NTNU
Norwegian University of
Science and Technology

Data-driven Avalanche Forecasting

Using automatic weather stations to build a data-driven decision support system for avalanche forecasting

Anders Asheim Hennum

Master of Science in Physics and Mathematics
Submission date: March 2016
Supervisor: Ingelin Steinsland, MATH
Co-supervisor: Eivind Juvik, Statens Vegvesen

Norwegian University of Science and Technology
Department of Mathematical Sciences

Thesis outline

This thesis consists of three parts:

Part I is a scientific paper about using automatic weather stations to build a data driven decision support system for avalanche forecasting.

Part II gives some background information about the models being used.

Part III contains notes on the implementation.

Preface

This thesis is written as a part of my master in Applied Physics and Mathematics at the Norwegian University of Science and Technology (NTNU). Statens Vegvesen (The Norwegian Public Road Administration) proposed the project, wanting to research what data science could do to help them improve avalanche safety on the road network. As a statistician with an interest in avalanches, it seemed like a very interesting project where knowledge from many different domains had to be combined. Also, using knowledge about statistics and technology to solve a practical problem was a motivational factor to me.

The main part of the thesis is an article which is written such that it can be published as a scientific paper. It is intended for researchers in avalanche science, engineers working with avalanches or others who work with avalanche safety. Knowledge about snow avalanche formation is assumed so no background information about avalanche theory is included in the article. Also, details about the mathematical properties of the models are kept to a minimum. Part II briefly explains the theory behind the models that are being used. This part is added to give some background information and is intended for a more mathematical oriented audience. The third and last part is about the implementation. Much of the work in this project was related to data analysis and programming, and also involved development of a prototype. To do that efficiently, I took advantage of many open source projects and I also developed one myself for easy retrieval of weather data. The prototype showing live avalanche predictions for Mefjorden at Senja is available at `http://52.19.132.210:5000/`.

Acknowledgments

I would like to thank my supervisor Ingelin Steinsland for good supervision and valuable reviews. Also, I would like to thank Eivind Juvik and Tore Humstad in Statens Vegvesen for support and for taking initiative to this project. At last, I would like to thank my brother, Endre, for a valuable review of the article.

Using automatic weather stations to build a data driven decision support system for avalanche forecasting

Anders Asheim Hennum
Department of Mathematical Sciences

NTNU, March 2016

Abstract

In this paper, a decision support system for avalanche forecasting based on data from automatic weather stations is developed and tested. 17 years of avalanche and weather observations from Senja in Northern Norway are processed and analyzed to identify meteorological factors important for avalanche formation. Current snow depth and precipitation the preceding days of an avalanche release are found to be most important. Further, a simple model based purely on snow depth, a logistic regression model and a random forest model are fitted to training data and used to forecast the probability of an avalanche on test data. The results show that the logistic regression model and random forest model performs better than the simple snow depth model. Random forest is able to detect 12 out of 19 avalanches, obtaining a true skill score of 0.6. This is better than logistic regression that detects 9 out of 19, obtaining a true skill score of 0.43. The study shows that it is possible to develop a decision support system for avalanche forecasting using already existing infrastructure. However, the results also shows that the models have their limitations. Many avalanches are not detected, and hence, a system based on these models should only act as decision support system and not be relied on solely. At last, a prototype is developed and tested live. Live testing showed that reliability of the weather stations in use is important for operational usage.

1 Introduction and background

In Norway, avalanches are one of the most frequent cause for roads to be closed. A total of 6500 avalanche events were registered on Norwegian roads between 1998 - 2008 [Norem, 2014]. Securing all roads exposed to avalanche activity by building tunnels or other protective structures, is not an option due to high costs. The road network is large, and in many rural regions the traffic volume is too low to justify such costs. Closing roads or artificial triggering of avalanches are in many regions the only possible actions to secure roads during periods of high avalanche risk. For many settlements, closing the road means isolation or long detours, so it should be kept to a minimum. Thus, to maintain a high level of safety without closing the road too often, it is important to estimate the avalanche risk as precisely as possible. Today, the contractors that operate the road are responsible for monitoring the road, determine avalanche risk and take action when it is required. Estimating avalanche risk is a difficult task, and the knowledge they base their decisions on varies much between contractors. No systematic way of estimating avalanche risk is established. Having a system that can utilize available information and support decision makers in these situations would be beneficial as it can reduce the costs of estimating avalanche risk and possibly contribute to better safety and minimize consequences of closed roads.

The relation between meteorological factors and avalanche formation has been studied for a long time. Atwater [1954] listed 10 factors he found to be important in avalanche formation and several studies has been undertaken since then. Much of this research is summarized by Schweizer et al. [2003]. To do predictive analysis, various statistical learning methods have been applied to avalanche and weather data. Bois et al. [1975] used multivariate discriminant analysis and Buser [1983] used nearest neighbors to predict avalanches. Nearest neighbors has also been implemented in the avalanche forecasting program NXD-2000 [Gassner et al., 2000] and used operationally. More recently Hendrikx et al. [2014] used classification trees to forecast avalanches and Marienthal et al. [2015] used both classification trees and random forest to predict the occurrence of deep slab avalanches.

This study follow the lines of previous work and explores how logistic regression and random forest performs as predictive models for avalanche activity on Highway 862 at Mefjorden, Senja in Northern Norway. Weather observations and avalanche data for 16 years, with a total of 78 avalanche days and 2717 non-avalanche days, are analyzed in the study. First, an exploratory analysis is performed to give an overview over the data and to verify the existence of meteorological factors important in avalanche formation. This is an important part of the study, as meteorological factors important in avalanche formation will depend on the climate and topography of the studied area. Exploratory analysis will reveal some of the information the models can utilize to predict avalanches in this area. Further, two logistic regression models and one random forest model are fitted to 13 years of training data, and then used to forecast the occurrence of avalanches on 3 years of test data. These models are examined in order to interpret what information they utilize and if this is in agreement with common avalanche theory. Rather than predicting the outcome directly (i.e avalanche or no avalanhce), the models in this study are set up to model the probability of an avalanche for a given day. The intention is that having a probability make the models more suitable as a decision support system. Uncertainty is then incorporated in the prediction and additional information (snow pack information, weather forecast, etc.) can be combined with the predicted probability to make a final decision. At last, to explore how such a system can be implemented and used operationally, a prototype that automatically collects weather data, predicts probabilities, and displays the avalanche probability is developed and described. The prototype was tested live for the season 2015/2016.

2 Study area and data

2.1 Study area

Senja is an island off the coast of Northern Norway, southwest of the city of Tromsø. The climate is subpolar oceanic and low-pressure systems coming in over the coast bring large amounts of snow during the winter. (Figure 2.2 shows the total precipitation (water equivalent) and average temperature over the winter months). Topographically, Senja mostly consists of mountains and fjords and several roads are passing through areas with regular avalanche activity. The relatively cold climate and large amounts of precipitation creates together with steep slopes conditions that are favorable to avalanche activity. In this study we focus on one specific road stretch on highway 862, going from Mefjordbotn to Senjahopen (figure 2.1). There are several avalanche paths along the 8 km road stretch and it is subject of regular avalanche activity during the winter months. Closing this road means a significant detour for the inhabitants at Senjahopen and logistic challenges for the local fishing industry. Figure 2.3 shows a map over the region.

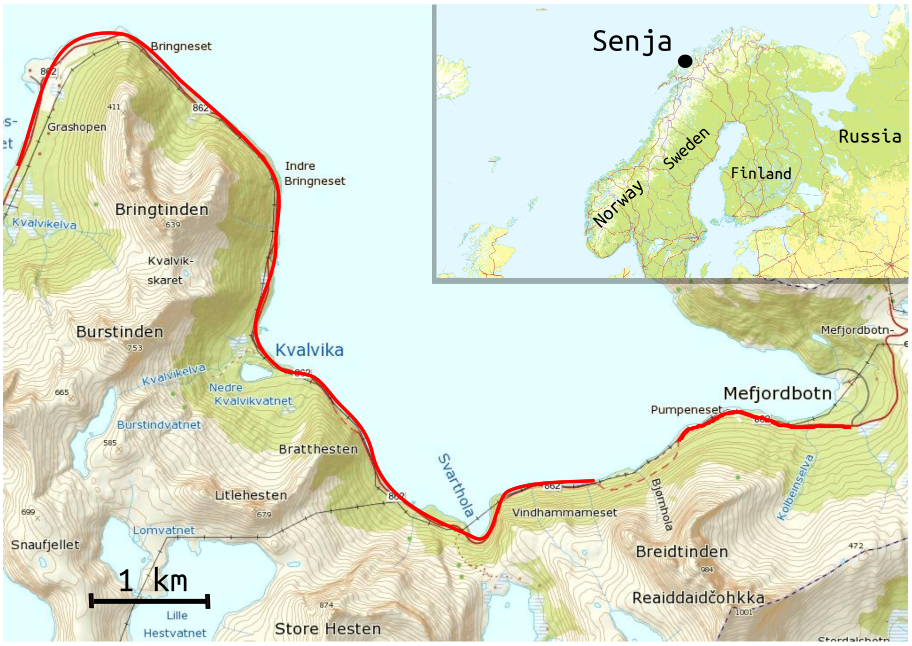

Figure 2.1: *Location of Senja and topographical map over the studied area.*

2.2 Avalanche data

From January 1995 to April 2013 a total of 315 snow avalanche events on 130 unique days were registered on highway 862 from Mefjordbotn to Senjahopen. Avalanches have been registered by snowplough drivers operating the road and only avalanches that hit the road are registered. As it can be several hours between each passing, especially during night, there is some uncertainty in the registration times of the avalanches. Only a few avalanches contain additional information like type of avalanche and size, so in this study all avalanches are treated equally. The number of avalanche events per season varies greatly from season to season (Figure 2.2). The avalanche season 2000, i.e winter 1999/2000, had over 40 avalanche events, while both the 1999 season and the 2004 season had zero registrations.

