Aaron Sun, Brendan Lee, Neel Bagora

CS 373, Professor Honorio

Team 15 Preliminary Report

Problem

Global warming is one of the many challenges that we are facing and will continue to face in the coming century. Seeing that weather has become increasingly unpredictable, we are interested in analyzing previous weather data to determine if weather can be predicted. Specifically, we are interested in determining if a given day's conditions can be used to determine if it will rain the following day. We aim to develop a Naïve Bayes Classification and a Decision Tree Classification model to determine if we can predict whether it will rain on the following day using only current day weather conditions in the city of Uluru, Australia.

Dataset

The dataset chosen is called "Rain in Australia" which includes about 10 years of weather data from various cities in Australia, specifically, the dataset has 23 features that are used to describe the weather conditions of each day. The features provided are *Date*, *Location, Minimum Temperature (MinTemp)*, *Maximum Temperature (MaxTemp)*, *Rainfall, Evaporation, Sunshine, Wind Gust Direction (WindGustDir), Wind Gust Speed (WindGustSpeed)*, *Wind Direction at 9 am (WindDir9am), Wind Direction at 3 pm (WindDir3pm), Wind Speed at 9 am (WindSpeed9am), Wind Speed at 3 pm (WindSpeed3pm), Humidity at 9 am (Humidity9am), Humidity at 3 pm (Humidity3pm), Pressure at 9 am (Pressure9am), Pressure at 3 pm (Pressure3pm), Cloud Cover at 9 am (Cloud9am), Cloud Cover at 3 pm (Cloud3pm), Temperature at 9 am (Temp9am), Temperature at 3 pm (Temp3pm), Rain on day of (RainToday), and Rain the day after (RainTomorrow).* The target variable that will be predicted in this data is "*RainTomorrow*", where "*Yes*" indicates that the amount of rain on the next day is greater than 1 mm whereas "*No*" indicates that rain on the next day is less than 1 mm.

The data in its raw form required preprocessing to be utilized efficiently by our model; we removed the features *Date*, *Location*, *Evaporation*, and *Sunshine* as these features did not add any significant value to the data. In addition, we mapped the categorical data to corresponding numerical data. For instance, Yes and No were mapped to be 1 and 0, respectively. In addition, we had categorical data describing the wind direction, such as West, East, North, South, etcetera. Using a numerical scale calculated by sklearn.preprocessing.LabelEncoder, these labels were mapped to numerical labels that were appropriate for their respective columns. Another issue we saw was our data contained several NaN (missing) cells, which required additional processing. To solve this, each NaN value in RainToday and RainTomorrow was filled with 0s seeing that these features only contained 1s and 0s (corresponding to Yes and No). In the remaining features, we filled NaN values with the mean of the column to ensure that we do not alter the data significantly by adding 0s. We also converted the RainToday and RainTomorrow columns to be integer columns for simplicity and compatibility with the models.

Finally, due to the large size of the dataset and the larger number of missing values in the earlier years of data, we narrowed down our data by choosing the city of Uluru and further selecting 1000 of the most recent samples in Uluru.

