EXAMENSARBETE INOM DATATEKNIK,
GRUNDNIVÅ, 15 HP
STOCKHOLM, SVERIGE 2018

Predictions of train delays using machine learning

Förutsägelser av tågförseningar med hjälp av maskininlärning

ROBERT NILSSON

KIM HENNING

KTH
SKOLAN FÖR KEMI, BIOTEKNOLOGI OCH HÄLSA

Predictions of train delays using machine learning

Förutsägelser av tågförseningar med hjälp av maskininlärning

ROBERT NILSSON

KIM HENNING

Examensarbete inom Datateknik,
Grundnivå, 15 hp
Handledare på KTH: Reine Bergström
Examinator: Ibrahim Orhan
TRITA-CBH-GRU-2018:46

KTH Skolan för kemi, bioteknologi och hälsa
141 52 Huddinge, Sverige

Abstract

Train delays occur on a daily basis in the commuter rail of Stockholm. This means that the travellers might become delayed themselves for their particular destination. To find the most accurate method for predicting train delays, the machine learning methods decision tree with and without AdaBoost and neural network were compared with different settings. Neural network achieved the best result when used with 3 layers and 22 neurons in each layer. Its delay predictions had an average error of 122 seconds, compared to the actual delay. It might therefore be the best method for predicting train delays. However the study was very limited in time and more train departure data would need to be collected.

Keywords: machine learning, prediction, weather, train delay, neural network, decision tree, adaboost

Sammanfattning

Tågförseningar inträffar dagligen i Stockholms pendeltågstrafik. Det orsakar att resenärerna själva kan bli försenade till deras destinationer. För att hitta den mest träffsäkra metoden för att förutspå tågförseningar jämfördes maskininlärningsmetoderna beslutsträd, med och utan AdaBoost, och artificiella neuronnät med olika inställningar. Det artificiella neuronnätet gav det bästa resultatet när det användes med 3 lager och 22 neuroner i varje lager. Dess förseningsförutsägelse hade ett genomsnittligt fel på 122 sekunder jämfört med den verkliga förseningen. Det kan därför vara den bästa metoden för att förutspå tågförseningar. Den här studien hade dock väldigt begränsat med tid och mer information om tågavgångar hade behövts samlas in.

Nyckelord: maskininlärning, förutsägelse, väder, förseningar, neural networks, decision tree, adaboost

Contents

1 Introduction

This chapter revolves around the reason why this paper was conducted. It describes the problem, limitations and goals of this paper.

1.1 Problem

There are about 10 million inhabitants in Sweden and in the year 2017, there were 965 000 trains on the Swedish railroads. This is 13% more than in 2013. The total distance driven by the trains increased by 8% between 2013 and 2016 [1]. This shows that there has been an increase in the scale of the public transport during the last years, in parallel with the increasing population that has increased with 3.6% during that time [2].

In Stockholm, the capital of Sweden, there are about two million registered inhabitants [3]. During 2017 more than 800 000 people traveled by the means of Storstockholms Lokaltrafik (SL) and the number of commuter train boardings was around 320 000 per day. [4]

However, during the past decade, train delays have become more common [5] and a large number of departures could increase the risk of train related problems. These problems could lead to a chain reaction of delays. According to numbers from Trafikverket, the speed trains punctuality throughout Sweden is the worst in Europe [6].

Signaling failure is one of the hindrances that causes train delays [7] and the following delay time has risen with 37% in the years of 2012-2017 according to SVT with their sources deriving from Trafikverket [8]. Trafikverket claims that an increased number of lightning strikes is the main cause of these signaling failures whilst Swedish Meteorological and Hydrological Institute, SMHI shows that there were more lightning strikes during the years prior to 2012 and the trains had a better punctuation then [8].

The paper by S. Choi, Y. J. Kim et al. mentioned in section 2.6 "Related Work", investigates the accuracy of three machine learning methods when predicting whether a plane would arrive late or not. They used weather data and it would impact the majority of methods in a positive way regarding the accuracy.

Many travelers that go by the commuter train within the Stockholm area become affected when issues such as train delays occur. Valuable time is lost for the travelers and the view of the public transport could be aggravated.

1.2 Goals

To find a solution for the problem, this thesis examines the possibilities of using machine learning to predict train delays. By doing so, it offers a chance of preparation for the commuters as well as the train companies. This chance would allow the commuter to know if they should go for an earlier departure and also enable train companies to use precautions in order to prevent delays to the furthest possible extent. It would also assist the train companies by giving a possibility to find frequent delays during certain times of the week. The companies could thereafter implement further delay preventions during these particular times of the week in order to maintain a good on-time arrival rate.

The aim of this thesis is therefore to:

- Find the most accurate machine learning method among two chosen candidates to predict future train delays, while using different types of data where one of them is weather data.
- Determine whether it is possible to predict train delays using these methods.
- Create a simple prototype of a program that implements the machine learning to predict and provide statistics about train delays.

1.3 Delimitations

Train departure data was exclusively gathered about the commuter trains of Stockholm. This made it easier to use the train data with the weather data since they were both from the same area. There was a time limit of ten weeks and therefore our investigation focused on two machine learning methods of our choice. The choice was based on the literature study that was conducted in the beginning of the investigation.

The majority of data had to be collected throughout this period and therefore the data was only extracted from 2018-03-20 to 2018-05-01.

2 Theory and background

This chapter contains general and detailed information about machine learning and the specific techniques used in this thesis to achieve our goal. It includes detailed information about decision tree, neural network and boosting. Earlier works that have a relation to this thesis are also presented in the end.

2.1 Artificial Intelligence

At the dawn of artificial intelligence it was discovered that problems which could be formally described by a list of mathematical rules could be easily solved by the machines. This enabled computers to solve logical problems that were difficult for humans [9]. The first successes in artificial intelligence mostly took place in a formal environment where the program did not need to have much knowledge about the rest of the world. Chess is an example of a simple world where the programmer easily can provide a brief list of formal rules and achieve great accomplishments such as defeating the world champion of chess, Garry Kasparov in 1997 [9].

This showed that artificial intelligence excelled in problems that could be described mathematically. Although, artificial intelligence would instead struggle with less formal problems. It was problems that humans solved naturally by intuition and automatic feelings, such as recognizing spoken words or familiar things in photos. It was difficult to describe these types of problems in a formal way [9].

