Group #2 Project: TKL Logistics

Mahmut Osmanovic Isac Paulsson Sebastian Tuura
Mohamed Al Khaled Clement Pickel
Celian Friedrich

December 8, 2024

Contents

1 Introduction

TKL-Logistics wanted to help their customers choose smarter freight booking options. With the following quote coming from a presentation given by the company.

> "Freight bookers that work in warehouses usually book a lot of shipments every day, having little to no possibility or time to compare freight options. This results in shipments booked with little to no consideration of sustainability, price, predicted ETA, or delivery performance."

This project focuses on the predicted ETA of shipments. Being able to accurately predict delivery times is a cornerstone of customer satisfaction when it comes to the logistics industry. This project aims to address the issues of rough and/or inaccurate estimations of delivery, a problem faced by every company in these types of industries. By leveraging predictive analytics through KNIME's regression tools, we developed a solution that predicts delivery times with greater accuracy, helping the company make more accurate estimations of delivery times.

In order to structure our approach we adhered to the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework. Through the six key phases: **Business understanding, data understanding, data preparation, modeling, evaluation and deployment**, the framework created a systematic methodology for addressing the business problem.

The project lays the groundwork for future advancements in predictive analytics within TKL Logistics, with the potential to incorporate additional factors like sustainability, performance, and cost-effectiveness into the freight comparison process. This project aligns with the company's broader goal of helping their customers to make smarter, data-driven logistics decisions.

2 Method

2.1 Overview

TKL Logistics required that our solution was able to rank transportation agents based on delivery time. The approach we opted for was to create an AI model for each vehicle attribute. Each model predicts an estimated time of delivery specified in days if given an agent and input attributes. The prediction would be computed for each possible agent of delivery for the specified route. The agents are subsequently ranked in virtue of the predicted time of delivery. The agents would be ranked in ascending days of delivery, the agent which is predicted to deliver the product(s) first will be ranked first.

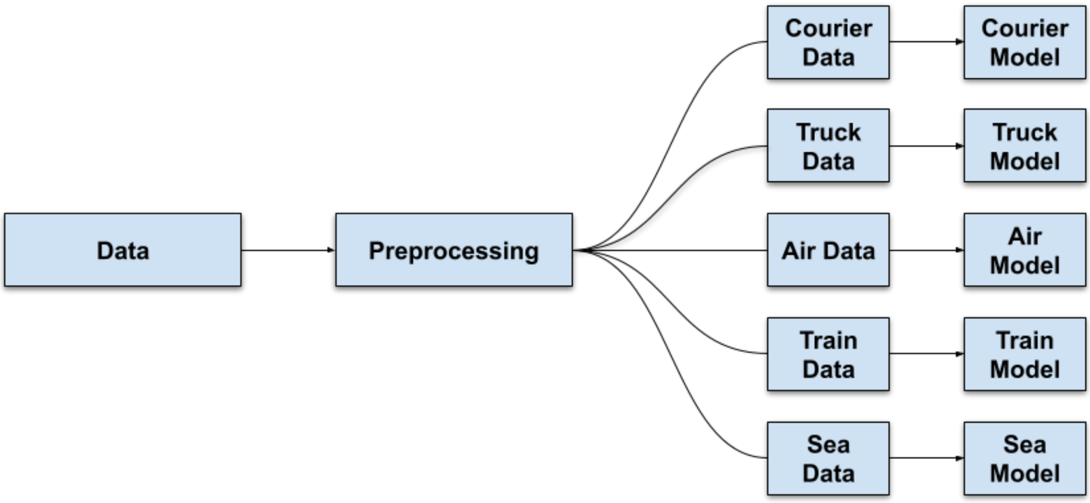

Figure 1: Final Model

2.2 The Data

The data attributes that were used in the final model are highlighted in the table below [1]. The attributes of data type string were subsequently one-hot encoded. All available regression models were tested, ultimately we found out that *Gradient Boosted Trees* Model outperformed the others. Outperformed in virtue of more or less all metrics, mean-absolute-error, R^2 and our own tolerance interval score. For each vehicle type, we also made sure to optimize its hyper-parameters to squeeze a few extra performance percentages. That was accomplished by for example increasing the amount of total ensembled models (*Gradient Boosted Trees* uses decision trees as its base learners).

Attribute Name	**Data Type**	**Description**
`FromISO`	String	From which country are goods sent
`ToISO`	String	To which country are goods sent
`TotalTransportationCost`	Number (double)	What is the total freight cost
`CO2`	String	What is the total freight cost
`MeansOfTransport`	String	Specifies means of transport
`Agent`	String	Specifies agent
`Distance`	Number (double)	Total distance for delivery
`Weight`	Number (double)	Weight of the shipment
`Agent`	String	Name of the delivery agent
`days_load_deliver`	Number (double)	Days taken to load and deliver
`Day of Week (Name)`	String	Day of the week
`Quarter`	String	Specifies Quarter of Year
`Year`	String	Year of loading
`Month (Name)`	String	Month name of loading
`Weekend`	Boolean value	Is delivery is on a weekend
`Midsummer`	Number (integer)	Midsummer holiday
`Christmas`	Number (integer)	Christmas holiday
`Easter`	Number (integer)	Easter holiday
`CN_NewYear`	Number (integer)	Chinese New Year
`CN_Mid-Autumn`	Number (integer)	Mid-Autumn Festival
`CN_NationalDay`	Number (integer)	National Day
`CostPerKg`	Number (double)	Cost per kilogram
`CO2PerKm`	Number (double)	CO2 emissions per kilometer
`Mean(days_load_deliver)`	Number (double)	Mean days by transport type
`Variance(days_load_deliver)`	Number (double)	Mean days by transport type
`Median(days_load_deliver)`	Number (double)	Mean days by transport type
`Percent(days_load_deliver)`	Number (double)	Mean days by transport type
`count`	Number (double)	Mean-monthly freights

