UPPSALA
UNIVERSITET

UPTEC STS 25009
Examensarbete 30 hp
Juni 2025

Machine Learning for More Efficient Traffic Flows

End-Time Predictions for Accidents

Hugo Asztély & Gabriel Martens

Civilingenjörsprogrammet i System i Teknik och Samhälle

UPPSALA
UNIVERSITET

Machine Learning for More Efficient Traffic Flows

Hugo Asztély & Gabriel Martens

Abstract

Road traffic accidents pose significant problems on Swedish roads, not only in their own devastating nature for the people involved but also because they tend to disrupt traffic flow. *The Swedish Transport Administration*, Trafikverket, actively works towards decreasing the negative consequences of accidents through coordinating traffic and communicating information to SOS, road assistance and other personnel in order to efficiently handle the incidents. Road traffic operators in Stockholm and Skåne publish announcements of accidents, in which they describe the type and location of accidents as well as give an end-time estimate, for when they think the state of the road and traffic flow will return to normal operation. The accident data is registered into a database called *Nationellt Trafikledningssystem*, NTS, where an array of information is stored for each accident, together with mentioned end-time estimate. This thesis explores the possibility of using predictive analysis, machine learning, in the task of predicting traffic accident end-times, based on historical road traffic accident data. The goal is to make estimates that outperform the manual ones set by the road traffic operators and hopefully prove that such techniques can be used in the future. Through preprocessing, NTS and meterological data were merged and transformed for a wide range of machine learning models with the top performers being Extreme Gradient Boosting, XGBoost and Support Vector Regressor, SVR. The thesis concludes that it is possible to successfully use machine learning models to predict end-times for accidents on the roads of Stockholm and Skåne, while also outperforming the ones set by Trafikverket. It is however important to consider the complex system of road traffic. Some accidents' end-times are more important to predict than others depending on time, place and severity. Suggested improvements include the use of more detailed attributes, more descriptive of the actual accidents, as well as better quality control for NTS data registration. The results of this thesis can be seen as a proof of concept and an assistive tool rather than an applicable method.

Teknisk-naturvetenskapliga fakulteten
Uppsala universitet, Utgivningsort Uppsala

Handledare: Beatrice Fritz Ämnesgranskare: Lars Oestreicher
Examinator: Elísabet Andrésdóttir

Populärvetenskaplig sammanfattning

Trafikolyckor på svenska vägar är ett stort problem, både för de drabbade men även för trafikflödet som ofta störs av sådana händelser. Trafikverket i Sverige strävar hela tiden efter att minska de negativa effekterna av olyckor genom att samordna trafiken och kommunicera med SOS, vägassistans och allmänheten för att hantera situationen effektivt. Vägtrafikoperatörer i Stockholm och i Skåne meddelar allmänheten via olika kanaler om en olycka skett. Informationen innehåller vanligtvis en kort beskrivning av olyckan, position och en så kallad sluttidsprognos för när trafikflödet förväntas återgå till det normala. Denna data registreras i en databas som kallas *Nationallt Trafikledningssystem*, NTS.

Trafikverket vill undersöka om sluttidsprognoserna går att förbättra. Denna uppsats undersöker om maskininlärning som prediktivt verktyg går att använda för att estimera sluttider baserat på historiska olycksdata. Målet är att göra mer exakta prognoser än de tidigare manuella som trafikoperatörerna anger. Genom att bearbeta NTS och meteorologisk data har flera maskininlärningsmodeller tränats.

Resultaten visar att maskininlärningsmodeller går att använda på ett framgångsrikt sätt för att estimera sluttider för trafikolyckor i Stockholm och Skåne. Utöver det producerar modellerna mer precisa estimat jämfört med Trafikverkets. Det är dock viktigt att förstå hur komplext ett vägtrafiksystem är och hur dynamiska trafikolyckor är. Vissa olyckor är mer kritiska än andra beroende på vad som hänt, var det hände och när det hände. Detta är något som de mänskliga operatörerna kan ta hänsyn till men inte en modell, vilken generaliserar verkligheten. Förslag på framtida förbättringar är attribut som bättre beskriver den faktiska olyckan och att fokusera på standardiserad kvalitetskontrollerad datainsamling till NTS. Resultaten kan ses som ett konceptbevis och ett assisterande verktyg, snarare än en färdig metod redo för implementering.

Acknowledgment We want to thank our supervisor Beatrice Fritz for the support during this thesis project. We also wish to thank Sweco and Trafikverket for generously sharing their expertise, which greatly enriched our work. Besides them, we also want to thank our subject reviewer Lars Oestreicher for guiding us in the right direction during the process of writing this thesis. Lastly, we extend our heartfelt gratitude to our examiner, Elísabet Andrésdóttir, for her unwavering support and dedication to our studies over the past five years.

Contents

1 Introduction

Swedish roads frequently suffer from sudden incidents, that directly or indirectly impact road users' plans, emergency response efficiency, and public transport, often resulting in traffic congestion. *The Swedish Transport Administration*, (Swedish: Trafikverket), actively works towards maintaining efficient traffic flows. When an accident inevitably occurs, road traffic operators quickly try to manage the situation and restore traffic flows to normal. This work includes communicating information of the accident to the public, as well as give an estimate for when traffic flow is expected to return to normal [1]. From this estimate, road users can adapt their travel plans accordingly and choose alternative roads to reach their destination, potentially avoiding a situation where a major congestion can occur. Currently, the estimates are based on experience and a simple mathematical formula. The experiences from prior incidents are important for providing an accurate estimate, while the formula is only a vague indicator, as it is based on a historical mean accident duration [2]. All data is collected in a database called *Nationellt Trafikledningssystem*, NTS, where 43 attributes for each accident are stored.

This thesis explores the possibility to improve the end-time estimation method with the use of machine learning. This is applied onto the NTS data, which is combined with weather data collected from *The Swedish Meteorological and Hydrological Institute (Sveriges Meteorologiska och Hydrologiska institut, SMHI)*. By improving the estimates, road networks could be used more efficiently, which could result in fewer and smaller congestions, less fuel emission, less negative impact on critical public services, and shorter travel times. By conducting interviews with operators and visiting the road traffic control center in Stockholm, a deeper insight in the subject has been obtained. These highlighted the importance of understanding the complexity of the concept of road traffic control, which was critical in order to develop a machine learning model that could support it. Further, this thesis aims to problematize and discuss how the current method for accident data collection at Trafikverket can be improved, with standardization and categorization of data, which could improve future research involving data analysis. In addition, this thesis investigates similarities and differences between applications from previous research and suggests future improvements.

Section 1 thoroughly explains the background, purpose, and previous research for this thesis. The following Section 2 focuses on the data and presents the structure and collection of the data used. Section 3 provides a more in-depth explanation of applied machine learning models and how they work. This section is not necessary to read unless the reader has a deeper interest in the mathematical breakdown of machine learning. Section 4 discusses how the data is handled, that is, how the original data together with additional data is merged, transformed, selected, filtered, and cleaned. This section also explains how the data is applied for machine learning purposes. The final three sections

5, 6, and 7 present the results of this thesis and the discuss all the issues encountered as well as give suggestions for future improvements.

1.1 Background

In Sweden there are four road traffic control centers that operate around the clock every day of the year, gathering information of incidents and overall traffic conditions on the roads and communicating it to the public. The four centers are located in the cities of Gävle, Stockholm, Göteborg and Malmö, as seen in Figure 1, where all operate under *Trafikverket*. Note that the focus for this thesis lies on Stockholm and Skåne counties. The center located in Stockholm is, besides Trafikverket, associated with *Stockholm Stad* and *Nacka Kommun* and operates in the counties of Stockholm, Gotland, Örebro, Östergötland, Uppsala, Södermanland, and Västmanland, with more than four million Swedish inhabitants. The southernmost central in Malmö covers the counties of Skåne, Kronoberg, Blekinge, Kalmar, and Jönköping. Their main purpose of operation is to ensure safe and efficient traffic flows, as disruptions in the road network can result in high costs to society. Among their goals are publishing information on unplanned and planned incidents that impact traffic flows, prioritizing public transport for incidents with high impact, as well as ensuring reliable travel times [1].

Figure 1 Road traffic control centers in Sweden in their respective operating zones [1].

However, although anybody can contact the road traffic control centers, most frequently their information comes from SOS personnel, entrepreneurs, sensors and cameras. A road

traffic operator verifies and registers an incoming incident in the NTS database. Depending on the information the operator provides the NTS, the system automatically generates an action plan. For example, if a major traffic accident, such as a fire, occurs in a tunnel, the NTS could suggest to close the entrance for all traffic but emergency vehicles. Other outputs of information from the road traffic control centers go out to local media such as radio, TV and websites like trafiken.nu. The information available to the public regarding an incident contains the location, type of incident, brief description of the incident, and the source with a corresponding start-time and an end-time estimate [1].

Since it is very difficult to estimate an end-time directly based on initial information, the road traffic control centers aim to set a secure estimate when a so called *Secure Lead Time* has been reached. In central Stockholm where road assistance is involved, the lead time is 12 minutes, while for the rest of Stockholm and Skåne, the lead time is 20 minutes. The internal goal at the road traffic control center in Stockholm for end-time estimates is to be within +-30 minutes of the actual end-time for at least 80% of all reported incidents, whereas in Skåne no such goal exists. For accidents in Stockholm, the end-time estimate is automatically set to 38 minutes after the start-time. This value is an estimate based on the mean duration of incidents calculated for some amount of years [2]. In Skåne, an estimate of 63 minutes is automatically pre-registered [3]. The individual operator can edit the estimate but as seen in Section 4.1, 38-39 minutes is by far the most common estimate for Stockholm. In Skåne a similar trend can be seen, but for the estimate of 63 minutes. Factors leading to the operator increasing an estimate are, for instance, if SOS is involved or if the incident involves heavy vehicles. The operator is also responsible for updating the incident record as the situation develops, for instance, editing the description, adjusting the end-time estimate, and registering or updating other missing or existing information. When the record is updated, a new record with the same ID and a new version number is created. As this procedure continues, newer versions give a more complete picture of the incident [4].

For incidents occurring on roads equipped with camera surveillance, it is straightforward for the operator to know when to start and close an incident, it is significantly more difficult when something happens outside the city centers. In this case, the operator relies solely on external information, often derived from SOS and road assistance personnel. If no further information is communicated as the estimated end-time is reached, it is common for the operator to end the incident [4].

If an incident is not closed or updated within 15 minutes of the estimated end-time, the operator receives a notification to update or end the incident. The operator also receives a notification as the estimated end-time is reached, leading to many incidents being closed at the same time as the estimate. This results in correct estimates despite the fact that information of the incidents' actual durations could potentially be false. This is depicted in Figures 6 and 7. The correct estimates can be partly explained from the

lack of information, as described in the paragraph prior to this, but also since some of the incidents are subjectively close enough to be regarded as finished [4].

Traffic flow is characterized by its rapidly changing and dynamic nature. Two disrupted lanes on the urban highway *Essingeleden* in Stockholm is estimated to cost approximately 12 000 SEK per minute, paid indirectly by the road users. This cost estimate considers only the actual delays that impact the affected road users [1]. The cost is an underestimate, as other factors, such as prolonged fuel consumption and the fact that accidents during rush hours tend to cause similar congestions in the opposite direction as well, as curious spectators slow down significantly to observe the situation [4]. As traffic flow conditions tend to change rapidly when accidents occur, traffic control operators want to produce as correct estimates as possible. The incidents often result in congestions and it is of great importance to communicate both quickly and correctly to the public how the road conditions have changed and when they are expected to return to normal. An estimate that is too high could mean that road users are able to reroute their journeys and use smaller, longer roads to get to their destinations. This is most often only a minor problem and, therefore, the preferred outcome if the incident is estimated to disrupt traffic flow conditions for a longer period of time. However, depending on the changed traffic flow on the incident-affected road, these smaller roads can in some cases be forced to support a traffic flow they are not meant to facilitate, creating a more widespread congestion over the nearby road network. On the other hand, a too low estimate can instead result in further and worsened traffic congestions, given that road users assume that traffic flow conditions are back to normal and end up getting stuck in the queues. This can increase delay costs and fossil fuel emissions, disrupt crucial operations such as emergency responses and services such as public transport. Since a perfect estimate is unlikely to be achieved considering the lack of initial information of an incident, the traffic control operators deliberately try to over-estimate incident durations, as this is preferred as explained above [4].

1.2 Purpose

The purpose of the work described in this thesis is to, on behalf of Trafikverket, explore the possibilities to generate end-time estimates for traffic incidents by training predictive machine learning models on historical data. This thesis's overall aim is to find and create models that will outperform the current manual estimates set by the operators at the road traffic control centers, evaluated on a selection of metrics. The evaluation of the results also focuses on how the estimates derived from the model differ from the ones produced by Trafikverket's method in terms of estimate prediction range and over- and underestimates. In addition to that, the thesis will also explore what attributes are most important when it comes to the correlation with the duration of accidents. Independently of the results regarding the evaluation metrics and performance towards Trafikverket's estimates, the

thesis will also explore how the data can be more efficiently structured in order to support future applications of machine learning.

1.2.1 Research questions

In this thesis, the aim is to answer the following research questions:

- How well do the implemented models perform compared to the currently employed manual end-time estimation method?
- How do the predictions from the proposed models differ from Trafikverket's?
- How can Trafikverket improve their data collection to facilitate future research and use in predictive modeling?

1.2.2 Delimitations

The research is on Trafikverket's request focused on Stockholm and Skåne counties because of these two regions' different infrastructures. While Stockholm is relatively compact with tunnels and widespread surveillance, Skåne is larger by area and also features roads with significantly less traffic. The results of this thesis are consequently not necessarily directly applicable to other regions in Sweden or abroad. It is also important to point out that the results are based on data from 2020 to 2024 which is the interval selected together with Trafikverket, because of the extensive data sharing a longer time span would result in.

The data used for the produced models are all limited to public information. The use of other external data sources was discussed in the beginning of the project, such as *Swedish Traffic Accident Data Acquisition*, STRADA. It is a database, handled and maintained by police, healthcare and coastguard. The police is responsible for covering all information on traffic accidents while the health care is responsible for reporting injuries into the database [5]. A database such as STRADA would probably complement the NTS data very well, since it handles the same accidents, but in more detail describes any human injuries and more thorough descriptions of the accident scenario. This would probably further improve performance of any produced model. STRADA could, however, not be used, since data is compiled long after the incidents took place and does not share any *Object ID*s with NTS which makes merging the two difficult.

Finally, the data is limited to only include actual accidents. There are two different values for the attribute *Object type* within the NTS data set, namely *Accident* and *Traffic message*, where the latter represents everything but an accident. Traffic messages could for instance be about animal sightings or fallen trees that are blocking lanes. The argument to remove these incidents is that the underlying nature of these tend to be challenging to predict. A common traffic message, for instance "animals on the road", prompts operators

to use a generic estimate, since it is almost impossible to predict an end-time for such an event. Training a model on such events would most likely not yield any valuable insight.

1.3 Previous research

In order to determine which machine learning models could be applied and how the existing data should be handled and characterized, research on both machine learning and the area of which it is intended to be applied was required. An important takeaway from this section is to see what attributes are most important for predicting durations and how they differ from the actual ones. This section focuses on scientific overviews and articles relating to machine learning and their application in the subject of end-time estimates for road traffic incidents. Next follows a selection of articles that provided guidance, valuable insight, and inspiration for this thesis.

In a review on publications of traffic incident duration prediction [6], authored by Grigorev et al., articles were gathered from the data bases *ScienceDirect*, *Google Scholar*, and *Research Gate* with queries relating to end-time prediction. They used a PRISMA methodology, which is a guideline that facilitates transparent and complete reporting of systematic overviews [7], in their process to filter down the information found in 1285 articles to just 174. Their findings concluded that traditional machine learning models such as XGBoost and Random Forest were commonly used. Key challenges found are presented as: Data quality issues, model interpretability, and the complexities following the use of high-dimensional data sets. The authors present recommendations for future research by noting the possibilities found in data fusion models that integrate heterogeneous data sets, utilization of natural language processing for extracting contextual information, and implementation of machine learning pipelines that incorporate anomaly detection, hyperparameter optimization, and sophisticated feature selection techniques. They provided a summary extracted from the different articles reviewed of key factors influencing incident duration. Some of these are peak traffic hours, type of incident, and time of incident. Peak traffic hours relate to the argument that response team preparation is impacted. The type of incident, in cases of more severe incidents, affected not only the emergency team preparation time, but also the clearance time. Time of incident relates to longer clearance times during nighttime and weekends.

Tang et al.[8] cover a selection of scientific articles that apply different statistical and machine learning models to assess whether the incident clearance time prediction models perform optimally, considering the different data they are applied to. The study investigates four different statistical models and four different machine learning models. More specifically, they look at the following statistical models: Accelerated Failure Time, Quantile Regression, Finite Mixture, and Random Parameters Hazard-Based Duration. The machine learning models tested are: K-Nearest Neighbor, Support Vector Machine, Back

Propagation Neural Network, and Random Forest. The review concludes that the Random Forest and Random Parameters Hazard-Based Duration models outperform the other in data fitting and model prediction relative to their respective methodological categories. It further indicates that Random Parameters Hazard-Based Duration, Finite Mixture and Quantile Regression outperform machine learning methods in model prediction when using the *mean absolute percentage error* as the evaluation metric. Machine learning is pointed out to be more consistent and stable in model prediction. Lastly, the review explores the underlying causality of influential factors and concludes that the incident type and the lane closure type have a notable impact on incident clearance time across all eight selected models.

Khattak et al. [9] discuss the prediction of incident durations using an online tool called *Incident Management Integration Tool*, iMiT, and how it is applied by the *Hampton Roads Traffic Operations Center* in Virginia, US. The tool can dynamically predict incident durations, secondary incident occurrence, and associated delays. It is based on statistical models that heavily relies on information about roadway conditions and incident information such as location, time of day, and weather conditions. The prediction tool contains five stages with each stage containing 28 different models, of which one is chosen for each stage depending on the operator input. The tool iterates over the five stages over time after a registered incident. The initial stage takes initial information available to the operator handling the incident and generates an end-time prediction based on this. The second stage produces a new prediction if parameters are updated and or new information is introduced after 10 minutes of the registration of the incident into the tool. It can then be updated similarly after 20-, 30-, and 45 minutes, if new information is presented. Results of the utilization of the iMiT tool for incidents in Hampton during 2007 shows that truncated regression models tend to under-predict incident durations, instead conservative OLS (Ordinary Least Squares) model was preferred in the iMiT tool. From 37 934 observations, the mean duration was 14.3 minutes, the standard deviation 20.2, and maximum duration 728 minutes. Their iMiT OLS model had the same number of observations and mean duration, however, the standard deviation was almost half, being 10.2, and the maximum duration predicted being only 83.1. The summary of these results is that the prediction model based on OLS in iMiT produced a narrower duration span compared to the real observation, and have a difficult time predicting longer durations for incidents.

