Master thesis - IEM

Driving last-mile delivery efficiency at Picnic by predicting travel times

Marnick Plomp

Picnic Technologies

July 16, 2024

Driving last-mile delivery efficiency at Picnic by predicting travel times

A thesis written as graduation assignment of the master Industrial Engineering & Management, at the University of Twente. The assignment was executed at Picnic Technologies.

This is a publicly available version of this thesis. Therefore, we removed axes in some figures due to the sensitive information they provide. For that same reason, some in-text numbers, tables and appendices are adjusted; sensitive information is hidden or multiplied with a random number such it does not represent reality anymore.

Author
M.A. Plomp (Marnick)
s2112957

Utrecht, 16 July 2024

University of Twente
Drienerlolaan 5
7547 RW, Enschede
Netherlands

First supervisor
Dr. Ir. M.R.K. Mes
Faculty of Behavioural, Management and Social Sciences (BMS),
Industrial Engineering & Business Information Systems (IEBIS)

Second examiner
Dr. Ir. E.A. Lalla-Ruiz
Faculty of Behavioural, Management and Social Sciences (BMS),
Industrial Engineering & Business Information Systems (IEBIS)

Picnic Technologies B.V.
Van Marwijk Kooystraat 15
1114 AG, Amsterdam
Netherlands

Company supervisor
Msc. M. Seggewiße
Business analyst
Picnic Technologies B.V.

Number of pages excluding appendices and references: 53
Number of pages including appendices and references: 64
Number of appendices: 8

Preface

You are about to read the master thesis 'Driving last-mile delivery efficiency at Picnic by predicting travel times'. This thesis was the last step to graduate from the Master Industrial Engineering & Management at the University of Twente. The past six years as a student were amazing, and graduating by means of this thesis marks the end of this period. This project started in February 2024, when I moved from Enschede to Utrecht to be within reach from Picnic's HQ in Amsterdam.

I want to thank Picnic for the opportunity to perform this project at an amazing, fast-growing, and open company. They took really good care of me, and I loved the culture and entrepreneurial mindset at the company from day one. I especially want to thank my main supervisor Michael Seggewiße for his engagement, guidance and input during this research. We had many sparring sessions and discussed how to approach this project guaranteeing both academic soundness, and practical relevance. Even though Michael was located in Dusseldorf Germany, we made it work really smoothly and I really appreciate his efforts and investments. Also, a big thanks to Arjan Braemer and Floor ten Damme. Arjan consistently asked critical questions, to enhance the thoroughness while also challenging me to take everything one step further. Floor helped me a lot with decisions on how to approach obstacles, and which findings were important to generate, and how to exhibit those.

Additionally, I was very fortunate to have Martijn Mes as lead supervisor from the University of Twente. He helped me with critical feedback to improve quality, but often also inspired me to research new aspects to generate even more relevant insights. Furthermore, he continuously confirmed that I was on the right track which made sure I felt comfortable throughout the entire process. And I am grateful to Eduardo Lalla-Ruiz, for assessing my work as member of the examination committee.

Finally, I want to thank my friends, who were also executing their graduation project. It was really nice to discuss challenges and assess each other's work. This peer feedback really helped improving the clarity of this thesis.

I wish you a pleasant read.

Marnick Plomp
Utrecht, June 2024

Management Summary

This research is conducted at Picnic Technologies, an e-grocer company operating in The Netherlands, Germany and France, which delivers groceries directly to customers' homes without physical stores. The focus is on optimizing the last-mile delivery of Picnic, where each trip starts from a hub and covers surrounding areas using an electric Picnic vehicle (EPV).

In the delivery process, EPVs drive between customers (drive segments) and stop to park and to drop off groceries (park and drop segments). Customers select delivery timeslots of either 1 or 1,75 hours for upcoming days and must be home during delivery. On the delivery day, Picnic communicates a twenty-minute time window (TW) around the estimated arrival time, defined as [-5; +15] from this estimate.

The planning of the last-mile delivery at Picnic results in significant errors since the actual segment times differ too much from planned segment times. This leads to too low performance of the main business KPIs *minutes per delivery* (M/D) and *on-time %* (OT). Hence, the formulated research objective is:

"Develop a method that increases planning accuracy to reduce the minutes per delivery and/or boost the on-time delivery %."

A delivery is considered on-time if it arrives within the communicated twenty-minute TW. Picnic aims to enhance efficiency (by reducing *minutes per delivery*, M/D) or improve *on-time* performance (OT) without compromising either. This research focusses on improving the planning accuracy of drive segments, as Picnic tends to overestimate drive times. We apply machine learning techniques to predict drive times, aiming to reduce M/D while maintaining OT.

Picnic currently uses drive time estimates from Mapbox, an external company. However, due to the special Picnic vehicle and the way they track segment times, these estimates are not as accurate as needed. Hence, Picnic modifies the from/to drive time matrix by multiplying all travel times by a factor. This factor is the slope of a linear regression between the 8 weeks historical actuals and Mapbox' originally planned drive times for those weeks in a specific dispatch plan. A dispatch plan includes all customers in a neighbourhood served in the same shift. The adjusted drive time matrix is then used in the vehicle routing optimization to determine the routes of the trips in that dispatch plan.

After a literature review, we developed and compared five prediction models: four common machine learning techniques and a linear regression. The models used location features, such as urbanity degree, postal code, car density and timing features such as shift and holiday week. Additionally, we included Mapbox' estimates since the from/to combinations are rarely identical. We trained and tested the models on nine representative hubs in the Netherlands (NL), using the MAE, MAPE and R^2-score to evaluate performance. With the best model, we reconstructed trips with the improved drive time predictions to assess their impact on trip planning, and OT and M/D.

The neural network (NN) dominated the other prediction models on all performance metrics and was significantly more accurate than Picnic's original approach, reducing the MAPE from 46,12% to 33,71% for test weeks 16 and 17. So, on individual drive segments, the NN is significantly more accurate. However, a potential bias in the NN's predictions on single drive segment could accumulate and negatively impact performance if we aggregate over all drive segments in a trip. Hence, for practical relevance, we aggregate to total trip drive errors, to learn how NN predictions impact the total trip performance. The total trip drive MAPE reduced from 17,74% to 12,09% with the NN. The median trip error (total trip actual – total trip predicted drive time) changed from -118,5 to -26 seconds. This indicates that for at least 50% of trips, the actual drive time is shorter than the planned drive time, i.e., we still plan a buffer in the drive segment.

The IQR of the total trip drive errors changed from 248 to 209 seconds, implying that apart from the more centred distribution, also the errors are less spread, improving overall trip accuracy.

With the NN predicted drive times, we reconstructed the trips. The key result is that we reduced the planned M/D with 6,8%, driven by shorter drive times and less blocking of quick runners. Quick runners outperform planning with more than 5 minutes. After some undesired waiting time, these runners will deliver in the first minute of the TW, as they are already at the customer when the TW opens. With the NN, the first-minute deliveries decreased from 10,84% to 3,66%, accounting for 2,55% of the M/D improvement.

However, tighter predictions make it harder for the slower runners to stick to planning. Resulting in an OT decrease of 1%. Slow runners relied on the original drive time buffer, but this is largely removed. The risk of lates increased, especially later in trips. Hence, we briefly analysed the potential of variable (and extending) TW placement to boost OT. This could improve the main KPIs significantly, but the practical implications for operations and customer satisfaction are yet unknown.

We extended the dataset to all 60 hubs in NL, to obtain country-wide results. The total trip MAPE reduced from 16,21% to 11,94%. The median total trip drive error changed from -108 to -12 seconds. The IQR changed from 241 to 220 seconds. We reduced the planned M/D with 6,9%, and the first-minute deliveries decreased from 12,50% to 4,07%. The OT decreased with 0,92%. These results mirrored the initial nine hubs, showing improved M/D at the cost of a slightly reduced OT.

For implementation, Picnic's distribution team should collaborate with the machine learning department, which will construct and maintain the NN. A similar collaboration already exists for predicting drop times. Therefore, the change this collaboration requires seems relatively small. Adopting a similar architecture to the drop time model is recommended to enhance last-mile planning accuracy and efficiency.

The limitation of this research is that tighter planning of compensatory drive segments can unbalance the overall trip, as drop and park segments are already tightly planned. Hence, improvements in drive time predictions should be accompanied by enhancements in other segments to further boost OT and M/D. Therefore, we recommend Picnic to investigate these segments and to explore the potential of variable and extended TWs. Especially at customers early in the trip many runners must wait, while at the end of the trip the risk of late a delivery increases. The practical implications and consequences for customer satisfaction should be known before deciding to implement drastic changes such as variable or extended TWs.

Abbreviations

Abbreviation	Full Form
AAA	Advanced analytics & algorithms
AHD	Attended home deliveries
cv	Cross validation
DCs	Distribution centers
DT	Decision tree
DTF	Drive time factor
EPV	Electric Picnic Vehicle
ETA	Estimated time of arrival
FCs	Fulfilment centers
IQR	Interquartile range
KPI	Key performance indicators
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ML	Machine learning
MPP	Master planning process
M/D	Minutes per delivery
NL	Netherlands
NN	Neural network
OT	On-time percentage
RQ	Research question
RF	Random forest
SVRPTW	Stochastic vehicle routing problem with time windows
TW	Time window
VROOM	Vehicle routing omnigenous optimization method
VRP	Vehicle routing problem
WA	Weighted average
XGB	Extreme gradient boost

Table of Contents

1. Introduction

This chapter provides the context of this master's thesis, conducted at Picnic Technologies in Amsterdam. It covers the company introduction (1.1), problem description and research motivation (1.2), and the proposed research design (1.3).

1.1. Company description

Picnic Technologies, often referred to as Picnic, is an online grocery supermarket launched in the Netherlands in 2015. Distinguished by its app-only approach, Picnic revolutionizes grocery shopping by delivering orders directly to customers' homes. Utilizing electric Picnic vehicles (EPVs), Picnic eliminates the need for physical stores, offering convenience and time savings to customers. Dubbed the modern milkman, Picnic operates in densely populated urban areas and visits customers at home, like the traditional milkman. Currently spanning over 120 cities across the Netherlands, Germany, and France, Picnic's expansion is driven by demand; cities are added based on waiting list size and local interest.

Figure 1. Example of a runner delivering groceries with the EPV.

Apart from just delivering groceries to customers, Picnic sets itself apart from competitors with its unique approach of managing a large part of the end-to-end supply chain in-house. As Figure 2 illustrates, Picnic manages distribution centres, fulfilment centres, hubs and the transportation network internally.

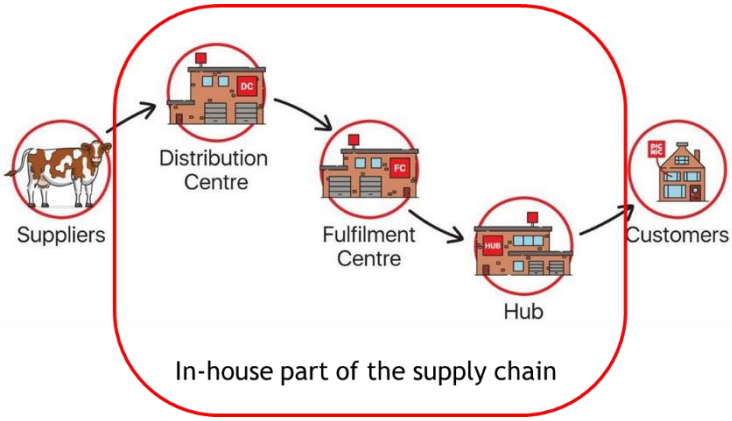

Figure 2. Flow of groceries through Picnic's supply chain.

The distribution centres (DCs) receive bulk groceries for sorting and storage before transferring to fulfilment centres (FCs). At FCs, customer orders are picked and packed into totes for delivery (see Figure 1). Totes are stored in racks and transported to hubs. At hubs, one rack with ambient and one with chilled/frozen totes are loaded onto EPVs for delivery by runners. In the Netherlands only, Picnic operates four DCs, seven FCs, and 60 hubs.

Picnic performs a wide range of activities, such as product forecasting, warehouse fulfilment, EPV design engineering, internal supply chain management and software development. By integrating this high-tech in-house philosophy with a vision for sustainability, flexibility and efficiency, Picnic offers the complete recipe for revolutionizing the grocery supply chain.

1.2. Problem description

The problem at hand concerns the planning accuracy of the last-mile deliveries and its impact on two main KPIs: *on-time deliveries* (also called *service* level) and *minutes per delivery*. Customers place orders through the Picnic app and after the planning process (dealt with in Chapter 2) they receive a twenty-minute time window determined by Picnic's sophisticated planning algorithm. This algorithm considers travel and service times between customers to estimate the time of arrival (ETA) at each customer. The *service level* is measured by the percentage of deliveries made within this twenty-minute window, a common metric in attended home delivery services (Agatz et al., 2008). Picnic's current planning accuracy is unsatisfactory, leading to insufficient performance of the two main KPIs. With planning, we mean the predicted time planned for performing deliveries. Each delivery has specific planned travel and service times, used to create the trip planning.

On the one hand, too tight planning leads to time windows that are hard to meet and therefore, could result in a bad *service level*. Too loose planning, on the other hand, could lead to runners arriving before the window opens, resulting in waiting and wasting valuable time. Improving planning accuracy could therefore increase the *service level* and reduce *minutes per delivery*, which in turn saves costs.

Each hub's performance is monitored based on these KPIs, i.e. *on-time %* and *minutes per delivery* and compared against Picnic's targets. The *on-time %* target is a weighted average across all hubs, considering the varying number of deliveries each hub handles. However, Picnic also aims to enhance the service level of individual hubs if it falls below expectations. The *minutes per delivery* targets are hub specific as delivery characteristics differ significantly between hubs, making a single target unfair.

1.2.1. Research motivation

From the previous section it is clear that the problem at hand focusses on last-mile deliveries. The last mile is, according to Waßmuth et al. (2023), generally recognized as the most challenging part of the fulfilment process, especially for attended home deliveries (AHD). In an AHD context, customers must be present to receive the goods and this concept is well established in the online grocery retailing, as the attended delivery is necessary because goods are perishable (Agatz et al., 2008). The e-grocery-sector is particularly challenging due to the low profit margins and the special care that is required for delivering (sometimes frozen) food to consumers, let alone the risk of the customer not being present during the planned delivery. Waßmuth et al. (2023) state that many online supermarkets struggle with being profitable while they also have a large desire for high customer service levels.

Picnic is also committed to providing high customer service, driven by its ambition to expand its market presence. Providing high customer service, by means of reliable grocery deliveries is vital to ensure repeat customers. Also, the deliveries should be as efficient as possible to reduce cost and increase profit margins. These two drivers, the desire for (i) reliable grocery deliveries and (ii) efficient last-mile deliveries, are drivers for a strong foundation that allows for the desired future growth.

The key motivator for this research is to improve the efficiency level by predicting travel and/or service times more accurate, while not deteriorating the *on-time delivery* percentage, or vice versa. Scheduling more time per delivery, could increase the *on-time delivery* percentage, as this results in delivery time windows that are easier to meet. However, such a solution reduces the last-mile efficiency since the main efficiency KPI increases: *minutes per delivery*.

1.2.2. Action problem

The presented problem is that the planning of the last-mile deliveries is not accurate enough. In this subsection, we quantify the problem in terms of an action problem. An action problem is a discrepancy between a (desired) norm and reality as the problem owner perceives it (Heerkens & van Winden, 2017). The problem owner is the distribution team of Picnic, as the *on-time delivery %* and *minutes per delivery* are their responsibilities. The action problem is formulated as:

The planning of the last-mile deliveries is not accurate enough, as the minutes per delivery and the fraction of on-time deliveries do not meet the targets.

The *service level (=% of on-time deliveries)* target is 96% and the *minutes per delivery* has variable targets depending on country and hub.

1.2.3. Problem identification

It is important to find the underlying (core) problem(s) in the context. This is done via a problem cluster. A problem cluster is used to map all problems along with their connections, and after some steps the core problem can be identified (Heerkens & Winden van, 2017).

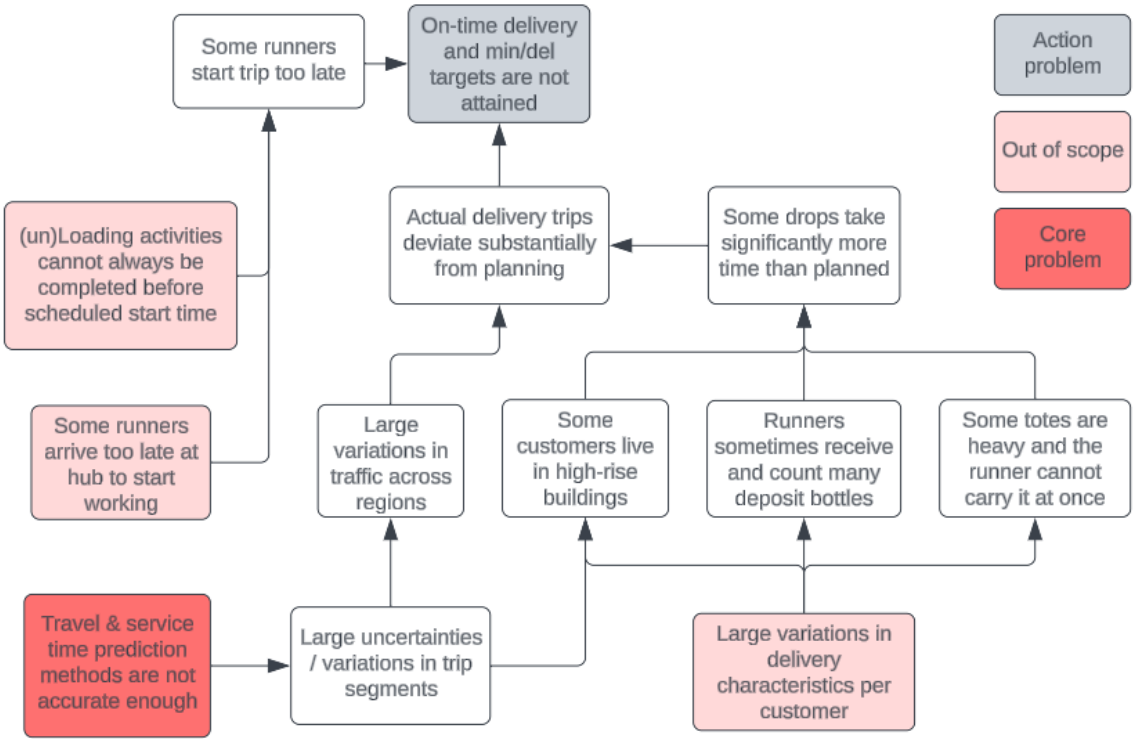

Figure 3. Problem cluster with all relevant problems at Picnic.

In Figure 3, three problems cannot be influenced by this research and are therefore out of scope. The (un)loading activities are restricted by hub traits, think of number of loading docks and available space. These traits cannot be influenced by the distribution team of Picnic. The same holds for the other two pink problems: late arrival of runners and the variations in delivery characteristics. Although they cannot be influenced, they are important to consider when solving other problems.

The core problem (Heerkens & Winden van, 2017) to be addressed in this research is that the prediction methods for the last-mile delivery segments are not accurate enough, which eventually leads to large deviations with the actual trip times and hence underperformance of *minutes per delivery* and *on-time deliveries*. The prediction methods provide the data used as input for the planning method of the delivery trip (outlined in Section 2.3). From Figure 3 shows that there is quite some uncertainty in Picnic's last-mile delivery, and the current prediction methods do not seem to capture that uncertainty well enough.

1.3. Research design

This section provides the research design. First the objective and main research question are formulated, then the approach is outlined by means of sub-questions and finally, the research scope is set.

1.3.1. Research objective

Clear from the problem statement is that the planning accuracy at Picnic should be improved. This could result in more efficient deliveries which should increase profitability, or better on-time performance as reality is approximated better. Hence, we formulate the research objective:

"Develop a method that increases prediction accuracy to reduce the minutes per delivery and/or boost the on-time delivery %."

This objective directly results in the main research question (RQ) of this thesis:

"How to optimize the last-mile delivery planning/predictions at Picnic, to minimize delivery time and increase on-time deliveries?

1.3.2. Research approach

The main research question serves as the vital point for achieving the research objective. To address this main question effectively, the report outlines a sequence of intermediate questions. These (sub-)RQs act as steps towards answering the main RQ. They are formulated with corresponding thesis chapters:

Chapter 2 – Current situation

1) What is the current situation at Picnic regarding the last-mile delivery process?
   - What steps are part of the last-mile delivery process and planning?
   - What is the current on-time and minutes per delivery performance?
   - How does the current planning process work and perform?

We will answer this RQ, through interviews with various company experts responsible for different aspects of last-mile planning or process. Additionally, performing actual delivery trips with runners will provide further insights into the current situation. Quantitative analysis will complement these efforts by diving into Picnic's data warehouse to extract relevant insights.

Chapter 3 – Literature review

2) What does literature tell us regarding improving last-mile attended-home deliveries?
   - What are key challenges in the planning of AHD settings?
   - What vehicle routing problems (VRP) exist, similar to Picnic's?
   - How to model uncertainty in travel and service times?

For the second RQ we will consult literature. We dive into AHD settings, VRP variants and modelling of uncertainties in VRPs. This exploration will reveal common scientific practices for creating last-mile delivery plans, considering various stochastic characteristics.

Chapter 4 – Solution method

3) How can we improve the planning accuracy at Picnic?
   - Which segment(s) should be improved?
   - What methods can be applied to predict(/plan) trip segment times more accurately?
   - What should the design of the proposed methods look like?

For the third RQ, we integrate our literature findings with the findings from the context analysis (RQ2), to determine a method to improve planning accuracy. We will research different prediction methods to improve planning accuracy of (a) specific segment(s).

Chapter 5 - Results

4) What performance can be expected from the best prediction method?
   - What is the predictive performance of each method, and which performs best?
   - How does the best method contribute to on-time and efficiency?

RQ4 focuses on the evaluation of the proposed methods. To evaluate the performance per model, we conduct experiments and calculate relevant metrics. Furthermore, the impact on the main KPIs, *on-time %* and *minutes per delivery*, is researched by reconstructing the trips in the experiments with new predicted segment times. It is vital to evaluate the impact of the best method on these KPIs to assess its practical relevance.

Chapter 6 - Implementation

5) How to implement the best method at Picnic?
   - How to set-up the required architecture to implement the method successfully?
   - What are the (long-term) practical implications?

This RQ deals with how to implement the best prediction method at Picnic, the steps that need to be taken to implement it and the required architecture. Furthermore, we also consider some practical implications for stakeholders that might occur due to the implementation.

1.3.3. Research scope

This research is scoped to Picnic in The Netherlands (NL) only. Picnic is well-established in NL, and therefore the focus on quality-of-service and efficiency is bigger than in France (FR) and Germany (GE), where business operations have not yet reached a steady-state but are focussed on growth. If results of this research lead to better performance in NL, the methodology can be extended to GE and FR as well, since eventually planning must be optimized there as well.

Furthermore, we want to improve the planning accuracy of (a) specific part(s) of the delivery trip. This implies that we will not look at any process before the departure of the EPV at hubs in NL. Of course, *on-time %* can probably be improved by, for example, increasing the reliability of truck deliveries from fulfilment centres to the hubs to ensure the EPVs depart on time, but that is outside the scope of this research. From now on, *minutes per delivery* (M/D) and *on-time %* (OT) are considered the main KPIs in this thesis.

2. Current situation

This chapter addresses the first research question by examining Picnic's current last-mile delivery situation and relevant aspects. Sections 2.1. and 2.2. introduce key terminology. Section 2.3. outlines the planning method and Section 2.4. exhibits insights regarding the main KPIs and planning accuracy.

2.1. Characteristics of a delivery trip

This section clarifies some key definitions used in this research regarding trip characteristics. We begin by segmenting delivery trips, then we look at the time terminology used at Picnic.

2.1.1. Trip segments

Picnic splits a delivery trip in multiple segments. This is done for a better understanding of the trips and for detailed planning per segment. Table 1 briefly states the segments and their definition.

Table 1. Explanation of Picnic's trip segments

Segment	*Definition*
Stem	Driving the EPV between the hub and the first/last customer
Park	Parking the EPV close to the customer's home, few meter radius from address
Drop	Delivering the groceries to the customer's home by foot
Drive	Driving the EPV between customers
Pre / Post	Time at hub for preparing and returning the EPV (e.g. load/unload)

In Table 1, the drop time is adjusted based on the runner's experience level. Starting runners receive a bit more time to get used to the job. After a certain number of deliveries, runners go from the starter level to intermediate and then to the experienced level.

Figure 4 illustrates the trip segments (Figure 4a) and tracks the time taken for each segment (Figure 4b), cumulatively forming the primary efficiency key performance indicator (KPI): *minutes per delivery*.

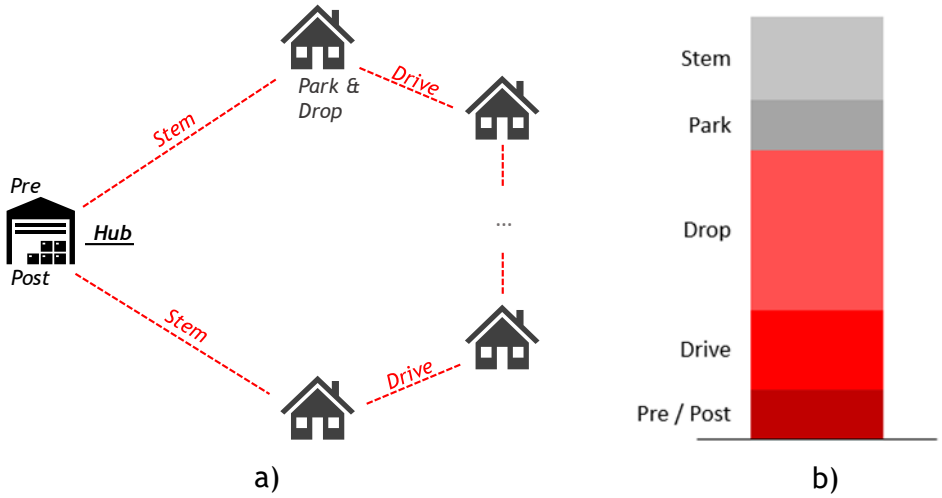

Figure 4. Visualisation of trip segments (a) and segment fraction per trip as % per delivery (b).

Currently, during an average trip, the two stem times collectively account for a similar fraction of the total *minutes per delivery* as all drive segments, despite there being only two stem segments in a trip. This relatively large fraction of stem time can be explained by the fact that hubs are usually located in industrial areas, resulting in considerable time spent driving to customer neighbourhoods. Conversely, the recurring drive times between customers are relatively short, as customers in a trip typically live close to one another.

The estimated time of arrival (ETA) at each customer in a trip, is calculated by summing the planned times of all preceding individual segments, before reaching the specific customer's front door. Then that sum is added to the trip's start time. Each segment towards every customer has a separately planned time. For example, to calculate the ETA at customer 2, we sum the pre-time, the stem time, the park and drop time at customer 1, the drive time to customer 2, and the park time at customer 2. Finally, a twenty-minute time window (TW) is placed around the ETA and communicated to the customer. Before May 2024, this window was placed symmetrical ([-10; +10]), but as of May 2024 this window switched to [-5; +15].

Runners perform trips with an electric Picnic vehicle (EPV). There are two types of EPVs: G4 and G6. The G4 is the most common EPV, and most hubs only have this type. The G6 is a larger EPV that can carry more totes, it can also drive faster and further. In the Netherlands (NL), only six hubs have G6 EPVs, from which two hubs have only G6 and four have both G4 and G6 EPVs.

The runner starts at a hub to load the EPV with the groceries of the customers in the trip (pre-time). Each hub serves (part of) a city, and the hubs vary in size depending on the delivery area they are serving. Some have 15 EPVs allocated, other hubs have 50. When an EPV is not driving around, it is usually recharging. During a shift, multiple runners perform a trip with on average 11-16 deliveries.