Table 1: Attributes for Delivery Data

Some of the attributes which are listed in table 1 are *derived*. Derived attributes include the target variable (days_load_deliver), "Day of Week (Name)", "Quarter", "Year", "Month (Name)", "Weekend", "Midsummer", "Christmas", "Easter", "CN_NewYear", "CN_Mid-Autumn", "CN_NationalDay", "CostPerKg" and "CO2PerKm" and the other five aggregated attributes based on monthly patterns. The data attributes that were initially provided by TKL Logistics are listed in table 2.

Attribute Name	**Data Type**	**Description**
`LogEntryID`	String	Freight ID
`FromISO`	String	From which country are goods sent
`ToISO`	String	To which country are goods sent
`FromZipCode`	Number (integer)	Destination zip code
`ToZipCode`	Number (integer)	Destination zip code
`ForwardingAgentID`	Number (integer)	ID of the forwarding agent
`ForwardingAgentName`	String	Name of the forwarding agent
`EntryDate`	String	Date of entry into the system
`LoadingDate`	String	Date the shipment was loaded
`UnloadingDate`	String	Date the shipment was unloaded
`OriginalETA`	String	Initially estimated delivery date
`DeliveryDate`	String	Actual delivery date
`TrackingNo`	String	Shipment tracking number
`CO2`	Number (double)	CO2 emissions for the shipment
`TotalTransportCost`	String	Total transportation cost
`NumberOfPieces`	Number (integer)	Total number of pieces in the shipment
`MeansOfTransport`	String	Mode of transportation
`Agreement`	String	Associated transportation agreement
`Distance`	Number (double)	Distance of the shipment
`Weight`	Number (double)	Weight of the shipment
`Agent`	String	Name of the agent managing the shipment

Table 2: Attributes Provided by TKL Logistics

Note that there were some attributes that were not included in the final model whilst others were transformed into format which was of greater utility (are derived). Specifically, "ForwardingAgentID", "ToZipCode", "EntryDate", "LoadingDate", "UnloadingDate", "DeliveryDate", "OriginalETA", "TrackingNo" and "Agreement". The target variable was derived as the difference in days between "LoadingDate" and "DeliveryDate". We did not use the difference between "EntryDate" and "Delivery" date since that difference is not the actual shipment time. The real shipment time is what we are interested in predicting, which is also what we extracted and dropped the other date related attributes. There are thousands of zip codes. For any model to distinguish between the zip code would be incredibly difficult. We reasoned that the information which it possibly could add to the model would also be approximated well enough by the union of three others. The three others are *FromISO*, *ToISO* & *Distance*. Specifically, the country of origin, the country to which the freight should be delivered and the distance. Nonetheless it is true that it misses some information, especially freights within a countries borders. However, we did not see any increase in accuracy when including postal codes. That is very likely due to the fact that the model does not contain enough of data points to meaningfully discriminate among freights with different postal codes. The *OriginalETA* had values filled in for less than 1% of the data hence was rendered useless, we dropped it. Tracking number and Log Entry ID similarly contains too many classes for our model to meaningfully differentiate between them in virtue of the provided dataset. *Agreement* contained information about which agent was responsible for that specific freight, however that information is also provided in the attribute *Agent*. Hence the attribute *Agreement* is also rendered obsolete and subsequently dropped.

2.3 Preprocessing

We noted that *NumberOfPieces* were distributed in an interesting manner, as exponential decay. We also noted that there did exist a small positive correlation between it and the target variable, hence it was of interest. However, there were too many possible number of pieces. To deal with the problem at hand, we binned the values in almost exponentially growing interval values (number of pieces) for the bins. For visual inspection, see figures 2 and 3.