There are difficulties that the systems face when they are relying on knowledge that is hard-coded into their software. A great example is an artificial intelligence system named Cyc that consisted of an inference engine and a database that contained statements written by human supervisors. Cyc failed to understand a story about a man named Dave. Cyc knew that people do not have electrical parts, yet in the morning when Dave was shaving, he held onto an electric razor. The inference engine of Cyc found this to be an inconsistency and the system wondered if Dave was still a person [9].

This suggests that the artificial intelligence need an ability to acquire its own knowledge from patterns found in a set of raw data and this capability is known as machine learning [9].

2.2 Machine learning in general

Machine learning is a data analytics technique where the machine learns from earlier experiences. Computational methods are used to train the machine to learn information directly from the offered data without the use of a predetermined equation as a model. The algorithms used by the machines will adaptively improve their performance as the number of samples of training data increases [12]. Two common types of machine learning are decision trees and neural networks.

2.2.1 Supervised and unsupervised learning

Machine learning can be either supervised or unsupervised. If supervised, the model is trained whilst having knowledge of the input data x_i and the associated output y_i, where $i = 1, ..., n$ and n is the number of inputs or outputs. After the training is done, there is a chance to accurately predict the output from future inputs, or to simply gain a better understanding of the relationship between them [12][16].

Unsupervised learning, on the other hand, has no knowledge of the desired output. It works by clustering the input dataset; by finding hidden patterns or by creating interferences between the data, forming them into isolated groups. In order to perform clustering, methods like k-means, hierarchical clustering, hidden markov models and more can be implemented [12].

Supervised learning can be used in classification and regression problems using methods such as linear regression, support vector machine, decision tree, naive bayes, and neural networks [12][16].

2.2.2 Regression and Classification

A variable can be characterized as either quantitative or qualitative, where the latter is also known as categorical. Regression problems has quantitative variable responses and they take on numerical values such as a person's age, income, a value of an item or the price of a stock. These are only a few examples of what a quantitative variable can hold. Classification problems have qualitative variable responses and they take on a value in one of the predetermined categories that are also known as classes, for example the brand of a purchased product (brand A, B or C), a person's gender (male or female), a fruit (apple or pineapple) [16].

2.3 Decision tree

There are many types of machine learning methods. One of them is decision tree. A decision tree is similar looking to a tree diagram. A tree diagram begins with a single node and from that node, branches will reach out to new nodes that represent mutually exclusive decisions or events, meaning that they cannot occur simultaneously and is not influenced by the other [26][27]. The diagram starts by the first node and thereafter decisions and events will bring it onto the next node [26]. So in other words, it is a sequence of events and this is useful in probability since it can record all the possible outcomes by adding more branches. If it is used to calculate probabilities, the probability is then put onto the separate branches and the outcome is the next-coming node [28].

A decision tree is a set of questions connected in a tree, so that answering one question leads to another and eventually to a final answer. It can be used for either regression or classification problems. When it is used to solve regression problems it is called regression tree.

A decision tree (see figure 2.1) is a directed tree and has one root node without any incoming edges [21]. Other nodes have exactly one incoming edge and they are grown by a conditional split that divides the result into two new branches where the tree can continue to grow. There can be consecutive questions with different conditions by each new branch, creating additional branches that form the tree in a downward matter, or else the branch will end there. The ends of a branch are called a leaf or a terminal node and it displays the observation that falls into that specific branch [22].

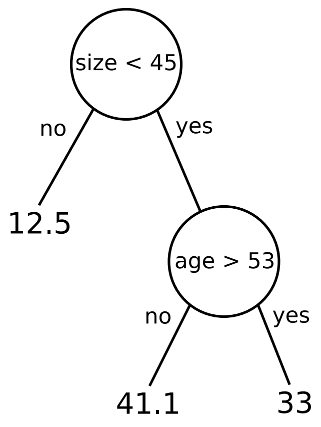


Figure 2.1: Example of a regression tree

The splits are done with a greedy top-down approach; the best split is done in each step even if another split could have created a better tree in a future step. The predictor space (the set of possible input values) is split into the regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ where a predictor X_j and cutpoint s are selected such that the sum of squared errors is reduced the most. This sum for the whole tree can be expressed as: [16]

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{2.1}$$

where R_j is the jth region or branch created by the splits, $y_i \in R_j$ is the ith value in the jth region and $\hat{y}_{R_j}$ is the mean value in the jth region.

Decision trees are similar to markov models since they are both used to form trees. yet the conditioning of markov models include events that may occur repeatedly over time. Since it is repetitive, state diagrams can be formed. The markov models accounts for time and cycles, whereas decision tree is not having that kind of focus [11].

Regression trees are trained by taking pairs of inputs x and expected outputs y where $x \in \mathbb{R}^n$, n is the number of input parameters and $y \in \mathbb{R}$. Each node is constructed in a way that creates the best partitioning of the input space x with the minimum squared difference [22].

If the tree is too deep and too wide, the model will become more complex and pose a risk for overfitting. If that would be the case then the model would have trouble finding a solution for a variety of problems. But if the tree is the opposite, it will be easier to use various data on the model with a promising result. Hence we are looking for a model that is not too complex by using proper tree pruning, which means to stop creating the conditional splits in the right time [16].

2.4 Neural network

Another type of machine learning method is artificial neural networks. The way an artificial neural network processes information is inspired by how a biological nervous system like the brain works through information [13].

2.4.1 Types of neural networks

Two common types of neural networks (NN) are feedforward NN and recurrent NN.

In a feedforward NN, information never flows backward. This is the type of NN that is inspired by biological NN. Feedforward NN do not have any feedback loops [33].

In a recurrent NN there is a loop that connects the hidden layers to the next input. Thus the state of the network is not only depending on the current input but also on the previous states. This way the output of the network can depend on previous inputs [14].

2.4.2 Feedforward neural network

A feedforward neural network (see figure 2.2) is composed of a large number of processors called neurons that are interconnected and work in unison to solve problems. The network learns by example and has to be configured for the particular application at hand, for instance voice recognition. The neural network is configured through a learning process to adapt the network to the specific application. It learns through the adjustments of the connections between neurons [13]. The connections are modified in a way that minimizes the sum of the squared errors of the outputs [20].