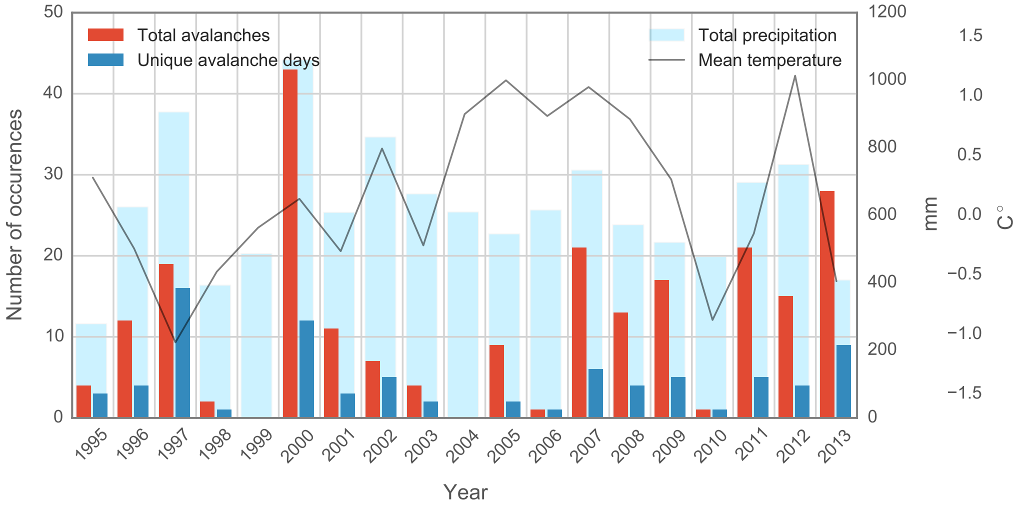

Figure 2.2: *A summary of total precipitation (mm), mean temperature (°C), number of observed avalanches and unique avalanche days over the period from November to April. Precipitation is observed at Botnhamn and temperature at Hekkingen Fyr.*

2.3 Weather data

Data from three automatic weather stations run by the Norwegian Meteorological Institute are used in the study. Hekkingen Fyr is located on a small island north of Senja and observes temperature and wind. Botnhamn is located on the north eastern part of Senja and observes precipitation and snow depth. Grunnfarnes is located on the south western part and also observes precipitation and snow depth. All stations are just above sea level. These are the three nearest stations that have been operational in the period from 1995 to 2013. Figure 2.3 shows a map over the area and the location of the stations. Two precipitation stations are included to get a better measure of precipitation as it can be large local variations due to the topography. Temperature and wind are measured hourly while precipitation is measured daily. Table 1 shows the details.

Table 1: *Details about the weather stations used in this study. The distance is from the station to the avalanche area.*

Location	Distance	m.a.s.l	Weather observed	Interval
Botnhamn	12 km	10 m	Precipitation and snow depth	Daily (7am)
Grunnfarnes	30 km	3 m	Precipitation and snow depth	Daily (7am)
Hekkingen Fyr	17 km	14 m	Temperature and wind	Hourly

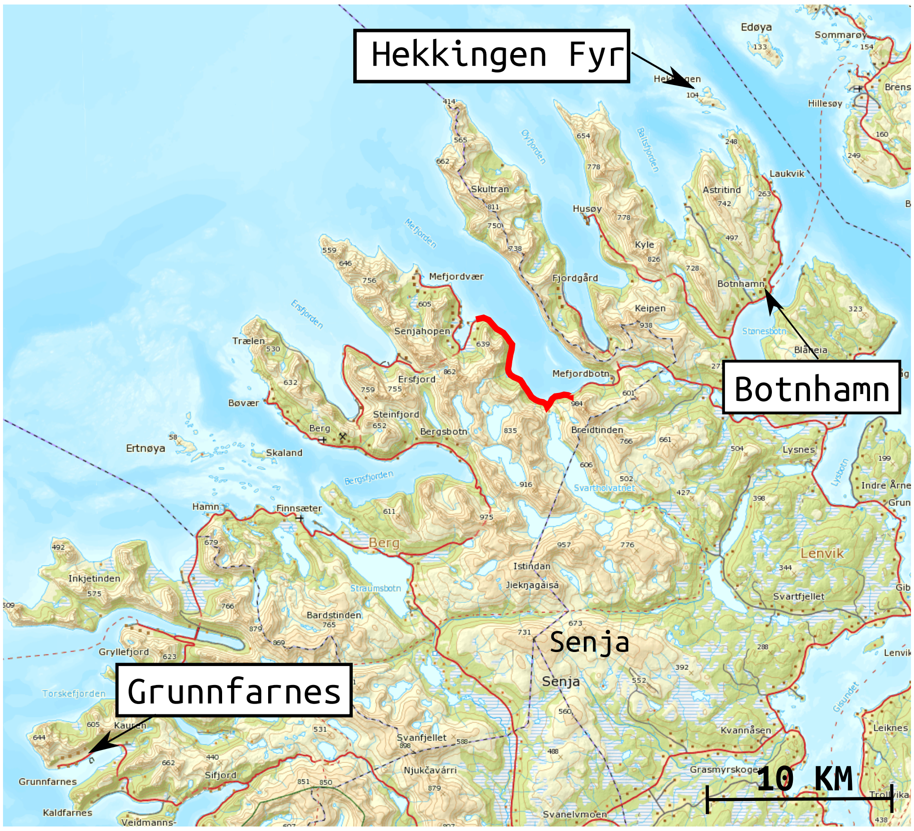

Figure 2.3: *Map over northern part of Senja showing the studied road (red) and the weather stations used in the study.*

3 Methods

3.1 Meteorological metrics

Before the data could analyzed and explored, raw weather observations had to be processed into meaningful meteorological metrics and aligned with avalanche observations. As precipitation is only measured daily, hourly wind and temperature data was processed into daily observations starting at 07:00 (NMT). Based on what previous research has found to be important in avalanche formation [Schweizer et al., 2003], the data was processed into variables that would expose relevant information, including mean temperature, maximum temperature, temperature trend, average wind speed and dominating wind direction. These quantities were aligned with daily measures of snow depth and 24 hour accumulated precipitation (water equivalent). In order to preserve relevant information from previous days, some additional variables were added to the daily observations. In particular, this was accumulated precipitation last three days, maximum mean temperature last 5 days (but not including the day itself) and change in snow depth last 24 hours. Table 2 lists the variables included in the final processed dataset.

3.2 Avalanche days

To combine avalanche observations with the processed weather data described above, a binary variable for indicating avalanche activity was added. 1 for indicating that one or more avalanche events were registered within the proceeding 24 hours of the weather observation and 0 for indicating that no avalanche activity was registered. As many days have several avalanche events, we loose some information by reducing avalanche activity to a binary variable, but having a binary variable is required when working both with the Kolmogorov-Smirnow two sample test and logistic regression.

3.3 Data cleaning and processing

The processed dataset was filtered by removing all rows with missing weather data. Also two full seasons, 1999 and 2004, were removed as they contained zero avalanche registrations. It is suspected that this is due to lack of registrations. After this, the dataset contained data from 17 avalanche seasons and a total of 2795 days were 78 were associated with avalanche activity. For exploratory analysis, the full dataset was used to utilize all data available. For model analysis, the dataset was split into a training set (avalanche seasons from 1995 to 2010) and a test set (avalanche season 2011, 2012 and 2013). The training set contained 59 avalanche days and the test set contained 19 avalanche days. Table 3 shows the details about the training set and test set. In the exploratory analysis, to compare the distribution of the meteorological variables for avalanche days and non-avalanche days, a random subset of non-avalanche days was generated. This is due to the imbalanced data. There is by far more non-avalanche days than avalanche days, and many of the non-avalanche days are from periods with little snow or other weather factors where one easily can discriminate between avalanche and non-avalanche days. Thus, to have a sample with more interesting data to analyze, a random sample of non-avalache days was generated by, for each avalanche day in the dataset, pick a random non-avalanche day from the same month and year as the avalanche day. With this method, we get a sample of non-avalanche days of same size as avalanche days, and from about the same periods of the season. This is more interesting to analyze as it is in the periods near an avalanche event a decision support tool will be most useful.

Table 2: *Variables included in the processed dataset, i.e. each day (row) in the dataset have the listed observations. The observation time is at 07:00 am.*

Name	Meteorological variable	Unit
SA_x	Snow depth at Botnhamn	cm
SA_y	Snow depth at Grunnfarnes	cm
SA_24_x	Change in snow depth last 24 hours at Botnhamn	cm
SA_24_y	Change in snow depth last 24 hours at Grunnfarnes	cm
RR_24_x	Precipitation last 24 hours at Botnhamn	mm
RR_24_y	Precipitation last 24 hours at Grunnfarnes	mm
RR_3_x	Precipitation last 3 days at Botnhamn	mm
RR_3_y	Precipitation last 3 days at Grunnfarnes	mm
TA_mean	Mean temperature last 24 hours	°C
TA_max	Maximum temperature last 24 hours	°C
TA_5	Maximum daily mean temperature last 5 days	°C
TA_grad	Change in mean temperature from 12-6 hours to last 6 hours	°C
FF	Mean wind speed last 24 hours	m/s
DD	Dominating wind direction last 24 hours	-
OA	Binary indicator for avalanche activity within proceeding 24 hours	-

Table 3: *Number of avalanche days and non-avalanche days in training set and test set.*

	Training set	Test set	Total
Avalanche days	59	19	78
Non-avalanche days	2308	409	2717
Total	2367	428	2795

3.4 Exploratory analysis

Using the processed dataset, some basic plots were examined to get an overview over the data. This included a plot of the scatter matrix along with estimated kernel densities. A Kolmogorov-Smirnov test was undertaken to compare the distribution of the variables grouped by avalanche activity. As the randomly generated sample of non-avalanche days introduce a random effect in the estimated p-values, 100 random samples of non-avalanche days were generated. The Kolmogorov-Smirnov test was then used to compare each of these to the sample of avalanche days. This resulted in 100 p-values for each of the variables in the dataset. The reported results from the test were the mean and standard deviation of these values.

3.5 Models

Two types of models were considered in this study: logistic regression [Aldrich and Nelson, 1984] and random forest [Breiman, 2001]. As the data is highly imbalanced (78:2717) in favor of non-avalanche days, both logistic regression and random forest will be biased towards the non-avalanche days as they aim to minimize the overall error rate [Chen et al., 2004]. This will lean the predictions towards correctly classifying non-avalanche days, rather than avalanche days. This is of less interest, as we are more concerned about correctly classifying avalanche days. To address this problem, avalanche days were assigned more weight than non avalanche days (weights are also referred to as misclassification cost). This way, the models will penalize misclassifying avalanche days more than non-avalanche days. We followed Hendrikx et al. [2014] and used weight 2 for avalanche days and 1 for non-avalanche days. The weights were set arbitrarily, but are supposed to reflect the actual cost of misclassifying in operational usage (i.e. misclassifying an avalanche day is twice the cost of misclassifying a non-avalanche day). Opposed to Hendrikx et al. [2014], where they predict the outcome directly (i.e 0 or 1) we predict the probability of an avalanche day and then map the probability to an outcome. In that sense the class weights are of less importance here as they will only act as a scaling parameter for the probabilities.

