Towards impressive titles

Tobias Axelsson

16 april 2018

Acknowledgements

I am a student blalsadf

Abstract

asdasd

Sammanfattning

asdasdasd

Contents

Chapter 1

Introduction

The chapter starts with a background describing why road condition monitoring is important and who Trafikverket are, how road condition data is collected today and why the technology behind it needs improvement. An objective for the project is defined followed by its delimitations. Lastly, a thesis structure is presented to simplify navigation through different parts of the project.

1.1 Background

Living in cold areas of the world usually means work for invididual people, municipalities and companies in trying to maintain a non-winter-like infrastructure. This of course, also involves winter road maintenance. Salting and plowing roads is an investment in not only saving lives, but also in lowering socio-economic costs: In two scenarios on a road with 2 cm snow and a daily traffic flow of 2000 vehicles, one with a salted and ploughed road taking four hours to drive, and another scenario on the same road without winter maintenance taking five hours to drive. The total socio-economic costs are 3.5% higher in the non-maintained road, mainly due to increased travel time and thus higher accident costs [1].

Despite the socio-economic savings in performing winter road maintenance, it still represents a notable economic cost. Trafikverket, the agency in charge of road state road maintenance in Sweden, reported that winter road maintenance were roughly 18% of the total road maintenance costs in 2013 [2]. Local contracters are hired to carry out the plowing and salting of state roads, with requirements on both ends regarding when to plow, which roads to prioritize etc. Trafikverket has over 800 Road Weather Information Systems (RWIS)(Fig. 1.1) distributed across state roads in Sweden which are used by contracters to carry out winter road maintenance work [3].

Figure 1.1: RWIS Station at sensor site Myggsjön [4].

The RWIS show different information depending on what instruments they have,

Operation	worst-case cost	time complexity
Insert x into l_i	2	$O(1)$
Update $count_i$	1	$O(1)$

Machine learning as formally defined by Mitchell [5]: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E". This means that machine learning algorithms are used to solve a set of problems, measure its performance in doing so and ultimately improve in some way from previous experiences. For example, imagine a program designed to determine if a human face is in a photo or not. Since photos are taken at different distances, angles and faces have different characteristics such as eye color, skin color, distance between eyes and nose shape, implementing this "manually" may prove cumbersome. Instead of programming an algorithm to recognize faces, it can be programmed *to learn to recognize faces*. If the algorithm is allowed to analyze a dataset with thousands of photos of human faces, it could learn to distinguish a human face by recognizing parts of the face such as eyes, nose, mouth and where those parts are most likely placed to oneanother.

In essence, machine learning algorithms improve/learn in some way from analyzing a dataset. How they learn can be used to broadly categorize machine learning algorithms as either having supervised or unsupervised learning [6]. Supervised learning algorithms processes a labeled dataset while unsupervised learning attempts to make sense of unlabeled data [7]. For that reason, it makes sense to see which of the RWIS sensors can be modelled using supervised learning algorithms.

1.2 Objective

The objective is to find which supervised learning algorithms that best models MS-4 surface temperature, precipitation type and precipitation amount by building and comparing supervised learning models available in Scikit-learn. The models are built from data provided by six weather stations along state road e6 with 171425 observations over two years measuring the following every 30 minutes: MS4 surface temperature, MS4 Precipitation type, MS4 Precipitation amount, DST111 surface temperature, DSC111 friction, DSC111 surface type.

1.3 Delimitations

There are many supervised learning algorithms, all of which are not evaluated in detail in this project. An algorithm is qualified for evaluation in this project if the following is true:

1. The algorithm is a supervised learning algorithm that solves regression and/or multiclass classification problems (see 2.1.1 and 2.1.2)
2. The algorithm is available in Scikit-learn [8]
3. The algorithm belongs to one of the following algorithm families (see 2.3):
    - Decision tree based learning
    - Instance based learning
    - Bayesian learning
    - Linear regression based learning
    - Deep learning
    - Ensemble learning

At least one algorithm per family, as listed above, are evaluated in this project. The reason for this is that algorithms from same families generally perform well on similar situations. Linear regression based learning is a subset of Regression based learning as the family is named in [9]. The reason for limiting this family to linear algorithms is because every other algorithm in the other families, except for Naïve bayes, are considered nonlinear, and are thus overrepresented in the project.

1.4 Thesis structure

Chapter 2

Literature Review

The chapter gives both general and specific information on theory used for this project. Mathematical statistics, regression and machine learning are covered in the first three sections, providing a general understanding of the field of study. Specific machine learning models are explained in the final three sections of the chapter.

