Predicting Avalanche Risk Factor using Machine Learning Techniques

Joel Kerfoot
V00855134
kerfootj@uvic.ca

Jonathan Skinnider
V00207396
jonathanskinnider@uvic

Tyler Harnadek
V00818721
tylerhar@uvic.ca

Rowan Burns-Kirkness
V00819316
rowanbk@uvic.ca

1 Introduction

Avalanches are one of the greatest threats facing outdoor enthusiasts in the winter. To assess the risk, organizations around the world prepare and distribute risk analyses. In Canada this work is done by Avalanche Canada and the data and reports are publicly available. Weather, snow, and risk data were compiled together into datasets for training machine learning models to predict avalanche danger ratings. Achieving 70% or greater accuracy would be a success and would mimic the results of A. Pozdnoukhov published in "Applying machine learning methods to avalanche forecasting" [1]. Support Vector Machine (SVM), Nearest Neighbour, and Random Forest models were used for this project.

1.1 Problem

Public avalanche forecasts are critical to the safety of non-professional users recreating in the backcountry. Avalanche forecasters combine years of experience, field observations and weather data to assign a danger rating (Low, Moderate, Considerable, High, Extreme) to different elevation bands (below treeline, treeline, and alpine). We are seeking to replace the years of experience and field observations with a machine learning algorithm that accurately predicts avalanche danger while only having access to historic weather data. We have decided to only predict for the alpine elevation band, as that is the most dangerous for large deadly avalanches.

2 Dataset

The dataset was sourced from the BC Government's Automated Snow Monitoring Stations. The two monitoring stations that we selected (3A22P and 3A25P) are located in the Sea-to-Sky avalanche forecasting region. This region encompasses much of the southern west coast of BC.

Data from the weather stations was combined with Avalanche Canada's forecasts to generate a data set. In total, data was collected for dates between January 2014 and December 2018, encompassing around 1800 days. We were able to select nearly 950 dates that contained both an avalanche forecast and sufficient weather monitoring data.

2.1 Measurements

Hourly snowfall and temperature measurements were sampled from each station. As the model is built around daily predictions the hourly samples were combined to create the following metrics:

Prepared for SENG 474: Data Mining, at the University of Victoria, Spring 2020.

- Total accumulated snow depth
- Change in snow depth over the course of the day
- Minimum daily temperature
- Maximum daily temperature
- Mean daily temperature

Temperatures are recorded in Celsius. Snow depth is measured using a Snow Water Equivalent (SWE) gauge, which approximates snow depth by measuring the mass of the snow. Temperature, snow depth, and the amount of snow falling/melting should encapsulate the mountain conditions as they relate to avalanche likelihood.

The dataset label is the danger rating for the day, as assigned by Avalanche Canada. The rating is a number from 1-5, where 1 is the lowest risk and 5 the highest risk. This is the best available source for quantifiable avalanche risk.

2.2 Normalization

The dataset had two issues that we fixed via normalization: first, the temperature data was clustered closely together below zero; second, the snowfall measurement was on a larger scale than the temperatures by a factor of 10 (SWE ranged 0-661). All of the temperature and snowfall measurements were normalized into a 0-1 scale. We decided to normalize the data from the two stations together to capture the difference in conditions at the two locations. If the stations had been normalized individually, it would be impossible to compare between them.

Scikit-learn's MinMaxScaler was used to perform all of the normalization.

2.3 Multi-day Datasets

Avalanche conditions build up over many days, so a single day's forecast is not enough to accurately evaluate risk. To capture this context we produced 4 separate datasets, containing 1, 2, 3, and 4 days worth of data. Each day's worth of data contains 10 columns: min, max, and mean temperatures, and total and daily snowfall for each of the two stations. Dataset 0 contained just the current day's data. Dataset 1 contained the current day's data as well as 1 previous day's data, and so on. We named the datasets Dataset 0, 1, 2, and 3, where the number indicates the number of previous days worth of data. Each model was tested with all four generated datasets.

We refer to these datasets as "dataset 0" or "dataSet_N_0" in the remainder of the report.