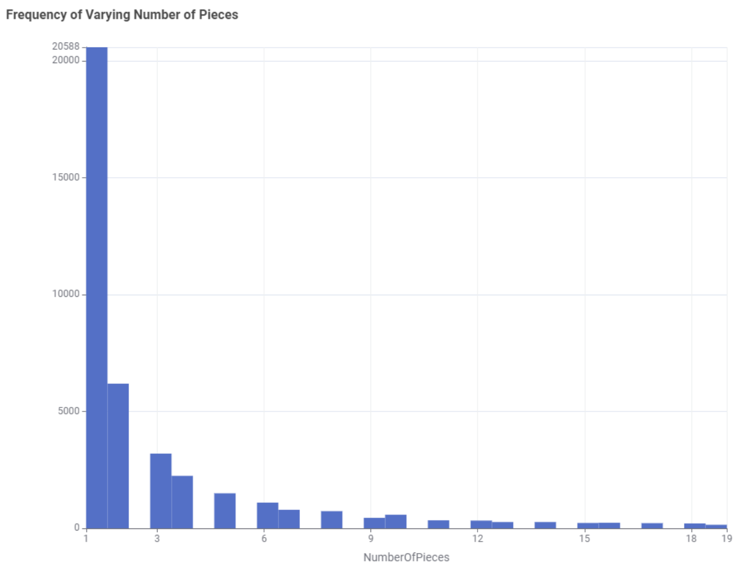

Figure 2: Histogram: Number of Pieces

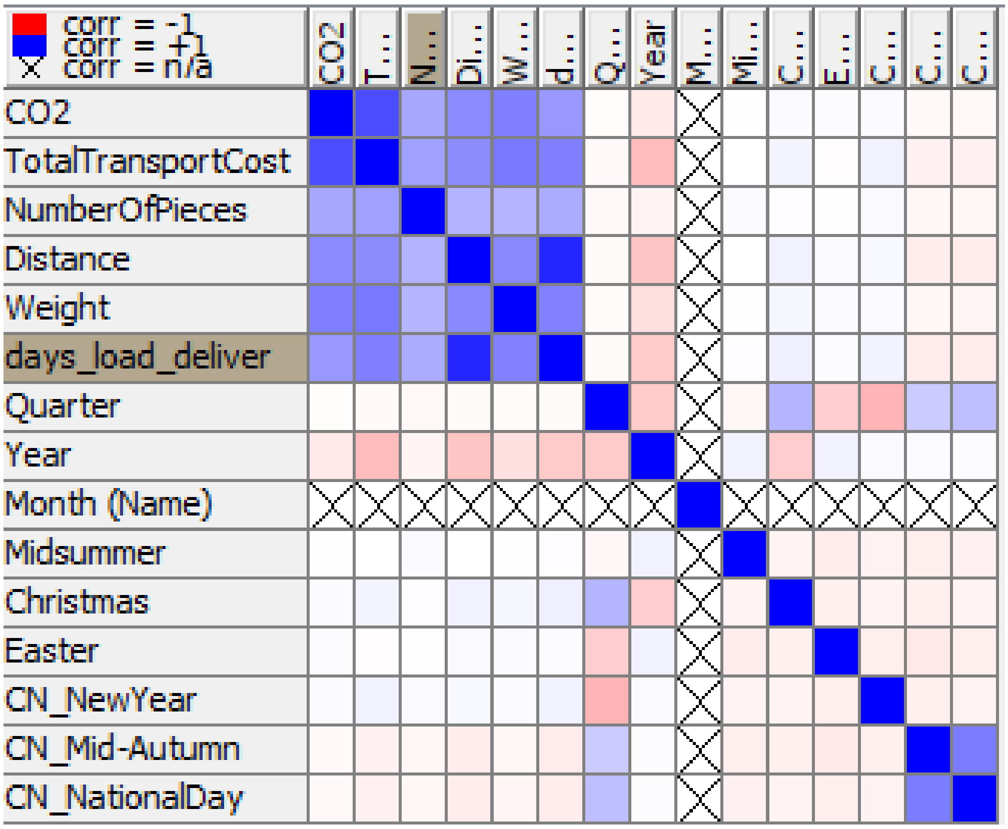

Figure 3: Linear Correlation - Number of Pieces

Although some of the attributes may shared a substantial amount of information with each other (weight and CO_2), they are not identical and all exhibit positive correlation with the target variable. The rest of the derived attributes can broadly be classified in three category types. They are either extractions relating to date based on day of entry, or derived attributes based on holidays in the most common countries which freights travel from and to. Lastly there are past monthly aggregates of say *Mean(days_load_deliver)* over the available past years for each means of transport type. The first category is meant to capture patterns which may exist on, say, a yearly or quarterly basis. Say that the agents are getting more effective at delivering goods at a fewer amount of days, that would then be captured. Another attribute we derived was the *Weekend attribute*, if the freight is scheduled during a *Friday*, *Saturday* or *Sunday*, that is noted with "true", otherwise "false". Scheduling a freight just before a weekend almost guarantees that the total freight time will be larger than if booked during other working days. The holiday attributes are supposed to track patterns in freight delays. The most common countries that packages were delivered from were China, Sweden, Norway, Finland, Taiwan and

Denmark. A similar pattern holds for the country which the goods are shipped to. The most common holidays in these countries include Midsummer, Christmas, Easter, Chinese New year, Chinese National Day and Chinese Mid-Autumn Festival. The specific date interval for which these holidays approximately have occurred over the last decades were analyzed and appropriate intervals were chosen. The intervals were chosen a few days before and after the actual beginning and ending of the holidays. That is as to account for winding up and winding down times. All string attributes were *one-hot encoded*.

We noted that the courier attribute contained a strange range of frequency of distances.