2.1 Supervised learning

In supervised learning, the learner (algorithm) receives a dataset of labeled observations which is used to predict correct values for unseen data[7]. A database table storing weather-related data could for example have thousands of records (observations) where data in each record belong to certain columns (features) such as wind speed w_s, wind direction w_d and time t. The goal of supervised learning is to build a mathematical model

$$y = f_{map}(x) \tag{2.1}$$

such that when new input data is used, f_{map} is able to predict a correct output value [10]. The model is built from a dataset which is typically split into three parts [11]:

- Training dataset: Used to fit the model.
- Validation dataset: Used to give an unbiased evaluation of a model built from the training dataset and potentially update its hyperparameters. Hyperparameters are model parameters that are used in some learning algorithms. They are usually fixed before the training process begins [12].
- Test dataset: Gives an unbiased evaluation of the final model.

How the dataset "should" be split is brought up in section 2.5.

Supervised learning can be thought of as having a teacher supervising the algorithm. The correct answers are in the training data and the algorithm learns from being corrected by the teacher [10]. Going back to the forementioned example of the weather

station to give a brief example of how a supervised machine learning algorithm works: Suppose a training, validation and test dataset is provided and one wishes to predict wind speed $y = w_s$ based on wind direction and time $x = [x_1, x_2] = [w_d, t]$. During the training process, a supervised learning algorithm goes through the training dataset to build a model, as seen in Eq. 2.1, and possibly updated when validated against the validation dataset. Suppose the supervised learning algorithm used is multiple linear regression (see section 2.3.4) and a model is built from the training process:

$$w_s = f_{map}([w_d, t]) = \beta_0 + \beta_1 w_d + \beta_2 t = 4 + 0.2w_d + 1.7t \tag{2.2}$$

The model can then be tested with the test dataset to see how it performs on unseen data.

Estimation of continous output variables, such as wind speed in the example presented above, is a regression problem. In supervised learning there are also algorithms associated with the problem of classification; how to categorize data [13].

2.1.1 Classification predictive modeling

In a classification problem, the computer is asked to place a new observation into one of k categories (labels), $k \geq 2$ [6]. The problem of categorizing new email as spam or not spam is an example of a classification problem. Google claims that their machine learning models can detect spam and phishing messages with 99.9% accuracy in their widely used Gmail application [11]. Classification on two labels, as in the forementioned example, is known as a binary classification case and problems with three or more labels to be classified is called multiclass classification case [14]. Classifying precipation type is a multiclass classification case since it has more than two labels.

Another example of a classification problem, one that may well be the first that machine learning novices encounter, is classification of the Iris flower dataset. The dataset consists of 50 observations with four features: length and width of the sepals and petals, in centimeters. Based on this information, the problem is to classify to which of the following labels each observation belongs to [15]:

- Setosa
- Versicolour
- Virginica

How the classification is carried out depends on the algorithm used to build the model. These kind of algorithms are commonly known as classifiers. There are several classifiers that can be used for the Iris dataset, but their performance in doing so may differ. Performance of classifiers are typically measured in accuracy, which is the amount of correct predictions divided by the number of observations in the test dataset [6].

$$\text{accuracy} = \frac{\#\text{correct predictions}}{\#\text{observations}} \tag{2.3}$$

2.1.2 Regression predictive modeling

In contrast to classification problems, such as categorizing incoming email as spam or not spam, regression problems are about predicting continous quantaties. Regression models can have either real-valued or discrete input variables. The model in eq. 2.2 is an example of a regression model since the goal is to predict a numerical value for wind speed. The problem could be translated into a classification problem by, for example stating that for given numerical intervals, the wind speed is categorized as being low, medium or high. This kind of conversion is known as discretization. But even if the conversion is useful, it can result in surprising and/or poor performance [13]. This is why both classification and regression modeling are covered in this project because one is used to classify, the other is used to predict.

Performance of regression models can be measured by computing the mean squared error (MSE) of the model on the test dataset.

$$MSE_{test} = \frac{1}{n}\sum_{i}^{n}(y'_{test} - y_{test})_i^2 \tag{2.4}$$

where y'_{test_i} are predictions on the test and y_{test_i} are actual values [6]. It's a measurement of how close each prediction was to its corresponding target value on average.