2.1.2. Shifts and time slots

An important time instance anticipated at the company is a *shift*. This is a timeframe on a day that Picnic uses to schedule runners and EPVs. The key take-away is that it covers the timeframe in which a runner can perform one delivery trip, including loading and unloading the EPV. Each hub operates multiple shifts on a day, allowing runners to perform multiple trips in adjacent shifts.

The first step of the last-mile delivery process at Picnic, is that customers order products in the app. At the end of their order, they choose a time slot in which they want their groceries delivered. This slot is typically on either of the coming few days. However, they cannot choose any time slot throughout a day, as Picnic offers a selection of time slots.

Which time slots are offered depend on a few factors. For most hubs it holds that Picnic operates two morning shifts (M1 and M2) and three afternoon shifts (S1, S2 and S3). All customers are offered both one morning and one afternoon slot. This implies that each neighbourhood is visited both in the morning and in the afternoon. Each shift takes approximately 2:30 hours of which 1 hour and 45 minutes is reserved for the drive, park and drop segments. The other 45 minutes are meant for pre/post and stem.

From the 1 hour and 45 minutes, customers can select the first hour, the second hour (15 min overlap, see Figure 5), or the full 1 hour and 45 minutes as time slot. Customers who choose this latter, grant Picnic more freedom in creating the routes and also contribute to more sustainable routes. The resulting enhanced freedom allows for more efficient routes and therefore less *minutes per delivery* (M/D).

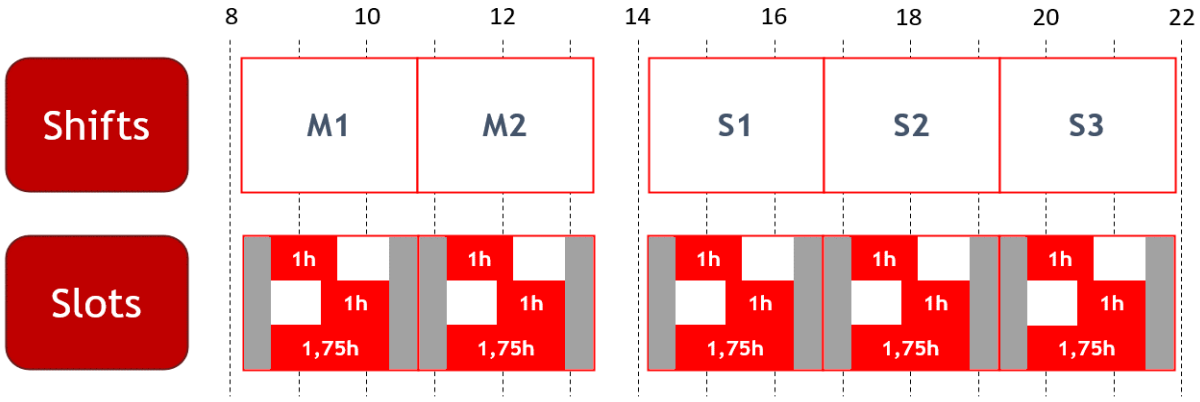

Figure 5. Shift and timeslot scheme for regular G4 shifts.

Some hubs also have extension shifts, next to the regular shift scheme of Figure 5. These shifts differ because their stem time is longer, as they serve areas further away. Extensions occur at hubs that serve a city, and small villages around it. The customers served by the extension shift, cannot be reached with the regular pre/post and stem time (grey parts in Figure 5), while still having the desired 1,75h of pure drive, park and drop time. Hence, more stem time is planned for extension shifts, basically extending the entire shift time a little bit. There are two extension shifts in the morning and two in the afternoon: EM1, EM2, E1 and E2 respectively. E-shifts overlap regular shifts.

For G4 EPVs, the spread of the number of deliveries over the weekdays is provided in Figure 6a. While Figure 6b exhibits the spread in delivery volume across the shifts.

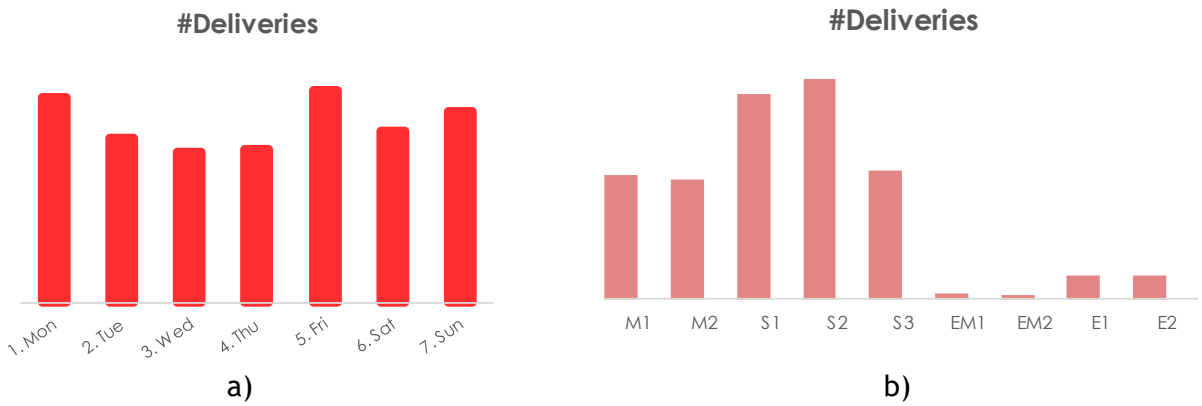

Figure 6. Spread of total G4 deliveries over weekdays (a) and shifts (b).

From Figure 6, we conclude that the company performs most deliveries on Friday, Monday and Sunday, and that the peak shifts are S1 and S2. It strikes that the extension shifts make up a small fraction of Picnic's total delivery volume. For G6 delivery trips, there is a similar spread over the weekdays and shifts as in Figure 6.

The costs of shifts are equal for Picnic in the direct sense; labour costs are equal for all shifts on all weekdays. However, peak shifts are causing indirect costs. As some EPVs are only procured to make sure the peak shifts can be performed. These EPVs are abundant in non-peak shifts. The same holds for indirect recruitment costs as there are more runners required just to handle the peak shifts. Further analysis regarding cost of timeslots is considered irrelevant for this research.

2.2. Dispatch plans

Each neighbourhood is visited by one shift in the morning (either of M1 or M2) and one shift in the afternoon (either of S1, S2 or S3), with each shift assigned to a dispatch area, representing a section of the service area. Picnic generates a daily dispatch plan for each dispatch area per hub the night before the delivery day. In the morning, a dispatch area accommodates 50% of the forecasted deliveries, while in the afternoon, it handles 33%, since there are two morning and three afternoon shifts. The assignment of dispatch areas to shifts, rotates daily to ensure variety, and to avoid visiting neighbourhoods at the same time every day, offering flexibility for customers' schedules. Figure 7 shows an example day in Maastricht.

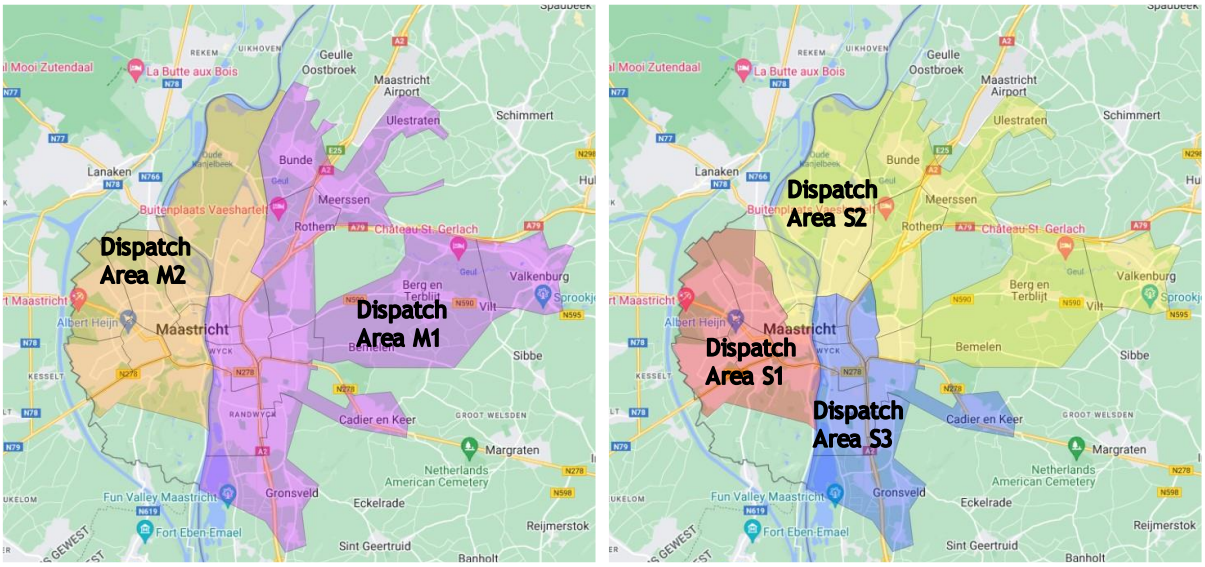

Figure 7. Example dispatch areas in Maastricht in morning and afternoon.

In Figure 7, customers in the purple area (M1) can receive groceries between 8.15 a.m. and 10.00 a.m., or in one of the afternoon shifts. In the app, customers can choose the day(part)-dependent timeslots, based on the rotating dispatch areas. On the delivery day, a customer receives a twenty-minute TW, either [-10; + 10] or, as of May 2024, [-5; + 15] as initiative to boost *on-time %* (OT).

The vehicle routing algorithm (discussed in Section 2.3) creates a dispatch plan per dispatch area (so also extension areas). A dispatch plan contains all delivery trips in the dispatch area. Either the G4 or the G6 EPVs are present in dispatch plan, not a combination. A dispatch plan contains the information as listed, for each EPV and hence trip:

- Start time of trip first trip segment (including pre time)
- Departure time at hub
- Route and sequence of visiting customers
- Opening time of time window per customer
- ETA per customer
- Drop time per customer
- Park time per customer
- Return time at hub
- End time of last trip segment (including post time)

2.3. Vehicle routing – VROOM

To create the dispatch plan for each dispatch area, Picnic runs their own vehicle routing algorithm, VROOM: the vehicle routing omnigenous optimization method. VROOM is method to solve a Vehicle routing problem (VRP) with time window and vehicle capacity constraints. Since each dispatch plan contains a single EPV type, VROOM is a VRP with homogeneous vehicles. As VROOM runs per dispatch area, the algorithm runs 60 (hubs) * 5 (daily dispatch areas per hub) = 300 times on average per day in NL. VROOM is part of the master planning process (MPP) of Picnic. MPP is the framework that facilitates the planning from a product in a fulfilment centre all the way to the customer. So, it needs to be picked in the right tote, the tote should be placed in the right truck, the truck should go to the correct hub etc.

Section 2.3.1 discusses VROOM's methodology and 2.3.2 deals with key drive and drop time inputs.

2.3.1. VROOM – Methodology

VROOM operates sequentially in three phases, per dispatch area:

1) *Initial solution generation*: This phase creates a feasible dispatch plan ensuring all customers receive their groceries within the chosen time slot. In this solution all customers are assigned to an EPV (and hence runner) and a sequence of customers per trip is created. The greedy heuristic used, assigns the customer that is furthest away from the hub to the runner with the most constrained return time. The return time is either the end time the runner agreed to work until, or the start of the next shift (plus break time).
2) *Runner minimization*: The algorithm improves the initial solution by minimizing the number of runners, thereby reducing total trips. It achieves this by removing a trip with its customers from the initial solution and re-inserts these customers in other trips depending on EPV capacity and time slots of the customers.
3) *Minimizing total driving* time: The last objective is to minimize the total driving time. Although minimizing the total time might seem logical, solely prioritizing this will favour experienced runners only, as they get tighter drop times than starters (Section 2.1.1) and hence total time reduces. The total drive time minimization is done by renowned heuristics:
    a. Moving customers, either to another position in the same trip or to different trips.
    b. Relocating a number of consecutive customers within the same route.
    c. Reversing the sequence of consecutive customers within the same route.
    d. Swapping two lists of consecutive customers between routes.

    Whenever either of these heuristics finds a better solution, it is accepted immediately. So, a first-accept approach is anticipated, rather than a best-accept approach.

The result of VROOM is a list of trips where each trip has a sequence of customers to visit. If we look at the regular G4 afternoon shifts, VROOM first runs for the dispatch plan in S3, then S2 and S1. As the start times of S3 imply constraints for the end times of S2 etc. If S3 is planned, S2 is planned such that it ends twenty minutes before the start time of S3. As runners require some legal break time. The same principle holds for S1 and other shift types.

As clear in the methodology, VROOM assigns runners to trips. However, after VROOM's execution, other systems within MPP can swap runners between trips to optimize breaktime. Also, runners swap about 20% of the trips among themselves. Thus, the assignment of specific runners to trips remains uncertain. Additionally, since customers often have specific delivery time preferences, the set of customers in a dispatch plan varies often. So, the routes VROOM generates are rarely identical. Implying that quite some drive segments are not travelled before, as the combination of from/to customer is usually unique.

The main inputs for VROOM are listed and described in Table 2.

Table 2. Inputs of VROOM

Input level	*Input*	*Description*
General	Drive time matrix	Drive times between all nodes (customers + hub) in the dispatch area.
	Set of available runners	This encompasses the number of runners, how many shifts they will perform and their experience level.
Per customer	Demand	Number of ambient and chilled/frozen totes.
	Location	Coordinates of the customer's home.
	Time slot	Time slot in which the EPV should arrive at the customer, 1h or 1,75hrs.
	Predicted stop time	Estimate of drop and park time, based on historical data as input for a neural network.
Per EPV / runner	Vehicle capacity	G4 can carry 48 totes and G6 can carry 64 totes. In both EPVs half of the totes are ambient and half are chilled.
	Available time	Availability of runner on a day. An EPV is available if a runner is available (working hours).
	Drop time delta	The additional component that influences the planned drop time, based on runner experience.

2.3.2. Drive and drop time predictions

VROOM requires predicted drive, drop and park times, see Table 2. This section outlines the way these predictions are made. It is key to understand how these processes work currently at Picnic.

The drive time matrix is created for all customers in a dispatch area, and it contains the drive times for all from/to combinations. Also, the hub is included in this matrix as node, so the potential stem times are also in this matrix. The first step in preparing the matrix for VROOM, is to source all drive (and stem) times from Mapbox, an external company that estimates these times. Many from/to combinations are not driven before; hence Picnic cannot accurately estimate these themselves. The requested matrix is sourced in both directions between customers, as distances are often asymmetrical (A-B ≠ B-A).

Picnic applies a correction factor to the sourced matrix: the *drive time factor* (DTF). The DTF typically ranges from 0,8 to 1,2 and is derived from a linear regression based on past actual drive times. This enhances planning accuracy by adjusting estimates from Mapbox. The DTF is the slope parameter of a linear regression of two features, historic actuals (of the eight most recent weeks) and Mapbox' estimates (for those eight weeks). This dataset is first cleaned on reliable datapoints (e.g., outlier or error removal).

The stem time is not only adjusted by the slope, but also by adding the intercept of the linear regression (red elements in Figure 8). These adjustments should account for the fact that EPVs are no regular cars and because Picnic's park time starts a few meters away from the exact customer address.

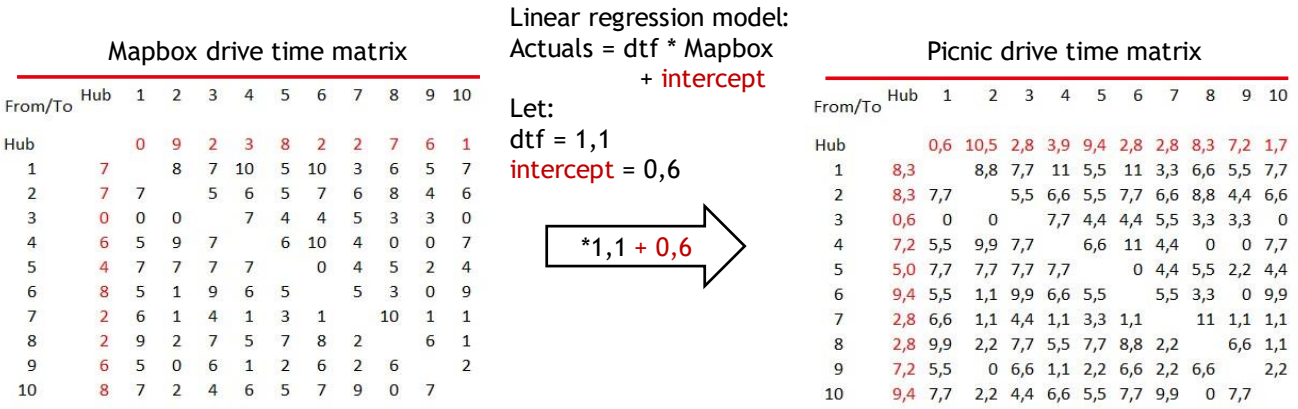

Figure 8. Current transformation of drive time matrix.

Table 3 presents the mean absolute error (MAE) on trip level of some random trips for two scenarios. The first involves Picnic utilizing the Mapbox estimates for drive & stem times, while the second reflects the error metrics resulting from adjustments made by Picnic to the drive time estimates.

Table 3. MAE of random trips using Mapbox drive & stem times or Picnic's adjusted estimates.

	Mapbox' drive times	*Picnic's adjusted drive times*
Total trip drive MAE (s)	254,72	124,69
1-Way stem MAE (s)	25,16	2,95

Similar behaviour as in Table 3, occurs among all trips. This justifies Picnic's decision to adjust Mapbox' estimates for drive and stem times, as Picnic's adjustments yield lower MAEs and thus better approximate the actual drive and stem times. However, drive time estimates still exhibit notable deviations on a trip level. While the MAE of the stem time closely aligns with reality (2,95) in this example. However, only the stem time towards the first customer is included in Table 3 since the data reliability of actual return stem times is very low due to many runners shutting of the tracking device before returning at the hub.

To summarize, the current approach of Picnic regarding drive times involves tailoring the estimates from Mapbox such they more accurately predict the real drive times as Picnic tracks them. However, with this approach all from/to drive times get the same conversion. Which might not capture the last-mile uncertainty (mentioned in Section 1.2.3) well, since in some drive legs there is more (or less) uncertainty.

Furthermore, the stop (park and drop) time at each customer in a dispatch area is predicted by Picnic and given as inputs to VROOM. Currently, Picnic estimates the drop times with an artificial neural network. It considers features such as total tote weight, amount of ambient and chilled totes, but also if the customer lives in a flat or high-rise buildings etc. For repeat customers, historical drop times are also taken as inputs. For new customers, the model uses an average drop time from neighbouring customers as input feature.

This drop time model is called the drop-time service and is owned by the machine learning department of Picnic, AAA. AAA requires several inputs from MPP of the distribution department, namely part of the input features regarding the tote and delivery characteristics. Then the drop-time service extracts the historical drop times from the data warehouse (if applicable) and runs the neural network. The park times are also predicted by a neural network that incorporate the historical park times at that customer. For new customers, the average park time of the neighbourhood is predicted as well. The predicted stop times (sum of drop and park) are returned to MPP, and then VROOM can start creating the trips.

AAA trains the drop neural network once every week. Every dispatch plan occurs once every week, so per week new actual drop times per dispatch plan are tracked and stored in the data warehouse. It trains the model on the last 52 weeks, from which the two most recent weeks are used as test set. If the predictive performance of the newly trained model on the test set is better than the previously trained model (also on the test set), the neural network is updated to predict for the coming week. Otherwise, the previously best trained model is kept for predicting drop times in the future. A similar procedure holds for park.

2.4. Current last-mile delivery performance

Now we study the planning accuracy on the different segments in terms of the main KPIs. The data is from 9th of September 2023 until March 8th 2024, from all hubs in the Netherlands (NL), TW is [-10; +10].

2.4.1. General hub performance

As mentioned in Section 1.2.1, the main KPIs for this research are OT and M/D. For each hub, these KPIs are weekly tracked, and Figure 9 provides a scatterplot with the performance of each hub on these two KPIs. Picnic identifies hubs with three-letter abbreviations.

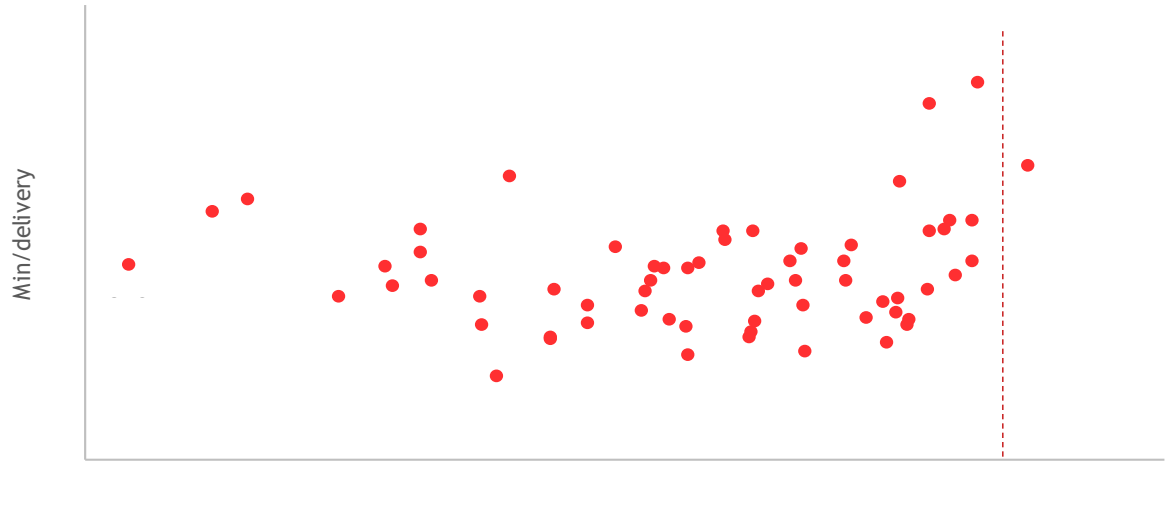

Figure 9. Scatter plot of general hub performance

Figure 9 shows that only one of the 60 hubs exceeds the 96% OT target (red line). And there is a very slight positive correlation between average OT and average M/D among the hubs in NL, this makes sense because if runners get more time per delivery, there is a bigger chance they will not deliver too late. For confidential reasons we removed the values at both axes.

2.4.2. On-time performance on shift level

We also investigate OT on shift level. Figure 11 provides insight in the OT for each of the regular (G4) afternoon shifts per week. It strikes that generally S1 has the best OT and S2 (generally) the worst. But from 2024 on, S2 and S3 show similar performance. S1's superior behaviour could be a result from the fact that it does not have a preceding shift, and therefore is not affected by delays in other shifts.

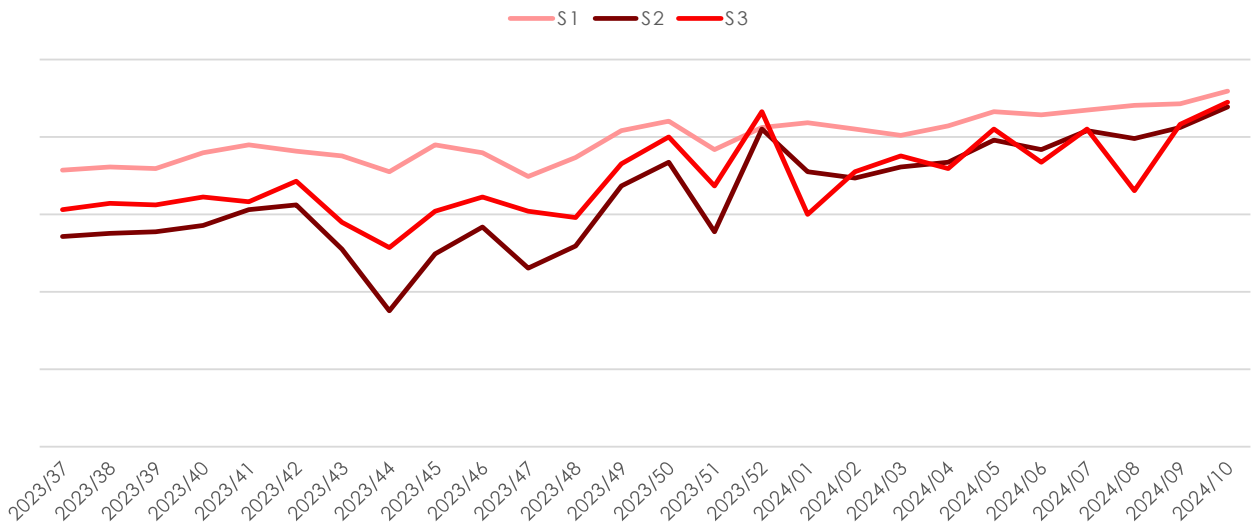

Figure 11. OT per week of S1, S2 and S3.

Then there are shifts for hubs that have G6 EPVs. These G6 EPVs travel longer distances and carry more totes. So, a trip takes more time, as more customers (that also live further away) can be served. For hubs that have the G6 EPVs, there are also two morning shifts, but only two afternoon shifts: G6-M1, G6-M2, G6-S1 and G6-S2 respectively. The time slots for customers work similar to G4 shifts, but with adjusted stem times to suit the G6 trip distances. Figure 10 shows the average OT of the regular morning shifts of both EPV types. Note that G6-M2 is a shift that is operational since end 2023.

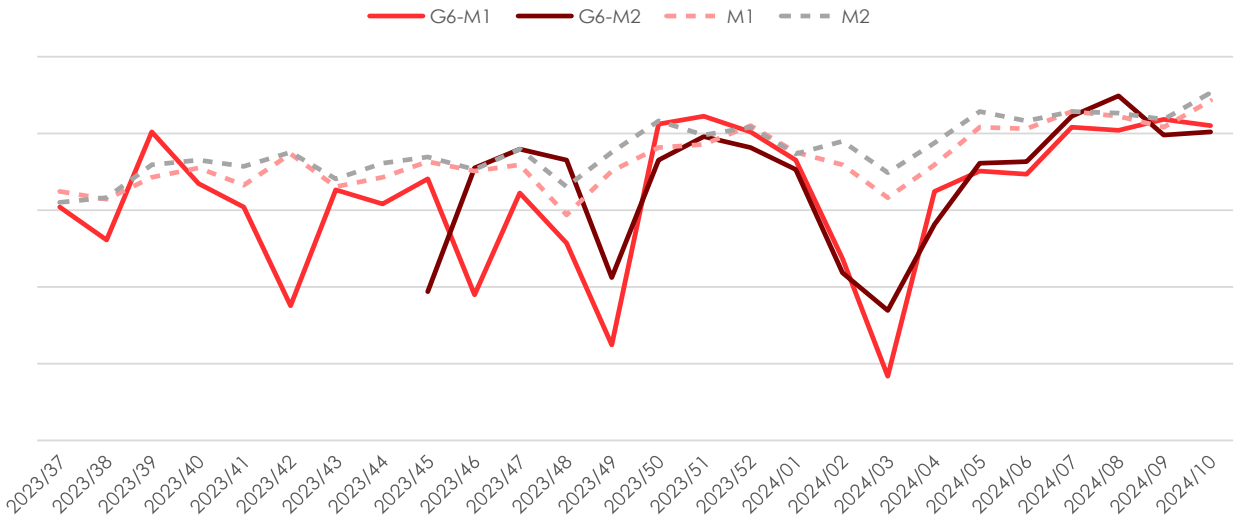

Figure 10. Morning shift OT per week of both EPV types.

In Figure 10, the dashed lines (G4 morning shifts) are more stable than the solid lines (G6 morning shifts) The unstable behaviour of G6 shifts is the result of only few datapoints compared to G4; only 4% of the total deliveries is performed by G6 EPVs. Only six hubs operate G6 EPVs and therefore, OT is impacted significantly, if these hubs have issues with supply from fulfilment centres or encounter heavy traffic congestions. Apart from the stability, the G4 morning shifts generally outperform the G6's. But in the final weeks all morning shifts seem to converge. This converging behaviour over the most recent analysed weeks is also present in Figure 10 and 11, indicating that Picnic is gradually improving their OT.

