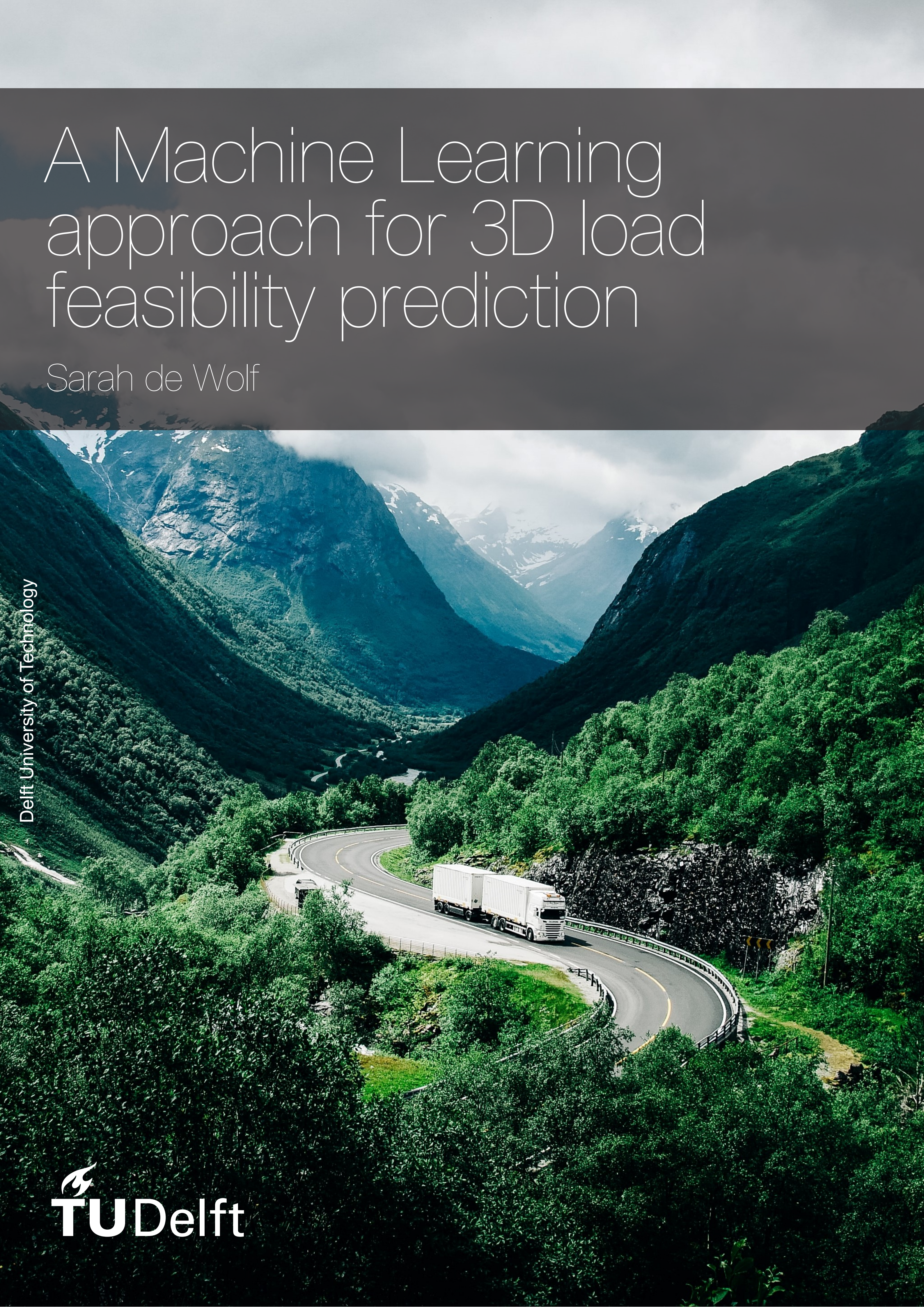
A Machine Learning approach for 3D load feasibility prediction

Sarah de Wolf



Delft University of Technology

TU Delft

A Machine Learning approach for 3D load feasibility prediction

by

Sarah de Wolf

Student number:	4364899	
Institution:	Delft University of Technology	
Faculty:	Faculty of EEMCS, Delft	
Company:	ORTEC, Zoetermeer	
Project Duration:	November, 2021 - June, 2022	
Thesis committee:	Dr. Neil Yorke-Smith	Thesis Advisor, TU Delft
	Dr. Rihan Hai	Examiner, TU Delft
	Ruggiero Seccia	Daily Supervisor, ORTEC

Cover Image: Truck in Norway from unsplash.com

TU Delft

Preface

Before you lies my thesis ”A Machine Learning approach for 3D load feasibility prediction”, which marks the end of my master Computer Science at the faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) at the Delft University of Technology.

This thesis was made as part of an internship at ORTEC, a company that creates planning, scheduling, routing, and forecasting solutions.

First, I would like to thank my ORTEC supervisor Ruggiero Seccia and my TU Delft supervisor, Dr. Neil Yorke-Smith. Both have contributed to the completion of my thesis by providing me with valuable suggestions and critical comments. I would also like to thank everyone at ORTEC for taking me into the team and giving me a great internship experience.

Sarah de Wolf
Delft, June 2022

Abstract

Vehicle routing problems (VRP) and Container Loading Problems (CLP) have been studied for decades. However, the combination of the two deserves more attention than in the literature to date. When solving VRP problems, computed routes must be checked for feasibility. Among the feasibility checks to perform, we need to guarantee that the load plan is feasible, namely that all the assigned products fit inside the truck. This involves solving a CLP.

Since the check of load plan feasibility is performed frequently, a short computational time is important. Hence, the load plan feasibility check is usually performed using approximation methods. Having rapid and reliable load plan feasibility estimations is crucial to reduce computational times when solving the VRP problem. However, if these estimations are conservative, the obtained routes are inefficient routes; if the estimations are opportunistic, the resulting load plans can turn out to be infeasible.

This work explored to what extent supervised Machine Learning (ML) methods can be used to rapidly yet accurately classify whether load plans will be feasible or not. These predictions can then be exploited in VRP algorithms to improve efficiency and computation time.

Several ML methods are considered and benchmarked on synthetic data and real data from a major company in the beverage sector. Extended experiments in different settings are performed, to check the effectiveness of ML in providing reliable load plan estimations and to extract insights on how load plan characteristics affect load feasibility.

Results suggest the effectiveness of applying ML models, with Random Forest models reaching an accuracy above 91.5% on all different experiments considered. Also, compared to the current estimations used for load feasibility checking, Random Forest models decreases computation time with 54.9%.

Contents

Acronyms

3L-CVRP	Three-Dimensional Loading Capacitated Vehicle Routing Problem
CLP	Container Loading Problem
FN	False Negative
FP	False Positive
KNN	k-Nearest Neighbour
KPI	Key Performance Indicator
LOG	Logistic Regression
LR	Linear Regression
ML	Machine Learning
NB	Naive Bayes
NN	Neural Network
NPV	Negative Predictive Value
OLB	ORTEC Load Builder
RF	Random Forest
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VRP	Vehicle Routing Problem

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1. Motivation

The Vehicle Routing Problem (VRP) and Container Loading Problem (CLP) have been studied for decades. However, the combination of the two deserves more attention than in the literature to date (Bortfeldt and Homberger 2013; Fuellerer et al. 2009; Gendreau et al. 2008; Iori 2005; Iori et al. 2007; Zachariadis et al. 2009). When solving VRP problems, computed routes must be checked for feasibility. Among the feasibility checks to perform, we need to guarantee that the load plan is feasible, namely that all the assigned products fit inside the truck. This involves solving a CLP.

Since load plan feasibility is checked frequently, a short computational time is crucial. Hence, the load plan feasibility check is usually performed using approximation methods. Having rapid and reliable load plan feasibility estimations is crucial to reduce computational times when solving the VRP problem. However, if these estimations are conservative, the obtained routes are inefficient; if the estimations are opportunistic, the resulting load plans can be infeasible.

This work explores to what extent supervised Machine Learning (ML) methods can be used to rapidly yet accurately classify whether load plans will be feasible or not. The computation time of standard heuristic methods depend on the number of items in a load plan, where ML approaches are size-independent. Moreover, ML approaches do not require human-based tuning of heuristics. Given the data, ML methods extract patterns in an automated way, decreasing the chance of biased predictions. The ML predictions can then be exploited in VRP algorithms to improve efficiency and computation time.

1.2. Introduction to ORTEC

This thesis is written in collaboration with ORTEC. ORTEC is now one of the world's leading providers of mathematical optimization software and sophisticated analytics, with over 1,000 people and offices in 13 countries worldwide. ORTEC creates planning, scheduling, routing, and forecasting solutions.

The research focuses on one client of ORTEC, a significant company in the beverage sector. This thesis uses data and loading rules of this beverage company data. For confidentiality reasons, the name of this company will not be mentioned.

1.3. Thesis goal and scope

This thesis aims to contribute to the improvement of the feasibility check of 3D load feasibility within route creation. The primary research goal is to examine if and how a ML solution could lead to improved productivity in route creation by improving the speed and/or accuracy of the load feasibility check. No literature is available on using ML methods for feasibility checking within route creation. The main research objective is to study the performance of different ML models and compare their capability to improve on the current solution.

This thesis focuses on a specific use case of a major beverage company. The most important Key Performance Indicator (KPI) is the precision of the model. It is not wished that the model predicts a load plan to be feasible when it is not. However, the overall performance of the model is also essential.

1.4. Research Questions

The primary research goal is to examine if and how a ML solution could lead to increased productivity of route creation. The research questions this thesis aims to answer are the following:

1. How accurately an ML approach can predict whether a 3D load plan will be feasible or not?
2. What are the improvements derived by a ML model for load feasibility checking versus current approximation algorithms?
3. What insights on the relation between input variables and 3D load feasibility can be extracted from the trained ML model?

1.5. Thesis Structure

This thesis is divided into chapters highlighting the research and development conducted throughout the thesis and the outcomes. The next chapter (Chapter 2) presents the relevant background for this thesis. This chapter provides background knowledge on cutting and packing problems, CLP, VRP, and their integration. At the end of the chapter, the research gap this thesis aims to fill is identified. Chapter 3 discusses the data used for the Machine Learning models. First, the data processing pipeline is explained. Then the chapter describes the datasets used for the final experiments. Chapter 4 presents the different Machine Learning models created for this thesis. The selection of the models is clarified. This chapter also goes over feature engineering and hyperparameter tuning. The approximation algorithm created for comparison with the ML performance is introduced in chapter 5. In chapter 6, the result of the models created in chapter 4 are explained, as well as the comparison with the approximation algorithm of chapter 5. Also, insights on the relation between the input features and 3D load feasibility are presented. The final chapter of this thesis concludes with answers to the research questions and the recommended future work.

2 Background and problem statement

This chapter presents the theoretical background needed to understand the thesis's methodology. First, an overview of cutting and packing problems is given, and the specific cutting and packing problem this thesis deals with is identified, namely the Container Loading Problem (CLP). The following section goes into depth on the CLP and presents the existing literature on the problem. Section 2.3 describes the Vehicle Routing Problem (VRP) and how CLP and VRP are integrated in both literature and reality. The last section identifies the research gap this thesis aims to fill.

2.1. Cutting and packing problems

Dyckhoff (1990) grouped the types of cutting and packing problems based on four main criteria, namely dimensionality, kind of assignment, the assortment of large objects and the assortment of small items. Wäscher et al. (2007) grouped the cutting and packing problem into five main categories. In figure 2.1 an overview of the taxonomy of Wäscher et al. (2007) is given.

Dimensionality
The meaning of dimensionality speaks for itself. The dimensionalities that are relevant for this research are 2D and 3D.

Kind of assignment
There are two different objectives/assignments a cutting and packing problem can have. The first is that all large objects can be used, and a selection of small items can be used. This is called output maximisation. The amount of container space is set in output maximisation, and the goal is to pack as many items as possible. The second cutting and packing objective is input minimisation. With input minimisation, a selection of objects is used to pack all items. So the amount of items that need to be packed is fixed, and the goal is to use the least amount of (container) space.

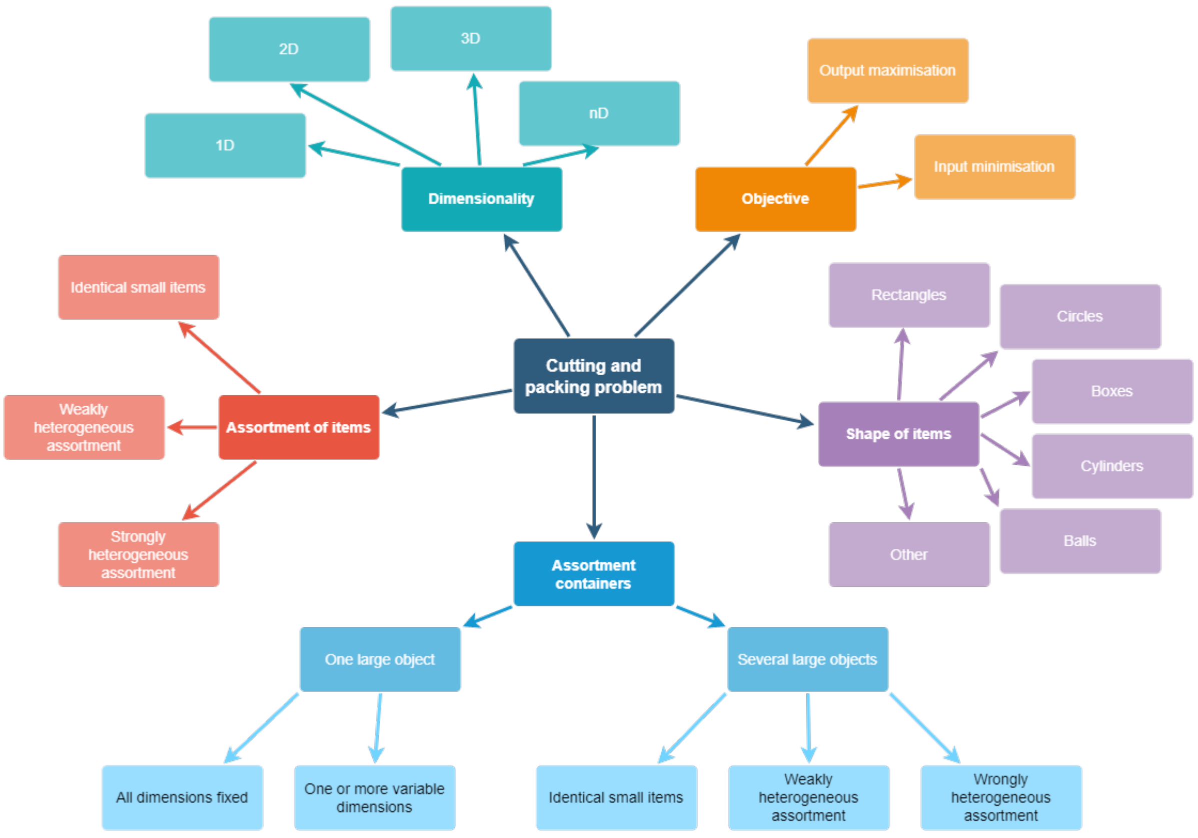


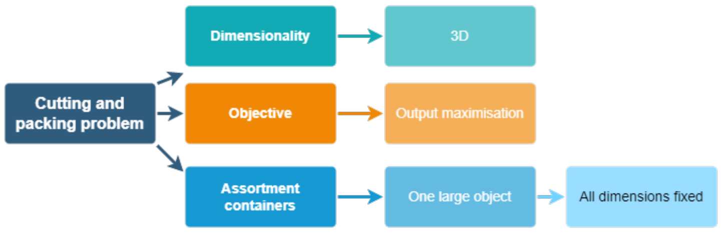
Figure 2.1: Overview of the criteria for the definition of cutting and packing problem types

Assortment of large objects
There are two different types of 'assortments of large objects', either 'one object' or 'several large objects'. In the case of one large object, the object can have either fixed dimensions or one or more variable dimensions. If there are several large objects, these can either be identical, weakly heterogeneous or strongly heterogeneous.

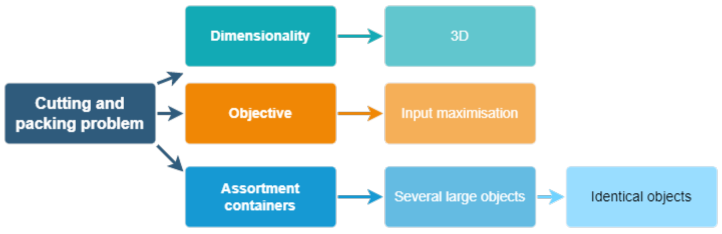
Assortment of small items
Concerning the assortment of small items, there are three different types. Namely 'identical', 'weakly heterogeneous' and strongly heterogeneous'. Identical items are the same in the problem-relevant dimensions, such as shape and size, but weight and required orientation can also play a role. Items are weakly heterogeneous if they can be grouped based on problem-relevant dimensions in relatively few classes. Items are strongly heterogeneous if there are only a few identical items.

2.2. Container Loading Problem

Dyckhoff (1990) found that the container loading problem can be defined in two different ways as a cutting and packing problem. In both cases, the problem is three-dimensional. The container loading problem can be defined as having the objective of output maximisation with only one large object with fixed dimensions. Alternatively, the problem can be defined as an input minimisation problem with several identical large objects. Figure 2.2 presents a schematic overview of the two different types of container loading problems as cutting and packing problems.



(a) Option 1



(b) Option 2

Figure 2.2: A schematic overview of the two possible criteria of a container loading problem as a cutting and packing problem

2.2.1. Constraints

The container loading problem is a geometric assignment problem where 3D items must be packed into 3D cubic large objects (containers). Three constraints are the same for every container loading problem. These constraints are:

- Each item is placed completely within the container
- Each item is not allowed to overlap with other items
- Each item is placed parallel with the walls of the container.

Aydemir and Yigit (2019) outlined the different constraints that can be accounted for in the Container Loading Problem. These can be split into Container-Related, Item-Related, Cargo-Related, Positioning and Load-Related constraints.

Container-Related Constraints

Weight Distribution Constraints: The weight should be spread across the container floor. Heavier items should be lower than lighter ones, reducing risks during transport.
Weight Limits: Take into account the weight limit a container can take.

Item-Related Constraints

Loading Priorities: This constraint can only occur in the case of input minimisation. Items with a higher contribution to the objective function have a higher loading priority. If all items need to be loaded, there could be constraints on (1) delivery locations, (2) items that are not allowed to be loaded together, or (3) obliged to be loaded together.
Orientation Constraints: This constraint is one of the most commonly used constraints. It depicts whether items are allowed to rotate and, if so, in what directions.
Stacking Constraints: Restricts how to stack items. Factors that influence this constraint are how much pressure items can handle or how many items can be placed upon another item

Cargo-Related Constraints

Complete-Shipment Constraints: This constraint can only occur in the case of output maximisation. Items that are part of a subset should be loaded as a group or not.
Allocation Constraints: This constraint should only occur in multiple container problems. Some items are not allowed to be packed in the same container, e.g. food and cleaning products.

Positioning Constraint

This constraint can occur when the items are for different customers. The items should be loaded in a specific route order so the unloading will not take too long.

Load-Related Constraints

Stability Constraints: This constraint ensures the items are loaded stable to avoid damage to the items, containers and employees handling the load.
Complexity Constraints: This constraint prevents the load plan from being too complex. Since this could be too difficult for manual loading and take too much time.

2.2.2. Approaches for Container Loading Problem

Bortfeldt and Wäscher (2013) states that heuristic algorithms are the most important class of algorithms for solving container loading problems. Since only heuristic algorithms can provide solutions within a reasonable computing time and volume utilisation if the problem size grows. Especially if different constraints are taken into account.

Pisinger (2002) stated that the most common heuristics use either the wall building, stack building, horizontal layer building, block building or guillotine cutting for placing items, in Pisinger's case boxes, in the Container Loading Problem.

1. **Wall Building Approach**, proposed by George and Robinson (1980), splits the

container into vertical layers, and the container is loaded per layer (wall) simultaneously.

2. **Stack Building Approach**, proposed by Gilmore and Gomory (1965), packs the items in stacks arranged on the container floor. Which reduces the problem to two dimensions.
3. **Horizontal Layer Building Approach**, proposed by, loads the container from bottom to top by loading horizontal layers simultaneously. Each horizontal layer aims to cover the maximum space of the layer.
4. **Block Building Approach**, proposed by Bortfeldt and Gehring (1998a), packs the container recursively with cuboid arrangements of similar items.
5. **Guillotine Cutting Approach**, proposed by Morabito and Arenales (1994),is based on a slicing tree. The slicing tree represents a 'guillotine' partition of the container, and the leafs represent the items to be packed.

The heuristic algorithms that can be applied to the Container Loading Problem can be split into three categories: Conventional Heuristic Methods, Meta-Heuristic Methods and Tree Search Methods (Fanslau and Bortfeldt 2010).