2 Data Overview

The main NTS data set received from Trafikverket consists of separate Excel sheets containing information about incidents on Swedish roads during the years of 2020-2024. Additional data included in the research was information on whether incidents resulted in local SOS or road assistance being involved, as well as information on weather conditions. All data was merged, cleaned, and transformed using *Feature Manipulation Engine*, FME. Below are more in-depth explanations of the different data sets used, how they were gathered, and what purpose they had in the research.

2.1 NTS accident data

The main data set used for this thesis consisted of 43 attributes describing the accidents. The bulk of data about the incidents was received directly from Trafikverket and consisted of several Excel sheets of data for different years which were combined into one single sheet, resulting in a total of around 600 000 records.

Listed in Table 1 are all the attributes of the original data set along with their explanations and translations, which will be used for reference in the thesis.

Table 1: Structure of the main data set over incidents from Trafikverket.

Attribute	**Explanation**	**Translation**
Objekt ID	Unique ID of incident	ObjectID
Versionslöpnummer	Version number of incident	Version number
Skapandetid	Time of creating incident in database	Creation time
År	Year of incident	Year
Månad	Month of incident	Month
Dag	Day of incident	Day
Veckodag	Weekday of incident	Weekday
Vecka	Week of incident	Week
Timme	Hour of incident	Hour
Halvtimme	Half hour of incident	Half hour
Kvart	15 minutes of incident	Quarter hour
Minut	Minute of incident	Minute
Versionens starttid	Time of update for version	Version's start-time
Versionens stopptid	Time of update for next version	Version's stop-time
Arkiveringstid	Time of archiving	Archive time

Attribute	**Explanation**	**Translation**
Beräknad starttid	Estimated start-time	Estimated start-time
Beräknad sluttid	Estimated end-time	Estimated end-time
Fastställd sluttid	Actual end-time	Actual end-time
Händelsetid	For how long the incident was active in system	Incident time
Händelsetid min	Same as above in minutes	Incident time min
Län	County of incident	County
Vägnamn	Road of incident	Road name
Vägnummer	Road number of incident	Road number
Standardväg	If the road is a standard road	Standard road
Lägestext	Location according to Location-code standard	Position
Xpkt	X-coordinate of incident in SWEREF99	X
Ypkt	Y-coordinate of incident in SWEREF99	Y
Frånplats	Incident's start position	From position
Tillplats	Incident's end position	To position
Planerad	Planned incident or not	Planned
Objekttyp	Accident or Traffic message	Object type
Händelsetext	First part of the Traffic information, Alert-C standard	Incident text
Tilläggstext	Second part of the Traffic information, Alert-C standard	Additional incident text
Påverkan	Severity of accident	Severity
Beskrivning	Free text description of incident	Description
Påverkad trafikriktning	Affected direction	Affected direction
Väg avstängd	Road closed or not	Road closed
Typ av tillfällig begränsning	Type of temporary restriction	Type of temporary restriction
Värde för tillfällig begränsning	Value of temporary restriction	Value of temporary restriction
Körfältskod	Bitmask for which lanes affected	Lane code

Attribute	**Explanation**	**Translation**
Utbredningsriktning länk	What road link is affected	Road link
Longitud	Longitude in WGS84	Longitude
Latitud	Latitude in WGS84	Latitude

For every new bit of information the operator receives, a new version under the same *ObjectID* is created with a new *Version number*. The updates between the versions were generally only minor, often just fine-tuning the end-time estimate and location of the incident. The incidents had on average six versions that had to be combined to one single row, one for each incident. In the end, the version to be used for the analysis was selected together with Trafikverket, to be based on the *Secure Lead Time*. More about this can be read in Section 4.1. A complete view of the NTS data set can be seen in appendix D, in Figure D1.

2.2 SOS and road assistance data

The second part of the data consisted of information about the involvement of SOS and road assistance. This set of data was merged with the main data set by the column *TRISSID* which was the same as *ObjectID*. The road assistance data was only available for Stockholm, and was mapped to the rest of the SOS data using the time and the date, prior to merging with the rest of the NTS data. In Tables 2 and 3 the attributes, explanations, and corresponding translations are presented.

Table 2: Structure of the NTS SOS data set.

Attribute	**Explanation**	**Translation**
Händelseid	Date and time of incident	IncidentID
Platsbeskrivning	Free text describing the location	Location Description
Länsnamn(kod)	County of incident	County
Händelsetext	Short description of incident	Description
Beskrivning när händelsen avslutades	Description of location after incident	End description
Vägen avstängd total tid minuter	How many minutes the road was closed	Road closed min
Skapad SOS	When SOS was contacted	SOS init
Första räddningsfordon på plats	Time when first responders arrived to the location	SOS arrival

Attribute	**Explanation**	**Translation**
Skapad händelse	Time when incident was registered in the database	Incident created
Avslutad händelse	Time when incident was was ended	Incident's end-time
Händelsens livstid [min]	Time in minutes for how long the incident lasted	Incident's duration
X	X coordinate for incident in SWEREF99	X
Y	Y coordinate for incident in SWEREF99	Y
Latitud	Latitude for incident in WGS84	Latitude
Longitud	Longitude for incident in WGS84	Longitud
TRISSID	Unique ID of incident	TRISSID

Since the only information of interest in the SOS data set was whether or not SOS was involved, it was condensed to a single binary attribute named *SOS on scene* which can be seen in Table 6. The other attributes were not relevant, since they either contained information already known or data which could not be used.

Table 3: Structure of the NTS Road assistance data set.

Attribute	**Explanation**	**Translation**
Händelseid	Date and time	IncidentID
Platsbeskrivning	Free text describing the location	Location description
Länsnamn(kod)	County of incident	County
Vägassistans på plats	Time when road assistance arrived at the location of the incident	Road assistance time

Since this data set did not include any unique ID, it had to be merged to the data set presented in Table 2 by using the *IncidentID*, followed by a new mapping of this combined data set to the main data set in Table 1. Similarly as for the SOS data set, the primary information of interest was whether road assistance was involved, which led to the creation of the new attribute *Road assistance on scene*.

2.3 Weather data

To complement the data from Trafikverket, weather data from *SMHI* was gathered. The two attributes used for this thesis project was *Temperature* as well as *Precipitation*, which

historical data was downloaded from SMHI's website [10]. For Skåne, historical data was collected from 6 weather stations on precipitation and from 8 stations on temperature data. For Stockholm, 4 weather stations were used for the precipitation data and 8 stations for the temperature data. At the time of fetching the weather data, the most recent data was from November 2024 which led to a limiting the rest of the NTS data to this date. Since the NTS data included attributes for coordinates, each accident could be mapped to the nearest station from SMHI's data and then inherit the temperature and precipitation for that specific date and time. The way that incidents, marked as red dots on the maps, are assigned to their corresponding weather stations is shown in Figures 2 and 3. The larger red and blue dots refer to weather stations for temperature and precipitation, while the black borders represent the zones of coverage for each station.

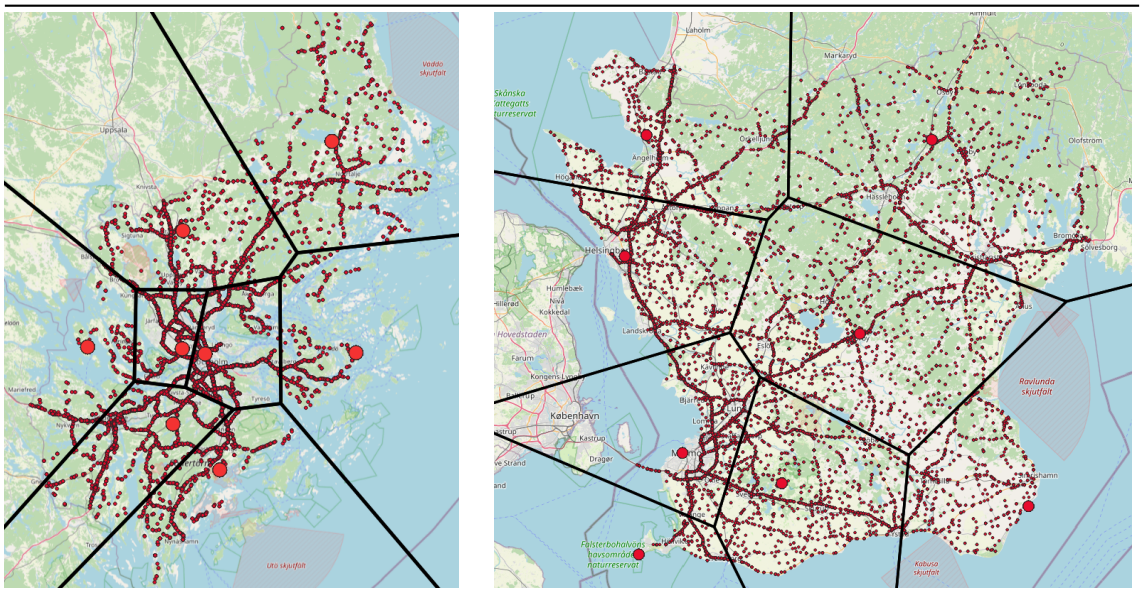


Figure 2 Maps showing how each incident is mapped to corresponding weather stations to gather information on temperature in Stockholm (left) and Skåne (right).

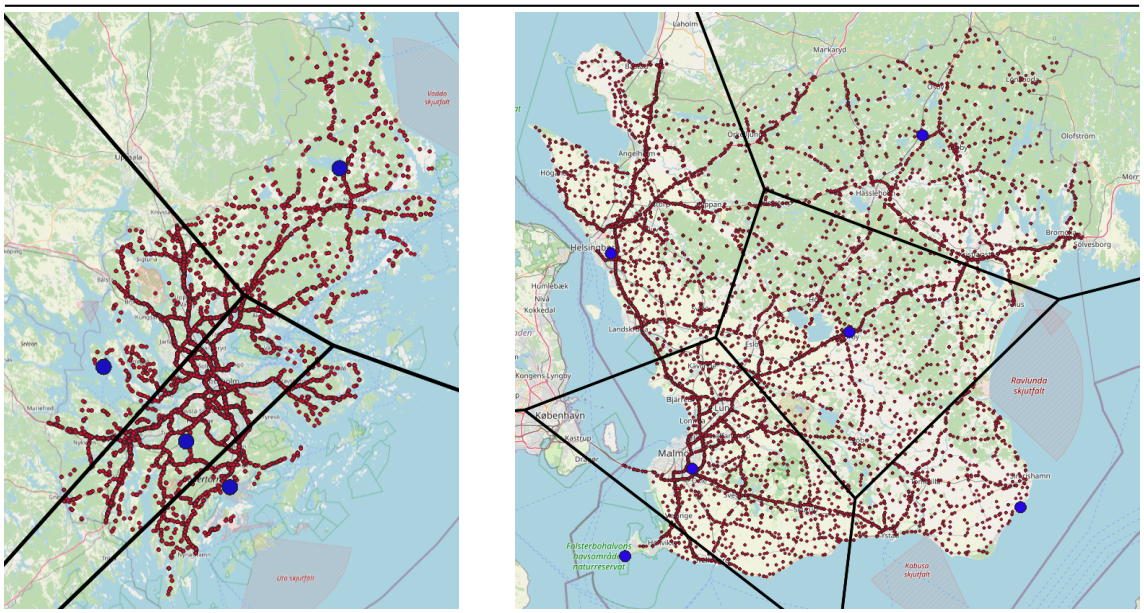


Figure 3 Maps showing how each incident is mapped to corresponding weather stations to gather information on precipitation in Stockholm (left) and Skåne (right).

Although there were additional weather stations available, not many had collected data on minute basis during the entire time period 2020-2024. This meant that the accuracy of the weather data mapped onto a given accident could not be guaranteed to be correct, as some accidents were mapped to stations far away.

3 Theory

In this section, the fundamental theory of *machine learning* is presented with its key concepts and methodologies. In Section 3.1 an overview of machine learning is given and principles such as supervised and unsupervised learning, classification, and regression are described. In addition, the section will touch on the subjects of *evaluation* and *tuning*. The section will hopefully provide a sufficient understanding of the subject, in order to accommodate further reading. In sections 3.1.1 and 3.1.2 more in-depth explanations are given for the models used in this thesis.

Note that these sections reach deeper into the specific mathematical frameworks of the machine learning models SVR and XGBoost. It is thus *not* mandatory to read these sections in order to get an adequate grasp of the thesis's results.

3.1 Machine Learning

The term machine learning refers to the construction of a software system that will learn how to make decisions or predictions based on data. The concept resides within the subject of data analysis and the two are associated with data science. A machine learning model consists mainly of statistical and mathematical models for data that show how different variables in the data correlate to each other and how new unseen data can be predicted from what it has previously learned [11].

There are two main forms of machine learning; *supervised* and *unsupervised*. When data is unlabeled, that is, it does not have any attributes or outputs, unsupervised machine learning is used. In this case, the model tries to cluster the data based on observed structural patterns. The data set used for this project is labeled, and hence only supervised machine learning will be used. In this case, the data from which the model learns is called *training data* which consists of an input variable `X` and an output variable `y`. The aim of the model is to learn from the training data in order to be able to predict `y` for previously unseen input data `X`. The model learns from previous examples to predict new ones. Since the data often contains many pairs of `X` and `y` the expression for training data can be expressed as Equation 1 below [11].

$$\tau = (X_i, y_i)_{i=1}^{n} \tag{1}$$

Variables or attributes can be of either *categorical* or *numerical* format. Numerical attributes are *ordinal*, that is, there is an explicit ordering between the values in the category. For example, the number 2 is greater than 1, when the values represent time durations. Categorical attributes are, on the other hand, *discrete*, that is, the values are representative placeholders. The numbers 2 and 1 could in this case represent something like 2 = "Stockholm" and 1 = "Skåne" for example. This concept leads to two different

problems in supervised machine learning, *regression* and *classification*. Either problem can take numerical or categorical input, whereas the output is what distinguishes the two, where the regression outputs numerical values and the classification outputs categorical [11]. In this project, the inputs will be both numerical and categorical, while the output will only be numerical, since it is a prediction on the duration.

A common method for evaluating the performance of a regression model is *Mean Absolute Error*, MAE, which essentially is the mean numerical value for how close to the actual value the estimate of the model lies. A closely related evaluation metric is *Mean Absolute Percentage Error*, MAPE, where the output is a mean percentage for how close the prediction is to the actual value. Another method is *R-squared*, R2, which outputs a number, n. If $n = 1$ this indicates that the regression model fits the data perfectly, while if $n = 0$ this indicates that the model does not explain any of the variance in the dependent variable. In other words, when $n = 0$, the independent variables have no linear relationship with the dependent variable. More specifically R2 indicates how much of the variance in the target or output attribute can be explained by the model [12]. A R2-score of 0 is reached when the best prediction a model can give is the mean value of the target variable, meaning that the model does not take into account any information from the attributes trained on. A negative value indicates that the model performs worse than always predicting the mean value.

In traditional machine learning, each data point is considered to be independent of each other. When working with data including date-time attributes, it is important to consider that data points can be both ordered and correlated, resulting in seasonal trends. Attributes with values in data-time form are considered to be *time series data*. Typical examples are trends that repeat on an hourly, daily, monthly, or even yearly basis [12]. For data in which such attributes can be gathered or engineered, they can significantly affect the results of a machine learning model.

3.1.1 Support Vector Regressor

Support Vector Regression (SVR) is a supervised machine learning technique that handles regression problems. SVR's principles stem from *Support Vector Machines*, where the former is used for regression problems and the latter for classification problems. The key idea is to find a function for a hyperplane that approximates the relationship between the input features and the target variable. For nonlinear models, the hyperplane approximation is done in a higher-dimensional space, referred to as the kernel space, using a kernel function. This allows the SVR to solve a nonlinear problem with a linear hyperplane in the kernel space. This is preferred compared to having to solve a complex high-order separating hypersurface in the data input space [13]. The kernels that are available in the Python library `sklearn.svm` include linear, polynomial, radial basis, sigmoid, and other pre-computed functions [14]. SVR uses a tolerance margin, ϵ, where data points

within this margin are considered acceptable and do not affect the model, while data points outside the margin contribute to the loss function and are penalized accordingly. Another way of explaining data points outside the margin is that they provide support vectors that influence the position and orientation of the regression line by determining the model boundaries [13]. A graphic representation of a linear regression line with its ϵ-margins included, containing generic data points, can be seen in Figure 4. As can be seen the ϵ-margins will form a 2-dimensional *ϵ-insensitive tube*, containing the values that support the predictions of the model.

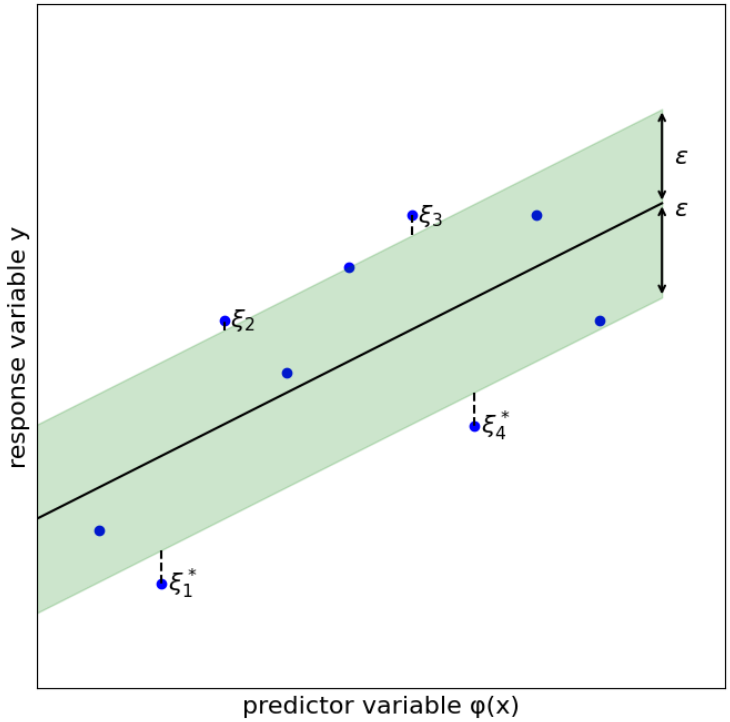


Figure 4 Plot containing generic data, showcasing how a linear SVR can be configured. The data points within the green ϵ-margins are considered acceptable, while the data points ξ_i and ξ_i^* outside this margin are slack variables that allows flexibility in the model and a softer margin. They also help deciding which points should be regarded as support vectors, ie. the data points that will decide the decision boundary or regression function of the model.