2.4.3. Minute per delivery performance

Next to OT, we also look at M/D per hub. In Figure 12, the total M/D (red line) is built up from direct (dark dots) and indirect (pink dots) minutes per delivery.

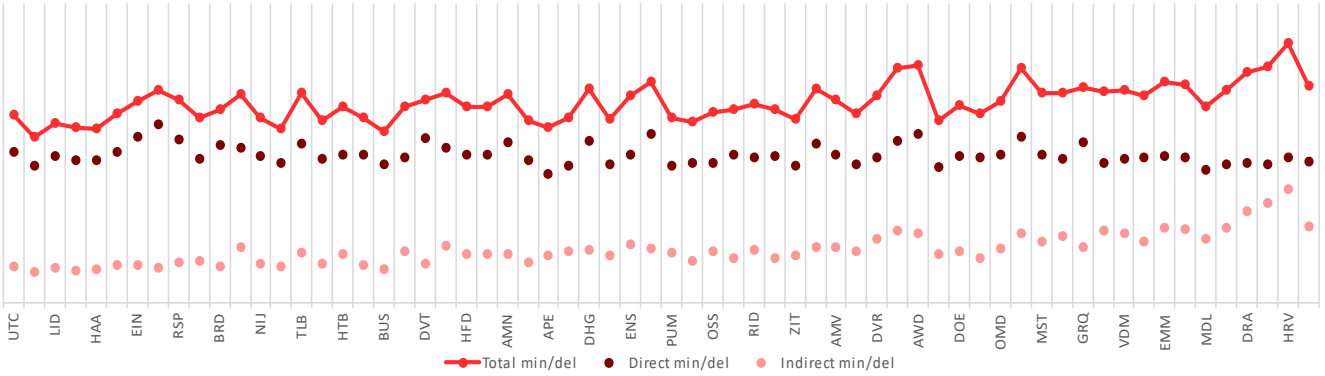

Figure 12. Minutes per delivery per hub in NL.

Direct time represents the time for directly performing the delivery, so all trip segments performed by the runner. The indirect time is the time colleagues require that do not directly perform the trip. For example, hub operators indirectly help perform the trip by maintaining the hub. There are quite some differences among hubs in Figure 12. Hub HRV requires the most time per delivery, however, ALS (lowest) requires 55,8% less. Mainly the indirect time at HRV is relatively large compared to the other hubs. A key driver for indirect time is the delivery volume a hub handles. The more deliveries, the fewer the impact of the indirect hours on M/D is, as the operator hours are quite stable across hubs.

2.4.4. Planning accuracy

Next to the OT and M/D performance, Picnic also aims to achieve high planning accuracy. Each segment (Section 2.1.1) is planned precisely per trip. The split per segment is required as these are inputs for the VROOM. The actual times for each segment for each hub, are tracked and stored in the data warehouse of Picnic. Then for each segment, in each trip the delta (actual seconds – planned seconds) is calculated. This gives insight in how accurate the planning is on each segment, and whether reality is closely approximated or not. Figure 13 shows two histograms with the delta (sec) of the actual trip time and the planned trip time of example hubs Groningen (a) and Meppel (b).

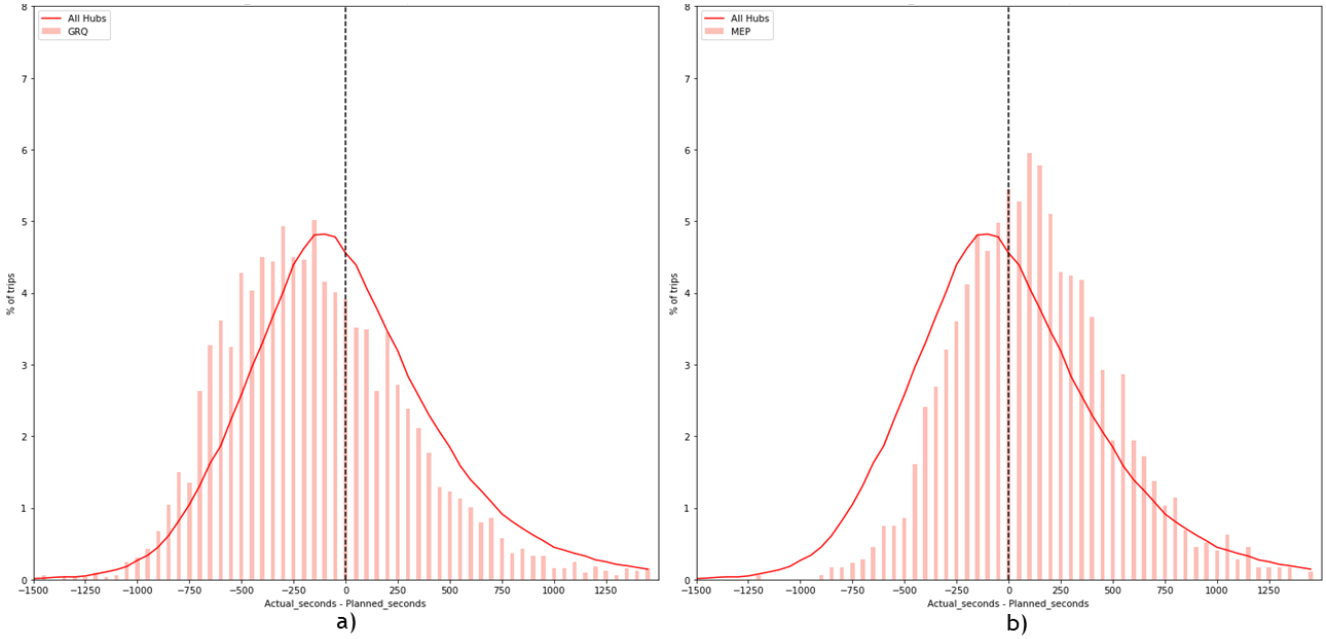

Figure 13. Trip error distributions of GRQ (a) and MEP (b) of actual vs. planned seconds of trips.

The red line in the figure represents the distribution of trip deltas over all hubs. Clear from Figure 13 is that most trips from Groningen (GRQ) are quicker than the scheduled time, whereas an average Meppel (MEP) trip takes longer than planned. The red line, which is exactly the same line in both plots as it represents the total trip error of all hubs in NL, is quite symmetrical around zero, but also somewhat shifted to the left. This implies that most hubs win time during an 'average' trip compared to the planning. This is also what Picnic strives for, to make the planning a little less tight, to build some safety seconds against uncertainty in the trips. Groningen wins more seconds, however, Meppel generally loses seconds compared to planning. This difference can be explained by the difference in hub characteristics and the service areas. The hub in Groningen serves only the city, and the hub in Meppel mainly serves villages around it as Meppel itself is not as large as Groningen.

For GRQ, we take a look at the delta seconds on segment level. So, the drop, drive, park and stem deltas are summed over the deliveries in a trip to get the segment delta per hub per trip. The stem segment is only the departure stem from the hub to the first customer, because there is few data available for the actual return stem time. Figure 14 provides the histograms for each segment of GRQ.

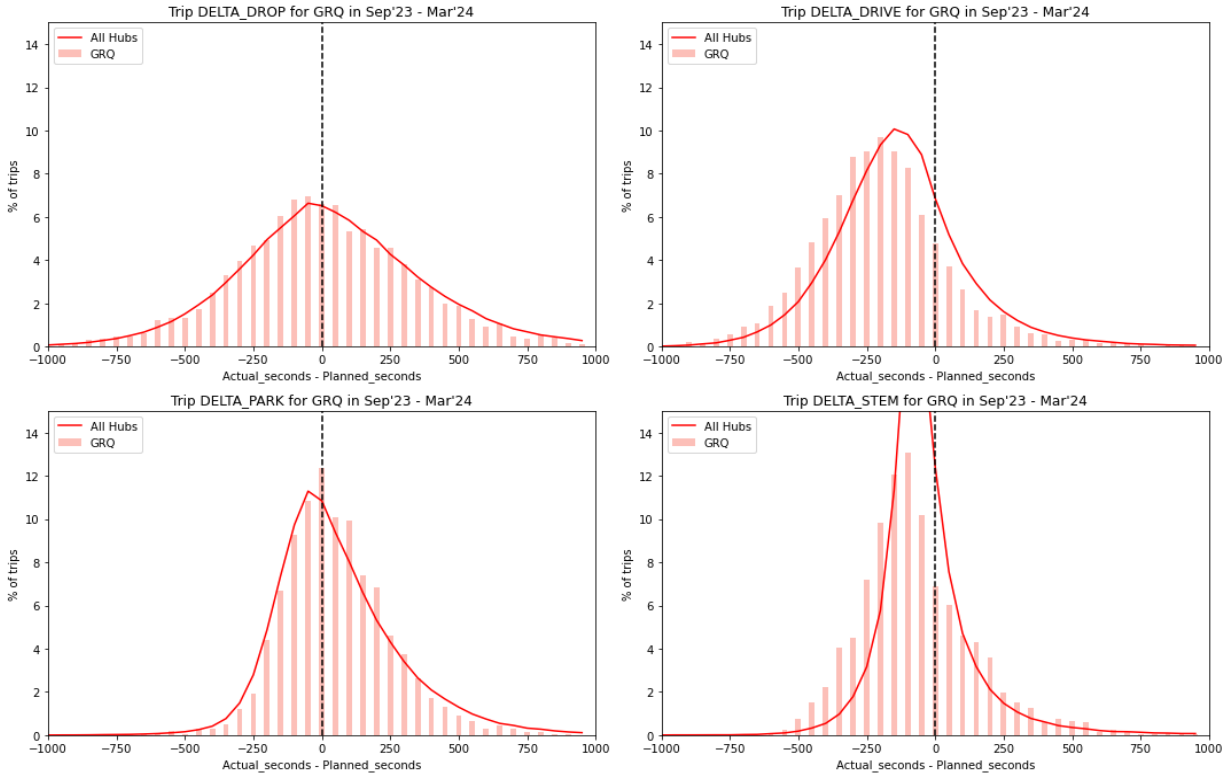

Figure 14. Histograms delta seconds per segment of GRQ of actual vs. planned trips.

It becomes clear that GRQ mainly wins seconds compared to planning on the drive and stem segment. Furthermore, we look at the segment histograms based on the runner experience. Figure 15 plots each segment of the starter level as example. Clear from this figure is that most starters win seconds (compared to the planning) on the drop segment, while losing seconds on the drive and park segments. The combination results in starters performing the entire trip quite similar to all experience levels when it comes to performance compared to planning. However, as explained in Section 2.1.1, the planning is more relaxed for starter runners.

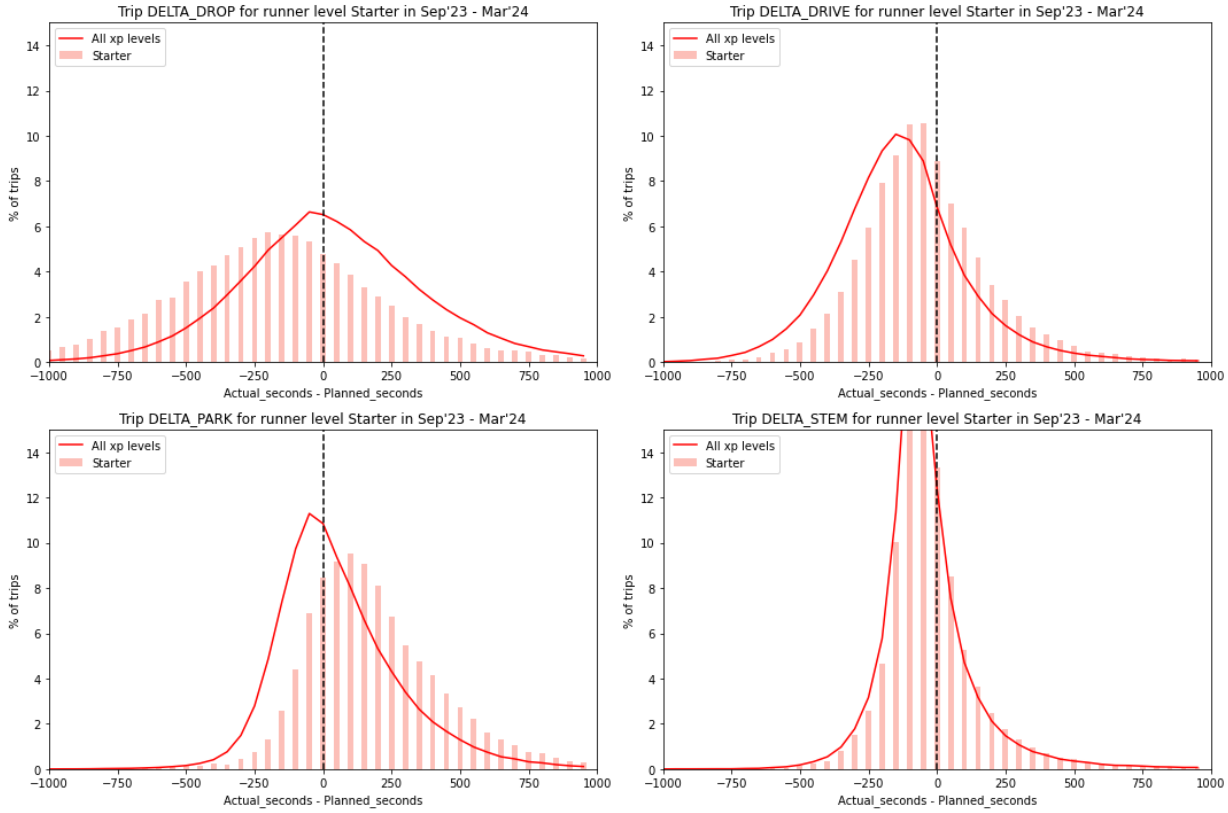

Figure 15. Starter runner histograms of delta seconds per segment

In Figure 14 and 15, we see identical red distribution plots, these depict the overall performance of Picnic in NL per segment. Table 4 provides segment-specific distribution statistics for overall leg performance in all of NL. So, just the red lines of Figure 14 and 15. Again, the trip errors (per segment) represent the sum of the delivery errors per segment, calculated as the difference between actual and planned times.

Table 4. Trip error distribution statistics per segment overall in NL from Sep'23 - Mar'24.

Segment	Q1 (s)	Median (s)	Q3 (s)	IQR (Q3 - Q1)	Average (s)
Drop	-162	42	268	430	61
Drive	-254	-117	19	273	-135
Park	-66	48	206	272	167
Stem	-98	-36	46	144	-4
Overall trip	-283	-13	303	586	89

Table 4 indicates that, on average, trips tend to be quicker than planned for drive and stem segments, as their median and average values fall below zero. Conversely, drop and park segments typically are slower-than-planned, with median and average values exceeding zero. The inter quartile range (IQR) for drop is the widest, while for stem, it is the narrowest. The relatively small stem IQR is due to including only one stem segment per trip, unlike the 12-13 segments summed in other categories, present at every delivery.

It strikes that the median for the drive segment deviates the most from zero. Moreover, 75% of trips have a total drive error of less than +20, since Q3 for drive is +19 seconds. This implies that a large fraction of trips closely adheres to or outperforms planning on this segment. The drive distribution is consequently most skewed to the left of zero (red lines in Figure 14 and 15). So, this segment is planned least accurate.

Overall, the errors per segment compensate each other such that on trip level, the median error is -13 seconds, belonging to the red line plots of Figure 13.

2.4.5. Delivery time error distributions

In Section 2.4.4, we explored the planning accuracy per leg. Now, we delve in how the planning accuracy converts to delivery time accuracy, because that is key in assessing Picnic's operational performance. We look at the error distribution of the actual delivery time (as timestamp) versus the planned delivery time, so not segment specific anymore. The distributions give an indication for main KPIs OT and M/D. The positive tail exceeding 600 seconds, indicates late deliveries while the negative tail below -600 seconds, indicates waiting time before the TW opens.

In Figure 11 of Section 2.4.2, we have seen that S2 generally has the worst OT while S1 generally has the best OT. Hence, we compare these delivery time error distributions in Figure 16. Each delivery is a datapoint representing the difference between the actual and the scheduled delivery time in seconds.

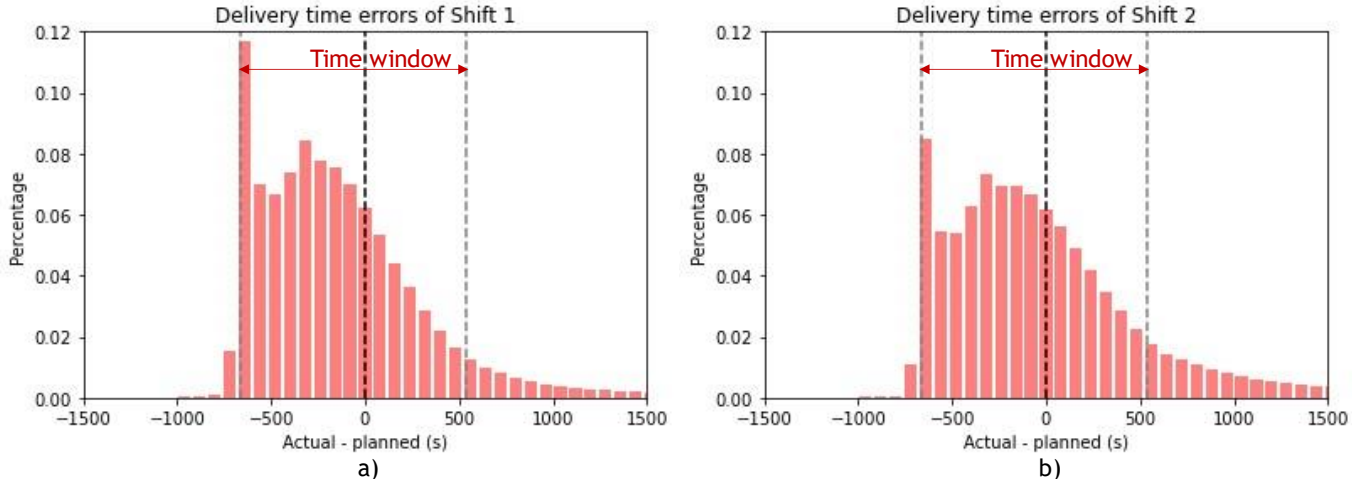

Figure 16. Delivery time errors of S1 (a) and S2 (b).

In Figure 16, there is a notable spike around -600 seconds, indicating many deliveries occur just as the TW opens. Runners arriving earlier wait until that moment, hence the large bar. S1's distribution appears a bit more left-skewed than S2's, aligning with what we have seen in Section 2.4.2.

Moreover, Picnic performs most deliveries on Friday (see Figure 6). Given that weekdays typically have heavier traffic compared to weekends, the error distribution of Friday is also compared to Sunday. Sunday also has relatively many deliveries (Figure 6). These plots are provided in Appendix A. Table 5 confirms the hypothesis that deliveries on Sunday are usually earlier than on Friday, derived from the lower median error, larger fraction of runners delivering in the first-minute, and less late deliveries.

Furthermore, we are interested in the progression of main KPIs during a delivery trip. So, we compare the delivery time error distribution of all four different sequenced customers (2, 6, 10 & 13). These plots are also provided in Appendix A. Table 5 offers all mentioned insights beyond visual comparison.

Table 5. Key statistics from filtered delivery time error distributions, where the late % is multiplied with a random number for confidentiality.

Filter value	*Median (s)*	*IQR (s)*	*% First-minute deliveries*	*% Late deliveries*
Shift 1	-170	556	8,89	5,25
Shift 2	-59	648	6,37	10,24
Friday	-96	590	6,87	7,73
Sunday	-136	581	7,90	6,88
Customer 2	-91	506	5,45	5,70
Customer 6	-108	562	6,20	7,12
Customer 10	-118	651	8,61	8,56
Customer 13	-114	713	9,75	9,55

From Table 5, several insights emerge. Firstly, the median error of -170 seconds in S1, indicates that the median delivery is almost three minutes ahead of schedule. In S2 this is only -59. The larger interquartile range (IQR) and the notably higher percentage of lates in S2 suggest greater variability than in S1.

Secondly, on Sundays, runners tend to deliver further ahead of schedule than on Fridays, indicated by the lower median error. While the IQR remains similar between these days, Sundays see more instances of runners waiting before the TW opens and fewer lates than Fridays. This aligns with the assumption that Friday is subject to heavier traffic. Since drive and stem times account for a large fraction of the time per delivery (Figure 4b), this heavier traffic has its clear impact on overall delivery time errors.

Thirdly, the timing of deliveries shifts as the sequence of customers in a trip progresses. The median error decreases, indicating that deliveries tend to occur earlier. However, variability increases significantly during trips, evident from the expanding IQR and distribution tails on both sides. This suggests that faster runners are consistently outpacing the planned schedule, leading to more waiting time for TWs as the trip advances. On the other hand, slower runners consistently lag behind, leading to more late delivery instances as visible in the late delivery % of Table 5.

2.5. Conclusions

In this chapter we have learned about the current process regarding the last-mile deliveries of Picnic and the terminology used. First, the dispatch areas are determined per hub, for which dispatch plans are created. VROOM creates the dispatch plans and assigns customers to trips based on their location, time slot and tote demand with the goal of minimizing the number of EPVs and total drive time. An important point is that EPVs of Picnic usually do not travel the same drive segments again, because the set of customers in a dispatch plan is almost never identical, implying that the routes between customers are also usually unique. Therefore, drive time estimates are sourced from Mapbox.

We have seen the current performance regarding the main KPIs on different levels: for different shifts and some hubs. Furthermore, we know the inputs of VROOM and that Picnic adjusts the Mapbox drive and stem times with a linear regression of datapoints with just two features: (i) Mapbox' estimate and (ii) actual drive times. Moreover, for predicting drive times, the linear regression model is partially trained on stem data. The predictions of drop times are done by a neural network.

Also, an additional time component in the drop time estimates, based on the runner's experience is anticipated. Next to that, we have a better understanding of the current planning accuracy per hub and runner experience level, on overall trip and on segment level. We have seen that on two segments, runners are usually quicker than planned (drive and stem) and on the other two (park and drop), runners tend to be slower. The drive segment has the largest median error and is most inaccurately planned.

At the end, we have seen how the planning impacts OT and M/D on different levels by plotting delivery time error distributions. In the busier S2, we note more instances of late deliveries and less waiting time before the TWs. Also, on Friday more late deliveries occur than on Sunday, despite both days have similar delivery volumes. At last, as trips progress, the spread of error distributions widens. Since the first few customers experience fewer late deliveries and less runners wait for the TW here, compared to the last few customers.

3. Literature review

This research is closely related to concepts that are common in literature. The problem in this thesis touches (i) the time slot management problem common in general AHD settings, (ii) the vehicle routing problem and (iii) stochastic travel & service times, that can be predicted with machine learning. The subsequent sections present existing literature on these topics. We cover RQ2 and its sub-questions.

3.1. Attended home deliveries

Mackert (2019) explains an attended home delivery (AHD) service model, as a business model in which goods are delivered to the front door of the customer, within a time slot agreed on by the e-grocer and the customer in advance. A driver for customers to shop their groceries online is the high customer service level through the provision of narrow time slots (Mackert, 2019). Agatz et al. (2008) support this statement and emphasize the need for the customer being home, as food can be perishable. As goods must be transported to customers within time slots, AHD settings are usually solved by a vehicle routing problem with time windows (Agatz et al., 2008; Mackert, 2019). Literature makes the distinction between same-day and next-day home deliveries. We focus on next-day home deliveries in which vehicles are being dispatched after all orders have been collected (Strauss et al., 2021), so there is no uncertainty in demand upon dispatching. This focus matches the business model of our organizational context. However, demand uncertainty can play a role in offering time slots in AHD. Section 3.1.1 presents key decisions regarding time slots, while Section 3.1.2 deals with policies regarding time slot offering.

3.1.1. Time slot decisions

Usually, companies offer a number of regular delivery timeslots, from which customers pick one that fits their personal schedule best. Some key decisions related to the time slots are:

- *Which and how many time slots should be offered to what regions?* This decision sets the conditions for delivery routing, that is based on actual customer orders and their locations (Agatz et al., 2008). It might be beneficial to offer only a limited number of slots in certain zip codes, to increase demand per time slot. All for the sake of cost-efficient delivery routes. Controlling the set of offered time slots is called slotting (Lang et al., 2021).
- *How wide should the time slot be?* Narrow delivery time windows lead to little flexibility in vehicle routing and therefore high fulfilment costs. Wide timeslots on the other hand are less desired by customers, even though some customers (pensioners, people working from home) do not require them to be narrow (Agatz et al., 2008; Strauss et al., 2021).
- *Do time slots overlap?* Agatz et al. (2008) state a marketing advantage could be achieved if time slots may overlap, as it offers more options and flexibility to customers.

A company's preferences must be balanced with the customers preferences. Customers want a well-balanced offering of time slots over a day, and over a week. But companies want a well-balanced demand over a day and week, especially if the demand is geographically smooth. Time slot design impacts expected demand in certain regions, but demand also drives time slot design (Agatz et al., 2008). A challenge that comes with this is to assess demand behaviour when certain time slots are not offered.

In literature, a time slot is usually offered to a new customer if the planning model can find a feasible solution with the new customer in that time slot. As this happens in real time, and the problem instance can be very large, this feasibility check can be slow in practice (van der Hagen et al., 2024).

3.1.2. Time slot offering policies

Literature distinguishes two time slot policy types: time slot allocation and time slot pricing. Time slot allocation policies seek what time slots to offer to what delivery area, whereas time slot pricing policies try to steer customer behaviour towards time slots that are cheaper for the company (Akkerman et al., 2022). Furthermore, a distinction between static and dynamic policies is made. Static methods use forecast data or static rules and can be used to make strategic and tactical decisions (such as capacity planning), whereas dynamic policies happen during the decision making, once real-time information is known (Akkerman et al., 2022). Dynamic decisions could be, for example, whether to close a time slot or open an extra time slot, given real-time demand. The goal of pricing and allocation is to minimize transportation costs and balance resources.

3.2. The vehicle routing problem

This research is about finding accurate predictions as inputs for the vehicle routing problem (VRP) solver of Picnic. Hence, we dive into VRP literature to learn about this context. The basic VRP was first introduced by Dantzig & Ramser (1959). The aim was to optimize routing between service stations and a depot for gasoline delivery with one vehicle. Clarke & Wright (1964) transformed the problem by including more vehicles, and they formulated the problem with a linear model. Lenstra & Kan (1981) have proven the VRP to be NP-hard, making exact algorithms impractical for large instances. Since then, numerous variants and solution methods have emerged. The next sections treat the VRP variants and solution methods (3.2.1), and the, for this research relevant, stochastic VRP with time windows (3.2.2).

3.2.1. VRP variants & solution methods

An early variation on the basic VRP was from Clarke & Wright (1964) as they introduced more practical restrictions to the delivery at customers. More recent literature (Braekers et al., 2016; Eksioglu et al., 2009; Elatar et al., 2023; Tan & Yeh, 2021) define variants based on criteria such as: capacity of vehicles (CVRP), heterogeneous fleet VRP, the need to return to depot (Open VRP), priority of customers, presence of (soft) customer time windows (VRPTW), time dependency (dynamic VRP), multi-depot (MDVRP) and randomness (Stochastic VRP). Braekers et al. (2016) emphasize the wide variety of objective functions, among the different VRP variants, although most variants have a routing-cost component in the objective function. A trending variant of the VRP is the green vehicle routing problem, where the focus is on minimizing emissions or it considers battery charging (Matijević, 2023). There exist multi-objective VRP formulations too, that usually result in a set of non-dominated solutions called the Pareto set (Giuffrida et al., 2022). Kara et al. (2008) proposed the first cumulative VRP (Cum-VRP). This Cum-VRP incorporated flow on the arcs as indicator for travel costs next to distance travelled. Vehicle load or fuel usage are used as variables to determine the flow. An extension to this Cum-VRP is the cumulative capacitated VRP (CCVRP). In which the objective is to minimise the maximum or the average arrival time at customers. Campbell et al. (2008) solve the CCVRP with well-known insertion and local search techniques and state that CCVRP has many real-world applications, for example in food delivery. Corona-Gutiérrez et al. (2022) performed an extensive analysis on the CCVRP literature.