Conventional Heuristic Methods
Bischoff, Janetz, et al. (1995) presented a heuristic approach where the loading arrangement is built up in layers from the bottom upwards. Bischoff and Ratcliff (1995) proposed a method specifically designed to produce stable, evenly distributed loading arrangements. Lim et al. (2003) created a method using a multi-faced buildup technique in the packing procedure, so items are not required to form flat layers. These methods have been the most used method types for the 3D Container Loading Problem in recent years (Sheng et al. 2016).

Meta-Heuristic Methods
Hemminki (1994), Gehring et al. (1997), Gehring and Bortfeldt (2002) and Bortfeldt and Gehring (2001) proposed genetic algorithms. Sixt (1996) and Mack et al. (2004) presented simulated annealing methods. Sixt (1996), Bortfeldt and Gehring (1998b) and Bortfeldt, Gehring, and Mack (2003) presented tabu search algorithms. Faroe et al. (2003) and Mack et al. (2004) developed local search algorithms. Moura and Oliveira (2005) and Parreño et al. (2008) suggested a greedy randomized adaptive search procedure (GRASP). Zhou and X. Liu (2017) introduced a swarm search algorithm.

Tree Search Methods
Morabito and Arenales (1994) introduced the AND/OR graph search method where possible loading patterns are represented as complete paths in an AND/OR-graph. Eley (2002) proposed an algorithm with a greedy heuristic that builds homogeneous blocks of identical items. These blocks are then loaded using tree search. Hifi (2002) presents a tree search algorithm and uses hill-climbing strategies to construct heuristics. Pisinger (2002) proposed an algorithm based on the wall-building approach. These 'walls' are then split into several horizontal or vertical strips. The size of the

layers and strips is decided through a branch-and-bound approach. These strips are then solved as a one-dimensional knapsack problem. Fanslau and Bortfeldt (2010) proposed two different methods. The first is a full-support method where full support from below for all packed boxes is guaranteed. The second is the non-support variant, where this is not considered. Partition-controlled tree search with the block-building approach is proposed. Zhang et al. (2012) created a heuristic block-loading algorithm based on multi-layer search based on depth-first search. S. Liu et al. (2014) presents a binary search tree algorithm based on the wall-building approach. Araya and Riff (2014) introduced a beam search approach, which can be seen as a variant of the branch-and-bound search that only expands the most promising nodes at each level of the search tree.

2.3. Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is the problem of finding the optimal routes for a set of vehicles visiting a set of locations. The VRP is a minimisation problem, where the objective is to minimise the route's costs. The VRP problem is NP-hard. So does not exist a solution that can optimally solve the VRP in a reasonable amount of time. Therefore, heuristics are used to give a non-optimal but good solution in a reasonable amount of time.

2.3.1. Integration CLP in VRP

Both CLP and VRP are NP-hard problems. Evidently, the combination of the two, known as the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP), is also NP-hard. Iori (2005) proposed the 2D version of the problem and solved it with an exact approach in 2007 (Iori et al. 2007). Gendreau et al. (2008) presented an approach using a tabu search algorithm for the routing part of the problem and tackling the loading as a two-dimensional strip packing problem. Zachariadis et al. (2009) introduced an algorithm called Guided Tabu Search. In terms of routing, it uses a search method based on tabu search, driven by an objective function alteration mechanism, to explore the solution space. In the approach proposed by Fuellerer et al. (2009), for the loading component, multiple heuristics are used, and for the overall optimisation, an ant colony optimisation (ACO) method is used. Bortfeldt and Homberger (2013) proposed a method where the loading is solved prior to the routing, in contract with the beforementioned approaches.

2.3.2. 3D load feasibility estimation

The methods mentioned in subsection 2.3.1 require a relatively adequate time to solve if the loading is not infeasible often. Nonetheless, these methods do not consider all the different aspects of a route plan that must be feasible in real life. The loading capacity is just one aspect of the problem. In reality, when creating routes, other feasibility checks need to be performed, such as time windows, picking time, and maximum pollution or costs. Generally speaking, creating a route plan that satisfies all these constraints and is close to an optimal solution is an iterative process. Consequently, these routes need to be checked for feasibility repeatedly. So the computational time for the load plan feasibility check must be short. Therefore, the load plan feasibility check is usually performed using approximation methods. Rapid and reliable load plan feasibility estimations are crucial to reducing computational times when solving the VRP problem. However, if these estimations are conservative, the obtained routes are inefficient; if the estimations are opportunistic, the resulting load plans can be infeasible. No literature was found on these approximation algorithms for feasibility checking during route creation.

2.4. Research gap

Talking to experts in the routing industry revealed that companies use very basic approximation algorithms to predict feasibility during route creation. In Chapter 5 an approximation algorithm based on the one ORTEC is using is presented. According to the lacking literature and the industry experts, a ML approach for 3D load feasibility checking during route creation has not been explored. This thesis aims to fill the research gap on how ML approaches may be used to quickly and reliably determine if load plans are feasible. Approximation methods have a computation time proportional to the number of items in a load plan, whereas ML approaches are size-independent.

3 Data

This thesis intends to bridge a knowledge gap in how ML methods may be used to identify whether load plans are feasible or not. In order to create an ML model, data is required. This thesis focuses on the beverage industry. Information and data from a major company in the beverage sector are utilised.

At first, a minimal dataset was available from the focus beverage company. The dataset consisted of 242 load plans, including 12 truck types, 4 pallet types and 118 unique item types. This dataset was insufficient to train an ML model. A second issue was that this dataset consisted of unusual cases. In Figure 3.1 an example is shown. The image on the left shows an almost empty truck. After talking to an industry expert, we discovered that this does happen in exceptional cases. Nearly empty trucks like that will drive if there is no efficient way to combine the load with other trucks, and the customer's orders need to be done that day. For these two reasons, synthetic data had to be created.

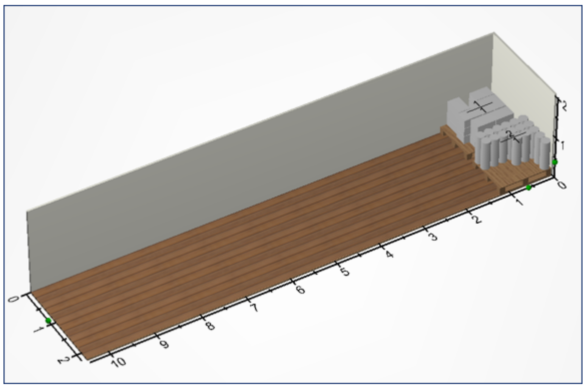


Figure 3.1: Visualisation of a random example of the initial company data

This chapter first explains the data processing pipeline. Second, a general overview of the pipeline is given. After which, every step of the process is explained in more detail. The last section of this chapter presents the datasets used for the experiments performed in this thesis.

3.1. Data Processing Pipeline

In Figure 3.2 one can see a flowchart of the data processing pipeline. The operation is part of the synthetic data processing pipeline if a square is blue. The procedure is part of the company data processing pipeline if a square is orange. If a square is half blue and half orange, it is part of both the synthetic and the company data processing pipeline.

Load plan input has to be created for the synthetic data processing pipeline. This input consists of information on what orders must be placed in the truck. This information is then passed on to the ORTEC Load Builder (OLB). The input file is read, and with the use of the ORTEC loading algorithm, the orders are placed in the truck(s). From this point on, the data processing pipeline is the same for the synthetic and the company data. OLB outputs where every item and pallet should be placed in the truck. Features are extracted from the OLB output and transformed to a row in the tabular ML input dataset.

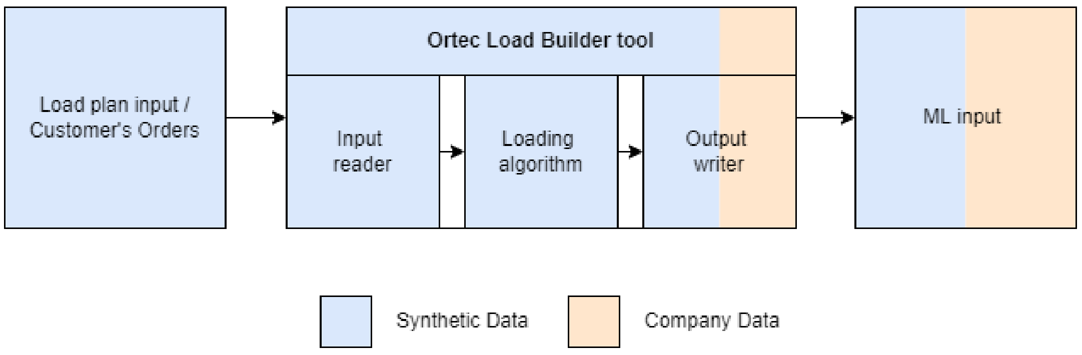


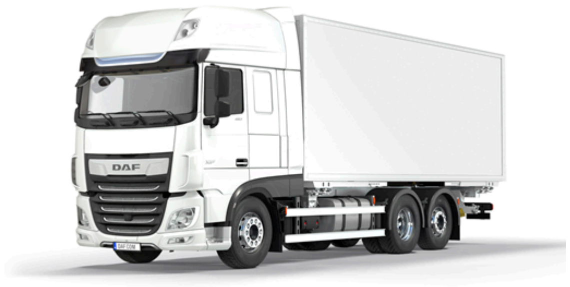
Figure 3.2: Data processing pipeline

3.1.1. Load Plan Input

Trucks, pallets, and items are needed to create synthetic load plan input. These trucks, pallets, and items are obtained from the small initial company dataset.

Trucks
The features considered for trucks in this research are the truck type, length, width, height, tare weight and the allowed total weight. A truck type can either be a Standard Truck or a Bay Truck.

Pallets
For pallets, we consider the features length, width and height of the pallet, its tare


(a) Standard Truck

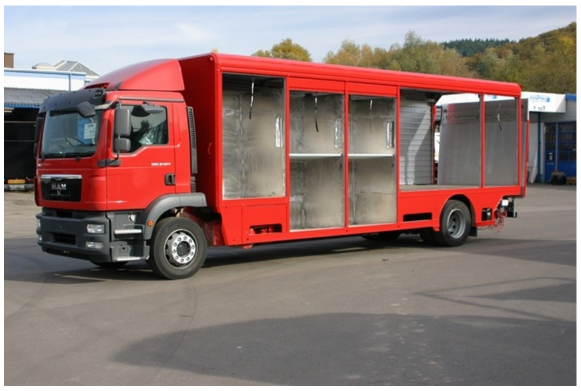

(b) Bay Truck

Figure 3.3: Examples of the two different truck categories included in the data

weight and allowed total weight, and the maximum allowed height of the items on the pallet.

Items
There are different features we look at when we consider items. First is the shape of the item; with generalisation, we consider every item either cuboid or cylinder-shaped. Other than the weight and dimensions of the items, we also consider the rotatability and stackability of the items. Some items have specific stacking rules. For example, a crate of beer cannot be stacked on a bag of potato chips. An example of rotatability constraints is that a bottle should not be placed on its cap.

After extracting the trucks, pallets and items, these could be used to create synthetic load plans and thus an experiment dataset. Figure 8 presents a scheme of how the synthetic load plans are created.

For every experiment, a specification CSV is given as input. This CSV provides information such as the customer types, number of truck, pallet and item types and loading rules. Based on this CSV, a subset of trucks, pallets, items and customer types is created. A customer type is specified through different features. Namely, its item type preference is either small, big or no preference.

Load plans are created using a lower- and upper-bound of volume utilisation. First, a load plan is created with a volume utilisation equal to the lower bound. This load plan is then exported, after which a new customer with its items is added to the same load plan. Again, this load plan is exported. This iterative process is repeated until the volume utilisation equals the upper bound.

The aim is to create synthetic datasets that are balanced. This was achieved by trying smaller test datasets with different lower and upper bounds. These tests were performed separately for the Standard Truck and the Bay Truck since they have different manners of loading. These test sets exposed the critical boundary where a load plan becomes infeasible to fit into one truck. The load plans close to this boundary are the most challenging load plans to predict. The datasets are balanced around this critical boundary.

3.1.2. ORTEC Load Builder

OLB uses operational research (OR) algorithms to simulate and optimise how to pack items into containers or onto pallets in the most efficient way. The result is exported as a configurable list and interactive 3D graphic. In this thesis, OLB 's output is used as a ground truth infeasibility for the data generation. OLB can perform a cartonizing, palletising and loading algorithm on the load plans. Different loading rules can be applied to these algorithms. Finally, OLB outputs XML files with the packing instructions.

A few of the most significant loading rules are loading sequence, pallet margins, multi-customer pick pallets, stackability of pallets and the height of items on a pallet.

Loading sequence
Pallets are loaded and unloaded from the back in the Standard Truck type. Therefore the location in the truck where pallets are placed is essential. The first customer should not be placed in the back of the truck. Bay Trucks are loaded and unloaded from the side, so this situation generally does not occur with Bay Trucks.

Pallet margins
Some companies have rules on how much space should be between pallets or rows of pallets. The beverage company considered in this thesis demands at least a 10cm spacing between the truck's left and right row of pallets.

Multi-customer pick pallets
A typical process in the beverage industry is that a customer's items are mixed through the truck on different pallets. If multiple small orders were all located on individual pallets, the volume of the truck would be utilised very poorly. A remedy is to combine orders from different customers on the same pallet. Furthermore, the picking process in the warehouse and loading of the trucks could be optimised by picking per item type of isle instead of the customer. So, if a customer allows, so-called multi-customer pick pallets are built. When arriving at the customer, the driver picks the items from the multi-customer pick pallet. In general, Bay Trucks are used for multi-customer pick pallets.

Stackability of pallets
If the items on a pallet are strong enough, it could be possible to stack pallets on top of each other. Some companies do not allow for this pallet stacking. Pallets can be dirty and have splinters and therefore could damage the products they are stacked on.

Height of items on a pallet
The rules on the maximum allowed height on the pallets are location-based. Generalising, the people in Germany are taller than the people in Japan. Therefore Germany will allow for higher item stacks on pallets than Japan. Otherwise, the unloading will be uncomfortable.

3.1.3. Machine Learning Input

Finally, the XML file outputted by OLB needs to be translated to rows of features. Section 4.2 will go into depth on what features are included in the ML models.

3.2. Experiments

This section will give insights into the three datasets used for the experiments. There are two synthetic datasets and one beverage company dataset. Table **??** gives insights into the different datasets.

3.2.1. Experiment Synthetic A

The first dataset is the simplest of all. This dataset consists of 12,625 synthetic instances, of which 45% is feasible. The dataset is built from 6 trucks, 4 pallets, 50 items and 3 customer types. The six different trucks are solely Standard Trucks. The three customer types include one that prefers a small number of big items, one that prefers a large number of small items and one that prefers an average amount of medium-sized items. In this dataset, it is allowed to have precisely one customer per pallet. The loading sequence is not taken into account.

3.2.2. Experiment Synthetic B

The second synthetic dataset consists of 21,934 instances, of which 51% is feasible. This dataset is made from 11 trucks, 6 pallets, 100 items and 5 customer types. The trucks include both Standard Trucks and Bay Trucks. The five customer types include the same types as Experiment Synthetic A. Also, two customers with dominant big orders of one item type are added. One customer's order consists of small items, the other of big items. Like the other synthetic dataset, it is allowed to have precisely one customer per pallet. However, the loading sequence is taken into account.

3.2.3. Company Experiment

The dataset of the Company experiment consists of 143,279 instances, of which 91% is feasible. It includes both the Standard Truck and the Bay Truck. It consists of primarily two special scenarios: Most of the time, the load plans employing a Standard Truck have just one customer and at most three different items. According to experts on the beverage company in question, this is either haulage or a cross-dock shuttle scenario. Haulage means the replenishment of distribution centres. A cross-dock shuttle scenario means that a bulk truck brings the goods to the cross-dock, where they are unloaded and loaded on distribution trucks. The Bay Trucks have the customer's items mixed over different pallets on so-called multi-customer pick pallets. In section 3.1.2 the motivation for these pallets is explained in more detail.

3.2.4. Summary

In summary, both synthetic datasets are balanced and controlled. It is known what truck, pallet and item types are included, and they represent different scenarios through the different customer types. Dataset Synthetic A only includes the loading rule of one customer per pallet. In the second synthetic dataset, the loading sequence is taken into account. The company dataset includes all the beverage company's loading rules, which will not be mentioned due to confidentiality reasons. The company dataset is imbalanced and includes particular scenarios that are not generalisable.

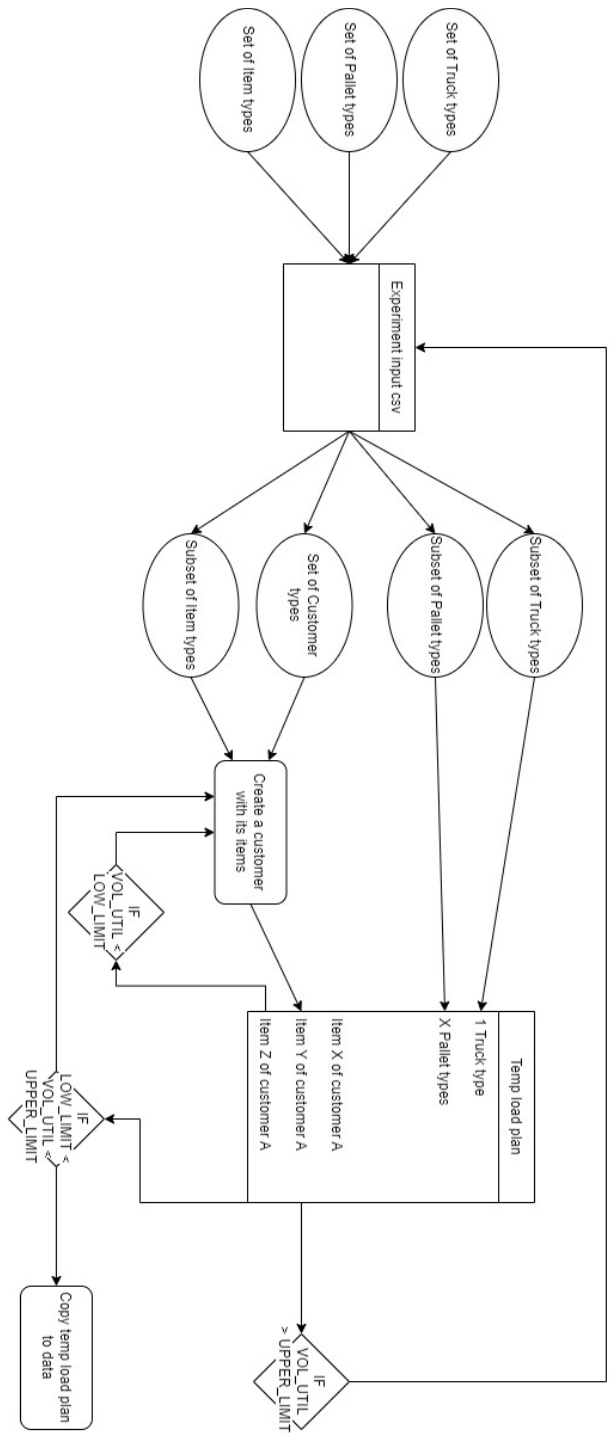


Figure 3.4: Load plan creation design

4 Machine Learning Models

This chapter presents the different ML methods that were explored. The first section elaborates on the different ML models that will be investigated. The following section demonstrates the feature engineering methods used. The last section elaborates on the strategy used for hyperparameter tuning.

4.1. Model Selection

In ML, there are two main approaches: supervised learning and unsupervised learning. With unsupervised learning, the model identifies patterns and trends in unlabeled data, for example, by clustering the data into groups. An unsupervised learning algorithm aims to find insights and trends in the data. A drawback of unsupervised learning is that the results can be inaccurate if not checked by humans and that it is challenging to make the outcomes explainable. Supervised learning is an ML approach that uses labelled datasets for training the model and finds a pattern between the input and output data. The goal of supervised learning algorithms is to predict outcomes for the new input data. A drawback of supervised learning is that the training can be very time-consuming. Experts sometimes must label the data manually, which is very time-consuming.

This thesis considers supervised learning methods to classify load plans (input) as feasible or infeasible (output). A selection was made from a list of IBM (2021) containing the most commonly used supervised learning algorithms: Neural Network (NN), Naive Bayes (NB), Linear Regression (LR), Logistic Regression (LOG), Support Vector Machine (SVM), k-Nearest Neighbour (KNN) and Random Forest (RF). LR is excluded because it is not a classification algorithm. NN is not considered because it is a bit too complicated for the use case. Also, it is unsure if the datasets consist of enough instances for NN. NB is not considered because it assumes that all features are independent, which is not the case. Therefore this thesis assesses, LOG, KNN, SVM and

RF.

4.1.1. Logistic Regression

Logistic regression is a linear classification model that predicts binary outcomes. It predicts the probability that an event occurs. This probability is then mapped to a discrete outcome class using the Sigmoid function. LOG is an efficient simple model. However, LOG supposes a linear relationship between the input and output variables.