The first logistic regression model used only snow depth as the variable to explain avalanche activity. This model was included to see how much of the avalanche activity a very simple model could explain. This is useful when evaluating the performance of more complex models and for verifying their potential advantage over a simple model with only snow depth. The logistic regression model with only snow depth is refereed to as the snow depth model.

For the second logistic regression model, a variable selection was performed to reduce the number of variables. The dataset contains many variables and many of them are highly correlated (table 9), so it seems like a good idea to reduce the number variables and thus the potential for overfitting.

The results from a simpler model will also be easier to interpret and give information about which meteorological factors that are most important in avalanche formation at this specific location. Selecting the most important variables is a difficult task itself and several methods exist. Here, we used a stability selection method based on work by Meinshausen and Bühlmann [2010]. The idea is very general:

- Fit a logistic regression model to a random subset of training data and use the l_1 penalty (known as Lasso [Tibshirani, 1996]) to determine the regression coefficients. When using Lasso, less important variables will tend to be excluded by getting a regression coefficient equal to zero.
- Repeat the step above an appropriate number of times, each time with a different random subset of data. Important variables will tend to be selected more often than less important variables.

Using this stability selection method, a subset of variables was selected. A logistic regression model with ordinary least square penalization (l_2) was then fitted to the reduced dataset (i.e the dataset with only the selected variables), and used to predict the probability of an avalanche day on the test dataset. This model is refereed to as the logistic regression model.

For the random forest model, all variables were used. 500 classification trees were grown by splitting bootstrapped samples of data recursively into groups of avalanche and non-avalanche days. At each split, two variables were selected randomly and the Gini index [Breiman et al., 1984] was used to determine the values of these that would best split the data. This methodology is known as the CART methodology [Breiman et al., 1984]. The nodes were split until they were homogeneous, i.e contained only non-avalanche days or avalanche days, or until the node contained less than 7 samples. Setting a minimum criteria for splitting a node helps to reduce overfitting of the trees, i.e having many nodes with only a few samples. A prediction for a given weather observation is from a single tree calculated by the respective class weights times the number of avalanche days and non-avalanche day in the terminal node for the given observation. The final predicted probability from the random forest is the average over all individual trees. To get a better overview of what information the random forest found to be most important, the variable importances were plotted. The variable importances are calculated by mean decrease in impurity [Breiman et al., 1984] over all nodes in all trees.

For both logistic regression and random forest, `python` [Rossum, 1995] was used with the package `scikit-learn` [Pedregosa et al., 2011] that contains implementations of both models. For data cleaning and filtering, `pandas`

[McKinney, 2010] was used. All models were trained on the training dataset and then used to predict avalanche probabilities on the test dataset.

3.6 Model performance

To evaluate the models predictive performance on the test set (seasons 2011, 2012 and 2013), the unweighted average accuracy (RPC), true skill score (TSS), false alarm ratio (FAR), probability of detection (POD), probability of non-events (PON) and probability of non-detection (FSR, i.e false stable ratio) was used. Definitions are in table 5. These are identical to the scores used in similar studies on avalanche forecasting by Schweizer et al. [2009], Hendrikx et al. [2014] and Marienthal et al. [2015]. The scores are based on values from the confusion matrix (Table 4). This requires that the predicted probabilities have been mapped into avalanche days (1) or non-avalanche days (0). This was done by setting a fixed threshold at 15% and for days with probability above this, an avalanche was predicted. The threshold was set arbitrarily at a value that gave a reasonable ratio between true positives and false alarms.

The scores gives a good indication of the models performance, and makes it easy to compare them. But the scores alone do not tell the full story. To get a better visual interpretation of the models performance, the probabilities were plotted as a time series along with the true avalanche observations. This also shows the time dependence in the probabilities and shows how the probability of an avalanche behave before and after an observed avalanche event.

Table 4: *Confusion matrix*

		Observed	
		Avalanche	No avalanche
Predicted	Avalanche	True positive (TP)	False positive (FP)
	No avalanche	False negative (FN)	True negative (TN)

3.7 Implementation of a prototype

Based on the results from the model analysis, a prototype was developed and implemented. The prototype consisted of a web application that automatically pulled weather data from the Norwegian Meteorological Institute and calculated the probability of an avalanche. The predicted probabilities were shown on web page and plotted as a time series together with observed

Table 5: *Mathematical definitions of the scores used to validate the models. TP is true positive, FP is false positive, TN is true negative and FN is false negative.*

Score	Definition	Interval	Optimal value
RPC - Unweighted average accuracy	$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{FP+TN}\right)$	$\{0,1\}$	1
TSS - True skill score	$\frac{TP}{TP+FN}-\frac{FP}{FP+TN}$	$\{-1,1\}$	1
FAR - False alarm ratio	$\frac{FP}{FP+TP}$	$\{0,1\}$	0
POD - Probability of detection	$\frac{TP}{TP+FN}$	$\{0,1\}$	1
PON - Probability of non-events	$\frac{TN}{TN+FP}$	$\{0,1\}$	1
FSR - Probability of non-detection	$\frac{FN}{FN+TP}$	$\{0,1\}$	0

weather information. The weather forecast was also included to provide additional relevant information. The prototype was tested live for the 2015/2016 season.

4 Results

4.1 Exploratory analysis

The scatter matrix in figure 4.1 indicates that for some variables there is a difference in the distribution of the variable depending on avalanche activity. Particularly for snow depth (SA_x) and wet precipitation last three days (RR_3_y), where values for avalanche days tend to be higher than on non-avalanche days. For maximum mean temperature preceding days (TA_5), the difference is not that clear, but values for avalanche days seems to be slightly skewed towards lower values. For mean wind speed, the values for avalanche days is also more to towards lower values than non-avalanche days.

The differences in distribution seen in the scatter matrix is further supported by the results from the Kolmogorv-Smirnov test (Table 6). The test indicates that there is a difference in distribution, especially for variables related to precipitation, but not all of them. For the precipitation station closest to the avalanche area (x is Botnhamn), 3 out of 4 variables are significant on a $p < 0.05$ level. For the precipitation station further away (y is Grunnfarnes), only 1 out of 4 variables are significant on the same level. For variables related to temperature and wind, the difference is not significant according to the Kolmogorv-Smirnov test. It is worth to note that the Kolmogorov-Smirnov test only consider one variable at a time, so even though these variables seems less important, they may have interaction effects which can be of importance.

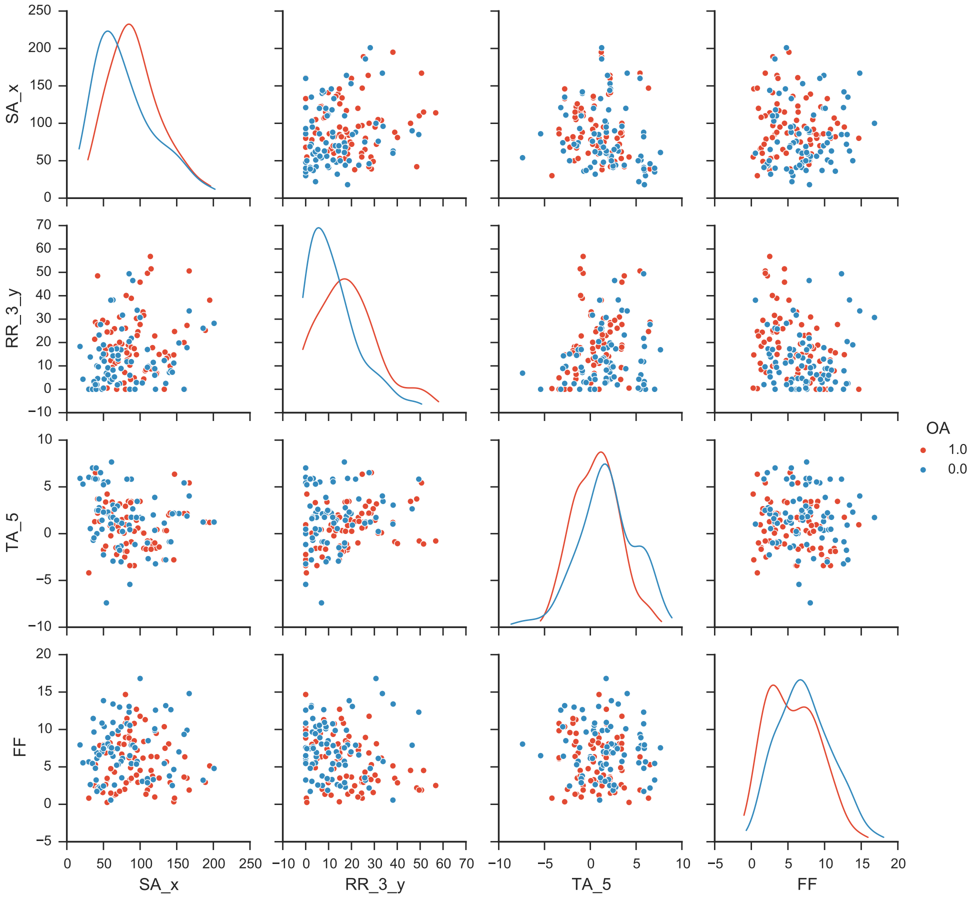

Figure 4.1: *Scatter matrix.*

4.2 Model analysis

A stability selection method applied on a logistic regression model with all variables selected snow depth at both stations (SA_x and SA_y), accumulated precipitation last three days from both stations (RR_3_x and RR_3_y), maximum mean temperature preceding days (TA_5) and change in snow depth last 24 hours at Botnhamn (SA_24_x). Table 7 shows the resulting coefficients when fitting a logistic regression model with these variables to the training data. The coefficients shows that snow depth has the strongest linear effect on avalanche activity. More snow is causing more avalanches. Precipitation last

Table 6: *Mean p-values and standard deviation from the Kolmogorov-Smirnov test where 100 random subsets of non-avalanche days was compared to the sample of avalanche days. The subset non-avalanche days was generated such that the days were from the about the same periods as the avalanche days. This was done to reduce seasonal dependency.*

Variable	mean P-value	Standard deviation
SA_x	0.013	0.02
SA_y	0.12	0.11
SA_24_x	0.039	0.065
SA_24_y	0.27	0.18
RR_3_x	0.01	0.02
RR_3_y	0.002	0.007
RR_24_x	0.05	0.07
RR_24_y	0.1	0.13
TA_5	0.19	0.17
TA_mean	0.19	0.17
TA_max	0.31	0.24
TA_grad	0.35	0.26
FF	0.16	0.16

three days is also positively correlated with avalanche activity. Both these results are supported by previously stated avalanche theory. Maximum mean temperature last 5 days is negatively correlated with avalanche activity. A period without any warm days seems to increase avalanche activity.

