Project 3

Logistic Regression and Support Vector Machine

FYS-STK4155: Applied data analysis and machine learning
Department of Geosciences, University of Oslo

Mina Tangen and Tham Le

https://github.com/minatang1/project3

December, 2019

Abstract

The seemingly easy question of whether precipitation fell as rain or snow is not as straightforward as it may seem. Correctly determining precipitation phase is of crucial importance to hydrological modeling [8]. Liquid water and solid water (snow) behaves very different in a catchment, and errors in precipitation type classification can therefore propagate into other parts of the model. Methods used in hydrological models today are typically too simple to capture the full complexity of the boundary-layer processes that govern precipitation type.

In this project, two machine learning (ML) methods have been used to classify precipitation phase. These methods have included other meteorological variables in addition to near-surface air temperature (T_a), which is the most used parameter in models today.

We found that Support Vector Machines (SVM) performed better at classifying precipitation type than the method using only T_a. Logistic regression performed even worse than the single temperature threshold method.

Contents

1 Introduction

The water cycle, shown in figure 1, describes how water continually moves through the Earth system and changes between the liquid, gaseous and solid phases: Liquid water evaporates from the land surface and oceans into the atmosphere as vapor, where clouds are formed [1]. When the air is saturated, vapor particles can condensate to form either solid og liquid particles, depending on cloud and atmospheric characteristics. When the condensed cloud particles grow large enough, gravity leads them back down to the surface as precipitation. Most of the precipitation formed at our latitudes originates as snow from cold clouds, but can melt on its way down and thus precipitation falls as both solid and liquid in our region [12].

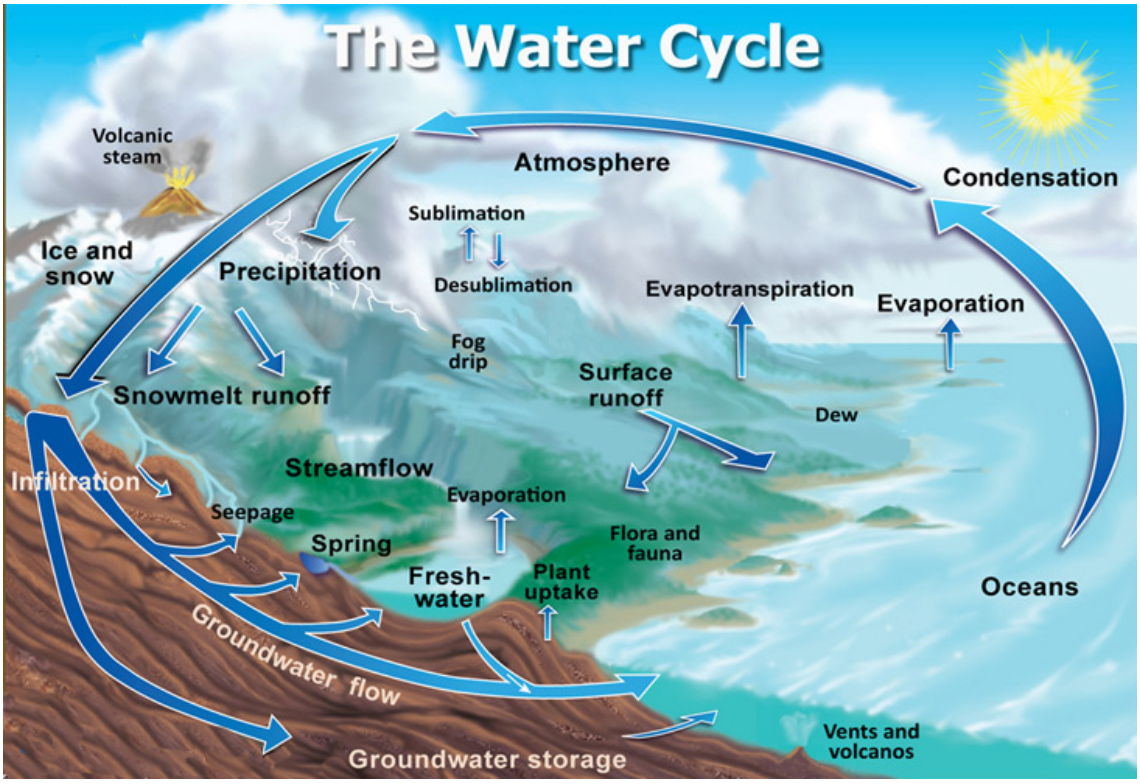


Figure 1: The hydrological water cycle [14].

The purpose of a hydrological model is to simulate the hydrological cycle or parts of it [1]. Hydrological models consist of different routines representing the different parts of the water system. Correctly determining precipitation phase is of crucial importance to hydrological modeling, as errors in classifying precipitation type can propagate into other parts of the model [8]. Snow and rain behave very differently in a catchment [2] . If precipitation falls as snow, water is stored until the snowpack melts in the spring. Rain, on

the other hand, will enter the rivers and streams in the catchment and affect stream flows immediately. This means that spring flow might be less than expected if precipitation that really fell as rain is reported as snow. Too much snow will also give a higher albedo, which will affect energy budgets.

Hydrological models are quite simple in terms of the input data they require. Typically, only temperature, discharge and precipitation data is needed. The latter is normally given as a quantity in units of length, without any information on precipitation type. This means that we do not know if the precipitation fell as rain or snow, and it is up to the model to determine precipitation phase. The most common method to determine precipitation phase used in hydrological models today are simple and based on near-surface air-temperature alone. Precipitation phase is the results of complex meteorological processes in the boundary-layer and using temperature only will not give an accurate representation of those processes [12]. In the literature, there has therefore been developed other methods including additional variables, such as humidity, to improve the accuracy of the models [8]. These methods have generally performed better than methods using temperature only, but results varies across different climates and locations.