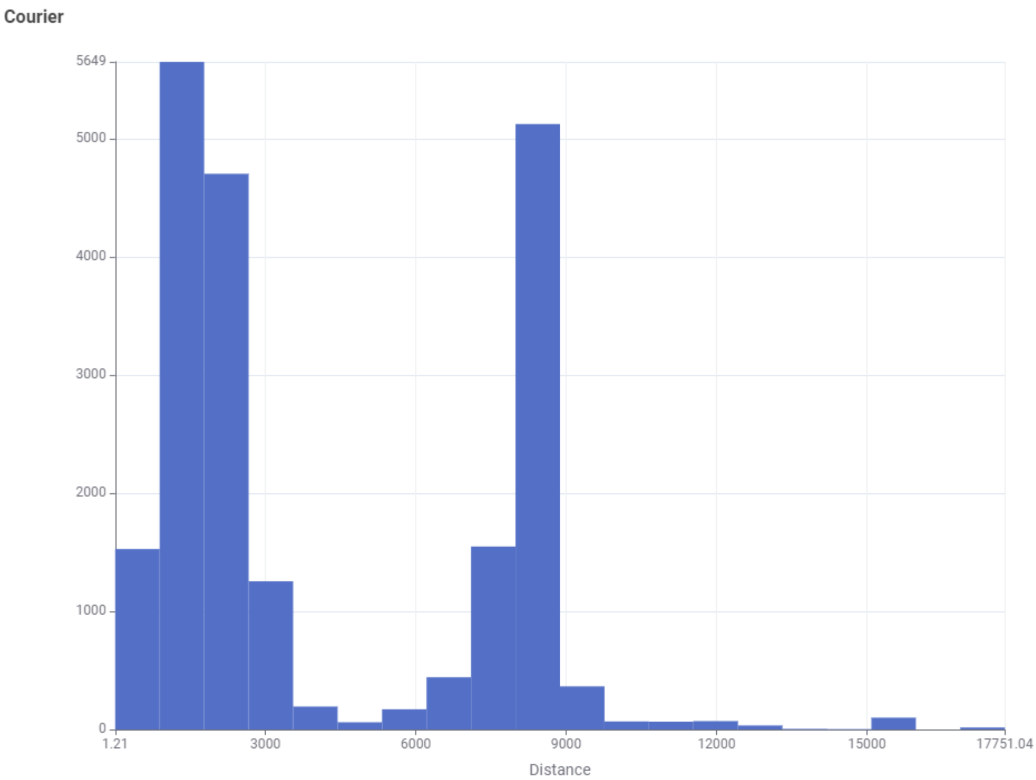

Figure 4: Histogram: Courier vs. Distance

TKL informed us about that the attribute actually also contains freights transported by *Air*. That occurs when a regular say, package-van or UPS truck could not reach its destination in appropriate timing. We cut out the second chunk of data, specifically between 7000 and 9000 kilometers and instead joined it with the data that belongs to freights by airplanes. This may also be processed live as a customer is booking their freight (in virtue of distance).

Lastly the data was cleaned. Outliers were analyzed with box-plots and histograms. They were not included as they are not expected to be part of the actual distribution but affected by unique irregularities that are not able to be capture by our models. However, we were careful about not excluding too large chunk of the data. Excluding a disproportionate amount of data would decrease the interval in which we actually will be evaluating our model up to. All the data which our model is not trained on will be data it must extrapolate upon. Given that our model is not over-fitted on our data, that may not be a problem. However, if the underlying distribution in extrapolation space differs from the space that the model is trained on, its reliability shrinks. Thus it is of vital importance to have a balanced training set. It should be large, but not large to the extent that it begins fitting noise. See figure 5.