2.2 Generalization

During the training process in supervised learning, a model is typically built based on its training data, and updated in order to reduce its training error. But the fundamental goal of machine learning is to generalize beyond observations in the training dataset since it's unlikely that the same exact observations are found again on unseen data [16]. This is why test datasets are used to measure performance of regression and classification models as seen in sections 2.1.2 and 2.1.1. Both training error, how well a performs on its training data, and generalization error, how well a model performs on unseen data, need to be considered in machine learning [6].

The terminology used to explain how well machine learning models learn and generalizes to new data is overfitting and underfitting [13] which are two central challenges in machine learning [6].

- Overfitting: Random fluctuations and statisticical noise is learnt to the extent that it affects the model's ability to generalize [13]. Instead of learning the data trend in the training data, the model "memorizes" it [17]. Non-linear (see section **??**) and non-parametric (see section **??**) models are prone to overfitting as they are flexible in choosing mathematical functions to fit its training data [13].

- Underfitting: A model that performs poorly on both its training data and on generalization. If a good performance metric is used, underfitting is easy to detect [13].

The goal then, is to select a model that is somewhere between underfitting and overfitting. Underfitting is typically remedied by choosing alternative models, but the most common problem in applied machine learning is how to avoid overfitting [13]. According to Davide [17] the mere awareness of the issue of overfitting along with two powerful tools: cross-validation and regularization, can be enough to overcome the problem.

2.2.1 Cross-validation

There are several cross-validation techniques out of which the most commonly used is k-fold cross validation: Once a randomly shuffled dataset is split into a training and test dataset, the training dataset is further split into k folds. One fold is used as a validation dataset and the remainding for training. The idea is to iterate this process k times so that every fold has been used once as a validation set, and ultimately average the performance over k iterations . Using a value of $k = 10$ is a common choice in practice and in which case it is called 10-fold cross-validation [17]. Furthermore, using $k = 10$ seems to be optimal when it comes to optimizing run-time for the test, limiting bias (underfitting) and variance (overfitting) [18].

There is a variant of this technique called stratified k-fold cross-validation which is mostly used in classification problems. In stratified k-fold cross-validation the folds are created in such a way that each fold contains similar proportions of predictor labels as the full dataset. For example, think of the classification problem of classifying email as spam or not spam. If the ratio of spam/not spam is 20%/80% in the original dataset, then the same proportion is attempted to be maintained in each of the k folds. This technique tends to generate less bias and variance when compared to regular k-fold cross validation [19]. It can also be applied to regression problems [20] but the results from Breiman and Spector [21] indicate that there is little improvement from using this technique for regression problems.

2.2.2 Regularization

The second method of overcoming overfitting is regularization. This technique discourages complexity and flexibility of models by regularizing its coefficients toward zero [22]. The magnitude of the regularization can be controlled by a hyperparameter λ [23]. The higher value of λ, the higher impact regularization has on the model, but high values on λ can result in underfitting and should therefore be controlled carefully [22]. Three different types of regularization methods as covered by [23]:

- L_1 regularization (Lasso): Adds a penalty equal to the sum of the absolute values of n coefficients. This kind of regularization can nullify parameters and for that reason it can be seen as a way to limit the amount of features in the model (see 2.2.3).

$$Error_{L_1} = Error + \lambda \sum_{i=1}^{n} |\beta_i| \tag{2.5}$$

- L_2 regularization (Ridge): Adds a penalty equal to the sum of the square value of the coefficients. This exhibits a different behavior than L_1 regularization in that the coefficients are slowly reduced to zero.

$$Error_{L_2} = Error + \lambda \sum_{i=1}^{n} \beta_i^2 \tag{2.6}$$

- L_1/L_2 regularization (Elastic-net): A combination of L_1 and L_2 regularization. Here, a value α is set to determine the impact ratio of L_1 and L_2 regularization, where $0 \leq \alpha \leq 1$.

$$Error_{L_1L_2} = Error + \lambda((1-\alpha) \sum_{i=1}^{n} |\beta_i| + \alpha \sum_{i=1}^{n} \beta_i^2) \tag{2.7}$$

Regularization can be used in any parametric (see section **??**) machine learning algorithm [24].

2.2.3 Feature selection

- Filter method:
- Wrapper method:
- Embedded method:

2.3 Supervised learning algorithms

There are many algorithms that can be used to solve regression and classification problems, some of which can solve both. There is a famous theorem called the "no free lunch" theorem which contradicts an ideal case where a single machine learning is best at solving all possible machine learning problems [21]. It is therefore interesting to study the effect of different supervised learning algorithms in this project, but to explain the functionality of them in detail is out of scope. Instead, the reader is encouraged to look up details on specific algorithms elsewhere. The algorithms are grouped by similarity in this section, which is referred to as algorithm families. The algorithm families are named the same way as in [9]. Each family listed in 1.3 are covered.