The biggest challenge we are facing today is climate change. As the climate is changing, so must our models. In order to adapt, we must dare to challenge the methods traditionally used in hydrology. Numerical models allow us to study how climate change will affect our catchments. It is therefore interesting for hydrologists to look to the world of machine learning algorithms to find solutions to the climate crisis.

In this project, we aim to use machine learning methods to classify precipitation type. We have applied logistic regression and Support Vector Machines (SVM) to this classification problem. But which method will give the best accuracy?

2 Study area and data

We have used data measured at at time interval of 10 minutes from *Fv-37 Jønjiljo* station, lThe station is located in Notodden municipality in Telemark county, as shown in figure 2. It is located at 200 m.a.s.l. The station is run by Statens Vegvesen (SVV). Data from October 1st to December 1st 2019 was used, however, as explained in implementation below, there was a lot of missing data from the beginning of the period. To retrieve data, we used SVV's vegvesen.no/vegvar and frost.met.no (run by the Norwegian Meteorological Institute (MET)). Figure 3 shows the air temperature and dew point temperature from this station during the period we looked at. The dew point temperature is the temperature an unsaturated parcel of air must reach in order to be saturated [1].

Figure 2: Map showing the study area Jonjiljo located in Nototodden in Telemark [4].

Both the feature and target data have quality flags. The quality categories used for the data from MET is shown in table 1. See table 1 for values of the two quality categories for the data used in this project. Although of great importance when dealing with large datasets in natural sciences, we have chosen to somewhat ignore the quality flags of the data, simply because we wanted to focus more on implementing the actual algorithms rather than creating a "perfect" dataset. A "perfect" dataset is also not achievable, so we decided to still use the data set in this project. See Discussion for additional thoughts on errors in the dataset and how those errors may affect the results produced in this project.

EXPOSURE CATEGORY	DESCRIPTION
1	Fullfulls all WMO's demands on location
2	Unknown location, assumed good
4	Unknown location, assume bad
5	Bad location
PERFORMANCE CATEGORY	**DESCRIPTION**
A	The sensor type fulfills the requirements from WMO/CIMOs on measurement accuracy, calibration and maintenance
B	Rutines for calibration and maintanance are known. Control of the montage exists. The precision of the measurment is lower than the WMO/CIMO requirements
C	The sensor type is assumed to fulfill the WMO/CIMO requirements. Missing measurement for control, rutines for calibration, or maintanence
D	The sensor type is assumed to fulfill the WMO/CIMO requirements. Some controls show deviations from the WMO/CIMO requirements
E	Less valuable. Possibly useful for extraordinary purposes. Unknown performance

Table 1: *Quality categories of MET data. Based on* frost.met.no/dataclarifications.html

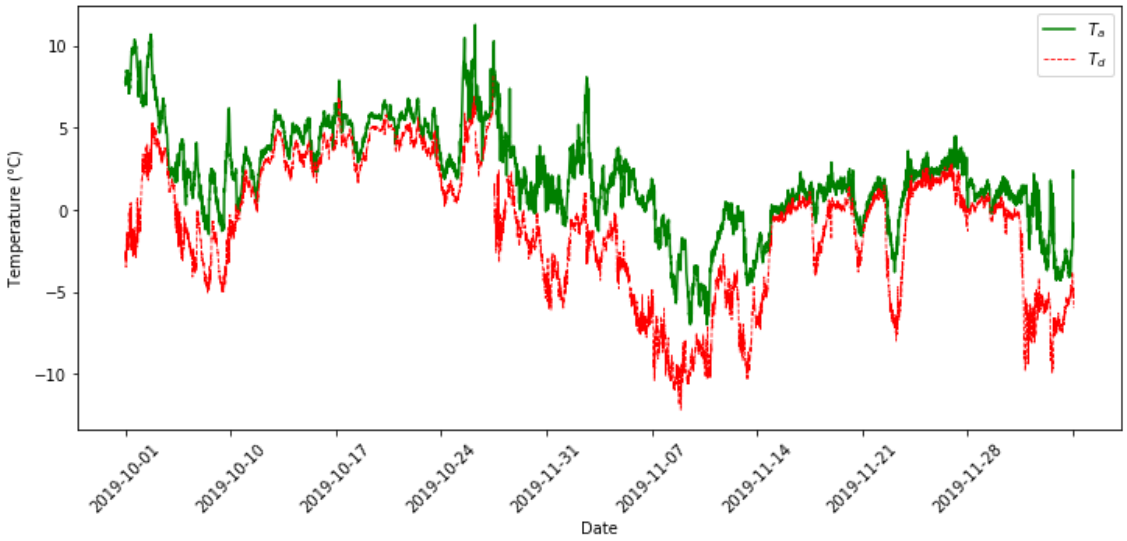


Figure 3: The air temperature and dew point temperature for station Fv 37 Jønjiljo.

2.1 Data preparation and cleaning

The precipitation type data from SVV is given as description of type (strings) in Norwegian. These had to be converted to binary data, i.e. to categories snow and rain, see tables 3 and 2. After converting the precipitation type into numeric codes, we chose a subsample containing only values with measured precipitation, i.e. we filtered out all dates where precipitation type is set as 9999 is removed. This drastically removes the number of samples, from 8610 values (for each feature) to 4536.

The meteorological data included as feature data are shown in table 4.

Before applying the methods to the data, we had to clean the data. There were a lot of dates with missing information on precipitation. Some of the other variables were also missing values. All these instances were removed. This again reduces the amount of instances to 1060. As shown in Figure 4, the data set is skewed, and contains more instances of rain than snow.