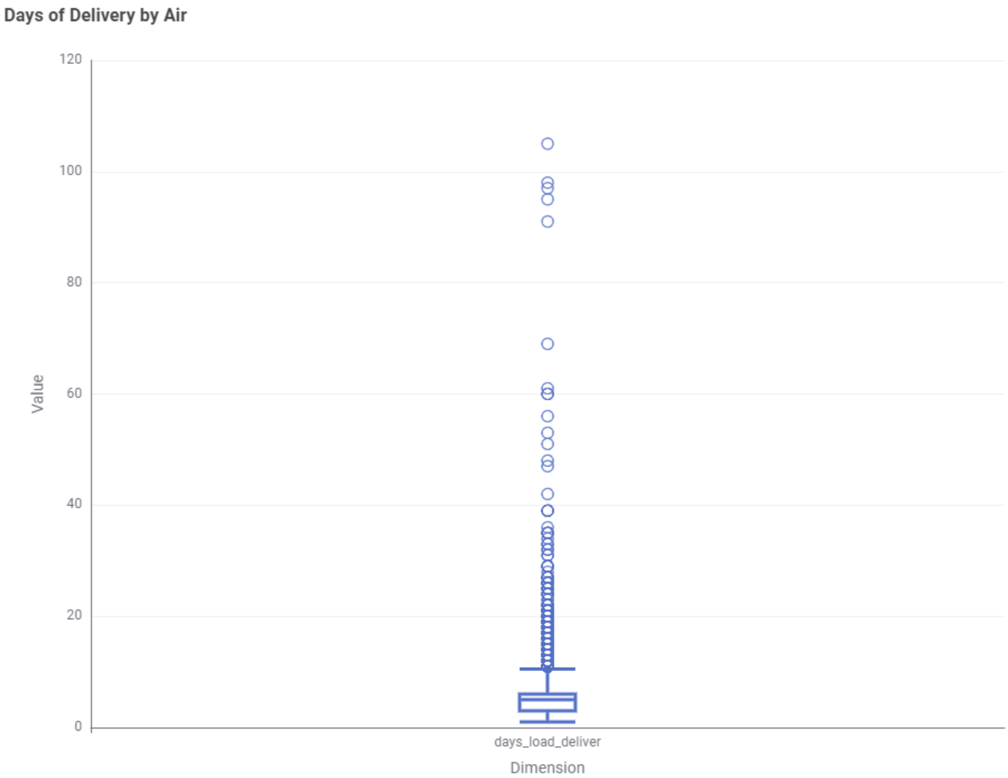

Figure 5: Days of Delivery by Air

There are over 10000 data points of freights transported by airplane. By visual inspection of figure 5, there are a couple of hundred of substantial outliers. The box plots upper bound is 10.5 days, we included freights up to 11 days. A similar analysis was done for each means of transport.

We noticed that the days of delivery appeared to be heteroscedastically distributed. To ensure that the model is not biased or dominated by outliers we applied a log transformation on it. It is supposed to ensure that variability is reduced, allowing the model to create more balanced splits and hence improve prediction accuracy. After training, the target variable is transformed back into its initial state.

We Z-normalized all numeric attributes since the numerical data varies in magnitude. The reason is likewise to minimize the risk of attributes with large numerical values to dominate the model.

This gives equal credence to all potential agents within each of the five models, we made sure that all possible prediction classes (distributions) contained an identical number of data points. If all agents with a low amount of attributes were bundled into a "REST" category. All categories were randomly sampled from in virtue of the lowest common dominator. The lowest common dominator was the category with the least amount of data points that was contained in it. Thus we used a under-sampling technique, as we reduced the size of the majority class(es). There was an option to over-sample, however, we did not want to risk changing the underlying distributions to something which they may not be. Figure 6.

count *Number (integer)*	**Agent** *String*
5821	DHL EXPRESS (Sweden) AB
1189	GLA
1980	REST
589	UPS

Figure 6: Air Categories - UPS is the Lowest Common Denominator

2.4 The Models

After the data is preprocessed, we feed it into the model, more precisely, the *Gradient Boosted Tree Learner*. The actual model structure can be inspected in figure 7.

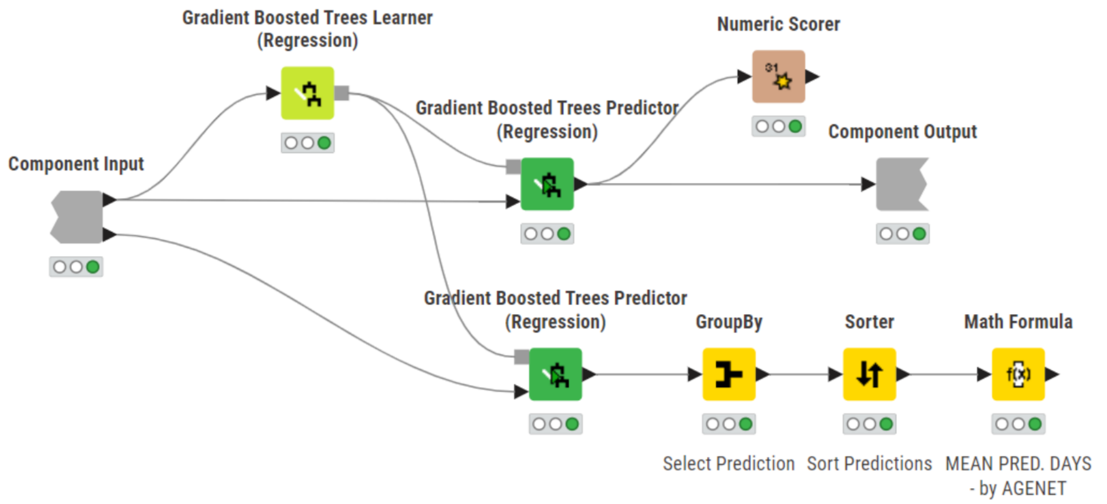

Figure 7: Air Model

The actual predictions can be seen in figure 16. All of the models have a similar structure.

2.5 Evaluation Metrics

There are several ways in which we have evaluated our models by. In the beginning we evaluated them by the R^2 value. In the final model, we looked at three core metrics. One was the *MAE* (mean absolute error), another was *MAPE* (mean absolute percentage error) as well as our own performance metric.

The absolute mean error is generated by:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\text{PredictedDays}_i - \text{ActualDays}_i|$$

.

The metric informs us about how many days on average our predictions are wrong in relation to the actual amount of days the freight took. A lower value is better as it indicates a tighter fit. However, it is a mean and although the model might perform well on average, there may be instances where it performs worse.

Another useful metric is "mean absolute percentage error", defined as:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{PredictedDays}_i - \text{ActualDays}_i}{\text{ActualDays}_i}\right| \times 100$$

It informs us of how wrong our model is on average as a percentage. For example, in the sea model, it is on average over or underestimating the delivery time by 9.3%.