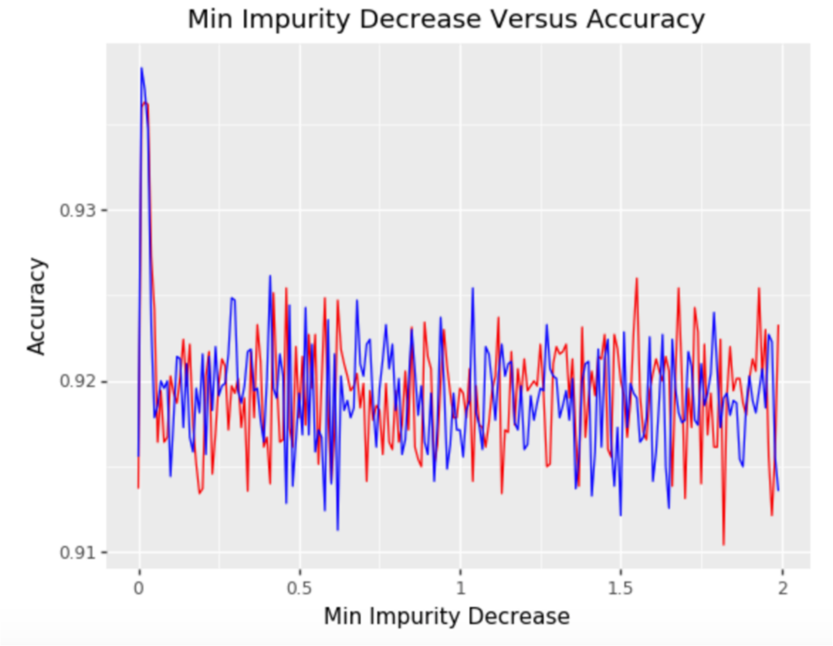
Experiment Setup

For our Decision Tree Classifier model, we are implementing the scikit-learn Decision Tree Classification (sklearn.tree.DecisionTreeClassifier). In the experiment, our tree will employ the use of the Gini impurity to measure the quality of each split. With the amount of data that the model will have to analyze, there is a tendency to overfit the decision tree so using the min_impurity_decrease hyperparameter, we aim to reduce the likelihood of overfitting the tree. In order to obtain an ideal min_impurity_decrease value, we will run several cross-validation iterations on the algorithm to determine what value yields the highest accuracy and to develop a thorough analysis on the model. Specifically, we will test the values (0, 2) with a step of 0.01 as this precision will be enough to identify outstanding values. Values beyond 2 would be considered redundant as they would be greater than the Gini value of the decision tree without splits.

In addition to Decision Tree Classification, we are also utilizing a Bernouilli Naïve Bayes Classifier, using the implementation from scikit-learn (sklearn.naive_bayes.BernouilliNB). Our experiments with this algorithm will involve tuning the alpha additive smoothing hyperparameter to minimize the occurrences of zero probability problems that may arise. We plan to implement the hyperparameter to find the best alpha value. Like the decision tree, we will also run several cross-validation iterations on the algorithm to determine an optimal alpha value for this algorithm and for developing a thorough model analysis.

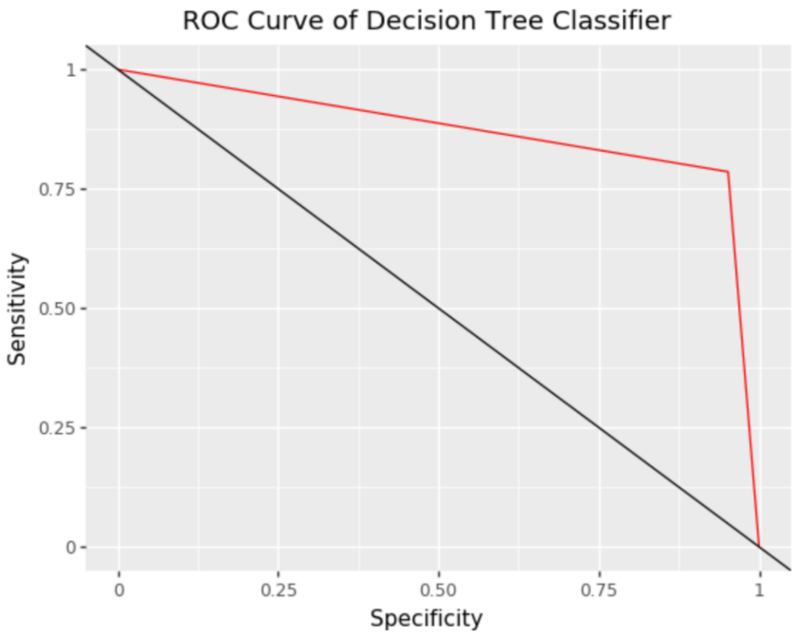
For our cross-validation technique, we will be using bootstrapping cross validation where we will run 35 iterations of training, validation, and testing. We will first randomly allocate each sample into three subsets where 60% of the samples will be used for training (D_{train}), 20% will be used for validation (D_{val}), and 20% will be used for testing ($D_{testing}$). In order to avoid training our model with the predictor variable, the "RainTomorrow" (y) column will be removed from D and kept for analysis. After training the models with X_{train} and y_{train}, the validation subset is supplied, yielding the prediction ($\hat{y}_{val)}$ from the set, X_{val}. This prediction is then compared to y_{val} and analyzed to generate metrics quantifying how well the models have been trained. Specifically, this will help us test many different hyperparameters and to identify the optimal values for the models. Seeing that we are using bootstrapping cross validation for identifying the hyperparameter values, we will use the average of validation accuracy score and the average of the testing accuracy for each hyperparameter to quantify the importance of the value.