The objective of SVR is to create a function that predicts data points with deviations no greater than ϵ from their true values. This is achieved by constructing the *ϵ-insensitive tube* mentioned above around the estimated function, thus allowing for slight inaccuracies within a specified margin. For a linear SVR, the function can be derived as $f(x) = w^T x + b$, where $w^T x$ is the dot product of the weight factor w^T and the input data x, and b is the bias term. An *ϵ-insensitive tube* as flat as possible can be achieved by minimizing the norm of w, as in Equation 2 [13].

$$min \frac{1}{2}||w||^2 \tag{2}$$

This, in turn, is subject to Equation 3.

$$\begin{cases} y_i - w^T x_i - b \leq \epsilon \\ w^T x_i + b - y_i \leq \epsilon \end{cases} \tag{3}$$

In essence, the ϵ-margin affects the flatness of the tube. A lower value means a low tolerance for prediction errors, whereas a higher value leads to a higher error tolerance. This is important to keep in mind when applying a higher dimensional kernel function, as a small ϵ-margin in combination with, for instance, a higher polynomial kernel function might lead to overfitting, while underfitting can be achieved by making it too broad [15]. The loss function used in this project is linear, which is shown in Equation 4.

$$L(y, f(x)) = \begin{cases} 0, & \text{if } |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon, & \text{otherwise} \end{cases} \tag{4}$$

To account for noisy data points at the hyperplane boundaries, slack variables ξ and ξ^* are introduced. These protect against outliers, as seen in Figure 4. These variables determine how many of the data points are tolerated outside the ϵ-margin. The optimization problem can now be updated with the use of these variables as in Equation 5 [13].

$$min_w \frac{1}{2}||w||^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \tag{5}$$

$C > 0$ denotes the regularization parameter, which states the trade-off between the flatness of function $f(x)$ and the prediction errors. Equation 5 is, in turn, subject to Equation 6 [13].

$$\begin{cases} y_i - w^T x_i - b \leq \epsilon + \xi_i \\ w^T x_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{6}$$

The first constraint ensures that the predicted value $w^T x_i + b$ does not deviate from the actual value y_i by more than ϵ plus an additional slack variable ξ_i. The slack variable allows for some flexibility in fitting the model, accommodating points that lie outside the ϵ-insensitive tube. The second constraint works similarly but on the other side of the ϵ-insensitive tube. The third condition enforces that the slack variables are non-negative. What separates the linear kernel function from the others, that is, the ones with the radial and polynomial basis, is that the latter transform the data input from a linear input space to a higher-dimensional kernel space. In a nonlinear setting, the optimization problem is to solve the flatness in the kernel space instead of the input data space. The transformed kernel, x can be denoted as $\varphi(x)$, which for a linearly stated problem can be described

with the equations discussed earlier in this section, with instances of x being replaced by $\varphi(x)$ [13].

When training an SVR, there are a few parameters that can be changed. Hyperparameter tuning is the process of trying combinations of parameters in order to see how they affect the model's performance. The parameters used for the application of this thesis are presented in Table 4.

Table 4: Hyperparameters used for tuning the Support Vector Regressor. [16]

Name	**Range**	**Effect**
kernel	'linear', 'poly', 'rbf'	Specifies the kernel type
degree	[0,∞]	Degree of polynomial kernel function
gamma	'scale', 'auto' [0, ∞]	Kernel coefficient
C	(0, ∞]	Regularization parameter
epsilon	[0, ∞]	Specifies the epsilon tube

3.1.2 XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is a popular tree-based machine learning library for both classification and regression problems. With success within many areas, the model is now widely recognized and known to generally perform well with many award-winning applications. The main advantage over other models is that it can handle large data more efficiently [17].

XGBoost is based on *Classification and Regression Trees*, CART, which is a tree-based model that splits data based on the attribute values that best separate the data. When a split occurs, the model chooses the attribute that maximizes the separation of classes or minimizes the variance to the output. The splitting continues until reaching a maximum depth or until the data would not allow for more splits. Assume someone is predicting house prices and available attributes are size, number of bedrooms, and location, the model would split the data based on the attribute that best separates the data concerning the target variable, house price. One case could be that a specific location really stands out in pricing. A first split being made there would result in all other locations being separated from it. This process results in a downward growing tree as can be seen in Figure 5, where each feature is represented by a node and each leaf represents a continuous predicted value. When given new data, the model follows the paths in the tree structure to get a prediction. CARTs are known to be prone to overfitting, which means that a model learns training data too well and, therefore, performs poorly on unseen data. To prevent this, pruning can be used, which removes unnecessary features, resulting in lower complexity [12].

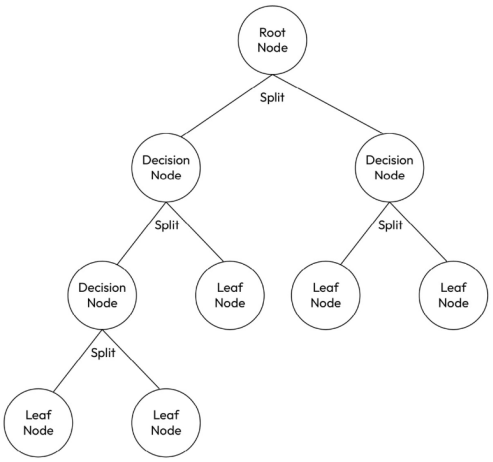


Figure 5 Structure of a CART [12].

In order to reduce both bias and variance, as well as to increase performance, ensembled models are created by training multiple models in order to find the best. In other words, ensembled models combine the predictions of multiple models to mitigate the weaknesses of individual models, leading to improved accuracy and stability. One method utilizing this concept is *bootstrap aggregation* which samples the original data and trains on it separately. However, the data points used for the different models are not exactly the same. In some, data points are missing, and in others the same data points are used in multiple models. The final prediction is obtained by taking the average of all the predictions, which will reduce the variance. Another method is *boosting* where multiple models are trained on the data where each tries to correct the mistakes from the previous, by adjusting the weights, as seen later. In return, this process reduces bias. The final prediction is then given by a weighted sum of all the models in which the top performers weigh the most [12].

XGBoost falls into the category of gradient-boosting tree models, which work by growing trees and minimizing a loss function. Some factors differ XGBoost from basic gradient-boosting models, such as the ability to handle sparse data and avoid overfitting. Sparse data sets are essentially incomplete, consisting of zeros, missing values, and NaN. To handle overfitting, the model uses *shrinkage* to scale the weights after each boosting round and *subsampling* which enables the model to use fewer rows when computing the loss function, preventing the model from becoming dependent on some specific data points [12].

Although the XGBoost model works well with its default parameters, there is room for improvement since every data set is different. In Table 5 the parameters are presented with their corresponding ranges and explanations.

Table 5: Hyperparameters used for tuning the XGBoost Regressor. [18]

Name	**Range**	**Effect**
eta	[0,1]	Shrinks feature weights after every boosting round
gamma	[0, ∞]	Minimum loss reduction required to make a split at at leaf node
max_depth	[0, ∞]	The maximum depth of a tree
min_child_weight	[0, ∞]	If a split results in a leaf's instance weight less than the value, the splitting is stopped
max_delta_step	[0, ∞]	The maximum delta allowed each leaf output. Helps with imbalanced data
subsample	[0,1]	Percentage to randomly sample the training data before growing trees
colsample_bytree	[0,1]	Subsample ratio of columns when constructing a tree
reg_alpha	[0, ∞]	Performs L1 regularization on the absolute value of weights. Higher value leads to more conservative model
reg_lambda	[0, ∞]	Performs L2 regularization on the sum of squares of the weights. Higher value leads to a more conservative model

The rest of this section presents the inner workings of XGBoost. Statistical and mathematical descriptions show how the loss functions are calculated and how the splits are chosen. For a gradient-boosting tree model, the final prediction $\hat{y}_i$ is the mean of the prediction from all CARTs, calculated as Equation 7.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{7}$$

Each K is the number of regression trees, f_k is the function of each tree, x_i represents the input parameters, and $\mathcal{F}$ defines the space containing all the regression trees. Each tree is associated with a tree structure q and weights w, so each tree function $f(x)$ constitutes a prediction that depends on the weight assigned to the leaf for the input x. $\mathcal{F}$ can therefore be defined as $\mathcal{F} = \{f(x) = w_{q(x)}\}$. Each leaf is associated with a weight for the i-th level represented as w_i. The result of summarizing all the weights is the final prediction.

For XGBoost, the *regularized objective* as in Equation 8 must be minimized [17].

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{8}$$

where l is a loss function that essentially represents the difference between the prediction

and the target value. $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$ is a penalizing term that prevents overfitting by smoothing out the learning weights, preventing trees to grow too deep and complex with γT adding cost for the number of leaves T and $\frac{1}{2}\lambda||w||^2$ adding penalty for large weights in the trees, in turn preventing the model from relying heavily on single features. Without the second term, the *regularized objective* is the same as for basic gradient tree boosting. Since the expression contains functions, the model needs to be trained by using the prediction one time step before. For example, if the prediction for t is computed, t-1 is needed. Equation 7 can therefore be expressed as:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{9}$$

Using Equation 9 in Equation 8 then the expression for the total loss at time t is derived. The goal is to find the $f(t)$ that minimizes this loss.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}^{(t-1)} + f_t(x_i)) + \Omega(f_k) \tag{10}$$

The expression in Equation 10 can be approximated as in Equation 11 by using a second-order approximation [17].

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{11}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are first- and second-order gradient statistics on the loss function [17]. By removing constants, the approximation for the objective at time t can be expressed as Equation 12:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{12}$$

Rewriting Equation 12 with an expansion of Ω, gives the following expression:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{13}$$

Equation 13 can be expressed for each leaf j as seen in Equation 14. The instance set for leaf j is defined as $I_j = \{i|q(x_i) = j\}$

$$= \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \tag{14}$$

For a leaf j, the optimal weight w^* can be calculated by summarizing the first- and second-order derivative of the loss function for all instances of leaf j adjusted by the regularization

parameter λ as in Equation 15.

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{15}$$

The optimal value of the loss function can then be calculated as in Equation 16.

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{16}$$

I_j is defined as the instance set of leaf j. In other words, I_j is the set of all data points that are assigned to leaf j with the function $q(x)$. Equation 16 is used to measure the quality of the tree structure. Since all structures q would be close to impossible to enumerate, a greedy approach is used which iteratively adds branches to the leaves. I_L and I_R are defined as the set of instances for the left and right nodes, respectively, after a split [17]. If $I = I_L \cup I_R$ the loss reduction after a split can be defined as Equation 17. The first two terms within parentheses represent the contribution to the loss function for instances going to the left and right nodes, respectively. The last term represents the current loss before the split for all instances in the parent node.

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \tag{17}$$

In summary, XGBoost uses a combination of advanced tree boosting techniques, regularization, and second-order derivative optimizations to produce robust, scalable, and accurate predictions. The mathematical framework behind it, including the calculation of loss functions and optimal weights, ensures efficient model training and meaningful splits, ultimately enhancing performance and preventing overfitting.

4 Method

During this project, the research has been both quantitative and qualitative. Quantitative, in the sense of collecting and analyzing measurable data to find certain patterns and trends; Qualitative, in the sense of conducting interviews and gathering more information on the subject of traffic accidents [19]. During the process of writing this thesis, several interviews were held with road traffic control operators from both Stockholm and Skåne as well as other people working at Trafikverket. The opportunity to visit the operators in Stockholm was also taken in order to see how traffic accidents are handled in real time. All interview questions are presented in Appendices E and F.

The main objective of this thesis is to predict the duration of traffic accidents. To accomplish this, predictive models that utilize historical data to find patterns in order to make predictions were constructed. Predictive analysis is the process of combining statistical modeling, data mining, and machine learning to make estimations of future results [20]. The goal is for the model to learn patterns in the data to then be able to make estimations when tested on unseen data, minimizing the error between the prediction and the target value [21].

The main tools during this project have been *FME Workbench* for data handling, *Google Colaboratory*, Colab for the code writing and predictive models, and *QGIS* for spatial data visualization. FME is a platform used for data integration which enables great support for geographical and spatial data. The environment consists of a workspace where the data is handled by readers, writers, and transformers, used to clean and manage the attributes [22]. Although FME is a no-code software there are transformers, so called *PythonCallers*, where scripts can be run to perform some calculation.

Colab is a Google-hosted Jupyter Notebook service that is well suited for data science and machine learning. Instead of a user using the computing power of their own devices, Colab provides free computing resources such as GPUs and TPUs [23]. When visualizing the data, which is presented in Section 4.1 and when training the machine learning models, sections 3.1.1 and 3.1.2, necessary libraries were imported to the Notebook, such as `Numpy` for computation, `Pandas` for analysis, and `Matplotlib` for plotting.

The Open Source Geospatial Foundation aims towards helping and promoting open-source geomatic softwares for collaborative development. One of their products is QGIS, formerly known as Quantum GIS, is a free-to-use open-source software used to store, analyze, manage, and collect spatial and geographic data. The development of the software lies in the hands of volunteers who work with everything from Python plugins to anomaly detection [24]. Since every traffic incident is related to a specific location as seen in Section 2.3, it was important to depict the incidents on a map to see how some locations and distances to the city centers were correlated with the durations. The map background used in QGIS was OpenStreetMap, which is another free service that does not require a

license [25].

The following subsections explain in depth how the tasks of handling the data were solved. First in Section 4.1 all the data preprocessing and transforming performed in FME are presented, as well as the final data set. The modeling methodology and preparation for machine learning are presented in Section 4.2.

4.1 Handling the data

The main NTS data set was split into several Excel files that were all read and combined to form a single workflow in the FME Workbench. In order for FME to recognize the geographical attributes *X* and *Y* a *CoordinatSystemSetter* transformer was added that set all values to a coordinate in the *SWEREF-99* format. Attributes deemed as not important for future modeling were all removed leaving about half of the original ones. The attribute *Object type* had two values; 'accident' for actual accidents involving vehicles as well as 'traffic message' which essentially is everything else. The two types of incidents were separated leaving only accidents split up into Stockholm or Skåne. The reason for splitting up the data into two sets, each representing a county, is because of how differently the data sets are gathered in the two counties as well as how the individual operators process the information received. Furthermore, a model predicting accidents durations in Skåne, should not be trained on data from Stockholm. The next step was to integrate the SOS and road assistance data described in Section 2.2 into the main data set. By first joining the SOS data to the Road assistance data by using *IncidentID*, it could then be joined to the main data by *TRISSID* and *ObjectID*. Since the data contained date and time for when SOS or road assistance arrived at the scene, it was concluded that it would be more useful to create new binary attributes for if SOS or road assistance was involved. This was done by creating two new attributes *SOS on scene* and *Road assistance on scene*. In addition, each incident was mapped using the transformer *NeighborFinder* to the closest weather stations, one for precipitation and one for air temperature. By then comparing the start time for the incident with the timestamps of the corresponding closest weather stations, each accident inherited the precipitation and air temperature. Since the weather stations only reported the conditions at their exact respective positions, one can not be sure that the weather conditions for where an accident took place were the same as for the station. Even if weather data is derived in this way, it was considered a viable solution to obtain at least an indication of the weather for any given accident.

As described in Section 2 each incident had several versions with the same *ObjectID*. Initially, the decision was made to keep only the first version of each incident since that version contained the initial end-time estimate which is the one that the model will be compared against. The second option was to keep the last version since that contained the most information. However, both options were discarded, referring to the fact that the

first version of each incident could not provide enough information to train a model on, while going with the last version would mean that the prediction has already been rendered meaningless, as any accident would already almost have been fully handled. The decision was made to keep the version that was registered in the closest time to the *Secure Lead Time*, which would also be fair to compare against. By request from Trafikverket, incidents shorter than 5 minutes or with only one version were discarded since these were said to be incorrect data. Furthermore, attribute values containing null were removed as well as all accidents prior to 2020-01-01 and after 2024-11-02, since this was the interval for which weather data were available at the time of gathering. A *PythonCaller* transformer was used to translate a binary code from which lanes were affected to how many lanes. Next, two new attributes were created, *Estimated duration* by calculating the difference between *Estimated start-time* and *Estimated end-time* as well as *Actual duration* by calculating the difference between *Estimated start-time* and *Actual end-time*. The attributes were rounded to the nearest minutes. Note that there is no *Actual start-time* in the NTS data set, therefore using the *Estimated start-time* is the only viable option.

As seen in Figure 6, there was a seemingly excessive amount of correct estimations. After asking the operators in Stockholm how often they think one would make a correct guess, the paraphrased answer, "a few times over the span of a career", [4] concluded that the amount of correct estimations, 11% in Stockholm and 15% in Skåne, was misleading and would deteriorate the performance of a machine learning model. This led to the decision to remove the correctly estimated incident durations. The reason for the excess amount of correct guesses, also explained in Section 1.1, is that when no more information is received by the operators, they close the incident when their estimated end-time passes, in turn leading to the *Estimated end-time* and *Actual end-time* being the same. The spikes seen in Figures 6 and 7 represent the fact that the suggested prediction of 38-39 minutes is the most common prediction, while 62 minutes is the most commonly used in Skåne.

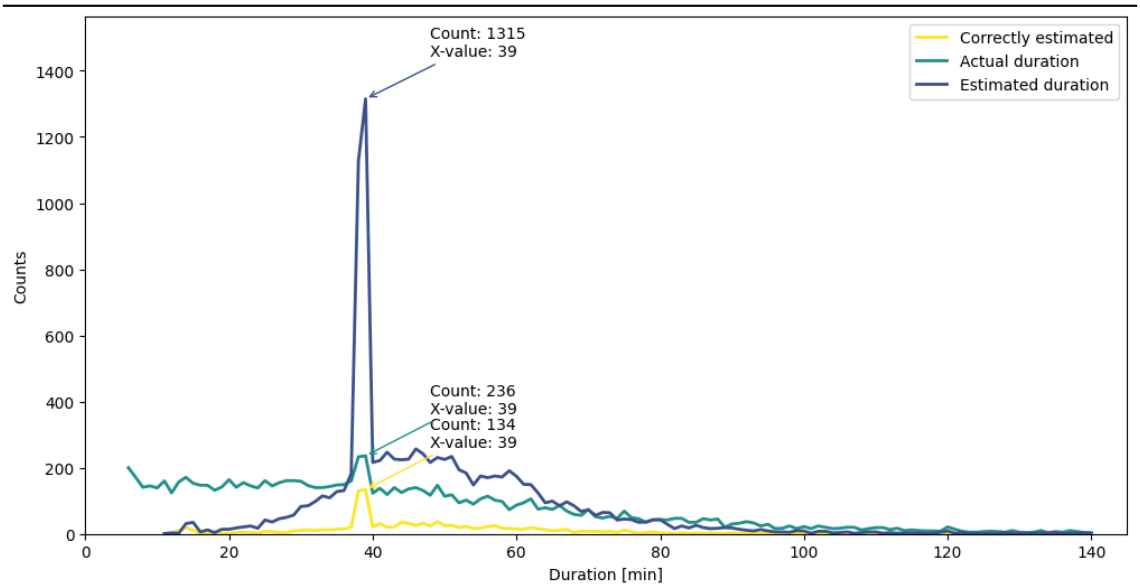


Figure 6 Combined line graph showing the counts of actual, estimated and correctly estimated reports of accidents in Stockholm. Note that the by far most common estimation is 39 minutes, for all plotted values.

