 5, 10
min_samples_leaf	Minimum samples required at a leaf node	1, 2, 4
max_features	Features to consider for best split	'auto', 'sqrt', 'log2'
bootstrap	Whether bootstrap samples are used	True, False

Table A.1: Random Forest Classifier Hyperparameters

Parameter	**Description**	**Values**
C	Inverse of regularization strength	0.001, 0.01, 0.1, 1, 10, 100
penalty	Type of regularization	'l1', 'l2', 'elasticnet', 'none'
solver	Optimization algorithm	'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'
class_weight	Weights associated with classes	None, 'balanced'
l1_ratio	Elastic-Net mixing parameter	0.0, 0.5, 1.0

Table A.2: Logistic Regression Hyperparameters

Parameter	**Description**	**Values**
n_estimators	Number of boosting stages	100, 200, 500
learning_rate	Contribution shrinkage of each tree	0.01, 0.1, 0.05
max_depth	Maximum depth of regression estimators	3, 5, 10
min_samples_split	Minimum samples required to split node	2, 5, 10
min_samples_leaf	Minimum samples required at leaf node	1, 2, 4
subsample	Fraction of samples for base learners	0.8, 1.0
max_features	Features to consider for best split	’auto’, ’sqrt’, ’log2’

Table A.3: Gradient Boosting Classifier Hyperparameters

Parameter	**Description**	**Values**
n_neighbors	Number of neighbors to use	3, 5, 7, 10, 12, 14, 20, 23, 26, 40
weights	Weight function used in prediction	’uniform’, ’distance’
metric	Distance metric for the tree	’euclidean’, ’manhattan’, ’minkowski’

Table A.4: K-Nearest Neighbors Hyperparameters

A.1.2 Best Parameters

Model	**Parameter**	**Value**
Random Forest	bootstrap	true
	max_depth	20
	max_features	“sqrt”
	min_samples_leaf	2
	min_samples_split	2
	n_estimators	500
K-Nearest Neighbors	metric	“manhattan”
	n_neighbors	23
	weights	“distance”
Gradient Boosting	learning_rate	0.1
	max_depth	5
	max_features	“sqrt”
	min_samples_leaf	1
	n_estimators	500
Logistic Regression	C	1
	class_weight	null
	l1_ratio	0.0
	penalty	“l2”
	solver	“newton-cg”

Table A.5: Best Hyperparameters by Model

A.2 Feature Importance Analysis

Feature	**Importance**	**Impact Description**
Operational Categories		
category_THRGH	2.688	Strongest positive correlation, indicating through containers significantly predict shorter dwell times
category_ TRSHP	-1.195	Strong negative correlation, suggesting transshipment containers tend toward longer dwell times
category_STRGE	-1.101	Significant negative influence, indicating storage containers typically have extended dwell times
category_EXPRT	-1.078	Notable negative correlation for export containers
category_ IMPRT	0.686	Moderate positive influence on dwell time predictions
Container Characteristics		
nominal_length_ NOM45	-0.385	Strongest negative correlation among container sizes
nominal_length_NOM20	0.197	Moderate positive influence for 20-foot containers
nominal_length_NOM40	0.189	Similar positive influence as 20-foot containers
freight_kind_FCL	0.147	Positive correlation for full containers
freight_kind_ MTY	-0.147	Corresponding negative influence for empty containers
Temporal Features		
time_in_ weekday	-0.264	Most influential temporal feature, indicating weekly patterns
time_in_business_day	-0.059	Modest negative influence of business day status
time_in_week_of_year	0.040	Minor positive correlation
time_in_hour	0.020	Slight positive influence of hour of day
time_in_week_of_month	-0.011	Minimal negative influence
time_in_month	-0.010	Minimal negative influence
Other Features		
requires_power	-0.041	Slight negative correlation for powered containers

Table A.6: Logistic Regression Model Feature Importance Analysis

Bibliography

[1] Denise Lindstrom Bandeira, João Luiz Becker, and Denis Borenstein. “A DSS for integrated distribution of empty and full containers”. In: *Decision Support Systems* 47.4 (2009), pp. 383–397.

[2] Mohammad Bazzazi, Nima Safaei, and Nikbakhsh Javadian. “A genetic algorithm to solve the storage space allocation problem in a container terminal”. In: *Computers & Industrial Engineering* 56.1 (2009), pp. 44–52.

[3] Bram Borgman, Eelco Van Asperen, and Rommert Dekker. “Online rules for container stacking”. In: *OR spectrum* 32 (2010), pp. 687–716.

[4] Héctor J Carlo, Iris FA Vis, and Kees Jan Roodbergen. “Storage yard operations in container terminals: Literature overview, trends, and research directions”. In: *European journal of operational research* 235.2 (2014), pp. 412–430.

[5] Lu Chen and Zhiqiang Lu. “The storage location assignment problem for outbound containers in a maritime terminal”. In: *International Journal of Production Economics* 135.1 (2012), pp. 73–80.

[6] Rommert Dekker, Patrick Voogd, and Eelco Van Asperen. “Advanced methods for container stacking”. In: *Container Terminals and Cargo Systems: Design, Operations Management, and Logistics Control Issues* (2007), pp. 131–154.

[7] Myriam Gaete et al. “A novel storage space allocation policy for import containers”. In: *Operations Research and Enterprise Systems: 6th International Conference, ICORES 2017, Porto, Portugal, February 23–25, 2017, Revised Selected Papers 6*. Springer. 2018, pp. 293–316.