4.1.2. k-Nearest Neighbour

Simply put, KNN works as follows: for every new data instance, knn looks at the k-nearest neighbours. What are the k-nearest neighbours mainly depends on two things: the distance metric and k. Examples of distance metrics included in python's SKLearn are Eucleadian, Manhattan and Minkowski. Deciding what k to use is a difficult trade-off. If the k-value is too small, this could cause overfitting. If the k-value is too big, this could cause over-generalisation. KNN is a fast prediction model and can handle non-linear data.

4.1.3. Support Vector Machine

SVM creates hyperplanes to separate classes. This hyperplane is placed in such a way that the distance between the hyperplane and the nearest point on each side (margin) is maximised. SVM is not sensitive to overfitting.

4.1.4. Random Forest

RF is an ensemble of multiple decision trees. In short, RF models creates n decision trees. Every decision tree is based on a randomly sampled subset of the dataset. The prediction of the RF model is the class that was predicted most often within all n decision trees. RF models are powerful and can also handle non-linear data. However, it is sensitive to overfitting and difficult to interpret.

4.2. Feature Engineering

Feature engineering is an essential aspect in the creation of an ML model. Feature engineering intends to organise the information/data in a way that is compatible with ML algorithms and improve the model's performance. In this thesis, the information about the load plans needs to be decoded into relevant features, which is called feature extraction. Understanding of the discipline is required to extract influential features. Therefore this is a manual process. Then these features need to be reduced. This process is called feature selection. Feature selection will reduce the feature calculation time and the training time. It also decreases overfitting en the model will be easier to interpret.

This thesis aims to create a model that will fast and precisely predict whether a load is feasible or not. So, the time to compute the features needs to be short, while the precision is high. It is impossible to optimise both, so a balance between the two has to be found. Where this balance is, depends on the use case, but even then, it is not a trivial decision. A minimum accuracy or precision can be set in stone. However, from there on, it is a trade-off between adding another feature that improves performance and increases computation time. Therefore, this process needs to be done manually.

The feature engineering was performed on Experiment Synthetic B (section 3.2.2). It was performed on this experiment because it is a more complicated and realistic experiment than Experiment Synthetic A. Moreover, in contrast with the Company Experiment, the dataset was balanced, and it consisted of more generalisable instances.

4.2.1. Feature Selection

Feature selection is the process of selecting essential input features for an ML model. Excluding a feature reduces the computation time of the input data for the ML model, but it could also decrease the performance of the model.

Table 4.1 gives an overview of the 40 initial features that were taken into account for the feature selection. A short computation time is one of the most important KPI's in this thesis. The feature selection performed in this thesis is based on the theory that including highly correlated features will increase the computation time but will not increase precision (much).

Algorithm 1 shows the pseudocode for the feature selection method this thesis uses, which will hereafter be explained. The feature selection method was performed on both the RF and KNN model. First, the 40 features were sorted on feature importance for the model prediction. Then, the Pearson correlation between all 40 features was calculated. Figure 4.1 presents the correlation heatmap between all these 40 features. For readability, correlations with itself were set to None, and only correlation values above 0.7 are presented. Also, features were excluded from the heatmap if they did not have a Pearson correlation above 0.7 with any other feature.

The next step is to exclude highly correlated features. Different correlation bounds have been tested, namely 0.7, 0.75, 0.8 and 0.85. Features from the residual feature set were added incrementally from three up to 20 features, while the model's performance was tested every step.

The calculation time per feature was measured on a subset of 1678 instances of Ex-

Table 4.1: All features included in the feature selection procedure with the median time it takes to calculate 1678 randomly selected instances of Experiment Synthetic A

Feature	Time (s)	Feature	Time (s)
Truck type	0.01	Total item height	0.33
Truck volume	0.00	Volume of biggest customer	32.84
Truck length	0.00	Weight of biggest customer	8.68
Truck width	0.00	Volume of last customer	3.09
Truck height	0.00	Weight of last customer	0.87
Max allowed weight	0.01	Volume utilization	0.01
Pallet EURO	0.00	Weight utilization	0.00
Pallet 48x40	0.00	Number of unique items	0.25
Pallet 36x36	0.00	Number of cylinder items	0.00
Pallet plastic 1x1.2	0.00	Number of cuboid items	0.00
Pallet plastic 1x1	0.00	Total cylinder weight	0.00
Pallet prostack	0.00	Total cuboid weight	0.00
Pallet NoDescr	0.00	Total cylinder volume	0.00
Unknown pallet type	0.00	Total cuboid volume	0.00
Total number of items	0.17	Number items allowed to be placed LxW	0.00
Total number of customers	0.23	Number items allowed to be placed WxL	0.00
Total item weight	0.50	Number items allowed to be placed LxH	0.00
Total item volume	2.90	Number items allowed to be placed HxL	0.00
Total item length	0.32	Number items allowed to be placed WxH	0.00
Total item width	2.85	Number items allowed to be placed HxW	0.00

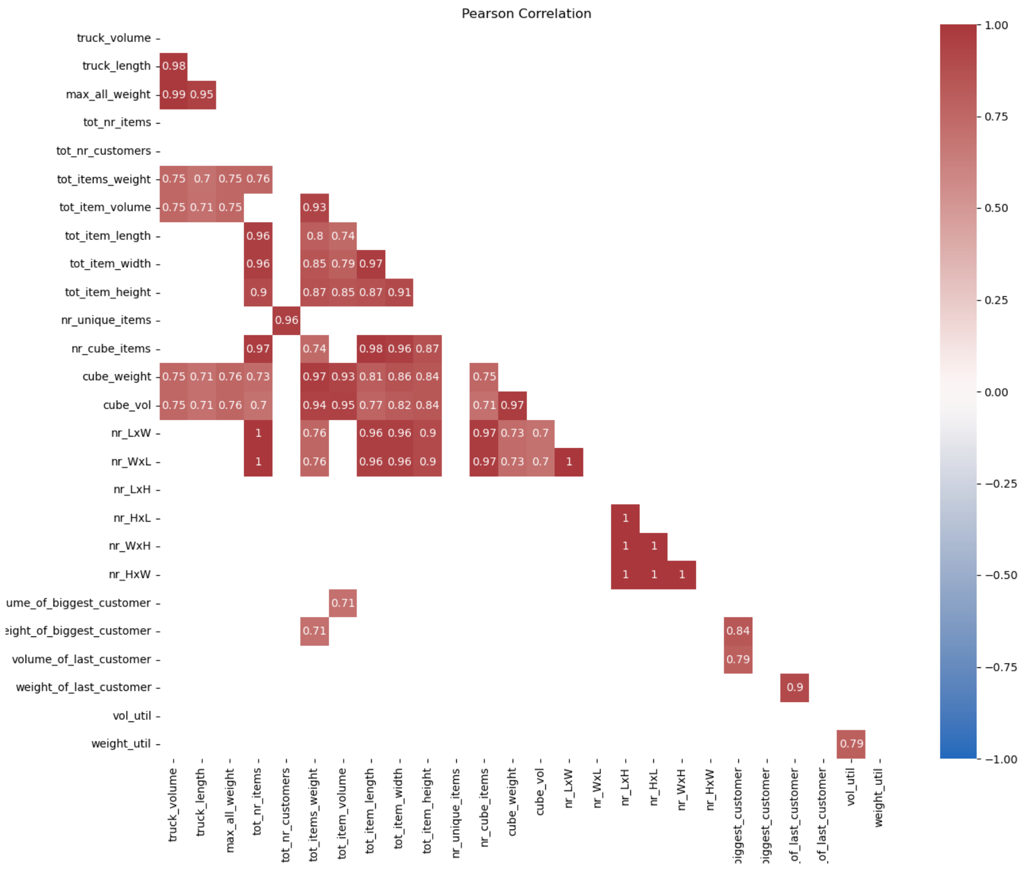


Figure 4.1: Pearson correlation heatmap for the features before feature selection. Only correlations > 0.7 are displayed.

periment Synthetic A. These load plans were converted to ML input features. The time it took to calculate a feature was summed for all 1678 load plans per feature. This process was repeated 100 times. Most features took so little time to create that it was impossible to measure. In Table 4.1 the median time it took to create the features for these 1678 load plans is presented.

Table C.1 shows the results of this method on the RF model on Experiment Synthetic B. Figure 4.2 presents a plot of these results with on the x-axis precision, and on the y-axis the time it takes to calculate the accompanying features. The plot shows a clear Pareto-front of 5 data points. Table 4.2 presents these 5 Pareto front feature sets and their performances. As said before, the trade-off between performance and time is very use case dependent. For this thesis, feature set 2 is chosen because this feature set showed the best overall performance compared to the other 4 in the Pareto front. The precision of feature set 1 was higher than that of feature set 2. However, feature set 2 had higher accuracy and recall and a lower computation time. Feature set 5 shows that very few simple computable features can perform well. Concluding, the final selected features for the ML models created in this thesis are:

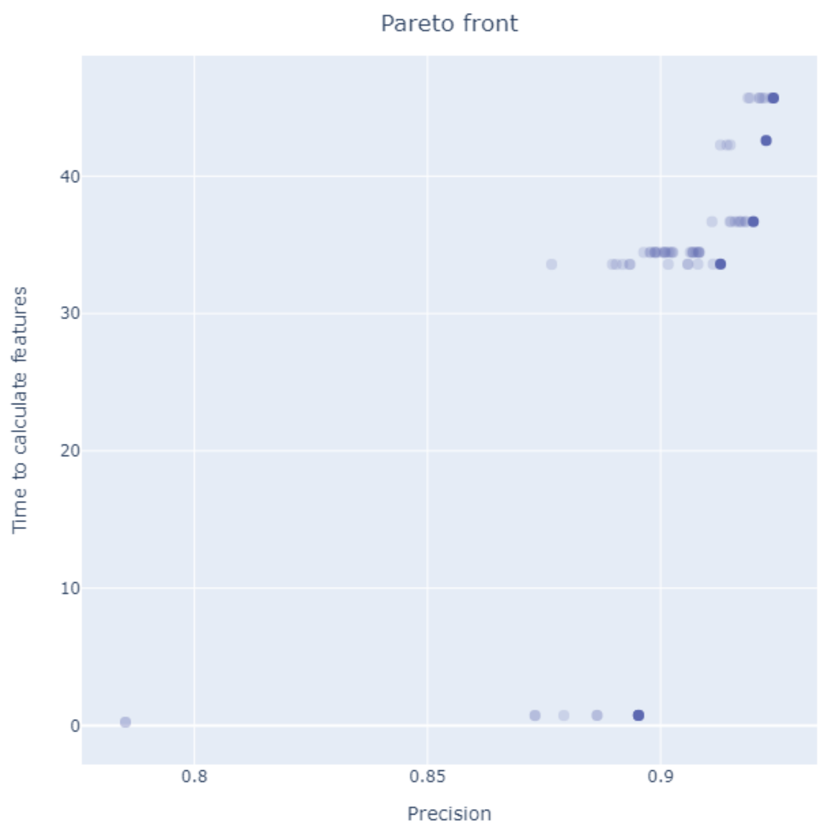


Figure 4.2: The precision for the different feature combination vs the time to calculate these features

1. Weight utilization
2. Volume utilization
3. Total number of customers
4. Total weight of the items
5. Truck Volume
6. Volume of the biggest customer
7. Weight of the biggest customer
8. Truck Width
9. Truck Type
10. Total Item Length

4.3. Hyperparameter Tuning

Hyperparameters are configuration arguments in a ML model that allows the model to be customised for a specific dataset. In contrast with model parameters, which are set through training, hyperparameters are set manually before training. ML models have different hyperparameters that also interact with each other. Therefore, a combination of hyperparameters should be found to achieve optimal model performance. The process of finding this set is called hyperparameter tuning.

This thesis uses a combination of Grid Search Cross-Validation and Randomised Search Cross-Validation.

With Grid Search, first, it is manually specified which ranges of hyperparameters are included. Then, all hyperparameter combinations are tried out with cross-validation, and their performance is measured. The best hyperparameter set is the combination that maximises the average value in cross-validation. Grid Search is a time-consuming technique since it considers all possible combinations, and for every combination, k-fold cross-validation is employed.

Randomised Search Cross-Validation is comparable to Grid Search, but instead of trying out every hyperparameter combination, it tests a randomly selected subset of all combinations.

This thesis first uses Randomised Search Cross-Validation, with accuracy as the performance measure. Even though precision is the most critical KPI, high accuracy is also essential. A model with very high precision could be a flawed model in general. The Randomised Search Cross-Validation gives an idea in which direction to look for models that perform well in general, so a high accuracy.

Then, a grid is built around the Randomised Search outcome. This grid is used as input for the Grid Search Cross-Validation. The Grid Search Cross-Validation will use

Table 4.2: Overview of the feature sets in the Pareto front and their performance

	Features	Feature Time	Precision	Accuracy	Recall	CorrBound	NrFeat
1	weight utilisation, volume utilisation, total number of customers, total item weight, truck volume, volume of biggest customer, weight of biggest customer, truck width, truck type, total item length, volume of last customer, number of cylinder items, total cylinder weight, truck height	45.6893	0.9242	0.9143	0.9072	0.85	14
2	weight utilisation, volume utilisation, total number of customers, total item weight, truck volume, volume of biggest customer, weight of biggest customer, truck width, truck type, total item length	42.5966	0.9226	0.9157	0.9120	0.85	10
3	weight utilisation, volume utilisation, total number of customers, total item weight, truck volume, volume of biggest customer, truck width, truck type, number of items allowed to be placed LxW, volume of last customer, number of cylinder items, total cylinder weight, truck height, total cylinder volume	36.6921	0.9199	0.9126	0.9087	0.8	14
4	weight utilisation, volume utilisation, total number of customers, total item weight, truck volume, volume of biggest customer, truck width, truck type, number of items allowed to be placed LxW	33.5979	0.9128	0.9091	0.9096	0.8	9
5	weight utilisation, volume utilisation, total number of customers, total item weight	0.7425	0.8864	0.8810	0.8809	0.8	4

Algorithm 1 Feature Selection

```
 1: function FeatureEngineering
 2:     for $correlationBound \leftarrow 0.7, 0.75, 0.8, 0.85$ do
 3:         $featureList \leftarrow$ ExcludeHighlyCorrelatedFeatures$(correlationBound)$
 4:         for $nrFeatures = 3, \ldots, 20$ do
 5:             $currentFeatures \leftarrow featureList[: nrFeatures]$
 6:             $currentFeaturesTime \leftarrow$ timeToCalcFeature$(currentFeatures)$
 7:             Fit Random Forest model on $currentFeatures$
 8:             Compute $precision, recall, accuracy$
 9:             Write $currentFeatures, currentFeaturesTime, correlationBound, nrFeatures,$
10:             $precision, recall, accuracy$ to csv line
11:         end for
12:     end for
13: end function

14: function ExcludeHighlyCorrelatedFeatures($correlationBound$)
15:     $finalFeatureList \leftarrow []$
16:     Sort $allFeaturesList$ on feature importance
17:     for all feature pairs $A, B$ in $allFeaturesList$ do
18:         Compute $pearsonCorr$ between $A$ and $B$
19:         if $pearsonCorr > correlationBound$ then
20:             Add $A, B, pearsonCorr$ to $correlationDF$
21:         end if
22:     end for
23:     for $f$ in $allFeaturesList$ do
24:         $abort \leftarrow False$
25:         $highlyCorrFeatures \leftarrow correlationDF[correlationDF['A'] = f]$
26:         for $highCorr_f$ in $highlyCorrFeatures$ do
27:             if $highCorr_f$ in $finalFeatureList$ then
28:                 $abort \leftarrow True$
29:             end if
30:         end for
31:         if $abort = True$ then
32:             continue
33:         else
34:             Add $f$ to $finalFeatureList$
35:         end if
36:     end for
37:     return $finalFeatureList$
38: end function
```

precision as the performance measure since the models within this grid will have good accuracy and the main KPI is precision.

Different k-values (3,5 and 10) for the k-fold cross-validation have been tested on Experiment Synthetic B. K=5 and k=10 clearly outperformed k=3 when considering bias. The difference in performance between k=5 and k=10 was neglectable. Therefore, k=5 was the best value for k since k=10 takes more computation time.

Table 4.3 presents the hyperparameters found through the method explained.

	Synthetic A	*Synthetic B*	*Company*
Algorithm	Auto	Ball tree	Auto
Leaf size	16	24	16
Metric	Euclidean	Manhattan	Manhattan
Num Neighbours	19	25	12
Weights	Distance	Distance	Distance

(a) KNN

	Synthetic A	*Synthetic B*	*Company*
Bootstrap	TRUE	TRUE	FALSE
Max depth	15	13	16
Max Features	Sqrt	Sqrt	Sqrt
Min samples leaf	3	3	4
Min samples split	2	4	2
Num estimators	44	72	48

(b) RF

	Synthetic A	*Synthetic B*	*Company*
C	100	1000	1
Gamma	auto	Scale	auto
Kernel	rbf	rbf	poly

(c) SVM

	Synthetic A	*Synthetic B*	*Company*
C	10	10	10
Penalty	None	None	l2

(d) LOG

Table 4.3: Hyperparameters for all models

5 Approximation Algorithm

This chapter introduces the approximation algorithm used for the comparison of the results. It is based on ORTEC's approximation algorithm. Thus is it representative of the type of (non-learning) algorithm used in practice today.

Recall that this thesis aims to use ML methods to rapidly yet accurately classify whether load plans will be feasible or not. The load plan feasibility check is usually performed using approximation methods. This thesis aims to outperform the current solution. Therefore, the performance of the ML feasibility checker should be compared to an existing solution. At ORTEC, an approximation method called 'Cube Calculation' is used. This algorithm was unavailable for this thesis, but a high-level description was. Therefore this chapter aims to recreate ORTEC's approximation algorithm.

Due to time limitations and scope, a simplified version is created. The three main simplifications are: The simplified algorithm only works with the Standard Truck type, whereas ORTEC's version also works with the Bay Truck type. The simplified algorithm assumes every load plan uses one specific pallet type, even though this is not the case. ORTEC's algorithm receives information on which item needs to be loaded on what pallet and considers this when calculating feasibility. The simplified algorithm assumes that customers cannot be combined on pallets. So that every pallet is loaded with the items of precisely one customer. First, for every item type, how many items fit on one pallet was calculated such that the pallet is maximally packed. Second, it was calculated for every Standard Truck type how many pallets fit in one truck, such that the truck is fully packed. Both only need to be calculated once. They are used as input for the algorithm together with the load plan in question.

Algorithm 2 present the pseudocode of the simplified algorithm. The algorithm works as follows. First, it is checked whether the total weight of the items in the load plan does not exceed the maximum allowed weight in the truck. If it does exceed the maximum allowed weight, it can immediately be concluded that the load plan is infeasible. Else, it is checked whether the total volume of the items does not exceed the volume of the truck. Again, if it does exceed the maximum allowed weight, it can immediately be concluded that the load plan is infeasible. Else, the algorithm continues. For every customer in the load plan, it is calculated how many pallets that customer fills. For example, if a customer has 20 items of type A and 15 items of type B. It is known that 40 items of type A fill a pallet, and 20 items of type B fill a pallet. Therefore this customer fills 20/40 + 15/20 = 1.25 pallet. This is rounded to 2 because the algorithm assumes only one customer per pallet. The amount of pallets per customer is summed for all customers. If this exceeds the number of pallets that fit in the truck in question, the load plan is predicted infeasible; else, it is predicted to be feasible.

Due to the simplifications of the algorithm, it can only be used on Experiment Synthetic A.

Algorithm 2 Approximation Algorithm

```
1:  function ApproximationAlgorithm($loadplan$, $maxItemsPerPallet$, $maxPalletsPerTruck$)
2:      if $totalItemWeight > maxAllowedWeightTruck$ then
3:          $feasibility \leftarrow 0$
4:          return $feasibility$
5:      end if
6:      if $totalItemVolume > maxAllowedVolumeTruck$ then
7:          $feasibility \leftarrow 0$
8:          return $feasibility$
9:      end if
10:     $nrPallets \leftarrow 0$
11:     for customer in loadplan do
12:         $itemsOfCustomer \leftarrow$ getItemsOfCustomer($customer$)
13:         $nrCubes \leftarrow 0$
14:         for $itemLine$ in $itemsOfCustomer$ do
15:             $nrCubes \leftarrow nrCubes + (itemLine['noItems']/itemLine['maxItemsPerPallet']$
16:         end for
17:         $nrPallets \leftarrow nrPallets + math.ceil(nrCubes)$
18:     end for
19:     if $nrPallets > maxPalletsPerTruck$ then
20:         $feasibility \leftarrow 0$
21:     else
22:         $feasibility \leftarrow 1$
23:     end if
24:     return $feasibility$
25: end function
```

6 Results

This chapter first section presents the models' performance. The following section introduces insights on the relation between the input features and 3D load feasibility. In the last section, the performance of the best ML model is compared with the approximation algorithm.

6.1. Model Performance

This section presents and evaluates the performance of the models built in Chapter 4. Models' performance on Experiment Synthetic A, Synthetic B and the Company Experiment are compared. Various evaluation techniques are used to analyse the model's performance, such as the confusion matrix, accuracy, precision, npv, McNemar's test, confidence, training time and testing time.

The box plots in this section are based on the performance of the in Chapter 4 created ML models on 100 samples of 30% of the test set of the experiment in question.