2.4 Classification Goals

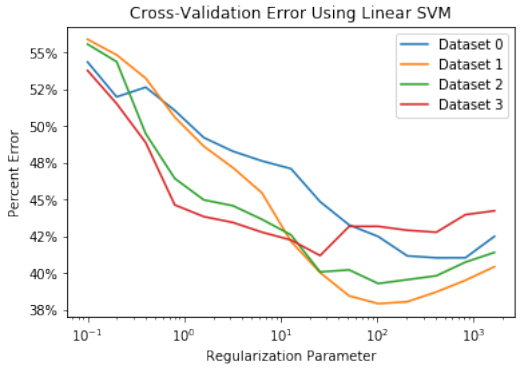
A classification that exactly matches Avalanche Canada's risk factor for a given day is considered a success. Because there are 5 potential classifications, 1: low through 5: extreme, a classifier that randomly selects a label would have an accuracy of 20%. Given the significant safety risk the goal would be to correctly classify more than 70% of the samples. An additional consideration will be given to classifications that are one higher on the risk scale. For example, if a model predicts a risk factor of 4 while the expected risks was 3 this will be considered a partial success. While not an exact match, predicting one point higher is safer than predicting a lower risk but, it is only considered if the value is immediately adjacent.

3 Results

Presented below are the experiments which were performed on the dataset. Support Vector Machine, Nearest Neighbour, and Random forest are tested and compared. 5-fold cross validation was used as the primary means of determining train error while a clean validation set was reserved to report a validation error for well-preforming methods.

3.1 Support Vector Machine

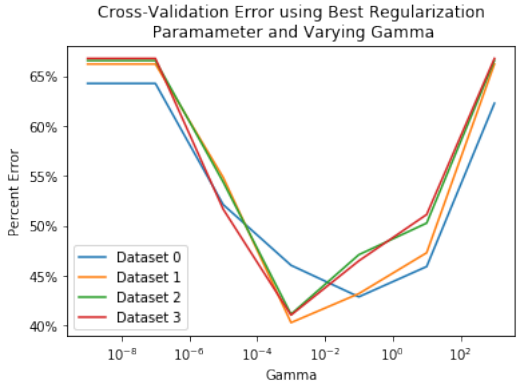
We did preliminary testing with a linear SVM on the four data sets, and used 5-fold cross-validation to determine how much regularization to apply. Each dataset followed a very similar trend, requiring a relatively high regularization parameter to succeed. Surprisingly, dataset 1 which only considered the previous day's weather performed the best. It was able to achieve a cross-validation error of 37.9%.



Using the optimal regularization parameter found above for each dataset, the final algorithms where able to achieve a test error of 57.5% on dataset 0, 35% on dataset 1, 37.5% on dataset 2, and 42.5% on dataset 3. These results show that given the very limited information of dataset 0, even the optimal algorithm was not able to do better than guessing. This supports our intuition that an avalanche forecaster must have knowledge about historic weather events to make an accurate report.

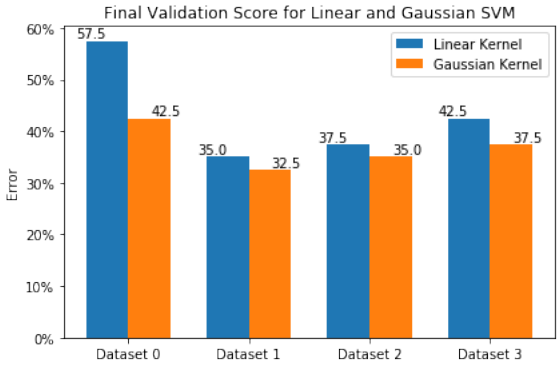
More interesting is how the test error on dataset 2 and 3 was higher than that on dataset 1. When adding the additional the previous days weather, the probability that it is linearly separable would intuitively go down, especially with highly convoluted weather data. In addition, many of the data points are clustered densely around a local mean over several days, making it more challenging to separate.

We also trained a non-linear SVM with a Gaussian kernel. We varied both the regularization parameter and Gamma in order and used 5-Fold cross-validation in order to find the best parameters. Bellow is a figure with the best training scores as γ varies.



On dataset 0, the best result was with $C = 1000.0, \gamma = 0.1$, however datasets 1,2, 3 all did the best with $C = 10000.0, \gamma = 0.001$. Using these optimal hyperparameters we trained the SVM's on the full training set and then tested on fresh test data. Our final results where 42.5% test error on dataset 0, 32.5% test error on dataset 1, 35% on dataset 2, and only 37.5% test error on dataset 3.
Again, the Gaussian SVM performed the best using dataset 1. The following figure shows our final validation scores for the optimally trained SVM's on both datasets. The Gaussian kernel outperforms