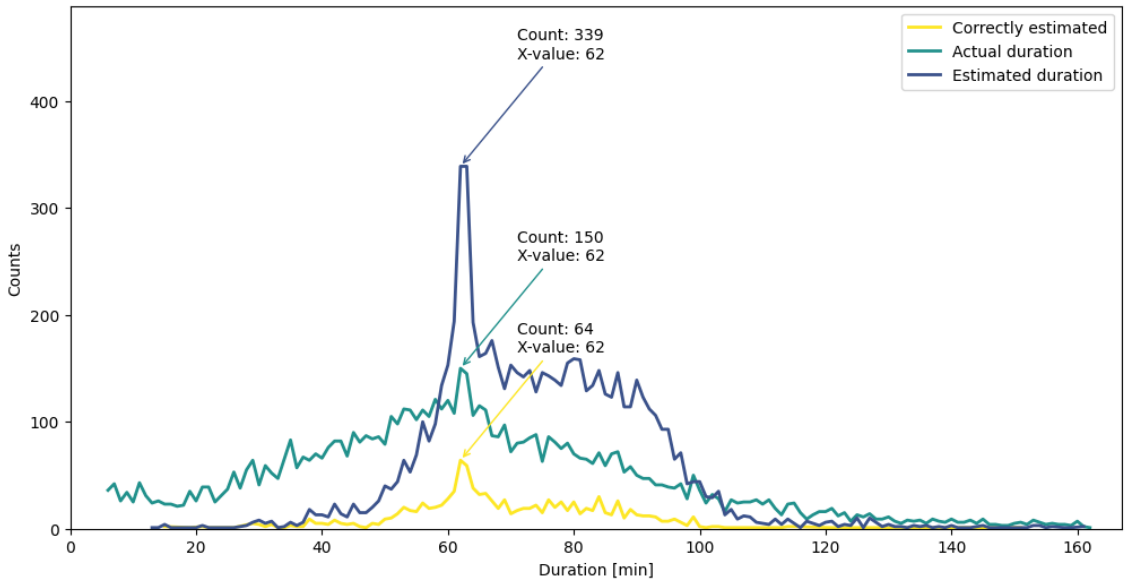


Figure 7 Combined line graph showing the counts of actual, estimated and correctly estimated reports of accidents in Skåne. Note that the by far most common estimation is 62 minutes, for all plotted values.

The positions, namely the coordinate attributes X and Y in the NTS data, themselves did not indicate any relation to the end-time. However, when these were plotted as clusters using `KMeans` on a map, a vague pattern appeared that shows a relation between the *Actual duration* and distance to central Stockholm. Therefore, the attribute *Distance to center* was introduced, where *Sergels torg* was used as the middle point. This geographical point was chosen as it could be considered the most central point in Stockholm, as well as the fact that it is also the middle point of the circular area in which *Road assistance* is active.

Rather disappointingly, the new attribute did not improve the initial models at all, and it was concluded that the visual pattern was overly generalized. Instead, it was replaced with the attribute *Near center*. This was a binary type that checked if an incident was inside a five kilometer radius from Sergels torg or in Skåne, from Malmö or Helsingborg center. In Figures 8 and 9 the clustered accident data in Stockholm and Skåne county are presented respectively.

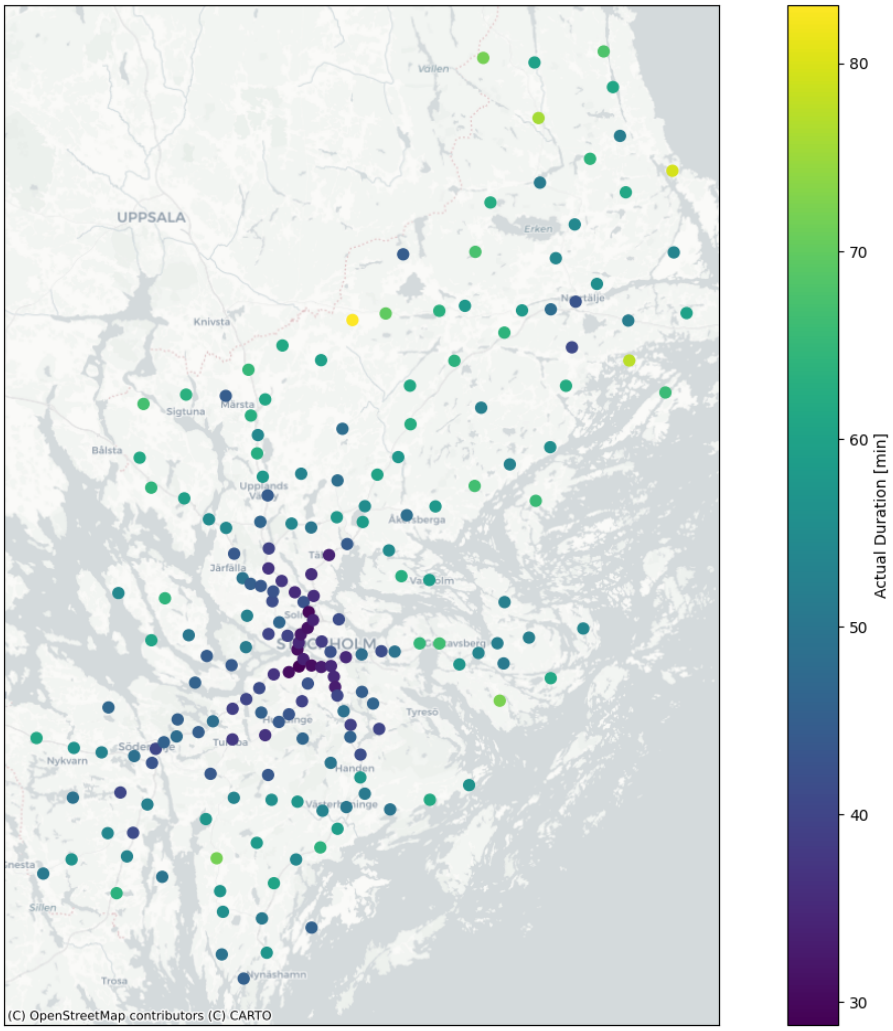


Figure 8 Map plot showing the *Actual duration*s for accidents in Stockholm county. `KMeans` is used to condense 8919 records into approximately 200 clusters of averaged durations, related through their geographical proximity. The plot indicates that geographical location of an accident is a factor when predicting its duration, as durations in general seems to become smaller as one gets closer to Stockholm city. This pattern is partially caught using the *Near center* attribute, which is a binary attribute that labels accidents as "Yes" if they are inside a 5 km radius from Sergels Torg (the chosen midpoint of Stockholm) and "No" if outside.

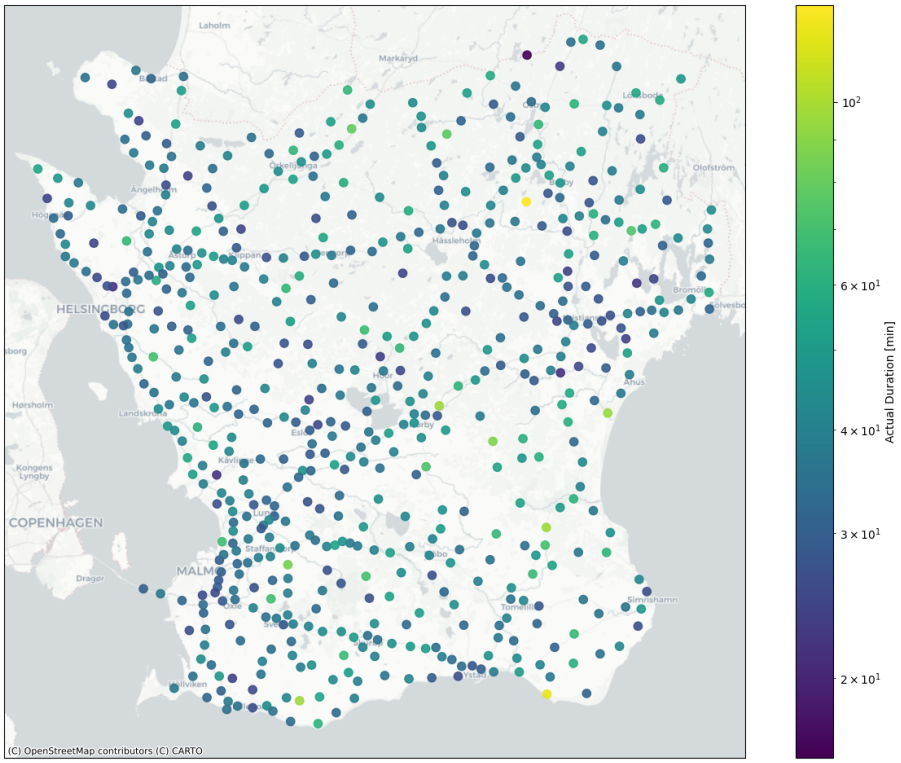
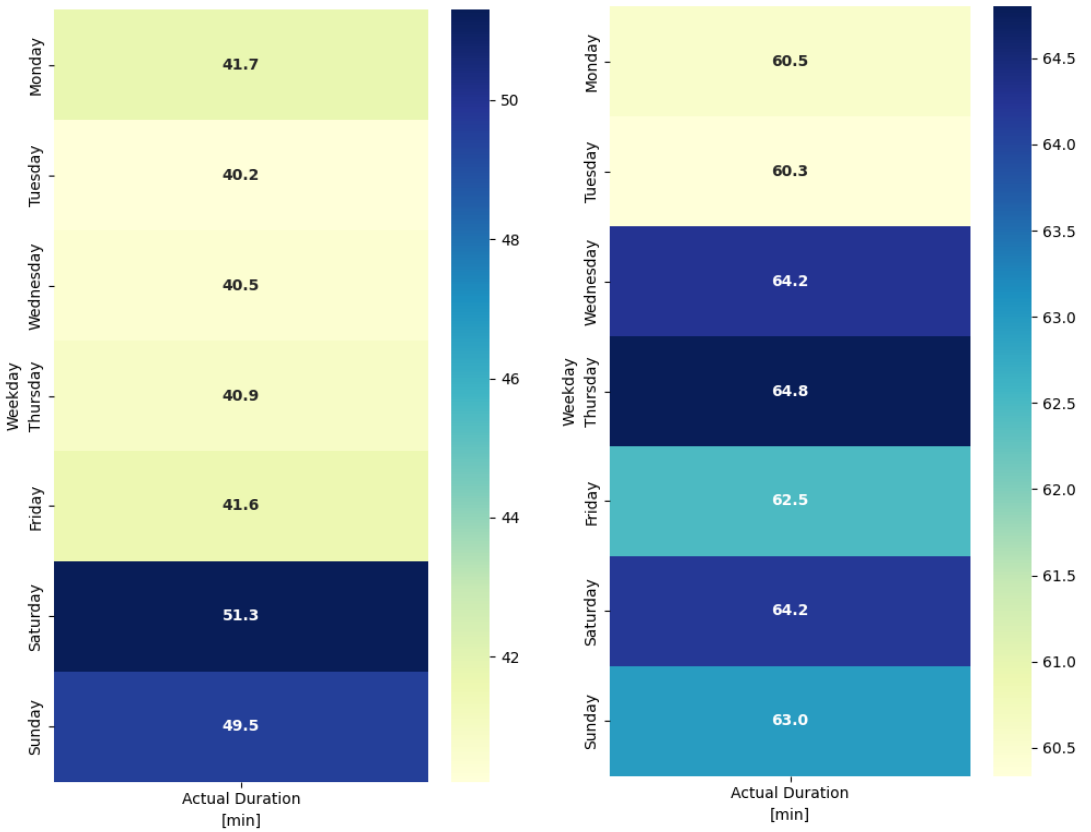


Figure 9 Map plot showing the *Actual duration*s for accidents in Skåne county. KMeans is used to condense 5931 records into approximately 500 clusters of averaged durations, related through their geographical proximity. Despite the color hue being scaled logarithmically to facilitate observation of patterns, these can still not be clearly determined.

Differences in *Actual duration* could also be seen for the *Day* attribute. Specifically, there was a difference in duration depending on if it was a work day or a weekend, as seen in Figure 10. This led to the attribute being converted to the binary type attribute *Weekday*. In Skåne, patterns for *Actual duration* over time could not be as easily discerned from the accident data, however, when applying the same generalization of attribute values, model complexity was reduced while also resulting in a minor improvement in performance.

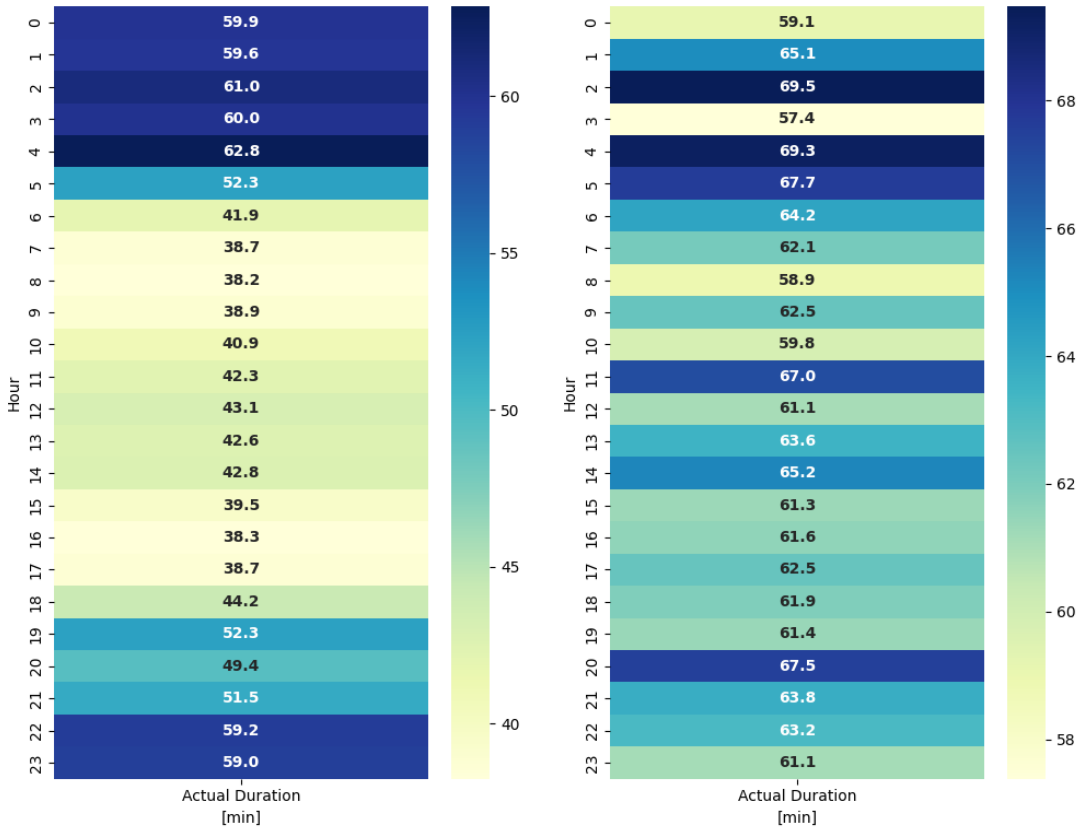


(a) Heatmap of accidents showing how the mean *Actual Duration* differ depending on day of the week in Stockholm.

(b) Heatmap of accidents showing how the mean *Actual Duration* differ depending on day of the week in Skåne.

Figure 10 Heatmaps showing how *Actual duration* for accidents in differ depending on day of the week.

With a similar method as with *Day*, *Hour* could also be partitioned into fewer values, generalizing the data. In this case, the average *Actual duration* was higher during the night, early morning, and evening, in contrast to being shorter during the rest of the day. *Hour* was subsequently divided into 'Morning' (00-06), 'Day' (06-17) and 'Night' (17-24). Just as for the day of the week, patterns in Skåne were more difficult to find, however, when applying the same generalization of attribute values, model complexity was reduced while also resulting in a minor improvement in performance. The heat plots for *Actual duration* in relation to the hour of day can be seen in Figure 11 below.



(a) Heatmap of accidents in Stockholm showing how the mean *Actual Duration* differs depending on hour of the day.

(b) Heatmap of accidents in Skåne showing how the mean *Actual Duration* differs depending on hour of the day.

Figure 11 Heatmaps showing the mean *Actual Duration* for accidents depending on hour of the day.

Other data transformations included creating the attributes *Tunnel* and *Main road* by using the *StringSearcher* transformer on the attributes *Description* and *Road number*. These two attributes were created to capture the well-observed part of the road network, potentially with higher data quality. Lastly, the numerical attributes *Temperature* and *Precipitation* were combined to a single attribute *Weather* taking categorical values as seen in Table 6. The categorical values were made through combining the noted temperature whilst also considering the precipitation. If precipitation was zero, the attribute would receive the value "Clear". If precipitation was non-zero, the temperature would decide if it was either "Rainy" or "Snowy". This solution of translating a wide range of numerical values into few categorical values, was based on the idea that fewer values were necessary in order to make generalization of the data possible. It is also important to note that this is not a perfect solution, as other weather types exist, it is however, indicative of the weather on scene. At the end of the data handling process, a manual inspection was performed to find obvious errors that would not be found through standard preprocessing. This could, for instance, be misclassified *Incident texts* of accidents, or attributes that directly contradict each other for a given accident. More about this can be read in Section 6.1.3.