[8] Virgile Galle et al. “The stochastic container relocation problem”. In: *Transportation Science* 52.5 (2018), pp. 1035–1058.

[9] Fabio García. *Predictive Container Dwell Time Modeling for Optimized Port Yard Placement*. GitHub repository. 2024. URL: https://github.com/fgar174/UOA-fgar174_dissertaion.

[10] AH Gharehgozli et al. “A decision-tree stacking heuristic for large scale reshuffling problems at a container yard”. In: *International Journal of Production Research* 52.9 (2014), pp. 2592–2611.

[11] Marc Goerigk, Sigrid Knust, and Xuan Thanh Le. “Robust storage loading problems with stacking and payload constraints”. In: *European Journal of Operational Research* 253.1 (2016), pp. 51–67.

[12] Roberto Guerra-Olivares et al. “A heuristic procedure for the outbound container space assignment problem for small and midsize maritime terminals”. In: *International Journal of Machine Learning and Cybernetics* 9 (2018), pp. 1719–1732.

[13] Ceyhun Güven and Deniz Türsel Eliiyi. “Modelling and optimisation of online container stacking with operational constraints”. In: *Maritime Policy & Management* 46.2 (2019), pp. 201–216.

[14] Ioanna Kourounioti, Amalia Polydoropoulou, and Christos Tsiklidis. “Development of models predicting dwell time of import containers in port container terminals–an Artificial Neural Networks application”. In: *Transportation Research Procedia* 14 (2016), pp. 243–252.

[15] Pasquale Legato and Rina Mary Mazza. “A decision support system for integrated container handling in a transshipment hub”. In: *Decision Support Systems* 108 (2018), pp. 45–56.

[16] Dung-Ying Lin, Yen-Ju Lee, and Yusin Lee. “The container retrieval problem with respect to relocation”. In: *Transportation Research Part C: Emerging Technologies* 52 (2015), pp. 132–143.

[17] Yanbin Liu et al. “A decision support system using soft computing for modern international container transportation services”. In: *Applied Soft Computing* 10.4 (2010), pp. 1087–1095.

[18] Sebastián Maldonado et al. “Analytics meets port logistics: A decision support system for container stacking operations”. In: *Decision Support Systems* 121 (2019), pp. 84–93.

[19] Filip Merckx. “The issue of dwell time charges to optimize container terminal capacity”. In: *Proceedings IAME 2005 Annual Conference, Limassol, Cyprus, 23-25 June 2005*. 2005, CD–ROM.

[20] Nadereh Moini et al. “Estimating the determinant factors of container dwell times at seaports”. In: *Maritime economics & logistics* 14 (2012), pp. 162–177.

[21] Katta G Murty et al. “A decision support system for operations in a container terminal”. In: *Decision support systems* 39.3 (2005), pp. 309–332.

[22] Etsuko Nishimura et al. “Container storage and transshipment marine terminals”. In: *Transportation Research Part E: Logistics and Transportation Review* 45.5 (2009), pp. 771–786.

[23] Taejin Park et al. “Dynamic adjustment of container stacking policy in an automated container terminal”. In: *International Journal of Production Economics* 133.1 (2011), pp. 385–392.

[24] Jana Ries, Rosa G González-Ramírez, and Pablo Miranda. “A fuzzy logic model for the container stacking problem at container terminals”. In: *Computational Logistics: 5th International Conference, ICCL 2014, Valparaiso, Chile, September 24-26, 2014. Proceedings 5*. Springer. 2014, pp. 93–111.

[25] Jean-Paul Rodrigue and Theo Notteboom. “The terminalization of supply chains: reassessing the role of terminals in port/hinterland logistical relationships”. In: *Maritime Policy & Management* 36.2 (2009), pp. 165–183.

[26] Jyoti Sharma and Ravi Shankar Singhal. “Genetic algorithm and hybrid genetic algorithm for space allocation problems-a review”. In: *International Journal of Computer Applications* 95.4 (2014), pp. 33–37.

[27] WS Shen and CM Khoong. “A DSS for empty container distribution planning”. In: *Decision Support Systems* 15.1 (1995), pp. 75–82.

[28] Andresson da Silva Firmino, Ricardo Martins de Abreu Silva, and Valéria Cesário Times. “A reactive GRASP metaheuristic for the container retrieval problem to reduce crane’s working time”. In: *Journal of Heuristics* 25 (2019), pp. 141–173.

[29] Evrim Ursavas. “A decision support system for quayside operations in a container terminal”. In: *Decision Support Systems* 59 (2014), pp. 312–324.

[30] KM Van Hee and RJ Wijbrands. “Decision support system for container terminal planning”. In: *European journal of operational research* 34.3 (1988), pp. 262–272.

[31] Bart Van Riessen, Rudy R Negenborn, and Rommert Dekker. “Real-time container transport planning with decision trees based on offline obtained optimal solutions”. In: *Decision Support Systems* 89 (2016), pp. 1–16.

[32] Fan Wang and Andrew Lim. “A stochastic beam search for the berth allocation problem”. In: *Decision support systems* 42.4 (2007), pp. 2186–2196.

[33] Chuqian Zhang et al. “Storage space allocation in container terminals”. In: *Transportation Research Part B: Methodological* 37.10 (2003), pp. 883–903.