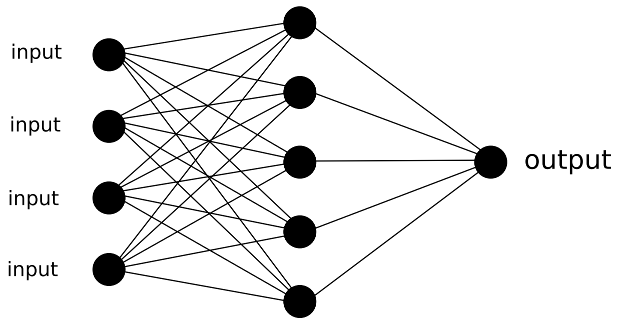


Figure 2.2: Example of a simple neural network

Each neuron has many inputs and one output. If a neuron receives a certain input pattern, it fires and activates the succeeding neurons. There are two kinds of neurons: input neurons and weighted neurons. The former is activated through inputs and the latter is activated through the weighted connections from the previously activated neurons [13]. The output of a weighted neuron is calculated using the sum of the outputs of the previous neurons, after those outputs have been multiplied with the weights of their connections. This sum is then processed by a sigmoidal function and the output of that function becomes the neuron's output. The calculation of the output can be expressed as in the equation 2.2: [20]

$$o_{pi} = f_i(\sum_j w_{ij} o_{pj}) \tag{2.2}$$

where f_i is the function that activates the neuron. The function can be for example [32]:

identity: $f(x) = x$

logistic: $f(x) = \frac{1}{(1+\exp(-x))}$

tanh: $f(x) = \tanh(x)$

relu: $f(x) = \max(0, x)$

The weights are a part of the neural networks learning process and determines the behaviour of the network. The neural network works in layers, which helps the system solve more complex problems [15].

A neural network is trained by calculating the sum of squared errors and then propagating the sum back through the layers in the network and adjusting the weights in order to minimize this sum. The sum is calculated for each pattern p:

$$E = \sum_{p} \frac{1}{2} \sum_{i=1}^{k} (t_{pi} - o_{pi})^2 \tag{2.3}$$

where t is the target and o is the output of the network. [20]

2.5 Boosting

Boosting is a general approach that is used to improve the predictions in many statistical learning methods for regression or classification. It uses multiple predictors that are trained on different subsets of the training set.

The training is done sequentially. In the first iteration all training patterns have equal probability. The patterns which are most in error then have a higher probability to be picked for the second iteration. In the second and following iterations, each tree that is grown uses information from previously grown trees. In this way, different predictors become better at predicting different patterns and the best predictors are weighted more heavily to give them more influence. The boosting approach is a slow learner [16][17].

2.5.1 AdaBoost

AdaBoost takes in a training set and calls the given weak or base learning algorithm repeatedly. The number of rounds is determined by the user. The main idea is to keep a distribution of weights over the training set and initially they are all equal. As the rounds go on, the misclassified data's weight is increased in order to have the weak learner focus on the difficult examples of the training set. Therefore it adapts to the error rates of each weak hypotheses [18].

The AdaBoost turns weak learning algorithms into strong learning algorithms [18].

2.6 Related work

In this section, information is given about past investigations that relate to this thesis.

2.6.1 Machine learning and weather data

"Prediction of weather-induced airline delays based on machine learning algorithms" [25] is a paper written by Choi S, Jin K. Y and Briceno S. It investigates the impact of inclement weather conditions on aircrafts in terms of whether the scheduled flight would be on-time or not. This is a classification problem and a few different machine learning methods were implemented. Decision tree was one of them, along with AdaBoost, K-nearest neighbor and random forest. Domestic flight data and weather data from 2005-2015 was used with and without various sampling techniques to avoid imbalanced training data.

The result of the conducted tests got worse when sampling techniques were applied and the best accuracy was achieved without the use of sampling techniques for all four candidates. The result was very even among the candidates and the random forest slightly outperformed the others along with AdaBoost. Although they were also the slowest ones. The fastest one was decision tree which also came in third place. When the sampling method was used, the distance between random forest and AdaBoost had increased in terms of percentage, closely followed by decision tree. K-nearest neighbor had an accuracy that was rather close to random. However, this does not imply that the use of sampling methods is a bad choice.

They also compared the impact of weather data: The K-nearest neighbor method did not show any difference with or without it and random forest got improved with weather data. AdaBoost and decision tree was equal to random chance without weather data but got improved while using it.

2.6.2 Disease prediction

"Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database." [19] As of this study's conduction there had not been any demonstration that deep neural networks outperform that time's state-of-the-art machine learning algorithms in disease prediction tasks.

A large population-based Electronic Medical Claims database that contained 800,000 patients was used to predict the occurrence of a diversity of diseases.

Prediction of future diseases can help the client to become proactive and interventional measurements can be conducted. Deep neural networks was compared with decision tree with a gradient booster applied to it. It was also compared to logistic regression and support vector machine approaches to predict a 5-year stroke occurrence. The programming language python was used along with the library sklearn.

The result was that deep neural networks had the best prediction accuracy, tightly followed by gradient boosting decision tree. They were better than logistic regression and support vector machine approaches. The deep neural networks obtained optimal results when using minimal amount of data, compared to the gradient boosting decision tree.

3 Methodology

This chapter describes how the thesis was practically conducted. It begins with explaining the structure and main process.

There were a few methods that were used to reach the conclusion. One of them was a literature study that was done to choose the machine learning methods. Data was collected from two different sources; it was combined and used in the performed tests to find the most accurate method. Then the obtained test-results were analyzed and compared. A prototype was also made in order to apply the obtained information into an environment. The chapter ends with information about the tools that were used.

3.1 The structure and process

This thesis is about finding the best method for predicting train delays. To reach that goal we divided the project into four phases: the preliminary investigation, the main process, the analytic phase, and the prototype.

The four phases are described in the subsections below:

3.1.1 Preliminary investigation

A literature study about the various machine learning methods and how to make predictions was conducted.

3.1.2 Main process

1. Data about past departures was collected from Trafikverket's API using the request in listing A.1. This was done between 2018-03-21 and 2018-05-02.
2. Data about past weather observations was downloaded from SMHI's API using the script in listing A.2.
3. Each departure was combined with the weather data from the time of the departure, and the combinations were saved in a JSON file.
4. The combined data was split into two parts: 80% was used as training data and the rest as testing data.

5. The machine was trained using the training data.

6. The machine learning algorithms were executed using the testing data. Their average accuracies were determined by comparing the predicted delays with the actual delays.

3.1.3 Analytic investigation

The result from the main investigation was analyzed and compared. Future changes and improvements were discussed.

3.1.4 Prototype

The collected data and the machine learning implementations were used to develop a website that allowed the user to see statistics about train delays.