Only some small changes had to be made before the data was ready for modeling. This included removing some temporary attributes, fixing some coordinate issues and running all data through a *Validator* transformer to ensure that there were no null values. Most of the data in terms of *Actual duration* are in lower intervals, with only a small portion of the incidents having a duration longer than 150 minutes. The low number of records in the higher intervals was a problem since the models had too few records to train on, leading to difficulty recognizing patterns in the higher intervals. The approach was taken to use *Inter Quartile Range, IQR* presented in Equation 18. By applying IQR on *Actual duration* for all incidents, an upper limit was set, in turn, removing outliers [26]. After applying IQR, around 5% of the data was removed and an upper limit was established at 147 minutes for accidents in Stockholm and 174 minutes for accidents in Skåne.

$$\begin{aligned} IQR &= Q_3 - Q_1 \\ LB &= Q1 - 2 \cdot IQR \\ UB &= Q3 - 2 \cdot IQR \end{aligned} \tag{18}$$

Q1 and Q3 represent the first and third quartile, respectively, while LB and UB represent the lower and upper bound. Only the upper bound was used since the lower bound was already set to 5 minutes as explained earlier in this section.

In Table 6 is the structure of the final data sets after processing in FME. Two of the attributes *Tunnel* and *Road assistance on scene*, are only present in the data set for Stockholm, while the attribute *Main road* was only used in Skåne.

Table 6: Attributes and corresponding format after handling the data in FME.

Attribute	**Format**	**Example**
Time of day	categorical	morning, day or night
Weekday	binary	yes or no
Month	categorical	1,2,3,...
Weather	categorical	clear, snow or rain
Incident text	categorical	heavy vehicle, object on road etc.
Severity	categorical	no, low, high or very high
Affected direction	categorical	one or both
Road closed	binary	yes or no
Road link	categorical	one or two
Number of lanes	numerical	0,1,2,...
SOS on scene	binary	yes or no
Road assistance on scene	binary	yes or no
Road number	categorical	4, 8 etc.

Attribute	**Format**	**Example**
Near center	binary	yes or no
Tunnel	binary	yes or no
Main road	binary	yes or no
Estimated duration	numerical	0,1,2,...
Actual duration	numerical	0,1,2,...

4.2 Modeling of data

The models were trained with the common 80/20 split, which means that 80% was used to train the models and 20% was used to test the results. The split was done randomly, minimizing the chance of only training on specific parts of the data. Since the data included both categorical and numerical features, scalers and encoders had to be utilized. For this `StandardScaler()` and `OneHotEncoder()`, both from the `sklearn.preprocessing` library was used. For numerical attributes, the value after scaling is calculated as $z = (x - u)/s$ where x is the original value, u is the mean, and s is the standard deviation. Standardization of numerical attributes is common practice for preprocessing in machine learning. The result is an interval where the mean is 0, which means that all the values are centered around 0 [27]. The encoder works by converting categorical values into binary columns such that each unique value for an attribute gets assigned either a 0 or a 1, resulting in a matrix like format [28].

To evaluate the hyperparameters used for the models, a grid search was used. The tool, also from the `sklearn.preprocessing` library, used for this was `GridSearchCV`. By defining all the hyperparameters in a grid, the function iterates through every combination producing a model with the optimal hyperparameter tunings. For the modeling, the scoring was set to `neg_mean_absolute_error`, which returns the model that minimizes the mean absolute error in minutes. The parameters used are presented in Section 5. In addition to that, the function also includes cross-validation that splits the training data into several subsets and trains on them separately. This ensures that the model is evaluated on different parts of the data, resulting in a more robust model capable of generalizing unseen data [29].

The free text attribute *Description* had varying values in terms of quality, sometimes including long descriptions of the incidents and sometimes just a few words. The decision was taken to generate a number of keywords from the descriptions and assign each accident with one of them, replacing the description attribute. The feature extraction technique used was *Term Frequency Inverse Document Frequency*, TF-IDF, which is a text classification method. The model essentially determines the relative frequency of words in a document and secondly measures how important this word is within all documents. The product of these two is the final score [30]. The problem encountered in the case of this thesis was

that the keywords generated were unevenly distributed among the accidents and did not tell much about the actual accident itself; hence, they were not correlating much to the duration. This led to the decision to not use keywords in the modeling.

To reduce the complexity for the modeling, important features for the XGBoost model were explored using the library `feature_importances_` which by default calculates how many times on average a feature is used to split the data among all the trees in the model [31]. A lower limit was set for the importance score, and only features with a score above the limit was used to retrain the model. However, no increase in accuracy or mean errors were achieved, leading to the decision to keep all attributes.

5 Results

This section presents the results of the two machine learning models used. The estimates produced will be compared against each other through the evaluation metrics; accuracy, MAE, MAPE, and R2. The aim of this section in to produce enough information to, in the last sections, discuss the research questions formulated in Section 1.2.1.

The results of the two models are presented together in table format as well as in plots. Since the main NTS data set presented in Table 1 included the estimates made by operators in the road traffic control centers, the estimates produced will also be compared against them. These estimates are presented as the model *Trafikverket* in Tables 7 and 8.

The two machine learning models used in this thesis were trained using grid search with a large combination of hyperparameters. In the following Tables 7 and 8 are the best performances of the two models with the corresponding hyperparameters presented. The scoring used in the grid search was `neg_mean_absolute_error` meaning that the parameters chosen are those that minimize the absolute error in minutes. Accuracy refers to the internal goal set by the road traffic control centers, preferably that the estimates are within +-30 minutes of the actual duration. The green filled cells refer to the best performance for that specific column.

5.1 End-time estimations in Stockholm

Table 7: Results for traffic accidents in Stockholm.

Model	**Hyperparameters**	**Accuracy**	**MAE**	**MAPE**	**R2**
Trafikverket	-	0.74	23.0	1.03	0.02
XGBoost	eta: 0.6 gamma: 0 max_depth: 5 min_child_weight: 5 max_delta_step: 1 subsample: 1 colsample_bytree: 0.6 reg_alpha: 0 reg_lambda: 0	0.81	19.9	0.74	0.19
SVR	C: 1 epsilon: 1 kernel: 'poly' degree: 4 gamma: 0.1	0.80	19.5	0.61	0.15

The results show that the estimates from the applied models perform better considering the chosen evaluation metrics, compared to those produced by Trafikverket. Trafikverket's internal goal of 80% is reached with the XGBoost model with an average absolute error of 19.9 minutes, which translates to a mean absolute error of 74%. The R2 score of 0.19 is not a particularly great fit, presented in Figure 13, but it is, however, better than Trafikverket's 0.02. Although, the estimates from Trafikverket ranged from 12 to 147 minutes, the XGBoost model predicted a duration span between 13.5 to 82.2 minutes, which can be seen in Figure 12. The median estimate done by the XGBoost model was 41.5 minutes, which is very close to Trafikverket's median of 45 minutes. Using XGBoost's `feature_importance_` the most important features for the model were *Road closed_yes*, *SOS on scene_yes*, and *Near center_yes*. The least important attributes were different values for *Road number*. In the context of weather attributes, the most significant factor for the model was *Weather_snow*, which ranked 20th out of 221 features in terms of importance. This suggests that the duration has a very slight correlation with snowy weather, likely due to its impact on visibility or road conditions.

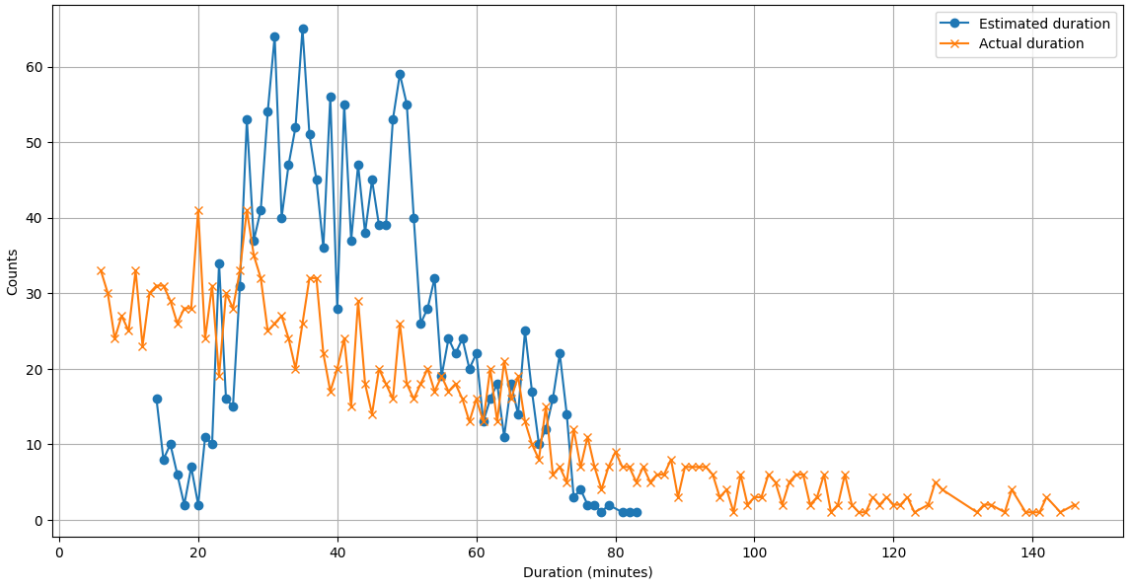


Figure 12 Estimated durations from the XGBoost model for traffic accidents in Stockholm compared to actual durations.

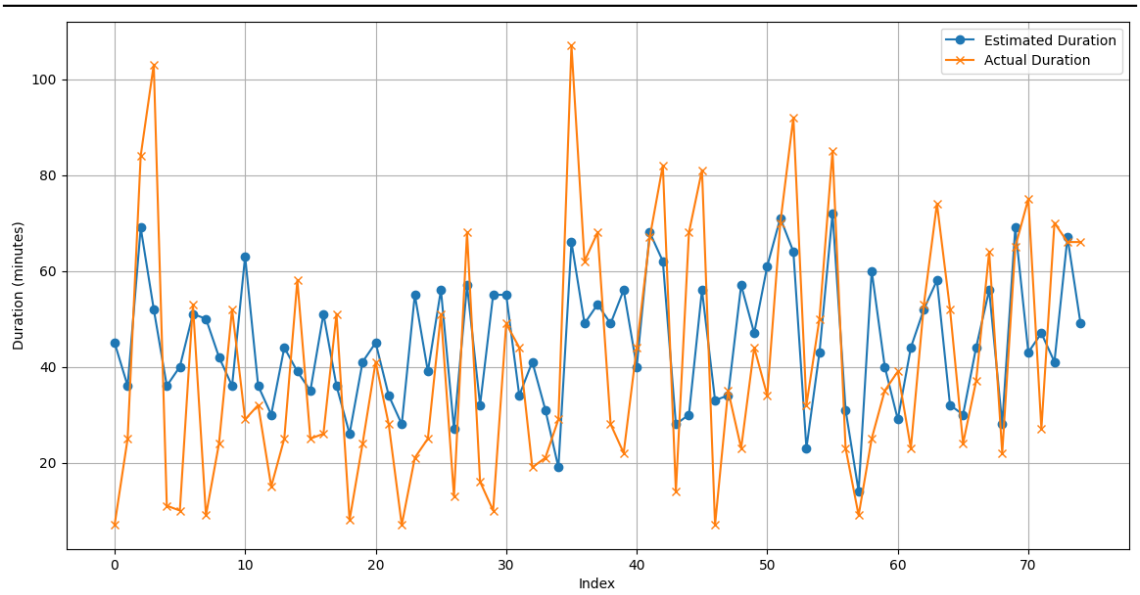


Figure 13 XGBoost model performance for 75 randomly selected traffic accidents in Stockholm compared to the actual durations.

The SVR model performed better on mean absolute errors, probably because the median estimate was lower than that of the XGBoost model at 36 minutes. The estimates from the SVR showed a duration range between 6.6 to 78.9 minutes which can be seen in Figure 14, compared to XGBoost it is lower for both sides of the interval. XGBoost still outperforms both models evaluated in respect to R2. A visualization of how well SVR performs in contrast to the actual durations can be seen in Figure 15.

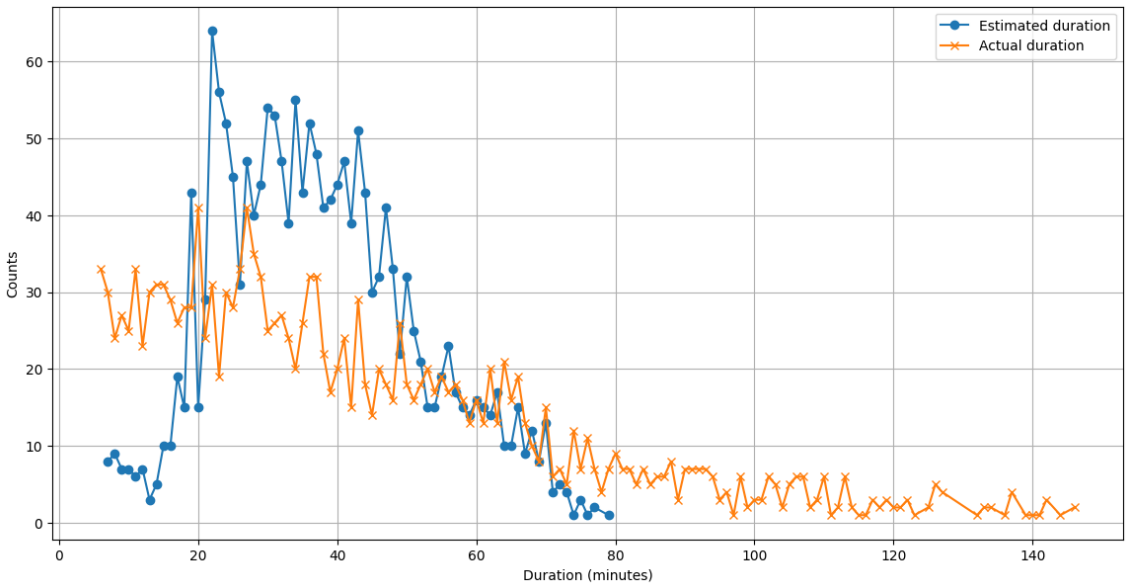


Figure 14 Estimated durations from the SVR model for traffic accidents in Stockholm compared to actual durations.

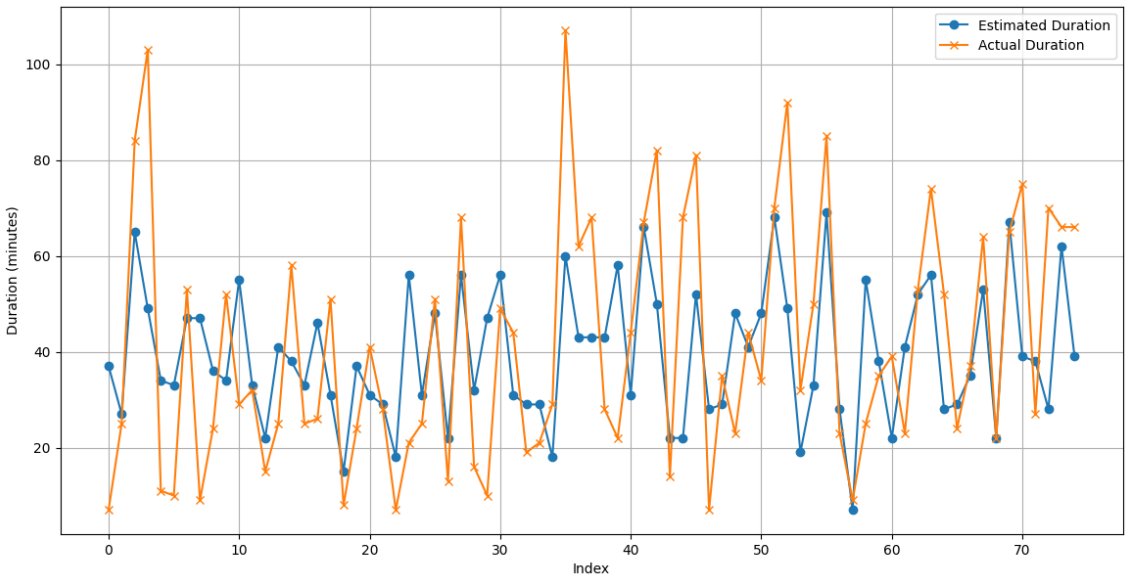


Figure 15 SVR model performance for 75 randomly selected traffic accidents in Stockholm compared to the actual durations.

5.2 End-time estimations in Skåne

Table 8: Results for traffic accidents in Skåne.

Model	**Hyperparameters**	**Accuracy**	**MAE**	**MAPE**	**R2**
Trafikverket	-	0.62	28.6	0.90	-0.28
XGBoost	eta: 0.2 gamma: 0.1 max_depth: 5 min_child_weight: 5 max_delta_step: 3 subsample: 1 colsample_bytree: 0.6 reg_alpha: 1 reg_lambda: 0	0.74	22.0	0.53	0.18
SVR	C: 50 epsilon: 0.01 kernel: linear degree: 2 gamma: 'scale'	0.75	21.5	0.48	0.19

The estimated durations for traffic accidents in Skåne are also better than the estimates from Trafikverket, considering the chosen evaluation metrics. The goal of an accuracy of 80% is not reached by either model. However, this goal was not set by the operators in

Skåne, since it was an internal goal for Stockholm. For XGBoost, the minimum estimate was 8.3 minutes, while the maximum was 118.1 minutes, which can be seen in Figure 16. In Figure 18 a more narrow duration span of estimates between 6.2 minutes and 84.7 minutes can be seen for the SVR model, which was also the case for estimates in Stockholm. The median estimate was 45 minutes for XGBoost and 58 minutes for SVR compared to Trafikverket's median of 74 minutes. In terms of evaluation metrics, the XGBoost model outperformed Trafikverket with a mean absolute error of 22.0 minutes or 53% compared to 28.6 minutes or 90%. However, SVR had the best overall performance, barely outperforming the XGBoost model. Different configurations of the grid search were performed with the linear kernel, resulting in the highest R2. However, the relatively low R2 scores are higher than those of Trafikverket. In Figures 17 and 19 the models' ability to follow trends and explain variance are visualized. The three most important features for the XGBoost model were *Road closed_yes*, *Incident text_unprotected accident scene*, and *SOS on scene* while the least important was *weekday_yes*. In the context of weather attributes, accidents in Skåne tend to be more correlated to them compared to accidents in Stockholm. For example, the attribute *Weather_Rain* placed 11th out of 567 on the list of most important features, indicating at least some correlation between durations and weather conditions.