6.1.1. Confusion Matrix

This section will not compare the confusion matrices of the different models and experiments. It is meant to give the reader a general feeling of the performances of the models in the different experiments and to provide basic knowledge that helps to understand the metrics in the following subsections.

One way to evaluate the performance of a classification model is by creating a confusion matrix. A confusion matrix is an NxN table, where N is the number of classes. One axis represents the predictions of the model, and the other axis represents the ground truth. So, a confusion matrix not only gives information on how many test instances were (in)correctly classified, but it also shows, if an instance was wrongly classified, what other class it was predicted to be. In Figure 6.1 a confusion matrix for a binary classifier is displayed. The following list explains what each cell in this matrix represents.

- True Negative (TN) is the number of instances that were predicted as negative and are truly negative.
- True Positive (TP) is the number of instances that were predicted as positive and are truly positive.
- False Negative (FN) is the number of instances that were predicted as negative, but are truly positive.
- False Positive (FP) is the number of instances that were predicted as positive, but are truly negative.

ORTEC's KPI is to minimize the number of FP's.

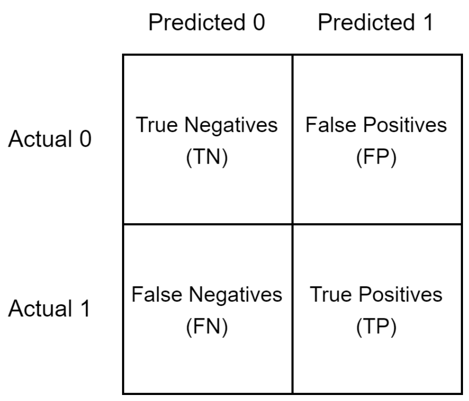


Figure 6.1: Confusion matrix explained

	Pred 0	Pred 1
Actual 0	1827	185
Actual 1	136	1640

(a) LOG

	Pred 0	Pred 1
Actual 0	1852	160
Actual 1	125	1651

(b) SVM

	Pred 0	Pred 1
Actual 0	1886	126
Actual 1	143	1633

(c) KNN

	Pred 0	Pred 1
Actual 0	1854	158
Actual 1	135	1641

(d) RF

Table 6.1: Confusion matrices for Experiment Synthetic A

	Pred 0	Pred 1
Actual 0	2843	364
Actual 1	331	3043

(a) LOG

	Pred 0	Pred 1
Actual 0	2865	342
Actual 1	297	3077

(b) SVM

	Pred 0	Pred 1
Actual 0	2888	319
Actual 1	318	3056

(c) KNN

	Pred 0	Pred 1
Actual 0	2931	276
Actual 1	317	3057

(d) RF

Table 6.2: Confusion matrices for Experiment Synthetic B

	Pred 0	Pred 1
Actual 0	3663	272
Actual 1	3913	35136

(a) LOG

	Pred 0	Pred 1
Actual 0	1479	2456
Actual 1	871	38178

(b) SVM

	Pred 0	Pred 1
Actual 0	3478	457
Actual 1	146	38903

(c) KNN

	Pred 0	Pred 1
Actual 0	3659	276
Actual 1	362	38687

(d) RF

Table 6.3: Confusion matrices for Company Experiment

6.1.2. Accuracy

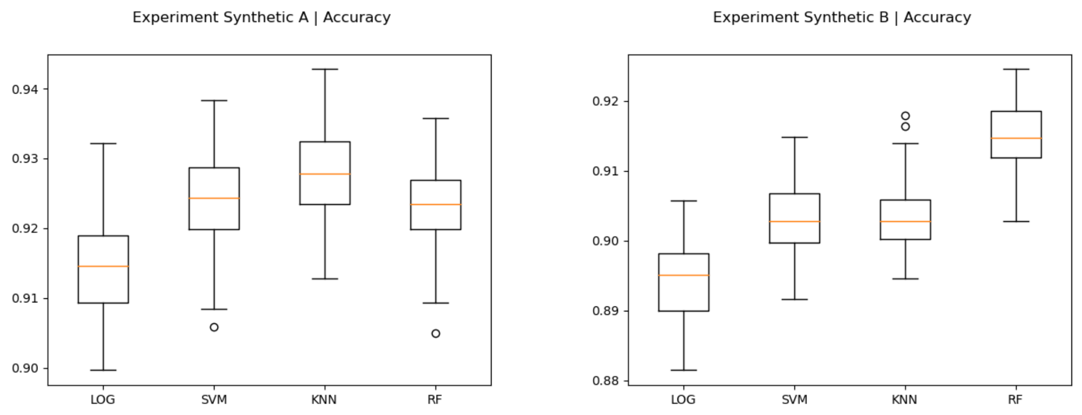
Accuracy is the sum of the correct predictions, divided by the total number of predictions. For binary classification, this is equal to the sum of TP and TN, divided by the sum of TP, TN, FP, and FN.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

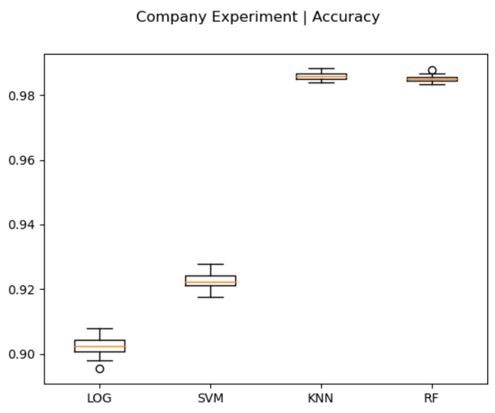
In Figure 6.2 an overview of the accuracies of all four models in the three different experiments is given. In Figure 6.2a, the simplest experiment, Experiment Synthetic A, is considered. The median accuracy for LOG is 91.5%. It can be observed that the accuracy improves with SVM, KNN and RF. The accuracy of these three models is almost identical. KNN shows the highest accuracy with a median of 92.8%. The performance of SVM and RF is 92.4% and 92.3%, respectively.

Looking into the accuracies of Experiment Synthetic B (Figure 6.2b), a clear distinction with Experiment Synthetic A is noticeable. In general, the accuracies of all models are lower. When comparing the different models within this experiment, RF clearly outperforms the other models, with a median accuracy of 91.5%. This suggests that when the use case is more complicated, RF is the best classifier. SVM and KNN show similar results, with a median accuracy of 90.3%. LOG clearly had the worst performance with a median accuracy of 89.5%.

Figure 6.2c shows the accuracies of the models in the Company Experiment. LOG (90.3%) and SVM (92.2%) clearly underperform in terms of accuracy compared to KNN and RF. KNN and RF show similar results with a median accuracy of 98.6% and 98.5%, respectively.



(a) Experiment Synthetic A **(b)** Experiment Synthetic B



(c) Company Experiment

Figure 6.2: The accuracy of LOG, SVM, KNN and RF on all three experiments

However, accuracy is not a good indicator of performance when considering class-imbalanced datasets, like the Company Data Experiment. In this dataset, 91% of the load plans are feasible. If a classifier predicted every load plan to be feasible, it would still reach an accuracy of 91%. This means that LOG performs worse than the most unintelligent model imaginable.

At the same time, accuracy should not be the only metric for balanced datasets either. If one classifier's prediction contains 8% FP and 2% FN, it has an accuracy of 10%. When another classifier's prediction contains 2% FP and 8% FN, it also has an accuracy of 10%. Whereas a specific use case would prefer one or the other.

6.1.3. Precision

Precision tells what proportion of positive predictions is actually positive. Equation 6.2 shows how this is calculated.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.2}$$

Figure 6.3 presents an overview of the precision of the four models in the three different experiments. Figure 6.3a presents the results of Experiment Synthetic A. KNN clearly shows the best performance with a median precision of 92.8%. SVM and RF perform very similarly with a median precision of 91.3% and 91.6%, respectively. LOG performs the worst with a median precision of 89.8%.

The precisions of the models in Experiment Synthetic B are shown in Figure 6.3b. In general, the precisions of the synthetic B experiments are slightly lower than those of the synthetic A experiments but are very close in relation. In this experiment, it is clear that RF outperforms the other models with a median precision of 92.2%. KNN is next best with a median precision of 90.5%. SVM achieved a median precision of 90% and LOG of 89.2%.

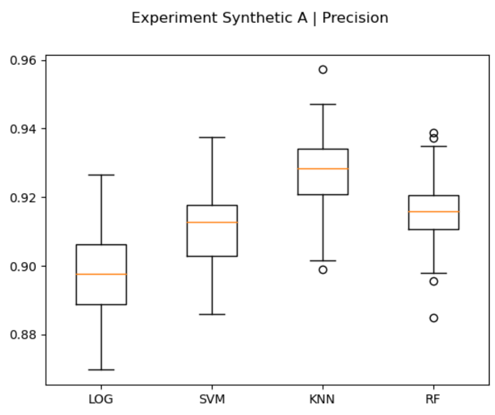
Figure 6.3c shows the results of the Company experiment. The first thing that stands out is that SVM performs poorly compared to the other models. SVM reached a median precision of 94%, while the other models all have a median precision greater than 98%. Namely 98.8%, 99.2% and 99.3% for KNN, LOG and RF respectively.

It is remarkable, that LOG performs so well in this experiment considering that it reached an accuracy of 90.3%, which was an under-achievement compared to all other models. The confusion matrix of LOG on the Company Experiment (Table 6.3a) shows that indeed when the model predicts that a load plan is feasible, it is correct in around 99% of the cases. However, when the model predicts that a load plan is not feasible, it is wrong about 50% of the time. This last metric is the NPV.

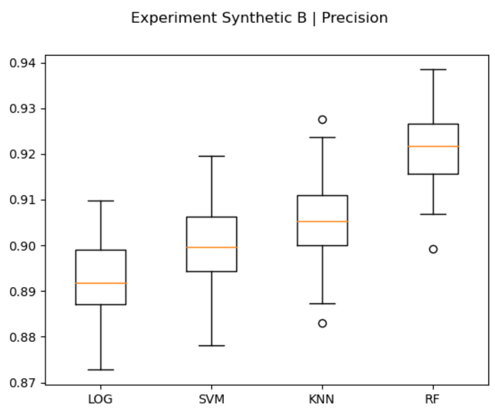
6.1.4. Negative Predictive Value

Negative Predictive Value (NPV) tells what proportion of negative predictions is actually negative. Equation 6.3 shows how this is calculated.

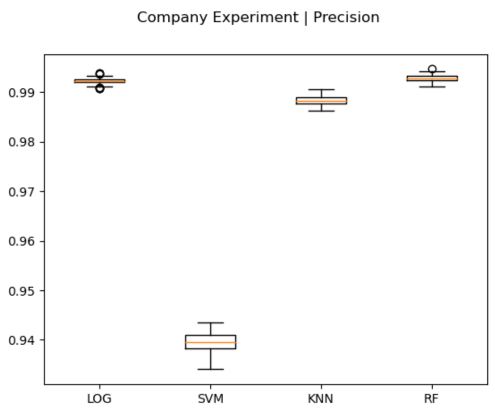
$$\text{NPV} = \frac{TN}{TN + FN} \tag{6.3}$$



(a) Experiment Synthetic A



(b) Experiment Synthetic B



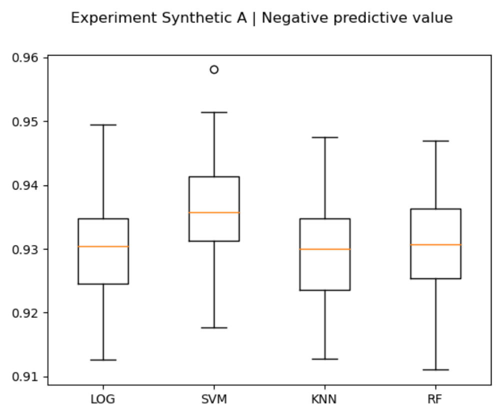
(c) Company Experiment

Figure 6.3: The precision of LOG, SVM, KNN and RF on all three experiments

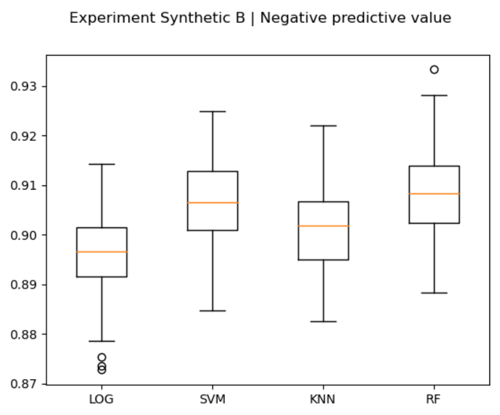
Figure 6.4 shows the NPV on the experiment for all models. Figure 6.4a gives an overview of the performance of the models on Experiment Synthetic A in terms of NPV. SVM shows the best performance with a median NPV of 93.6%. The other three models perform similarly with RF having a median NPV of 93.1% and KNN and LOG of 93%.

In experiment Synthetic B (Figure 6.4b RF and SVM perform almost identical, with a median NPV of 90.8% and 90.7%, respectively. KNN comes right behind that with 90.2%, and LOG performs the worst with 89.7%.

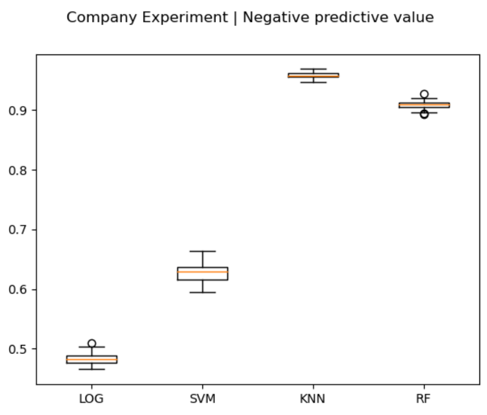
The performance on NPV of the models in the Company Experiment is presented in Figure 6.4c. This figure clearly shows that LOG and SVM are not suitable models for this classification task. LOG shows a median NPV of 48.2% and SVM of 62.9%. Where RF and KNN achieved a median NPV of, respectively, 90.9% and 95.8%. Therefore, the remainder of the result chapter will focus on KNN and RF.



(a) Experiment Synthetic A



(b) Experiment Synthetic B



(c) Company Experiment

Figure 6.4: The NPV of LOG, SVM, KNN and RF on all three experiments

6.1.5. McNemar's Test

This section investigates whether there is a statistical difference in performance between the KNN and RF model, for this McNemar's test is used. McNemar's test is a statistical test that can be used to compare the predictions of two ML models. The test uses a 2x2 contingency table of the predictions of the two models that are being compared. In figure 6.5 an overview of a contingency table is given. In cell A the number of instances both models predicted right is given. In cell D the number of instances both models predicted wrong is given. Cell B shows how many instances were correctly predicted by model 1, but incorrectly by model 2. Cell C shows the opposite of cell B.

The null hypothesis in McNemar's Test is formulated as follows: Neither of the two models outperforms the other. As a result, the alternative hypothesis is that the two models' performances are not the same, and one does outperform the other. The McNemar test statistic is chi-squared.

	Model 2 *correct*	Model 2 *wrong*
Model 1 *correct*	A	B
Model 1 *wrong*	C	D

Figure 6.5: Contingency table explained

For this research, we set the significance threshold to $\alpha = 0.05$. If the p-value is lower than this significance threshold, the null hypothesis that the two model's performances are equal is rejected.

	Correct	Wrong
Correct	3459	40
Wrong	60	229

(a) Synthetic A

	Correct	Wrong
Correct	5809	211
Wrong	135	426

(b) Synthetic B

	Correct	Wrong
Correct	42090	256
Wrong	291	347

(c) Company Experiment

Table 6.4: Contingency Matrices of KNN (x-axis) vs RF (y-axis)

For experiment Synthetic A, KNN slows a slightly better performance with 20 misclassifications less than RF, as can be seen in Table 6.4a. The Chi^2 value is 3.61, so the p-value equals 0.05743. This means that the null hypothesis is not rejected and thus that RF is not significantly better than KNN.

In experiment Synthetic B, KNN misclassified more instances than RF (see Table 6.4b). The Chi^2 value equals 16.257 and therefore the p-value equals 0.00005. This means the null hypothesis of the models' performance not differing is rejected. So RF is a significantly better model than KNN for experiment Synthetic B.

Table 6.4c shows the contingency matrix for the Company experiment. The percentage of correctly classified instances is very similar. The Chi^2 value is equal to 2.113 and thus the p-value equals 0.14601. This implies that there is no significant difference in the performance of KNN and RF on the Company experiment.

6.1.6. Confidence

Since the difference in performance is not significant for Experiment A and the Company experiment. We will look into the certainty that the models predicted the wrongly classified instances. Suppose one model predicted an infeasible load plan to be feasible with 51% certainty, and another model predicted it with 100% to be feasible. Arguably, the last model performs worse, even though they both made an incorrect prediction.

To investigate this, confidence plots were created. Figure 6.6 and 6.7 present confidence plots on the misclassified data. In both figures, the two plots at the top represent the confidence plots for RF and the bottom two for KNN. The blue histograms show the prediction probability the model had that an instance was feasible for the FN. These load plans are feasible, but the models return a prediction probability below 0.5, classifying them as infeasible. The red plots show exactly the opposite, namely the FP. These load plans were infeasible, but the model returned a probability of more than 0.5 that they were feasible.

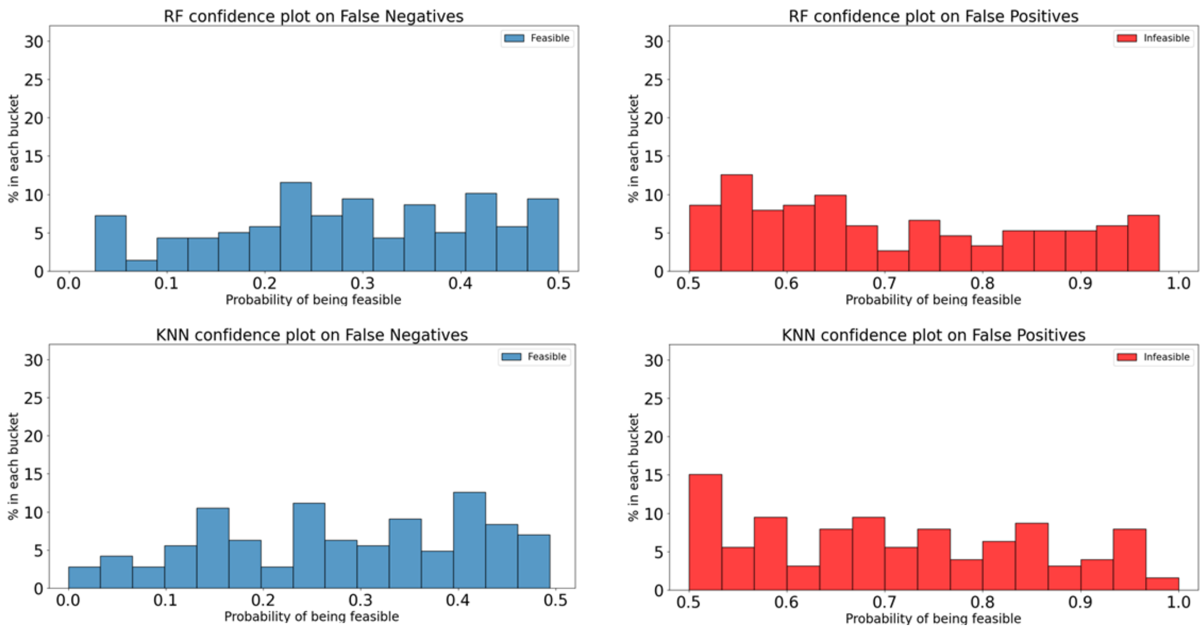


Figure 6.6: Confidence plots for misclassified data for Experiment Synthetic A for KNN and RF

In addition to the confidence plot, this section will also consider the average distance to correct classification. The distance meant is the distance from the prediction probability to 0.5. 0.5 is the threshold value that decides whether a prediction classifies as feasible or infeasible. The lower this distance is, for both FN and FP, the better the model. These distances are also compared using the unpaired t-test. The results for these tests are presented in Table 6.5.

Looking at Figure 6.6 the confidence distributions of experiment Synthetic A are quite evenly spread for both RF as KNN. The average distance for the RF model is 0.212, for KNN this is 0.217. An unpaired t-test was performed on the list of distances to the threshold value to check whether this difference is significant. This test returned a t-statistic of 0.420 and a p-value of 0.674. So, there is no significant difference in the performance regarding prediction probability on the misclassified data for this experiment.