SVV name	**SVV name (translated)**	**Conversion**
Opphold	No precipitation	9999
Snø	Snow	1
Regn	Rain	0
Yr	Drizzle	0
Uspesifisert	Unspecified	9999

Table 2: Conversion of SVV precipitation type names. Drizzle is considered as rain, although there might be instances of freezing drizzle. As this is not indicate in the data, and drizzle would not contribute greatly to the total precipitation amount (especially at a time scale of 10 minutes), it is classified as rain.

Type	(Binary) code
Snow	1
Rain	0
No precipitation	9999

Table 3: Binary codes for precipitation type classes

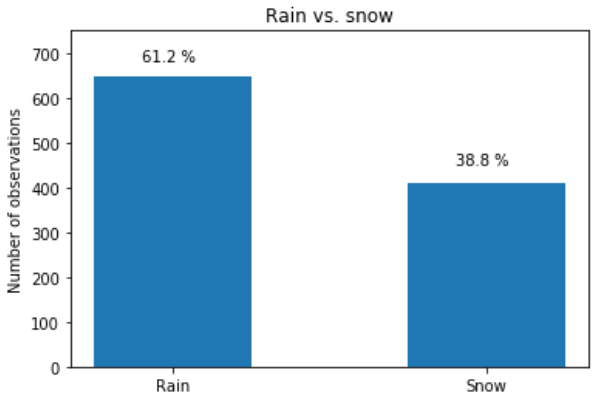


Figure 4: Rain vs. snow histogram

VARIABLE	**UNIT**	**SYMBOL**
Air temperature	$°C$	T_a
Relative humidity	Percentage	RH
Dew point temperature	$°C$	T_d
Wind speed	m/s	w_u

Table 4: Meteorological variables included as feature data. All variables are measured at 2 meters above the ground, except for wind speed, which is measured at 10 meters above the ground.

2.2 Preprocessing

We split the data into training and tests set using *train_ test_ split* from **scikit-learn** [1] (70 % training data and 30 % test data). After this we used *StandardScaler* to scale the data.

[1] This is the only machine learning library used in this project. If not referenced specifically, **scikit-learn** was used

3 Methods

3.1 Single temperature threshold method

The most used method for classifying precipitation **(P)**. in hydrological models today is the single temperature threshold method [8] This method is relatively simple and is based on equation 1.

$$P = \begin{cases} R, & \text{if} \quad T_a > T_c \\ S, & \text{if} \quad T_a \leq T_c \end{cases} \tag{1}$$

I.e., if the temperature is below a certain critical temperature threshold T_c, precipitation is classified as snow **(S)**. If the air temperature is above T_c, precipitation is classified as rain **(R)**. T_c must be calibrated for different locations.

3.2 Logistic Regression

In project 2 [13] we used logistic regression, the same procedure will be used in this project. The logistic model classify the observed data x_i to be binary outcomes such as passed/failed, winning/losing, spam/not spam etc. [10] Classification aimed to predict the outcome correctly by finding the patterns based on discrete variables. The probability that x_i belongs to a category $y_i = (0, 1)$ is given by the sigmoid function 2.

$$f(x) = \frac{1}{1 + \exp^{-x}} \tag{2}$$

and the probability of these two classes with y_i are either 0 or 1 and $\hat{\beta}$ are the weights.

$$p(y_i = 1|x_i, \hat{\beta}) = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0) + \beta_1 x_i)} \tag{3}$$

$$p(y_i = 0|x_i, \hat{\beta}) = 1 - p(y_i = 1|x_i, \hat{\beta}) \tag{4}$$

We can use the **Maximum Likelihood Estimation** to find the total likelihood for all possible outcomes from the dataset $D = (y_i, x_i)$, with the binary labels $y_i = (0, 1)$. The Maximum Likelihood Estimation is y_i [10] defined as

$$P(D|\hat{\beta}) = \prod_{i=1}^{n} [p(y_i = 1|x_i, \hat{\beta})]^{y_i} [1 - p(y_i = 1|x_i, \hat{\beta})]^{1-y_i} \tag{5}$$

from which we can obtain the log-likelihood and our cost function

$$C(\hat{\beta}) = \sum_{i=1}^{n} (y_i \log p(y_i = 1|x_i, \hat{\beta}) + (1 - y_i) \log[1 - p(y_i = 1|x_i, \hat{\beta})] \tag{6}$$

3.3 Support Vector Machine

Support Vector Machine is a simple algorithm but yet highly desired in machine learning since it can be used for both regression and classification and produces significant accuracy. The objective of the algorithm is to find the optimal hyperplane in an n-dimensional space where n is the number of features, a optimal hyperplane is one that clearly manages to classify the data points [3]. There are many possible ways to choose a hyperplane to classify the data points. We want to find the optimal one which has the maximum margin, also to the maximum distance between the data points in both classes. Maximizing the margin will strengthen the classification of future data points.