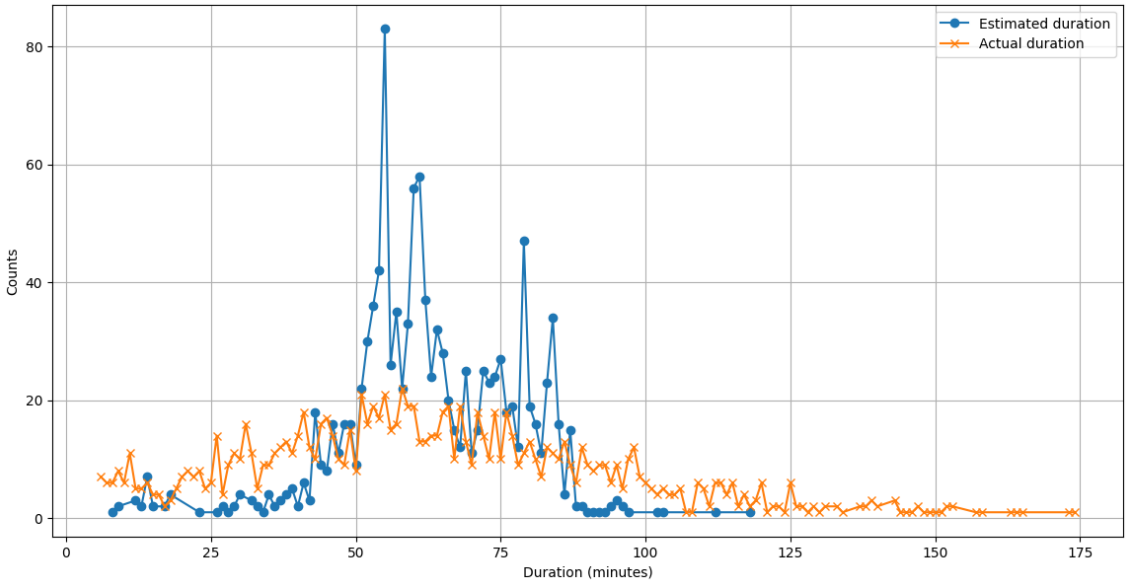


Figure 16 Estimated durations from the XGBoost model for traffic accidents in Skåne compared to actual durations.

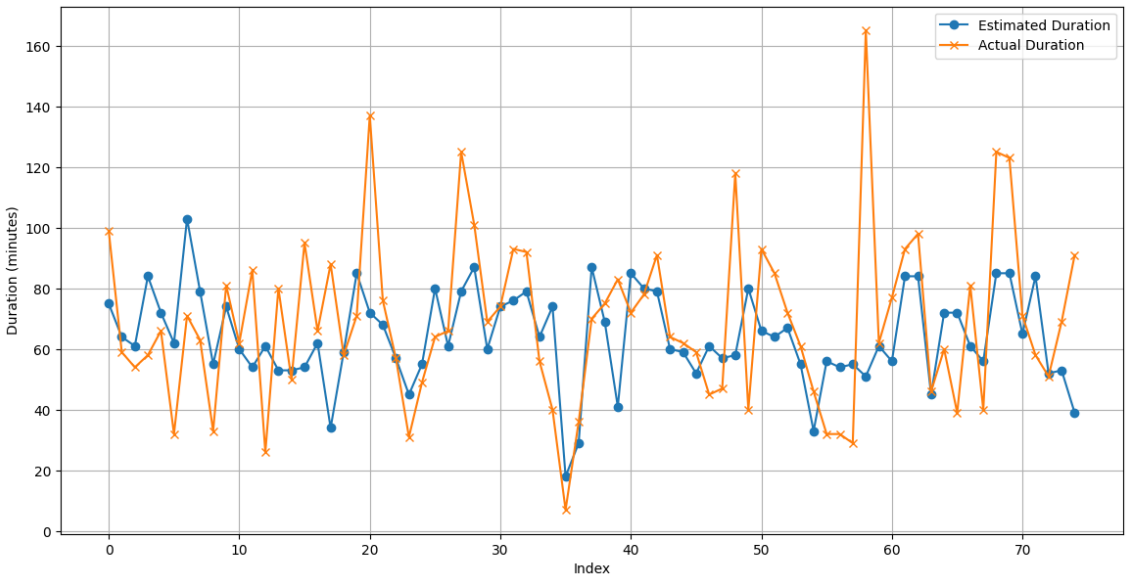


Figure 17 XGBoost model performance for 75 randomly selected traffic accidents in Skåne compared to the actual durations.

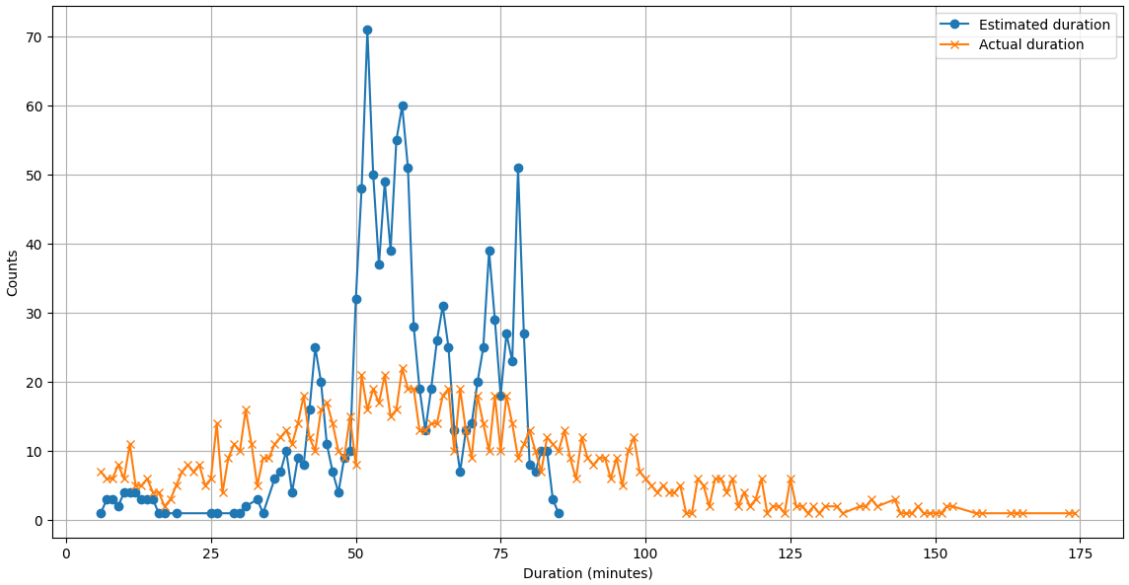


Figure 18 Estimated durations from the SVR model for traffic accidents in Skåne compared to actual durations.

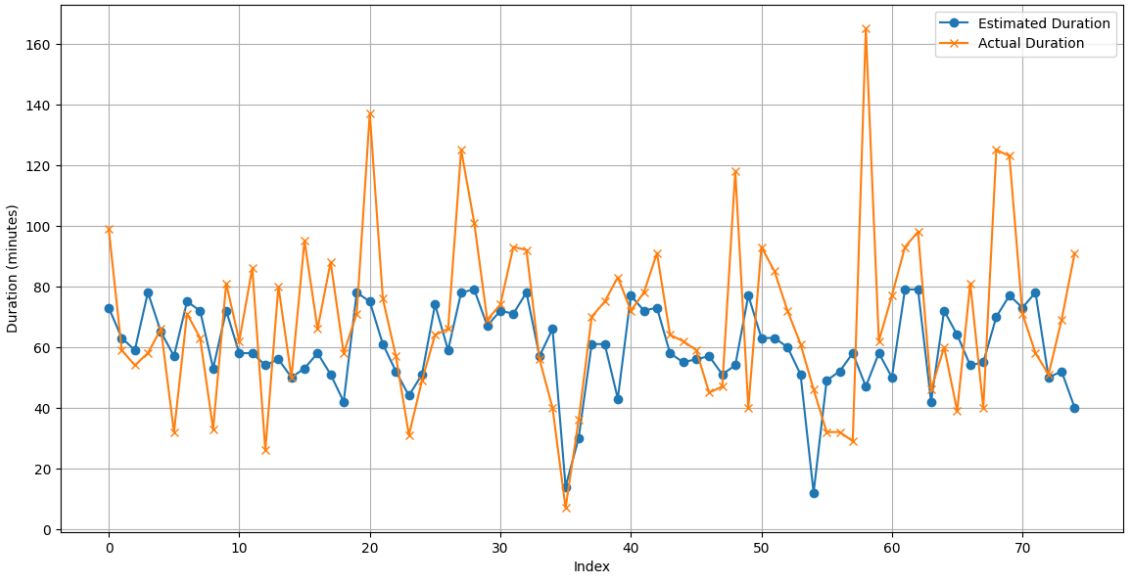


Figure 19 SVR model performance for 75 randomly selected traffic accidents in Skåne compared to the actual durations.

To conclude the results, it seems that the proposed models produce more accurate predictions than Trafikverket. By reviewing the Tables 7 and 8 one can see that on all evaluation metrics measured, the applied models outperform the manual estimates made by the operators at the road traffic control centers in Stockholm and Skåne. Although the models produced estimates on a much narrower span than Trafikverket, the median prediction at least in Stockholm was similar across all models, including those of Trafikverket. Another important difference is that Trafikverket tends to overestimate their durations, where 69% of durations were higher than the actual ones in both Stockholm and Skåne. Compared to the XGBoost model that only overestimated 57% of the time in Stockholm and 51% in Skåne, it is significantly higher. When comparing the different machine learning models, they generally tend to produce similar results in terms of accuracy. Since both models were trained on the same preprocessed data, it is unclear how the results would differ if more sparse data was used, but it is likely that XGBoost would outperform SVR in that case. Another factor in favor of XGBoost is its efficiency. During the model training process, it was observed that the XGBoost model completed the training process faster than the SVR model.

6 Discussion

This section presents a discussion of the entire process, from data analysis to machine learning modeling. The aim is to nuance the results and explain what and how things could have been done differently, why the results look like they do, and how future research can gain by taking part of this work. The section is introduced with a discussion of data quality, followed by discussions of the machine learning models, results, limitations, and ultimately future work.

6.1 Data quality

High data quality is essential for machine learning. It helps models become more consistent, accurate, and robust, whilst the opposite can negatively influence models by skewing predictions in unnatural ways. The issues of the data for this thesis range from low to high in respect to impact on modeling. They relate to many factors including lack of standardization, missing data, false data, anomalous data, and under-dimensioned data, all discussed further in the following subsections.

6.1.1 Non-standard data

Some attributes in the NTS data are either standardized or binary in the operator registration entry. Some of these standardized attributes can still be subjectively interpreted. For example, the attribute *Severity* which has the options 'none', 'low', 'high', and 'very high'. Individual operators have their own interpretation of how these are scaled. These subjective patterns cannot be traced, and therefore this fact is completely ignored in the preprocessing and modeling for this thesis.

Description is another non-standardized attribute. It is entered with freely expressed text. That is unfortunate, as it could be argued that it has a significant relation to the duration of an accident, as it focuses explicitly on the accident itself and not on its consequences. After multiple attempts to derive information from this attribute through keyword extraction, it was found to be relatively useless since no real connections could be made. Its non-standardized phrasing meant that it could not be included as an independent data attribute for the machine learning modeling. The description field only facilitated the creation of the binary attribute *Tunnel*, which was believed to have at least some relation to the duration of the accident. Trafikverket wants to prioritize these accidents, as congestions in tunnels hinder the emergency response that helps alleviate and solve the situation, as tunnels tend to lack space. Other than that, no valid options containing any already used information were to be extracted. Misspellings and different phrasing further complicated the issue of keyword extraction.

6.1.2 Missing data

One aspect that makes predicting durations for accidents difficult is the fact that a prediction is only useful if it is made quickly after an accident happens. For this thesis, the *Secure Lead Times* 12 and 20 minutes were used. This means that an accident response is still developing and that in many cases a complete picture of the scene is yet to be achieved. For the preprocessing, this meant that many accidents were filtered out as critical attribute fields were empty.

Filtering missing data can also be traced to the idea that different operators at the road traffic control centers have different ideas of what is essential when it comes to data registration. A non-critical accident on a county road may not receive the same attention to detail in the data entry to NTS compared to a major traffic impeding event in urban Stockholm or Malmö. This means that data during preprocessing might be filtered away due to lack of detail during registration.

6.1.3 False data

The possible existence of false data is troublesome for machine learning purposes. There are various types of poor data that may conflict with the machine learning modeling for this thesis. While some can be addressed directly, others are more challenging to identify and can, to an unknown extent, hinder the models' performance. Following are some handled or at least noted patterns of misregistered and anomalous data.

Misregistered data directly deteriorate the algorithm's ability to represent reality, making the algorithm less accurate as well as more difficult to interpret. Finding and solving these entries has been ongoing since the beginning, and it would be bold to claim that all misregistered data has been handled. For instance patterns in how attributes directly contradict others were observed.

An example is when the attribute *Road closed* was set to 'yes', the attribute *Number of lanes* was always set to '0'. This contradictively meant that no lanes were affected, despite the road being closed for road users. This was solved by translating the *Number of lanes* to '1' if *Affected direction* was set to 'One', or '2' if it was set to 'Both'. This interpretation simplified the underlying nature of the data, as it made the assumption that all roads affected by this anomaly had exactly one lane going each direction. Solving this properly would be very time-consuming and was not a viable option. However, it is safe to say that the data quality improved.

Another misregistration of data was the pattern of classifying accidents involving heavy vehicles as the more general 'accident', instead of 'accident involving heavy vehicle'. This was observed through manual inspection of the data, as the *Description* often included information of heavy vehicles being involved, often directly contradicting the respective *Incident text* as this could be registered as an accident (not involving a heavy vehicle).

The solution was to change the *Incident text* entry to 'accident involving heavy vehicle' by using the *StringSearcher* transformer in FME. This transformer looked for words relating to heavy vehicles, such as "truck", "crane truck", "snowplow truck" etc. in the *Description* attribute, and subsequently reclassified the respective values in *Incident text* according to these. This solution is based on the assumption that the *Description* information is more reliable in these cases than the *Incident text*. Furthermore, it was deemed more important to classify accidents with heavy vehicles than without. The three reasons behind this are that *Description* was not used in the modeling, accidents involving heavy vehicles can drastically change the overall duration to handle a "general" accident and accidents involving heavy vehicles are more scarce, meaning this solution would produce more data of such type.

The Covid period is a factor that potentially could harm the models' performance. Some of the attributes are indirectly related to the amount of traffic. For example, the attribute *Road Closed* might not have been deployed to the same extent since traffic was lighter, rendering such a significant response unnecessary. The same goes for *Severity*, as events in general might have been less traffic impeding. Further the *Actual duration* could be affected due to the lighter traffic flow, meaning that the emergency response vehicles could reach and handle accidents more effectively. It is difficult to determine to what extent Covid actually skewed the data, however, it might be of interest to note since the modeling uses time series data.

One of the most important phenomena is the fact that many estimates, 11% in Stockholm and 15% in Skåne are perfectly estimated to their actual durations. This was discussed with the operators, which pointed to the fact that many accidents in rural areas lack update communication from emergency responders, which means that operators simply close an event right as the estimated time is reached. After further inspection, this tendency of closing events was in fact not only common in rural areas but also frequently observed in urban areas. Even if the reason behind the phenomenon remains unclear, the solution of simply removing them was deemed satisfactory, even if this meant also removing a few valid perfect estimates made by skilled operators. The case of accidents in rural areas being closed right at the estimated time can be further problematized. Even though operators receive alerts for ongoing incidents, it seems unlikely that the operators always pay attention to the closure of an event at the exact point of time when the estimated duration is reached. This means that there are an unknown number of data entries that are probably closed right before or right after the estimate is reached, meaning that an operator closes it whenever they physically focus on the task at hand. Therefore, estimates that are very close to their actual duration can be argued to be less reliable than estimates that are off. Estimates close to their actual durations were, however, kept, since the reason behind potentially removing them was too speculative.

6.2 Data limitation solution using IQR

Limiting the data set in terms of the actual durations included in the modeling was necessary. The reason is that the relationship between the single attributes and the duration of an accident was relatively low, and it would be impossible for the model to separate similarly described accidents in the NTS data that had large differences in their duration. The following Table 9 presents an example taken from the data set.

Table 9: Example of two accidents in Stockholm with similar attributes but very different durations.

Incident text	**Road closed**	**Affected direction**	**SOS on scene**	**Actual duration (minutes)**
Heavy vehicle	yes	both	yes	1679
Heavy vehicle	yes	both	yes	20

As seen in Table 9 above, it is impossible to distinguish the two accidents and conclude the reason why one accident takes more than a full day to handle, while the other takes only 20 minutes, when considering the available data. Note also that the attributes in the table are the most important for the algorithm and have the highest relation to duration as described in Section 5. This phenomenon is a major problem for this thesis when applying machine learning, as it only confuses the algorithm. By applying IQR onto *Actual duration* in the data set, the idea was that a smaller yet interesting duration span could support modeling, since it would be easier for a machine learning model to find general patterns made on denser data. This was done to minimize similarly described data on both sides of the spectrum. It should be noted that there might be other attributes that actually separate the given examples. However, pointing to the fact that those in general had extremely low relation to duration, it is doubtful that these would improve the modeling much when including a larger duration span.

Another problem with the data can be sourced from the contradictive intuitive nature of some data. Thankfully, it seems fairly common for accidents involving heavy vehicles to be handled with great efficiency. Having homogeneous data in mind, this makes it challenging for the algorithm to decide when to raise the prediction for a duration. This might be an explaining factor when discussing the fact that the best performing algorithm for Stockholm data, namely XGBoost, does not predict any higher value than 82.2 minutes when the underlying actual duration can reach 147 minutes.

A test involving applying IQR on actual durations for the attribute *Incident text*'s respective categorical values was also carried out. This was believed to increase the amount of data available for different incident texts, which it did. This is because different incident text values, in general, had different duration spans, and applying IQR to the *Actual duration* in the data set meant some incident types almost completely disappeared as they

were found in the higher duration spans. Despite the fact that more data was available in the different types of accidents, the performance of any given machine learning model did in fact not improve. The reason for this, is difficult to assess at the moment.

6.3 STRADA data

It was argued that the low relation between single attributes and duration could be derived in part from the reason described in Section 6.2. Furthermore, one can argue that the attributes do not explain durations well, as they do not focus on the accidents themselves, but rather on the consequences of them. If roads are closed, lanes are closed, traffic is partly impeded, etc., does not explain how long it will take to handle an actual accident. With this in mind, the emergency response database STRADA appeared to be a great fit to complement the data used with more detailed information on accidents themselves. Emergency personnel provide more in-depth descriptions of any humanitarian and vehicular damage that has occurred. Unfortunately, it quickly became apparent that STRADA could not be used for this purpose. The main reason is that the data in STRADA is registered after an accident has been handled, rendering it useless for a predictive model. For instance, data registered by police are only saved to STRADA after they close an errand. Furthermore, STRADA and NTS do not share any common feature or attribute on which to relate their records, rendering it useless in the aim of this thesis. Since no other similar option to STRADA was found, the models were created without any additional data that better described the accidents. It is important to note that future similar projects would probably gain a lot from having more in depth information on actual accidents.