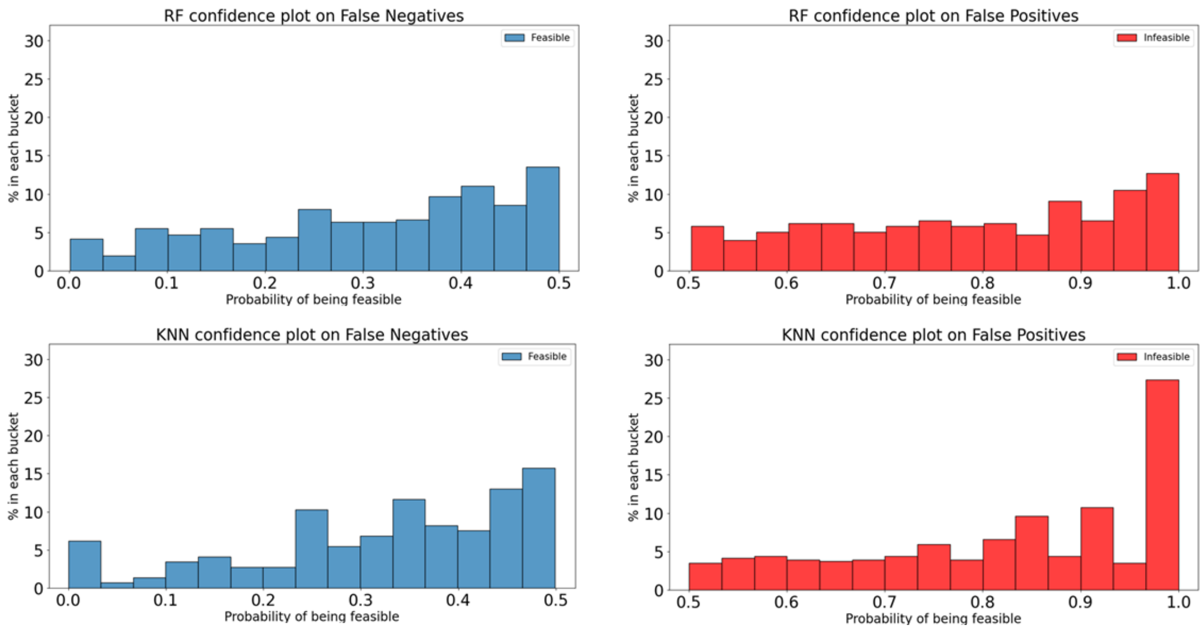


Figure 6.7: Confidence plots for misclassified data for Company Experiment for KNN and RF

For the Company experiment, the confidence distributions are displayed in Figure 6.7. One thing that immediately stands out is the peak in the FP plot of KNN. Solely visual analysis would suggest a difference in performance in terms of prediction probability on the misclassified data for the Company experiment. The average distance for RF equals 0.234, where the average distance for KNN is 0.292. The t-test returns a t-statistic of 6.409 and a p-value of 0.000, indicating a significant difference.

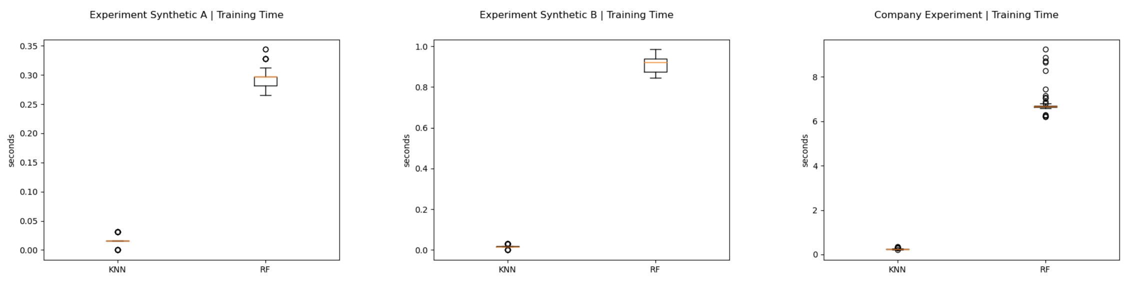
Table 6.5: Unpaired t-test on prediction probability distance to correct prediction

Experiment	Subject	t-statistic	p-value
Synthetic A	FN+FP	0.420	0.674
Company	FN+FP	6.409	0.000
Synthetic A	FN	0.594	0.991
Company	FN	-1.225	0.000
Synthetic A	FP	-0.011	0.991
Company	FP	3.570	0.000

6.1.7. Training and Testing Time

When two models show similar performance, training and testing time could be the deciding factor; the quickest in training and testing is typically the selected one. It is worth noting that training and testing time vary per machine, but the models can be compared because all models were run on the same system. The training and testing time mentioned in this thesis were recorded on a machine with an intel i7 processor, 4-core CPU and 16GB RAM.

Figure 6.8 shows the training time for KNN and RF in all experiments. A clear trend is visible, namely that RF takes relatively a lot more time to train than KNN. However, for the Company experiment, which consists of over 100,000 instances, the average training time is still just 6.78 seconds. It is important to remember that the training process is only performed occasionally.

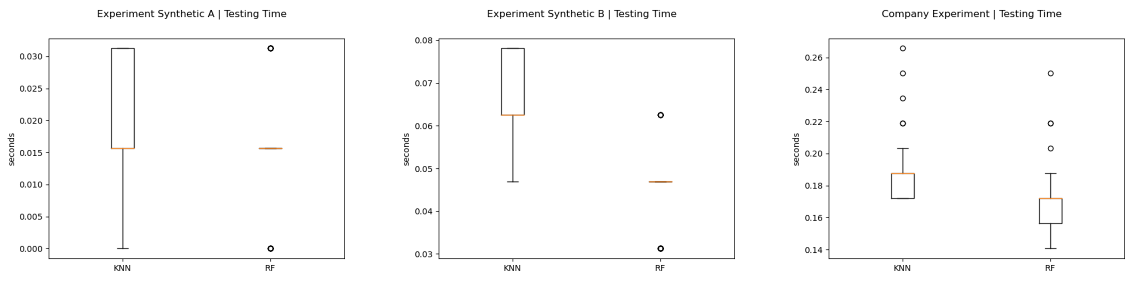


(a) Synthetic A **(b)** Synthetic B **(c)** Company

Figure 6.8: Comparison of training time, on the 70% training sets, between the approximation algorithm and the RF model

On the other hand, the testing time gives information about the time it takes to predict whether a load plan is feasible. This is an action that is performed very often in the iterative process of creating routes, as explained in 1.1. Doing this fast and precise was the initial motivation of this thesis.

In Figure 6.9 the testing times for KNN and RF in all experiments are presented. Again, a trend is visible, namely that RF is faster in terms of testing time. A paired t-test was performed for every experiment to check whether this difference is significant. The results of these tests are presented in table 6.6. It applies to all experiments that the testing time of RF is significantly shorter than that of KNN.



(a) Synthetic A **(b)** Synthetic B **(c)** Company

Figure 6.9: Comparison of testing time, on the 30% test sets, between the approximation algorithm and the RF model

Table 6.6: Results paired t-test on training time different between the approximation algorithm and the RF model for all experiments

Experiment	*t-statistic*	*p-value*
Synthetic A	-5.460	0.000
Synthetic B	-24.905	0.000
Company	-11.640	0.000

6.1.8. Conclusion

For Experiment Synthetic B, RF is a significant better model than KNN. For the other two experiments, KNN seems to perform slightly better, but this is not a significant difference. Taking into account the confidence for misclassified data and the testing time, RF does significantly outperform KNN. It is all those reasons combined that the rest of this chapter will solely focus on the RF model.

6.2. Insights

This section presents insights on the relation between the input features and 3D load feasibility. First every experiment will be evaluated individually using SHAP summary plots.

SHapley Additive exPlanations (SHAP) is a game theory technique that can help to understand how an ML model reaches its prediction. In the SHAP summary plots (Figure 6.10, 6.13 and 6.18), the feature names are listed on the y-axis in order of importance from top to bottom. The SHAP values are plotted on the x-axis. The colour represents the value of that feature. In the case of boolean features, the colour will be either pink for one or blue for zero. For non-boolean features, the colour will range in the spectrum of colours, with pink presenting high and blue low values. Each point represents one instance from the dataset.

6.2.1. Experiment Synthetic A

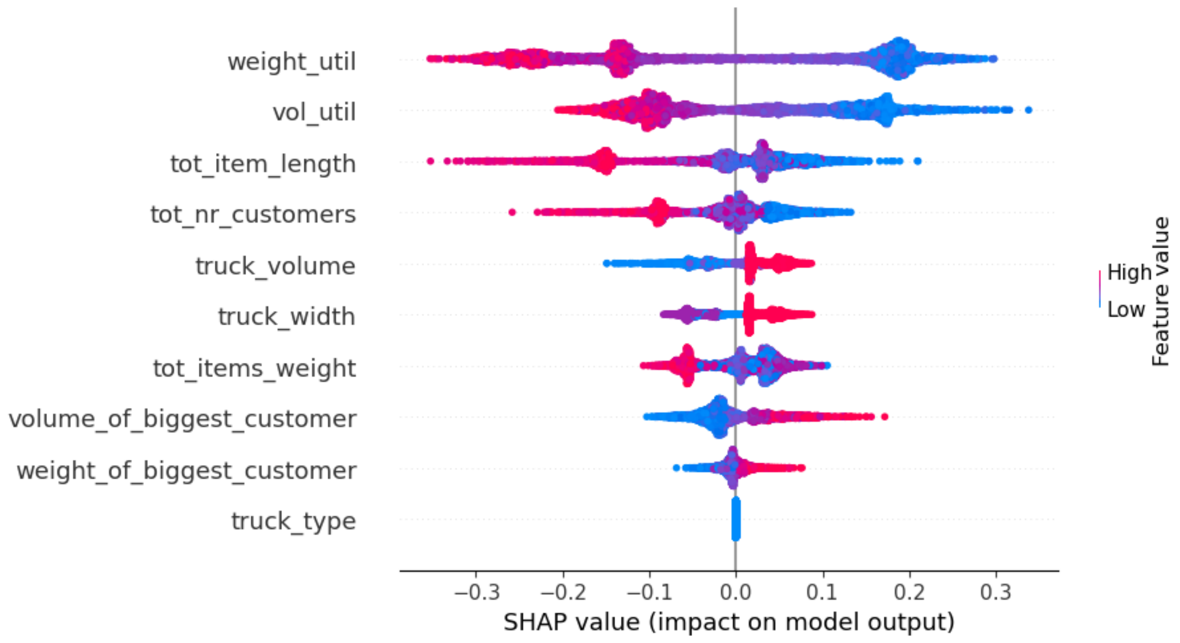


Figure 6.10: Shap summary plot for the RF model on Experiment Synthetic A

Weight utilization and **volume utilization** are the most important features for Experiment Synthetic A. For both features, it holds that the chance of feasibility decreases if the value is high and vice versa. The impact of weight utilization is more substantial than that of volume utilization.

The biggest instances of **total item length** decrease the chance of feasibility, and the smallest cases increase the chance of feasibility. However, for the less extreme instances, there is not such a clear relation. The same pattern is visible for the **total number of customers**. **Total item weight** exclusively has a clear relation when its value is high; the other instances do not clearly relate to the model output. For these unclear instances, the output does not depend solely on the value of one feature but on a combination of features. The data distributions (Figure 6.11) indicate that the combinations of total item length, the total number of customers or total item weight with truck volume are stronger predictors of feasibility.

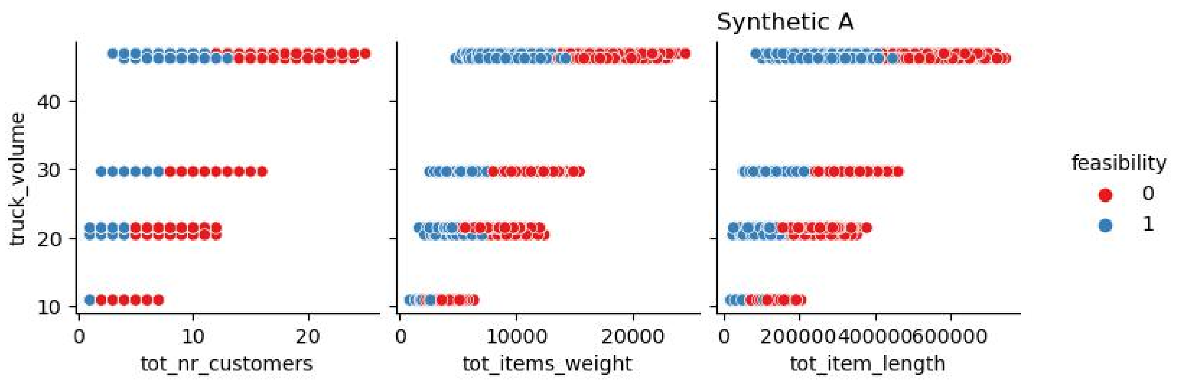


Figure 6.11: Scatterplot of truck volume vs total number of customers, total item weight and total item length of Experiment Synthetic A

Truck volume shows a positive relation with feasibility. So a load plan with a big truck has a higher chance of being feasible.

The SHAP values for **truck width** show an interesting pattern. A high truck width has a positive relation with feasibility, a low truck width does not influence feasibility much, and a medium truck width has a negative effect on feasibility. Figure 6.12 displays the distribution of truck width with the colour being feasibility. This plot indicates just three different truck widths: 2000, 2100 and 2200 mm and that the feasibility rate is lower for the trucks with a width of 2100 mm. This is because the data is created based on volume utilization (see chapter 3). It tells us that 2100 mm wide trucks are inefficient for optimal volume utilization. The pallets included in this experiment are 1200x1000, 1200x800 or 940x940. For a 2100 mm wide truck, there will always be at least 100 mm space left broadwise. However, some companies have rules about space between the pallets. For these companies, a 2100 mm truck would be a good option.

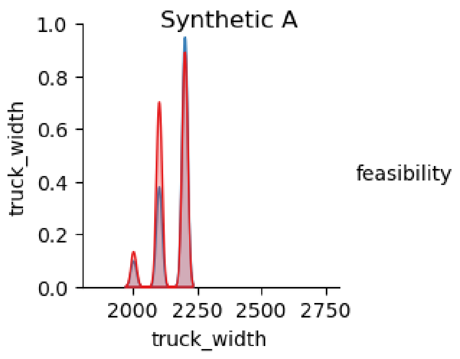


Figure 6.12: Distribution for truck width in experiment Synthetic A

The **volume of the biggest customer** and the **weight of the biggest customer** show a positive relationship with the feasibility. So if the volume of the biggest customer is low, the chance of the load plan being feasible is also low. This seems counter-intuitive at first, but taking into account that this dataset allows for one customer per pallet, the volume utilization of the truck will be low if the orders of the customers are small. Because the volume and weight of items are strongly correlated, the same reasoning holds for the weight of the biggest customer.

The **truck type** is irrelevant for this experiment since it only includes the standard truck and not the bay truck.

6.2.2. Experiment Synthetic B

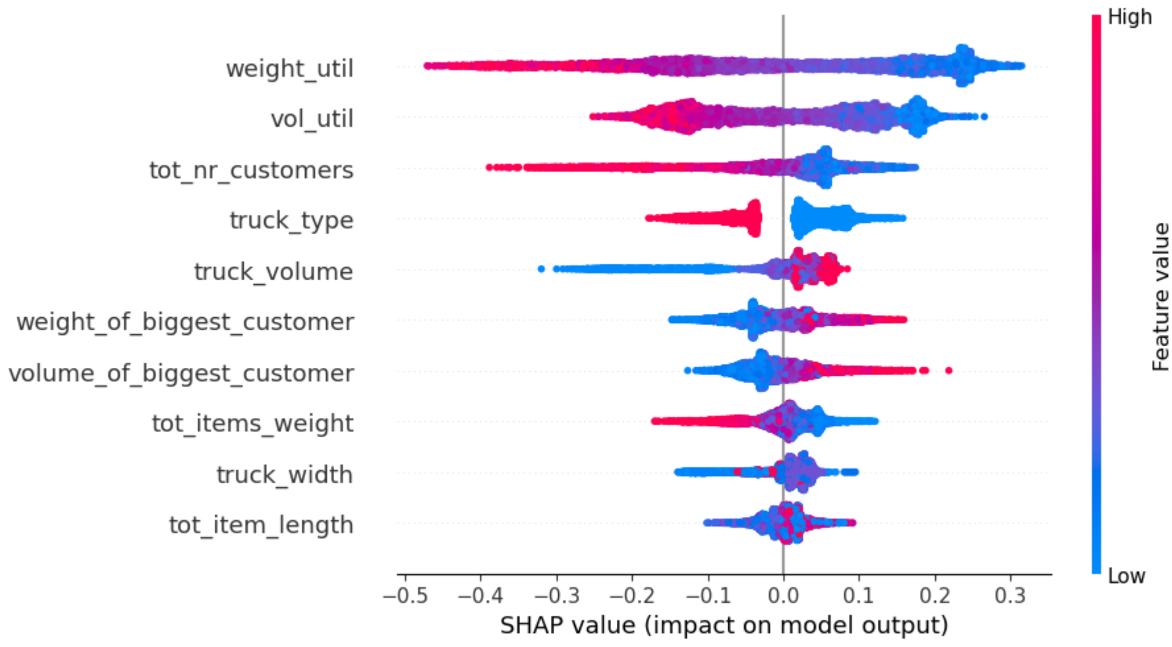


Figure 6.13: Shap summary plot for the RF model on Experiment Synthetic B

Just as in Experiment Synthetic A, **weight utilization** and **volume utilization** are the most influencing features for Experiment Synthetic B. For both features, the chance of feasibility decreases if the value is high and vice versa. The impact of weight utilization is more substantial than that of volume utilization.

Also, the combination of the **total number of customers** and the **total item weight** with the **truck volume** shows a strong relation with feasibility, just like in Experiment Synthetic A. However, the **total item length** does not in this experiment.

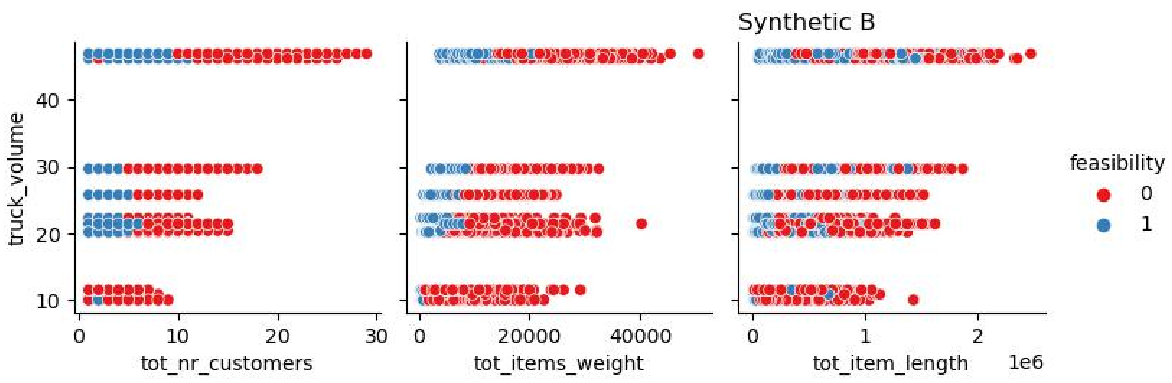
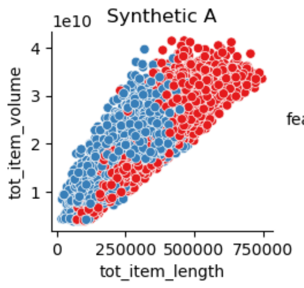
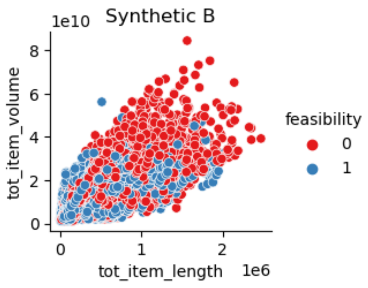


Figure 6.14: Scatterplot of truck volume vs total number of customers, total item weight and total item length of Experiment Synthetic B

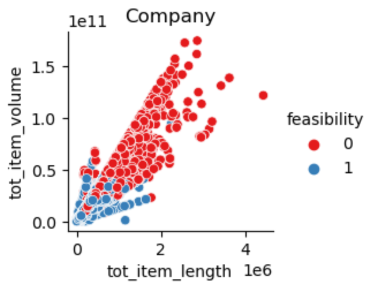
This reflects that the items from experiment synthetic B are a more heterogeneous group than the items from experiment synthetic A. The relation between item length - item volume, and item length - item weight is not as linear as in experiment synthetic A. Figure 6.15 depicts that, generally speaking, the items in experiment Synthetic A increase in volume when the length increases. For the items of Experiment Synthetic B, this relation is not that clear. The same holds for the relation between item length and weight. For this reason, item length is not the best predictor.