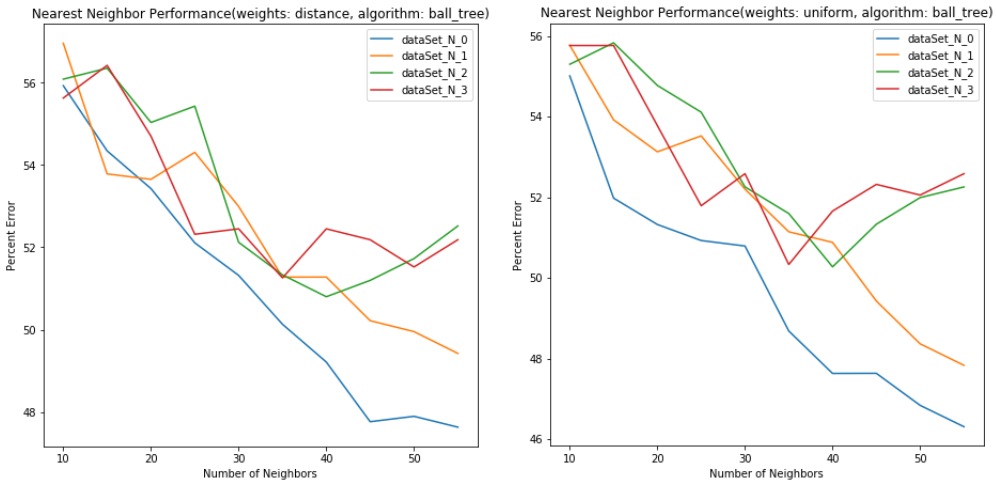
the linear kernel in every case. In the end, we where able to achieve a validation error of 32.5% using a Gaussian kernel SVM on Dataset 1.



In our testing we trained and tested and Gaussian kernel SVM with the optimal hyperparameters found for dataset 0 (namely $C = 1000.0, \gamma = 0.1$), on datasets 1, 2 and 3. Very interesting, we where able to achieve a 27.5% final validation error on dataset 3 when using these parameters. Of note, however, the 5-fold cross-validation error when using these hyperparameters and dataset 3 is a staggering 50.2%. This huge discrepancy is likely an indication that we do not have enough data, or that our training and test sets do not contain a uniform distribution of data points. We where able to train an SVM that does very poorly on 5-fold cross validation, however does extremely well on fresh test data.

3.2 Nearest Neighbour

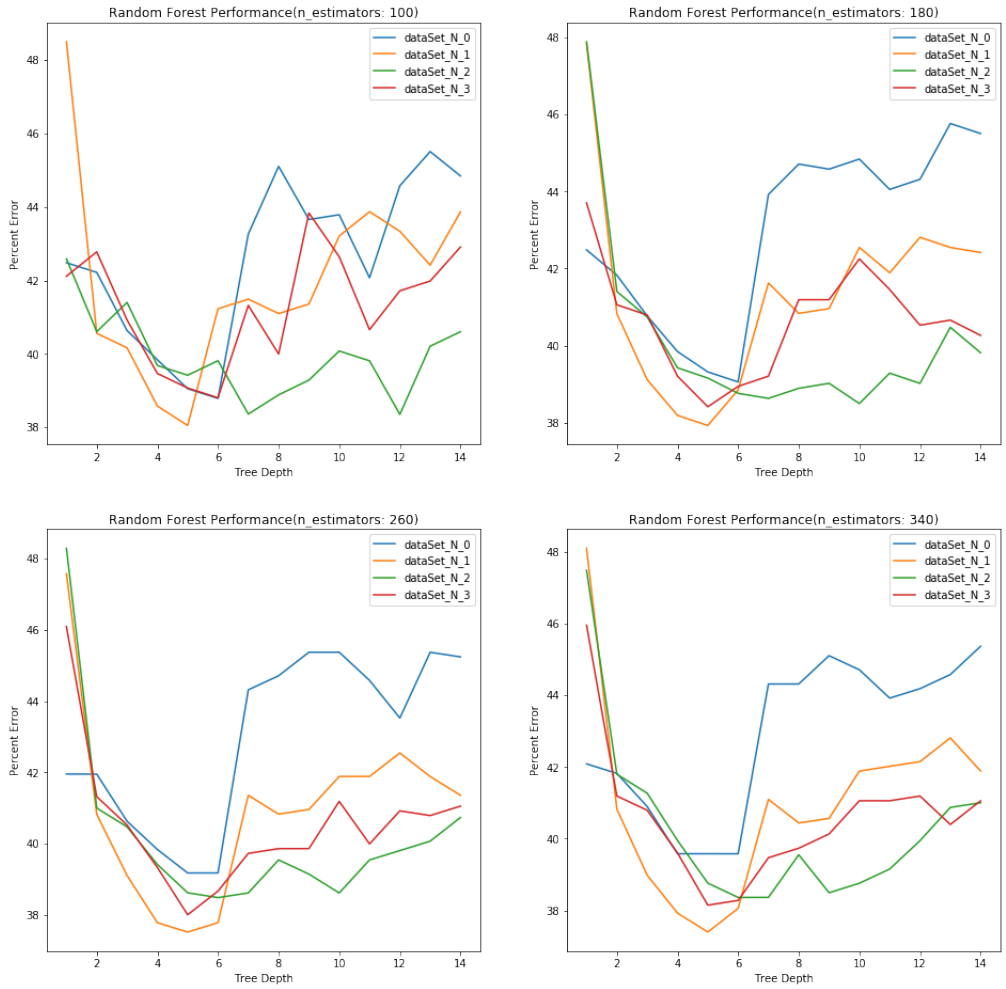
The Nearest Neighbour classifier used for this project is the KNeighborsClassifier from scikit-learn's neighbour classifier library. K was varied between 10 and 60 due to the higher dimensionality of the dataset. Both the uniform and distance weighted neighbour contribution metrics were tested, as well as both the ball tree and KD tree as shown in the figure below. Selecting between Ball Tree and KD Tree had no effect on the results, so KD Tree graphs are omitted below. As clearly shown