Our own metric is one which classifies how many of the points that we predicted fell into a set bounded interval. Specifically, it sets a lower bound

$$\textbf{lower\_bound} = \text{actual\_days} \times (1 - \frac{k_{perc}}{100}) - m_{days}$$

and

$$\textbf{upper\_bound} = \text{actual\_days} \times (1 - \frac{k_{perc}}{100}) + m_{days}$$

Points within the bound are colored green and those outside of the bound are colored red. We could have used a more complicated function to express the bounds, however, we thought that the two constants expressed a sufficient amount of complexity to capture most of the variability within the dataset. The tighter we can make this interval the better. Ideally both k_{perc} and m_{days} would equal 0, that would mean that the predicted values are identical to the actual values. The m_{days} sets the interval to be at least $2 \times m_{days}$. This is of vital importance when the means of transport have a lot of points that fall within just a few days of delivery, otherwise they would have been categorized as errors. The k_{perc} increases the allowed tolerance of error as the amount of days grows, which aligns well with the heteroscedasticity of our data. Our aim is to make each bound as tight as possible whilst ascertaining a performance score of at least 80%.

Scorer View

Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	2120	236	89.98%
1 (Actual)	0	0	undefined
	100.00%	0.00%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classifi
89.98%	10.02%	0.000	2120

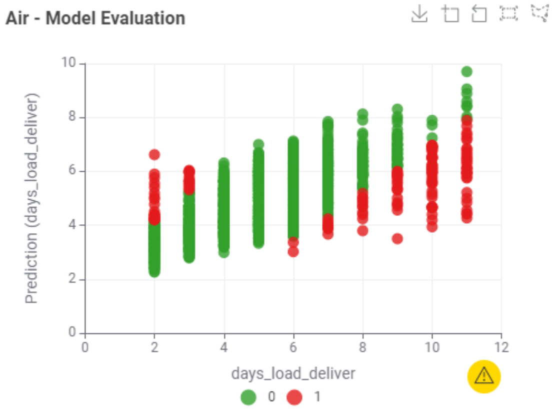

Figure 8: Air Model - Evaluation

3 Results & Analysis

3.1 Model Results

Our first model attempted to predict delivery time across the different modes of transport, this initially gave promising results in terms of R^2 with a value of around 0.9 which would be considered a good representation. After dividing the model by means of transport the R^2 value of our prediction on the individual cases was closer to 0.2. This leads into a flaw with evaluation by R^2 where the value is inflated based on the distribution of delivery times between different modes of transport. From this we decided to look for other performance metrics.

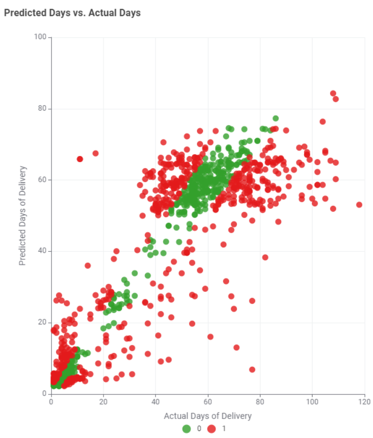

Figure 9: Error bounds for model with all Modes of Transport

RowID	Prediction (days_till_delivery) Number (double)
R^2	0.889
mean absolute error	4.45
mean squared error	63.024
root mean squared error	7.939
mean signed difference	-0.3
mean absolute percentage error	0.612
adjusted R^2	0.889

Figure 10: R^2 Metrics, all Modes of Transport

Attempts were also made to divide the problem further with a model for each agent based on the three most common agents and an additional model for the rest. To evaluate the individual models to the general model box plots were made to compare the distributions of the predicted values. With matching distributions and negligible accuracy differences we decided that dividing the model on Means of Transport was sufficient.

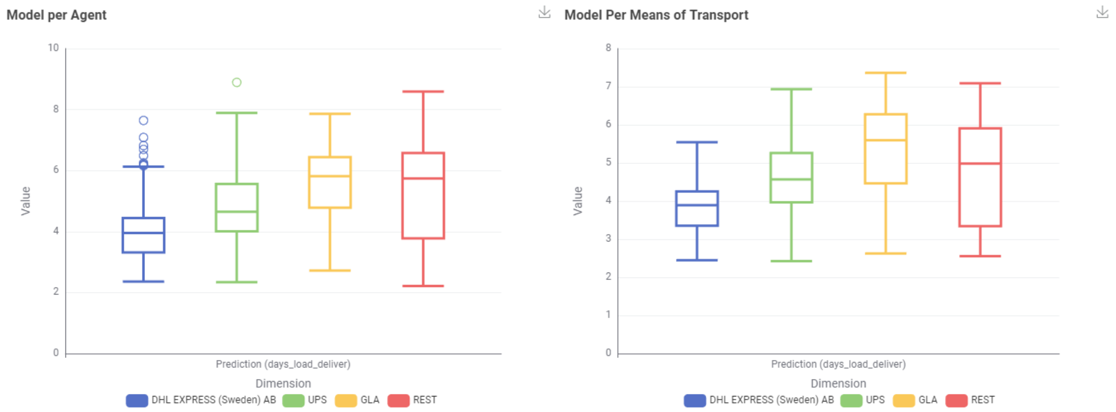

Figure 11: Enter Caption

To grade model performance the three following metrics were chosen, Mean absolute error (MAE), Mean Absolute Percentage Error (MAPE) as well as our Error bound. The metrics are further explained in section 2.5.