(a) Experiment Synthetic A

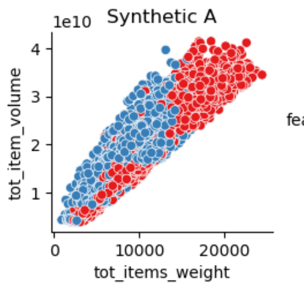


(b) Experiment Synthetic B

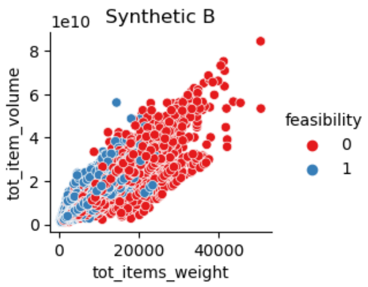


(c) Company Experiment

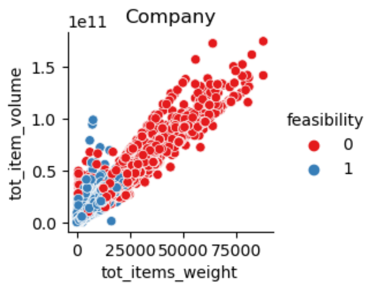
Figure 6.15: Relation item length and item volume



(a) Experiment Synthetic A



(b) Experiment Synthetic B



(c) Company Experiment

Figure 6.16: Relation item length and item weight

The **volume of the biggest customer** and the **weight of the biggest customer** show the same positive relationship with the feasibility, as explained for Experiment Synthetic A (subsection 6.2.1).

The SHAP plot shows that a high value for the **truck type**, meaning bay trucks, negatively affects the feasibility prediction, and a standard truck positively influences the feasibility prediction. This truck type feature is powerful in combination with other features, such as the total item weight. Figure 6.17 shows that the maximum allowed weight, in general, is a lot higher for standard trucks than for bay trucks.

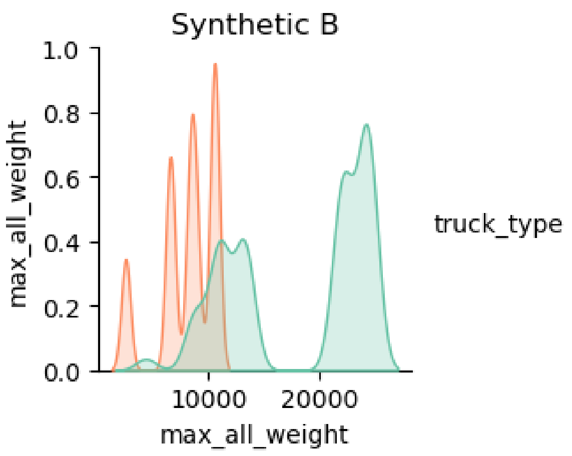


Figure 6.17: Distribution of maximum allowed weight in a truck for Experiment Synthetic B. Orange represents bay trucks, green represents standard trucks.

6.2.3. Company Experiment

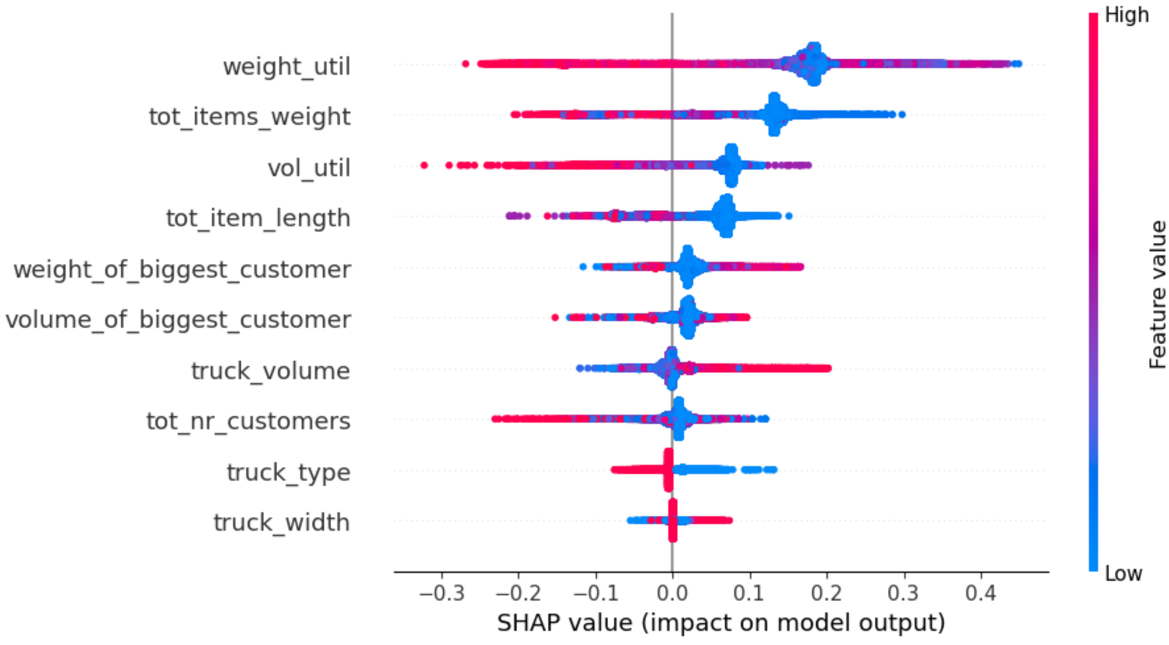
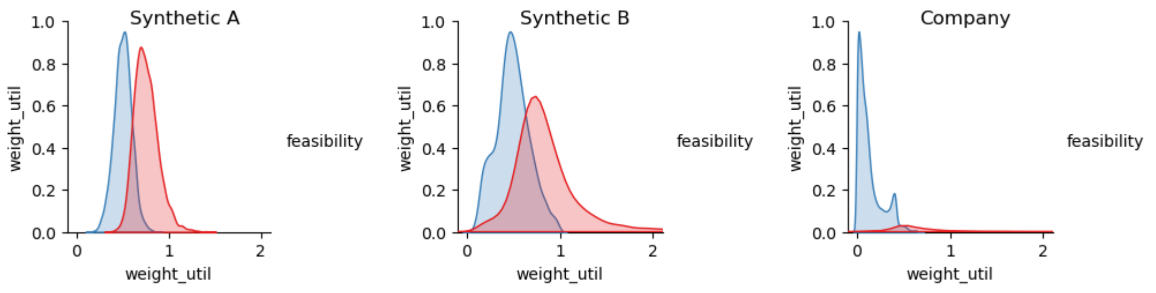


Figure 6.18: Shap summary plot for the RF model on the Company Experiment

Again, **weight utilization** is the most influencing feature. However, a difference in the colour assignment is visible. As shown in Figure B.1, in the synthetic experiments, the highest weight utilization is a lot bigger than for this experiment. Since the colour scale is relative per experiment, a dot's colour can represent different numbers. Even though the colours are different, the overall trend the SHAP plots indicate in all experiments is the same: a negative relation.



(a) Synthetic A **(b)** Synthetic B **(c)** Company

Figure 6.19: Distribution of the weight utilization for all experiments

In contrast to Experiment Synthetic B and just like Experiment Synthetic A, there exists an almost linear relation between item length - item volume and item length - item weight. This is visualized in Figure 6.15 and 6.16.

The **volume of the biggest customer** and the **weight of the biggest customer** have the opposite relation with the model output compared to the synthetic experiments. This is caused by the fact that 58% of the load plans have only one customer in this dataset. The volume and weight of the biggest customer are, in those cases, equal to the total volume and weight of the load plan. Therefore a low value increases the chance on feasibility.

Figure 6.20 shows the heatmap of the absolute means of the Shap interaction values. This plot shows the important main effects and interaction effects of features. The main effects of the features are given on the diagonal, corresponding to the Shap plot. The other values represent the interaction effects. One interaction effect that stands out is weight utilisation with total item weight. This interaction effect influences the outcome just as much as volume utilisation. The relation between these two features is based on the maximum allowed weight in a truck. The maximum allowed weight is a good indicator of which truck is considered. To clarify, the dataset includes 21 distinct truck widths, 38 truck heights, 40 truck lengths and 77 distinct truck volumes and 80 maximum allowed weights. So the maximum allowed weight is the best indicator of what truck is considered. This suggests that knowing what truck is considered is valuable. The model found a trend that the pattern found to predict feasibility is different for every truck.

6.3. Comparison with the Approximation Algorithm

This section compares the performance of the RF model and the approximation algorithm created in chapter 5 on Experiment Synthetic A. This is the only experiment possible for comparison, because the approximation algorithm does not allow for bay trucks. The box plots in this section are based on the performance on 100 samples of 30% of the test set of the experiment.

6.3.1. Accuracy

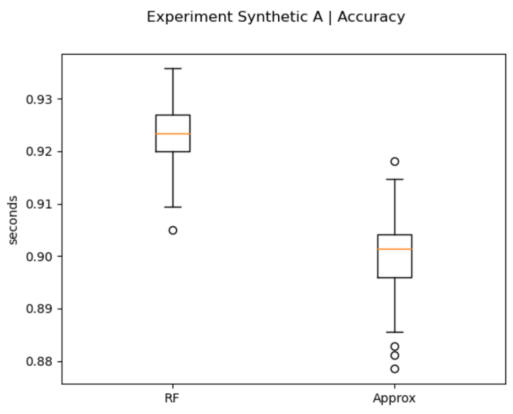
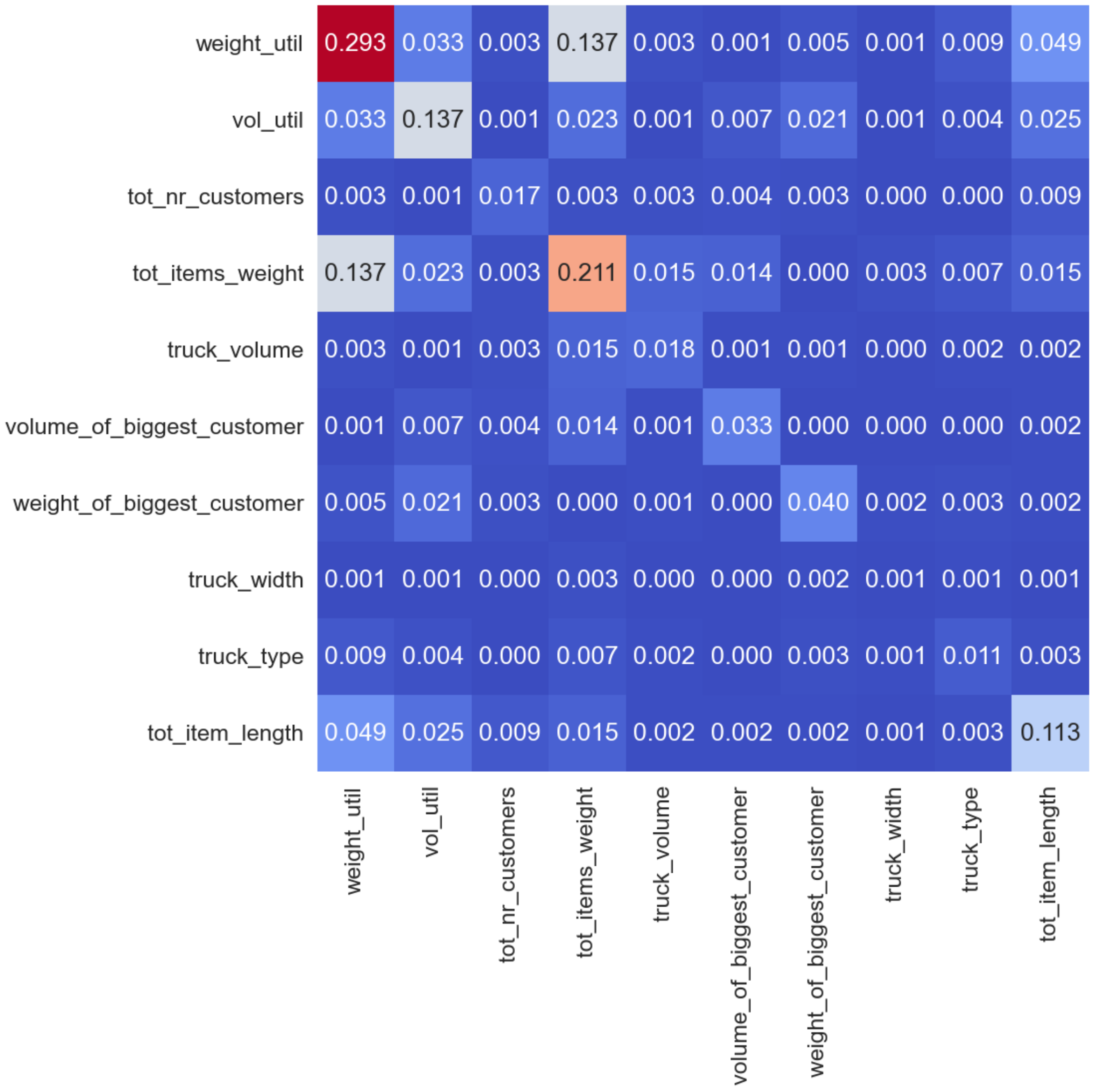


Figure 6.21: The accuracy of RF and the Approximation Algorithm



	weight_util	vol_util	tot_nr_customers	tot_items_weight	truck_volume	volume_of_biggest_customer	weight_of_biggest_customer	truck_width	truck_type	tot_item_length
weight_util	0.293	0.033	0.003	0.137	0.003	0.001	0.005	0.001	0.009	0.049
vol_util	0.033	0.137	0.001	0.023	0.001	0.007	0.021	0.001	0.004	0.025
tot_nr_customers	0.003	0.001	0.017	0.003	0.003	0.004	0.003	0.000	0.000	0.009
tot_items_weight	0.137	0.023	0.003	0.211	0.015	0.014	0.000	0.003	0.007	0.015
truck_volume	0.003	0.001	0.003	0.015	0.018	0.001	0.001	0.000	0.002	0.002
volume_of_biggest_customer	0.001	0.007	0.004	0.014	0.001	0.033	0.000	0.000	0.000	0.002
weight_of_biggest_customer	0.005	0.021	0.003	0.000	0.001	0.000	0.040	0.002	0.003	0.002
truck_width	0.001	0.001	0.000	0.003	0.000	0.000	0.002	0.001	0.001	0.001
truck_type	0.009	0.004	0.000	0.007	0.002	0.000	0.003	0.001	0.011	0.003
tot_item_length	0.049	0.025	0.009	0.015	0.002	0.002	0.002	0.001	0.003	0.113

Figure 6.20: Absolute mean of the Shap interaction values for the Company Experiment

Figure 6.21 presents the accuracy of RF and the approximation algorithm on Experiment Synthetic A. RF clearly shows a better performance with a median accuracy of 92.3% versus 90.1% for the approximation algorithm.

6.3.2. Precision

Figure 6.22 compares the precision of RF with the approximation method on Experiment Synthetic A. In contract with accuracy, for precision the approximation technique clearly outperforms RF, with a median precision of 95.9% against 91.6%.

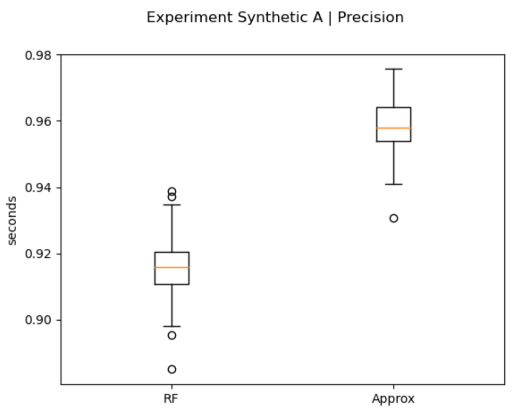


Figure 6.22: The precision of RF and the Approximation Algorithm

6.3.3. NPV

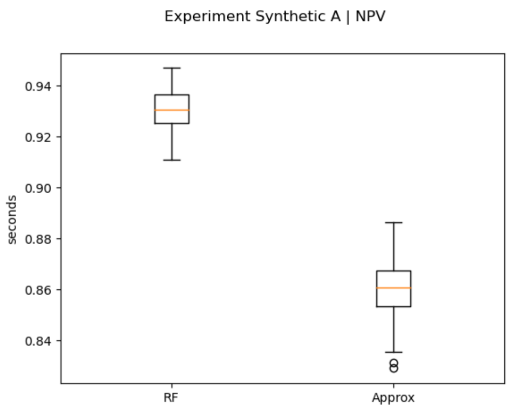


Figure 6.23: The NPV of RF and the Approximation Algorithm

Figure 6.23 depicts the NPV of RF and the approximation method on Experiment Synthetic A. RF clearly outperforms the approximation approach, with a median accuracy of 93.1% vs 86.1% for the approximation algorithm.

6.3.4. McNemar

For experiment Synthetic A, RF shows a better performance with 87 misclassifications less than the approximation algorithm, as can be seen in the contingency table (Table 6.7). The Chi^2 value is 18.444, so the p-value equals 0.000. This means that the null hypothesis is rejected and thus that RF is a significantly better model than the approximation algorithm.

	Correct	Wrong
Correct	3255	244
Wrong	157	132

Table 6.7: Contingency Matrices of Approx (x-axis) vs RF (y-axis) on Experiment Synthetic A

6.3.5. Time

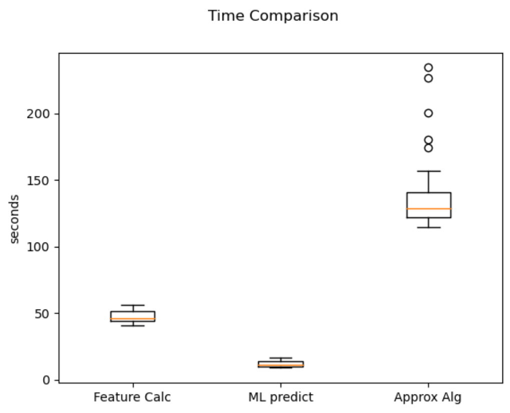


Figure 6.24: The computation time of RF and the Approximation Algorithm

Figure 6.24 presents a comparison between the computation time for the RF model and the approximation algorithm. The computation time is split into two parts: features calculation time and prediction time. The results are based on 50 runs on the same subset of 1000 instances of Experiment Synthetic A. It is run per instance, and the time of the 1000 instances is summed up. This process is repeated 50 times. The median feature calculation time is 46.34 seconds, with a standard deviation of 4.91 seconds. The median ML prediction time is equal to 11.59 seconds with a standard deviation of 2.10 seconds. For the approximation algorithm, the computation time is much higher, namely a median of 128.49 seconds with a standard deviation of 22.12 seconds.

Even though the feature calculation time and the ML prediction time are both not normally distributed, for comparison, their medians are summed. It will give a better sense of the improvements in computation time. The sum of the feature calculation time and the ML prediction time equals 57.93 seconds. This implies that switching to an ML approach improves the computation time by 54.9%.

6.4. Summary

This chapter first looks at the performance of the four models, created in Chapter 4, on the three different experiments explained in Chapter 3. The performance of the models is very similar in Experiment Synthetic A. However, in Experiment Synthetic B and especially the Company data experiment, it is evident that KNN and RF outperform SVM and LOG. McNemar's test showed that in Experiment Synthetic B, the RF model significantly outperforms the KNN model. This difference was not significant for Experiment Synthetic A and the Company Experiment. The analysis of the confidence plots on the misclassified data in Section 6.1.6 showed that the RF model was less sure about the instances it wrongly classified. On average, the KNN model felt more sure about the wrong predictions it made. This difference was significant for the Company Experiment. With regards to the training and testing time, RF has a higher training time, but still acceptable. RF also has a lower testing time. Since training only needs to be done occasionally, and predictions are often made, a short testing time is more important than a short training time. Combining all these reasons concludes that RF is the best model.

Section 6.2 uses SHAP to help understand how an ML model reaches its predictions. It was shown that weight utilisation and volume utilisation are fundamental features in predicting feasibility for all experiments. The effects of the other features are more experiment (and thus scenario) dependent. For example, in Experiment Synthetic A, the total item length is a powerful predictor, whereas this is the least important feature for Experiment Synthetic B. It was also shown that features' interaction effect could be precious. In the Company Experiment, the interaction effect of the weight utilisation and the total item weight is significant. In this case, this interaction effect symbolises the maximum allowed weight. Moreover, the maximum allowed weight is a good identifier for the truck used.

Table 6.8: Comparison of the performance of the RF model and the approximation algorithm

	RF	**Approx**
Accuracy	92.30%	90.10%
Precision	91.60%	95.90%
NPV	93.10%	86.10%

Table 6.8 gives an overview of the performance of the RF model and the approximation algorithm on Experiment Synthetic A. The KPI is to reach the best precision. However, the general performance, thus accuracy, should be good. Even though the

approximation algorithm has better precision, this is not a good predictor. McNemar's test showed that the RF model performs significantly better in general. Also, the ML model improves computation time by 54.9%.

7 Conclusion

This chapter first provides the answers to the research questions defined in section 1.4. Second, possible future research directions are presented.

7.1. Answers to Research Questions

RQ1: How accurate can an ML approach predict whether a 3D load plan will be feasible or not?
In section 6.1 we see that the RF model outperforms the LOG, SVM and KNN models created in chapter 4. This model reaches an accuracy of 91.5% for Experiment Synthetic B, 92.3% for Experiment Synthetic A and 98.5% for the Company Experiment. This means that the performance of a ML model is dependant on the scenario, but overall shows great potential.

RQ2: What insights on the relation between input variables and 3D load feasibility can be extracted from the trained ML model?
In Section 6.2, it was found that weight utilisation was the most important feature in predicting feasibility for all experiments. The volume utilisation came in second for both synthetic experiments and third for the Company Experiment. Therefore, we conclude that weight and volume utilisation are significant predictors of load feasibility in general.