2.3.1 Decision tree based learning

Decision tree algorithms uses a decision tree datastructure to solve classification and regression problems. Furthermore, Non-leaf nodes represent conditions for a specific feature and leaves are values of the target feature. One of the main advantages of decision tree algorithms is that they are easy to understand [25]. Figure 2.1 depicts an example of a decision tree used to solve a classification problem with two features:

sex and age. Starting at the root node, a decision is made once a leaf-node is reached. Ideally, the feature that best divides the dataset would be represented in the root of the tree, followed by the second best in the second level etc. However, constructing such optimal decision trees has been proved to be a NP-complete problem [26].

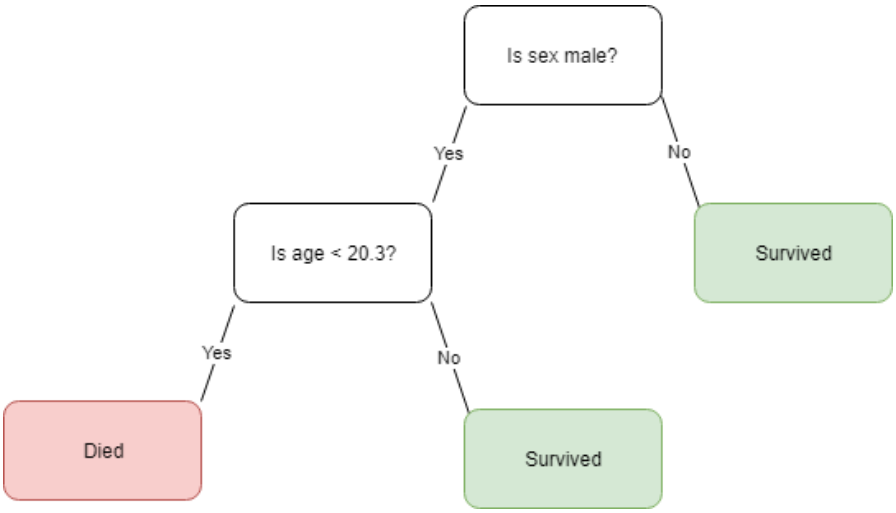

Figure 2.1: Example of a decision tree which predicts if a passenger survives a car crash or not.

Decision tree algorithms are allowed to capture non-linear patterns in the training data, which makes them flexible, but also prone to overfitting [27]. Kotiantis [25] claims there are two common solutions to battle overfitting in decision tree induction algorithms:

1. Stop the training before it fits the training data perfectly
2. Prune the decision tree

The second method has shown to be more successful in practice than the first one [28]. Pruning is a process in which subtrees of the decision tree are replaced by leaves. In other words, the size of the tree is reduced. There are different pruning methods that can be used, but exploring such details are not in the scope of this project.

Among several decision tree induction algorithms, CART and C4.5 are the most commonly used [28]. CART can solve both regression and classification problems and it is available in Scikit-learn [29].

2.3.2 Instance based learning

Instance-based learning (IBL) is about storing the provided training data, and using it to predict/classify a new observation. In other words, when a new observation is received, an IBL algorithm retrieves related observations from memory and uses it to predict/classify the new observation. This behavior means that IBL algorithms process

training data quickly while the classification/prediction process takes more time when compared to, for example, decision trees [30].

An example of an IBL algorithm is k-nearest neighbor (kNN). Generally, observations can be considered to be points in an n-dimensional space and kNN is based on the principle that these observations in a dataset are generally close to other observations with similar properties. For a new observation, the algorithm finds the k closest neighbors and uses the properties of the k neighbors to classify the observation. kNN can be used to solve both regression and multiclass classification problems in Scikit-learn [31], [32].

Wu et. al [33] argue that kNN is suitable for multiclass classification problems. Another study by L. Zhong et. al [34] showed success in predicting CT images from MRI data by using kNN regression. However, [33] mentions that the algorithm is sensitive to the choice of k: smaller values can mean it is sensitive to noise and larger values may include too many points from other classes. Overall, Okamoto and Yugami [35] showed that an optimal choice of k grew linearly with an increase in the amount of training instances. This indicates that k should be higher for higher amount of training data. However, scientific methods for choosing a specific optimal value of k are lacking [25].