The different VRP variants try to capture different real-life aspects of logistics. Consequently, most authors propose highly problem specific solution methods, which are not applicable to other VRP variants (Braekers et al., 2016). VRPs are primarily solved by heuristics due to their NP-hard nature.

Literature distinguishes two main heuristic classes for solving VRPs. *Classical heuristics*, developed between 1960 and 1990 and *metaheuristics* (Laporte et al., 2000). Many classical heuristics are common construction heuristics, which find initial feasible solutions. Some well-knowns are the savings algorithm (Clarke & Wright, 1964), the sweep algorithm (Gillett & Miller, 1974) or the cluster-first, route-second algorithm (Fisher & Jaikumar, 1981). These are often followed by improvement heuristics like reinsertion

or λ-opt to improve solutions (S. Lin, 1965). More complex metaheuristics like tabu search, simulated annealing, ant colony systems, GRASP and variable neighbourhood search are frequently applied to VRP variants (Boussaïd et al., 2013), with tabu search and variable neighbourhood search being the most commonly used methods (Elshaer & Awad, 2020). However, since the goal of our research is to enhance accuracy of the (stochastic) inputs of VRPs, we do not go into more detail on solving VRPs.

3.2.2. Stochastic vehicle routing problems with time windows – SVRPTW

Stochastic versions of the VRPs also receive attention in literature. The stochasticity can be present in different elements of the VRP. Uncertainty is usually in the travel or service times, the presence of customers and/or demand. In literature, many SVRPTW models are presented (Dror, 1993; Dror et al., 1993; Taş et al., 2014) and in this thesis' context, the uncertainty is in the travel and service times which are inputs of the VRP. Hence, we dive in a few existing SVRPTWs that also have this characteristic.

A commonly used heuristic to solve stochastic VRPs is to replace the random elements with their means and solve the corresponding deterministic model (Kenyon & Morton, 2003). Russell & Urban (2008) model uncertain travel times with an Erlang distribution to solve the VRPTW, which is a special case of the gamma distribution. The gamma distribution is used to model travel times by Taş et al. (2014) who solve the problem by column generation. Zhang et al. (2012) model the travel times by a normal distribution, from which different sets of travel vectors are drawn and used in the chance-constrained optimization model. The normal distribution is also used by Li et al. (2010) to model both uncertainty in travel and service times. Li et al. (2010) employed chance constrained and stochastic programming models to minimize the transportation costs. A tabu search based heuristic was applied, however both their approaches proved computationally intensive.

Ehmke et al. (2015) also research a SVRPTW, in which a certain service level must be guaranteed. Different than Li et al. (2010), only stochastic travel times are considered. They define a set of routes acceptable if the probability of arriving at each customer within their time window is above the service level. They used a parallel insertion heuristic, followed by a tabu search algorithm, randomly selecting either of four neighbourhood operators. Similarly, Kenyon & Morton (2003) formulate a SVRPTW where the objective is to maximize the probability of completing the vehicle tours within a given deadline, and they incorporated both uncertain travel and service times.

Braaten et al. (2017) investigate the SVRPTW with uncertain travel times and hard time windows, modelling uncertainty via delay patterns per vehicle, which in turn determines the travel time. They propose a robust heuristic capable of handling any realization of the uncertain travel time patterns. They build on the same problem presented first in Agra et al. (2013).

Han et al. (2014) also study the SVRPTW with uncertain travel times, modelling uncertainty with scenarios. In each scenario, the arcs have a random travel time from a range. They seek to find a robust solution against the worst-case travel times in each scenario, using a branch-and-cut algorithm. Eventually they take the optimal route via minimizing the expectation over all scenario solutions.

Another commonly researched variant of the SVRPTW incorporates stochasticity by means of fuzzy variables. Fuzzy variables incorporate uncertainty in VRPs by taking some value from a fuzzy set. Take for example, the fuzzy set 'short' or 'medium' in case of travel times. Then, the membership function indicates the degree of membership that a travel time of, for example, 10 minutes belongs to the fuzzy set 'short'. Fuzzy sets are introduced by Zadeh (1965) and hereafter studied by multiple researchers in the context of SVRPTWs where the travel times were fuzzy (Tang et al., 2009; Zheng & Liu, 2006).

Considerable research has been devoted to the SVRPTW from various perspectives. Typical strategies for addressing uncertainty include substituting the uncertainty with its mean and solving the deterministic

model, constructing chance-constrained models, employing stochastic programming with recourse, conducting robust optimization focusing on worst-case scenarios, and utilizing fuzzy sets. However, in this thesis we research the impact on business KPIs by enhancing the prediction accuracy of the VRP inputs (travel or service times), hence we dive in the literature regarding this aspect.

3.3. Predicting travel and service times

In AHD, travel and service times are key inputs to VRP-solvers to get accurate results. Apart from the methods in Section 3.2.2, stochastic travel and service times are also often predicted. Taghipour et al. (2020) mention three groups of travel time prediction models: (i) naïve methods, which are simple but often inaccurate. (ii) Traffic-theory-based models which lay relations between traffic variables and travel time and (iii) data-driven models, which require a large amount of data, and use statistical methods to predict. Here it holds that the more data available, the more accurate results will be. This third group can be divided in parametric and non-parametric methods, where parametric methods have a pre-defined function / structure in which parameters need to be estimated (Taghipour et al., 2020).

Gmira et al. (2020) describe parametric methods, such as linear regression, the Kalman filter and ARIMA to predict short-term traffic characteristics like speed. Non-parametric methods commonly used are support vector machine, neural networks, random forests, and K-nearest neighbours (Taghipour et al., 2020). Gmira et al. (2020) eventually propose a neural network to predict travel times as it performed best of all methods in their experiments. Qiu & Fan (2021) compared the performance of four different ML techniques to predict travel times in a case study, with the goal of low-variance predictions. They found the random forest to give the most accurate, and stable results with the lowest mean absolute percentage error as key metric. It outperformed the decision tree, the extreme gradient boost and a long short-term neural network. With a five-fold cross-validation they tune the maximum number of features, the number of trees and the minimum leaf size for the random forest, as they are the primary features that can be tuned to optimize predictive power (Qiu & Fan, 2021). Taghipour et al. (2020) predicted travel times with a neural network, k-nearest neighbours and a random forest. Here random forest outperformed the other models, which was especially accurate with drive times below 15 minutes.

Wolter & Hanne (2024) predicted service times of home deliveries of furniture. They used features like weight of order, location of customer, service technician experience and more. Their artificial neural network outperformed linear regression and support vector machine models. But deliveries with relatively long durations were harder to predict as there was not enough data available on those cases. Wolter & Hanne (2024) explicitly mention that data quality is a key issue for practical implementation, what tolerances are used and how to categorize are essential aspects to consider.

L. Lin et al. (2018) state that for traffic predictions, non-parametric methods are often more flexible than other models, when dealing with complex datasets containing nonlinearities. Therefore, even though they have limited transparency and might be computationally expensive, we decide also to research non-parametric methods for our travel time predictions.

3.4. Key findings literature study

In this chapter, we have looked at what literature tells us about the general concepts that relate to this research. We have looked in more detail at stochasticity in VRPs and how to deal with that. From this latter part, we summarize the key findings from our literature study:

- In AHD and VRP settings, many researchers mention the need to meet the time windows, but not necessarily waiting time before the time windows is considered.
- There are many ways to deal with stochasticity in the SVRPTW, by modelling or by predicting.
- An often-studied topic is stochasticity in demand and travel times.
- Predicting travel times is often done by a neural network, a random forest and sometimes with an extreme gradient boost, decision tree or linear regression.
- Hyperparameter tuning in similar settings is done with a five-fold cross-validation.

In chapter 2, we learned about Picnic's complex context, where many VRPs run on a daily basis and many customers are served with rarely recurring routes and hence Mapbox' estimate is required. This makes it impractical for Picnic to create fuzzy sets, scenarios or delay patterns. Also, these methods are considered too general and not individual drive segment specific enough for the highly specific planning Picnic desires. Therefore, in this research we want to enhance the VRP travel time input accuracy by prediction methods as data availability at Picnic is high and predicting is common in literature to enhance VRP input accuracy. All with the goal to improve business KPIs. The contributions of this thesis to existing literature are therefore two-fold:

1) Studying different prediction methods to predict travel times, by conducting a case study that incorporates real-life characteristics of an e-grocer in an AHD setting, where historical routes rarely recur. We compare the methods to determine the most suitable.
2) Translating predictions of travel times into business KPIs by reconstructing last-mile deliveries using machine learning-predicted drive times. This involves balancing efficiency (with minimizing *minutes per delivery*) and service level (by increasing *on-time %*) in a SVRPTW.

4. Solution design

This chapter provides the solution design. We will address RQ3. We briefly recap the problem and then dive into the proposed solution design, for improving the planning accuracy of the drive segment.

As introduced in Section 1.2, the last-mile delivery planning of Picnic is not as accurate as desired. In Chapter 2, we have seen that Picnic plans four segments in each delivery trip: stem, park, drop and drive. However, each segment shows errors of actuals compared to planned times. By increasing planning accuracy, improvements in both *on-time %* (OT) and *minutes per delivery* (M/D) are expected. In Section 2.4.4, we have learned that the drive segment has the biggest error. Next to that, the current drive planning model is rather simple and not many improvement rounds have been executed. Summarized, we find considerable improvement potential in this segment and prioritize our focus accordingly.

To achieve the goal of increasing OT or reducing M/D, we now dive in the proposed model construction approach to increase drive planning accuracy. Consequently, we assess how the best model impacts OT and M/D.

4.1. Model construction approach

The aim is to enhance planning accuracy of Picnic's drive segment by refining the current linear regression model (see Section 2.3.2), to reduce the gap between actual and planned drive times.

Section 3.3 outlines common prediction methods for travel times in literature. Non-parametric methods that have led to good results are the random forest and the neural network. Furthermore, we learned from literature that decision trees are also used to predict travel times. This model is also relatively simple, interpretable and efficient (Kuhn & Johnson, 2013). The last non-parametric model that is used in literature to predict travel times is the extreme gradient boost (XGB). This model should give high predictive performance and is computational efficient. Additionally, the parametric linear regression is common in literature, which is adopted by Picnic in some form. This model also has high interpretability and is efficient, which can have its practical advantages for Picnic. In this research, we will construct these mentioned models as they were either common or performing well in similar research.

Exploring a linear regression for drive time prediction, might seem redundant given Picnic's current use, but it is valuable to assess its performance with more features than the current has (only two). Additionally, the current linear regression is trained on the eight most recent weeks (e.g., the eight past dispatch plans for ALS on Monday in S2), to make predictions for the next one (so, the coming ALS-Monday-S2). In this study, we train across dispatch plans, because it is expected that machine learning (ML) methods could capture patterns that hold across dispatch plans, also this ensures that we only have to train the models once (per time unit) instead of training each model for the different dispatch plans, which is expected to be rather time-consuming. Picnic's current linear regression, which is trained per dispatch plan following the existing procedure, serves as a benchmark for comparison.

To develop accurate prediction models, we need to follow some steps. First, the dataset is built and then a certain training, hyperparameter tuning and testing procedure must be followed for the machine learning models. All with the goal of increasing predictive ability of each individual model. Figure 17 provides the general procedure for constructing each of the five methods in this research.

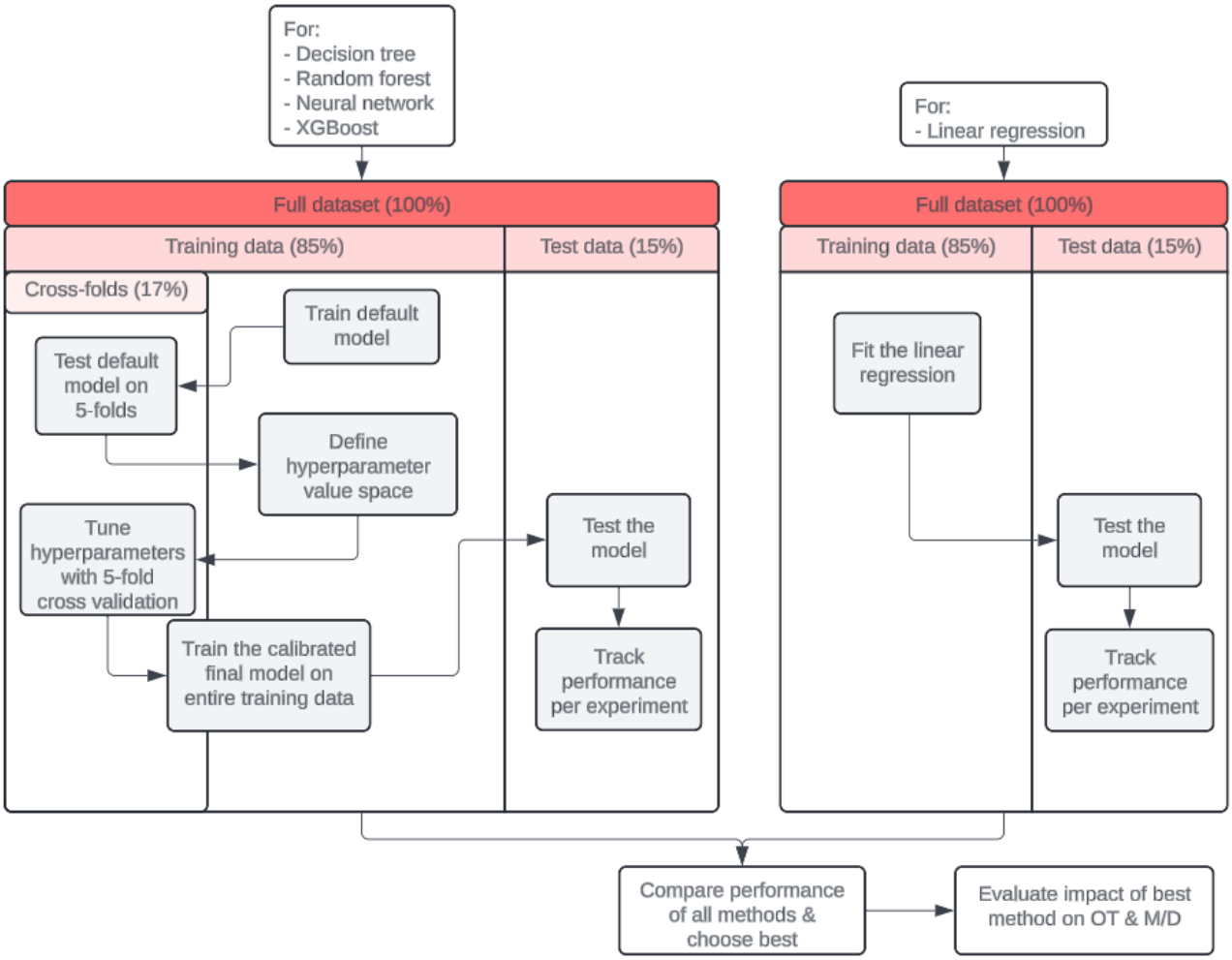

Figure 17. Model construction approach for each of the five methods.

After building and preparing the dataset (Section 4.1.1), we had 14 predictor variables left. With this relatively small number, we consider a feature selection step irrelevant. Additionally, the decision tree, random forest and XGB already perform a feature selection inherently, by splitting data in nodes based on the most important features. The higher the impact a feature has (considered by the model), the more often it gets picked to split on. The earlier mentioned linear regression is a non-tuneable prediction method, and therefore does not require the same steps as the other four ML methods. In the remainder of this section, we will explain the steps of Figure 17, starting with the dataset description in Section 4.1.1. Section 4.1.2 deals with the models and 4.1.3 with the hyperparameter tuning. In Section 4.1.4 we discuss the experimental set-up accuracy performance, while Section 4.1.5 outlines the main KPI evaluation of the best method.

4.1.1. Dataset description

The dataset for this part of the research contains all the deliveries of nine hubs in NL, as building the models and performing experiments for 60 hubs would be too time-consuming. The hubs are selected based on their current performance on the drive segment: three hubs that perform the drives quicker than average hubs, three average hubs and three slower than average hubs. This way, we can see for different hub types, how each model performs. With the aim to draw an aggregated conclusion for the whole of NL. Appendix B provides the current drive performance of the selected hubs.

The dataset covers the first three months of 2024, totalling thirteen weeks. The reason we do not include a longer period, e.g., add December of 2023, is that in the last week of 2023 an update of the drive time planning was done at Picnic, so older data would not be representative anymore. Also, with this

timespan, we include holiday weeks and do not increase the number of weeks too much, compared to Picnic's current approach where it looks back the eight most recent weeks. However, we increase the number of weeks a bit because we expect the ML models to perform better with more data included. Furthermore, we decided to only include the regular G4 shifts because they account for 90,5% of Picnic's deliveries. These shifts are: M1, M2, S1, S2 & S3. After the data preprocessing steps (Section 4.1.1.1), the dataset contains just over 291,200 deliveries, and hence records.

In Appendix C, a full description of the input features of the dataset can be found. The features relate to neighbourhood characteristics of the customer, like address and car density, postal zip, urbanity degree and average distance to a main road. Also, we included features that tell something about the timing of deliveries, like weekday, daypart, holiday week (or not) and shift. And of course, the prediction of Mapbox which is key because, as stressed in Chapter 2, from/to combinations of customers are almost never the same. For that reason, we only include features that are known before VROOM. For example, the rank of the runner is uncertain at that stage, as it is not yet certain who will perform which trip.

Approximately 85% of the data, representing the first eleven weeks of 2024, is allocated for training and cross-validation. The remaining 15%—the last two weeks of March 2024—is held as unseen test data, never used in training or validation, ensuring fair evaluation of model performance. In the 85% training data, we can create five folds of 17% each for cross-validation in hyperparameter tuning. The unseen test data is split in experiments representing single dispatch plans, aligning with the required VROOM inputs. The 85% training data is not split in dispatch plans, because we train across dispatch plans.

4.1.1.1. Data preprocessing

The dataset with the nine hubs in the chosen period underwent data cleaning and some preprocessing steps following the guidelines of Kuhn & Johnson (2013). First of all, the records with empty values in either of the input features were removed, this was approximately 4% of the dataset. Furthermore, we removed the records where the drive time was stored as 'unreliable', to make sure no mistake is made in the recording of the datapoints which could influence model performance.

We also excluded outliers in the dataset. Picnic considers a datapoint an outlier if the following condition is not met:

$$\frac{1}{4} \leq \frac{Actual\ driving\ seconds}{Mapbox\ estimated\ seconds} \leq 4$$

If Mapbox' estimate is more than four times larger, or smaller than the actual, Picnic assumes that something unusual happened during the drive time, or that Mapbox made unreliable (or extremely inaccurate) predictions. Picnic does not want to predict for these outliers, because they will impact the other 'regular' predictions too much. With this outlier removal condition, we excluded 1.67% of the datapoints. This condition is also used in Picnic's current drive time approach (Section 2.3.2).

Additionally, we created the correlation matrix of the predictor variables, to identify redundant variables, that only add computational complexity, without being relevant to the result. This matrix can be found in Appendix D. We found that the population density is highly correlated to the car and address density. So, we removed the population density predictor. Other correlated features, such as longitude and latitude, are considered relevant enough to retain. At this point, there are 14 predictor variables left.

At last, we scaled the features of datatype float. This results in a loss of interpretability, but the linear regression and neural network will benefit from the predictors being on the same scale, by improving numerical model stability (Kuhn & Johnson, 2013). The other ML models will be unaffected.

4.1.2. Default models

Before we train and cross-validate the performance of each of the chosen ML methods with default hyperparameters, we briefly outline each method we selected (and supported at the start of Section 4.1):

1) A decision tree (DT) is a tree-based method that recursively splits data based on feature conditions at nodes and results in leaves based on the value of the feature. The splits are chosen in such a way that information gain is maximized. Eventually predictions are made after passing through the nodes.
2) A random forest (RF) builds multiple decision trees and, for regression tasks, makes a prediction based on the aggregation of the predictions of each tree. It anticipates the 'wisdom-of-the-crowd' principle as each decision tree is built differently and therefore makes a different prediction. In a RF, the individual DTs are independently trained on a random subset of the training data with bootstrap sampling, which reduces the risk of overfitting.
3) A neural network (NN) makes a prediction by passing data through connected nodes, organized in layers. While training, the network learns complex patterns by iteratively adjusting the weights at each node. A neural network includes activation functions to deal with non-linearity in the model, which allows for learning complex patterns.
4) Finally, the extreme gradient boost (XGB) constructs multiple weak learners (decision trees) sequentially by the boosting principle, where each newly built tree compensates for the errors in the previous tree, by optimizing on a certain loss function. The final prediction is obtained by (weighted) aggregating the predictions of all trees.

To get performance benchmarks for each ML method, we initially train default models and evaluate them using a 5-fold cross-validation (cv) procedure. Also, bagging training is considered. However, that is recommended with limited data availability, which is not the case here. Then we can compare the default models with the eventually tuned models. We perform this in Python 3.11 and use the Scikit-Learn library. In Section 4.2, we compare the 5-fold cv performance of the default models, with the tuned models to learn the impact of tuning. The linear regression does not require hyperparameter tuning, therefore only the 'default' version exists. Now, we proceed with the hyperparameter tuning of the four ML methods.

4.1.3. Hyperparameter tuning

Hyperparameter tuning is crucial for enhancing predictive performance of ML models. Hyperparameters namely determine the complexity of the models. Thereby influencing both speed and the chance of under- or overfitting (Akkerman & Mes, 2022). We determine the space the hyperparameters can take and tune them with a Bayesian optimization approach, aiming for the hyperparameter combination that maximizes the R-squared value (R^2) through a 5-fold cv, as also seen in Qiu & Fan (2021) in Section 3.3. The R^2 measures the variability in the data set that can be explained by the model (Akkerman & Mes, 2022). It is a value between 0 and 1, with 1 indicating that all variability is captured by the model.

Bayesian hyperparameter optimization is an efficient algorithm that balances exploration and exploitation with knowledge of previous iterations to guide the sampling for the hyperparameters (Brochu et al., 2010). Bayesian hyperparameter optimization is considered faster than the conventional grid search, and more guided towards promising values than the quick random search, as it maximizes a probability function that approximates the expected predictive performance (Mantovani et al., 2018). The 5-fold cv prevents the hyperparameters to overfit a subset of training data. For this purpose, we use the Scikit-optimize library.

Table 6 provides the hyperparameters that we will tune. The models have more hyperparameters, but we do not tune those and keep their default values from the Scikit library in Python 3.11. The chosen hyperparameters are well-known to-be-tuned parameters per corresponding model.

Table 6. ML methods with 'to be tuned' hyperparameters.

Model	*Hyperparameter*	*Description*	*Value range*
Decision tree	Maximum features	This limits the number of features considered at each split in the tree	[1, 14]
	Maximum depth	This value limits the number of splits the tree can take	[1, 14]
	Minimum samples leaf	The minimum number of samples that must be present in each end node (leaf)	[5, 20]
Random forest	Number of estimators	The number of trees in the RF	[100, 300]
	Maximum features	Same as for a decision tree	Best value of DT
	Maximum depth	Same as for a decision tree	Best value of DT
	Minimum samples leaf	Same as for a decision tree	Best value of DT
Neural Network	Initial learning rate	The rate at which the model learns and converges to the final model	[0,001; 0,1]
	Batch size	Number of samples after which the NN updates the weights	[10, 100]
XGBoost	Number of estimators	Number of trees the XGB builds	[100, 300]
	Maximum depth	Same as for a decision tree	Best value of DT
	Initial learning rate	Sets the convergence rate and each tree's contribution to the final model	[0,01; 0,3]

For a DT, we limit the maximum features to fourteen, matching the number of predictor features in the dataset. Maximum depth has the same range, since more possible splits than features results in a high risk of overfitting. The minimum samples leaf is typically between 5 and 20. A too low number could result in a too complex tree structure that overfits the training data, and hence, has bad generalizability. But a too high value might not offer enough learning flexibility (Mantovani et al., 2018). For the RF, we choose a value range for the number of trees between 100 and 300. This is symmetrical around the 200 of Akkerman & Mes (2022) who balance performance and computational effort with this number. The other hyperparameters are fixed on the best DT values. These hyperparameters were also tuned in similar research by Qiu & Fan (2021) as discussed in Section 3.3

The learning rate of the NN and XGB are typically small positive values. We choose a lower range for the NN, because the NN has a more complex structure than an XGB. Hence, a lower range is chosen to better capture the complexities than in an XGB. For both models, we keep the initial learning rate constant throughout the process to reduce overall computational complexity. Furthermore, for the NN we have the batch size range similar as Gmira et al. (2020), who also predicted travel times in AHD (Section 3.3). And we set the range for the number of trees in the XGB equal to the number of trees of the RF.

After the best hyperparameters are found to fit the training cross-validation data, we will fix these values and train each model on the entire 85% of training data. Then, we move on to the experiment phase.

4.1.4. Prediction model experiments & performance evaluation

The last step before selecting the best prediction model in Figure 17, involves conducting experiments on the 15% unseen test data and tracking predictive performance of all models. For that sake, the test data is partitioned into experiments. Since Picnic requires the drive time matrix for each dispatch plan, we also split the 15% test data into dispatch plans. Each experiment is, hence, a unique combination of hub, weekday and shift.

For each experiment, we make predictions using the trained models. The goal of these experiments is to get the performance of each model on the test data and compare it to the current prediction performance of Picnic on week 12 and 13 of 2024 (see Section 5.1). Consequently, the best prediction model will be selected for further evaluation regarding the impact on the main Picnic KPIs OT and M/D.

As performance metrics per experiment, we use the mean absolute error (MAE) and the mean absolute percentage error (MAPE). Next to that, we also calculate the R^2 value.

$$MAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n} \qquad \textit{Equation 2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i} \times 100\% \qquad \textit{Equation 3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2} \qquad \textit{Equation 4}$$

The MAE, while easy to compute and interpret, is scale-dependent. Hence, we also include the MAPE, which is scale-independent and therefore preferable for comparing datasets of varying scales (Chen et al., 2017). With the MAE and MAPE, outliers in the unseen test data are not given extra weight, unlike the (root) mean squared error does (Chen et al., 2017). Furthermore, the R^2 value indicates how well the trained models are generalizable to unseen data. Alongside these KPIs, we track the model training and prediction time to assess computational efficiency. Subsequently, we aim to aggregate our experimental results to decide upon the best prediction model.

Table 7 provides the approach for estimating the predictive performance of each prediction model.

Table 7. Experimental set-up for finding the best performing prediction model.

	Experimental set-up and KPI storage for (calibrated) prediction models
1	***Initialize sets***
2	*Models* = [DT, RF, NN, XGB, LinReg]
3	*Hubs* = [UTC, LID, ALE, APN, BUS, EMM, HFD, LEY, HRV]
4	*Days* = [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday]
5	*Shifts* = [M1, M2, S1, S2, S3]
6	*Training_data* = full dataset from week 1 to 11
7	*Test_data* = full dataset from week 12 to 13
8	
9	**For** *model* in *Models*:
10	Fit *model* on *Training_data*
11	Store the trained *model* as *trained_model*
12	**End**
13	
14	***Initialize*** *final_results*
15	**For** *hub* in *Hubs:*
16	***Initialize*** *hub_results*
17	**For** *day* in *Days:*
18	**For** *shift* in *Shifts:*
19	*Experiment* = (*hub*, *day*, *shift)* in *Test_data*
20	**For each** *trained_model:*
21	Make predictions with *trained_model* on *Experiment*
22	Store *MAE*, *MAPE* & R^2 together with *run_time* in *hub_results*
23	**End**
24	**End**
25	**End**
26	**For each** *trained_model:*
27	Extract weighted average KPIs and *run_time* from *hub_results* & store in *final_results*
28	**End**
29	**End**
30	**For each** *trained_model:*
31	Extract weighted average KPIs & *run_time* from *final_results*
32	**End**

Line 27 and 31 of Table 7 ensure performance assessment for each model at both hub level and over all hubs respectively. In our experiments, we predict for weeks 12 and 13 at once. This slight variation from

Picnic's current process (predict weekly) enhances robustness in the test set, which is desired if we want to generalize our results. Furthermore, we compare the performance of our five models, with Picnic's current approach: the linear regression with two features, trained on both drive and stem datapoints (see Section 2.3.2). For the current approach we compute the same KPIs.