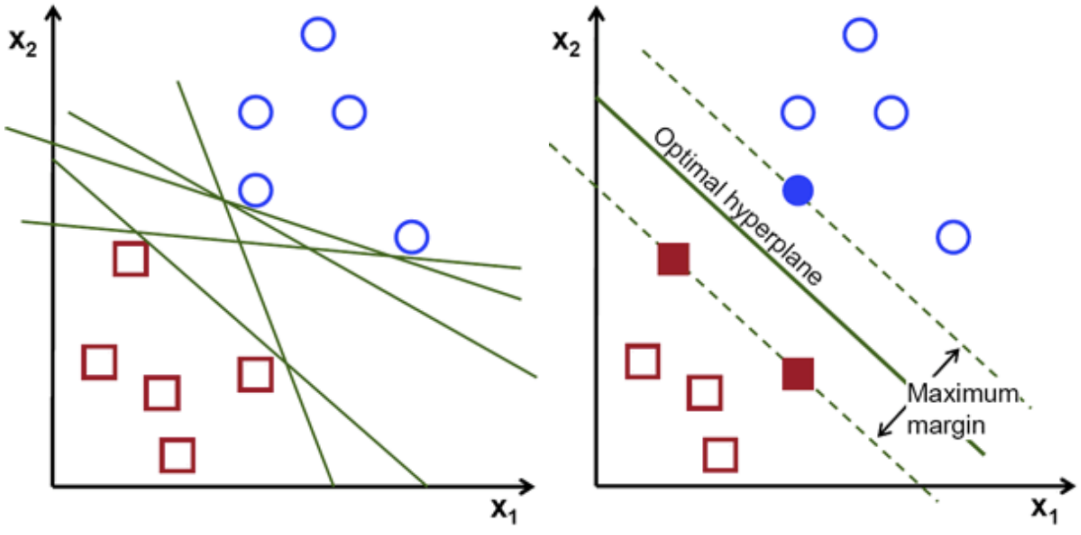


Figure 5: Possible hyperplanes [3].

3.3.1 Hyperplanes and support vectors

Hyperplanes are fundamental in SVM, they determine where the boundaries are set to classify the data points. The data points are assigned to a certain class depending on which side of the hyperplane the data point falling on. The dimension of the hyperplanes depends on the number of input features [3]. If there are 2 input features, the hyperplane will be a line (6). When the number of input of features exceeds 3, it will be difficult to illustrate it.

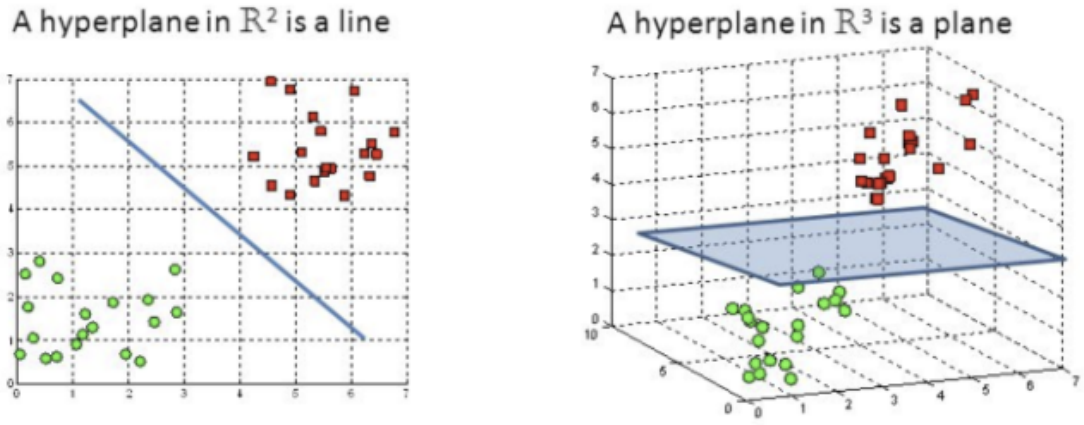


Figure 6: Hyperplanes in 2D and 3D space [3].

Support vector is data points that are closer to the hyperplane, these determine the position and orientation of the hyperplane. These data points thats build our SVM, so by removing the support vectors will change the position of the hyperplane. By using these support vectors we will maximize the margin of classification [3].

3.3.2 Cost function and Gradient Updates

We want to maximize the margin between the data points and the hyperplane. The cost function that helps us with that is **hinge cost function** [3]

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if} \quad y \cdot f(x) \geq 1 \\ 1 - y \cdot f(x), & \text{else} \end{cases} \tag{7}$$

The cost function is 0 if the predicted and observed value give the same output. If not, we can calculate the cost function. We add a normalization parameter to the cost function,

the normalization parameters is to balance the maximization of the margin and loss. The loss function with regularization parameter would look like

$$\min_w \lambda = ||w||^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+ \tag{8}$$

To find our Gradient Updates, we take partials derivates of the cost function with respect to the weights.

$$\frac{\delta}{\delta w_k} \lambda ||w||^2 = 2\lambda w_k \tag{9}$$

$$\frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & \text{if} \quad y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases} \tag{10}$$

We can now update our weights by using the gradients, α is the learning rate

$$w = \begin{cases} w - \alpha \cdot (2\lambda w), & \text{no misclassification} \\ w + \alpha \cdot (y_i \cdot -2\lambda w), & \text{misclassification} \end{cases} \tag{11}$$

3.3.3 Kernel function

As mentioned, SVM is a simple algorithm but highly preferred in machine learning. This is due to kernel functions that we can use to perform better in both non-linear and high-dimensional tasks [16]. The kernel trick allow us to operate in the original feature space and offer a more efficient way to transform data in higher dimension without computing the coordinates in a higher dimensional space.

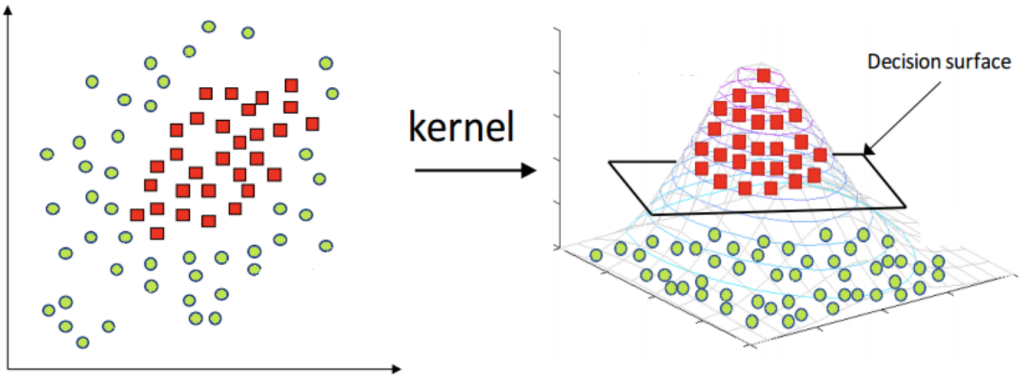


Figure 7: Kernel function maps the data from 2D to 3D-space to find the hyperplane [16].