Wu et. al [33] talks about the issues of choosing a distance metric for kNN. They say that among several choices, it is desirable to have a distance measure in which a smaller value between two nodes implies a stronger connection. Scaling is another issue they talk about when it comes to distance metrics. If one feature f_1 varies from, say 1.5 to 1.8, and another f_2 from 10,000 to 1,000,000 then f_2 will have higher impact on the computation of distance.

2.3.3 Bayesian learning

Bayesian algorithms are those that apply the Bayesian inference theorem. This can be used to calculate the probability of a hypothesis H after seeing the data D.

$$p(H|D) = \frac{p(H)p(D|H)}{p(D)} \tag{2.8}$$

The basic notion of this kind of algorithms is that classification can be achieved if assumptions can be made from existing data.

Naïve Bayes classifiers assume that the features in a dataset are independent from one another. Which means that each feature contribute independently to classify an observation, regardless of any correlations that might exist among the features, which is why it is called naïve. Despite this naïve assumption and its ease of use, Naïve Bayes classifiers have proved successful, even when strong dependencies among features are present [36], [37]. Naïve Bayes can be applied to regression problems through discretization, but it has limited success in comparison to classification problems when the independence assumption does not hold [38]. Naïve Bayes can be applied to multiclass classification problems [39].

The Gaussian Naiïve Bayes classifier assumes that the features have a gaussian (normal) data distribution.

2.3.4 Linear regression based learning

As stated in 1.3, only linear algorithms are algorithms are covered. A linear algorithm is one where its model is specified as a linear combination of input features, any algorithm that are not part of this algorithm are the opposite: nonlinear.

Although the names might be confusing, regression based learning is not the same as regression predictive modelling, which are referred to as "regression problems" throughout this project. Regression based analysis referes to regression analysis: a number of statistical methods for estimating relationships among variables. This is a well-known tool in statistics and it is also used in machine learning.

There are three components in a regression model: scalars β_i, independent variables X_i and dependent variables Y. In regression based learning, X_i represent input features, Y is the target feature and β_i are parameters which are tuned during the training process. The example shown in 2.2 is an example of a trained regression based learning algorithm called Ordinary Least Squares (OLS). OLS is used in regression predictive modelling and a different regression based learning algorithm called Logistic regression (LR) can be used for classification problems.

Some of the assumptions listed by [9] that most regression based learning algorithms make of the training data:

- Linear behavior: If a target feature and an input feature have a linear relationship, it would look like a straight line when plotted against oneanother. Any non-linear dependencies between the target feature and other features are assumed to not be considered input features.
- Normal distribution: Assumes the data of the target feature is normally distributed.
- Outliers: Are expected to be handled.
- Data accuracy: Values of the target feature that are consider for classification/prediction are assumed to be deleted. For example, error states that are not to be classified.

In addition to applying cross-validation, overfitting in regression based learning algorithms can be thwarted by applying the regularization techniques as seen in 2.2.2. Feng et. al [40] chose one of the regularization hyperparameters λ based on which value of λ maximised their cross-validation scores. They did so by performing a grid-search (see 2.4) on λ.

2.3.5 Deep learning

Deep learning algorithms are those that aim to learn an artificial neural network (ANN) [9]. ANNs take inspiration from biological neural networks found in the human brain. An ANN is a directed weighted graph where each node represents an artificial neuron. The graph is organized from left to right by having three type of nodes:

- Input nodes

- Hidden layer nodes
- Output nodes

The input nodes are the different input features and the output nodes are possible outcomes. Without going into too much detail, the hidden layers are there to transform the input to an output. For example, imagine a neural network model trained to recognize triangles. The input nodes represent all of the different pixels. The first hidden layer of that model distinguishes edges in the picture, the second layer recognizes when these edges form complex shapes, and so on. Figure 2.2 depicts an example of a deep learning algorithm model with the same kind of problem as seen in 2.1.