The effect of the other features is dependent on the scenario. For example, in Experiment Synthetic A, the total item length is a significant predictor, whereas this is the least important feature for Experiment Synthetic B.

RQ3: What are the improvements derived by an ML model for load feasibility checking versus current approximation algorithms?
The answer to this question is based on comparing the performance of the RF model

and the approximation algorithm on Experiment Synthetic A. These results suggest switching to an ML approach for feasibility prediction decreases computation time by 54.9%. Also, the RF model is significantly better in predicting load plan feasibility. The RF model offers precision and accuracy of respectively 95.9% and 92.3%. Against 91.6% and 90.1% for the approximation algorithm. This indicates that the ML approach increases the precision with 4.3%-point and accuracy with 2.2%-point. Further, it is expected that ML scales better in performance and time if the problem instances are larger.

7.2. Discussion and Future Work

This section discusses the drawbacks of the thesis and possible future work.

The approximation algorithm used for comparison
Since the approximation algorithm used by ORTEC was not available and due to time limitations, a simplified version of ORTEC's algorithm was created (Chapter 5). Therefore, the comparison between the performance of the ML model and the approximation undertaken in section 6.3 is not entirely realistic. ORTEC's actual approximation algorithm is expected to be a better and faster feasibility predictor than that of chapter 5. Nonetheless, the advantages of ML approach are expected to remain.

Comparison only on Experiment Synthetic A
Another fallback of the simplified approximation algorithm is that it could only be used on Experiment Synthetic A. This thesis hypothesises that the ML solution scales better in terms of time and performance when the experiments get more complicated. This is something that should be tested in the future.

Performance in different real-world use cases
This thesis uses one real dataset. The first fallback is that this dataset is imbalanced (91% instances are feasible). Also, this dataset consists of (1) specific scenarios, (2) from one company, (3) in one specific region (4) of one industry. This thesis aims to prove a concept. However, in future research, it should be investigated how well ML models perform load feasibility checks for different scenarios, companies, regions and industries.

Investigate the performance of a minimal feature set
The feature selection method in Section 4.2 showed that the RF model shows good performance on Experiment Synthetic B with just the features weight utilisation, volume utilisation, total number of customers and total item weight. It would be valuable to further investigate the performance of such a simple model.

Combination of different algorithms
Such simple models are specifically interesting for a combinatorial solution. For example, a very simple model could predict the more straightforward instances. If the prediction probability is below a certain threshold, that instance will be predicted by a

more complex model or heuristic algorithm.

Insights on how often feasibility check is performed
It is recommended to get better insights on how often the feasibility check is performed. Then it could be analysed what the best balance between computation time and accuracy is.

Time measurement
The scripts created for this thesis are not optimised in terms of computation time. Therefore, the time measurements performed in this are sufficient for comparison with each other, but the time on its own is invaluable. Further improvements such as parallel programming should be used when the project is deployed.

Bibliography

Araya, I., & Riff, M.-C. (2014). A beam search approach to the container loading problem. *Computers & Operations Research*, *43*, 100–107.

Aydemir, M., & Yigit, T. (2019). A review of the solutions for the container loading problem, and the use of heuristics. *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*, 690–700.

Bischoff, E. E., Janetz, F., & Ratcliff, M. (1995). Loading pallets with non-identical items. *European journal of operational research*, *84*(3), 681–692.

Bischoff, E. E., & Ratcliff, M. (1995). Issues in the development of approaches to container loading. *Omega*, *23*(4), 377–390.

Bortfeldt, A., & Gehring, H. (1998a). Applying tabu search to container loading problems. *Operations research proceedings 1997* (pp. 533–538). Springer.

Bortfeldt, A., & Gehring, H. (1998b). A tabu search algorithm for weakly heterogeneous container loading problems. *OR SPEKTRUM*, *20*(4), 237–250.

Bortfeldt, A., & Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research*, *131*(1), 143–161.

Bortfeldt, A., Gehring, H., & Mack, D. (2003). A parallel tabu search algorithm for solving the container loading problem. *Parallel computing*, *29*(5), 641–662.

Bortfeldt, A., & Homberger, J. (2013). Packing first, routing second—a heuristic for the vehicle routing and loading problem. *Computers & Operations Research*, *40*(3), 873–885.

Bortfeldt, A., & Wäscher, G. (2013). Constraints in container loading–a state-of-the-art review. *European Journal of Operational Research*, *229*(1), 1–20.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, *44*(2), 145–159.

Eley, M. (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research*, *141*(2), 393–409.

Fanslau, T., & Bortfeldt, A. (2010). A tree search algorithm for solving the container loading problem. *INFORMS Journal on Computing*, *22*(2), 222–235.

Faroe, O., Pisinger, D., & Zachariasen, M. (2003). Guided local search for the three-dimensional bin-packing problem. *Informs journal on computing*, *15*(3), 267–283.

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, *36*(3), 655–673.

Gehring, H. et al. (1997). A genetic algorithm for solving the container loading problem. *International transactions in operational research*, *4*(5-6), 401–418.

Gehring, H., & Bortfeldt, A. (2002). A parallel genetic algorithm for solving the container loading problem. *International Transactions in Operational Research*, *9*(4), 497–511.

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks: An International Journal*, *51*(1), 4–18.

George, J. A., & Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers & Operations Research*, *7*(3), 147–156.

Gilmore, P. C., & Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations research*, *13*(1), 94–120.

Hemminki, J. (1994). *Container loading with variable strategies in each layer*. University of Turku.

Hifi, M. (2002). Approximate algorithms for the container loading problem. *International Transactions in Operational Research*, *9*(6), 747–774.

IBM. (2021). Supervised Learning. https://www.ibm.com/cloud/learn/supervised-learning

Iori, M. (2005). *Metaheuristic algorithms for combinatorial optimization problems* (Doctoral dissertation). Springer.

Iori, M., Salazar-González, J.-J., & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation science*, *41*(2), 253–264.

Lim, A., Rodrigues, B., & Wang, Y. (2003). A multi-faced buildup algorithm for three-dimensional packing problems. *Omega*, *31*(6), 471–481.

Liu, S., Tan, W., Xu, Z., & Liu, X. (2014). A tree search algorithm for the container loading problem. *Computers & Industrial Engineering*, *75*, 20–30.

Mack, D., Bortfeldt, A., & Gehring, H. (2004). A parallel hybrid local search algorithm for the container loading problem. *International Transactions in Operational Research*, *11*(5), 511–533.

Morabito, R., & Arenales, M. (1994). An and/or-graph approach to the container loading problem. *International Transactions in Operational Research*, *1*(1), 59–73.

Moura, A., & Oliveira, J. F. (2005). A grasp approach to the container-loading problem. *IEEE Intelligent Systems*, *20*(4), 50–57.

Parreño, F., Alvarez-Valdés, R., Tamarit, J. M., & Oliveira, J. F. (2008). A maximal-space algorithm for the container loading problem. *INFORMS Journal on Computing*, *20*(3), 412–422.

Pisinger, D. (2002). Heuristics for the container loading problem. *European journal of operational research*, *141*(2), 382–392.

Sheng, L., Hongxia, Z., Xisong, D., & Changjian, C. (2016). A heuristic algorithm for container loading of pallets with infill boxes. *European Journal of Operational Research*, *252*(3), 728–736.

Sixt, M. (1996). *Dreidimensionale packprobleme: Lösungsverfahren basierend auf den meta-heuristiken simulated annealing und tabu-suche*. Lang.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European journal of operational research*, *183*(3), 1109–1130.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, *195*(3), 729–743.

Zhang, D., Peng, Y., & Leung, S. C. (2012). A heuristic block-loading algorithm based on multi-layer search for the container loading problem. *Computers & Operations Research*, *39*(10), 2267–2276.

Zhou, Q., & Liu, X. (2017). A swarm optimization algorithm for practical container loading problem. *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, 5690–5695.

A
Data Pairplots

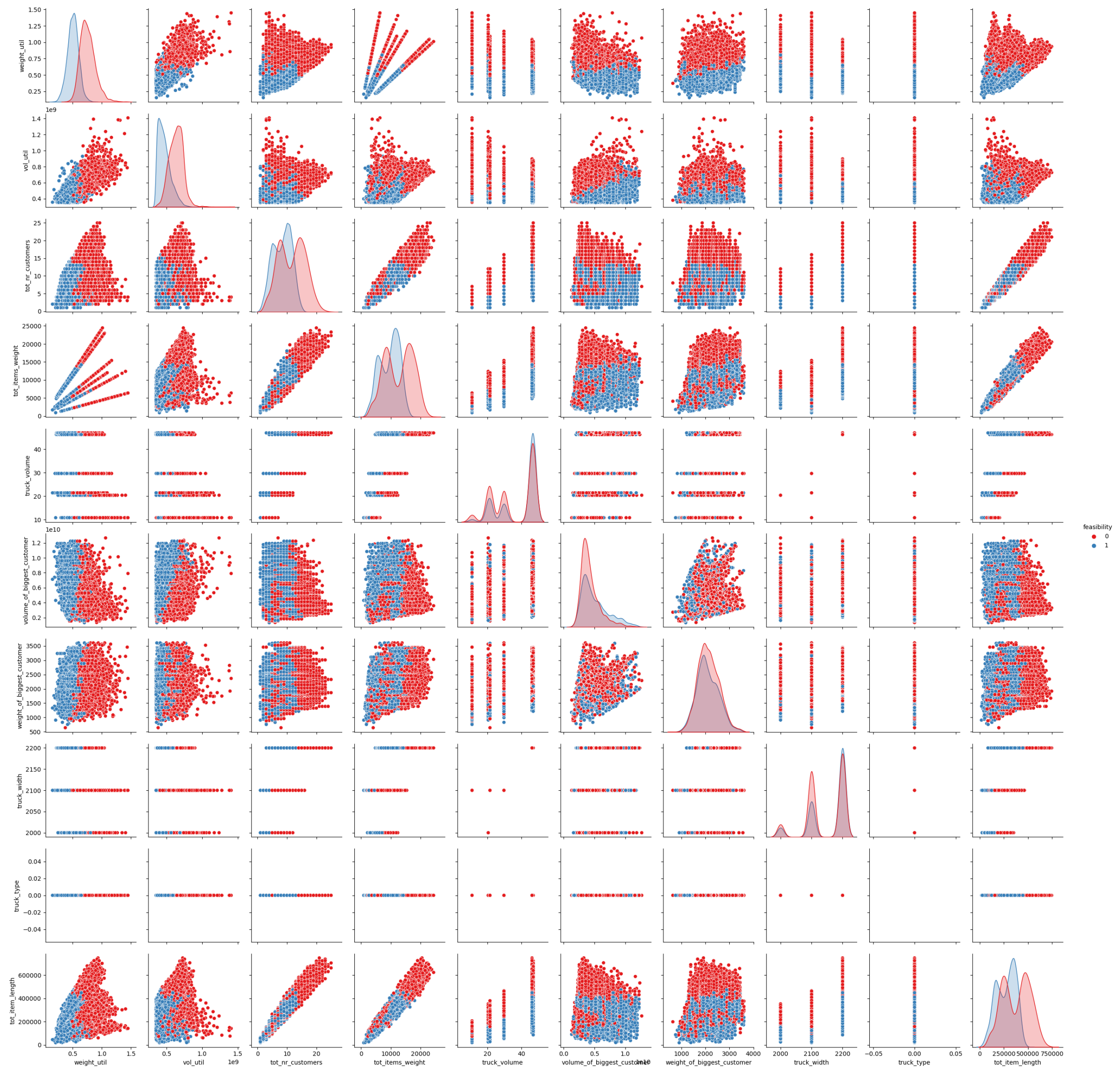


Figure A.1: Pairplot Experiment Synthetic A

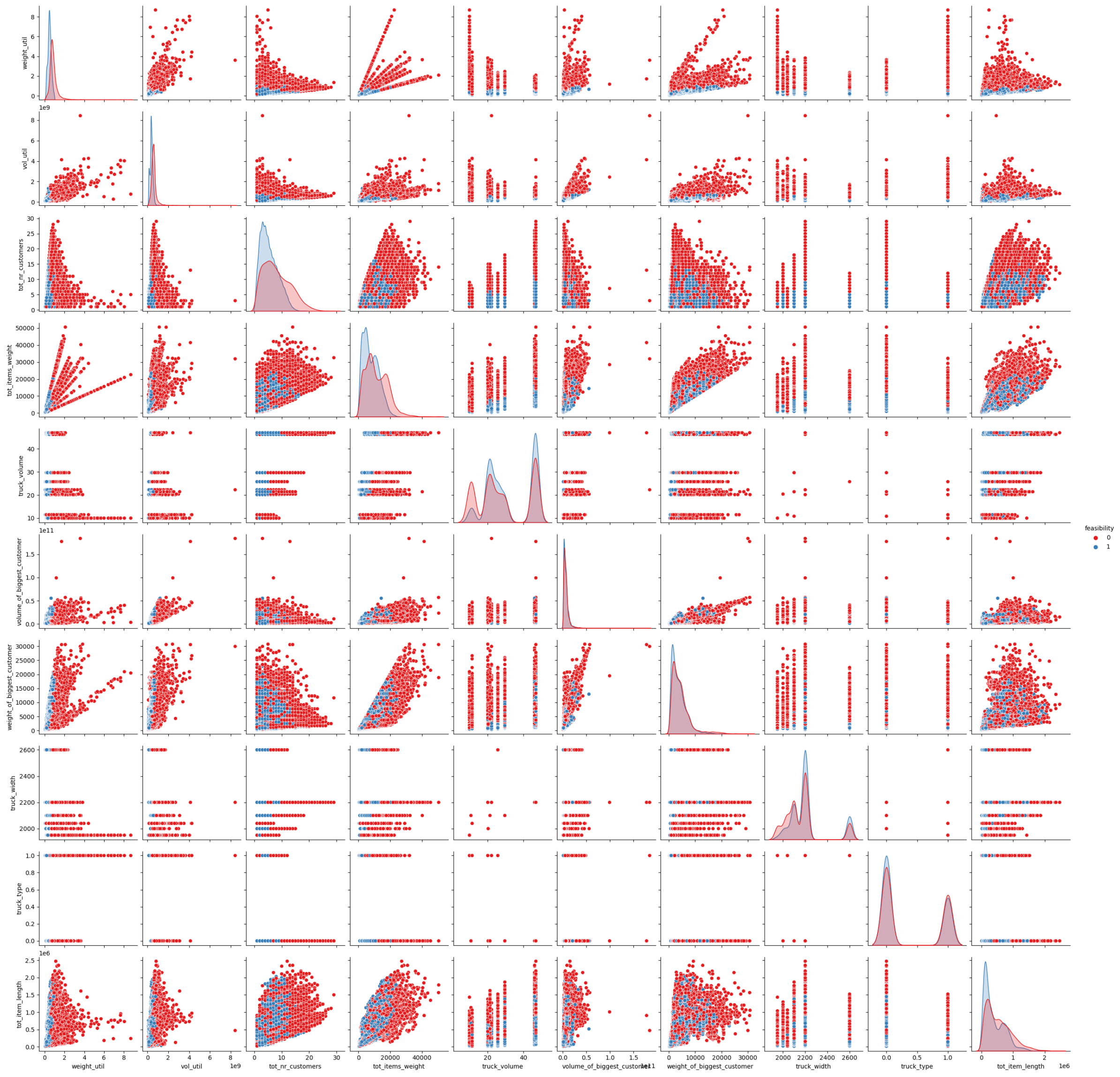


Figure A.2: Pairplot Experiment Synthetic B

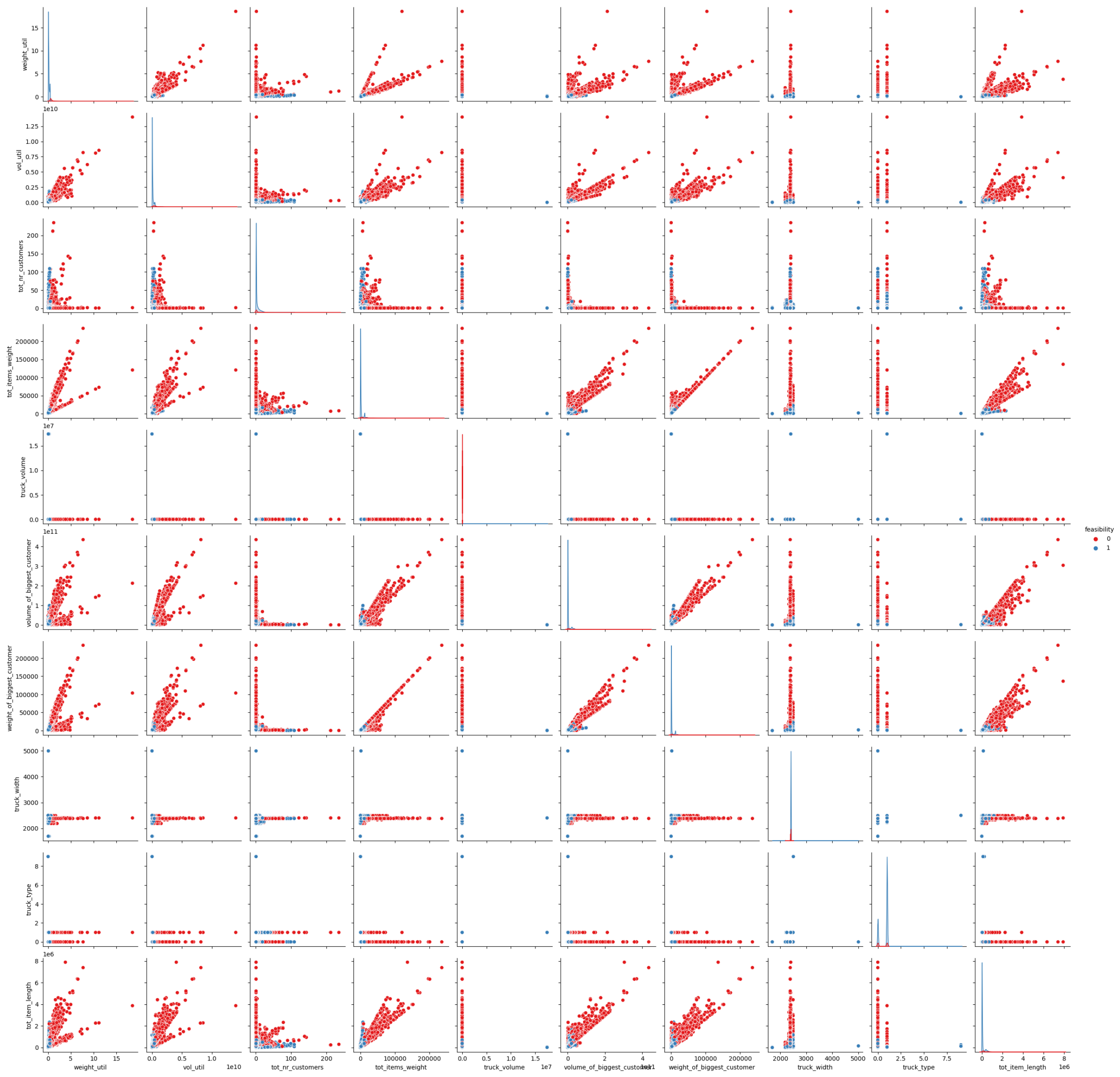


Figure A.3: Pairplot Company Experiment

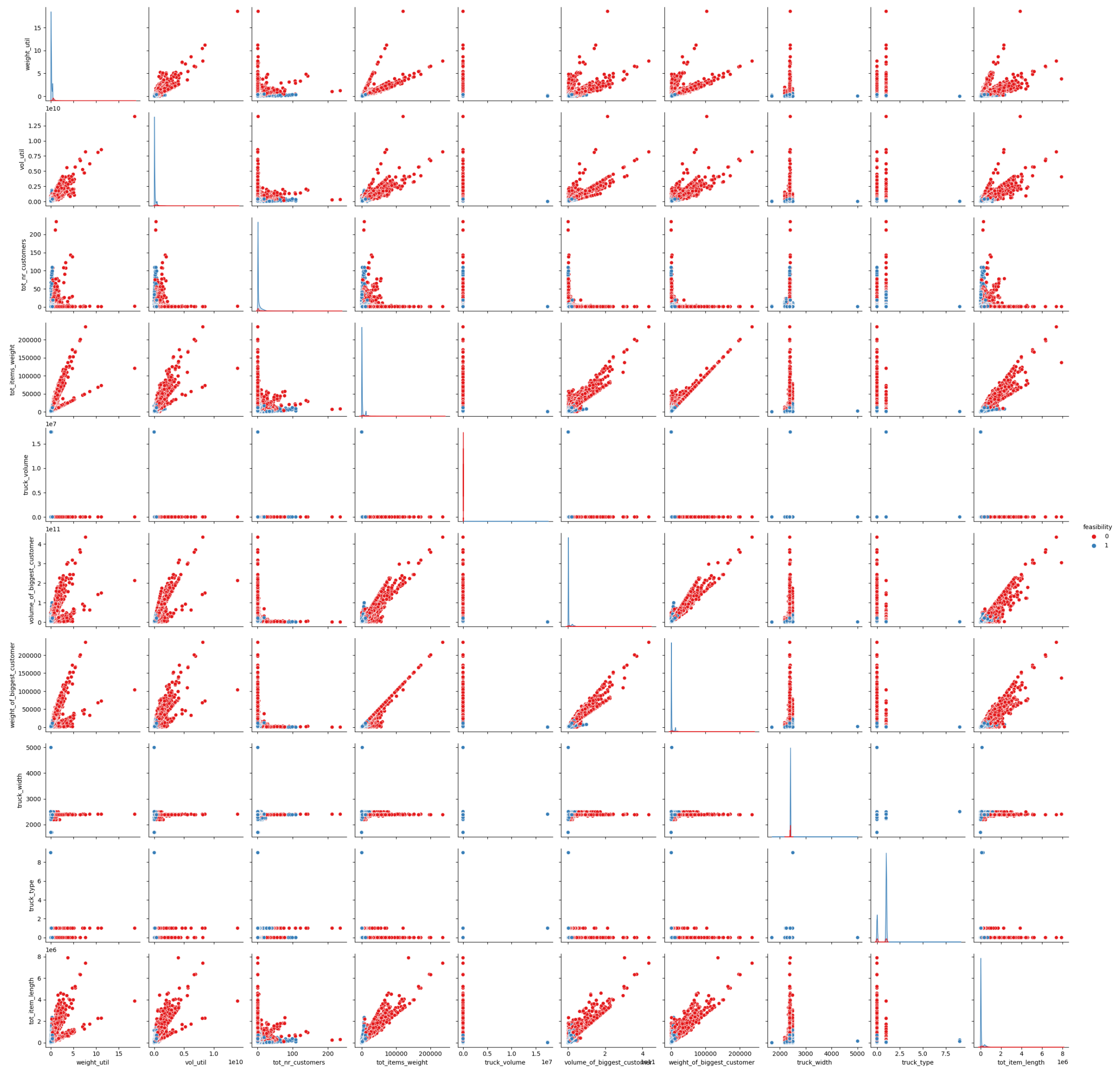
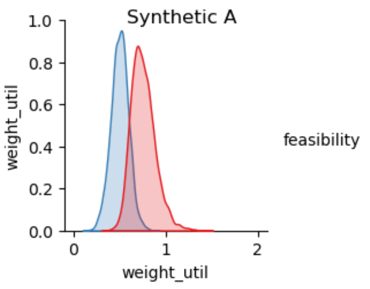
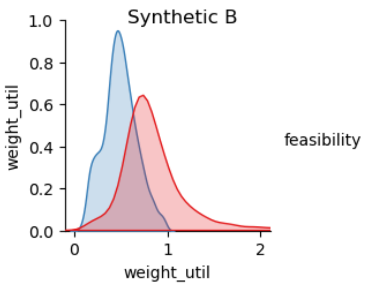