There are many different kernel tricks, the most common one are the radial basis function (RBF) kernel. The RBF kernel is a non-linear function of Euclidean distance defined as

$$k(\mathbf{x}, \mathbf{y}) = e^{-\gamma||\mathbf{x}-\mathbf{y}||^2}, \qquad \gamma > 0 \tag{12}$$

where the values depends on the distance from the origin (or from some point) and by using the distance in the original space we calculate the dot product of $\mathbf{x}$ and $\mathbf{y}$ [16].

$||\mathbf{x} - \mathbf{y}||$ is the Euclidean distance and γ controls the shape of the "peaks" where the points raise as shown in figure 7. A large value of gamma (γ) give a "softer" peak in higher dimensions, while a small γ gives a more pointed one.

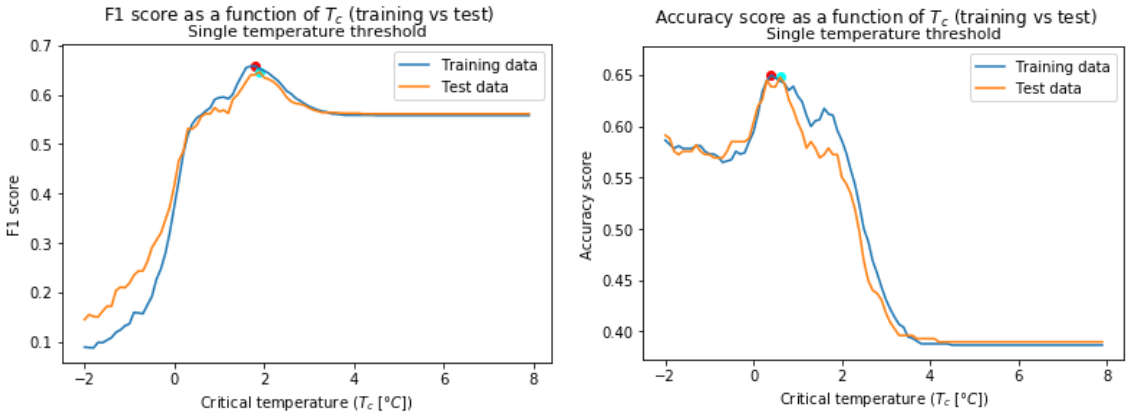
4 Implementation

4.1 Evaluation metrics

We applied mean squared error (MSE), accuracy score, F1 score and confusion matrix to evaluate the performance of the methods. As the dataset is unbalanced, we focus mainly on using the F1 score and confusion matrix.

4.2 Single temperature threshold

We applied the simple temperature threshold method. As explained above, this method uses a critical temperature, T_c, to classify precipitation type. To determine this critical temperature for our station, we looped through a list of values ranging from -2 °C an 8 °C using the training data set. Figure 8a and 8b shows how the F1 and accuracy scores respectively, varies with the T_c for test and training data. The dots show the optimal value for the critical temperature, i.e. whcih value for T_c that gave the highest F1 and accuracy scores. For F1 this temperature was found to be 1.9°C and 0.6°Cfor the accuracy score. These are substantially different, however, as explained, the F1 score gives the best estimate for our case due to imbalance in the data set. The optimal value for T_c was then used when running the model for the testing data, to get the final prediction. Note that the steps of determining the optimal T_c using the training set corresponds to calibration of a hydrological model.



(a) F1 score

(b) Accuracy score

Figure 8: F1 score and accuracy score for critical temperature T_c.

4.3 Machine learning methods

Similarly to the previous method, we found an optimal regularization parameter C for logistic regression by running through a set of values. We found the optimal value of C to be 0.0091, i.e. a very strong regularization. Figure 9 shows how the F1 scores varies with the regularization parameter.

We used the standard kernel *'RBF'* for SVM.

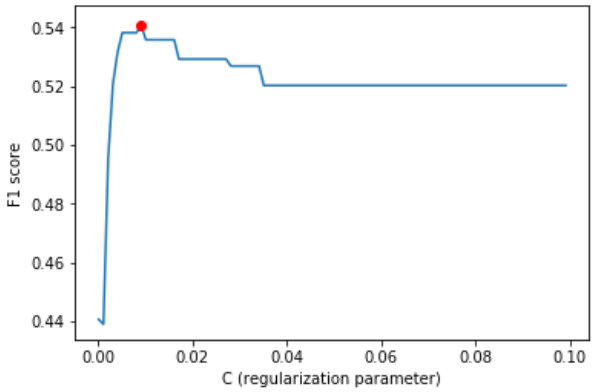


Figure 9: F1 scores varies with the regularization parameter.

4.4 Time series and machine learning assumptions

Most machine learning algorithms are based on the assumption of iid data, i. e. data that is independent and identically distributed. This means that each data point should be independent from the others and come from the same distribution. When dealing with time series data, this assumption is often violated, as time series data might have a strong autocorrelation. In our case, we have time series of temperature, humidity and other variables of the Earth system that clearly depend on itself the day before. This can lead to problems when applying machine learning algorithms. An important step of machine learning is to split the data randomly into a training and test set. The data is normally shuffled randomly before being split into two different sets. If we randomly shuffle a temperature time series and then split the data, we will have values in the different sets that depend on other values. This will give the model high variance and a low generalization error.