3.2 Machine learning methods

A preliminary investigation was conducted to find two machine learning methods to compare. The chosen methods were decision tree with and without AdaBoost, as well as neural network.

They were chosen on the grounds that they usually achieve good results in studies. One of the studies is a former investigation about predictions of disease occurrences conducted by Hung C-Y, Chen W-C, Lai P-T et al. showed that the deep neural network proved to have the best accuracy compared to three other machine learning methods. It was closely followed by decision tree with gradient booster. The logistic regression and support vector machine approaches had a worse result regarding the prediction accuracy. This investigation weighs heavily upon our choice because it also contained predictions of future events.

Another study is a paper by Jun Lu [34] that compared Linear regression, Neural network, Decision tree, K-nearest neighbors, AdaBoost, Random forest, and Uniform blending. AdaBoost had the best mean performance in classification. In the regression test, random forest had the best mean performance but it had a high variance. Therefore AdaBoost was considered the preferred method because it had a high mean performance and low variance.

But in the end, it was a personal choice of the two (one with and without a booster) machine learning approaches that we found interesting and very promising for this kind of problem. That it would give us a nice comparison in the way that they are quite similar, and yet they are not.

3.3 Data collection

Our primary focus was to predict train delays, therefore various types of data needed to be collected to make the predictions possible. We chose to collect data from two sources: Trafikverket's API [24] and SMHI's API [23]. The former offered data about train departures and arrivals, including scheduled time, actual time, line number and more, all concerning the local traffic of Stockholm. The latter, SMHI, offered data about the weather, such as humidity, precipitation, wind speed and more, for many areas in Sweden. The reason for collecting weather data was that the weather sometimes affects the public transport.

Scripts were used to collect the data about historical train departures and weather observations. The gathering of departure data had to be done on a 24-hour basis since Trafikverket's API only handed out data for the current and previous day. SMHI gave both historical weather data for many years and forecasts and therefore it was not necessary to collect the information as regularly.

3.4 Data preparation

Before the training was done, the data from the two sources was combined in order to prepare it for use as machine learning input. For each departure, only the essential information was extracted about that departure, and about the weather for that point in time. The information was combined and put in a file as an array in JSON format, where each element had information about a departure and the weather for that departure.

If there was no weather data for a certain departure, that departure was skipped. This was done due to the fact that the weather data was needed in order to make the predictive calculations.

3.5 Input attributes

The data that was used as input variables was:

- which day of the week it was (monday, tuesday etc, one attribute for each),
- whether is was weekend (saturday or sunday),
- time of the day (one for morning/evening and one for day/night),
- weather attributes (temperature, snow depth, wind speed, visibility),
- and which train line the departure happened on (40, 41, 41X, 42X, 43, 43X, 44, 48).

In total there were 22 attributes.

3.6 Splitting the data for training and testing

Machine learning is done in two stages: The first stage consists of training the model and the second stage is about testing it and producing a result. Different sets of data were used for the two stages.

The prepared data was split into two different arrays, one for training and one for testing. The data was allocated such that all departures on a day of month that was divisible by 5 was used for testing, and the rest for training. In this way, the groups were spread out and approximately $\frac{1}{5}$ of the data was for testing and $\frac{4}{5}$ for training.

3.7 Training and testing

The training data was thereafter used to train the machine learning model to find connections between the inputs and train delays. In the second stage of the machine learning process, the model was tested by using the testing data to determine its accuracy.

The chosen machine learning methods were decision tree and deep learning using a neural network.

One test was run for each setting in each machine learning method. The setting that was tested with decision tree (both with and without AdaBoost with 10 estimators) was the maximum depth (1-10). Example figures of a tree with depth 4 versus a default depth is given in appendix B. Neural network was tested with different number of layers and layer sizes ($a \times b$) where $a \in \{1, 2, 3\}$ is the number of hidden layers and $b \in \{5, 10, 22\}$ is the number of nodes per layer. Randomness was removed from the tests by setting the `random_state` argument of decision tree, adaboost, and neural network to 0. This caused the result to always remain the same when using the same settings.

The number of estimators used with AdaBoost was 10. This number was not varied because of the time limit of the project and increasing it did not make any difference the few times it was tested.

The tests were executed by running the machine learning implementation once for each departure in the testing data. The error was calculated as the absolute value of the difference between the predicted delay and the actual delay. The average error was then calculated.

3.8 Tools

3.8.1 Data format

All data was downloaded in JSON format. The departures were stored in a MongoDB database and the weather data was stored in JSON files. JSON files were also used to store the combined data.

3.8.2 Database

The NoSQL database management system MongoDB was chosen because:

- Trafikverket's data contains arrays that would increase the difficulty of implementation if a relational database was used.
- There was no need for the ACID [10] properties.

3.8.3 Programming language

The programming language that was used was Python. It was chosen based on the grounds that it is a programming language with great viability, especially in terms of machine learning. The reason for this is that python has a great number of external libraries that are easily used. The drawback is that it might not be as efficient as other languages but due to the time limit, we chose the language with most ease and usability. We used the implementations from the python library sklearn for decision tree and neural network.

4 Result

This chapter presents the information that was obtained from the performed tests. It begins with individual tests performed on each machine learning method in order to find the settings that give the most accurate result. We test decision tree and also decision tree using AdaBoost. Thereafter the neural network was tested with various settings.

4.1 Prediction tests

The prediction tests comprise different tests for each machine learning method to find an approximation to the setting that achieves the most accurate result. The average error is the average time difference $\hat{y} - y$ where $\hat{y}$ is the predicted delay for a departure and y is the actual delay.

4.1.1 Decision tree

Decision tree was tested with varying maximum depth setting as shown in table 4.1. The maximum depth that gave the most accurate result was 4. The tests were run once for each setting.

Table 4.1: Average error for different tree depths when not using boosting.

Max depth	**Avg error (s)**
10	197
9	198
8	191
7	197
6	199
5	168
4	166
3	201
2	181
1	178

4.1.2 Decision tree with AdaBoost

AdaBoost with 10 estimators was added to decision tree. The result is shown in table 4.2. The maximum depth that gave the most accurate result was 5. The average errors of decision tree with and without AdaBoost are compared in figure 4.1. The tests were run once for each setting.

Table 4.2: Average error for different tree depths when using AdaBoost with 10 estimators.