We run the hyperparameter tuning procedure of Section 4.1.3 as well as the experimental set-up, aiming to find the best prediction model, on a computer with an Intel Core i7-10850H CPU processor, which has 2.60GHz and 32GB RAM.

Afterwards, we can extract the *hub_results* and *final_results*. The results are aggregated from hub-specific to general outcomes, by calculating the weighted average MAE and MAPE, where the number of deliveries per hub are used as weights. Subsequently, we select the method that scores best on the weighted average MAPE and MAE. With the selected method, we will evaluate the impact on Picnic's main KPIs, OT and M/D, as outlined in Section 4.1.5.

4.1.5. Assessing impact on main KPIs

Once we selected the best prediction model for drive times, the final step of Figure 17 is left: evaluating the impact on OT and M/D. While an enhanced prediction model is valuable, it is essential to translate its predictions to Picnic's main KPIs to draw well-founded conclusions. To achieve this, we follow two approaches that we elaborate on in this section. One regarding trip drive error distributions and one regarding reconstructing the trip planning.

Total trip drive error distributions

The first approach deals with constructing error distributions, as we have seen in Chapter 2 as well. We start by creating the total trip drive error distribution of the trips in the test data. By going from single delivery errors to trip level, we get a more comprehensive view on the performance of the prediction model. At the end of a trip, we want to know how the prediction error of each drive segment accumulates. This accumulation will tell if the selected model is indeed suitable to predict drive times. A model can accurately predict individual drive segments, but consistent under- or overestimation can have noticeable negative impact on OT or M/D, if we accumulate over the twelve/thirteen drive segments in a trip. On the contrary, if it intermittently under- and overestimates drive times, the errors can compensate one another, which in turn will likely not negatively impact OT and M/D. This is visualised in Figure 18.

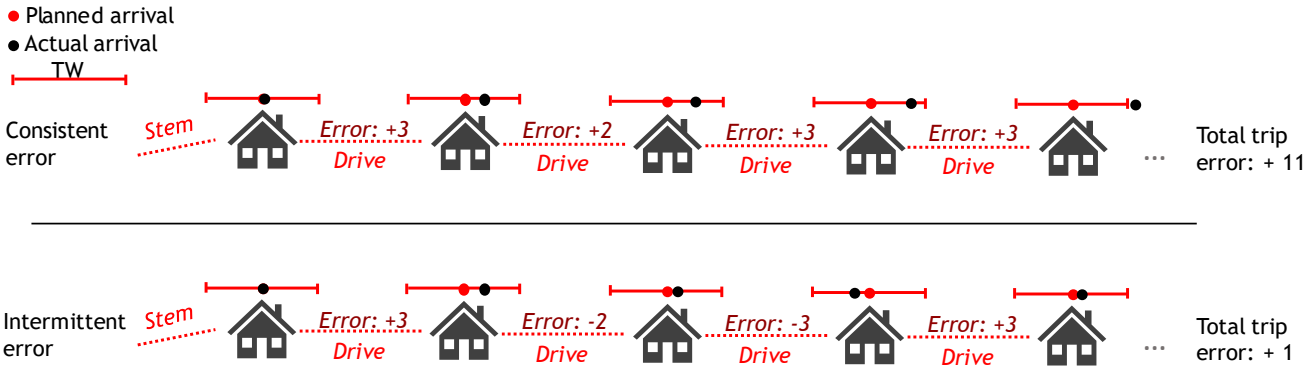

Figure 18. Visualization of consistent and intermittent error impact on total trip drive error.

With the consistent positive error in Figure 18, the fifth delivery will already not be within the time window (TW) anymore and therefore will impact the OT. We assume that stem, park and drop times are performed exactly according to planning, which we elaborate on later in this section.

To evaluate the impact with this approach, we focus on the tails of the total trip drive error distribution. Figure 19 provides an example of the distribution and the shaded relevant tails, where each trip with its corresponding total drive error is used as datapoint.

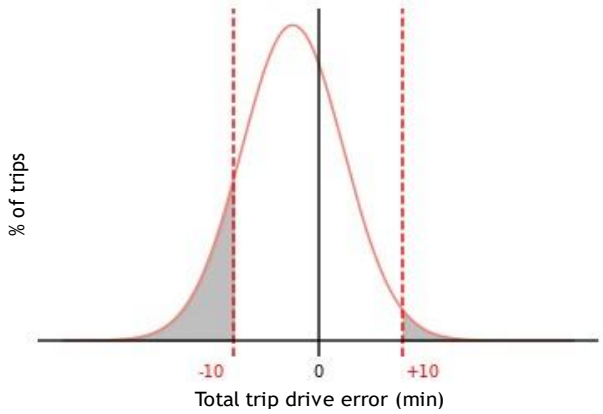

Figure 19. Example of total drive error distribution of trips (actuals – planned).

The total trip drive error is calculated as the sum of all individual drive errors (actual – planned time) in the trip. Given that the other segments have zero error, this total trip drive error distribution, provides insights in the impact on the main business KPIs purely caused by the new drive time predictions. The shaded tails in Figure 19 contain the trips where the TW ([-10; +10]) is not met, purely caused by drive time errors. The aim is to reduce the size of the tails with the new approach because that indicates the drive times are predicted more accurate on trip level. Hence, we compare the sizes of the tails between Picnic's current approach and our proposed prediction method. Note, in May 2024 this [-10; +10] changed to [-5; +15], so the relevant tails in Figure 19 then also shift accordingly.

The tail of the error distributions associated with OT is the right tail of Figure 19, indicating trips where the drive segment is consistently underpredicted. Trips exceeding a cumulative delay of more than ten minutes on drive will arrive late at ≥1 customer in that trip, due to the closing of the TW. This directly impacts Picnic's OT. For M/D, our focus shifts to the left tail of Figure 19, representing the fraction of trips that drive at least ten minutes quicker than planned. Trips that cumulatively drive more than ten minutes ahead of schedule, will arrive early at ≥ 1 customer in that trip. Implying waiting time for the TW. This results in wasting valuable time and, hence, increasing the total trip length and thus M/D.

The sizes of the tails provide indications of how OT and M/D are affected, solely caused by different drive time predictions. The key assumption for this analysis is that delay or time ahead of schedule is not compensated by another segment (e.g., drop). A runner could have a total delay of 15 minutes on drive segments for example, however this delay can be compensated if he/she performs drop times way quicker. In reality the runners deviate from planning on the other segments, but with this analysis we consider the change in risk of late or early arrivals, purely caused by differently planned drive segments.

It is worth noting that runners have less buffer before falling into the right tail if they depart late at the hub. Consequently, an early departure implies that arriving at a customer before the TW opens, becomes more attainable. We do not take late or early departures into account.

Reconstructing trip planning

The second approach to evaluate the impact on OT and M/D involves reconstructing trip planning of the trips in the test set. This entails replacing each originally planned drive time with its new predicted drive time and adjusting the ETA and TW accordingly. The planned stem, park and drop times are unchanged. The change in planned drive times will influence the ETA and TW (as timestamps) of all subsequent customers in the trip. See Figure 20 for an illustration.

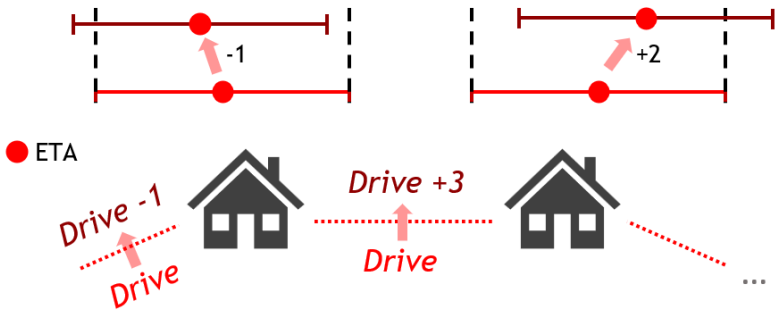

Figure 20. How ETA and TW shift if planned drive times change.

In Figure 20, if the new planned drive time for the first customer is one minute less than the originally planned drive time, the ETA and TW shift accordingly. If the new planned drive time to the second customer is three minutes longer than the original, the ETA and TW at this second customer shift two minutes later than the originally planned timestamps, accounting for the change at customer one as well.

We construct the table where we adjust the planned drive times for each delivery. Then in other columns we change the planned ETA and TW accordingly. From Picnic's data warehouse, we know the actual drive times realized in the test set. Then we can track what the OT would be given the new drive times and hence, new planned timestamps of ETA and TW. This simulated OT is then compared to the original OT in the test weeks. Additionally, for both Picnic's current approach and our chosen method, we sum all planned drive times for the deliveries in the test data. The total planned drive times directly impact the planned M/D, as the number of deliveries remains constant. By comparing this, we gain insight in the M/D impact of the selected method for predicting individual drive times.

Furthermore, from the reconstructed trip planning, we also construct delivery time error distributions, similar to Section 2.4.5, by subtracting the actual delivery timestamps from the planned delivery timestamps. Analysing the distribution tails, will in this case help to assess the fraction of late deliveries and instances where runners wait for time windows to open.

Unlike the drive segment analysis, this distribution accounts for all segments in a trip, reflecting errors in actual versus planned delivery times. Thus, the second part of this approach directly reveals the impact of a revised drive time prediction model on OT, and it offers indications for its effect on M/D, through the fraction of deliveries requiring waiting time.

Now, every step of Figure 17 is explained and we move on to the hyperparameter tuning in Section 4.2.

4.2. Calibrated model design & importance of tuning

At this point, we have executed the first steps of Figure 17 for each of the tuneable methods, on which we elaborate in this section: (i) The default model training, (ii) cross validating its performance as benchmark and (iii) hyperparameter tuning. Table 8 provides the configuration of the calibrated models.

Table 8. Final hyperparameter configuration per ML method.

Model	*Hyperparameter*	*Value*
Decision tree	Maximum features	14
	Maximum depth	8
	Minimum samples in leaf	20
Random forest	Number of estimators	291
	Maximum features	14
	Maximum depth	8
	Minimum samples in leaf	20
Neural Network	Learning rate	0,0013

	Batch size	68
XGB	Number of estimators	246
	Maximum depth	8
	Learning rate	0,0509

As mentioned in Section 4.1.3, the other hyperparameters of the models are kept at their default value, resulting in, for example, the NN activation function being the ReLu, as used in similar work (Akkerman & Mes, 2022) and the NN having one hidden layer with 100 neurons.

To evaluate the importance of the hyperparameter tuning, we also track the average performance on the 5-folds of cv data for both the default and the tuned models. Table 9 provides this benchmark performance of the four machine learning methods and the linear regression. Note: This is not the performance on unseen test data. Also, the run time is the combined time of training & predicting in the 5-fold cv and we trained (and hence predicted in this cv) across dispatch plans.

Table 9. Average 5-fold cross-validation performance of default and tuned models.

Model	*Type*	*Avg MAE (s)*	*Avg MAPE (%)*	*Avg R^2*	*Run time (s)*
Decision tree	Default	45,78	48,00	0,698	8,14
	Tuned	31,78	34,20	0,851	3,06
Random forest	Default	32,95	36,40	0,848	707,11
	Tuned	31,39	34,01	0,855	584,71
Neural network	Default	31,46	33,74	0,854	646,73
	Tuned	31,52	34,07	0,855	1048,03
XGBoost	Default	30,80	33,10	0,858	2,66
	Tuned	30,74	33,15	0,859	11,20
Linear regression	Default	33,48	34,25	0,834	0,51

In Table 9, all tuned models demonstrate equal or superior performance across the error metrics compared to the defaults, except for the NN. The tuned NN has a worse MAE and MAPE, but slightly better R^2. Due to the significantly longer runtime of the tuned NN compared to the default with only a marginal improved R^2, we decide to perform our analysis with the default NN model. For the other three ML methods, we utilize the tuned models since their performance is superior without run time issues. The XGBoost shows the best cv predictive performance, since for both the default and the tuned, this model has the best average MAE, MAPE and R^2.

4.3. Conclusions

In this chapter, we formulated the solution design. First, we decided to focus on improving the drive segment, as the most improvement potential is expected in that trip segment. We also identified the methods to research and compare. We selected four tuneable ML techniques along with an extended linear regression model to predict drive times.

Furthermore, we defined the dataset which we will use in this research, containing data from nine hubs during the first 13 weeks of 2024. We train and cross-validate on the first eleven weeks, and test on week 12 and 13. We chose our performance metrics (MAE, MAPE and R^2) and formulated an approach to learn the impact on OT and M/D of the best performing prediction model.

We already conducted hyperparameter tuning for the ML methods, and obtained benchmark performance for each model using a 5-fold cv. As the tuned NN showed only a slight improvement in the R^2 value, at a large increase in the computational cost, we will perform our experiments in Chapter 5 with the default NN, while using the tuned other ML methods, next to the linear regression model.

5. Results

This chapter presents the main results. Section 5.1 covers the predictive performance per model, while Section 5.2, addresses the best model's on-time and minutes per delivery performance. In Section 5.3, we extend our approach to all Dutch hubs. Overall, in this chapter we address RQ4 and its sub-questions.

5.1. Predictive performance

To learn the predictive performance of each of the five models, we execute the experimental set-up from Table 7. For aggregated model performance, we calculate the weighted average (WA) metrics with hub delivery volume as weights. Table 10 provides the results together with Picnic's current performance.

Table 10. Predictive performance of proposed models and Picnic's current, for week 12 & 13.

Prediction model	WA MAE (s)	WA MAPE (%)	WA R^2	Training time (s)
Decision Tree	31,89	33,60	0,828	0,6
Random Forest	31,46	33,44	0,832	144,7
Neural Network	30,80	30,25	0,834	149,0
XGBoost	31,26	32,73	0,832	3,0
Linear Regression	33,32	34,20	0,823	0,2
Picnic's current	37,86	46,74	0,809	NA

Opposite to the benchmark performance in Table 9 (the average 5-fold cv performance), the XGB does not show superior performance on the unseen test data. Rather the NN dominates the other models on each metric with the lowest WA MAE of 30,80 seconds, WA MAPE of 30,25% and an R^2 value of 0,834. However, the R^2 values of the RF, NN and XGB are close to one another. Due to the superior performance, we select the NN as the best method to predict individual drive times at Picnic.

Now we identified the NN as the best prediction method, we assess its predictive performance on hub level, compared to Picnic's current drive time planning approach. Table 11 lists the MAE, MAPE and each hub's delivery volume in week 12 & 13 (the test set) for both approaches. Again, the delivery volumes serve as weights for calculating the overall weighted average (WA) MAE and MAPE in the last row of Table 11.

Table 11. Predictive performance of NN and Picnic's current per hub.

Historic drive performance	Hub ID	Picnic's current		Neural network	
		MAE (s)	MAPE (%)	MAE (s)	MAPE (%)
Faster-than-average	UTC	44,23	56,83	34,33	32,86
	APN	37,15	51,06	27,68	29,96
	HFD	39,08	45,12	30,23	27,48
Average	LID	36,84	44,68	31,49	30,75
	BUS	37,89	43,24	32,60	32,16
	LEY	28,45	37,54	23,27	25,36
Slower-than-average	ALE	31,14	34,53	27,66	27,64
	EMM	27,40	33,56	24,73	25,01
	HRV	23,21	31,81	21,21	23,65
Overall WA		37,86	46,74	30,80	30,25

The results in Table 11 align with the analysis of Chapter 2, since the MAPE reduces as the hub's drive performance becomes slower. In Section 2.4.4, we have seen that the drive segment of an average trip in NL tends to be faster than planned. Consequently, hubs that generally perform drive segments slower-than-average have a lower MAPE on this segment, as the error (actuals – planned) is closer to zero.

The overall WA MAE and MAPE are largely influenced by hubs UTC, LID and BUS as the delivery volume is relatively large for these hubs. It strikes that there are no large outliers in MAPE values, when comparing the NN to Picnic's current performance. Such outliers are present in Picnic's current approach

for UTC and APN. The NN seems to have more stable predictive performance across the different hubs, both for the difference in historic hub drive performance, as for different delivery volumes. The performance of all prediction models per hub, is provided in Appendix F.

To validate that the predictive performance is generalizable, and that weeks 12 and 13 are representative to use as test data, we also train and test the models on random other subsets of the full dataset. Here, we test on week 3 and 11, while training on the other eleven weeks of the dataset. Then we run the same approach as for the previous test set (Table 7) and provide the results in Table 12.

Table 12. Predictive performance of proposed models and of Picnic's current, for weeks 3 & 11.

Prediction model	*WA - MAE (s)*	*WA - MAPE (%)*	*WA - R^2*	*Training time (s)*
Decision Tree	32,21	33,76	0,829	0,7
Random Forest	31,82	33,61	0,834	146,6
Neural Network	31,23	30,17	0,833	152,6
XGBoost	31,47	33,14	0,838	2,4
Linear Regression	33,80	33,58	0,816	0,2
Picnic's current	38,54	46,64	0,809	NA

Table 12 shows similar behaviour as Table 10, while using a different train/test split. However, in Table 12 the XGBoost has the highest R^2 value instead of the NN. But the NN still has the best MAE and MAPE. Despite minor variations in overall predictive performance between Tables 11 and 12, the similarity suggests that weeks 12 and 13 are representative and that the prediction models are generalizable.

5.2. Impact of neural network predictions on OT & M/D

As described in Section 4.1.5, we assess the impact of the selected prediction method, the NN, on *on-time %* (OT) and *minutes per delivery* (M/D) in multiple ways:

1. **Trip drive error distributions**: We assess the drive distributions on total trip level instead of on individual drive segment. This enhances practical relevance as errors on individual deliveries might compensate each other. Hence, Picnic is interested in predictive performance on trip level.
2. **Trip planning reconstruction**
   a. **Tracking OT and M/D**: We reconstruct the trips of the test set with the NN predicted drive segments. We solely change the planned drive times and assess the change in OT and M/D.
   b. **Delivery time error distribution:** We assess changes in planned delivery times (solely due to change in planned drive times) by evaluating the delivery time error distribution. Picnic values delivery time errors, as they reflect performance across all trip segments.

As mentioned in Section 2.2, as of May 2024, the TWs are placed [-5; +15] instead of [-10; +10]. Although TW placement has not affected predictive performance for individual deliveries, it plays a crucial role in assessing OT and M/D. Namely, OT is directly deduced from the TW, while potential waiting time due to the TW affects M/D. So, for practical relevance, we evaluate the main KPIs on weeks where the TW is [-5; +15] because this TW is used in the future and therefore the evaluation is more relevant with this TW. Hence, we shift our dataset to weeks 5-17, using weeks 16 and 17 for testing, and 5-15 for training the NN, keeping the same number of weeks as before. Using the default NN without hyperparameter tuning, and by taking the same preprocessing steps as in Section 4.1.1.1. This gave a test data set of 42.809 deliveries in 4284 trips. The resulting predictive performance on delivery level is provided in Table 13.

Table 13. Predictive performance of individual deliveries of NN and Picnic's current, for weeks 16 & 17.

Prediction model	*WA - MAE (s)*	*WA - MAPE (%)*	*WA - R^2*	*Training time (s)*
Neural Network	31,70	33,71	0,830	159,7
Picnic's current	37,53	46,12	0,804	NA

In Table 13, we see a similar R^2 value for the NN as in Table 10. Although the improvements in weighted average MAE and MAPE are not as big as in Table 10, the NN still shows substantial better performance versus Picnic's current. Hence, we will proceed our OT and M/D evaluation on weeks 16 and 17.

5.2.1. Total trip drive error distributions

The trip level predictive performance of the NN is assessed by creating the total trip drive error distributions of Picnic's current planned drive times and of the NN predicted drive times. On trip level, we gain insights regarding the compensatory behaviour of individual drive segment errors. Table 14 provides the relevant statistics belonging to the total trip drive error distributions of Figure 21, with the [-5; +15] TW boundaries in grey.

Table 14. Total trip drive error distribution relevant statistics belonging to Figure 21.

	Picnic's current	*Neural network*
Median error (s)	-118,5	-26
Inter Quartile range (s)	248	209
Total trip drive MAE (s)	191,29	142,85
Total trip drive MAPE (%)	17,74	12,09
≥5min ahead of drive schedule (%)	15,38	4,46
≥15min total drive delay (%)	0,14	0,16

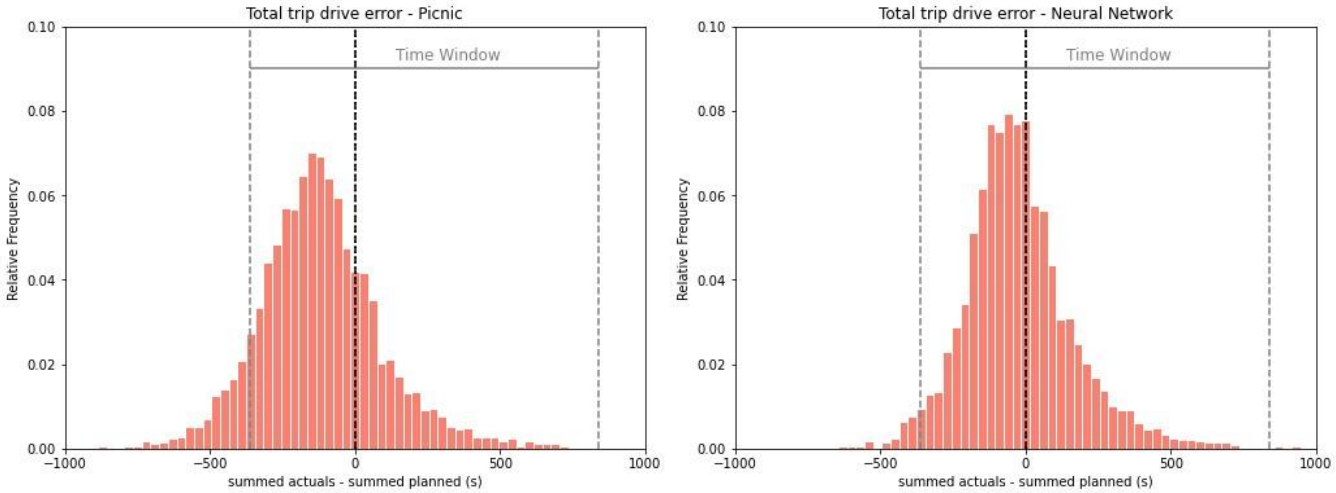

Figure 21. Total trip drive error distributions of Picnic's approach (a) and the NN drive predictions (b).

As Table 14 indicates, runners drive 118,5 seconds quicker than planned in the 'median' trip, whereas with the NN, this reduces to 26 seconds quicker. This supports the visual observation in Figure 21, that the NN error distribution is more centred and narrower, which is also indicated by a reduced IQR (from 248 to 209 seconds). Not only is the gain in the shift of the distribution, but the narrower distribution indicates more errors concentrated around zero. This suggest that the improved predictions on delivery level, positively impact the total trip drive error. Resulting in a better approximation of actual total drive times.

This is also reflected in the total trip MAE and MAPE. The NN improves predictive performance by reducing the total trip drive MAE from 191,29 to 142,85 seconds, which corresponds to reducing the total trip drive MAPE from 17,74% to 12,09%. Additionally, the fraction of runners who are excessively ahead of schedule reduced from 15,38% to 4,46% with the NN, indicating less waiting time.

However, these improvements come with a slight increase in trips with excessive total drive delay, rising from 0,14% to 0,16%. Hence, there is a slight increased risk that runners will not meet the time window anymore at ≥1 customer, purely caused by drive time delays.

5.2.2. Trip reconstruction

From Section 5.2.1, we learned that more accurate predictions of full trip drive segments indicate a significant reduction in waiting times at the cost of a slight increased risk of late deliveries. To learn what this exactly implies for M/D and OT, we reconstruct the full trips of the test set by changing the original planned drive times to the NN predictions. This way, we assess the operational implications for Picnic.

We incorporate the other segments as well and keep their planned times as they originally were. The same holds for all actual segment times. So, we assess the change in ETA, and hence TW placement, at each customer by changing the planned drive times only.

Main KPI performance

Since we do not change actual segment times, we should reconsider our definition of OT. Picnic considers a delivery on-time if it falls within the twenty-minute TW. So, early deliveries are not considered on-time. With the new TW placement due to NN predictions and unchanged actual segment times, runners could theoretically arrive at customers before the new (NN-based) TW opens. In such cases, runners would in reality wait for the TW to open. Therefore, in our reconstruction early runners are also counted as *on-time*. Which basically implies the definition of OT to shift to *not-late* deliveries. To keep the assessment fair, also the OT of Picnic's current approach is calculated according to this *not-late* definition. Additionally, we calculate the total planned drive times for both approaches and divide that by the total number of deliveries, to measure the M/D change. These key results provided in Table 15.

Table 15. Impact of NN drive time predictions on OT and M/D.

	Relative delta compared to Picnic's current (%)
OT	-1,00
Drive planned M/D	-6,77

The improved accuracy of drive time predictions by using a NN would have decreased the OT with 1,0% while improving the planned M/D with 6,77%. This improvement in M/D is due to overall tighter drive planning (Table 14 & Figure 21). It makes sense that this comes with a decrease in OT, since the risk of late deliveries increases, as the planning gets tighter. The NN, on average, predicts shorter drive times and therefore, addresses Picnic's tendency to overestimate drive times. However, this leads to a lower OT. For reference, Picnic's current OT is not above the desired 96% target, but well above 90%.

One could expect the NN to predict longer drive times in areas with greater uncertainties. Hence, that some originally late deliveries will not be late given the NN predictions. This is the case for 58 of the original late deliveries in the analysed data. So, it is not that the NN is only predicting shorter drive times.

Delivery time error distribution

Additionally, from the trip reconstruction, the delivery time error distributions are constructed. It is crucial to note that the delivery time error is calculated per delivery as the difference between the scheduled and the actual delivery times *as timestamp*. This error can be reviewed as *delay*. The error distribution inherently considers all trip segments and by assessing the tails, we gain more relevant insights beyond those presented in Table 15. We compare the delivery time error distribution of Picnic's current approach for planning drive times with the NN-based approach in Figure 22, with the TW in grey.

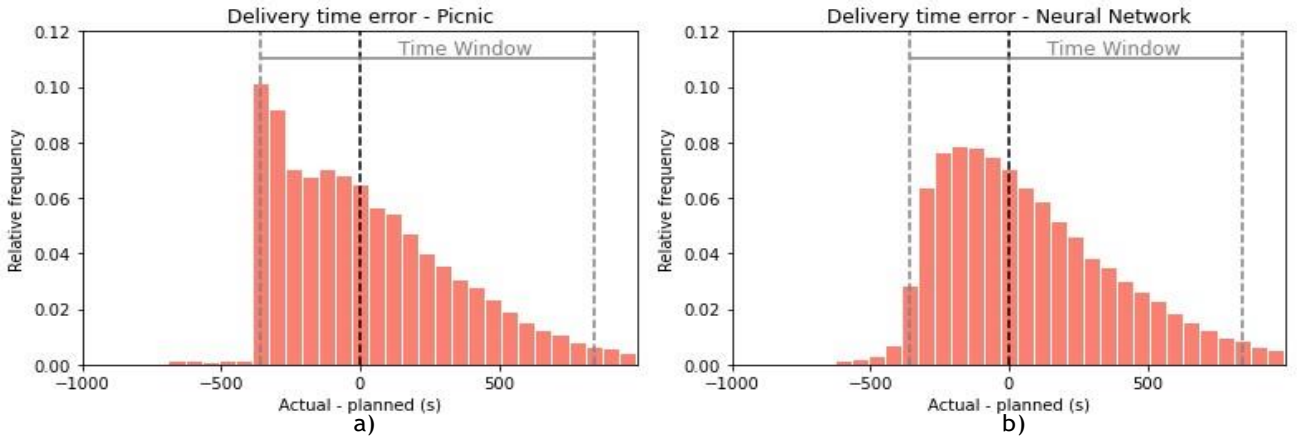

Figure 22. Delivery time error distribution of Picnic's current (a) and of the NN-based approach (b).