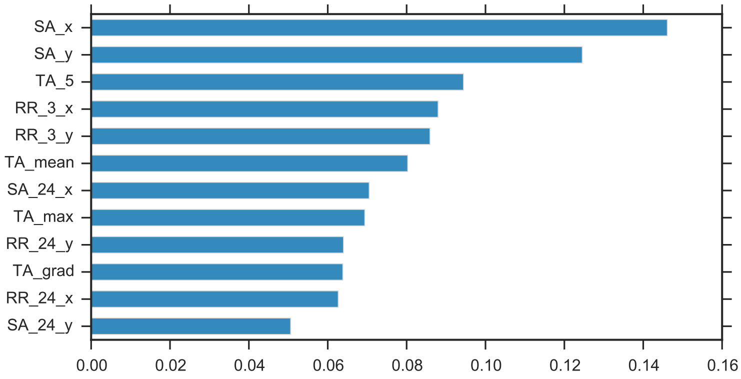

Figure 4.2: *Random forest variable importances for the model where wind was excluded.*

Figure A.1 shows which variables the random forests found to be most important based on mean decrease in node impurity. As with logistic re-

Table 7: *Variables selected when the stability selection method is applied on a logistic regression model. The listed coefficients are from a logistic regression model with the selected variables fitted to training data. The variables were normalized before fitting the model.*

Name	Coefficient
Intercept	-1.92
SA_x	0.71
SA_y	0.18
SA_24_x	0.25
RR_3_x	0.14
RR_3_y	0.35
TA_5	-0.52

gression, snow depth is found to be most important. The top five variables found to be most important by random forest are the same as selected by the stability selection method applied on logistic regression. Further, mean temperature and average wind speed, which is not selected by the stability selection method, is more important in random forest than change in snow depth last 24 hours, but the difference is small. Wind direction seems to be of less importance to avalanche activity in these data.

In the exploratory analysis, the effect of wind seemed to be opposite of what one would expect from avalanche theory. To further examine the importance of wind a random forest model without wind, was also included. The variable importances for this model is shown in figure 4.2. The top five list is unchanged compared to the random forest where wind was included (figure A.1).

4.3 Model performance

Table 8 shows the evaluated scores after the predicted probabilities has been mapped to outcomes. A threshold at 15% was used so an avalanche was predicted for days with a probability above this. Random forest without wind predicts correctly (TP) 12 out of 19 avalanches, and by that obtains the highest probability of detection at 0.63. With 12 false positives it has a false alarm ratio at 0.5. This is better than random forest with wind that detects 9 out of 19. Random forest with wind has with 8 false positives the lowest false alarm ratio at 0.47. Logistic regression with only snow depth predicts correctly 8 out of 19, resulting in a POD at 0.42. With 22 false positives it has the highest false alarm ratio at 0.73. Logistic regression with

Table 8: *Evaluated prediction scores. To calculate the scores, a fixed threshold at 15% has been used, i.e an avalanche is predicted on days with an avalanche probability greater than 15 %. Values in bold are the best.*

Score	Snow depth	Logistic Regression	Random Forest	RF no wind
TN	387	392	**401**	397
FN	11	9	10	**7**
FP	22	17	**8**	12
TP	8	10	9	**12**
RPC	0.68	0.74	0.73	**0.8**
TSS	0.37	0.48	0.45	**0.6**
FAR	0.73	0.63	**0.47**	0.5
POD	0.42	0.53	0.47	**0.63**
PON	0.95	0.96	**0.98**	0.97
FSR	0.58	0.47	0.53	**0.37**

selected variables predicts correctly 10 out of 19 avalanches resulting in a POD at 0.53. With 17 false positives the false alarm ratio is 0.63.

Plotting the predicted probabilities for avalanche season 2011, 2012 and 2013 along with observed avalanche activity shows that both the random forest models and the logistic regression model are better to predict avalanche activity than the simple snow depth model (Figure 4.3 to 4.4). The random forest models and the logistic regression model give similar predictions, i.e are highly correlated. Random forest without wind seems to be best correlated with avalanche activity when inspecting the plot. The difference is small, but the model without wind is detecting some more avalanches and has in general higher probabilities on days with avalanche activity than the model including wind.

For most avalanches, there is a notable change at or before the event, but there are some avalanches that occurs at low predicted avalanche risk and no sign of avalanche activity is indicated in the probability plot.

5 Discussion

5.1 Exploratory and model analysis

The results from the exploratory analysis demonstrates the importance of snow depth and precipitation in snow avalanche formation. For temperature and wind, the importance is not so clear. The scatter matrix in figure 4.1 shows that the two variables with the most significant difference in the

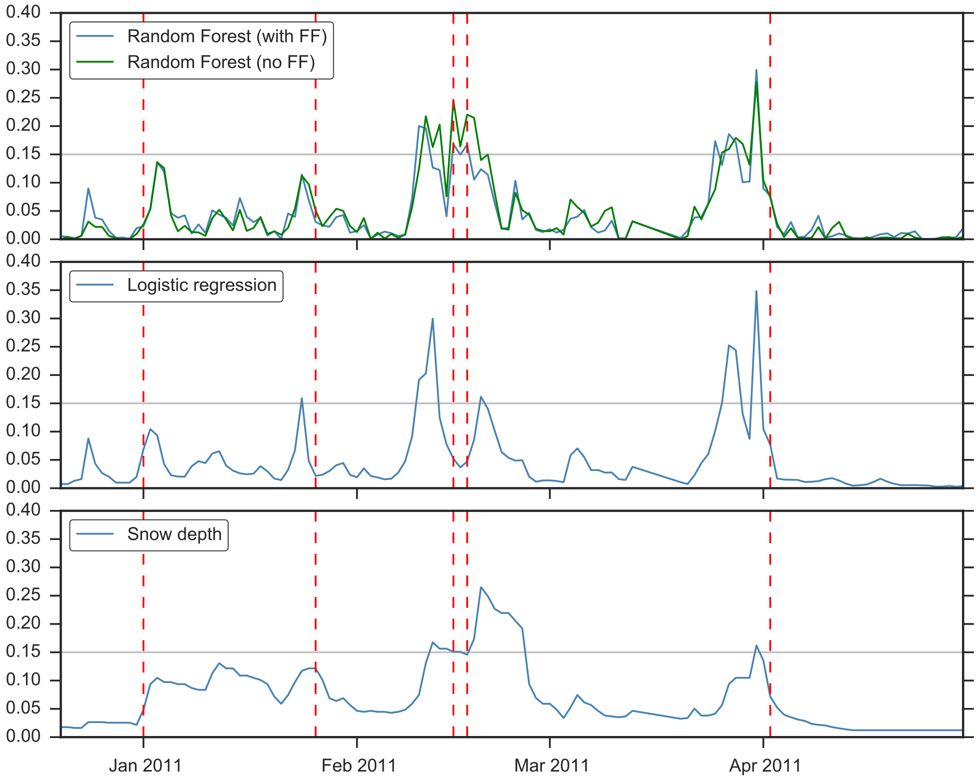

Figure 4.3: *Predicted probabilities for avalanche season 2011. The red dashed lines are true avalanche observations.*

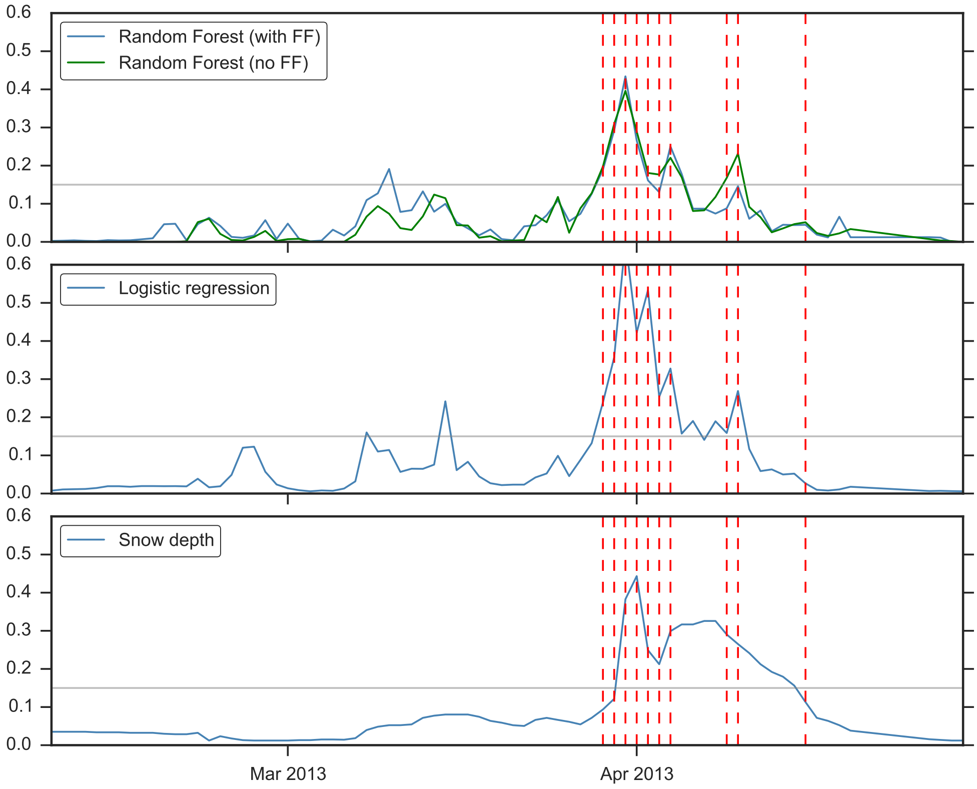

Figure 4.4: *Predicted probabilities for avalanche season 2013. The red dashed lines are true avalanche observations.*

Kolmogorv-Smirnow test (SA_x and RR_3_y), have clear visual differences in the estimated kernel densities. In general it seems to be more snow on avalanche days and the amount of precipitation last three days is greater than on non-avalanche days. This is in agreement with previously stated avalanche theory [Schweizer et al., 2003]. For maximum daily mean temperature last 5 days before the avalanche event (TA_5) the difference is not so clear, but there is an indication to be slightly colder in the period before an avalanche event. One explanation is that if the maximum mean temperature in the preceding period has been high, the snow deck might have had time to stabilize already, or the avalanche was triggered during the period of warmer temperature. Further, the exploratory analysis shows that there is much overlap in the data and one can not easily discriminate between avalanche days and non-avalanche days by only consider one variable. It appears that one has to deal with significant uncertainty. Therefore, using models that can predict probabilities rather than predicting the outcome directly are appealing. A probability contains more information and is more useful to combine with other information such as snow stratigraphy and weather forecast.