Max depth	**Avg error (s)**
10	168
9	196
8	204
7	252
6	165
5	145
4	181
3	271
2	314
1	346

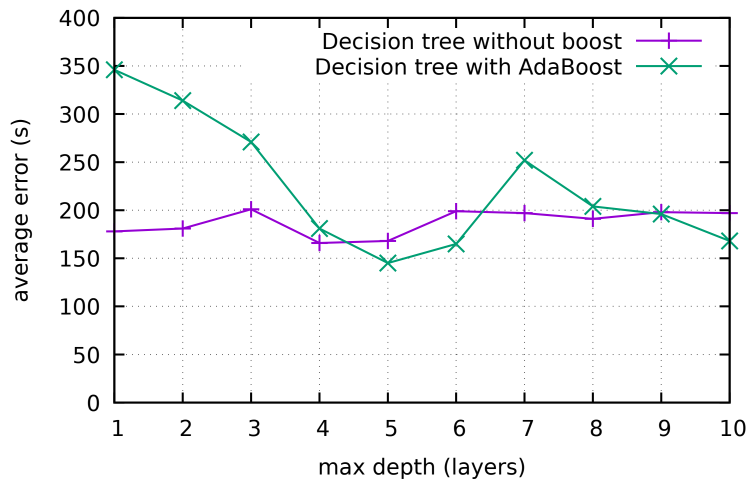


Figure 4.1: Two graphs showing the average prediction errors in seconds while using decision tree with and without AdaBoost.

4.1.3 Neural network

Neural network was tested by changing the number of hidden layers and nodes per layers. This is shown in table 4.3 and figure 4.2. The most accurate configuration was 3 hidden layers with 22 nodes per layer. The tests were run once for each setting.

Table 4.3: Average error for different layer configurations in neural network.

Hidden layers	**Nodes per layer**	**Avg error (s)**
3	22	122
3	10	129
3	5	326
2	22	227
2	10	346
2	5	322
1	22	816
1	10	1156
1	5	685

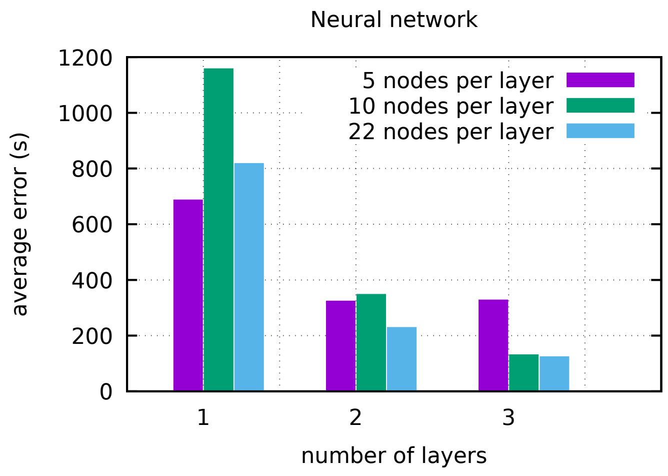


Figure 4.2: The average errors for different neural network configurations.

4.2 Prototype

A website was made that had a map of all train stations of the Stockholm commuter rail. When the user clicked on a station, information about that station appeared. This website allowed the user to see statistics about past train delays. It could not be used to predict delays. The information comprised:

- the average delay at that station since the start of the measurements,
- the average delay at that station in the last 7 days,
- lists of the most important features that determine the delay, e.g. wind speed or whether it is Friday.

Figure 4.3 shows the information flow from the APIs to the prototype.

The website was written in Python with the web framework Flask. There is a screenshot of the prototype in appendix C.

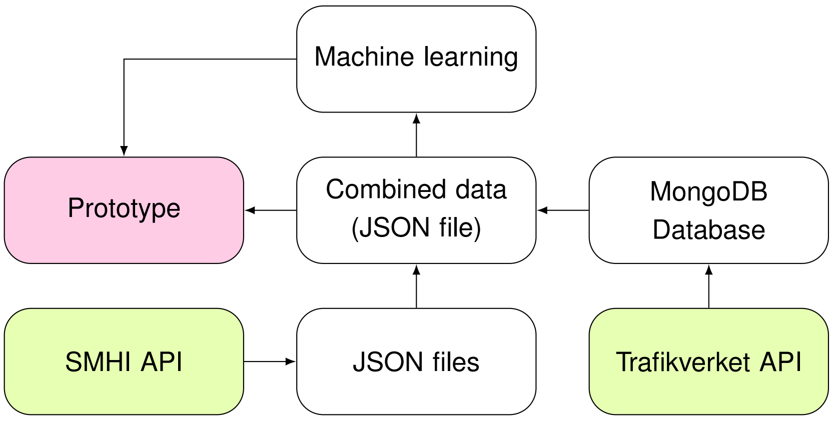


Figure 4.3: Block diagram that shows the paths taken by the data that derives from SMHI and Trafikverkets API. The train data that originates from Trafikverkets API is stored in a database whilst the SMHI weather data is stored in JSON files. The weather data is thereafter combined with the stored train data. The combined data is used to train the machine learning methods. Lastly, data about important features from the trained decision tree and other train departure data were added to the prototype.

5 Analysis and discussion

This chapter contains the analysis and discussion of the result and methodology. It includes our reasoning and thought process.

5.1 Result

The most accurate configuration of decision tree had an average error of 166 seconds which is almost 3 minutes. AdaBoost which was used with decision tree had a best average error of 145 seconds and that is an improvement since it is less than 2.5 minutes. Neural network had the best average error with 122 seconds, that is, almost exactly two minutes.

The result also shows that the decision tree did not have much variation between different settings whereas the neural network had more drastic shifts in terms of average error when predicting the train delays.

5.2 Reliability

Neural networks best result was with 3 layers and 22 nodes. We have a sense that this model might have become too complex and cause overfitting, making it difficult to work with varying data. The second best result was 2 layers and 22 nodes, which also might be too complex of a model.

In summary, the methods had an average estimation error of 2-3 minutes. This could probably be improved by optimising the settings of the machine learning methods and investigate other methods and boosters too.

5.3 Data collection and usage

It was found that a minor change in the settings of the machine learning methods, especially the neural network, could change the outcome drastically. The reason behind this could be that there was not enough data to form a consistency to the trained model, causing it to get influenced by noise.

Since it was not possible to retrieve past train data, we could only collect data for the timespan of this project. The gathered data was therefore lacking of useful information for the rest of the year. Despite this, we figured that it was possible to get an idea about which one of the algorithms that was the best suited one for this particular problem due to them both sharing equal pre-conditions.

