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 of these types of data may be available. Our question was this: Would using image data as features to train a machine learning model produce better results than using only the instrument data?

# Data Collection and Cleaning

Weather data taken at Vancouver International Airport was obtained from the Canadian Government. Hourly readings from various instruments were provided, along with a short description of the weather for some hours. Kat Kam provided an image taken every hour during daylight hours from English Bay. The images were scaled down to 256×192 pixels. Our initial clean removed the unnecessary columns of the weather data. The time and date data were unnecessarily split into multiple columns, so only the column with both date and time was kept in the form of Datetime objects. This conversion was needed such that the date and time could be converted to a timestamp to be used in the analysis.

The data as given contained many unique, seemingly human generated descriptions for the weather. Looking at the images associated with various descriptions revealed that the descriptions were not well defined. The difference between “mostly cloudy” and “cloudy” was unclear, for example. Similarly, “rain showers” and “rain” images had no discernable differences. Many of these descriptions were reduced to be less verbose and simplified to just the weather component. Photos without a corresponding weather description were also discarded. It may have been possible to impute an appropriate label based on the weather descriptions for the surrounding hours of the same day but this would have been quite intensive work with unclear effectiveness.

Each RGB image was opened and reshaped from a 256×192×3 array to a 1-dimensional numpy array. A 2-dimensional array was constructed with each row representing a single image, so that the data could be fitted to a principal component analysis (PCA). A PCA was used to transform the data, reducing the number of features per image from 147456 to 250. The image data were then joined with the weather instruments data based on the time the images were taken and the time the instrument data was recorded. With the data now cleaned, it was saved in CSV file format. Manipulating the data into a practical size and then saving it allowed us to clean the data once and then use it in any analysis afterward. This saved memory and computing power, and allowed our analysis greater efficiency.

# Techniques Used

The question we sought to answer was which among the image data, the instrument data, or both, would provide a better prediction for a description of the weather. To test this, three different machine learning models were applied to three groups of features. The models chosen were a Gaussian Naïve Bayes, a Support Vector Machine, and a K-Nearest Neighbors. The three groups of features were the instrument data by itself, the image data by itself, and both together.

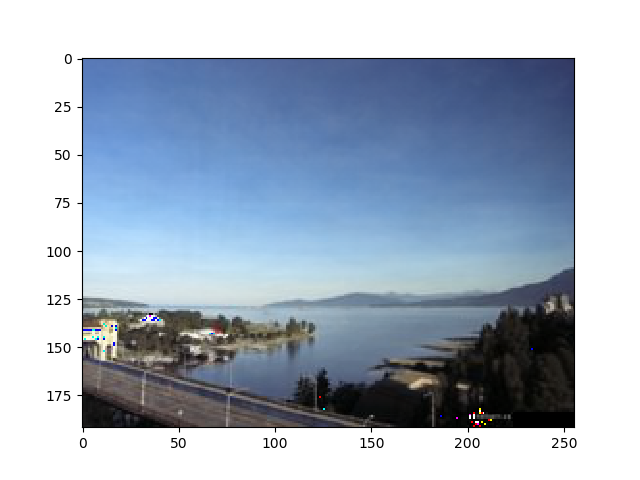
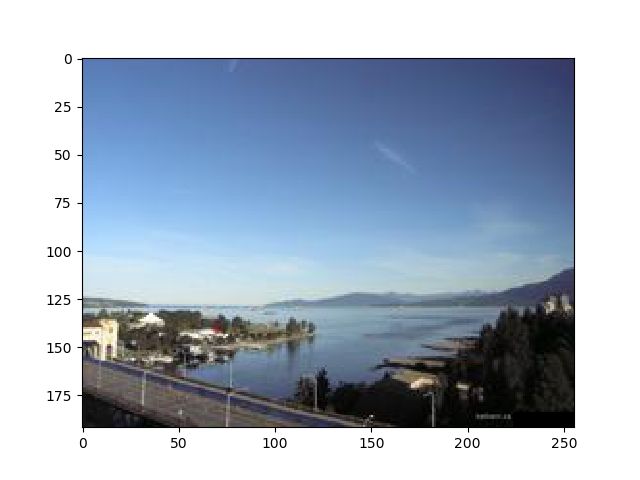
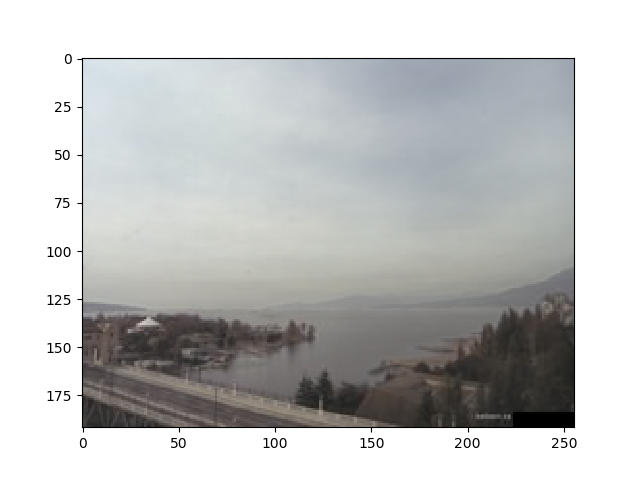
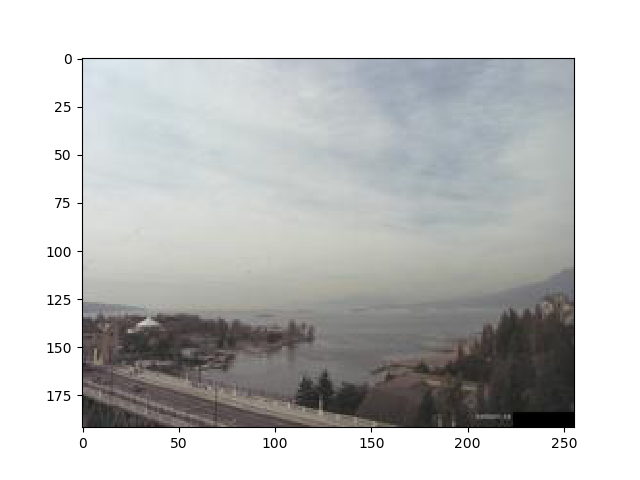
In order to train a machine learning model with the image data in a reasonable amount of time, a dimensionality reduction needed to be done. To do this, we used a principal component analysis and reduced the number of features to 250. Despite reducing the number of features by 99.8% (147456 original features), we chose 250 because it still maintained a very large explained variance ratio of 98%, and increasing the number of features further resulted in a marginal increase in explained variance. Despite the RGB images taking up significantly more space than equivalent greyscale images, we still chose to keep the RGB values because we believed that the colour of the pixels, especially the blue values (for detecting a clear sky), were relevant to the analysis. This increased the memory and time cost of the dimensionality reduction but also improved the accuracy score of the models.