One way of avoiding the problems of autocorrelated data is to select sub-samples containing values with lower dependence. In a temperature time series the first values will depend on each other, but at some point, a chosen value will not depend on the first value as strongly as the values closer to it (in time). To decide when this occurs, we

look at the autocorrelation of the different time series. The subplots in figure 12 shows the autocorrelation of the feature data. Autocorrelation will never be zero, but we can loosely define a threshold to decide when we think the variables are no longer strongly dependent. As we can see, the autocorrelation of the different variables varies greatly. For wind speed, we can define a threshold after only about 200 lags, whereas air temperature is still dependent after 1250 lags. In our dataset, we would therefore be left with extremely little data if we were to choose this approach to avoid problems caused by dependent data. In addition, the autocorrelation plots were done for the whole dataset, and as explained above, we are left with little data after cleaning. Obviously, having that little data will also cause problems. Therefore, in this project, we chose to use the full data set without changing it to deal with dependence. If the errors associated with this is not depending on the chosen algorithm, then at least all algorithms will have the same error in them (as we are using the same train and test sets for all algorithms).

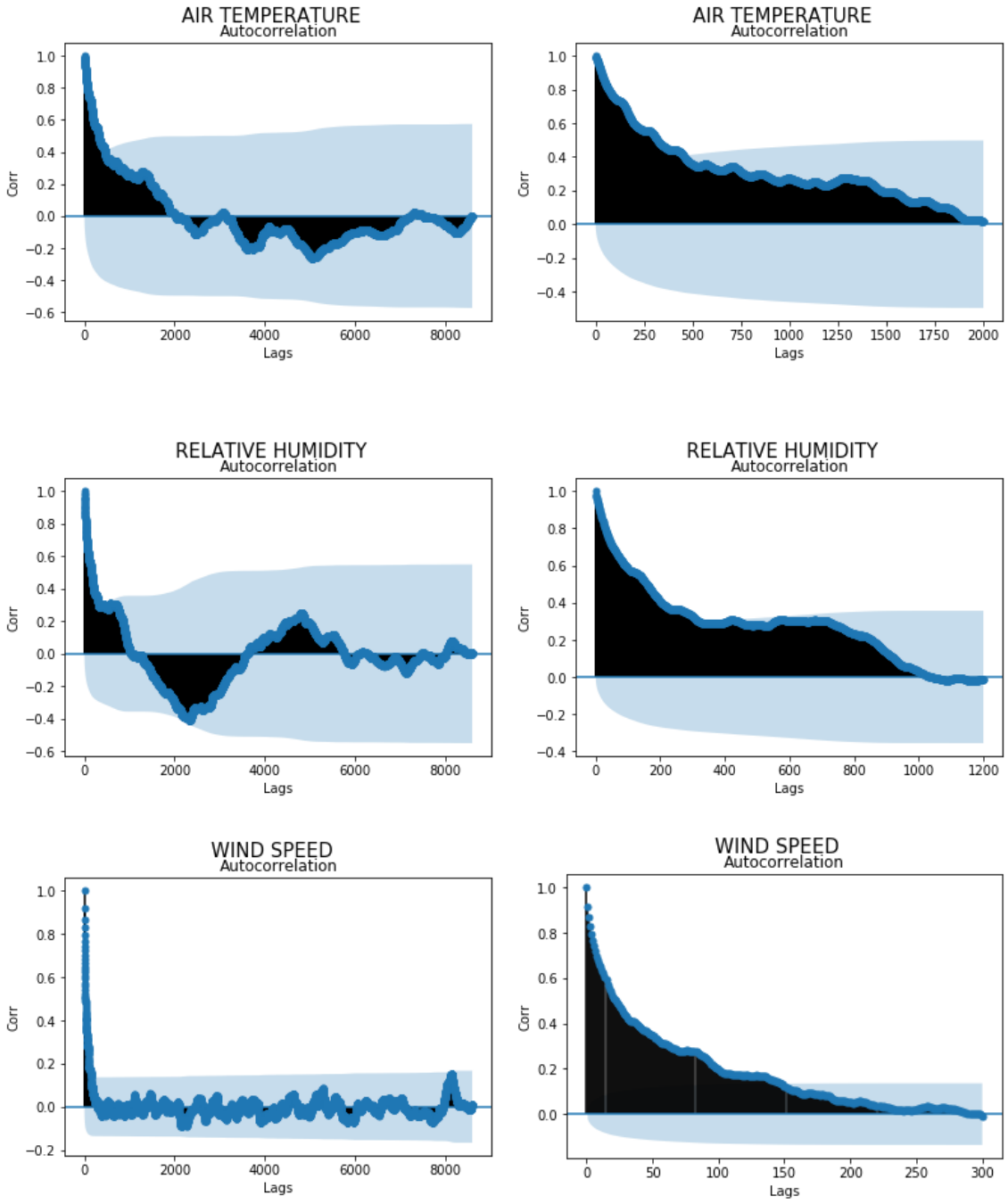
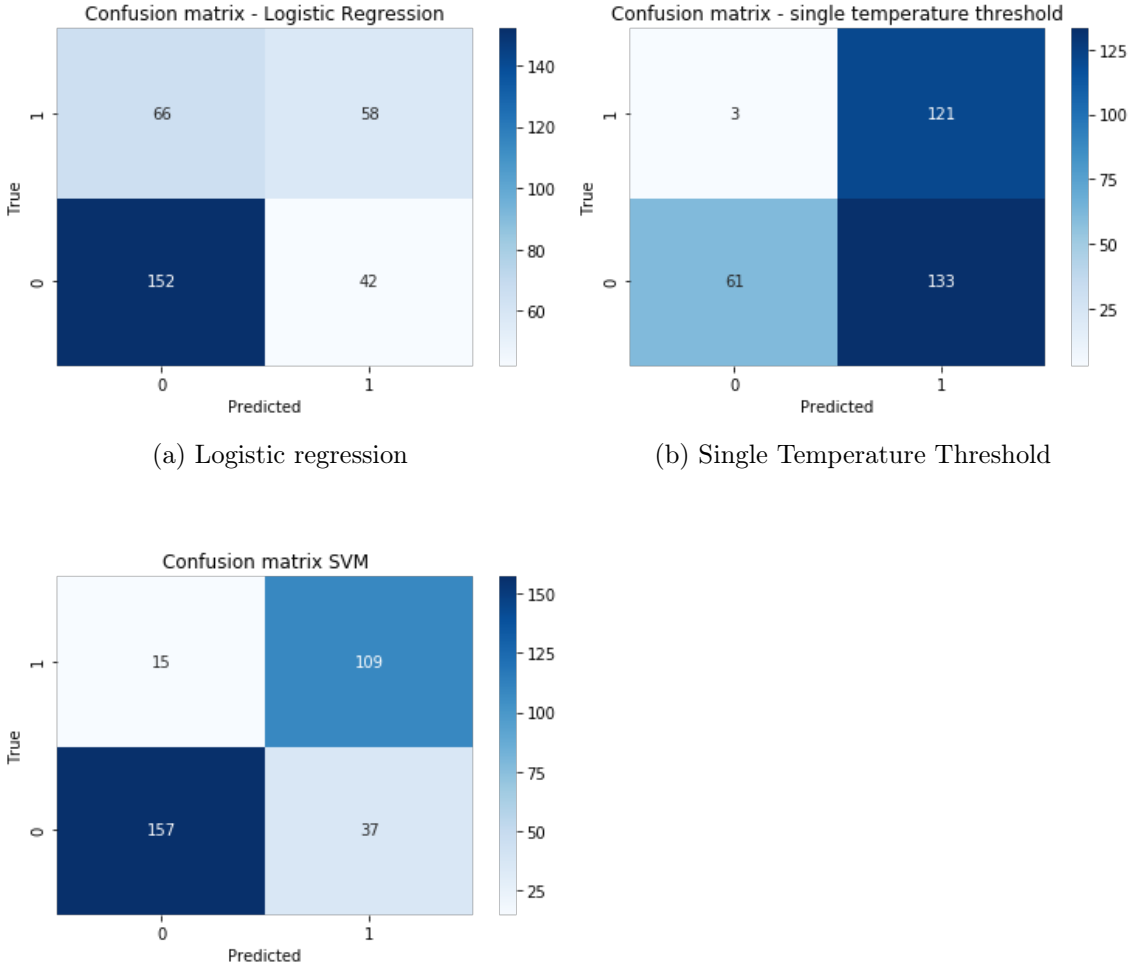