6.4 Model and result discussion

In this section, the created machine learning models will be discussed, from their performance metrics, dynamic nature, hyperparameterization, and reliability. This will then be compared to both the benchmark model, i.e., Trafikverket's estimates, and a naive model which only estimates the median value of the *Actual duration*.

The main concern in modeling was balancing the algorithms to obtain adequately dynamically adapted models without either underfitting, by avoiding high bias, or overfitting, by avoiding high variance. Multiple methods were applied to avoid overfitting, such as utilizing cross-validation, where the model is trained on different subsets of the entire training data set. Other methods include the use of regularization parameters in the XGBoost model, as well as feature selection, which, however, did not improve the model performance. Lastly, a larger data set could have been used, enabling the model to better generalize the data and find patterns more easily.

Instead, it was more challenging to avoid underfitting, as the models produced relatively conservative predictions, as seen in Figures 13 and 17. This can be argued to be due to

the fact that the attributes do not provide sufficient information about the target variable, *Actual duration*, since they have very low correlation with it. The nature of the data, thereby, makes the models conservative by default. As explained in Section 1.3 the tendency of narrow and conservative predictions was also discovered by Hampton Roads Traffic Operations Center when using iMiT OLS, which also had difficulty predicting longer durations.

By using `GridSearchCV` from the scikit-learn library, different arrays with hyperparameter tunings were processed to evaluate the optimal combination of hypertunings. Having no access to high computational power, the hyperparameter tuning candidates were limited to the values seen in listing 4. Additional tests, not documented here, showed that the models were sufficient enough and the performance did not improve noteworthy when testing other hypertunings. However, not using hypertuning as such worsened performance.

The next two subsections discuss these settings for the applied XGBoost and SVR models used for Stockholm and Skåne.

6.4.1 Predictive Models for Stockholm

For Stockholm, it is debatable which model of XGBoost and SVR performed the best. One thing that stands out is the models lowest predictions. Both were trained on the same data where the shortest *Actual duration* was 6 minutes. SVR's lowest prediction was just above 6 minutes, while XGBoost had a more conservative lowest estimate of around 13 minutes. The most interesting performance key values are MAPE and R2, where SVR had a lower MAPE by a factor of 0.21 and XGBoost had a higher R2 by a factor of 0.27. Both outperformed Trafikverket's current default prediction method by far in all categories. A possible explanation to this is that the grid search was set up to choose the parameters where the mean square error was minimized. Combined with a relatively short interval, this results in the models opting to predict close to the median.

The hyperparameterization of XGBoost using `GridSearchCV` can be observed to create a model with relatively drastic tuning, where some hyperparameter values indicate overfitting while others indicate underfitting. An `eta` of 0.6 means a relatively high learning rate, which can lead to overfitting. `gamma` of 0 means that there are no constraints for tree splits based on loss reduction. This generally tends to overfit models. However, using a `max_depth` of 5, makes sure the tree is not deep enough to overfit, instead it only gets wider from the low gamma tuning. The `min_child_weight` being set to 5, also helps the tree becomes more conservative, as it sets the lower limit for when a leaf node should be partitioned. The parameter `max_delta_step` was set to 1, which helps with handling imbalanced data and helps the model find patterns in the minority classes, that is instances in the data set with low coverage. `Subsample` and regularization parameters were tried but did not improve the performance.

The hyperparamerization of SVR using `GridSearchCV` found an optimal solution in a dynamic model. The regularization parameter `C` being 1, is not very aggressively punishing small errors and allows the model to be fairly attentive to the training data, i.e., it can be argued to be balanced. `Epsilon` of 1, is a fairly high value, which means that the model allows quite large deviations from the actual target value without incurring a penalty. It makes the model focus on larger trends in the data, which is a good thing when referring to the ability to generalize, of which downside can be found in the risk of underfitting. A `kernel` and `degree` chosen as polynomial with degree 4, makes the model able to capture quite complex, nonlinear data patterns. Using a higher degree always poses a risk referring to overfitting, however, after using both techniques mentioned in Section 6.4 and listed hyperparameter tunings, the model is evidently not overfitted.

6.4.2 Predictive Models for Skåne

Skåne was an even more challenging modeling task than Stockholm. For a start one can observe that for the operators in Skåne the lead time of 20 minutes is notably higher than the average of Stockholm's 12 and 20 minutes lead times, and despite this, they contradictively produce worse estimates. With an accuracy of 62%, MAE of 28.6 minutes, MAPE of 90% and a negative R2 score of -0.28, the prediction task is evidently very complicated. SVR performed slightly better at predicting accident durations than XGBoost in all categories, which is troublesome when looking at the hyperparameter tuning of SVR. `C` set at 50 while having an `epsilon` value of 0.01, means that the margin for errors where no penalty is given is very narrow, while the penalty for values outside this margin are severely penalized. This means that the model tries to fit the training data very closely. At the same time, a linear `kernel` regression function is applied, meaning the relationship between the target variable *Actual duration* and the input features is assumed to be linear. Knowing that the data are not linear with respect to the output variable, this optimal regression function seems ridiculous. All in all, the parameter tunings for margin and penalizing errors are very prone to overfitting, however, this is impossible considering that the linear regression function is applied onto nonlinear data. The model resulted in an accuracy of 75%, MAE of 21.5 minutes, MAPE of 48%, and R2 of 0.19. Especially MAE, MAPE, and R2 are important for concluding that SVR can generate a much better prediction output than the current method. Comparing the hyperparameters of XGBoost for Skåne to those for Stockholm, the most evident differences are *eta* which is lower in Skåne at 0.2 meaning more boosting rounds are required, in turn reducing overfitting. Another difference is that regularization is used in the XGBoost model for Skåne, which indicates that the model attempts to prevent overfitting while still keeping significant attributes.

It is difficult to understand why modeling in Skåne was so much more difficult than for Stockholm. It can be related to many different things. The geographical locations of accidents in Skåne are much more evenly distributed over the entire county, while Stockholm

have the majority of the accidents inside the municipality of Stockholm. In addition to this, Stockholm has excellent camera surveillance of the streets inside Stockholm, which means that the data can be more accurate, gathered more quickly, and registered in the NTS database. In addition, the accident duration span in Skåne is broader, meaning more pressure onto the model's ability to produce dynamic output. Given the poor data quality, this dynamic ability is evidently difficult to obtain.

6.4.3 Naive model

If a model were made to solely produce estimates set at the median value of *Actual duration* in Stockholm, that is, 36 minutes, the accepted *Actual duration*s would fit in the duration span of 6 to 66 minutes according to Trafikverket's goal. Given that 7201 out of 8919 entries in the processed data set for Stockholm have an *Actual duration* within this duration span, a static estimate for any given accident set at 36 minutes would produce an accuracy of 80.7%. Even if this number seems relatively satisfactory, it is important to note that it is static, thus meaning that all *Actual duration*s that are outside of this span will be mispredicted with very high errors. This becomes obvious when other metrics are introduced. A naive model using the median estimate producing the accuracy of 80.7%, would actually result in an R2 score of -0.06. A negative R2 score indicates that the predictions are not following the real underlying data and, as matter of fact, it even shows that they actually represent the real pattern negatively. The median estimate in its static nature thus provides a high accuracy, but it does not represent the accident durations at all, and is therefore not viable. Trafikverket currently employs a suggestion based on the mean value of the *Actual duration* in both Stockholm and Skåne. This results in relatively good accuracies, but in a similar manner, the R2 scores being close to 0, indicates that the default pre-filled predictions do not actually represent any variance in the data.

Altogether, it would seem suitable for the operators to continue being able to update their estimates manually, since the manual estimate means that they can make use of significantly more information compared to what is entered into NTS. However, the underlying default estimate could benefit from a more dynamic tool. As the results show, for accidents that fit the duration spans of 6 to 147 minutes for Stockholm and 5 to 174 minutes for Skåne, both XGB and SVR can improve the current method not only for achieving a higher accuracy, but also for a lower mean average error, a smaller mean average percentual error, and an R2 that actually to some extent represents the variance in the data.

6.5 Future work

The concept of using machine learning to estimate the end-times of traffic accidents has been proven to be successful. Still, with the current generally positive results, there is also

room for improvement and, of course, a full implementation of a system.

During the project, it has been discovered that there are some attributes that would have greatly impacted the performance of the models. For example, an attribute explaining the traffic density on minute basis for each accident. An attribute like this would give insight in how quickly emergency personnel could reach the scene of the accident, in turn extending the time before the traffic flow could return to normal. Another attribute would be lane closure, which is one of the most important attributes that Tang, J et al. discovered to impact the duration. Although NTS has an attribute for how many affected lanes, it says nothing about the proportion of lanes closed. For example, the closing of one lane on a six-lane motorway would have less impact than closing one lane on a road with a total of two lanes. Another important attribute also discovered by Tang et al. was the type of accident. Once again, the attribute exists in the data set used but could be more descriptive, for example, by including other types like: collision, number of vehicles, and insensitive information of injuries. Another obvious improvement is the data quality that was thoroughly discussed earlier. In fact, issues regarding data quality and complexities following high-dimensional data sets were found to be key challenges in earlier research by Grigorev et al. Suggested improvements in line with Grigorev include standardized attributes and integrated anomaly detection techniques.

Implementing such a prediction system in reality would certainly be a lot of work. The system would have to be very fast to generate an estimate within the set *Secure Lead Time*. In addition to that, the system would have to be easy to use and react on the input from the operator while also being able to handle missing values without compromising the functionality. To narrow down the test space, a similar approach to the iMiT model could be used where the operator continuously updates the information, resulting in the model generating a new prediction from the best-performing model.

7 Conclusions

This thesis has investigated the possibility of, based on historical data, generating end-time estimates for traffic accidents in Stockholm and Skåne. As presented in Section 5, it is possible to do so, while also outperforming the manual estimates set by the operators at the traffic control centers. The largest difference between the predictions produced by the models proposed in the thesis, compared to Trafikverket's estimations was the duration span. The proposed models generated more conservative predictions while Trafikverket had predictions ranging over the entire span for *Actual duration*. When looking at evaluation metrics, the overall largest takeaway is that the accuracy within +-30 minutes is relatively constant among all models, including Trafikverket's, both in Skåne and Stockholm. However, the models still outperform previous estimations, with XGBoost being the preferred in Stockholm and SVR in Skåne. The models proposed in this thesis stands out, especially in the ability to explain the variance in the output. While Trafikverket barely exceeds an R2 score of 0, the models reach around 0.2 which is not great, but still an improvement over previous predictions. As seen in Figures 6 and 7, the most common estimate by Trafikverket is 39 minutes in Stockholm and 63 minutes in Skåne, however there are predictions on the entire span set by the IQR. The XGBoost and SVR models did not produce such spikes, although the duration span is narrower, hence resulting in more conservative predictions.

Section 6.5 clearly highlights that enhancements are necessary before such a system could be implemented in reality. Improvements in terms of data quality explained in Section 6.1 would be absolutely necessary, as well as incorporating new features closer correlated to the durations. As discussed earlier, traffic density can have a significant impact on the speed with which emergency vehicles arrive at the scene, in turn relating to the duration of the actual accident. Such attribute would must probably have positively impacted the result of this project and is something to consider when collecting accident data in the future.

In essence, the findings of this thesis represent a significant step towards a more effective traffic accident management system. With continued research, development, trial and error, the vision of more accurate, reliable, and efficient end-time estimates for traffic accidents can become reality, ultimately contributing to a safer and more efficient traffic flow.

References

[1] Trafik Stockholm, "Trafik stockholm," PowerPoint presentation, February 2025, presented at field trip at Stockholm Traffic Control Center 2025.

[2] Trafikverket, Correspondence with Alexander Nilsson and Rodrigo Marquez-Lucero, February-May 2025, email conversations and meetings.

[3] Trafik Skåne, "Trafikledningen," Interview with Maria Jönsson by Gabriel Martens and Hugo Asztély, March 2025, teams meeting.

[4] Trafik Stockholm, "Trafikledningen," Interview with Raimo Nilsson by Gabriel Martens and Hugo Asztély, February 2025, field trip.

[5] Transportstyrelsen. (2025) Om olycksdatabasen strada. Fetched: 2025-05-07. [Online]. Available: https://www.transportstyrelsen.se/strada

[6] A. Grigorev, A.-S. Mihăiţă, and C. Feng, "Traffic incident duration prediction: A systematic review of techniques," *Journal of Advanced Transportation*, vol. 2024, pp. 1–36, 2024.

[7] M. J. Page, D. Moher, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, and J. Glanville, "Prisma 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews," *The BMJ Group*, vol. 372, p. 1, 2021.

[8] J. Tang, L. Zheng, C. Han, W. Yin, Y. Zhang, Y. Zou, and H. Huang, "Statistical and machine-learning methods for clearance time prediction of road incidents: A methodology review," *Analytic Methods in Accident Research*, vol. 27, p. 100123, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213665720300130

[9] A. J. Khattak, X. Wang, and H. Zhang, "Incident management integration tool: dynamically predicting incident durations, secondary incident occurrence and incident delays," *Iet Intelligent Transport Systems*, vol. 6, pp. 204–214, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:110790936

[10] SMHI. (2025) Ladda ner meteorologiska observationer. Fetched: 2025-02-05. [Online]. Available: https://www.smhi.se/data/meteorologi/ladda-ner-meteorologiska-observationer/airtemperatureInstant/98210

[11] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön, *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press, 2022. [Online]. Available: https://smlbook.org

[12] P. P. Deka and J. Weiner, *XGBoost for Regression Predictive Modeling and Time Series Analysis - Learn how to build, evaluate and deploy predictive models with expert guidance*. Packt Publishing, 2024.

[13] F. Zhang and L. J. O'Donnell, "Chapter 7 - support vector regression," in *Machine Learning*, A. Mechelli and S. Vieira, Eds. Academic Press, 2020, pp. 123–140. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128157398000079

[14] scikit learn. (2025) Svr. Fetched: 2025-03-27. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[15] M. Awad and R. Khanna, *Support Vector Regression*. Apress, Berkeley, CA, 01 2015, pp. 67–80.

[16] Scikit-learn. (2025) Svr. Fetched: 2025-04-15. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016, pp. 785–794.

[18] DMLC XGBoost. (2025) Xgboost parameters. Fetched: 2025-04-09. [Online]. Available: https://xgboost.readthedocs.io/en/stable/parameter.html

[19] C. Barnham, "Quantitative and qualitative research: Perceptual foundations," vol. 57, no. 6, 2015, pp. 837–854. [Online]. Available: https://doi.org/10.2501/IJMR-2015-070

[20] IBM. (2025) What is predictive analysis. Fetched: 2025-02-14. [Online]. Available: https://www.ibm.com/think/topics/predictive-analytics

[21] GeeksForGeeks. (2024) What is predictive analysis. Fetched: 2025-02-14. [Online]. Available: https://www.geeksforgeeks.org/what-is-predictive-modeling/

[22] Sweco Sverige. (2025) Fme platform. Fetched: 2025-02-14. [Online]. Available: https://www.sweco.se/vart-erbjudande/digitala-losningar/geografisk-information/fme-platform/

[23] Google Colaboratory. (2025) Google colaboratory. Fetched: 2025-02-14. [Online]. Available: https://colab.google/

[24] C. Mallet, M. Zribi, and N. Baghdadi, "Introduction to qgis," in *QGIS and Generic Tools*. United States: John Wiley Sons, Incorporated, 2018.

[25] OpenStreetMap. (2025) Openstreetmap förser tusentals webbsidor, mobilappar, appar och fysiska apparater med kartdata. Fetched: 2025-05-06. [Online]. Available: https://www.openstreetmap.org/about

[26] G. Gowthami and S. S. Priscila, "Classification of intrusion using cnn with iqr (inter quartile range) approach," in *Advancements in Smart Computing and Information Security*, S. Rajagopal, K. Popat, D. Meva, and S. Bajeja, Eds. Cham: Springer Nature Switzerland, 2024, pp. 259–269.

[27] Scikit-learn. (2025) Standardscaler. Fetched: 2025-03-20. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[28] Scikit-learn. (2025) Onehotencoder. Fetched: 2025-03-24. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

[29] Scikit-learn. (2025) Gridsearchcv. Fetched: 2025-03-24. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[30] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1. Citeseer, 2003, pp. 29–48.

[31] DMLC XGBoost. (2025) Python api reference. Fetched: 2025-04-10. [Online]. Available: https://xgboost.readthedocs.io/en/latest/python/python_api.html#xgboost.XGBRegressor

A Python Scripts

Below are some code snippets from the modeling in Python.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.svm import SVR
from sklearn.metrics import r2_score
from math import sqrt
```

Listing 1: Necessary imports in Python.

```
#Defining Q1 and Q3 for traffic accidents in Skåne (df_sko)
Q1 = df_sko['Fastställd varaktighet'].quantile(0.25)
print("25 % av datan har lägre fastställd varaktighet än", Q1)
Q3 = df_sko['Fastställd varaktighet'].quantile(0.75)
print("75 % av datan har lägre fastställd varaktighet än", Q3)
IQR = Q3 - Q1
lower_bound = Q1 - 2 * IQR
upper_bound = Q3 + 2 * IQR

#Setting the upper limit with the bound above
df_sko = df_sko[(df_sko['Fastställd varaktighet'] >= lower_bound) & (
    df_sko['Fastställd varaktighet'] <= upper_bound)]
df_sko.shape
```

Listing 2: Setting the IQR limits for the data set with traffic accidents in Skåne.

```
#Defining input and output for traffic accidents in Stockholm (df_sto)

X = df[['Väder','Händelsetext', 'Påverkan', 'Påverkad trafikriktning',
    'Väg avstängd', 'Utbredningsriktning länk', 'Antal körfält', 'SOS p
    å plats','Vägassistans på plats', 'Månad', 'Helgdag','Tid på dygn',
    'Vägnummer', 'Nära centrum', 'Tunnel', 'Huvudväg']]

y = df['Fastställd varaktighet']

#Defining categorical and numerical attributes

categorical = ['Väder','Händelsetext', 'Påverkan', 'Påverkad
    trafikriktning', 'Väg avstängd', 'Utbredningsriktning länk', 'SOS p
    å plats', 'Vägassistans på plats','Månad', 'Helgdag','Tid på dygn',
    'Vägnummer', 'Distans centrum', 'Tunnel','Huvudväg']

numerical = ['Antal körfält']


#Creating a columntransformer for the encoder and the scaler

preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical),
        ('cat', OneHotEncoder(), categorical)
    ])

X = preprocessor.fit_transform(X)
```

Listing 3: Defining the input and the output for the machine learning models as well as the encoding and scaling the attributes. The attributes differ slightly between Stockholm and Skåne as explained earlier.