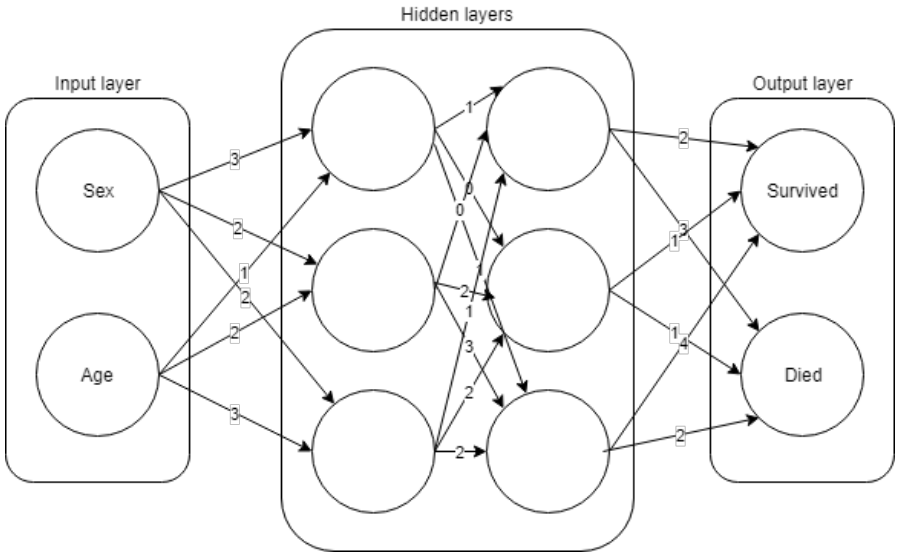

Figure 2.2: Example of a neural network with two hidden layers. The goal is to predict if a passenger survives a car crash or not.

The number of hidden layers h and the number of nodes per hidden layer N_h can be configured as hyperparameters in the different deep learning methods. So the question is, what is the optimal value of h and N_h? Heaton [41] writes that for most practical problems, having $h = 1$ is enough for most practical problems and that there is no theoretical reason to use more than two hidden layers. As for N_h, Heaton writes that a low value of N_h can lead to underfitting and the opposite may introduce overfitting and a significant increase in training time. The author encourages practitioners to try different values for N_h but provides several rule-of-thumb methods for simplicity, one of which is the following: "The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer".

All that's been covered so far of ANNs is the structure, but how are decisions made? Choices are made in the ANN by having each non-input node n_i process information from the previous node whose edge to n_i have the highest value among all nodes connected to

n_i. In other words, which of the connected nodes to n_i to choose depends on the weights of the edges that connects them. How the weights are set correlates to the learning process, which is what distinguishes the different types of deep learning algorithms.

Backpropagation is one deep learning algorithm that can be used to solve supervised learning problems. Broadly speaking, backpropagation updates the weights of the edges in the ANN based on a loss function. Whenever a "bad" classification/prediction is made (based on the loss function), the information is propagated backwards in the network from that classification/prediction and weights are adjusted accordingly.

2.3.6 Ensemble learning

Ensemble learning is based on the principle that combining the results of several classifiers/predictors into a collective score can produce better results than individual scores. Chowdhury et. al [42] conducted a study to see if ensemble methods were superior to CART, kNN, support vector machine and neural network in classifying physical activities. One of the ensemble methods used in the study was Random forest, which showed a constistent high score. Random forest performed second best in an empirical comparison of ten supervised learning algorithms [43].

Random forest averages the results from several decision tree classifiers/predictors to achieve a result. Random forest can be applied to classification and regression problems [31].

2.4 Choosing hyperparameter values

There are approaches to how hyperparameter values can be tested

2.5 Dataset split

2.6 Data preparation

2.6.1 Imbalanced data

Chapter 3

Method

The chapter covers strategies and methods used to achieve the objective of the project. Reasons for each choice of method or strategy are motivated and described in the sections, which are ordered chronologically.

3.1 Overview

3.2 Data overview

3.3 Literature study

3.4 Tools

3.5 Methodology

3.5.1 Data preparation

3.5.2 Study Multicolinearity

3.5.3 Spot check algorithms

3.5.4 Configure hyperparameters

3.6 Tools

Chapter 4

Data preparation

This chapter describes the process of preparing the data.

4.1 Data selection

4.2 Data preprocessing

4.3 Data transformation

Chapter 5

Results and analysis

Describe the process of collecting data, training and implementing machine learning algorithms with different methods.

5.1 Classifying precipitation type (Optic Eye)

5.1.1 Pre-analysis

5.1.2 Spot checking

5.1.3 Mid-analysis

5.1.4 Improving results

5.1.5 Analysis

5.2 Predicting precipitation amount (Optic Eye)

5.2.1 Pre-analysis

5.2.2 Spot checking

5.2.3 Mid-analysis

5.2.4 Improving results

5.2.5 Analysis

5.3 Predicting road surface temperature (Track Ice road sensor)

5.3.1 Pre-analysis

5.3.2 Spot checking

5.3.3 Mid-analysis

5.3.4 Improving results

5.3.5 Analysis

5.4 Predicting road surface temperature (DST111)

5.4.1 Pre-analysis

5.4.2 Spot checking

5.4.3 Mid-analysis

5.4.4 Improving results

5.4.5 Analysis

Chapter 6

Conclusions and recommendations

6.1 Conclusions

6.2 Recommendations

Chapter 7

Discussion

7.1 Thesis process

7.2 Validity and reliability

Validity and reliability of the conclusions. Needed?