Figure 12: Autocorrelation of the feature data

5 Results



(a) Logistic regression

(b) Single Temperature Threshold

(c) Support Vector Machine

Figure 13: Confusion matrices for all three methods.

	T_a	**Logistic regression**	**SVM**
F1 score	0.64	0.52	0.81
MSE	0.43	0.34	0.16
Accuracy	0.57	0.66	0.84

Table 5: Evaluation metrics for the three methods.

6 Discussion

As we can see in table 5, SVM performs best of all the methods, scoring best at all evaluation metrics. The confusion matrix for SVM is shown in Figure 13c. The method is not perfect, but gives a decent classification of precipitation type, at least compared to the other two methods. Logistic regression performs worst of all methods. The single temperature threshold method performs a bit better than (but comparable to) the logistic regression method when looking at score values in table 5, although not for the accuracy score. When looking at the confusion matrices however (figures 13b and 13a for single temperature threshold and logistic regression respectively), we see an interesting difference in how the two models classify incorrectly. The single temperature threshold method predicts a lot more instances of snow than what is really reported. This simply means, as the method only relies on a temperature threshold, that a lot of the air temperatures measured (in the test data), was below this threshold temperature. This suggests that the method used to determine this threshold is not the best. Logistic regression does the opposite and classifies most precipitation as rain; the method is unable to both correctly and incorrectly classify snow. This might have to do with the fact that the data is skewed and the amount of test data.

6.1 Support Vector Machine

Support vector machine have been used in many other studies in the field of hydrology, in Raghavendra and Deka's article [11] they review previous studies with applications of SVM. The article states that the support vector machine represents the most important development for coping with hydrological parameters, while traditional regression models were found to be inefficient and provided low accuracy [9]. It was also found that SVM is sensitive to the choice of kernel functions, but it is possible to achieve good results by optimizing the parameters in SVM and selecting kernel. This was not carried out in this project, but is an important consideration for further applications of the support vector machine in hydrology.

6.1.1 Advantages and disadvantages of Support Vector Machine

SVM is capable of producing significant accuracy and provides good robust classification results, regardless of whether the dataset is linear or not. This helps to evaluate the relevant information from the data in a more practical way. The data points that have been embedded as support vectors in the training set contribute to SVM being more likely to correctly classify future data points based on the experience of the training set. [6].

Another important feature of SVM is that it automatically identifies and incorporates support vectors during the training part to prevent the algorithm from being influenced by non-support vectors. This makes the model work well with noisy conditions. [6]

Although SVM is known as one of the best algorithms in machine learning, there are some drawbacks. Choosing an appropriate kernel function and parameters depending on the trial and error process is a time-consuming approach. In this project we only tested with a kernel function, so we did not get tested with different kernel functions and see how SVM performs. SVM linearizes the data set using the kernel function, which means that the accuracy of the results is not a result depending on the quality of human expertise assessment for the optimal choice of the kernel function for nonlinear input data [5] and SVM only produces point error statements and is not designed for probability forecasts. [11]

6.2 Time series and machine learning assumptions

As shown above, we found that SVM performs better than the single temperature threshold method and logistic regression.