In Figure 22a, a striking peak shows that a high fraction of deliveries is performed as the TW opens. This indicates that runners who are five minutes or more ahead of schedule, must wait for the window to open. This implies that a significant fraction of runners is too fast for the [-5; +15] TW to be suitable for them. In contrast, Figure 22b lacks this peak because the NN usually plans shorter drive times than Picnic's current method, preventing more runners from being that far ahead of schedule and improving the efficiency. Table 16 provides relevant information regarding Figure 22.

Table 16. Median delay and first-minute delivery statistics for both approaches.

	Picnic's current	*Neural network*
Median delivery error / Median delay (s)	18	71
First-minute deliveries (%)	10,84	3,66

If Picnic would have used the NN to predict drive times, the average drive segment would have been planned 6,77% tighter (Table 15). In Table 16 we see this results in the median delivery being 71 seconds delayed compared to 18 seconds with the current approach. Also, the NN would have reduced the fraction of (undesired) first-minute deliveries from 10,84% to 3,66%. This should reduce waiting time. To estimate this reduction, we should understand where this waiting time is locked up.

Runners can either wait away from the customer, to stay out of sight, or when they have parked at the customer within a 50m radius from the destination. Waiting times for runners who wait further away are included in actual drive times and cannot be extracted with certainty. However, of those who wait close to the customers, we can extract waiting times within the park segment. In the park segment, first-minute runners wait on average almost one minute in the park segment, assuming that they do not park slower than an average runner. With less first-minute deliveries, the saved waiting time already accounts for 35,25% of the total planned drive time reduction (which totalled 6,77%).

In conclusion, from the delivery time error distributions we have learned that 35,25% of the M/D improvement, already come from the reduction in waiting times, potentially more. The other 64,75% is derived from the overall tighter drive time predictions of the NN.

5.2.3. Trip reconstruction – filtered delivery time error distributions

In Section 2.4.5 the delivery time error distributions are plotted alongside relevant statistics for specific filters: the peak shifts (S1 & S2), the peak days (Friday & Sunday), and different customer sequences in a trip. The same comparison is conducted here for the deliveries in our test set, using the current approach and our NN drive predictions. The comparison of the peak shifts and days can be found in Appendix G. However, the results from different customer sequences in a trip differ significantly from those in Section 2.4.5. Table 17 provides key insights corresponding to the error distributions shown in Figure 23.

Table 17. Filtered delivery time error distribution of current approach and NN drive predictions. Lates % adjusted for confidentiality reasons.

	Picnic's current			NN drive predictions		
Customer sequence	Median delay (s)	First-minute (%)	Lates (%)	Median delay (s)	First-minute (%)	Lates (%)
2	-2	8,66	3,22	4	6,29	3,24
6	15	9,26	4,01	51	3,53	4,49
10	27	12,34	5,04	94,5	3,07	5,69
13	42	13,08	6,07	147	2,45	7,19

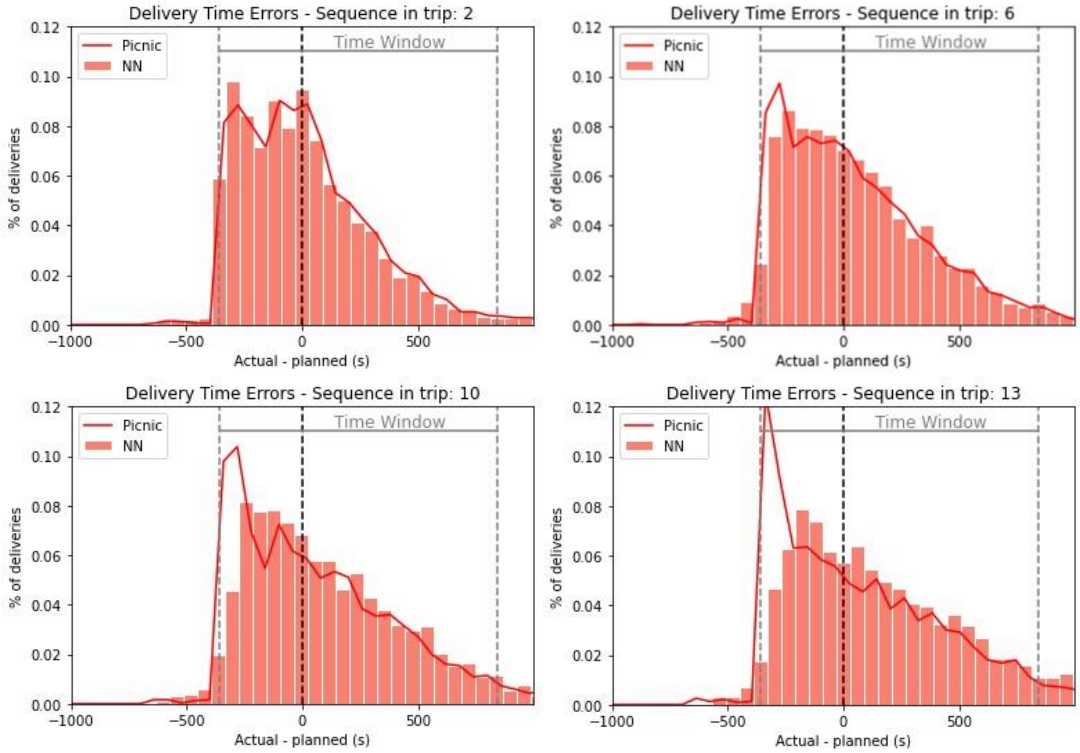

Figure 23. Error distributions of the two approaches for increasing customer sequence.

In Section 2.4.5, we learned that as the trip progresses, both the fractions of first-minute and late deliveries increase significantly. In Table 17, this pattern is also present at Picnic's current approach. However, with NN drive predictions this not the case. The fraction of late deliveries does also increase further on in a trip, but contrarily, the fraction of first-minute deliveries decreases. Suggesting that the error distribution does not widen like it does when using Picnic's current approach.

In Picnic's current approach, the quick runners could consistently be quicker than the planning, resulting in a rising peak of first-minute deliveries as the trip progresses (see Figure 23), and the slow runners were consistently losing time compared to planning resulting in more late deliveries. This latter also is the case when using NN drive predictions, but the peak of first-minute deliveries does not arise. Rather the opposite occurs: fewer runners are able to outperform the planning because of the NN's consistently tighter planned drive times. Hence, the first-minute delivery percentage decreases as the trip progresses.

At the second customer, 6,29% of the deliveries would occur in the first-minute and 3,22% would be late. At the thirteenth customer, only 2,45% of the deliveries occurs in the first-minute of the TW. This suggests that only few runners are waiting for the TW to open further on in the trip, compared to the second customer. However, 7,19% of the deliveries at these later customers is after the TW has closed. This suggest that the fixed [-5; +15] TW placement is not optimal for the different customer sequence. Hence, in Section 5.2.4 we will dive into variable TW placement.

5.2.4. Variable TW placement

In Section 5.2.2, we observed that OT (now defined as not-late %) decreased with 1,0% with the NN drive time predictions, despite the M/D improving with 6,77%. The aim of this research was to improve either OT or M/D without deteriorating the other metric. Hence, the negative OT impact should be addressed. As suggested in Section 5.2.3, the fixed TW placement may not be optimal throughout a trip when using NN drive time predictions. Since TW placement directly affects Picnic's OT, investigating variable TW placement throughout trips is worthwhile.

In the bar charts of Figure 23, at the second customer, a large bar appears when the TW opens, indicating wasted time. Few deliveries are made close to the end of the TW, suggesting that shifting the TW earlier could benefit the M/D without significantly impacting OT. However, for customers 10 and 13, relatively large bars appear after the TW, implying potential OT improvement if the TW shifts a (few) minute(s) later. Shifting it too much, however, could result in blocking quick runners at customers 10 and 13. Hence, M/D would deteriorate again due to increased waiting time. We propose three strategies for variable TW placement to enhance main KPI performance:

1) Shift the TW.
    i. Shift it earlier for customers at start of trip. This should even further reduce waiting time but might slightly decrease OT.
    ii. Shift it later for customers at the end of the trip. This should increase OT but will slightly reduce the saved waiting time.
2) Extend the TW.
    i. Extend TWs to 21 or 22 minutes later in the trip, to create more buffer after the ETA to boost OT.
    ii. Open the TW earlier at the start of the trip to accommodate quick runners.
3) Shift and extend the TW.

Per strategy, we formulated two scenarios to reach the desired OT of Picnic's current approach. With these scenarios we reconstructed the trips with the new TW placement. The scenarios are provided in Table 18. We cut all trips to a maximum of 14 deliveries, to keep the number of datapoints stable between different sequences. Some trips have more deliveries, but this is too few to fairly compare with customers early in the trip.

Table 18. Scenarios for shifting and extending TWs for different customer sequences.

		TW placement per sequence				
Strategy	*Customers*	*1 to 2*	*3 to 5*	*6 to 10*	*11 to 12*	*13 to 14*
-	Picnic's current	[-5; +15]	[-5; +15]	[-5; +15]	[-5; +15]	[-5; +15]
-	NN - Original	[-5; +15]	[-5; +15]	[-5; +15]	[-5; +15]	[-5; +15]
Shift	NN - Scenario 1	[-5; +15]	[-5; +15]	[-5; +15]	[-3; +17]	[-3; +17]
Shift	NN - Scenario 2	[-6; +14]	[-5; +15]	[-5; +15]	[-4; +16]	[-3; +17]
Extend	NN - Scenario 3	[-7, +15]	[-5; +15]	[-5; +15]	[-5; +16]	[-5; +17]
Extend	NN - Scenario 4	[-5; +15]	[-5; +15]	[-5; +15]	[-5; +17]	[-5; +17]
Shift + extend	NN - Scenario 5	[-6; +14]	[-5; +15]	[-5; +16]	[-5; +17]	[-5; +17]
Shift + extend	NN - Scenario 6	[-6; +14]	[-5; +15]	[-5; +16]	[-5; +16]	[-5; +17]

Per scenario, we calculate the OT (not-late %) and the fraction of first-minute deliveries. Since the planned drive times of the NN remain unchanged in this analysis, the planned M/D reduction remains 6,77%. However, different TWs will affect the M/D as they determine waiting time, meaning the real

M/D cannot remain the same. Originally, 35,35% of the M/D improvement was due to reducing first-minute deliveries with an average waiting time of just under a minute. Therefore, 64,75% of the M/D improvement is due to tighter drive time planning. In this analysis, we aim to estimate again the total saved M/D by translating the reduction in first-minute deliveries to saved waiting time.

Table 19 provides the relevant performance metrics for each scenario of variable TWs.

Table 19. Performance metrics per scenario of variable TW placement.

	Relative OT difference with Picnic's current (%)	*First-minute (%)*	*Relative M/D difference with Picnic's current (%)*
Picnic's current	-	10,84	-
NN - Original	-1	3,66	-6,77
NN - Scenario 1	-0,26	6,75	-5,76
NN - Scenario 2	-0,37	5,22	-6,27
NN - Scenario 3	-0,34	3,34	-6,90
NN - Scenario 4	-0,26	3,88	-6,73
NN - Scenario 5	+0,05	3,36	-6,90
NN - Scenario 6	-0,04	3,36	-6,90

In Table 19, all six scenarios show superior OT performance compared to the original NN OT. However, for Scenario 1, 2 and 4, the reduction in waiting time is not as significant as in the original NN scenario. It strikes that only two scenarios reach an OT close to Picnic's current: Scenarios 5 and 6. These also have the largest waiting time, and hence, M/D savings. These scenarios also feature wider TWs, which are better placed per different customer sequences due to the strategy of both shifting and extending TWs.

Scenarios 3 and 4 achieve almost similar OT performance as Scenarios 1 and 2 but with significantly better overall M/D reduction due to more saved waiting time. Scenarios 1 and 4, which have equal OT, share the same ending time of the TWs. Scenarios 5 and 6 have the same first-minute deliveries due to matching opening time of the TWs. Although Scenarios 1 and 2 show the worst combined performance, they are the only scenarios where the TWs are not extended, which can be risky.

It is uncertain how extending TWs will impact operations and customer satisfaction. Extending TWs might better capture the uncertainty, and therefore width of the error distributions in Picnic's trips. However, it may also influence runner behaviour. If runners perceive more buffer to still be 'on-time', they may not hurry as much. This might affect their performance in subsequent delivery trips. Especially because runners must have their legal break time between trips, what can cause undesired late departures from the hub, negatively impacting both M/D and OT.

Moreover, extended TWs can negatively affect customer satisfaction. Less precision of the attended-home delivery communicated TW, probably leads to lower customer satisfaction. Furthermore, customers who selected a time slot between (for example) 16.00 and 17.00 hrs, will more often receive a TW that can give the impression that the runner might deliver at 17.01 or 17.07 hrs. Even though the planned delivery time is before 17.00 hrs. Extended TWs will increase such instances, probably also leading to more actual deliveries after, in this case, 17.00 hrs. This will likely increase customer complaints.

In summary, variable TW placement can have uncertain implications for Picnic. These uncertainties are beyond this thesis' scope. Hence, the results of this variable TW analysis must be considered carefully.

5.3. Extension to all hubs

So far, we examined a subset of nine representative Dutch hubs in weeks 16 and 17. Now, we aim to assess the performance of the NN to predict drive times for all hubs in NL. To achieve this, we train a new NN

including all 60 hubs. Training and testing are conducted on the same weeks (5-15, and 16-17, respectively), and identical features are used (Appendix C). We evaluate the predictive performance per individual drive segment and on trip level. Table 20 presents the results, corresponding to the error distribution plots in Figure 24.

Table 20. Performance on individual and total trip drive error for all hubs in NL.

	Performance metric	*Picnic's Current*	*Neural network*
Individual drive segment error	R^2	0,813	0,833
	MAE (s)	37,853	32,934
	MAPE (%)	44,271	33,434
	Median error (s)	-18	-7
	IQR (s)	42	39
Total trip drive error	MAE (s)	183,451	145,860
	MAPE (%)	16,212	11,942
	Median error (s)	-108	-12
	IQR (s)	241	220

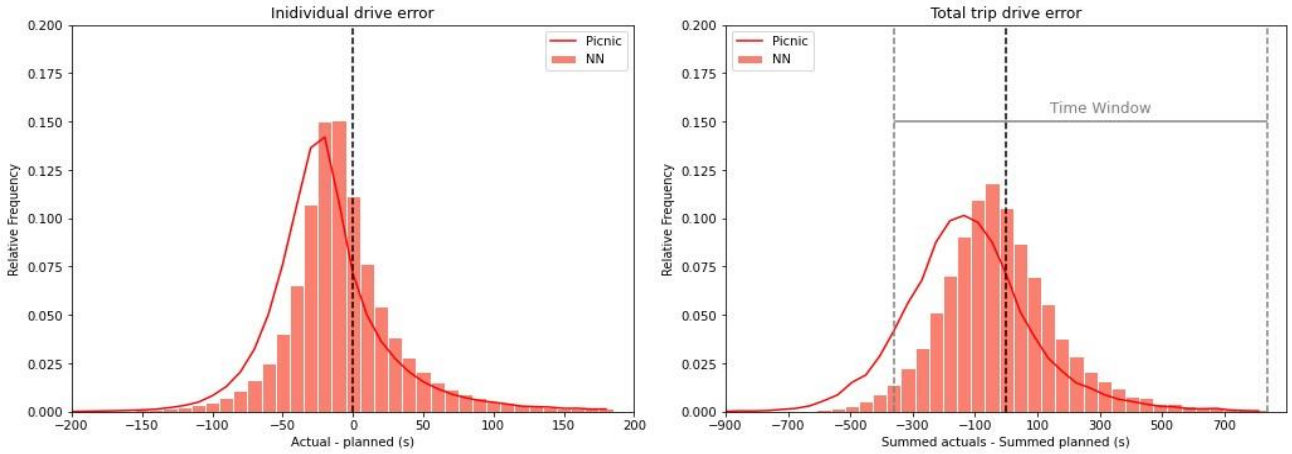

Figure 24. Individual drive error distribution (a) and total trip drive error distribution (b) of all hubs in NL.

Across all Dutch hubs, predictive performance per individual drive segment improved, resulting in a more centred error distribution with a median of -7 seconds (originally -18). The thinner IQR of 39 seconds (originally 42) also suggests that more drive segments are predicted closer to zero error. Aggregating to entire trip drive error, we observe a much lower MAE of 145,86 seconds, which originally was 183,45 seconds. The median is more centred with -12 compared to -108 seconds and the IQR decreased from 241 to 220 seconds.

Comparing these results with those of the nine hubs in Table 14, we note similar improvements, although the IQR reduction not as significant. Also, it strikes that the median error across all hubs is still negative, while it was significantly positive (+71 seconds) when evaluating only the nine hubs. This suggests that the median trip still maintains a small drive buffer when using a NN, ensuring runners usually stay on schedule. Figure 24b illustrates how the total trip drive error distribution better aligns with the TW.

The enhanced trip drive prediction performance, resulted in the main business KPIs and insights as provided in Table 21, belonging to the delivery time errors (=delays) of Figure 25.

Table 21. Impact of NN drive predictions on KPIs on all hubs in NL for weeks 16 and 17.

	Relative delta compared to Picnic's current (%)
OT	-0,92
Drive planned M/D	-6,87
First-minute deliveries	-67,44

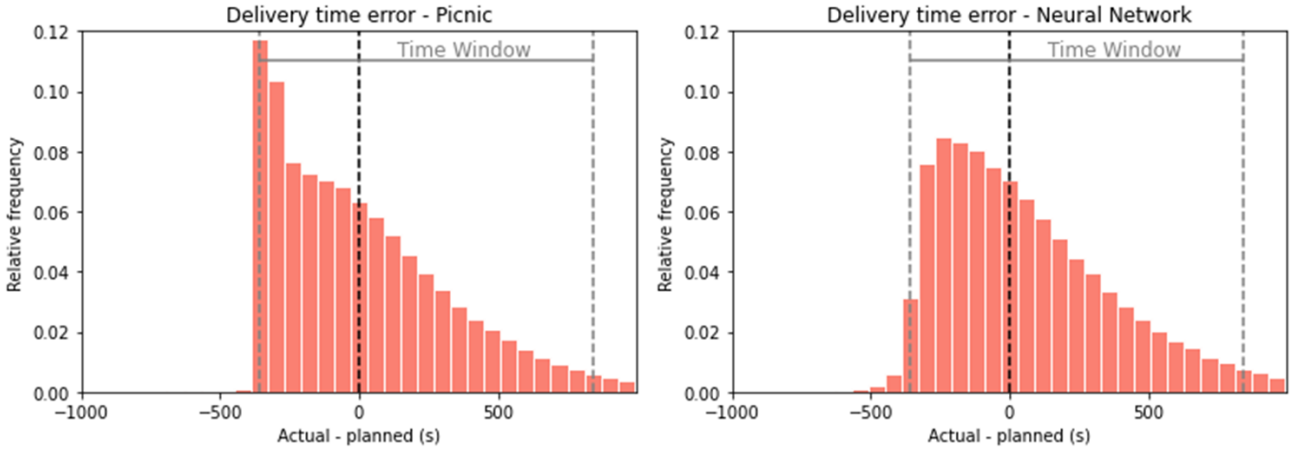

Figure 25. Delivery time error of Picnic's current (a) and the NN (b) approach for all hubs in NL.

In Table 21 and Figure 25, similar behaviour as for the nine hubs in Section 5.2.2 is observed. The OT decreased with 0,92% while improving M/D with 6,87%. With tighter drive times the risk of arriving at the opening of the TW reduces significantly to 4,07%, while the risk of late deliveries increased slightly. The delivery time error distribution of Figure 25b better fits the TW than Figure 25a, which we also observed in Section 5.2.2. For reference, with Picnic's current approach, the OT was again well above 90% and the planned drive M/D was between 2 and 3 minutes.

These results were discussed with the distribution team of Picnic, and even though the OT decreases slightly, and the fact that the 96% target is not yet met, the results are promising to discuss potential implementation steps (Chapter 6). Especially the significant efficiency increase is important to Picnic, as the company aims to enhance its profitability by reducing costs.

Al in all, for all hubs in NL the drive planning accuracy is enhanced when using a NN to predict the drive times. This method could also be applied to predict stem times more accurately, since stem times are variations of drive times and are also included in Mapbox' drive time matrix. However, the proposed approach for drive segments cannot be directly applied for stem times as well. This is because in our approach, we included quite some features regarding neighbourhood characteristics of the next customer. Stem segments are usually long drive segments, since the EPV does not drive between customers in a specific neighbourhood, but the runner must get to the neighbourhood first.

This is usually done via large roads where the runner is on for a significant amount of time. Hence, it does not make sense to include the neighbourhood characteristics of the first customer in the trip to predict the stem times, since only a small fraction of the stem time the runner will be in the neighbourhood and hence most of the stem uncertainty occurs at the big roads. Probably more features regarding the route should be included, such as number of traffic lights to pass or total distance outside neighbourhoods. However, this is not part of this thesis' scope.

5.4. Extension to all hubs – performance insights

In this section, we delve into overall results of Section 5.3 on a more granular level. So, all NL hubs are involved. We will analyse performance for different shifts, urbanity levels and the customer sequence in a trip. To illustrate the analysis, we create scatter plots where the horizontal axis provides the MAPE of single drive segments, while the vertical axis shows either the resulting OT or median delay. In the plots, the red data series represent the performance of Picnic's current method, while the blue series shows the performance of the NN predictions.

Figure 26 provides the OT (26a) and the median delay (26b) per shift against the horizontal drive MAPE. Delay is defined as the number of seconds the actual delivery is before (negative delay) or after (positive delay) the planned time, essentially the delivery timestamp error in seconds.

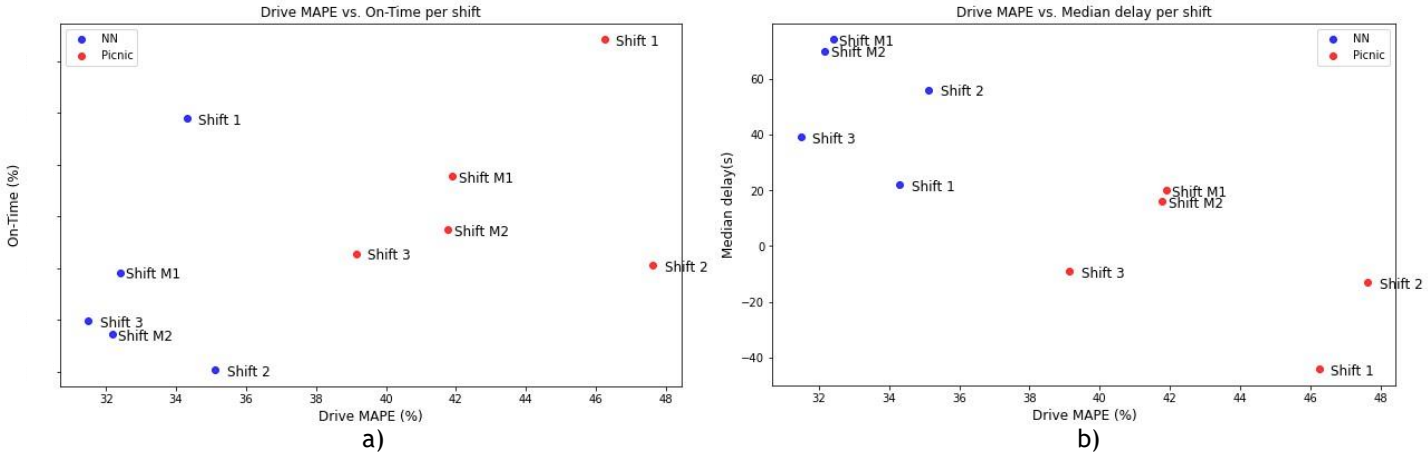

Figure 26. Drive MAPE vs. OT (a) and vs. median delay (b) per shift.

In the red series of Figure 26, S1 shows the lowest median delay and highest OT, with the median delivery more than 40 seconds ahead of schedule, indicating easy planning adherence. The relatively high drive MAPE of S1 is likely due to light traffic, leading to faster actuals compared to planning. S2 also has a negative median delay but lower OT, assumed to be due to high variability in drive segments (high MAPE) during peak traffic hours, especially between neighbourhoods traffic jams likely occur. Interestingly, S2 and S3 have similar OT and median delays despite differing MAPE performance.

In the blue series, all MAPE values are much closer to each other than in the red series, indicating that the NN effectively captures the different shift drive behaviours. Similar to the findings in Section 5.3, the NN generally shows lower OT performance than Picnic's current method for all shifts. Comparing S2 and S3 for the NN, S3 has significantly better OT and a lower median delay than S2, suggesting that the NN's tighter drive times better fit S3, which experiences lighter traffic than S2.

All hubs are divided into an urbanity level. For example, the hubs in big cities are 'metropolitan' hubs. In this way, all hubs are assigned to an urbanity level. For this distribution, please consult Appendix E. Figure 27 provides the scatter plot of each of the five urbanity degrees for both the NN and Picnic's current approach. The vertical axis provides the OT of the hubs in the different urbanity levels.

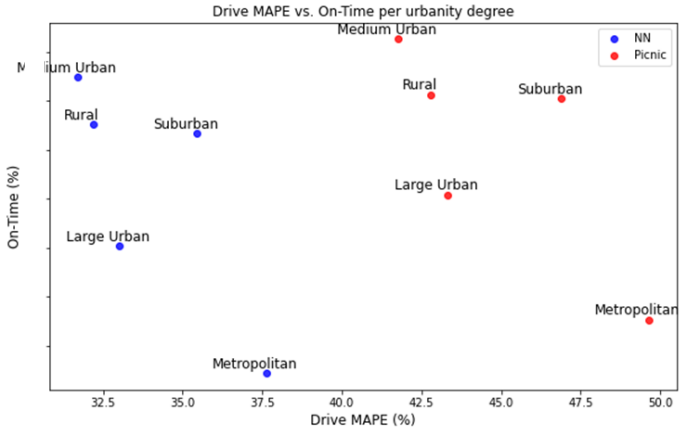

Figure 27. Drive MAPE vs OT per urbanity level.

From Figure 27 it is clear that the worst OT is achieved in metropolitan hubs. Also, the prediction error is the largest for this degree, in both the blue and red series. Apart from drive errors, OT is also largely influenced by park and drop segments, which are more uncertain in metropolitan areas as well due to building and street characteristics. Comparing the blue and red series, we see similar differences between the urbanity levels with a slightly worse OT, but more concentrated MAPEs for the blue series.

Figure 28 shows the OT (28a) and median delay (28b) per customer sequence against the MAE of drive time predictions. We exclude the first customer in the trip since we have the stem time towards that customer rather than the drive time.

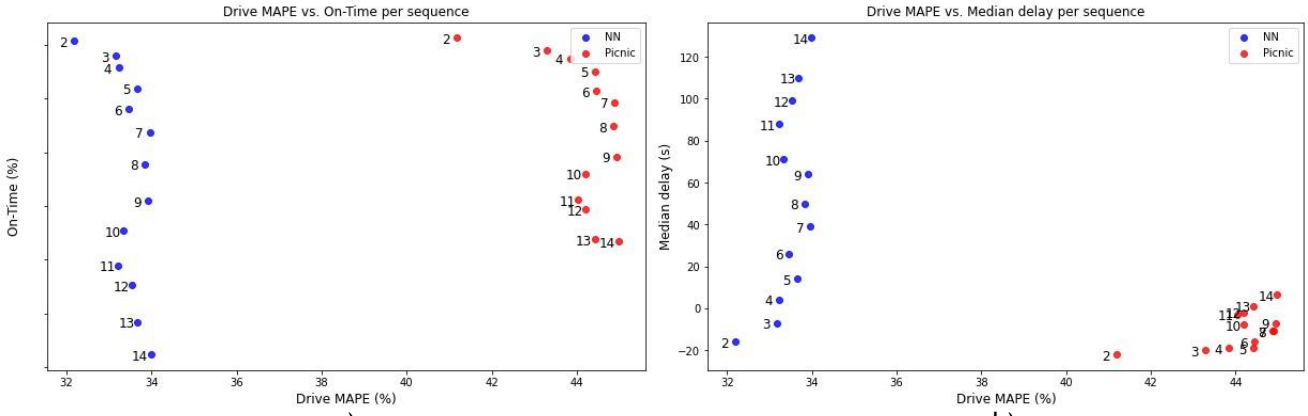

Figure 28. Drive MAPE vs. OT (a) and vs. median delay (b) per customer sequence.