The collected weather data was extracted in the midst of Stockholm, where most of the trains travel below ground level and that makes the data less relevant for these stations. In addition to this, some of the trains travel quite far in the outskirts of Stockholm and therefore the acquired weather forecasts are not as accurate in those specific cases. But considering the limitations of time, this method gives a good average measurements of the overall weather in those locations.

If a departure was cancelled, it was skipped and not used in the training of the machine. This was done because the machine had to work with numbers representing the delays. Skipping these departures could have had an effect on the result. One alternative solution to this could be to use a large value representing the delay of the cancelled departures, for example 15 minutes.

5.4 Alternative methodology

There are a few parts of the methodology that could have been done differently. One of them is to implement the machine learning methods manually instead of using a library. This would allow full control over the code but there is a chance the library methods have a better efficiency and accuracy.

Another part that could have been done differently is to create fake data instead of gathering real data. In this way, there would be more control of the data over its variations and content, although this would consume more time if written by hand, than the automatic collection of the real data. But if we were to use a code that would auto-generate the content of the various data, this might have been the fastest. This would have been useful due to the time limit. But there might be an issue with fictional data.

The problem lies in the arrangements of the prediction. If the data is fake, it might not be as correct as the real data in terms of connections between the delays and the rest of the data. Therefore the training could be flawed and the result incorrect.

Also, there might be a problem in arranging the prediction if all the data is fake. And the data might not be as correct as the real data in terms of connections between the delays and the other data.

5.5 Consequences in the long run

The sustainable development is an important aspect of today's science. By doing predictions of future train delays, commuters will be able to adapt by taking an earlier departure. If connections between delays and time of the day, day of the week or month are found where the risk of a delay is significantly increased, the train companies get a chance to take measures in order to prevent the delays as much as possible in these particular high-risk periods. This might produce a lowered cost for the companies leaving a financial margin for further improvements and more content passengers. That could induce people who travel by their private vehicle to gain trust in the public transports reliability.

If these people would begin to travel by the public transport more often instead of using their own private means of transport, there will be a positive effect on the environment. Thus, predicting delays might have both social and environmental implications that will be beneficial in the long run. Hence, a more sustainable development is attained.

6 Conclusions

Neural network and decision tree with and without AdaBoost were compared to find the most accurate machine learning method for predicting train delays in the commuter rail of Stockholm. They were compared by testing different settings to find the most accurate settings for each method.

Neural network was the most accurate method when using 3 hidden layers and 22 nodes per layer. The average error was 122 seconds. It was slightly more accurate than decision tree with AdaBoost, that got 145 seconds with max depth of 5. The least accurate method was decision tree with average error of 166 seconds at max depth 4.

One of the goals in this thesis was to find the most accurate machine learning method. The obtained result suggest that a deep neural network can be the most accurate method for predicting train delays. However, this study was very limited and the result has to be taken with a grain of salt since the data that was used was limited to the certain time period this thesis was conducted.

Another goal was to determine whether it is possible to predict train delays using the machine learning methods. Whether it is possible is not very clear from the result, mainly because of the short time of data collection and the limitations of the tests.

The last goal was to make a simple prototype. A website was made with a clickable train map that used the train departure data to display statistics about train delays.

6.1 Future work

Include the weather from several days before each day. That would make it possible to find sudden changes of the weather which may have an impact on the train delays. Perhaps determine if there is a difference between constant cold and sudden cold.

The accuracy of the results can be improved by running each test multiple times with a varying `random_state` argument.

It could also be investigated how much varying the number of estimators would influence the result.

In a future study it would be important to collect data for a longer duration, probably at least one or a few years. Doing that would give the machine a greater chance to find the determinant variables and filter out the noise. It would also enable the machine to learn about more types of weather.

This study was limited to two machine learning methods and one booster. More methods and boosters should be investigated.

References

[1] Trafikanalys, "Punktlighet på järnväg 2017", https://www.trafa.se/globalassets/statistik/bantrafik/punktlighet-pa-jarnvag/2017/statistikblad-punktlighet-pa-jarnvag-2017.pdf? Published: 2018-03-23 Retrieved: 2018-04-30

[2] Statistiska Centralbyrån, "Befolkningsstatistik i sammandrag 19602017". http://www.scb.se/hitta-statistik/statistik-efter-amne/befolkning/befolkningens-sammansattning/befolkningsstatistik/pong/tabell-och-diagram/helarsstatistik--riket/befolkningsstatistik-i-sammandrag/ Retrieved: 2018-05-23

[3] Statistiska Centralbyrån, "Folkmängd i riket, län och kommuner 31 mars 2018 och befolkningsförändringar 1 januari31 mars 2018", https://www.scb.se/hitta-statistik/statistik-efter-amne/befolkning/befolkningens-sammansattning/befolkningsstatistik/pong/tabell-och-diagram/kvartals--och-halvarsstatistik--kommun-lan-och-riket/kvartal-1-2018/ Retrieved: 2018-05-23

[4] SL, "Storstockholms Lokaltrafik, SL", http://www.sll.se/verksamhet/kollektivtrafik/sl/ Retrieved: 2018-04-30

[5] Dagens Nyheter, "Kraftig ökning av pendeltågsförsenningar", https://www.dn.se/sthlm/kraftig-okning-av-pendeltagsforseningar/ Published: 2016-01-22 Retrieved: 2018-04-30

[6] The Local, "Swedens high speed trains are the least punctual in Europe", https://www.thelocal.se/20160824/swedens-high-speed-trains-are-the-least-punctual-in-europe Published 2016-08-24 Retrieved 2018-04-30

[7] Nyteknik, "Trafikverket: Här är orsakerna till sena tåg", https://www.nyteknik.se/fordon/trafikverket-har-ar-orsakerna-till-sena-tag-6894428 Published: 2018-01-22 Retrieved: 2018-05-01

[8] Svt, "Fler tågförseningar på grund av signalfel" https://www.svt.se/nyheter/nyhetstecken/fler-tagforseningar-pa-grund-av-signalfel Published: 2017-09-25 Retrieved: 2018-04-30

[9] Goodfellow I, Bengio J, Courville A. "Deep Learning", http://www.deeplearningbook.org Published: 2016 by MIT Press

[10] Haerder T, Reuter A. "Principle of transaction-oriented database recovery". CSUR. 1983 Dec;15(4):287-317

[11] Glick H A. Ph.D. "Introduction to Markov Models", http://www.uphs.upenn.edu/dgimhsr/acadcrs/korea07/08.markovmodels.pdf Retrieved: 2018-05-11 Uploaded: July 2007

[12] Mathworks, "What is machine learning?",https://se.mathworks.com/ discovery/machine-learning.html Retrieved: 2018-04-01

[13] Stergiou C, Siganos D. "Neural networks", https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html Retrieved: 2018-04-19

[14] Yang T H, Tseng T H, Chen C-P. "Recurrent neural network-based language models with variation in net topology, language, and granularity". IALP. 2017.