The variables selected by the stability selection method applied on the logistic regression model and the variables found to be most important by random forest, coincides with the results from the exploratory analysis. Both models finds snow depth to be most important, and also agrees on precipitation last three days and maximum mean temperature last 5 days. It is encouraging that two different models selects the same variables, and it strengthens the importance of those variables.

Regarding the less importance of temperature and wind, this could be related to the location of the weather station that observes wind and temperature. The distance to the avalanche area is about 17 km and it is placed on a small island further out in the ocean. Measuring wind and temperature out there might not capture the effect of these variables as they are in the mountains where the avalanches are triggered. Additionally, it could be related to the more complex effect wind and temperature has on snow avalanche formation. Snow depth and precipitation is generally more directly coupled to the formation of avalanches (i.e. more snow and precipitation generally means more avalanches), where as temperature and wind has a more complicated effect on snow avalanche formation.

5.2 Model performance

Both the evaluated scores in table 8 and the probability plots in figure 4.3, 4.4 and A.2, indicates that the random forest models and the logistic regression model with selected variables, are better to explain avalanche activity than the simple snow depth model. This suggest that including more information than just snow depth increase the predictive capabilities of the models. The random forest models and the logistic regression model seems to behave similarly, which is expected as they do contain much of the same information. Random forest seems to be favorable over logistic regression as it detects some more avalanches and have fewer false alarm events. This indicates that there could be more complex interactions than linear in the data such that a model that can detect non-linear effects is appropriate. Also, it indicates that the variables excluded by the stability selection method, but are included in random forest, do contain some information.

There are some avalanches that occur at low risk, and when inspecting weather observations related to these days, there is no indication of avalanche risk (no precipitation, no increase in temperature, etc.). These avalanches could be caused by factors not available in the data included here. For example, avalanches late in April might be caused by strong sun radiation, a meteorological variable which is not available. Another example are avalanches caused by deep persistent weak layers that have sustained in the snowpack for a long time. Under such conditions, even small changes in temperature and wind can cause a release of an avalanche. For the models, it will not be possible to distinguish these days from similar days where the snowpack is stable. Two observations that are similar for the models can have completely different snowpacks. This is some of the limitation with models that only consider weather information. Including information about the snow stability could possibly improve this, but this information is simply not available. Limitations like these are reasons to why the models should only be used to support decisions, and not take decisions. For a decision maker, the predicted probabilities should be treated as the risk of an avalanche *given the recent weather conditions only.*

Even though using a model that includes weather information only has its limitations, it has a strong advantage of being automatic. For example, this makes it possible to set this up as an automatic warning system that notifies responsible people when the predicted probability exceeds a certain threshold. Having such a system in addition to the existing workflow could possibly help to detect dangerous situations that would otherwise be undetected.

At last it should be noted that the performance of the models varies greatly from season to season. For these test data, the avalanche season 2013 greatly improves the evaluated scores for the models. Given the limitations discussed above, it seems reasonable that the performance will vary depending on whether the main cause for most avalanches a given season is detectable recent weather changes or if it is more complex snow stability conditions. For the 2013 season, where the main cause for a series of avalanches was a huge snowfall, the performance is very good. But for the 2012 season, two out of four avalanches occurs at very low risk. Thus, testing the models on more data is necessary to better evaluate their performance.

6 Operational testing and prototype

The results from the model analysis indicates that a decision support system based on already existing infrastructure can be useful in an operational setting. Even though we are dealing with great uncertainty and the models needs to be tested on more data, it is still useful to test how such a system would work in practice. This can help to guide further development of the models, identify possible technical issues and to get experience on how such a system can be implemented in the current workflow of determining avalanche risk.

To start testing how a predictive model would work in an operational setting, a prototype was developed. The prototype consisted of a web application that automatically pulled data from the Norwegian Meteorological Institute, processed and stored the data and then used a pre-trained model to predict the probability of an avalanche event. The model used in the prototype was the random forest without wind and it was trained on all available data. To visualize the results, the avalanche probabilities for the last five days were shown as a time series, aligned with observed weather data. Also the weather forecast for the next 24 hours were shown. The prototype was tested live, but not used to take actual decisions.

The prototype gave useful insight on the operational potential and challenges. Including both the observed weather data and the weather forecast, gave a better understanding of the situation than only showing the probability without any further background. Some issues arose as one of the stations providing precipitation observations was unreliable. On several occasions, weather observations were available first after two or three days. Thus, the station is not useful in an operational setting.

7 Conclusion

Based on 17 years of weather and avalanche data, this study has identified meteorological variables important in avalanche formation at Mefjorden, Senja. Exploratory analysis has indicated that snow depth, precipitation last three days and maximum mean temperature the 5 preceding days are weather characteristics that are important in the formation of avalanches at

Mefjorden.

Fitting a logistic regression model and random forest models to the data, confirmed the results from the exploratory analysis about which variables that were most important. Random forest were the model that performed best at predicting avalanches and obtained a true skill score at 0.6. Logistic regression with a selection of variables obtained a TSS at 0.48 and performed better than a simple logistic regression model that only used snow depth and obtained a TSS at 0.37. The results indicates that using a model that is not restricted to linear effects is preferable, and including more information than only snow depth does increase the accuracy of the models.

A prototype was developed and tested live. Combining predicted probabilities with other relevant information is a promising approach for how a decision support system based on weather observations can be set up.

8 Ackowledgements

I would like to thank Statens Vegvesen (The Norwegian Public Road Administration) for taking initiative to this project, particularly Eivind Juvik and Tore Humstad. I am also very thankful for the thorough review comments from Ingelin Steinsland, professor at NTNU.

References

Aldrich, J. and Nelson, F. (1984). *Linear Probability, Logit, and Probit Models*. Number Bd. 45;Bd. 1984 in Linear Probability, Logit, and Probit Models. SAGE Publications.

Atwater, M. M. (1954). Snow Avalanches. *Scientific American*, 190:26–31.

Bois, P., Obled, C., and Good, W. (1975). Multivariate data analysis as a tool for day-to-day avalanche forecast. *International Association of Hydrological Sciences Publications*, 114:391–403.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

Buser, O. (1983). Avalanche forecast with the method of nearest neighbours: an interactive approach. *Cold Regions Science and Technology*, 8(2):155–163.

Chen, C., Liaw, A., and Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*.

Gassner, M., Etter, H., Birkeland, K., and Leonard, T. (2000). NXD2000: An improved avalanche forecasting program based on the nearest neighbor method. In *Proceedings of the 2000 International Snow Science Workshop*, pages 52–59.

Hendrikx, J., Murphy, M., and Onslow, T. (2014). Classification trees as a tool for operational avalanche forecasting on the Seward Highway, Alaska. *Cold Regions Science and Technology*, 97:113 – 120.

Marienthal, A., Hendrikx, J., Birkeland, K., and Irvine, K. M. (2015). Meteorological variables to aid forecasting deep slab avalanches on persistent weak layers. *Cold Regions Science and Technology*, 120:227–236.

McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Norem, H. (2014). *Veger og snøskred (Roads and avalanches).* Number V138 in Håndbøker i Statens Vegvesen.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rossum, G. (1995). Python reference manual. Technical report, Amsterdam, The Netherlands, The Netherlands.

Schweizer, J., Bruce Jamieson, J., and Schneebeli, M. (2003). Snow avalanche formation. *Reviews of Geophysics*, 41(4). 1016.

Schweizer, J., Mitterer, C., and Stoffel, L. (2009). On forecasting large and infrequent snow avalanches. *Cold Regions Science and Technology*, 59(2):234–241.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

A Additional tables and figures

Table 9: *Correlation matrix for the variables used in the study.*

Variable	SA_x	SA_y	SA_24_x	SA_24_y	RR_3_x	RR_3_y	RR_24_x	RR_24_y	TA_5	TA_mean	TA_grad	TA_max	FF	OA
SA_x	1.00	0.82	0.08	0.02	0.07	0.08	0.04	0.05	-0.32	-0.21	-0.05	-0.21	-0.08	0.23
SA_y	0.82	1.00	0.09	0.11	0.04	0.11	0.02	0.07	-0.45	-0.34	-0.00	-0.33	-0.06	0.23
SA_24_x	0.08	0.09	1.00	0.52	0.17	0.12	0.22	0.14	-0.06	-0.26	-0.03	-0.26	-0.08	0.09
SA_24_y	0.02	0.11	0.52	1.00	0.10	0.14	0.09	0.23	-0.05	-0.23	-0.06	-0.23	-0.09	0.05
RR_3_x	0.07	0.04	0.17	0.10	1.00	0.78	0.71	0.52	0.23	0.15	-0.03	0.17	0.01	0.09
RR_3_y	0.08	0.11	0.12	0.14	0.78	1.00	0.60	0.71	0.17	0.17	-0.01	0.19	-0.07	0.11
RR_24_x	0.04	0.02	0.22	0.09	0.71	0.60	1.00	0.67	0.15	0.20	0.00	0.23	0.06	0.06
RR_24_y	0.05	0.07	0.14	0.23	0.52	0.71	0.67	1.00	0.09	0.20	0.10	0.23	-0.05	0.08
TA_5	-0.32	-0.45	-0.06	-0.05	0.23	0.17	0.15	0.09	1.00	0.62	-0.20	0.61	-0.06	-0.09
TA_mean	-0.21	-0.34	-0.26	-0.23	0.15	0.17	0.20	0.20	0.62	1.00	-0.05	0.97	-0.09	-0.07
TA_grad	-0.05	-0.00	-0.03	-0.06	-0.03	-0.01	0.00	0.10	-0.20	-0.05	1.00	-0.03	0.11	0.01
TA_max	-0.21	-0.33	-0.26	-0.23	0.17	0.19	0.23	0.23	0.61	0.97	-0.03	1.00	-0.09	-0.07
FF	-0.08	-0.06	-0.08	-0.09	0.01	-0.07	0.06	-0.05	-0.06	-0.09	0.11	-0.09	1.00	-0.07
OA	0.23	0.23	0.09	0.05	0.09	0.11	0.06	0.08	-0.09	-0.07	0.01	-0.07	-0.07	1.00