In the red series of Figure 28a, a lower MAPE is observed at customers two and three. Early customers typically have longer drive times since routes often start with farther customers before visiting customers close to one another, making the distances for customers two and three longer, and hence, relative errors smaller. For other customers, the MAPE is relatively stable while the OT gradually decreases. With the NN, the MAPE difference between early and later customers is smaller. Also, the OT decreases more linearly than in the red series. In Figure 28b, the red series show a stable median delay, indicating that original (loose) drive times compensated for delays in other segments, keeping overall delay stable throughout the trip. With the NN, these buffers are removed, and median delay increases as the trip progresses. This could support further investigation of variable (or extended) TW placement in combination with NN drive predictions, as briefly discussed in Section 5.2.4.

In Appendix H, a few more insights are provided regarding granular performance, based on urbanity degree, customer sequence and weekday.

5.5. Conclusions

This chapter presented the results from the solution design of Chapter 4. First of all, the NN was selected as the best method for predicting drive times, demonstrating the greatest MAE and MAPE reduction compared to Picnic's current prediction process. The MAE decreased from 37,86 to 30,8 seconds, with a corresponding MAPE reduction from 46,74% to 30,25% per drive segment.

Despite individual segment accuracy improvements, potential bias can arise when assessing the total trip. Therefore, we translated the individual drive segment accuracy improvement to total trip drive errors. The NN total trip drive error distribution showed a more centred median error (-26 instead of -118,5 seconds) and a lower IQR (209 instead of 248 seconds), indicating a thinner error distribution and more precise predictions. Additionally, trips were reconstructed with the new planned drive times, resulting in a tighter planning of drive segments and a 6,77% M/D improvement. However, this led to a 1,0% OT reduction.

From the trip reconstruction we also evaluated the change in delivery time errors, and the fraction of first-minute deliveries. First-minute deliveries waited on average much more in the park segment, before delivering to the customer. A significant amount of the M/D improvement is driven by the reduction in waiting time (waiting for the TW to open) in first-minute deliveries from 10,84% to 3,66%.

This suggested that the TWs are not placed optimally, especially when looking at different customer sequences in the trip. Hence, we formulated some scenarios for variable TW placement with the aim of increasing the NN's OT, or to better capture the width of the error distribution at specific customers. This could result in the NN achieving an OT higher than the current approach, with even more saved waiting time. However, this requires extending TWs, which has unknown impact on customer satisfaction, hub operations and runner behaviour.

Finally, expanding our scope to all operating hubs in NL, similar patterns were observed, with a 0,92% decrease in OT, and a M/D gain of 6,87%. From Section 5.4 we learned that the NN makes drive predictions with better capturing the differences between shifts, urbanity degrees and customer sequences by showing more concentrated MAPE values. With NN drive predictions the OT gradually decreases, with an increasing median delay as the trip progresses, supporting the need for further exploring the potential of variable TW placement.

Despite not meeting Picnic's 96% OT target, these results are convincing for the distribution team of Picnic, to discuss the potential implementation.

6. Implementation

In this chapter we will deal with the implementation of the new method to predict drive times. Section 6.1 outlines the architecture and ownership of the model, while Section 6.2 provides (other) practical implications of the implementation. We will answer RQ5.

6.1. Required architecture & ownership

Since the NN is a machine learning (ML) model, appropriate IT-architecture and clear responsibilities are crucial for its implementation. This section outlines the required architecture for developing the NN to predict drive times, focussing on its placement in the planning process and the roles involved.

In Section 2.3.1 & 2.3.2, we briefly introduced the master planning process (MPP), which oversees the entire Picnic supply chain, from the fulfilment centre to the customer. MPP covers more than the last-mile deliveries, optimizing steps like tote and rack assignments to trucks and hubs. Picnic's vehicle routing optimization model, VROOM, is part of MPP and determines delivery routes using the drive time matrix and customer drop times. Since VROOM is integrated into MPP, MPP ensures the right inputs for VROOM. In this research, we focussed on tweaking the drive time matrix. Currently, this drive time matrix undergoes a transformation from the requested Mapbox matrix to the Picnic-tuned drive time matrix using a linear regression model (Section 2.3.2). However, we propose a NN for this transformation.

Picnic's ML department, AAA, owns all company-wide ML models including the drop time model (Section 2.3.2), which also is a NN to predict customer drop times. The drop time model is a collaboration between MPP (owned by the distribution department) and AAA. MPP namely supplies features, such as delivery size and weight, while AAA extracts additional features from the data warehouse, like historical drop times, to predict drop times with the NN. Given AAA's experience with NNs, implementing another NN should be rather straightforward.

Figure 29 compares the current (As-is) and proposed (To-be) states of creating the drive time matrix, with the To-be architecture based on the drop time model architecture.

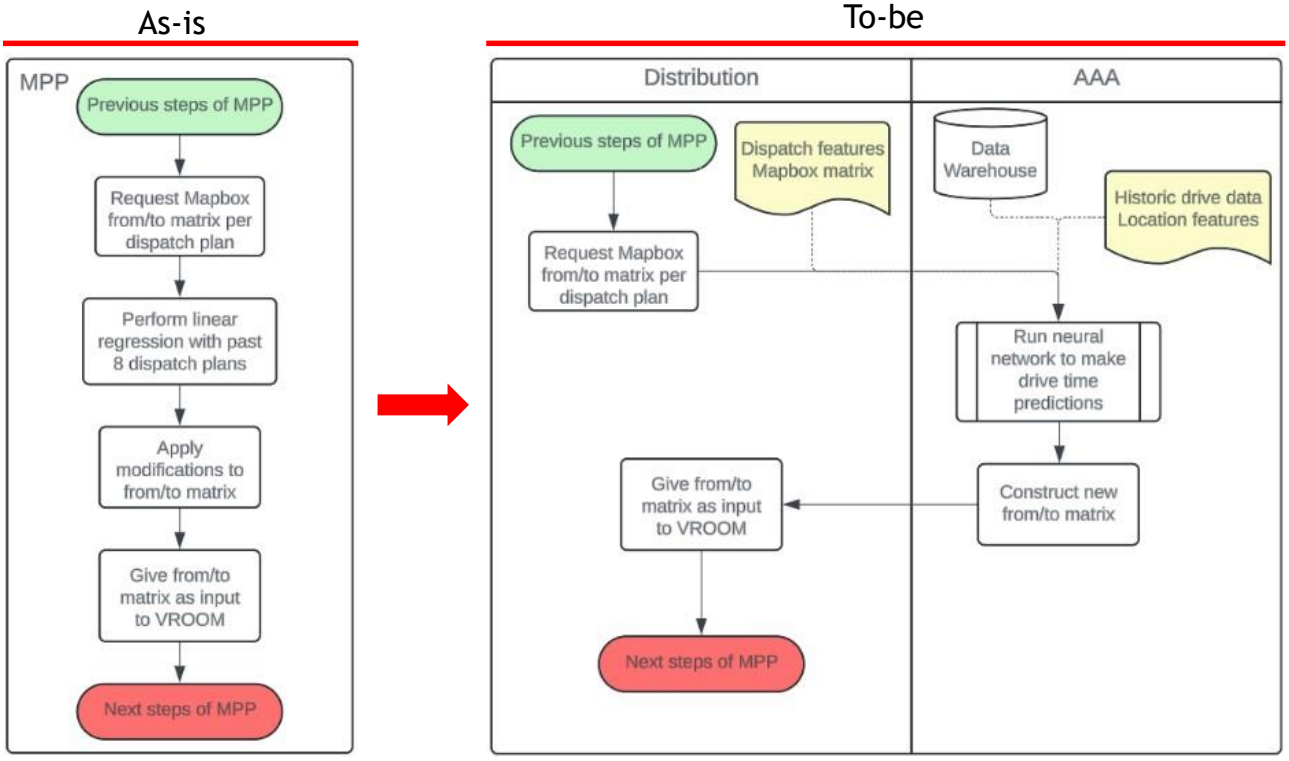

Figure 29. As-is and To-be architecture of the drive time prediction process at Picnic.

The To-be situation in Figure 29 shows the communication between the distribution team and AAA. With the NN to predict drive times, Picnic remains dependent on Mapbox' estimates, as the set of customers in a dispatch plan is rarely identical. Thus, Mapbox will still provide drive time estimates between all customers in a dispatch plan. MPP requests this Mapbox matrix and communicates it to AAA along with dispatch features (hub, day, shift). AAA extracts other relevant features from the data warehouse, such as historical drive times and customer location data. AAA then runs the NN to predict each drive time in the matrix. The newly constructed from/to matrix is communicated back to MPP to be used in VROOM.

The drop time model of AAA is weekly updated using the past year's data, with the two most recent weeks as the test set and the remaining 50 weeks for training. The drop time model is not updated if the newly trained model performs worse on the test set compared to previous week's model. This procedure applies to all ML models owned by AAA. Since each dispatch plan combination of hub, day and shift occurs weekly, the NN to predict drive times should also be trained weekly, and the same update condition can be applied. Experts of AAA estimate the entire implementation time to be 20 workdays, including construction, testing the model and conducting pilot phases for certain hubs.

6.2. Practical changes

Picnic is a young and dynamic company with a big focus on improvement. Therefore, the change required for successfully implementing a NN to predict drive times is probably welcome. However, we must consider the practical changes that come with implementing the proposed architecture.

The proposed change in model involves minimal process changes. All tasks and flows are not completely new, as they already occur at Picnic for the stop time model. Also, the proposed ownership is not new and therefore, the transition to this new procedure should be smooth.

At the distribution side of Picnic, the main change involves replacing the current system that performs the linear regression, by an information flow that sends the estimates of Mapbox together with the dispatch plan features to AAA. This implies that the distribution analyst responsible for maintaining the linear regression will become responsible for setting up this flow.

The ML department needs to develop a few new systems, information flows and performance tracking dashboards. For the other ML models, AAA also has performance dashboards which they check frequently. Once these construction steps are taken, the employees of AAA need to extend their daily tasks with also tracking the drive time performance. One important point to consider, is that in the future the drive time model might require improvements or adjustments. Though it is currently unknown what these improvement iterations entail, it is crucial to account for this in AAA's capacity planning to ensure the team can handle such iterations.

Effective implementation and maintenance of the NN model will require clear communication and collaboration between the distribution team and AAA. Once the model is running, both teams will share responsibility for tracking performance and making necessary adjustments and improvements. AAA will handle software adjustments, while the distribution team will provide operational insights to identify and address performance issues. AAA namely does not have the operational knowledge which the distribution team does have. However, given the stop time model experience this should not be an issue.

If the implementation is successful, it is essential to communicate the effects of this change to all runners and hubs through a central message. This message should inform runners that, in general, there will be less waiting time because of tighter drive planning. Also, it is important to explain that the new model considers additional features to predict individual drive segments more accurately.

During field discussions, some runners expressed a dislike for waiting because they prefer to complete their trips as quickly as possible. Also, they thought that drive planning could really benefit from granular distinctions regarding locations and timing. By involving runners in understanding the reasons behind the changes, they will feel more engaged in the project and probably will be convinced why the changes will benefit trip planning. This will increase the chance that they accept the tighter drive time plans, as it is important to acknowledge that tighter drive times may increase stress, especially for slower runners.

We summarize the practical implications for the stakeholders (i.e., people affected) briefly in Table 22.

Table 22. Stakeholder implications of implementing the NN at Picnic.

Stakeholder	*Potential impact*
Customers	Could experience slightly more late deliveries. But win in efficiency also benefits customers since Picnic might have less need to increase prices.
Picnic - Runners	Will generally experience less slack time in their trip. Which could lead to runners having to hurry more often. However, the quick runners can finish their trips earlier due to less waiting time and more accurate predictions.
Picnic - Hub operators	May face increased pressure to ensure that the vehicles depart on time. This results in sometimes helping with loading of EPVs to make sure the runners can take their legal break time while still departing on schedule.
Picnic - AAA	Owns an extra ML model, which should be trained and maintained. Furthermore, AAA will have to develop the ML model such it communicates and runs smoothly with MPP. This increases workload at AAA.
Picnic - Distribution team	Should monitor the planned drive time accuracy, together with OT and M/D performance. The reduced OT should be compensated for, by initiating other projects to improve this. Before the NN is implemented, the distribution team should first convince Picnic's management that the win in M/D is more important than the loss in OT. That this truly is the right step to improve Picnic's overall business performance.

6.3. Conclusions

This chapter outlined the proposed implementation of a NN to predict drive times. A similar architecture as the current drop time model owned by AAA is proposed, since that is a NN as well. Hence, a similar but extra collaboration between the distribution team (who owns MPP) and AAA is proposed. The estimated implementation time including a pilot phase is twenty working days. We also briefly assessed some practical implications that affect the relevant stakeholders. Mainly the tighter planning will have some practical implications for customers, runners and hub operators, while the slight loss in OT could imply a shift in Picnic's business strategy. However, we do not consider these implications deal-breakers.

7. Conclusion & recommendations

This chapter summarizes the key findings and concludes this research. In Section 7.1 we draw the main conclusions. In Section 7.2. we list our recommendations for Picnic. Section 7.3. discusses the limitations and opportunities for further research.

7.1. Conclusions

In this thesis, we aimed to improve Picnic's last-mile delivery planning accuracy. A too low fraction of on-time (OT) deliveries was experienced, and Picnic aimed to reduce minutes per delivery (M/D) to increase profitability by means of tighter planning and waiting time reduction. Our focus was on refining drive time planning accuracy, recognizing significant improvement potential within this segment due to the observed buffer, and rather simple current approach.

Picnic currently uses a linear regression model to plan drive times, which is fitted on the eight most recent dispatch plans of the to-be-predicted dispatch plan. The model contains the estimates of Mapbox and distinguishes between stem and drive datapoints. However, we explored five different prediction models to replace this procedure, with the end goal of enhancing planning accuracy.

After a literature review, we trained and tested five models to predict drive segments: (i) a linear regression model, (ii) a decision tree, (iii) a random forest, (iv) a neural network (NN) and (v) an extreme gradient boost. Utilizing eleven weeks of training data and two weeks of testing data, we incorporated dispatch plan features such as hub, day and shift, along with customer location features like postal zip, car density and urbanity degree. Another key input feature was still Mapbox' estimate because a lot of from/to combinations of customers are unique at Picnic. We started with a subset of nine representative hubs to execute the prediction model selection approach.

We tuned the machine learning (ML) models with a Bayesian hyperparameter optimization with five cross-folds to improve predictive performance while mitigating overfitting. Ultimately, we found that the (default) NN exhibited the highest predictive performance per individual drive segment in our test data. With an R^2-score of 0,834, a MAE of 30,80 seconds and MAPE of 30,25% it dominated all other models. Additionally, it significantly outperformed Picnic's current approach which had an MAE of 37,86 seconds and a MAPE of 46,74%. We aggregated the performance on individual drive segments to total trip level, to assess potential bias which could negatively impact overall trip planning.

Fortunately, the NN had no negative (in)consistencies, resulting in a total trip drive MAE of 142,85 seconds, which is a 25,3% improvement. However, the improved prediction accuracy was mainly due to shifting the entire error distribution closer towards the centre. While the distribution became thinner, as the IQR decreased from 248 to 209, the main impact was the tighter planning of most drive times. This ensured less runners outperforming planning, leading to less waiting for the time window (TW), and overall improved M/D of 6,77%. However, it also decreased the OT with 1%, as tighter planning makes it more difficult to deliver on time.

The TW placement significantly affected both M/D and OT, especially, when considering different customer sequences. During a trip, originally runners had more chance to outperform planning, but also the risk of late deliveries increased. Initially, runners had more opportunities to surpass planning, but this also increased the risk of late deliveries, leading to a widening error distribution as the trip progressed. The NN's tighter planning reduced instances of runners consistently outperforming planning, resulting in a narrower error distribution. Although, the risk of lates still increased significantly during trip progression. Hence, we explored the potential of flexible TW placement.

Flexible TW placement, or variably extending TWs for different customer sequences, can greatly improve OT and M/D. Variable TWs can better capture the delivery error distribution at specific customer sequences, as it can be shifted as needed. Additionally, an extended TW can better capture the uncertainty of the distribution as it is wider and hence, more deliveries will fall within the TW. However, the impact on runner behaviour, Picnic's hub operations and customer satisfaction remains uncertain.

Our drive time prediction approach extended to all Dutch hubs yielded similar predictive performance and findings regarding OT and M/D. In weeks 16 and 17, the OT decreased with 0,92% and the planned M/D improved by 6,87%, with the NN drive time predictions compared to Picnic's original approach. Due to the substantial gain in M/D, the OT decrease is considered acceptable.

Finally, we have learned that Picnic's dedicated machine learning department, AAA, owns the drop time model for predicting customer drop times with a NN. Since this is a collaboration between the distribution team of Picnic and AAA, it should be possible to also implement a NN to predict drive times using similar architecture and ownership. The practical implications of implementing a NN to predict drive times are not considered deal-breakers.

To conclude, this research proposed a NN to predict drive times at Picnic. This will lead to enhanced planning accuracy on this specific segment. As a result, on short term Picnic will gain efficiency by a decrease in M/D. But this comes at a cost of a reduction in OT performance as the drive segment plannings buffer is largely removed. Hence, we reached this thesis' research goal to some extent: despite the slight OT reduction, the planning accuracy is enhanced, and the M/D is decreased significantly.

7.2. Recommendations

Based on the conclusions in Section 7.1, we formulate the following recommendations for Picnic:

1) Implement a (default) NN to predict drive times between customers in the last-mile delivery. This model proved to have the best predictive performance, enhancing planning accuracy and M/D performance while slightly, but acceptably, decreasing OT due to tighter planning.
2) Implement the NN with similar architecture and ownership as the existing drop time model.
3) Document the purpose, procedure steps, input features and other relevant aspects of this drive time prediction well and publish this in the information repository. This ensures transparency and potential of improving the model / procedure even further.
4) Improve the prediction accuracy of other segments in the last-mile delivery. If drop and park are more accurately predicted, the removal of the drive buffer due to the NN becomes trivial due to (generally) tighter drive time predictions, leading to increased OT.
5) Revise the 96% OT-target together with the twenty-minute TW. It seems as if there is too much natural uncertainty in the last-mile of Picnic, such that 96% of deliveries within twenty minutes seems only theoretically feasible. Especially when also taking the late deliveries into account caused by supply issues of the Fulfilment centres, or insufficient capacity of runners.
6) Research the effects of shifted and/or extended TWs on customer satisfaction and hub operations. Variably changing the TWs has led to significant improvements for the nine hubs in our test set, but the practical implications of such changes are unknown.
7) Ensure equal data availability in Germany in France, to be able to develop and research a NN for drive time predictions in those countries as well. Currently, for Germany and France not the same data (and hence features) is available. This implies that the NN cannot be simply converted to these countries.

7.3. Limitations & further research

The research conducted includes some limitations, and therefore opportunities for further research. Here we list these:

1) Not all hyperparameters of the ML models were tuned. Fully tuning could even better fit the data of Picnic, and therefore increase predictive performance. Also, it could be the case that other models would have been better than the NN. However, it is greatly time-consuming to tune all parameters and the risk of overfitting increases. But it could still be interesting to further research the effect of tuning more hyperparameters, and different value ranges for the hyperparameters.
2) While predicting drive times, we only included location features of the customer to drive towards. We did not incorporate features regarding the previous customer. Therefore, future research on taking those features into account is interesting.
3) We only included features in our ML models that were directly present in the data warehouse of Picnic. But one could think of many more features to include, that could even further improve drive planning accuracy. Think of expected weather conditions, the average driving experience of a hub's runner pool, etc.
4) In our models, we excluded outliers in both training and testing. Picnic did not want to train and test on outliers, because they do not want to predict for those instances. However, in reality outliers always occur, so it might be relevant to explore performance with outliers included.
5) When Picnic requests the estimates of Mapbox, they receive estimates tailored to the coming dispatch plan, considering expected traffic conditions at that time. However, our NN also incorporates the shift as feature, together with the historical data. This way, time-dependency is both included in Mapbox' estimate and captured by our NN, which might be redundant or even generates noise in the data. Therefore, an interesting opportunity for further research is to obtain Mapbox' raw estimates without the time-dependency.
6) In this research, we consistently trained on eleven weeks of data because too much data could give an unfair comparison with Picnic's current approach and Mapbox had a big update at the end of 2023. If Picnic is to implement the proposed NN, AAA will likely train the model on 50 weeks. However, it might be interesting to explore and optimize the number of weeks to train on.
7) For weeks 16 & 17, we did not train all five prediction models again to select the best model. Given that the XGB was very close to the NN performance-wise for weeks 12 & 13, it could be the case that for weeks 16 & 17 it is better. However, we went into further analysis with the NN directly. In the future, it is worth testing all models again on a larger set of weeks, to arrive at a best model with even more certainty.
8) A feature that we did not include but could have high predictive value, is the exact customer ID. Especially for repeat customers this could be relevant. Even though the from/to combinations are rarely recurring, the model could learn a lot from previous actuals and previous Mapbox' estimates, if it can connect the coming drive segments directly to previous drive segments.
9) Should Picnic not decide to implement the NN to predict drive times, we recommend them to further research the current linear regression to predict drive times. That method is namely trained on both stem and drive segments in one dataset, whereas those segments have striking differences in terms of characteristics, but also in terms of times. Stem times are namely way longer. Therefore, it might make more sense to split these segments and then fit a linear regression. This way, Picnic predicts a segment by solely training on that segment. Rather than training on a dataset that contains noise of the other segment.

8. Bibliography

Agatz, N., Campbell, A. M., Fleischmann, M., & Savels, M. (2008). Challenges and Opportunities in Attended Home Delivery. In *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 379–396). Springer US. https://doi.org/10.1007/978-0-387-77778-8_17

Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L. M., Poss, M., & Requejo, C. (2013). The robust vehicle routing problem with time windows. *Computers & Operations Research*, *40*(3), 856–866. https://doi.org/10.1016/j.cor.2012.10.002

Akkerman, F., & Mes, M. (2022). Distance approximation to support customer selection in vehicle routing problems. *Annals of Operations Research*. https://doi.org/10.1007/s10479-022-04674-8

Akkerman, F., Mes, M., & Lalla-Ruiz, E. (2022). *Dynamic Time Slot Pricing Using Delivery Costs Approximations* (pp. 214–230). https://doi.org/10.1007/978-3-031-16579-5_15

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, *237*, 82–117. https://doi.org/10.1016/j.ins.2013.02.041

Braaten, S., Gjønnes, O., Hvattum, L. M., & Tirado, G. (2017). Heuristics for the robust vehicle routing problem with time windows. *Expert Systems with Applications*, *77*, 136–147. https://doi.org/10.1016/j.eswa.2017.01.038

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, *99*, 300–313. https://doi.org/10.1016/j.cie.2015.12.007

Brochu, E., Cora, V. M., & de Freitas, N. (2010). *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*.

Campbell, A. M., Vandenbussche, D., & Hermann, W. (2008). Routing for Relief Efforts. *Transportation Science*, *42*(2), 127–145. https://doi.org/10.1287/trsc.1070.0209

Chen, C., Twycross, J., & Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. *PLOS ONE*, *12*(3), e0174202. https://doi.org/10.1371/journal.pone.0174202

Clarke, G., & Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, *12*(4), 568–581. https://doi.org/10.1287/opre.12.4.568

Corona-Gutiérrez, K., Nucamendi-Guillén, S., & Lalla-Ruiz, E. (2022). Vehicle routing with cumulative objectives: A state of the art and analysis. *Computers & Industrial Engineering*, *169*, 108054. https://doi.org/10.1016/j.cie.2022.108054

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, *6*(1), 80–91. https://doi.org/10.1287/mnsc.6.1.80

Dror, M. (1993). Modeling vehicle routing with uncertain demands as a stochastic program: Properties of the corresponding solution. *European Journal of Operational Research*, *64*(3), 432–441. https://doi.org/10.1016/0377-2217(93)90132-7

Dror, M., Laporte, G., & Louveaux, F. V. (1993). Vehicle routing with stochastic demands and restricted failures. *ZOR Zeitschrift F*r Operations Research Methods and Models of Operations Research*, *37*(3), 273–283. https://doi.org/10.1007/BF01415995

Ehmke, J. F., Campbell, A. M., & Urban, T. L. (2015). Ensuring service levels in routing problems with time windows and stochastic travel times. *European Journal of Operational Research*, *240*(2), 539–550. https://doi.org/10.1016/j.ejor.2014.06.045

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, *57*(4), 1472–1483. https://doi.org/10.1016/j.cie.2009.05.009

Elatar, S., Abouelmehdi, K., & Riffi, M. E. (2023). The vehicle routing problem in the last decade: variants, taxonomy and metaheuristics. *Procedia Computer Science*, *220*, 398–404. https://doi.org/10.1016/j.procs.2023.03.051

Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, *140*, 106242. https://doi.org/10.1016/j.cie.2019.106242

Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, *11*(2), 109–124. https://doi.org/10.1002/net.3230110205

Gillett, B. E., & Miller, L. R. (1974). A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research*, *22*(2), 340–349. https://doi.org/10.1287/opre.22.2.340

Giuffrida, N., Fajardo-Calderin, J., Masegosa, A. D., Werner, F., Steudter, M., & Pilla, F. (2022). Optimization and Machine Learning Applied to Last-Mile Logistics: A Review. *Sustainability*, *14*(9), 5329. https://doi.org/10.3390/su14095329

Gmira, M., Gendreau, M., Lodi, A., & Potvin, J.-Y. (2020). Travel speed prediction based on learning methods for home delivery. *EURO Journal on Transportation and Logistics*, *9*(4), 100006. https://doi.org/10.1016/j.ejtl.2020.100006

Han, J., Lee, C., & Park, S. (2014). A Robust Scenario Approach for the Vehicle Routing Problem with Uncertain Travel Times. *Transportation Science*, *48*(3), 373–390. https://doi.org/10.1287/trsc.2013.0476

Heerkens, H., & Winden van, A. (2017). *Solving managerial problems sytematically*. Noordhoff Uitgevers.