As motivated above, a better classification algorithm would benefit hydrological models. However, hydrological models are, as explained, quite simple in terms of the amount of input variables they require. Implementing machine learning methods like those used in this project in actual hydrological methods will be difficult, because these methods include many meteorological variables that are typically not available or necessary for hydrological models (as pointed out by Harder & Pomeroy [7]). In other words, ML methods for classifying precipitation phase might simply be too complex to implement in a hydrological model. In atmosphere-coupled hydrological models however, they might prove useful.

It is also worth noting that we have only used data from one station. As shown in [8] methods including variables like humidity, have shown different results depending on location and climate. This might be the case for the methods we have used as well. It would be interesting to see how elevation and local/regional climate zones would affect the results.

Most hydrological models run on a daily scale. To apply the methods we have tested in this project, we would have to change the data into daily data. For numerical values, such as temperature values, humidity values etc., we can for instance take the mean to obtain a value that represent that day. For precipitation type, however, it is not as easy. As the data is categorical, there is no obvious way to statistically find one precipitation class that represents that day, in the case of days with reported instances of both classes. One option is to take the mode of the categorical data, i.e. choose the most frequent

value. However, it might be the case that the most frequent precipitation type during one hour, contributed with a smaller amount of precipitation than the other type did. I. e., there might be 98 instances of snow and 46 of rain during a day, but if those 46 instances of rain produced a more significant amount of precipitation than the 98 instances of snow, taking the mode would not represent the precipitation type for that hour in a satisfying way. This can be a challenge when applying these methods to hydrological models.

6.3 Improvement potential

In addition to include more stations and to care more about data quality, we have identified potential areas of improvement. If we had more time, we would have wanted to test resampling techniques such as k-fold cross validation to tune our models and find the best model parameters (not simply loop through values as we did in this project). We would also have liked to look into the imbalance of the data set, by choosing subsamples with an equal amount of rain and snow instances. In order to do this, we would have to have much more data than we had in this project. Ideally, we would have enough data so that we could make subsamples that are less dependent (by looking at autocorrelation, as describes above)

It would also be interesting to test the classification methods with actual values of precipitation amount, i.e. it would be interesting to see how classifying incorrectly affects the actual amount of precipitation, to possibly get a feeling for how much this will affect the streamflow.

7 Conclusion

The single temperature threshold method is the most used method in hydrological modeling to classify precipitation. We found that it does not perform well for our station. Our aim was to develop a method that would better classify precipitation. We applied two machine learning methods to try to improve classification performance. By including other meteorological variables than air temperature, we hoped that performance would increase, as it is suggested in the literature that air temperature alone is unable to capture the full range of processes determining phase. Support vector machines proved to be much better at classifying precipitation type. However, logistic regression proves to be worse than the single air temperature method. We can therefore not conclude that the inclusion of other variables alone will improve the result, as this clearly depend on the chosen method. We would like to test the methods for larger data sets of better quality, as the data set used in this project suffers from both dependence (violation of iid assumption) and poor quality at the source, in addition to being skewed. Further, we would have liked to test the methods for other climatic location to see if there is any difference in the performance.

It would also be interesting to do a feature selection analysis in order to see which variables actually matter.

References

[1] Dingman, S. L., 2015. *Physical Hydrology: Third Edition.* Waveland Press.

[2] Feiccabrino, J. et al., 2013. *Surface-based precipitation phase determination methods in hydrological models* . © IWA Publishing 2013.

[3] Gandhi, R., 2018. *Support Vector Machine - Introduction to Machine Learning Algorithms.* https://towardsdatascience.com/support-vector-machine-\introduction-to-machine-learning-algorithms-934a444fca47

[4] Google Earth Pro, 2018. earth.google.com/web/.

[5] S.Gunn, 1998. *Support vector machines for classification and regression.* ISIS Technical Report.

[6] D. Han. et al. 2017. *Flood forecasting using support vector machines.* Journal of Hydroinformatics 9, p.267-276.

[7] Harder, P. & Pomperoy, J.W., 2014. *Hydrological model uncertainty due to precipitation-phase partitioning methods.* Centre for Hydrology, University of Saskatchewan, 117 Science Place, Saskatoon, Saskatchewan, S7N 5C8, Canada

[8] Harpold et al., 2017. *Rain or Snow: Hydrologic Processes, Observations, Prediction, and Research Needs.*. Department of Geosciences, Boise State University.

[9] Jian, Y.L. et al., 2006. *Using support vector machines for long term discharge prediction.* Hydrological Sciences Journal 51, p.599-612.

[10] Hjort-Jensen, M., 2019. *Data Analysis and Machine Learning: Logistic Regression.* Department of Physics, University of Oslo.

[11] Raghavendra, S. & Deka, P.C., 2014. *Support vector machine applications in the field of hydrology: a review* Department of Applied Mechanics and Hydraulics National Institute of Technology Karnataka, Surathkal, India.

[12] Stewart, R.E., 2015. *On the Characteristics of and Processes Producing Winter Precipitation Types near 0 °C.* University of Manitoba, Winnipeg, Manitoba, Canada.

[13] Tangen, M. & Le, T., 2019. *Classification and Regression: from linear and logistic regression to neural networks.* Department of Physics, University of Oslo.

[14] Perlman, H. & Evans, J. 2019. (USGS) *The Natural Water Cycle* https://www.usgs.gov/media/images/water-cycle-natural-water-cycle

[15] Wolff et al., 2015. *Derivation of a new continuous adjustment function for correcting wind-induced loss of solid precipitation: results of a Norwegian field study.* Hydrology and Earth System Science (EGU).

[16] Zhang, G., 2018. *What is the kernel trick? Why is it important?* https://medium.com/@zxr.nju/what-is-the-kernel-trick-\why-is-it-important-98a98db0961d