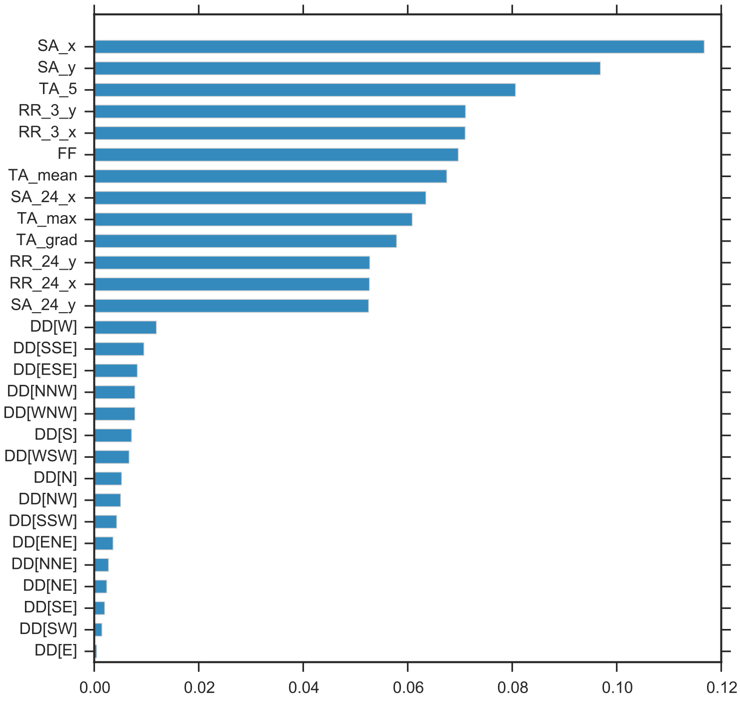

Figure A.1: *Random forest variable importances when wind and wind direction are included.*

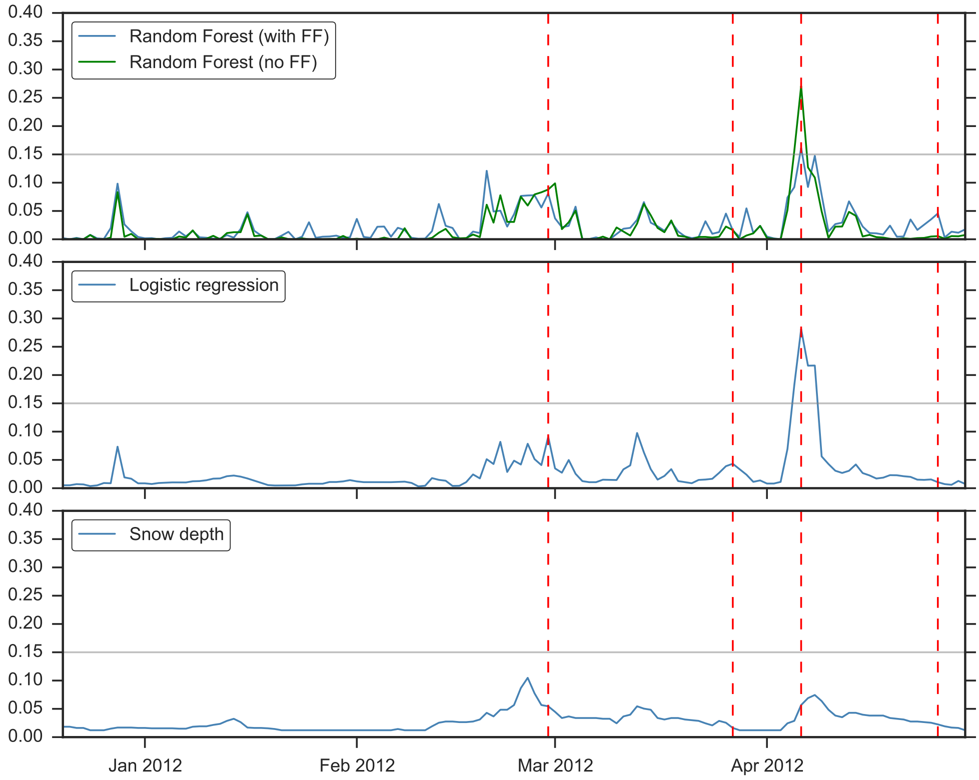

Figure A.2: *Predicted probabilities for avalanche season 2012. The red dashed lines are true avalanche observations.*

Logistic regression and random forest

1 Notation

To explain the methods used in this study, it is useful to define a general notation and relate this to the weather and avalanche data we have. The data consists of weather observations $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and corresponding responses $\{y_1, \dots, y_N\}$ representing avalanche activity. N is the number of observations, i.e. number of days considered. A single weather observation $\mathbf{x} \in \mathbb{R}^p$ is a vector of length p, where p is the number of exploratory variables (also refereed to as features), i.e. snow depth, wind speed, temperature etc. The response $y \in \{0, 1\}$ is a binary value that takes the value 0 for no avalanche activity and 1 for avalanche activity. Together $\mathbf{x}$ and y form a learning set $\mathcal{L}$ consisting of data $\{(\mathbf{x}_i, y_i),\ i = 1, \dots, N)\}$. With this notation, we can define a learning method $\varphi(\mathbf{x}, \mathcal{L})$ as a method that predicts the response y to the observation $\mathbf{x}$ based on the learning set $\mathcal{L}$.

Since we model the risk of an avalanche rather than the direct outcome y, we assume that

$$y \sim \text{Bernoulli}(\pi),$$

so that the response $y \in \{0, 1\}$ take the value 1 with probability π and 0 with probability $1 - \pi$. The learning methods used in this study are set up to model the probabilities π.

2 Logistic Regression

A logistic regression model is a common choice to inspect the relationship between a binary response like "Yes"/"No" and a set of exploratory variables. Here, we will use a linear logistic regression model with link function *logit* [Aldrich and Nelson, 1984]. The model assumes that the log-odds of the probabilities have a linear relationship to the exploratory variables, i.e

$$\text{logit}(\pi) = \mathbf{x}^\top \boldsymbol{\beta} = \boldsymbol{\eta},$$

where $\boldsymbol{\beta}$ is a set of regression coefficients. The log-odds, or *logit* function, is defined as

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1 - \pi}\right).$$

Applying the inverse of the *logit*, we can write the model as

$$\pi = \text{logit}^{-1}(\mathbf{x}^\top \boldsymbol{\beta}) = \frac{\exp\{\mathbf{x}^\top \boldsymbol{\beta}\}}{1 + \exp\{\mathbf{x}^\top \boldsymbol{\beta}\}} = \frac{\exp\{\boldsymbol{\eta}\}}{1 + \exp\{\boldsymbol{\eta}\}}.$$

It is a simple model that captures linear effects in the data such as *more snow → more avalaches*, *less snow → less avalaches*. By normalizing the variables by subtracting the mean and dividing by the standard deviation, the size of the estimated coefficients $\boldsymbol{\beta}$ can be compared. This gives us an indication of how strong the effects are and the variables relative importance. Since the model assumes linearity, more complex relations in the data will not be found by this model.

3 Random Forest (RF)

Random Forest [Breiman, 2001] is a statistical method used both for classification and regression. RF is based upon decision trees and bagging [Breiman, 1996]. To understand how it works, we need to understand these two concepts.

3.1 Decision Trees

Decision trees as a statistical method has been around for a while [Morgan and Sonquist, 1963]. It has shown to be a successful approach for many classification and regression problems. To define a decision tree formally, we use the notation as defined earlier. Let the space of all feasible $\mathbf{x}$ be denoted by $\mathcal{X}$ and similarly for y, $\mathcal{Y}$. We can then define a decision tree φ as a function $\varphi : \mathcal{X} \mapsto \mathcal{Y}$ by checking one or more conditions on $\mathbf{x}$ in a tree structured procedure. Because of this tree structure, decision trees can easily be visualized. Figure 1 shows an example of a tree classifier.

A random forest consists of many decision trees. Each single tree is grown using the CART methodology as described in "Classification and Regression Trees" by Breiman et al. [1984]. Briefly explained this involves picking m random variables and then find the values of these that best categorize the data. In our case, if $m = 3$, the randomly picked variables could be snow depth, mean temperature and new snow accumulation. An algorithm then find the values of these variables that best split the data into avalanche activity and no avalanche activity. The whole procedure is applied recursively on each split of the data until a terminal criteria is met. This is either when all the instances in the node is of the same class, the number of instances is less than a certain value or no possible best split is found. In our case, were we use

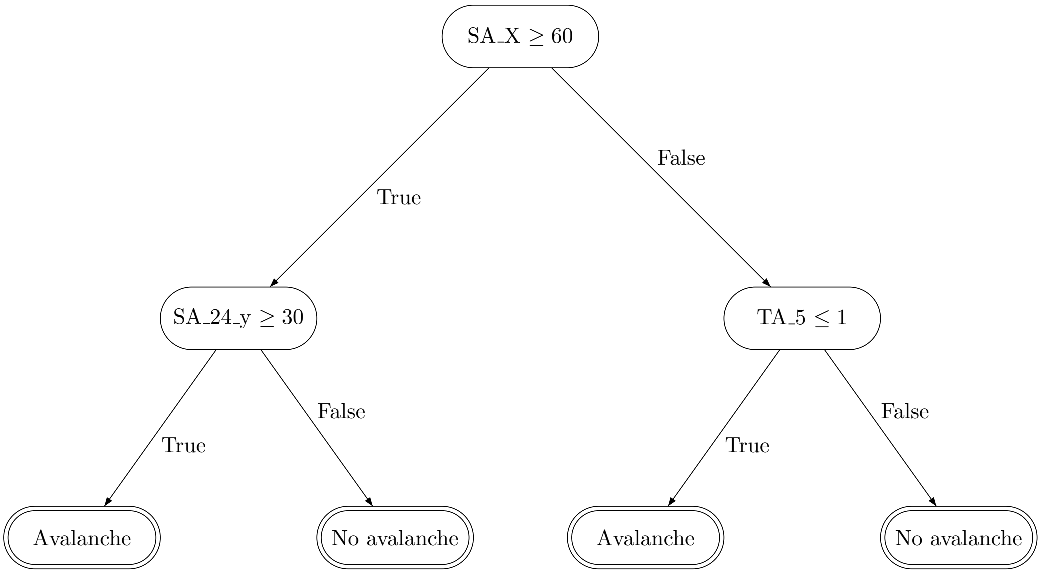

Figure 1: *Example of a decision tree. At each parent node a condition is checked for the weather observation. The conditions defines which way the observation should take down the tree until it reaches a terminal node. The terminal node then use the proportion of the remaining instances (from growing the tree) to cast a vote for which class the observation belongs to, or use them to calculate a probability for which class the observation belongs to.*

random forest for predicting probabilities, the proportion of the remaining instances will be returned instead of a single vote for a specific class. If class weights are assigned, these are used both as weights to determine the best split, and they are used as weights when the proportion of remaining instances in the terminal node are used to cast a vote or predict a probability for a given observation.