7.3 Future work

Bibliography

[1] A. Arvidsson, "The Winter Model – A new way to calculate socio-economic costs depending on winter maintenance strategy", *Cold Regions Science and Technology Volume 136, April 2017, Pages 30-36*, 2017.

[2] Trafikverket, *Trafikverkets årsredovisning 2015*, 2016.

[3] ——, (2017). Vinterväghållning, [Online]. Available: https://www.trafikverket.se/resa-och-trafik/underhall-av-vag-och-jarnvag/Sa-skoter-vi-vagar/Vintervaghallning/ (visited on 02/07/2018).

[4] Pelpet. (2010). Rwis Station at sensor site Myggsjön, [Online]. Available: https://commons.wikimedia.org/wiki/File:Rwis_station_Myggsjon_01.JPG (visited on 02/06/2018).

[5] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[7] M. Mehryar, R. Afshin, and T. Ameet, *Foundations of machine learning*. Ser. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2012, ISBN: 978-0-262-01825-8. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=msn&AN=MR3057769&lang=sv&site=eds-live&scope=site.

[8] scikit-learn developers. (). Supervised Learning, [Online]. Available: http://scikit-learn.org/stable/supervised_learning.html (visited on 04/05/2018).

[9] S. Gollapudi, *Practical Machine Learning*, ser. Community experience distilled. Packt Publishing, 2016, ISBN: 9781784399689. [Online]. Available: https://books.google.se/books?id=3ywhjwEACAAJ.

[10] J. Brownlee. (2016). Supervised and Unsupervised Machine Learning Algorithms, [Online]. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (visited on 02/28/2018).

[11] F. Lardinois. (2017). Google says its machine learning tech now blocks 99.9% of Gmail spam and phishing messages, [Online]. Available: https://techcrunch.com/2017/05/31/google-says-its-machine-learning-tech-now-blocks-99-9-of-gmail-spam-and-phishing-messages/ (visited on 03/19/2018).

[12] X. Amatriain. (). What are hyperparameters in machine learning?, [Online]. Available: https://www.quora.com/What-are-hyperparameters-in-machine-learning (visited on 03/20/2018).

[13] J. Brownlee. (). Difference Between Classification and Regression in Machine Learning, [Online]. Available: https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/ (visited on 03/22/2018).

[14] M. Aly, *Survey on Multiclass Classification Methods*, 2005.

[15] R. Fisher. (). Iris Data Set, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/iris (visited on 03/19/2018).

[16] P. Domingos, "A Few Useful Things to Know About Machine Learning.", *Communications of the ACM*, vol. 55, no. 10, pp. 78 –87, 2012, ISSN: 00010782. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=82151052&lang=sv&site=eds-live&scope=site.

[17] D. Chicco, "Ten quick tips for machine learning in computational biology.", *BioData Mining*, vol. 10, pp. 1 –17, 2017, ISSN: 17560381. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=126676440&lang=sv&site=eds-live&scope=site.

[18] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Ser. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont, CA, 1984, ISBN: 0-534-98053-8. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=msn&AN=MR726392&lang=sv&site=eds-live&scope=site.

[19] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", Morgan Kaufmann, 1995, pp. 1137–1143.

[20] S. Lowe. (). Stratified Validation Splits for Regression Problems, [Online]. Available: http://scottclowe.com/2016-03-19-stratified-regression-partitions/ (visited on 03/29/2018).

[21] "SUBMODEL SELECTION AND EVALUATION IN REGRESSION - THE X-RANDOM CASE.", *INTERNATIONAL STATISTICAL REVIEW*, vol. 60, no. 3, pp. 291 –319, n.d. ISSN: 03067734. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edswsc&AN=A1992KC62300004&lang=sv&site=eds-live&scope=site.

[22] P. Gupta. (). Regularization in Machine Learning, [Online]. Available: https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a (visited on 04/03/2018).

[23] EnhanceDataScience. (). Machine Learning Explained: Regularization, [Online]. Available: https://www.r-bloggers.com/machine-learning-explained-regularization/ (visited on 03/28/2018).

[24] J. Brownlee. (). Overfitting and Underfitting With Machine Learning Algorithms, [Online]. Available: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/ (visited on 03/29/2018).