PCA Transformation and Inverse

Original Image with cloudy weather

PCA Transformation and Inverse

Original Image with clear weather



Despite PCA drastically reducing the features of each image from 147456 to 250, a remarkable amount of data is kept intact, as seen in the comparison above. One area where details are lost is the street at the bottom of the image. The main difference between the images is the smoothing out of the sky. Details in the clouds are lost and other areas blend together.

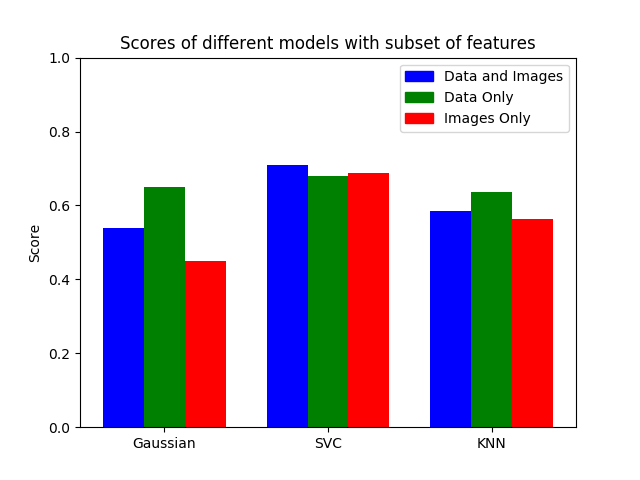
The features provided by the weather instruments were on different units and scales, so they were standardized with a standard scaler. For the Support Vector Machine, a c-value of 10 was chosen with a rbf kernel.

At first only one trial was done to get a score for each model based on the same train and test data split. Running different splits gave varied results as expected, resulting in an unclear true score for each model. We decided to run 10 trials and average the scores to account for the randomness of the train-test split. The time taken for each model on each group of features was also recorded.