Figure A.4: Pairplot Company Experiment

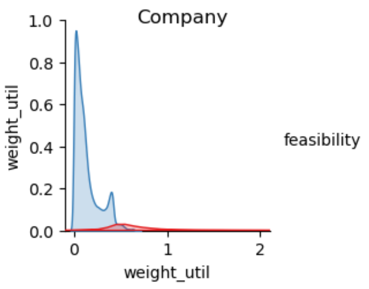
B Data Distributions per Feature



(a) Synthetic A

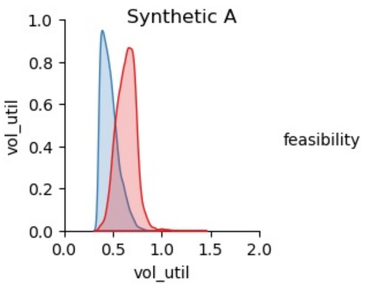


(b) Synthetic B

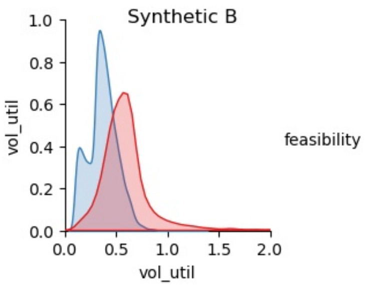


(c) Company

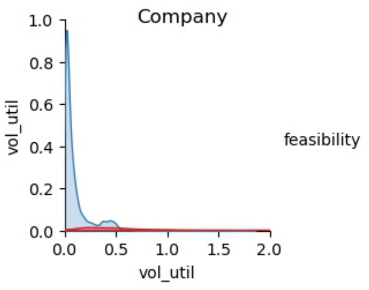
Figure B.1: Distribution of the weight utilization for all experiments



(a) Synthetic A

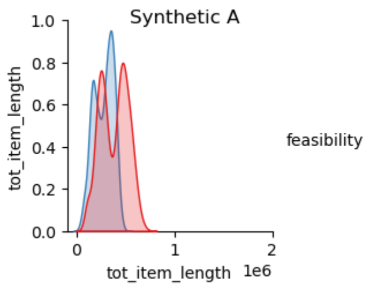


(b) Synthetic B

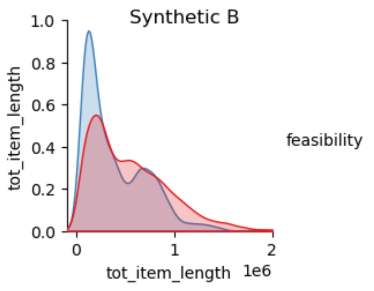


(c) Company

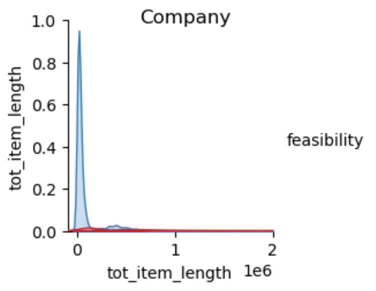
Figure B.2: Distribution of the volume utilization for all experiments



(a) Synthetic A

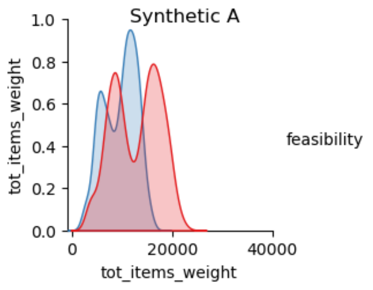


(b) Synthetic B

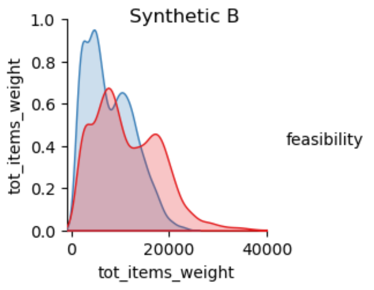


(c) Company

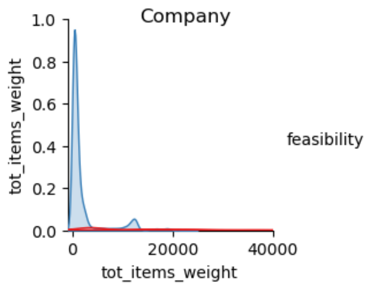
Figure B.3: Distribution of the total item length for all experiments



(a) Synthetic A

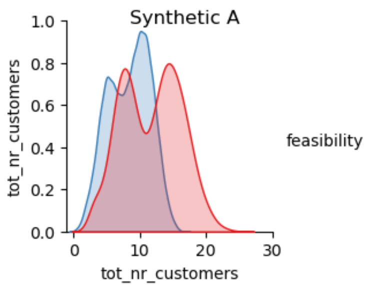


(b) Synthetic B

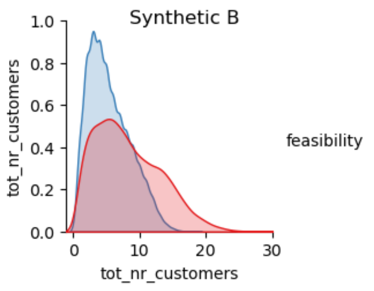


(c) Company

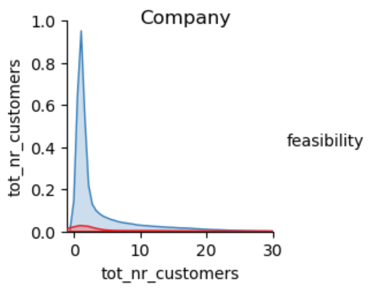
Figure B.4: Distribution of the total item weight for all experiments



(a) Synthetic A

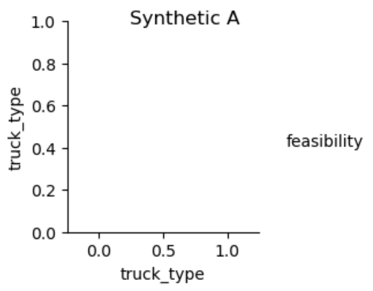


(b) Synthetic B

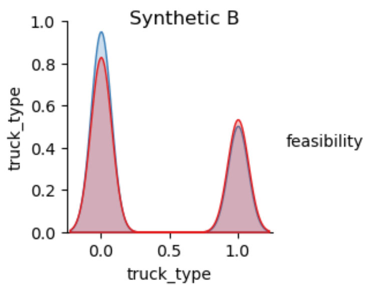


(c) Company

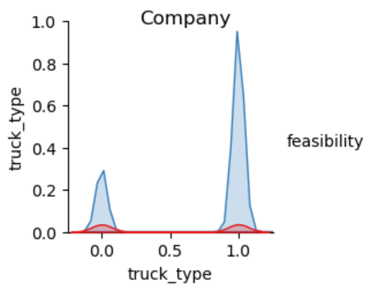
Figure B.5: Distribution of the total number of customers for all experiments



(a) Synthetic A

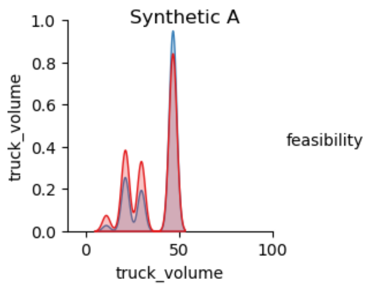


(b) Synthetic B

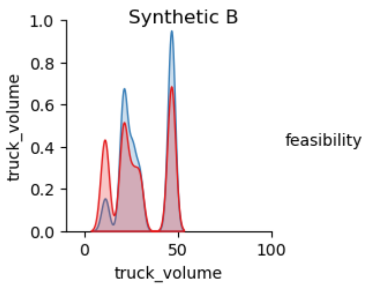


(c) Company

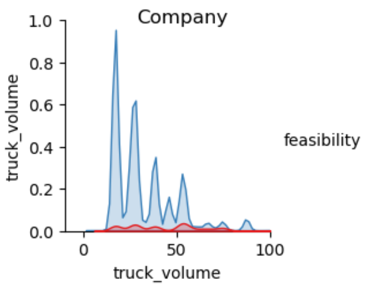
Figure B.6: Distribution of the truck type for all experiments



(a) Synthetic A

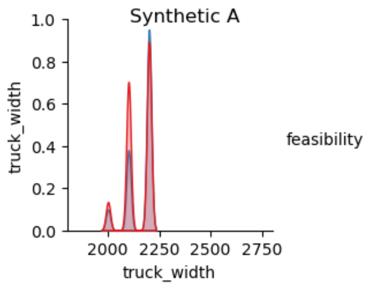


(b) Synthetic B

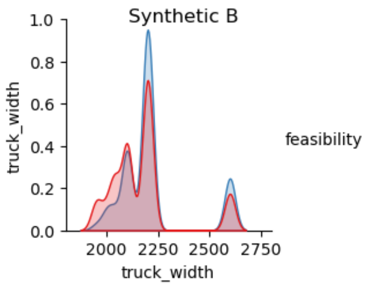


(c) Company

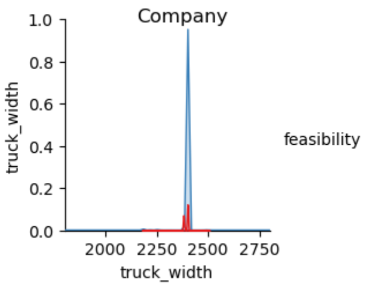
Figure B.7: Distribution of the truck volume for all experiments



(a) Synthetic A

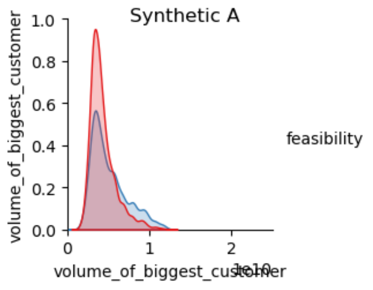


(b) Synthetic B

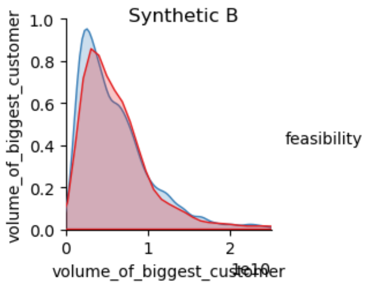


(c) Company

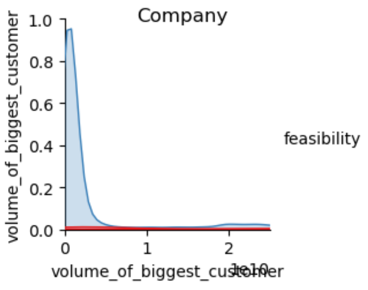
Figure B.8: Distribution of the truck width for all experiments



(a) Synthetic A

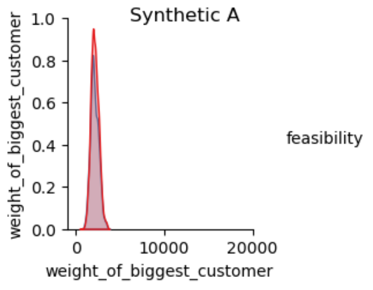


(b) Synthetic B

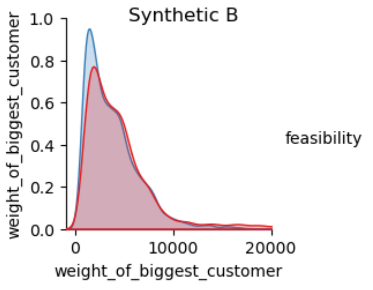


(c) Company

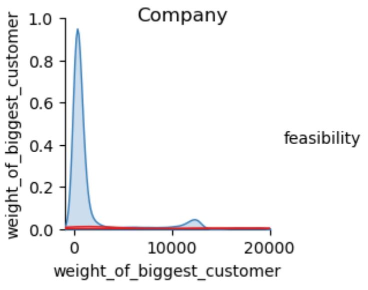
Figure B.9: Distribution of the volume of the biggest customer for all experiments



(a) Synthetic A



(b) Synthetic B



(c) Company

Figure B.10: Distribution of the weight of the biggest customer for all experiments

C

Feature Engineering

Table C.1: Feature sets and their performance outputted after performing feature selection on RF on Experiment Synthetic B

Precision	**Recall**	**Accuracy**	**CorrBound**	**NrFeat**	**Features**	**Time**
0.924	0.907	0.914	0.85	14	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height']	45.689
0.924	0.910	0.915	0.85	17	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2']	45.691
0.924	0.912	0.916	0.85	19	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36']	45.691

0.923	0.912	0.916	0.85	10	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length']	42.597
0.922	0.912	0.916	0.85	12	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items']	45.688
0.922	0.911	0.915	0.85	11	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer']	45.688
0.921	0.908	0.913	0.85	13	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight']	45.689

0.921	0.907	0.912	0.85	15	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol']	45.691
0.920	0.909	0.913	0.8	14	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol']	36.692
0.919	0.910	0.913	0.85	16	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL']	45.691

0.919	0.908	0.912	0.85	18	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type', 'tot_item_length', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40']	45.691
0.918	0.909	0.912	0.8	11	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items']	36.69
0.918	0.909	0.912	0.8	18	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36']	36.693
0.917	0.908	0.911	0.8	15	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL']	36.692

0.917	0.911	0.912	0.8	17	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40']	36.693
0.917	0.910	0.911	0.8	19	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36', 'EURO']	36.693
0.916	0.908	0.910	0.8	12	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight']	36.691
0.915	0.911	0.911	0.8	16	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2']	36.693

0.915	0.902	0.907	0.85	9	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width', 'truck_type']	42.273
0.915	0.910	0.911	0.8	13	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height']	36.691
0.914	0.900	0.905	0.85	8	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer', 'truck_width']	42.267
0.913	0.910	0.909	0.8	9	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW']	33.598
0.913	0.900	0.905	0.85	7	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'weight_of_biggest_customer']	42.267
0.911	0.898	0.903	0.8	8	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type']	33.598

0.911	0.904	0.906	0.8	10	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_LxW', 'volume_of_last_customer']	36.689
0.908	0.908	0.906	0.75	18	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36', 'EURO']	34.454
0.908	0.903	0.904	0.75	13	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol']	34.454
0.908	0.906	0.905	0.75	14	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL']	34.454

0.908	0.905	0.904	0.75	12	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height']	34.452
0.908	0.895	0.900	0.8	7	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer', 'truck_width']	33.591
0.908	0.909	0.906	0.75	16	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40']	34.454
0.907	0.908	0.905	0.75	11	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight']	34.452
0.907	0.908	0.905	0.75	17	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36']	34.454

0.907	0.902	0.902	0.75	15	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2']	34.454
0.906	0.903	0.903	0.75	19	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36', 'EURO', 'Pallet_plastic_1x1']	34.454
0.906	0.895	0.899	0.8	6	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer']	33.591
0.906	0.895	0.899	0.85	6	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume', 'volume_of_biggest_customer']	33.591

0.903	0.891	0.895	0.7	19	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36', 'EURO', 'Pallet_plastic_1x1', 'Pallet_prostack', 'Unknown_pallet_type']	34.453
0.902	0.902	0.900	0.75	10	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer', 'nr_cylinder_items']	34.451
0.902	0.902	0.899	0.75	9	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items', 'weight_of_last_customer']	34.45
0.902	0.889	0.893	0.7	11	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol']	34.453
0.902	0.899	0.898	0.75	8	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'nr_cube_items']	33.584

0.901	0.888	0.893	0.7	16	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36', 'EURO']	34.453
0.901	0.889	0.893	0.7	17	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36', 'EURO', 'Pallet_plastic_1x1']	34.453
0.901	0.894	0.895	0.7	9	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight']	34.451
0.900	0.892	0.894	0.7	10	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height']	34.451

0.899	0.889	0.892	0.7	18	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36', 'EURO', 'Pallet_plastic_1x1', 'Pallet_prostack']	34.453
0.899	0.888	0.892	0.7	13	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2']	34.453
0.899	0.887	0.891	0.7	8	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items']	34.45
0.899	0.889	0.892	0.7	15	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40', 'Pallet_36x36']	34.453
0.898	0.891	0.892	0.7	12	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL']	34.453

0.898	0.890	0.892	0.7	14	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer', 'nr_cylinder_items', 'cylinder_weight', 'truck_height', 'cylinder_vol', 'nr_HxL', 'Pallet_plastic_1x1_2', 'Pallet_48x40']	34.453
0.896	0.890	0.891	0.7	7	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type', 'weight_of_last_customer']	34.449
0.895	0.889	0.890	0.8	5	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume']	0.7462
0.895	0.889	0.890	0.85	5	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight', 'truck_volume']	0.7462
0.893	0.887	0.888	0.75	7	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width', 'truck_type']	33.583
0.893	0.886	0.887	0.75	5	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer']	33.577
0.892	0.885	0.886	0.75	6	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length', 'volume_of_biggest_customer', 'truck_width']	33.577
0.890	0.884	0.885	0.7	5	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width']	33.577

0.890	0.882	0.883	0.7	6	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer', 'truck_width', 'truck_type']	33.583
0.886	0.881	0.881	0.8	4	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight']	0.7425
0.886	0.881	0.881	0.85	4	['weight_util', 'vol_util', 'tot_nr_customers', 'tot_items_weight']	0.7425
0.879	0.876	0.875	0.75	4	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'truck_length']	0.7321
0.877	0.884	0.877	0.7	4	['weight_util', 'tot_nr_customers', 'tot_items_weight', 'volume_of_biggest_customer']	33.577
0.873	0.881	0.873	0.7	3	['weight_util', 'tot_nr_customers', 'tot_items_weight']	0.7321
0.873	0.881	0.873	0.75	3	['weight_util', 'tot_nr_customers', 'tot_items_weight']	0.7321

D

ESICUP conference certificate

EWG ESICUP CUTTING AND PACKING

UCLM Universidad de Castilla-La Mancha

18th ESICUP Meeting

11-13 May 2022, Toledo, Spain

This is to certify that

Sarah de Wolf presented

“A Machine Learning Approach for 3D Load Feasibility prediction”

by Sarah de Wolf, Ruggiero Seccia, Leendert Kok and Neil Yorke-Smith

at the “18th ESCIUP Meeting, EURO Special Interest Group on Cutting and Packing” celebrated in Toledo, Spain, from 11th to 13th May 2022.

[illegible]

Francisco Parreño

Chair of the Organizing Committe