As a baseline for mean absolute error we have used the general values provided for the different modes of transport, listed below.

$$[Air : 7, Courier : 2, Truck : 5, Train : 25, Sea : 50]$$

In addition to this we have calculated a mean of actual delivery time per agent in each category. These values are used as an alternative prediction. Mean absolute error compared to the actual delivery time is calculated for each alternative and is found in figure 14.

Model accuracy in the specified error bounds is tabulated in figure 13. This also includes an accuracy for the first model containing all modes of transport as a comparison.

Mean Absolute Percentage Error is tabulated in figure 15.

Means of Transport *String*	Percentage *String*	Days *String*
Air	10	2
Courier	15	1
Truck	10	1
Train	20	0
Sea	10	3

Figure 12: 80% accuracy bounds per Means of Transport

Means of Transport *String*	Accuracy (Model per Means of Transport) *Number (double)*	Accuracy (Large Model) *Number (double)*
Sea	0.83	0.629
Train	0.913	0.75
Truck	0.87	0.525
Courier	0.828	0.593
Air	0.9	0.718

Figure 13: Accuracy Error Bound

Means Of Transport *String*	Prediction (MAE) *Number (double)*	Mean per Agent (MAE) *Number (double)*	Original Estimate (MAE) *Number (double)*
Sea	4.581	6.212	6.706
Train	2.464	3.806	4.433
Truck	0.867	1.103	1.114
Courier	0.942	1.417	1.69
Air	1.213	1.733	2.509

Figure 14: Mean Absolute Error

Means of Transport *String*	Mean Absolute Percentage Error *Number (double)*
Sea	8.872
Train	8.639
Truck	15.094
Courier	34.359
Air	25.939

Figure 15: Final model Mean Absolute Percentage Error

The model provides an improvement in accuracy compared to the static general values as well as to the mean per agent in the Sea, Train, Courier and Air categories. For deliveries by truck the improvement in accuracy is negligible.

While the model generally does provide an improvement over both static values as well as mean per agent the model performance is still not optimal as seen in the bounds, figure 12 and MAPE evaluation, figure 15. The lack of accuracy probably lies in a lack of data for good indicators of delivery time. There is an abundance of variables that go into the actual shipping time that we can not derive nor proxy from the provided data leading to problems with model accuracy.

Some possible indicators that can not be derived from provided data could include, shipment scheduling, precise routing, port and route congestion, number of stops, variability in loading times, weather conditions, route conditions. Parts of this data could possibly be collected from the shipping agents to improve model accuracy.

The models could be used to make a prediction of delivery time based on agent. Example in figure 16. Here a prediction has been made for identical shipments with the difference being what agent performs the delivery. This could in turn be used to give a ranking suggestion for what agent might make the fastest delivery.

Predictions for shipments by train can not reasonably be estimated by agent since the category almost exclusively lists "TKL EGEN" as agent.

Agent *String*	Prediction (days_load_deliver) *Number (double)*
DHL EXPRESS (Sweden) AB	3.943
UPS	4.567
REST	4.774
GLA	4.967

Figure 16: Example prediction (Air)

4 Appendix

The CRISP-DM framework played a pivotal role in structuring our approach. With its six key phases it provided a us with a systematic methodology that ensured a clear, logical workflow and facilitated effective collaboration within the group.

- **Business understanding**
  The first phase helped align the group's goals with the company's needs. By analyzing the business case & understanding the challenges faced by freight bookers.
- **Data understanding & Preparation**
  These two phases helped structure the technical work. Guiding us in understanding the contents in the datasets as well as addressing any gaps or inconsistencies. Which in turn enabled a strong foundation for the analysis.
- **Modeling & Evaluation**
  The framework's focus on iterative modeling and evaluation enabled us to experiment with different regression techniques using KNIME's tools. The structured evaluation process ensured that we selected the most effective model based on the accuracy and reliability of ETA predictions. This iterative approach also allowed for regular team discussions and refinements, fostering a collaborative environment.
- **Deployment**
  This phase allowed us to think about he practical implications and how our solution could be integrated into the TKL Logistics' operations

Overall, the CRISP-DM framework contributed to the success of our project. It ensured that our efforts were directed and coordinated, minimizing inefficiencies and miscommunication.

