Jeff Beyer / Logan Heller / Anthony DePaul

Dr. Mooney

CSC 403

5/10/18

Using Machine Learning to Make Weather Predictions

**Introduction**

The problem domain of this project is classifying the weather description of a given data point based on the climate data at a given point in time. In order to solve this problem, we utilized a decision tree model as well as an SGD classifier, in order to evaluate which model performs better. Our data set is a record of the climate features, taken hourly over the course of several years in a given city.

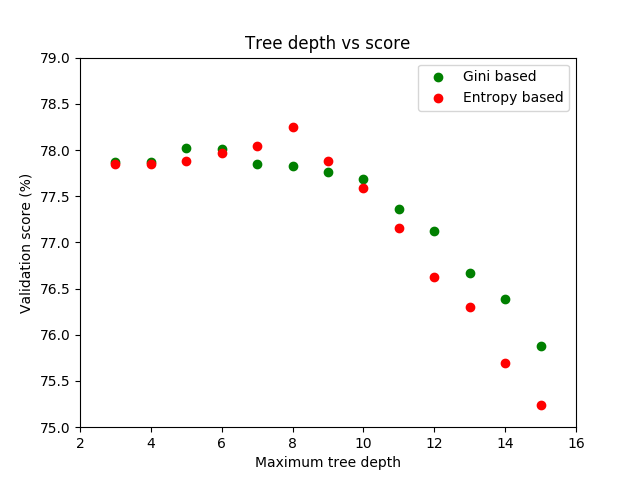
**Data Analysis**

The hourly weather data set contains a report on a number of climate features: Humidity, Pressure, Temperature, Wind Direction, and Wind speed. We combined all of these features from their respective files into a data-frame, so that they can be used to train our models. For each data point, a weather description is also reported. This was imported separately as it is the classification for each data point.

**// Talk about normalizing the wind direction and mapping the weather classifications to integer values to be machine readable**

**Decision tree model**

Decision trees were a natural pick for this problem domain. They fit well into discrete classification problems and are robust to inaccuracies in training data, which can occur on a dataset as large as ours. To accurately train the model, we explored how different Hyper-parameters affected the overall accuracy of classification. Accuracy is a viable performance metric here, as there are eleven possible classifications. If the model is placing an example in the correct category out of eleven a reasonable portion of the time, then it is performing well. We compared the use of Gini score vs Entropy to split nodes and how the maximum depth of the tree affected the overall validation score. To generate the score, we split our data set into 2/3 training data and 1/3 validation data. We then iteratively created a decision tree model of either scoring type and specified a depth for the nodes of the tree. The following is a summary of the results:



The results are slightly different with each execution of the training, so to gather results the model was trained 30 times at each point and the score was averaged. A few conclusions can be drawn from this run. Primarily, the maximum depth of the tree creates an optimal result between 6-9 levels. Beyond a depth of 10, the model begins over-fitting and the performance continues to decrease. Also, entropy based classification scored the highest possible value at a maximum depth of 8. However, there is generally very minute differences between Gini models and Entropy models (Typically less than .5 %) In order to compare the results with the regression model, we considered the overall accuracy of our model as 78%.

**SGD model**

The other model trained was the stochastic gradient descent model. Due to the model being much simpler than a decision tree, the results with the same data was less successful. To alleviate these issues, the data was manipulated to best suit the model. Originally there were eleven unique classes for the model to identify: clear, few/overcast clouds, broken/scattered clouds, mist/fog, light rain, moderate rain, heavy rain, thunderstorm, freezing rain/sleet, snow, and haze/smoke/dust. The model trained for this had an accuracy of 27.15% and f1 score of 0.2752. The f1 scores were calculated with the micro average method. Next, some classes were removed or combined to try to better suit the model. Stratified shuffle split was used to ensure that the weather description labels of the data set were distributed appropriately for the test and training sets.

|  |  |
| --- | --- |
| Class combinations | Test set f1 score |
| 11 classes | 0.2752 |
| 4 classes  (clear, rain, snow, clouds) | 0.4385 |
| 3 classes #1  (clear, precipitation, clouds) | 0.4355 |
| 3 classes #2  (no precipitation, rain, snow) | 0.7365 |
| 2 classes  (no precipitation, precipitation) | 0.7336 |

From this graph and table it can be determined that the model performs better when the classes are reduced to be more general. Reducing down to four classes shows that removing the specific classes improves the model, but still is not as high as it could be. The next attempt was to combine the rain and snow classes, but performance gained from that was negligible. The attempt after that was to keep the separate rain and snow classes, but remove the clouds class. This increased the f1score by 69.12%. Finally, combining the rain and snow classes was attempted again, but, once again, the results were negligible. In conclusion, the model trained to classify no precipitation, rain, or snow would be the best to use as it has the best f1 score and is the most informative with three different classes to classify. Compared to the decision tree model, this one performed worse and only performed well once the data was simplified.