Webcams, Predictions, and Weather

# Problem

Predicting the weather is a difficult task. Giving an accurate, human-readable description of the weather is an even more difficult task. Using given data about the weather, we explored how machine learning techniques could be used to give a human-readable description of this weather (i.e. “sunny”, “cloudy”, “raining”, and so on).

Two different types of data were available to us. There were quantitative measurements taken from a variety of weather instruments, and pictures taken from a fixed location at every hour during daylight hours. In a realistic setting, only one type of data may be available. Our question was this: Would using image data as features to train a machine learning model produce significantly better results than only using the instrument data?

# Data Collection and Cleaning

The Canadian Government provided the monthly weather data from the Vancouver International Airport. It contained the hourly readings from the instruments used and a description for some hours. Kat Kam provided images taken during daylight hours at English Bay. The images were scaled down to 256×192 pixels. The first aspect was removing the unnecessary columns of the weather data. The time and date columns were split up unnecessarily so only the one with both date and time was kept as a Datetime object. This conversion was done so that the date and time could be converted to a timestamp to be used in the analysis.

The data as given contained many unique descriptions for the weather. Looking at the images associated with the descriptions revealed that the descriptions were not well defined. The difference between ‘mostly cloudy’ and ‘cloudy’ was unclear. Similarly, ‘rain showers’ and ‘rain’ images had no discernable differences. Many of these categories were changed to remove the adverb and be simplified to the weather component. The times with no weather description were also dropped. It may have been possible to compute an appropriate label based on the descriptions for other times in the same day but we weren’t confident in its accuracy.

Each image was opened as an RGB image and converted to a 1-dimensional numpy array. The sample of images were then combined, so it could be used in a principal component analysis (PCA). A PCA was done to reduce the image features down to 250. The images were then joined with the weather data based on the time the images were taken and the time the weather was recorded. Joining the data and saving it as an csv file meant that the data could be cleaned once and be used in any analysis afterwards. Memory was saved, and the programs were more efficient.

# Techniques Used

The question we sought to answer was whether the image data, the weather data, or both, provided better prediction for a description of the weather. To test this, three different machine learning models were applied to three different groups of features. The models chosen were a gaussian naïve bayes, a support vector machine, and k-nearest neighbors. The three groups of features were the weather data by itself, the images by themselves, and both together.

To use the images in the analysis, a dimensionality reduction needed to be done. To do this we used a principal component analysis to reduce the number of features to 250. We chose 250 because it gave us a high enough total explained variance ratio of ???. Increasing the number of features resulted in a marginal increase in explained variance. We also chose to use the RGB values of the images instead of a single value to represent a pixel. This increased the memory and time cost of doing the dimensionality reduction but improved the accuracy of the models.

Insert PCA comparison

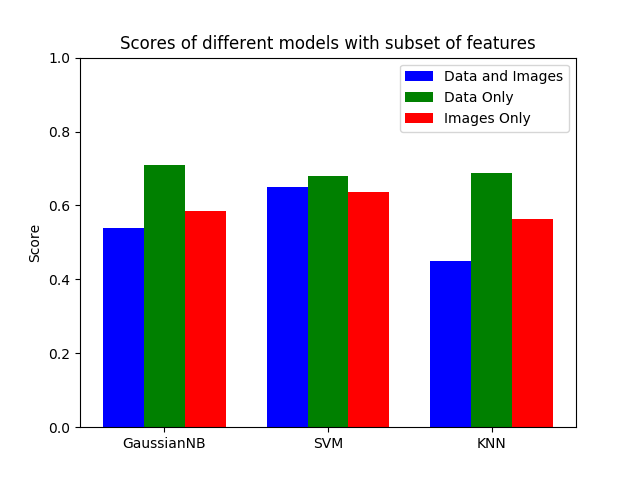
The features provided by the weather instruments were on different units and scales, so we standardized them with a standard scaler. For the support vector machine, a c-value of 10 was chosen with a rbf kernel.

At first only one trial was done to get a score for each model based on the same training data and test data split. Running different splits gave different results as expected resulting in a less accurate true score for a model. We decided to run multiple trials and average the scores to account for the randomness of the train-test split. The time taken was for each model on each group of features was also recorded.

# Conclusions

The results recovered were somewhat surprising. Before the analysis, we hypothesized that the images and the weather data together with a support vector machine model would produce the best results. The reason being that the SVM is best suited for image classification and the high number of features negatively affects the other two models. The results from our analysis revealed that the combination of the weather data and images consistently performed worse than just either one for all models. Our assumption was that adding more features would produce better results but that was not the case.

Another perhaps less surprising result was that the weather data alone was more accurate than the image data alone for all models. To answer our initial question, our analysis revealed that using images alone or in a combination with other weather features did not increase the accuracy of machine learning model’s prediction.



# Limitations

The main limitation we encountered was the dimensionality reduction. It required a significant amount of time and memory to perform the reduction. We could have read images differently or taken a larger number of components, but memory and time restrictions prevented this. We assumed that RGB values would give the best results because the colors in the sky are probably important in differentiating the weather. We would have liked to try out other color modes to see if there is a difference.

Choosing the parameters for the models was another limitation we encountered. We believe the parameters we chose were optimal, but we would have preferred a more exhaustive approach to find them. Tuning the hyper parameters was something we considered, but doing so would increase the complexity of the analysis.

We have not done a statistical test to be confident that the average scores between the models are different. The Gaussian naïve bayes had a larger difference between the groups of features as well as K-nearest neighbors. The support vector machine results were much closer, so we could not confidently say that one group of features gave a better score. To test this, we could have increased the number of trials to above forty and assumed normality to run a t-test. However, the randomness of the train-test split meant that some samples may not be used in any split resulting in biased samples. To avoid this issue, a k-folds cross validation could have been done, but we are unfamiliar with the technique.