Decision Tree Classifier Experiment Results



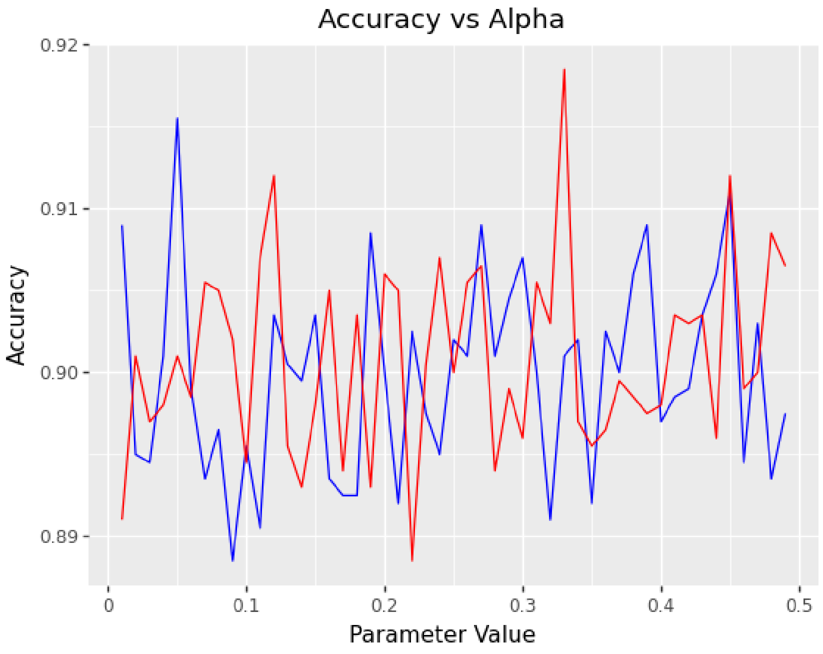
Blue* – *Testing Accuracy,* *Red* - *Validation Accuracy

After performing bootstrapping with cross validation and plotting the results, we can see that our model achieved its highest testing and validation accuracy at a very low "min_impurity_decrease" value, specifically, at a value of ~0.02. The testing and validation accuracy at that hyperparameter value is around ~0.935, which is considerably better compared to the other values, as shown in the plot.



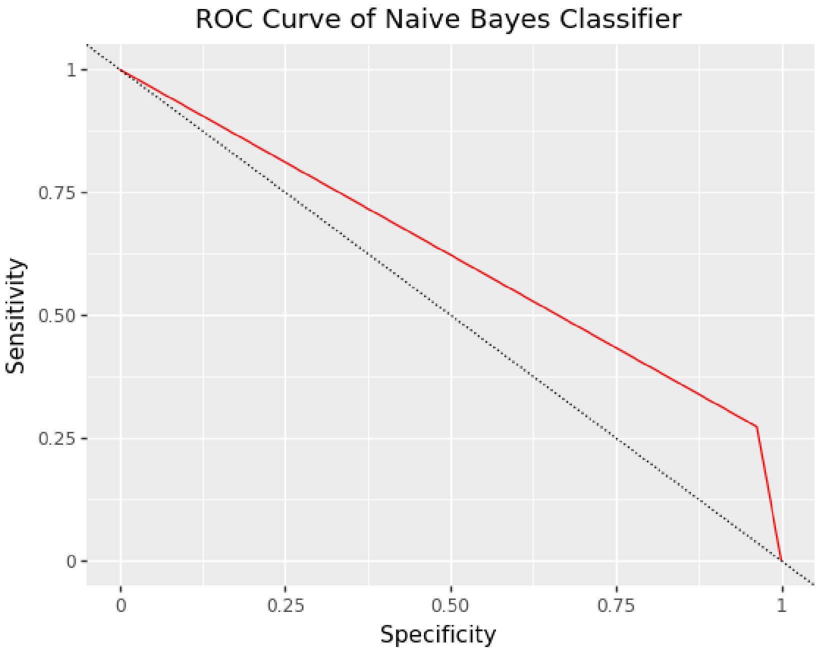
Our ROC Curve for the Decision Tree Classifier is relatively close to the top right corner of the plot - (1, 1), indicating that our model has a significant level of performance in terms of the tradeoff between specificity and sensitivity as well as accuracy.

Naïve Bayes Classifier Experiment Results



Blue* – *Testing Accuracy, Red - Validation Accuracy

After performing the cross validation, we plotted the results of the model to compare the accuracy of each test with the different test parameters. The graph shows that after multiple bootstrap iterations, our model achieves the highest overall accuracy at the value ~0.45. We found that with alpha values greater than 0.5 did not return any significant results. In addition, we can see that this new value yields greater accuracies than the initial value of 0.1.



As the ROC curve shows, Naïve Bayes Classifier is not nearly as well performing as the Decision Tree Classifier, in terms of the tradeoff between specificity and sensitivity. In addition, seeing that the curve is closer to the 45-degree baseline, that indicates that the naïve bayes classifier is not nearly as accurate as the decision tree classifier, which had an ROC curve closer to the corner.

Conclusion

Based on the performance of the Decision Tree Classifier and Naïve Bayes Classifier, we can conclude with the evidence presented that it is possible to predict, whether it will rain on the next day using only present-day weather conditions, however, the Decision Tree Classifier was able to more accurately predict whether it will rain with the same dataset. In addition, the Decision Tree Classifier and Naïve Bayes Classifier both yield accurate results, with the Decision Tree Classifier performing significantly better than the Naïve Bayes Classifier, as shown by the respective ROC curves. Finally, seeing that the Decision Tree Classifier and Naïve Bayes Classifier both yield accurate results with the city of Uluru, Australia, we think that they can be universally applied to many cities and can further improve weather prediction in an increasingly unpredictable climate.