3.2 Bagging

If we have N observations we define a learning set $\mathcal{L}$ as the set that consists of data $\{(y_i, \mathbf{x}_i),\ i = 1, \ldots, N)\}$. Assume further that we have a predicator $\varphi(\mathbf{x}, \mathcal{L})$. The notation means that the predicator is predicting the response to $\mathbf{x}$ based on the data from learning set $\mathcal{L}$. Now, create k learning sets $\{\mathcal{L}_1, \ldots, \mathcal{L}_k\}$ by, for each set, draw randomly N samples from the original set $\mathcal{L}$ with replacement. With k learning sets available we can create a set of classifiers $\{\varphi(\mathbf{x}, \mathcal{L}_1), \ldots, \varphi(\mathbf{x}, \mathcal{L}_k)\}$. Let all of these classifiers predict the response y to $\mathbf{x}$. Instead of having a single prediction, we now have an ensemble of predictions. If our problem is a classification problem, we can now pick the class for $\mathbf{x}$ with most votes or, if it is a regression problem, we can use the votes to calculate probabilities for the different classes. This procedure is called bagging, also called bootstrap aggregating, and was first introduced in a paper by Breiman [1996].

3.3 A randomized forest

Assume that we perform bagging on our learning set $\mathcal{L}$, k times. For each learning set $\mathcal{L}_n$ where $n = 1, \ldots, k$, we grow a decision tree as described in section 3.1. The result is a collection of trees, $\{\varphi(\mathbf{x}, \mathcal{L}_1), \ldots, \varphi(\mathbf{x}, \mathcal{L}_k)\}$, also called forest. In stead of having a single tree to predict the outcome of $\mathbf{x}$, we now have a collection of trees that can predict the outcome of $\mathbf{x}$. The idea with RF is that with a sufficient number of trees it should be able to capture most of the usable information available in the dataset. To get good results a large number of trees are required. The general rule is to use as many as computational affordable, since more trees than necessary do not decrease accuracy. A typical default value is 500.

References

Aldrich, J. and Nelson, F. (1984). *Linear Probability, Logit, and Probit Models.* Number Bd. 45;Bd. 1984 in Linear Probability, Logit, and Probit Models. SAGE Publications.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees.* The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58(302):pp. 415–434.

Notes on the implementation

Today, there exists many powerful open source projects that can help scientists do research more efficiently. Especially within data science, where much of the time is spent on cleaning and organizing data, choosing the right tools can save time. Also, many open source frameworks exists to make software development easier. Taking advantage of this, scientists can more easily take their ideas from the paper and turn them into prototypes. Getting feedback from a prototype early in the process gives valuable information, and it can help to present the research to others, for example people from the industry who could benefit from the research. Letting people see how the idea works in practice make it easier for them to understand how they can use the result. For applied research, that deals with solving practical problems, early development of a prototype helps to make sure that one solves the actual problem and that practical issues are detected early in the process.

This chapter briefly explains how the study was conducted technically and describes the implementation of the prototype. It is not a complete documentation of what has been done or how things work. I made a separate software package, `metnopy` [Hennum, 2016], to make it easy to retrieve weather data and some examples of usage are included. Hopefully, this chapter can give some inspiration on how similar projects and prototypes can be implemented.

1 Software

The language of choice for this project was `python` [Rossum, 1995]. As a high-level programming language with a rich ecosystem of packages, it is a language suitable for many tasks. Especially within scientific computing there exists many packages for doing common computational operations. This makes `python` an ideal choice for projects like this. For all the various operations throughout the project, it was possible to stick to one programming language. With one language, it is easier to connect different parts of the workflow together and it saves time as it require less learning than using two languages. `python` is also free and open source.

For data analysis, `pandas` [McKinney, 2010] was used. This python package provides high performance, easy-to-use data structures and a rich set of functions for typical data analysis operations. Among the data structures `pandas` provides are data frames. This structure is much like a spreadsheet, it contains columns and rows. Most of the data in this project was represented as data frames. With support for smart date indexes and functions for most kinds of common transformations, the data preparation necessary in this project was easy to implement. Under the hood `pandas` uses `numpy` [Van Der Walt et al., 2011] which provides multi-dimensional array objects to python. Most of `numpy` is written in C which ensures high performance. Both `pandas` and `numpy` is a part of the Scipy [Jones et al., 2001] ecosystem of open source software.

For fitting the models, `scikit-learn` [Pedregosa et al., 2011] was used. It is a huge library and contains implementations for the most commonly used machine learning methods. With its easy-to-use application interface and a strong community of researchers who supports it, it has become a very popular machine learning library. As a result, many good learning resources are available online. Further, `scikit-learn` and `pandas` are easy to use together, so one can easily fit models to the data structures provided by `pandas`.

2 Retreiving data

2.1 Weather data

All the weather data used in this study is retrieved from the Norwegian Meteorological Institute. They provide a web service, eKlima [The Norwegian Meteorological Institute, 2016], for accessing weather data and export it in CSV text format. It is possible, but cumbersome, to export data for research purposes with this service. In an operational setting, data retrieval should be automated, this does not work. For accessing weather data from applications, the Norwegian Meteorological Institute provide a web service which allow users to query the weather database with HTTP requests and retrieve weather data in XML format. This makes it easy for users to implement weather data access in their applications, but the XML format the data is returned in is not of a form which can be imported and used directly in python. To make it easy to retrieve weather data in a usable format, I developed a separate python package for this purpose only. The package was developed such that one can query eKlima directly from python and retrieve the data as a pandas data frame. In the returned data frame, dates are converted into a date-time index and the weather variables are casted to the correct type (int, float, etc.). The returned data frame can then be used directly without any further formating. This makes it very convenient to do exploratory analysis on weather data, or build an application that make use of weather data. The amount of coding necessary for accessing weather data and do some simple analysis, is reduced to a minimum. Splitting this part of the project out as a separate python package makes it easy for others to reuse the code for their own projects. The package is called `metnopy` [Hennum, 2016] and is available online with full source code and documentation. Two examples are shown under to give some ideas on how this package can be used.

Example 1: This example shows how air temperature (weather code TA) at 11 o'clock from the 10th of June 2015 to the 15th of June 2015 at Blidern, Oslo (station nr. 18700) and Voll, Trondheim (station nr. 68860) is retrieved. The function parameters in `get_met_data` are the same as used in the official web service.

```
In [1]: from metnopy import get_met_data

In [2]: get_met_data("2", "18700,68860", "TA", "2015-06-10",
   ...:              "2015-06-15", "11", "")

Out[3]:
                    TA_18700 TA_68860
date
2015-06-10 11:00:00     18.4      9.8
2015-06-11 11:00:00     20.1      9.8
2015-06-12 11:00:00     23.1     12.2
2015-06-13 11:00:00     18.8      9.5
2015-06-14 11:00:00     17.5      7.7
2015-06-15 11:00:00     16.5      6.8
```

Example 2: This example shows how one can plot the combined annual mean temperature for January, February and March at Blindern, Oslo from 1931 until 2016 together with a rolling mean and the average for the whole period. With an easy way to get data and the built-in power of pandas, it doesn't require much coding to do analysis on weather data.

```
import matplotlib.pyplot as plt
import pandas as pd
```

```
from metnopy import get_met_data

data = get_met_data("2", "18700", "TA", "1931-01-01",
                    "2016-03-31", "", "1,2,3")

data.groupby(data.index.year).mean()["TA"].plot()

pd.rolling_mean(data.groupby(data.index.year).mean()["TA"],
                window=5, center=False).plot()

plt.axhline(y=data["TA"].mean(), color="red")

plt.legend(["Annual mean", "Rolling mean (5yr)", "Overall mean"],
           loc=4)
plt.ylabel("Temperature [C$^\circ$]")
plt.xlabel("Year")
```

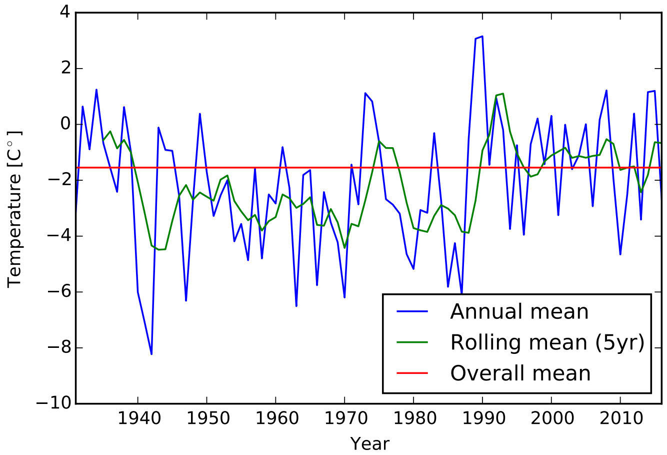

Figure 1: *Resulting plot from example 2. Shows the annual mean temperature for January, February and March at Blindern, Oslo, from 1931 to 2016, together with a rolling 5 year mean and the overall mean.*