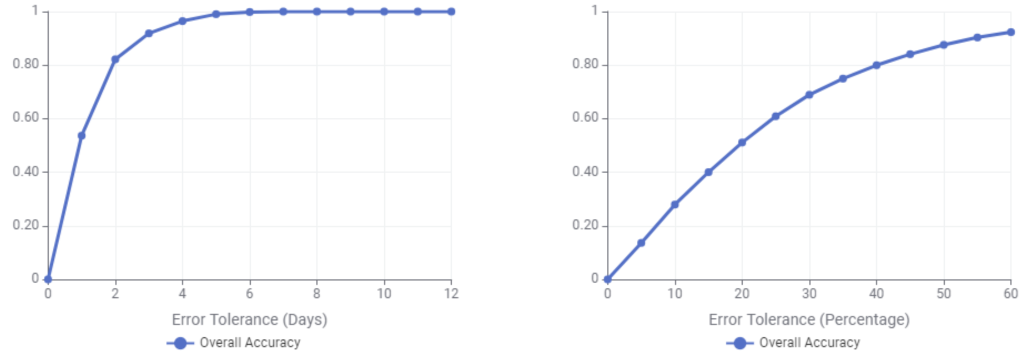

Figure 17: Air Bound Influence

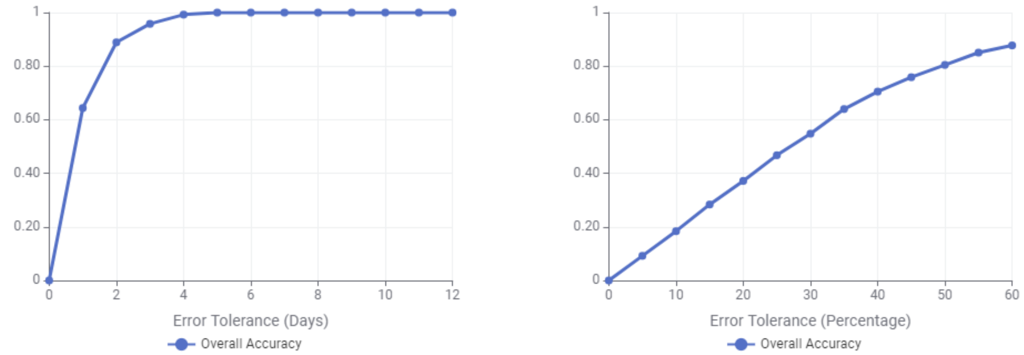

Figure 18: Courier Bound Influence

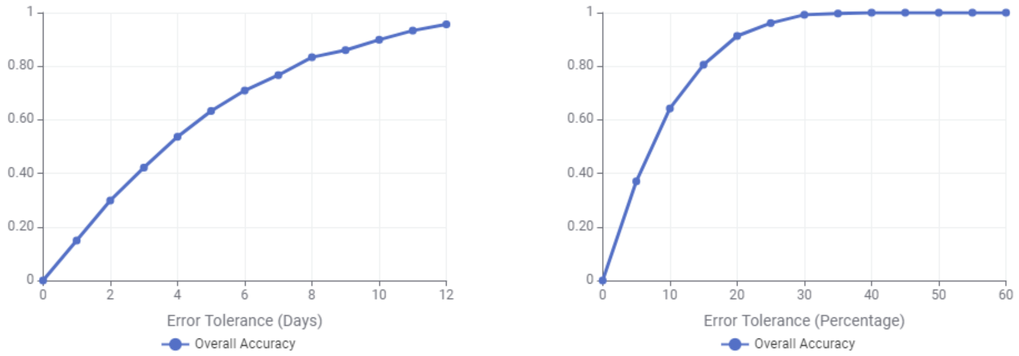

Figure 19: Sea Bound Influence

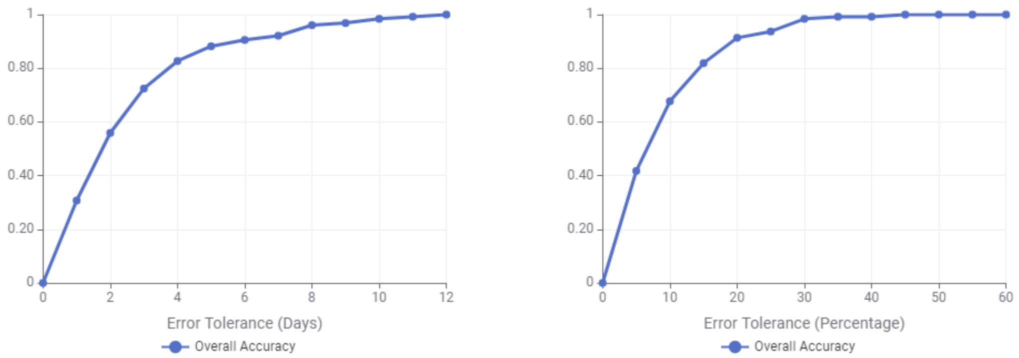

Figure 20: Train Bound Influence

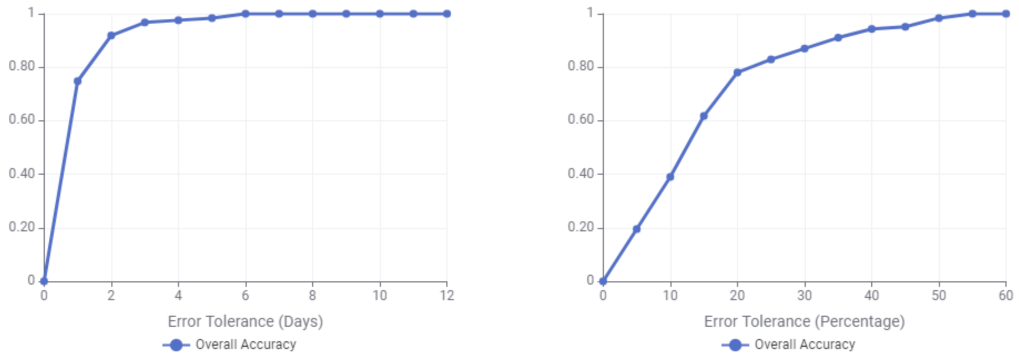

Figure 21: Truck Bound Influence

References