```
#Split data in train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.2, random_state=42)

model = xgb.XGBRegressor()
#model = SVR()
model.fit(X_train, y_train)

#Defining the hyperparameter grid for XBGRegressor

param_grid = {
    'eta': [0.1, 0.2, 0.6],
    'gamma': [0, 0.1, 0.2],
    'max_depth': [5, 7, 9],
    'min_child_weight': [5, 7, 10],
    'max_delta_step':[1, 2, 3],
    'subsample': [0,8, 1.0, 1.2],
    'colsample_bytree': [0.6, 0.8, 1],
    'reg_alpha': [0, 0.1, 0.5],
    'reg_lambda': [0, 0.1, 0.5],
}

#Defining the hyperparameter grid for SupportVectorRegressor
''''
param_grid = {
    'C': [0.1, 1, 50],
    'epsilon': [0.01, 1, 10],
    'kernel': ['linear', 'poly', 'rbf'],
    'degree': [2, 3, 4],
    'gamma': ['scale', 'auto', 0.1],
}
'''
grid_search = GridSearchCV(estimator=model, param_grid=param_grid,
    scoring='neg_mean_absolute_error', cv=5, n_jobs = -1, verbose=1)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
best_grid_model = grid_search.best_estimator_

#Predicting the duration
y_pred = best_grid_model.predict(X_test)
print(best_params)
```

Listing 4: Splitting the data into train and test, defining the model with its hyperparameter grid and predicting the durations.

B FME Workbench

Weather feature extraction in FME for precipitation in Skåne.

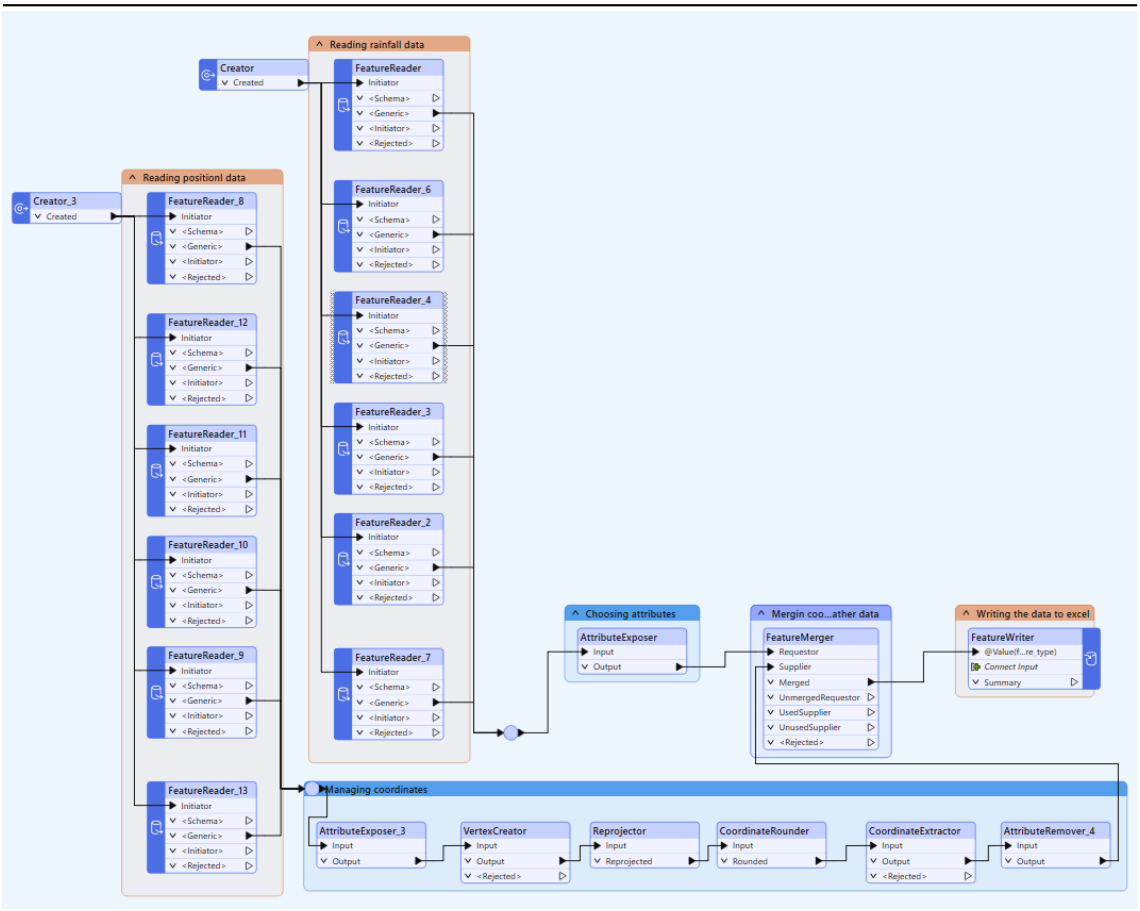


Figure B1 Reading the Excel files from SMHI for each weather station extracting the coordinates, precipitation and time of day.

Data cleaning, preprocessing and transforming in FME.

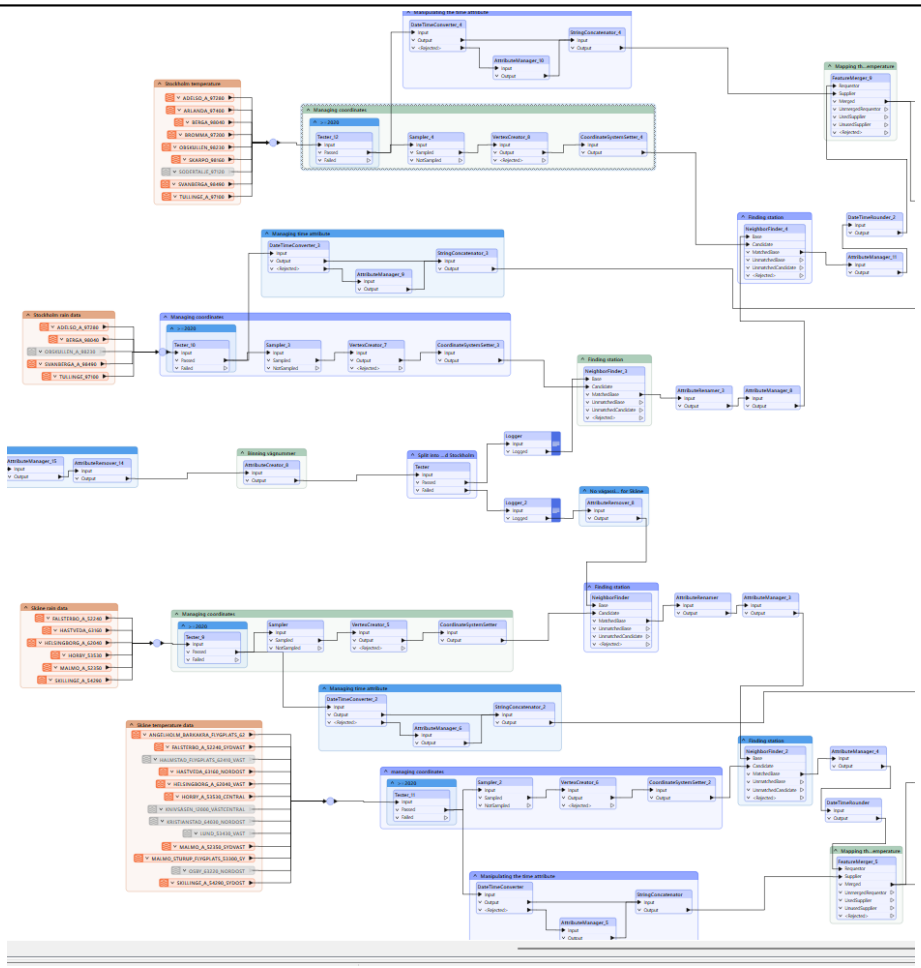


Figure B2 Small part of the FME workbench used for reading and transforming all data used in this thesis.

C Data plots

Plots representing the accident counts in for different time intervals in Stockholm and Skåne.

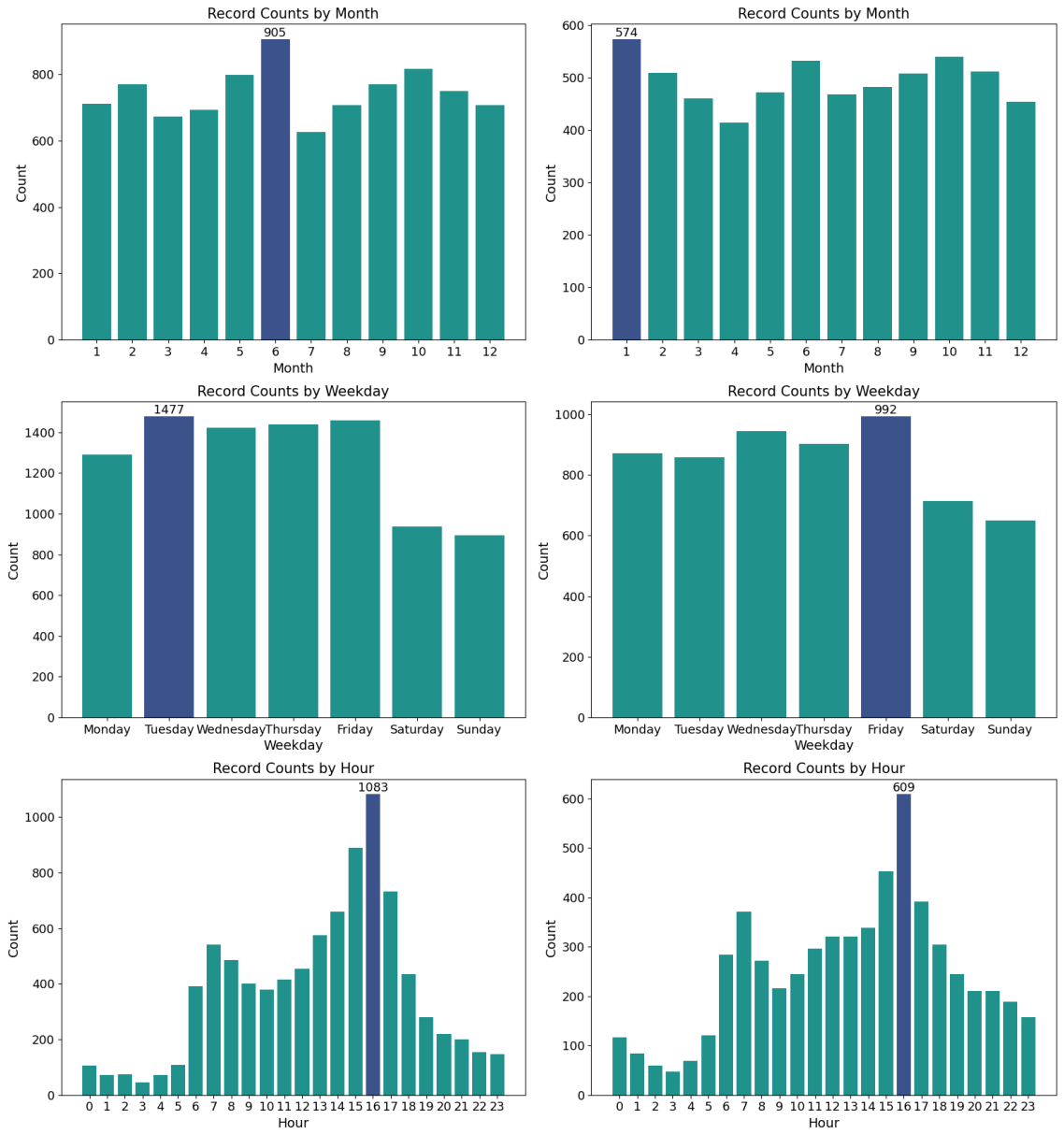


Figure C1 Plots showing how many accidents occurred in Stockholm for different time intervals.

D Data drafts

NTS data draft.

Objekt ID	Versions-lögnummer	Skapandetid	År	Månad	Dag	Veckodag	Vecka	Timme	Halvtimme	Kvart	Minut	Versionens starttid	Versionens stopptid	Arkiveringstid	Beräknad starttid	Beräknad sluttid	Fastställd sluttid	Händelsetid	Händelsetid min	Län	Vägnamn	Vägnummer	Standardavg
19546359	1	2023-05-01 00:32:14	2023	1	1	Söndag	52	00	30	30	32	2023-05-01 00:32:14	2023-05-01 00:33:41	2023-05-01 00:35:14	2023-05-01 00:31:53	2023-05-01 01:10:00	2023-05-01 00:35:14	00:01:26	1,4	Stockholms län		75	1
19546359	2	2023-05-01 00:32:14	2023	1	1	Söndag	52	00	30	30	33	2023-05-01 00:33:41	2023-05-01 00:35:14	2023-05-01 00:35:14	2023-05-01 00:31:53	2023-05-01 01:10:00	2023-05-01 00:35:14	00:03:00	3	Stockholms län		75	1
10313512	1	2023-05-01 00:34:30	2023	1	1	Söndag	52	00	30	30	34	2023-05-01 00:34:30	2023-05-01 00:36:33	2023-05-01 01:43:42	2023-05-01 00:32:51	2023-05-01 01:30:00	2023-05-01 01:43:42	00:02:03	2,1	Skåne län		22	1
10313512	2	2023-05-01 00:34:30	2023	1	1	Söndag	52	00	30	30	36	2023-05-01 00:36:33	2023-05-01 00:36:53	2023-05-01 01:43:42	2023-05-01 00:32:51	2023-05-01 01:30:00	2023-05-01 01:43:42	00:02:23	2,4	Skåne län		6	1
10313512	3	2023-05-01 00:34:30	2023	1	1	Söndag	52	00	30	30	36	2023-05-01 00:36:53	2023-05-01 00:39:37	2023-05-01 01:43:42	2023-05-01 00:32:51	2023-05-01 01:30:00	2023-05-01 01:43:42	00:05:07	5,1	Skåne län		6	1

Lägestext	Xpkt	Ypkt	Frånplats	Tillplats	Planerad	Objekttyp	Händelsetext	Tilläggstext	Påverkan	Beskrivning	Påverkad trafikriktning	Väg avstängd	Typ av tillfällig begränsning	Värde för tillfällig begränsning	Kontaktkod	Utlösningsriktning	Länk	Longitud	Latitud
Väg 75 vid Hammarbytunneln - Södra Länken i riktning mot Nacka i Stockholms län (AB)	6577066	675936	Anslutning Johannesfrovsvägen	Anslutning Johannesfrovsvägen	Oplanerad	Trafikmeddelande	Stillastående fordon	ett körfält blockerat	Liten påverkan		Nacka i Stockholms län (AB)							18,073765	59,29814
Väg 75 vid Hammarbytunneln - Södra Länken i riktning mot Nacka i Stockholms län (AB)	6577066	675936	Anslutning Johannesfrovsvägen	Anslutning Johannesfrovsvägen	Oplanerad	Trafikmeddelande	Stillastående fordon	ett körfält blockerat	Liten påverkan		Nacka i Stockholms län (AB)				4	Med		18,073765	59,29814
E22,10 från Trafikplats Sege till Trafikplats Kronetorp S båda riktningarna i Skåne län (M)	6166300	378965			Oplanerad	Olycka	Olycka		Stor påverkan	Trafikolycka med flera fordon i höjd med trafikplatz Kronetorp.	Båda				0	Med		13,077652	55,62751
Ett från Trafikplats Sunnanå (K) till Trafikplats Helenelund (17) i riktning mot Halmstad i Skåne län (M)	6165471	381149	Trafikplats Sunnanå	Trafikplats Helenelund	Oplanerad	Olycka	Olycka		Stor påverkan	Trafikolycka med flera fordon mellan Sunnanå och Helenelund.	Halmstad i Skåne län (M)				0	Med		13,112676	55,62061
Ett från Trafikplats Sunnanå (K) till Trafikplats Helenelund (17) i riktning mot Halmstad i Skåne län (M)	6165471	381149	Trafikplats Sunnanå	Trafikplats Helenelund	Oplanerad	Olycka	Olycka		Stor påverkan	Trafikolycka med flera fordon mellan Sunnanå och Helenelund.	Halmstad i Skåne län (M)				0	Med		13,112676	55,62061

Figure D1 Draft from NTS data set showing two incidents with all attributes before data handling.

E Interview 1

Interview questions Raimo Nilsson at Trafik Stockholm, 17th february 2025

- Hur ser processen ut från att ni får information om en händelse tills att den stängs?
- Vilka är de vanligaste anledningarna till att incidenter tar kortare respektive längre tid?
- Vad utgår ni främst från när ni estimerar en sluttid?
- Vilka ärenden är svårast att estimera vad gäller sluttid?
- Skiljer det sig mellan olika individer hur estimaten läggs eller har ni någon metod att utgå från?
- Vi har hört talas om att ni får notiser om att stänga ärenden efter ett visst antal minuter? Hur fungerar det?
- Vi ser klockslagen 15-16 är väldigt överrepresenterade för trafikincidenter. Påverkar tiden på dygnet varaktigheten?
- Vi har märkt att ni frekvent estimerar till exakt den fastställda sluttiden, alltså helt korrekt. Är detta enbart skicklighet eller kan ni ibland registrera den estimerade tiden precis innan ni stänger ärendet och på så vis få samma estimerade som fastställda sluttid?
- När det sker en olycka så är estimatatet relativt fastställda varaktigheten helt korrekt i ca. 10% av fallen, om ärendet däremot är ett trafikmeddelande är estimaten korrekta i ca. 15% av fallen. Hur skiljer sig registrering mellan de två typerna av incidenter?
- Vad ska ske för att incidenten ska uppdateras med en ny version i databasen?
- Varför noteras ofta "Oangiven påverkan"? Blir det automatiskt ifyllt vid frånvarande information, eller är det en indikation på att påverkansgraden är minimal?
- För attributet "Väg avstängd" finns bara "Ja" eller inget alls ifyllt. Kan ett tomt fält tolkas som "Nej" i detta fall?

F Interview 2

Interview questions Maria Jönsson at Trafik Skåne, 12th march 2025

- När trafikledaren skapar en ny incident, finns det någon färdigifylld estimerad sluttid likt Stockholms 38 minuter?
- Vad har ni för mål för era sluttidsprognoser? Är det samnma som i Stockholm att 80% av prognoserna ska vara inom +-30 minuter av fastställd sluttid?
- Vid vilka tidpunkter får ni notiser om pågående incidenter?
- Har ni någon form av vägassistans som jobbar med er? Isåfall, i vilka områden är de verksamma?
- Skiljer sig prioriteringar centralt kontra glesbygd? Tror du att ärenden går snabbare att reda upp om det är närmare en central ort?
- På vilka vägar har ni övervakning?