2.2 Avalanche data

Statens Vegvesen [2016] has recently made a web service were much of their

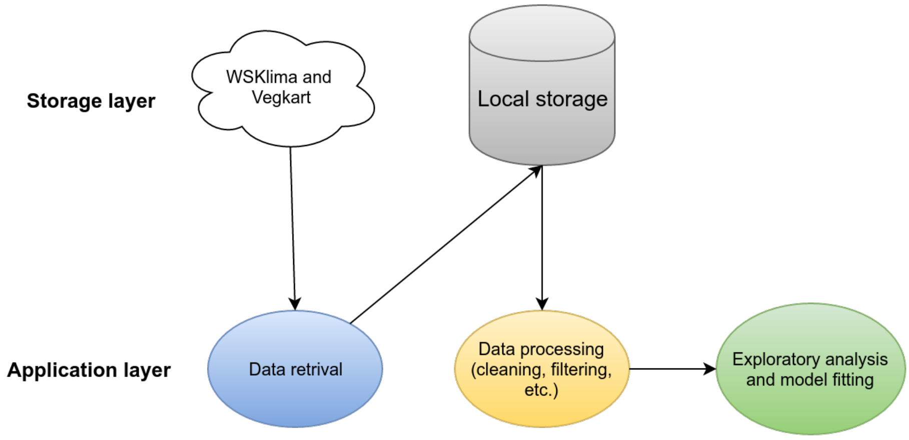
Storage layer
WSKlima and Vegkart
Local storage
Application layer
Data retrival
Data processing (cleaning, filtering, etc.)
Exploratory analysis and model fitting

Figure 2: *Shows how the data flows from the sources into the analysis process.*

data is publicly available. With this web service, it is possible to retrieve avalanche data in JSON format. The returned format did not require much coding to parse into a `pandas` data frame. Having such an API as they provide, with several filtering and format options, greatly simplifies the data gathering process. This function was not separated out of the project as it easy to implement and it is of limited use to others.

3 Data processing and model fitting

To process the weather data into the required format for further analysis (like the meteorological variables described in Chapter 2), a library of functions was developed. For every transformation that was necessary to apply to the data, there was a corresponding function. Having a set of small functions made it easier to keep control over the data transformation. Also, tests were written to ensure that the functions worked as intended. When the code base grow bigger and changes were made, having tests were crucial to make sure that the result was as expected and to avoid introducing bugs. The functions was then combined into a pipeline where raw data was retrieved and then transformed into the format required for the analysis and for model fitting with `scikit-learn`. Figure 2 shows how the dataflow was in the study. By setting up a pipeline consisting of modular functions, it was easy make changes by just setting changing function parameters somewhere in the pipeline, and then run the complete analysis again.

4 A prototype

A simple working prototype should be able to automatically retrieve new weather data, predict the probability of an avalanche and then visualize the result. Thus, to accomplish this, we needed:

- A predictive model trained on historical data
- A task to automatically retrieve new weather data
- A task to process new weather data into the meteorological metrics used in the predictive model
- A task to predict the avalanche probability for new data
- A way to store the data
- A way to visualize the results

As the code developed during the study was made as reusable modules and functions, the same code could be used to put together a prototype. This made the development of the prototype quite simple, and was more or less about gluing together already existing functionality and then automate the whole process. Figure 3 shows a flowchart of the prototype. First, a SQL database was set up with the necessary tables and columns to store the data. Secondly, based on the results from the study, a random forest was trained on historical data and stored (purple). Then, the following tasks were set up:

- Retrieve new weather data and store it in the database (blue)
- Read new weather data from the database, process the data into the format required by model and store it in the database (yellow)
- Read the processed data from the database, predict the probability of an avalanche event and store it in the database (green)

Each of these tasks were set up as separate services. This makes it easier to track down errors when the program fails. To automate the process, a task runner was configured to run the services sequentially every day at appropriate times. To avoid duplicates in the database, the database was set up with multiple constraints to ensure that rows in the database were unique. This made it possible to run the services many times without worrying about getting duplicates in the database.

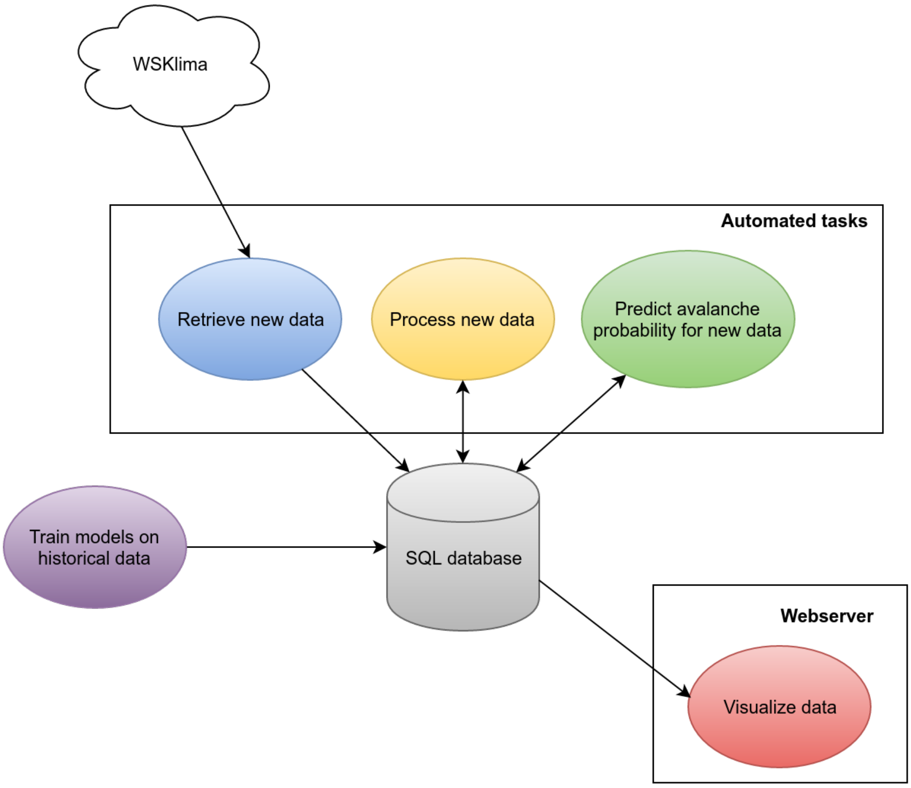

Figure 3: *Shows how the data flows in the prototype.*

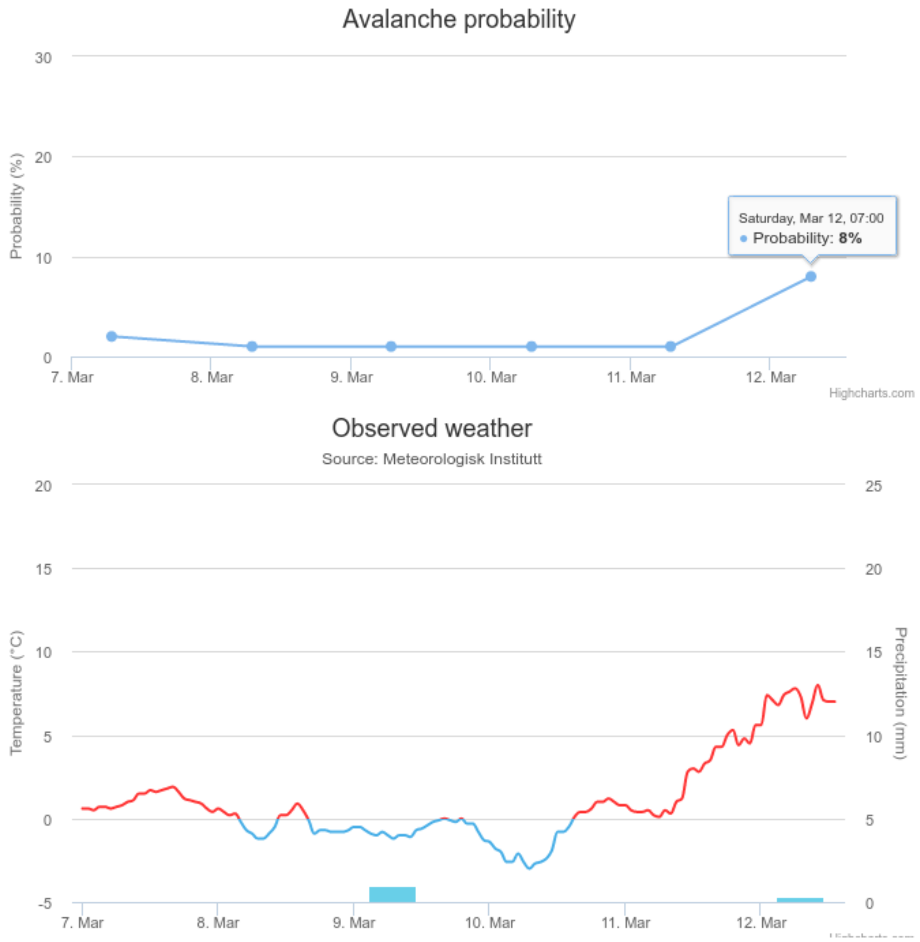

Figure 4: *A screenshot of the frontend. The map and the weather forecast is left out here to save space.*

To visualize the data, a simple web application was developed. The application was written in `flask` [Ronacher, 2016], a lightweight python framework for developing web applications, and Highcharts, a JavaScript library for drawing graphs. The application simply read the raw weather data and the avalanche probabilities from the database, and then graphed the data on a webpage. It also included a map over the area were the road stretch the forecast was valid for and the weather stations in use were marked out. The final frontend is shown in figure 4. The prototype was hosted in the cloud by using Amazon Web Services and is available at `http://52.19.132.210:5000/`.

References

Hennum, A. A. (2016). MetNoPy: A python package for retrieving weather data from the Norwegian Meteorological Institute. `github.com/hennumjr/MetNoPy`.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. `scipy.org`, Acessed online 2016-03-12.

McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ronacher, A. (2016). Flask: A microframework for python based on werkzeug, jinja 2 and good intentions. `flask.pocoo.org`, Acessed online 2016-03-12.

Rossum, G. (1995). Python reference manual. Technical report, Amsterdam, The Netherlands, The Netherlands.

Statens Vegvesen (2016). Vegkart: Public information about the Norwegian road network. `vegvesen.no/vegkart`, Acessed online 2016-03-12.

The Norwegian Meteorological Institute (2016). eKlima: Free access to weather- and climate data from norwegian meteorological institute from historical data to real time observations. `eklima.no`, Acessed online 2016-03-12.

Van Der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30.