[15] Schmidhuber J. "Deep learning in neural networks: An overview", https://www.sciencedirect.com/science/article/pii/S0893608014002135 Published: January 2015 Retrieved: 2018-04-19

[16] James G, Witten D, Tibshirani T H R. "An introduction to Statistical Learning with Applications in R" .New York. Springer; 2013

[17] Drucker H. "Improving Regressors using Boosting Techniques", https://pdfs.semanticscholar.org/6d82/26a52ebc70c8d97ccae10a74e1b0a3908ec1.pdf Published: July 08 1997

[18] Freund Y, Schapire R E. "A Short Introduction to Boosting", http://www.site.uottawa.ca/~stan/csi5387/boost-tut-ppr.pdf Published: September 1999

[19] Hung C-Y, Chen W-C, Lai P-T, Lin C-H, Lee C-C. "Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database". EMBC. 2017: 3110-3113

[20] Stites R L, Ward B, Walters R V. "Defect prediction with neural networks". ANNA ’91 Proceedings of the conference on Analysis of neural network applications. 1991: 199-206

[21] Kumar C, Dudyala A K. "Bank note authentication using decision tree rules and machine learning techniques". ICACEA. 2015: 310-314

[22] Bergstra J, Pinto N, Cox D. "Machine learning for predictive auto-tuning with boosted regression trees". InPar. 2012

[23] SMHI API `https://opendata.smhi.se/apidocs/` Retrieved: 2018-03-18

[24] Trafikverkets API `https://api.trafikinfo.trafikverket.se/API` Retrieved: 2018-03-18

[25] Choi S, Kim Y J, Briceno S, Mavris D. Prediction of weather-induced airline delays based on machine learning algorithms. DASC. 2016

[26] Tree Diagram `https://www.investopedia.com/terms/t/tree_diagram.asp`

[27] Mutually Exclusive `https://www.investopedia.com/terms/m/mutuallyexclusive.asp`

[28] An introduction to tree diagrams `https://nrich.maths.org/7288`

[29] Tree Diagrams `http://www.bbc.co.uk/schools/gcsebitesize/maths/statistics/probabilityhirev1.shtml`

[30] Probability Tree diagrams `https://www.mathsisfun.com/data/probability-tree-diagrams.html`

[31] What is a probability tree `http://www.statisticshowto.com/how-to-use-a-probability-tree-for-probability-questions/`

[32] sklearn.neural_network.MLPRegressor `http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html` Retrieved: 2018-05-13

[33] Zhang C, Xu W. "Neural networks: Efficient implementations and applications". ASIC. 2018.

[34] Lu J. "Machine learning modeling for time series problem: Predicting flight ticket prices". 2018. arXiv:1705.07205v2

Appendices

A API requests

```
<REQUEST>
  <LOGIN authenticationkey="(key)" />
  <QUERY objecttype="TrainAnnouncement">
    <FILTER>
      <EQ name="InformationOwner" value="SL" />
      <EQ name="TypeOfTraffic" value="Pendeltåg" />
      <LT name="AdvertisedTimeAtLocation" value="(date)" />
    </FILTER>
  </QUERY>
</REQUEST>
```

Listing A.1: The XML code that was used as the request to Trafikverket's API to collect train departure data for yesterday. (key) should be replaced by a key and (date) by the current date in the format YYYY-MM-DD.

```
#!/bin/bash

dl() {
    name=$1
    parameter=$2
    station=$3
    curl https://opendata-download-metobs.smhi.se/api/version/1.0/\
parameter/$parameter/station/$station/period/latest-months/data.json \
    > smhi-$name.json
}

dl temp 1 98230
dl snow 8 98210
dl wind 4 98210
dl vis 12 98210
```

Listing A.2: The script that collected data from SMHI's API.

B Decision tree images

This appendix presents selected decision trees that were tested and printed.

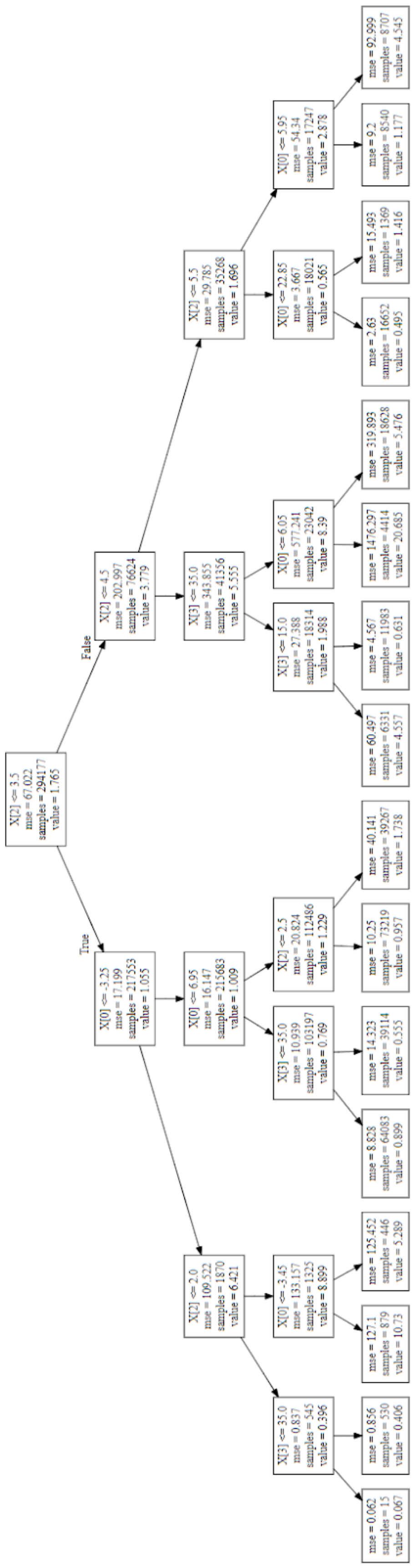


Figure B.1: Decision tree with the depth of four.