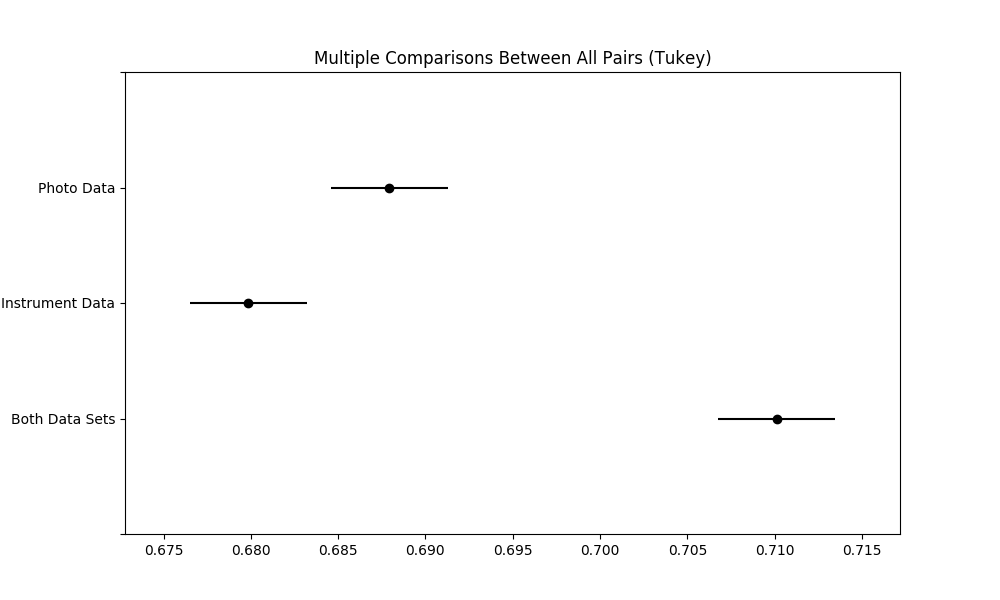
# Conclusions

The results recovered were somewhat surprising. Before the analysis, we hypothesized that the images and the weather data together with a Support Vector Machine (SVM) model would produce the best results. We theorized that the SVM is best suited for image classification and the high number of features would negatively affect the other two models. The results from our analysis revealed that the combination of the weather data and images did perform best, but only for SVM, and the accuracy scores from the other feature sets were quite close behind. Interestingly, our assumption that adding more features would produce better results was completely false in the case of both the GaussianNB and K-Nearest Neighbors models, where the instrument data alone actually scored significantly higher than the combination of the feature sets.

This analysis showed that the SVM model performed better than both other models in all three conditions. But more importantly, the analysis revealed that training an SVM model on both the instrument data and the image data can produce more accurate predictions than training the model on only one type of data. But is it enough of a different to be statistically significant? It certainly did not look like a large difference in the graph we plotted, shown below. Further analysis was needed.



We decided to run additional analysis on only the SVM model. The same three feature sets were compared, but 40 trials were ran for each so that normality could be assumed. An ANOVA test was computed on the data, giving a p-value of 5.1×10-19. With statistical significance confirmed, a posthoc Tukey test was performed. The plot below clearly shows statistical significance between the combined feature set and the individual feature sets. However, despite having statistical significance, it is worth noting that the combined feature set had an accuracy score of only 2.2% higher than the score of the photo feature set by itself. In a real-life scenario, this improvement in accuracy may not be worth the greater memory and computation required.



# Limitations

The main limitation we encountered was dimensionality reduction. PCA required a significant amount of time and memory to perform. Without this limitation, we could have read images in different formats (such as LAB colour space) 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 colour of the sky is likely important in differentiating the weather. We would have liked to try out other color modes for a potential improvement in accuracy.

Choosing the parameters for the models was another limitation we encountered. The parameters we used were manually chosen, and we would have preferred a more exhaustive approach to find them. Using hyper parameters to optimize our models was something we considered, but doing so would increase the complexity of the analysis.

Finally, the randomness of the train-test split meant that some samples may not be used in any split resulting in biased sample sets. To avoid this issue, a k-folds cross validation could have been done, but we were unfamiliar with the technique.

# Project Experience Summary

**Manvir Grewal:**

* Used the sklearn python package to train three different machine learning models on different groups of features to determine the best feature and model combination
* Performed a primary component analysis to reduce the number of features of an image by 99.8% to improve the training time of a support vector machine to under 3 seconds
* Effectively communicated the results of an analysis in a multiple bar plot to show the different scores between models
* Cleaned data in an extract, transform and load process to reduce memory usage and time taken to perform the analysis

**Michael Lam:**

* Used only photographs to train a machine learning model to predict the weather with 69% accuracy, obtaining an even greater score than predictions made with weather instrument data.
* Removed 99.8% of image features while maintaining 98% of relevant data in order to diminish run time and computing power needed for analysis, accomplished via Principle Component Analysis.