Kara, mdat, Yeti, B., & Kadri, M. (2008). Cumulative Vehicle Routing Problems. In *Vehicle Routing Problem*. InTech. https://doi.org/10.5772/5812

Kenyon, A. S., & Morton, D. P. (2003). Stochastic Vehicle Routing with Random Travel Times. *Transportation Science*, *37*(1), 69–82. https://doi.org/10.1287/trsc.37.1.69.12820

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York. https://doi.org/10.1007/978-1-4614-6849-3

Lang, M. A. K., Cleophas, C., & Ehmke, J. F. (2021). Anticipative Dynamic Slotting for Attended Home Deliveries. *Operations Research Forum*, *2*(4), 70. https://doi.org/10.1007/s43069-021-00086-9

Laporte, G., Gendreau, M., Potvin, J.-Y., & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, *7*(4–5), 285–300. https://doi.org/10.1016/S0969-6016(00)00003-4

Lenstra, J. K., & Kan, A. H. G. R. (1981). Complexity of vehicle routing and scheduling problems. In *Networks* (Vol. 11, pp. 221–227). https://doi.org/10.1002/net.3230110211

Li, X., Tian, P., & Leung, S. C. H. (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, *125*(1), 137–145. https://doi.org/10.1016/j.ijpe.2010.01.013

Lin, L., Handley, J. C., Gu, Y., Zhu, L., Wen, X., & Sadek, A. W. (2018). Quantifying uncertainty in short-term traffic prediction and its application to optimal staffing plan development. *Transportation Research Part C: Emerging Technologies*, *92*, 323–348. https://doi.org/10.1016/j.trc.2018.05.012

Lin, S. (1965). Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, *44*(10), 2245–2269. https://doi.org/10.1002/j.1538-7305.1965.tb04146.x

Mackert, J. (2019). Choice-based dynamic time slot management in attended home delivery. *Computers & Industrial Engineering*, *129*, 333–345. https://doi.org/10.1016/j.cie.2019.01.048

Mantovani, R. G., Horváth, T., Rossi, A. L. D., Cerri, R., Junior, S. B., Vanschoren, J., & de Carvalho, A. C. P. de L. F. (2018). *Better Trees: An empirical study on hyperparameter tuning of classification decision tree induction algorithms*. https://doi.org/10.1007/s10618-024-01002-5

Matijević, L. (2023). General variable neighborhood search for electric vehicle routing problem with time-dependent speeds and soft time windows. *International Journal of Industrial Engineering Computations*, *14*(2), 275–292. https://doi.org/10.5267/j.ijiec.2023.2.001

Qiu, B., & Fan, W. (David). (2021). Machine Learning Based Short-Term Travel Time Prediction: Numerical Results and Comparative Analyses. *Sustainability*, *13*(13), 7454. https://doi.org/10.3390/su13137454

Russell, R. A., & Urban, T. L. (2008). Vehicle routing with soft time windows and Erlang travel times. *Journal of the Operational Research Society*, *59*(9), 1220–1228. https://doi.org/10.1057/palgrave.jors.2602465

Strauss, A., Gülpınar, N., & Zheng, Y. (2021). Dynamic pricing of flexible time slots for attended home delivery. *European Journal of Operational Research*, *294*(3), 1022–1041. https://doi.org/10.1016/j.ejor.2020.03.007

Taghipour, H., Parsa, A. B., & Mohammadian, A. (Kouros). (2020). A dynamic approach to predict travel time in real time using data driven techniques and comprehensive data sources. *Transportation Engineering*, *2*, 100025. https://doi.org/10.1016/j.treng.2020.100025

Tan, S.-Y., & Yeh, W.-C. (2021). The Vehicle Routing Problem: State-of-the-Art Classification and Review. *Applied Sciences*, *11*(21), 10295. https://doi.org/10.3390/app112110295

Tang, J., Pan, Z., Fung, R. Y. K., & Lau, H. (2009). Vehicle routing problem with fuzzy time windows. *Fuzzy Sets and Systems*, *160*(5), 683–695. https://doi.org/10.1016/j.fss.2008.09.016

Taş, D., Gendreau, M., Dellaert, N., van Woensel, T., & de Kok, A. G. (2014). Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, *236*(3), 789–799. https://doi.org/10.1016/j.ejor.2013.05.024

van der Hagen, L., Agatz, N., Spliet, R., Visser, T. R., & Kok, L. (2024). Machine Learning–Based Feasibility Checks for Dynamic Time Slot Management. *Transportation Science*, *58*(1), 94–109. https://doi.org/10.1287/trsc.2022.1183

Waßmuth, K., Köhler, C., Agatz, N., & Fleischmann, M. (2023). Demand management for attended home delivery—A literature review. *European Journal of Operational Research*, *311*(3), 801–815. https://doi.org/10.1016/j.ejor.2023.01.056

Wolter, J., & Hanne, T. (2024). Prediction of service time for home delivery services using machine learning. *Soft Computing*, *28*(6), 5045–5056. https://doi.org/10.1007/s00500-023-09220-7

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zhang, T., Chaovalitwongse, W. A., & Zhang, Y. (2012). Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Computers & Operations Research*, *39*(10), 2277–2290. https://doi.org/10.1016/j.cor.2011.11.021

Zheng, Y., & Liu, B. (2006). Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm. *Applied Mathematics and Computation*, *176*(2), 673–683. https://doi.org/10.1016/j.amc.2005.10.013

Appendices

A. *Current delivery time error distributions*

Figure 30 provides the delivery time errors of Fridays and Sundays, given Picnic's current drive planning approach. The data analysed is the same as in the rest of Chapter 2. Picnic performs relatively many deliveries on Friday and Sunday, however both days are subject to different traffic conditions.

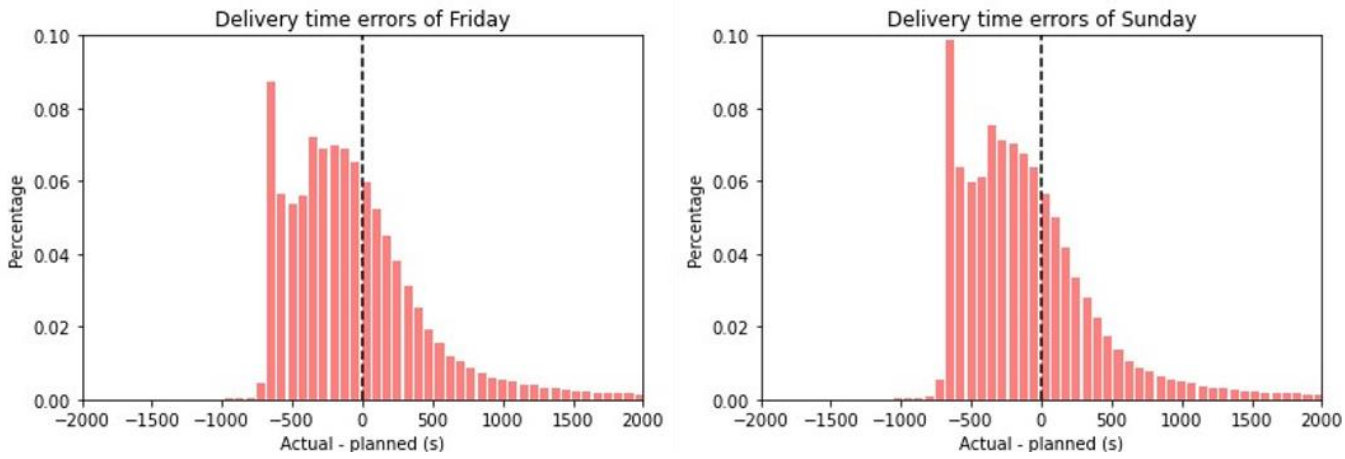

Figure 30. Delivery time errors of peak delivery days.

Even though the delivery errors incorporate all trip segments, and just the stem and drive conditions are assumed to be significant different between the delivery days, We see a larger peak at the opening of the TW on Sunday. Table 5 in Section 2.4.5 provided the corresponding statistics, and it became clear that apart from this larger opening peak, also less late deliveries occur on Sundays.

Figure 31 provides the distributions per customer sequence corresponding to Table 5.

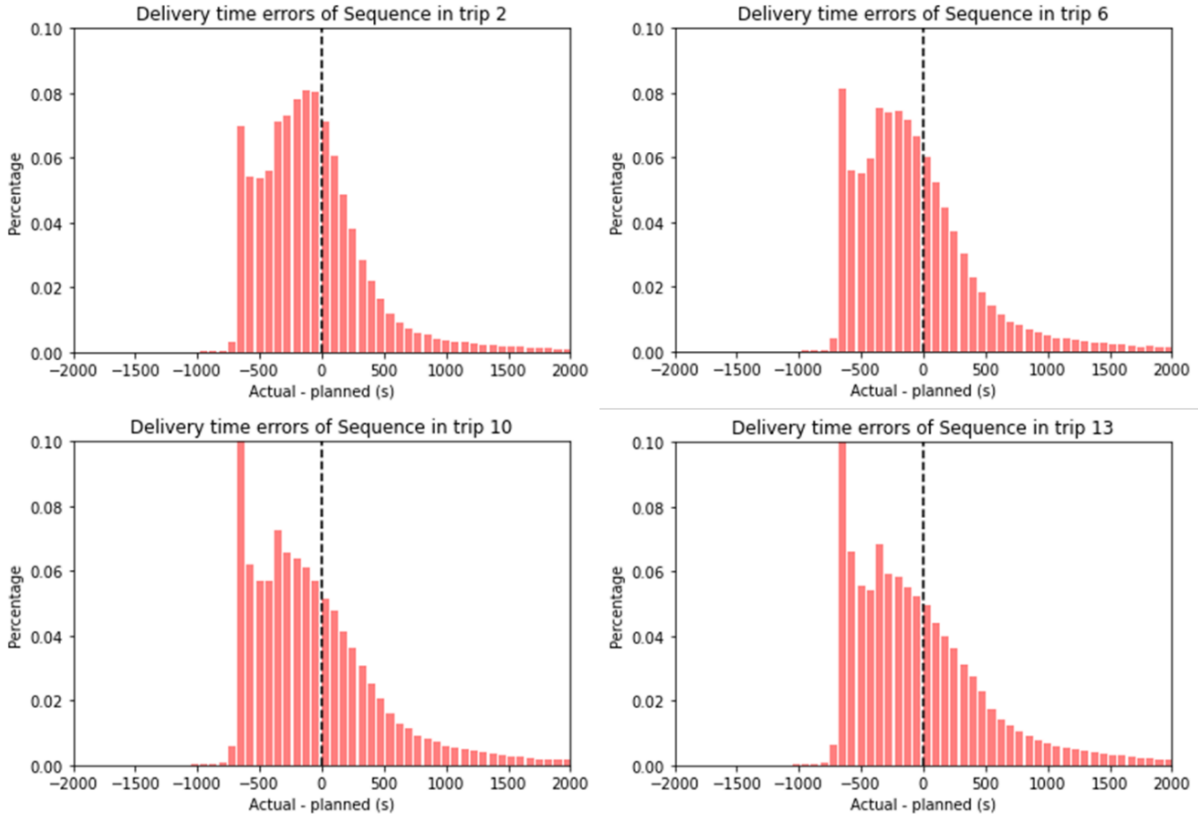

Figure 31. Delivery time error distributions per customer sequence.

B. Current drive performance of selected hubs

Table 23 provides the current performance of each of the nine selected hubs on the drive segment on trip level. We selected three hubs quicker than average on the drive leg, three hubs performing averagely on the drive leg and three hubs performing slower on the drive leg. The error is calculated as the actual drive time minus the planned drive time, and then summed over all drive segments in a trip.

Table 23. Picnic's current drive performance for the nine selected hubs.

Selection criterion	Hub	Q1 drive error (s)	Median drive error (s)	Q3 drive error (s)
Overall		*-275*	*-139*	*-6*
Faster-than-average on drive	UTC	-370	-215	-61
	APN	-335	-212	-88
	HFD	-342	-210	-71
Average on drive	LID	-269	-135	-5
	BUS	-275	-143	-9
	LEY	-206	-169	-17
Slower-than-average on drive	ALE	-170	-52	101
	EMM	-140	-55	50
	HRV	-159	-57	28

Clear is that all hubs have a negative median drive error. Implying that the median trip gets a nice buffer in the planning, for performing the drive segments. For the average and faster-than-average hubs, 75% of the trips have more time planned for the drive legs than they actually take, indicated by the negative Q3 error.

C. Dataset feature description

Table 24. Description of the features used in this thesis.

Feature	Description
Holiday week	Boolean value indicating whether the week is a holiday week or not
Hub ID	Categorical value containing the three-letter hub ID
Urbanity	Categorical value containing one of five urbanity degrees
Weekday	Categorical value containing one of the seven weekdays
Shift	Categorical value in the set {S1, S2, S3, M1, M2}
Day part	Categorical value indicating the morning or afternoon. Overarching of shifts.
Rounded latitude	Continuous value representing the rounded latitude to 2 decimals, implying approximately 1km in between
Rounded longitude	Continuous value representing the rounded longitude to 2 decimals, implying approximately 1km in between
Postal zip	Categorical value representing the postal zip code of the customer
Drive planned Mapbox	Continuous value representing Mapbox' prediction for the drive time
Address density	Continuous value representing the number of addresses per square kilometre in the neighbourhood
Singel family percentage	Percentage of households in the neighbourhood, being a single-family household
Avg distance to main road	Continuous value representing the average distance between the centre of the neighbourhood and a main road.
Population density	Continuous value representing the population density in the specific neighbourhood of the customer
Car density	Continuous value representing the car density in the specific neighbourhood of the customer

D. Feature correlation matrix

Figure 32 provides the correlation matrix of the predictor features in the dataset. From this matrix, we decided to remove the population density feature, as it was highly correlated to both address density and car density. The other features are kept, even though some high other correlations exist, because we think they are important to describe the specific data record completely and relevantly.

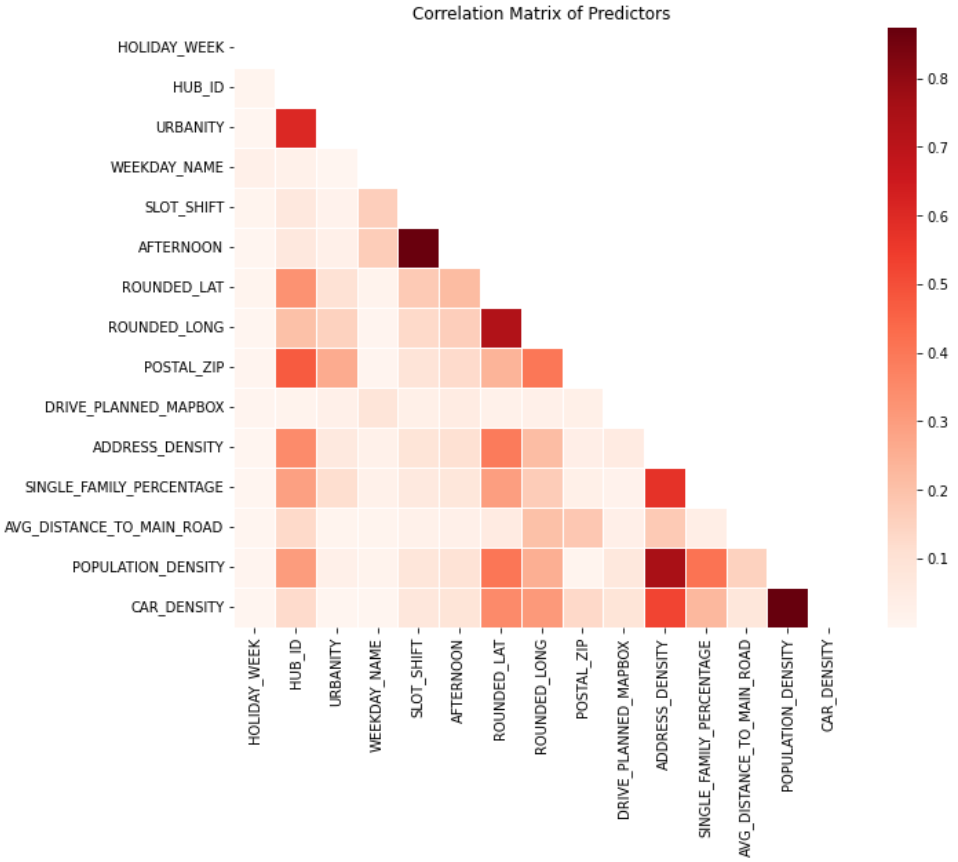

Figure 32. Correlation matrix of features in the dataset.

E. Urbanity degree distribution

Table 25 provides the distribution of the hubs among the five urbanity degrees in NL. This distribution is made by employees of Picnic and used in this research.

Table 25. Distribution of NL hubs among urbanity degrees.

Urbanity degree	*Hubs*
Metropolitan	AMS, AMW, DVT, RSP, DHG, UTC, EIN
Large urban	NIJ, BRD, TLB, HTB, DOR, APE, DVR, ENS, HAA, MST, GRQ, GOU, NWG, ZTM, AMN
Medium urban	HFD, HLM, ZIT, RID, HKE, ALK, AMV, ZWO, ROO, ALS, LID, HRN, OSS, HEE, PUM, ALM, LWR, DOE, HRV, DRA, ASS, EMM, OMD, VEN, ARN
Suburban	ALE, ALW, EDV, YPB, AWD, WAT, MEP
Rural	SPI, APN, BUS, BER, VDM, GOE, LEY, KTS, HRD, MDL

F. Predictive performance per model per hub

Table 26 provides the predictive performance per hub, of all constructed prediction models in this research. The performance is on weeks 12 and 13, so the test set 5.1.

Table 26. Predictive performance per hub for each researched prediction model.

Hub	Prediction model	WA - MAE (s)	WA - MAPE (%)	WA - R^2
ALE	Decision Tree	29,65	32,13	0,866
	Random Forest	28,8	31,92	0,873
	Neural Network	27,66	27,64	0,873
	XGBoost	27,7	28,82	0,877
	Linear Regression	32,5	33,92	0,834
	Picnic's Current	31,14	34,53	NA
APN	Decision Tree	28,62	32,61	0,867
	Random Forest	28,07	32,51	0,874
	Neural Network	27,68	29,96	0,873
	XGBoost	27,19	30,4	0,875
	Linear Regression	30,36	33,81	0,863
	Picnic's Current	37,15	51,06	NA
BUS	Decision Tree	33,04	32,28	0,802
	Random Forest	32,76	32,17	0,805
	Neural Network	32,6	32,16	0,809
	XGBoost	32,86	32,53	0,808
	Linear Regression	34,1	32,69	0,799
	Picnic's Current	37,89	43,24	NA
EMM	Decision Tree	25,04	26,58	0,87
	Random Forest	24,52	26,06	0,87
	Neural Network	24,73	25,01	0,871
	XGBoost	24,37	26,97	0,874
	Linear Regression	24,78	25,76	0,872
	Picnic's Current	27,40	33,56	NA
HFD	Decision Tree	31,45	31	0,866
	Random Forest	31,01	30,78	0,869
	Neural Network	30,23	27,48	0,869
	XGBoost	30,59	29,4	0,871
	Linear Regression	33,48	31,98	0,86
	Picnic's Current	39,08	45,12	NA
HRV	Decision Tree	20,99	25,72	0,865
	Random Forest	20,76	25,2	0,867
	Neural Network	21,21	23,65	0,86
	XGBoost	21,29	26,63	0,865
	Linear Regression	22,34	25,73	0,855
	Picnic's Current	23,21	31,81	NA

LEY	Decision Tree	25,41	30,38	0,89
	Random Forest	24,91	30,29	0,896
	Neural Network	23,27	25,36	0,901
	XGBoost	22,62	26,35	0,909
	Linear Regression	26,59	32,12	0,884
	Picnic's Current	28,45	37,54	NA
LID	Decision Tree	32,14	33,38	0,821
	Random Forest	31,78	33,4	0,826
	Neural Network	31,49	30,75	0,825
	XGBoost	32,06	33,83	0,822
	Linear Regression	33,34	33,86	0,819
	Picnic's Current	36,84	44,68	NA
UTC	Decision Tree	36,05	38,77	0,787
	Random Forest	35,68	38,55	0,791
	Neural Network	34,33	32,86	0,797
	XGBoost	35,63	37,6	0,789
	Linear Regression	37,52	39,09	0,784
	Picnic's Current	44,23	56,83	NA

For most hubs, the XGBoost has the highest R^2-value, however for LID, UTC & BUS the neural network has the highest R^2-value. These hubs have the highest delivery volume and, therefore greatly influence the R^2.

G. Delivery time error comparison for peak shifts and days

Table 27 provides the statistics, belonging to Figure 33. These figures provide the delivery time error distribution of both Picnic's approach and the NN to predict drive times for peak shifts S1 and S2 and peak days Friday and Sunday. The statistics and distributions belong to week 16 and 17 across the nine selected hubs. The error is the difference in seconds between de actual and the planned delivery timestamps.

Table 27. Key statistics for the peak shifts and days, belonging to Figure 33. Lates (%) multiplied with a random number for confidentiality.

	Picnic's current approach			Neural Network		
Filter value	*Median error (s)*	*First-minute*	*Lates*	*Median error (s)*	*First-minute*	*Lates*
Shift 1	-21,5	11,67%	2,47%	43	3,35%	2,98%
Shift 2	22	11,22%	6,27%	87	3,18%	7,11%
Friday	10,5	10,53%	4,84%	66	2,89%	5,40%
Sunday	81	10,16%	6,95%	103	4,93%	7,34%

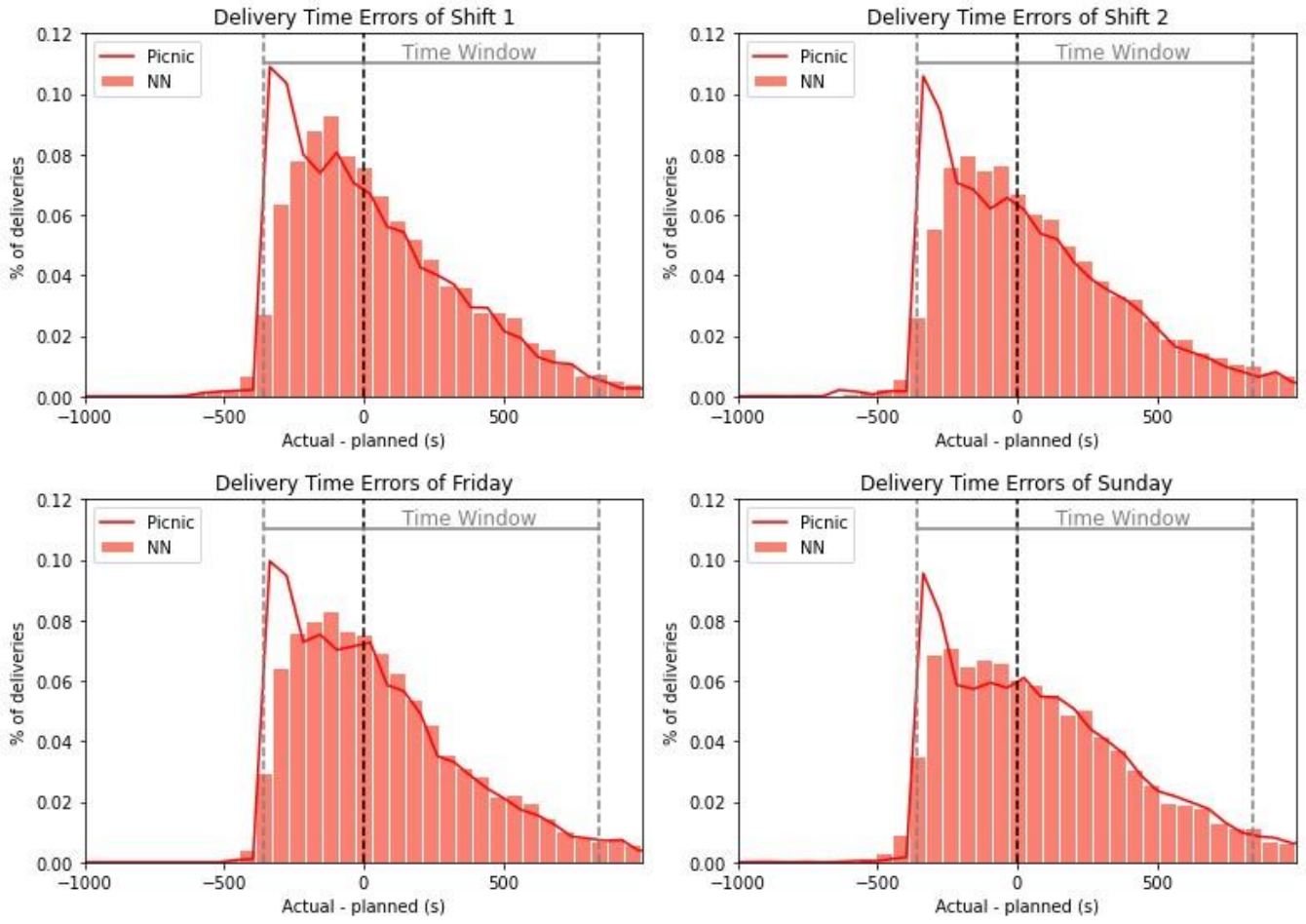

Figure 33. Delivery time error distributions of the peak shifts and days for the 9 hubs in week 16 and 17.

H. Insights across all hubs

In Section 5.4, we provided some granular insights (on shift, sequence and urbanity level) between the drive MAPE of individual drive segments, and eventual OT performance or median delays. In Figure 34, we provide some more relevant insights, where the x-axis also represents the median drive error rather than just the MAPE.

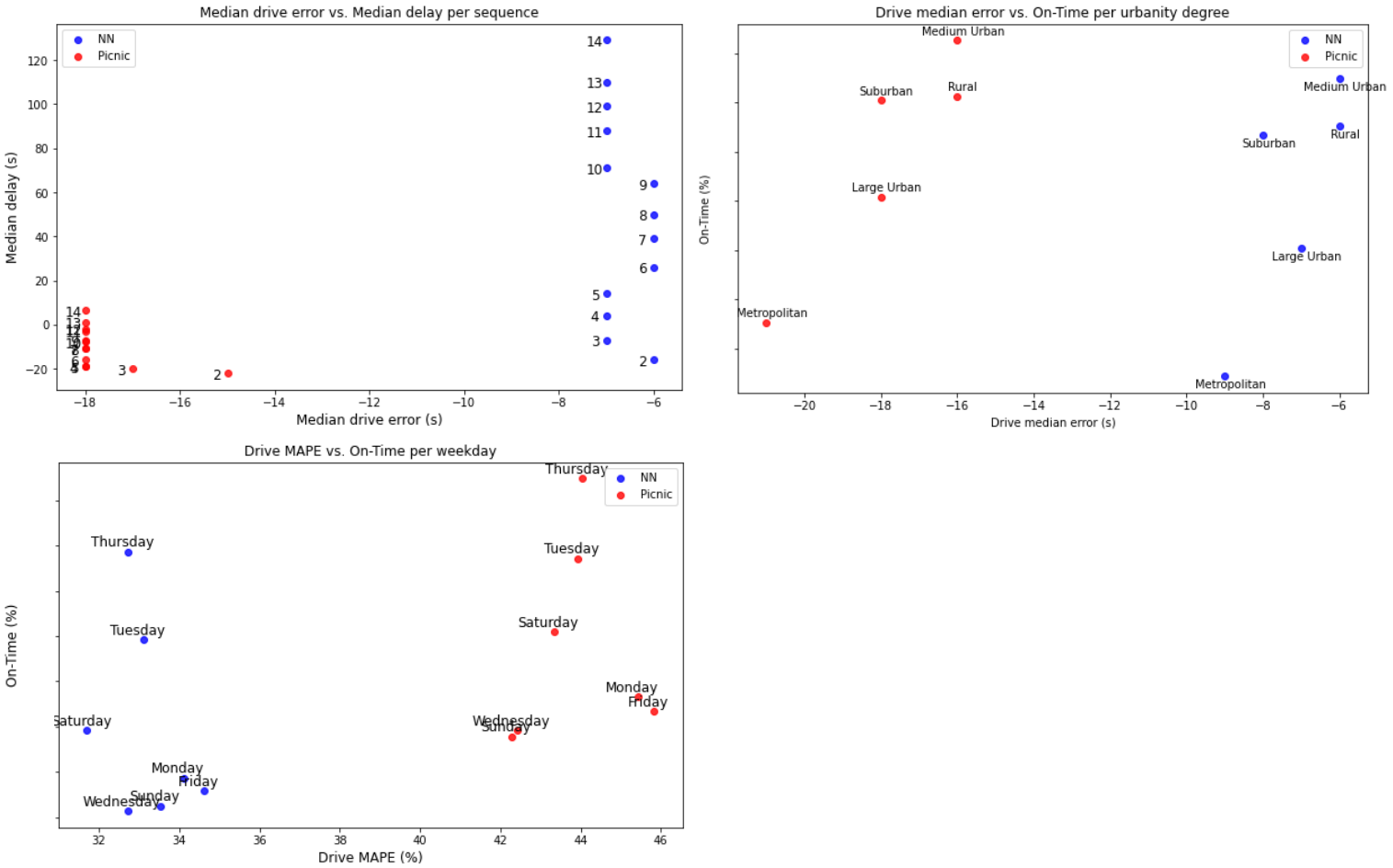

Figure 34. Granular insights on sequence, urbanity and weekday level. Across all hubs