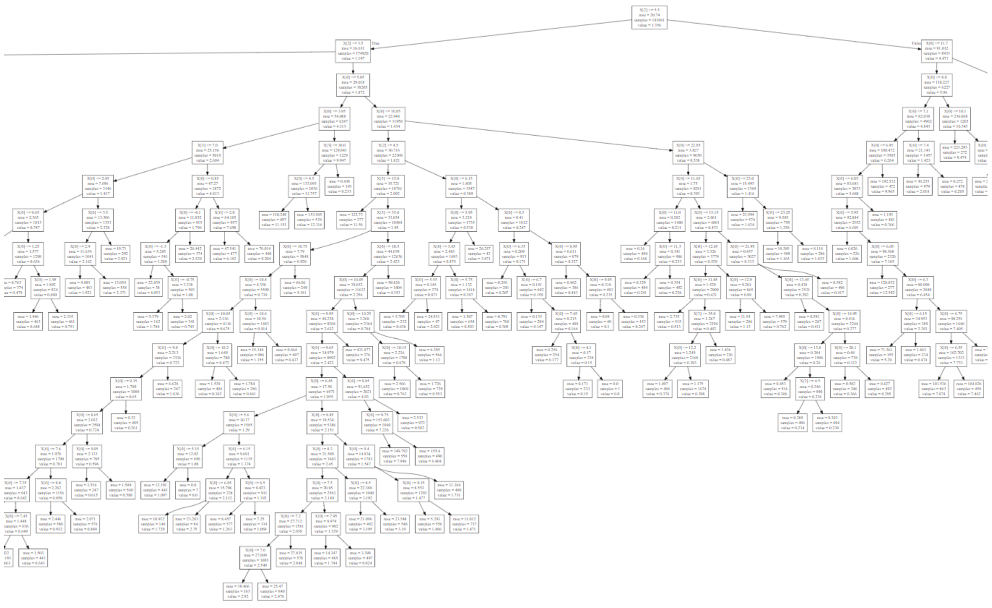
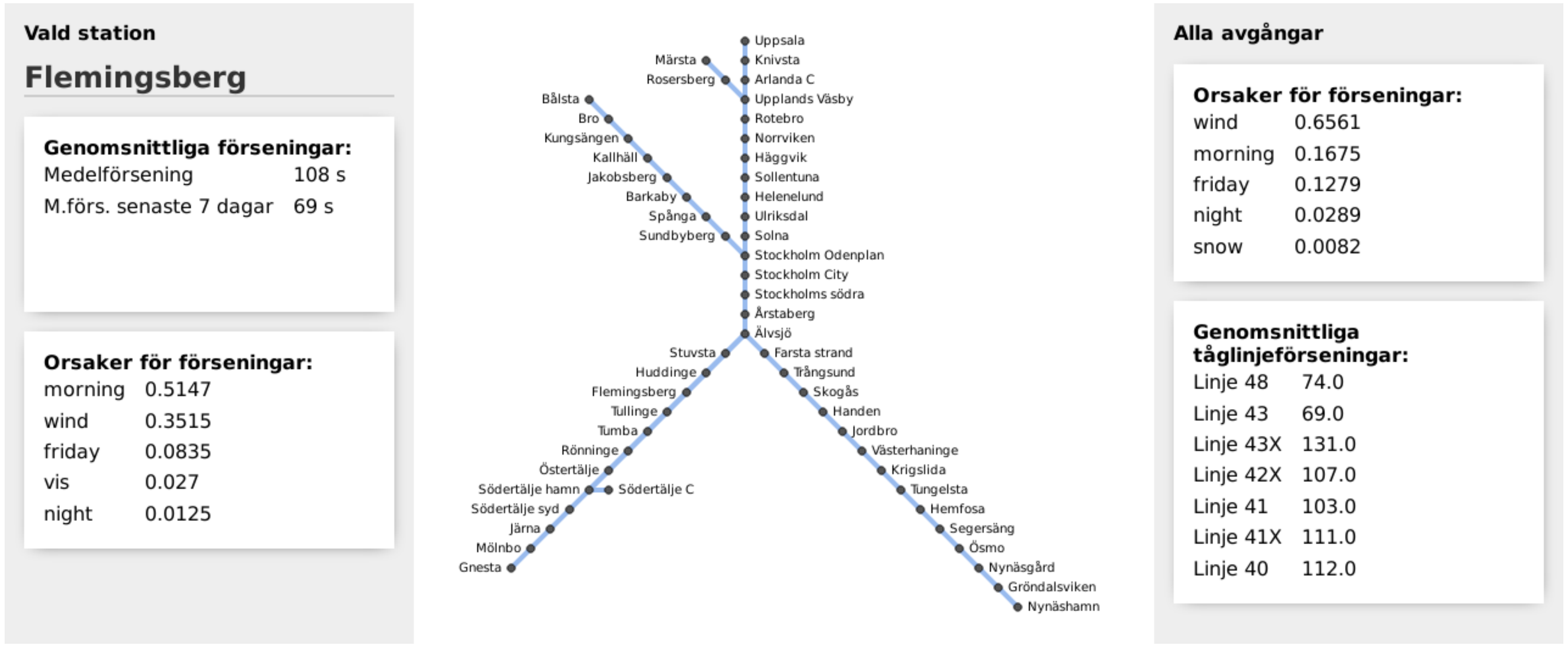


Figure B.2: A small part of the decision tree with a default depth. It got very large and complex.

C Prototype screenshot

Figure C.1 is a screenshot of the prototype that was developed.



Vald station

Flemingsberg

Genomsnittliga förseningar:

Medelförsening	108 s
M.förs. senaste 7 dagar	69 s

Orsaker för förseningar:

morning	0.5147
wind	0.3515
friday	0.0835
vis	0.027
night	0.0125

Alla avgångar

Orsaker för förseningar:

wind	0.6561
morning	0.1675
friday	0.1279
night	0.0289
snow	0.0082

Genomsnittliga tåglinjeförseningar:

Linje 48	74.0
Linje 43	69.0
Linje 43X	131.0
Linje 42X	107.0
Linje 41	103.0
Linje 41X	111.0
Linje 40	112.0

Figure C.1: A screenshot of the prototype. It has a train map in the middle and statistics about delays to the right and left.

TRITA CBH-GRU-2018:46