by the tick values on the Y-axes of the graphs above the Nearest Neighbour classifier was not at all effective on this dataset. The best accuracy achieved for any parameter values tested was around 54%, far lower than any of method discussed here. Nearest Neighbour is particularly sensitive to high dimensionality, often performing poorly on high dimension data. This is most likely the reason for its poor performance here as evidenced by the fact that it performs best on the N=0 version of the dataset. At N=0 the number of data points is minimized which seems to help the Nearest Neighbour classifier more than missing out on the historical data hurts it. Overall Nearest Neighbour is just not a good fit for our dataset and we would not continue to work on it going forward.

3.3 Random Forest

Random forest classification was done using the RandomForestClassifier from scikit-learn. Tests were conducted using varying forest sizes of 100, 180, 260, and 340 and varying tree depth from 2 to 14. In all cases dataset 1, containing just the previous days weather data, performed the best. The results from the experiments are shown below.



The number of trees in the forest minimally affected the classification error as the different sized forests were within 1% of each other. Had we had a larger dataset the forest size may have become more important attribute to fine tune. The depth of the trees in the forest had the larger impact on the results, where trees of depth 5 or 6 performed the best. Using the validation set the random forest classifiers were able to achieve 38.8%, 37.4% 38.4%, and 38.0% test error on data sets 0 through 3 respectively.

Assuming that properly classifying or classifying 1 higher than the actual index is acceptable for safety reasons, random forests gets a significant performance boost. The random forest classifier is able to achieve 22.5%, 20.0%, 25.0%, and 12.5% error on datasets 0 through 3 respectively.

Conclusion

Of the three models investigated through to experiments above one can be immediately discarded, Nearest Neighbour did not perform at an acceptable level. Random forest achieved accuracy around 61% on all the datasets, improving significantly when taking into account that a off-by-one higher danger warning could be acceptable.

SVM outperformed Random Forest across the board achieving a best validation accuracy of 67.5% on dataset 1 and a validation accuracy of 72.5% on dataset 3 for a model which only achieved around 50% training accuracy. As noted above this may suggest that our dataset is unbalanced or too small. The 72.5% validation accuracy achieved with SVM was our best overall result, and the only one to achieve our goal of 70%.

Recommendations

Further work is needed on the SVM model. It performed best in the experiments we did and offers promise of even better results in the future. We would like to experiment with a larger dataset by including more stations in the area. If the SVM model can achieve a higher test accuracy on that dataset we would also like to experiment with adding more days of history. Models which have a memory component, such as an LSTM should also be tested using the previous day's danger rating as part of their input.

Finally, more analysis should also be done to explore the value of adjacent-to-correct solutions. Consistently predicting a "too-high" risk factor is not the goal; instead, it may be possible that a well-trained model could react to changed conditions more quickly than a forecaster. Because of human error, could a computer model classify the risk factor more consistently than a forecaster? This would indicate that Avalanche Canada's ratings are relatively noisy. To determine whether this was the case, the model would need to be trained with more data than was available during this project.

References

[1] A. P. R. . K. M. Pozdnoukhov, "Applying machine learning methods to avalanche forecasting," Annals of Glaciology, no. 49, pp. 107-113, 2008.