[25] S. Kotsiantis, "Supervised machine learning: A review of classification techniques.", *Informatica (Ljubljana)*, vol. 31, no. 3, pp. 249–268, 2007, ISSN: 03505596. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-36749047332&lang=sv&site=eds-live&scope=site.

[26] L. Hyafil and R. Rivest, "Constructing optimal binary decision trees is NP-complete.", *Information Processing Letters*, vol. 5, no. 1, pp. 15–17, 1976, ISSN: 00200190. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-0001815269&lang=sv&site=eds-live&scope=site.

[27] N. Lavrač, *Machine Learning: ECML 2003: 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings*, ser. Lecture Notes in Artificial Intelligence v. 14. Springer, 2003, ISBN: 9783540201212. [Online]. Available: https://books.google.se/books?id=C_E4lDNIR1QC.

[28] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997, ISBN: 0070428077, 9780070428072.

[29] scikit-learn developers. (). Decision trees, [Online]. Available: http://scikit-learn.org/stable/modules/tree.html (visited on 04/10/2018).

[30] H. Almuallim, "An efficient algorithm for optimal pruning of decision trees", *Artificial Intelligence*, vol. 83, no. 2, pp. 347 –362, 1996, ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(95)00060-7. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370295000607.

[31] scikit-learn developers. (). API Reference, [Online]. Available: http://scikit-learn.org/stable/modules/classes.html (visited on 04/10/2018).

[32] ——, (). Multiclass and multilabel algorithms, [Online]. Available: http://scikit-learn.org/stable/modules/multiclass.html (visited on 04/10/2018).

[33] X. Wu, V. Kumar, M. Steinbach, Q. Ross, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z.-H. Zhou, D. Hand, and D. Steinberg, "Top 10 algorithms in data mining.", *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008, ISSN: 02191377. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-37549018049&lang=sv&site=eds-live&scope=site.

[34] L. Zhong, L. Lin, Z. Lu, Y. Wu, Z. Lu, M. Huang, W. Yang, and Q. Feng, "Predict CT image from MRI data using KNN-regression with learned local descriptors", in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 2016, pp. 743–746. DOI: 10.1109/ISBI.2016.7493373.

[35] S. Okamoto and N. Yugami, "Effects of domain characteristics on instance-based learning algorithms", *Theoretical Computer Science*, vol. 298, no. 1, pp. 207 – 233, 2003, Selected Papers in honour of Setsuo Arikawa, ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(02)00424-3. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397502004243.

[36] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss", *Machine Learning*, vol. 29, no. 2, pp. 103–130, 1997, ISSN: 1573-0565. DOI: 10.1023/A:1007413511361. [Online]. Available: https://doi.org/10.1023/A:1007413511361.

[37] H. Zhang, "The Optimality of Naïve Bayes", in *In FLAIRS2004 conference*, 2004.

[38] E. Frank, L. Trigg, G. Holmes, and I. H. Witten, "Technical Note: Naive Bayes for Regression", *Machine Learning*, vol. 41, no. 1, pp. 5–25, 2000, ISSN: 1573-0565. DOI: 10.1023/A:1007670802811. [Online]. Available: https://doi.org/10.1023/A:1007670802811.

[39] M. Aly, "Survey on multiclass classification methods", *Neural networks*, pp. 1–9, 2005.

[40] Y. Feng, J. Fan, Y. Feng, and Y. Wu, "NETWORK EXPLORATION VIA THE ADAPTIVE LASSO AND SCAD PENALTIES.", *ANNALS OF APPLIED STATISTICS*, vol. 3, no. 2, pp. 521 –541, n.d. ISSN: 19326157. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edswsc&AN=000271979600002&lang=sv&site=eds-live&scope=site.

[41] J. Heaton, *Introduction to Neural Networks with Java*. Heaton Research, 2008, ISBN: 9781604390087. [Online]. Available: https://books.google.se/books?id=Swlcw7M4uD8C.

[42] A. K. CHOWDHURY, D. TJONDRONEGORO, V. CHANDRAN, and S. G. TROST, "Ensemble Methods for Classification of Physical Activities from Wrist Accelerometry.", *Medicine and Science in Sports and Exercise*, vol. 49, no. 9, pp. 1965 –1973, 2017, ISSN: 01959131. [Online]. Available: http://proxy.lib.ltu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=s3h&AN=124667245&lang=sv&site=eds-live&scope=site.

[43] R. Caruana and A. Niculescu-Mizil, "An Empirical Comparison of Supervised Learning Algorithms", in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 161–168, ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143865. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143865.