Boston Weather Analysis

CS 699 Final Project

Ayan Ashkenov
Sarve Khorramshahi

Boston University
Spring 2023

Contents

Data Mining Goal

The purpose of this data mining exploration is to analyze weather data to help classify a snow day versus a day without snowfall based on various meteorological factors.

Dataset Description

This dataset contains meteorological data from the National Centers for Environmental Information (National Oceanic and Atmospheric Administration) collected from a weather radar placed at Boston Logan International Airport. We were able to select a custom date range of data which we limited to January 2019 to December 2022, and obtain daily meteorological data for these past three years. This ensured that we crafted our algorithms based on a dataset of recent and relevant values and allowed us to make inferences on when and how much snow we should expect given current climate and environmental conditions which are relevant in recent terms but may not be as relevant if we were to consider data from the more remote past. We believe limiting our dataset to the most recent years will provide the least margin for error as it will take into consideration the current state of the environment and effects of global warming and pollution.

The following provides a detailed description of all the data attributes which we mined from our original data source, the National Centers for Environmental Information.

Data Attributes:

- STATION: Alphanumeric station code, unique identifier
- NAME: Name of the station where the meters are located
- LATITUDE: Numeric latitude where the station meter is located
- LONGITUDE: Numeric longitude where the station meter is located
- ELEVATION: Numeric elevation above mean sea level in tenths of a meter
- DATE: Date of recording in the format yyyy-mm-dd
- AWND: Average daily wind speed (miles per hour)
- AWND_ATTRIBUTES: Wind attribute measurement, quality, and source flags (below)
- PGTM: Peak gust time (hours and minutes)
- PGTM_ATTRIBUTES: gust attribute measurement, quality, and source flags (below)
- PRCP: Precipitation (inches)
- PRCP_ATTRIBUTES: precipitation attribute measurement, quality, and source flags (below)
- SNOW: Snowfall (inches)
- SNOW_ATTRIBUTES: snow attribute measurement, quality, and source flags (below)

- TAVG: Average temperature (Farenheit)
- TAVG_ATTRIBUTES: average temp attribute measurement, quality, and source flags (below)
- TMAX: Maximum temperature (Farenheit)
- TMAX_ATTRIBUTES: max temp attribute measurement, quality, and source flags (below)
- TMIN: Minimum temperature (Farenheit)
- TMIN_ATTRIBUTES: min temp attribute measurement, quality, and source flags (below)
- WDF2: Direction of average 2 minute wind (degrees)
- WDF2_ATTRIBUTES: direction of avg 2 minute wind attribute measurement, quality, and source flags (below)
- WDF5: Direction of average 5 second wind (degrees)
- WDF5_ATTRIBUTES: direction of avg 5 second wind attribute measurement, quality, and source flags (below)
- WSF2: Fastest 2 minute wind speed (miles per hour)
- WSF2_ATTRIBUTES: fastest 2 minute wind speed attribute measurement, quality, and source flags (below)
- WSF5: Fastest 5 second wind speed (miles per hour)
- WSF5_ATTRIBUTES: Fastest 5 second wind speed attribute measurement, quality, and source flags (below)
- WT01: Weather type where there is fog, ice fog, or freezing fog
- WT01_ATTRIBUTES: Fog weather type attribute measurement, quality, and source flags (below)
- WT02: Weather type where there is heavy fog or heavy freezing fog
- WT02_ATTRIBUTES: Heavy fog weather type attribute measurement, quality, and source flags (below)
- WT03: Weather type where there is thunder
- WT03_ATTRIBUTES: Thunder weather type attribute measurement, quality, and source flags (below)
- WT04: Weather type where there are ice pellets, sleet, snow pellets, or small hail
- WT04_ATTRIBUTES: Ice pellet weather type attribute measurement, quality, and source flags (below)
- WT05: Weather type where there is hail
- WT05_ATTRIBUTES: Hail weather type attribute measurement, quality, and source flags (below)
- WT06: Weather type where there is glaze or rime
- WT06_ATTRIBUTES: Glaze or rime weather type attribute measurement, quality, and source flags (below)
- WT08: Weather type where there is smoke or haze

- WT08_ATTRIBUTES: Smoke or haze weather type attribute measurement, quality, and source flags (below)
- WT09: Weather type where there is blowing snow
- WT09_ATTRIBUTES: Blowing snow weather type attribute measurement, quality, and source flags (below)

Please see below for a key to the flags which were present in the original data source.

Measurement Flag

Blank = no measurement information applicable

A = value in precipitation or snow is a multi-day total, accumulated since last measurement

B = precipitation total formed from two twelve-hour totals

D = precipitation total formed from four six-hour totals

H = represents highest or lowest hourly temperature (TMAX or TMIN) or average of hourly values (TAVG)

K = converted from knots

L = temperature appears to be lagged with respect to reported hour of observation

O = converted from oktas

P = identified as "missing presumed zero" in DSI 3200 and 3206

T = trace of precipitation, snowfall, or snow depth

W = converted from 16-point WBAN code (for wind direction)

Quality Flag

Blank = did not fail any quality assurance check

D = failed duplicate check

G = failed gap check

I = failed internal consistency check

K = failed streak/frequent-value check

L = failed check on length of multi day period

M = failed mega-consistency check

N = failed naught check

O = failed climatological outlier check

R = failed lagged range check

S = failed spatial consistency check

T = failed temporal consistency check

W = temperature too warm for snow

X = failed bounds check

Z = flagged as a result of an official Datzilla investigation

Source Flag

Blank = No source (i.e., data value missing)

W = WBAN/ASOS Summary of the Day from NCDC's Integrated Surface Data (ISD).

Data Mining Tools

As our data was obtained from an online repository, we were able to specify the range of dates which we wanted to include and download the dataset directly from the National Centers for Environmental Information website. We then proceeded to load the .csv file to R Studio to continue with our preprocessing and analytic tasks.

Classification Algorithms

The following provides a brief description of the five classification algorithms we chose to train and test our dataset.

Naive Bayes (‘nb’):

The Naive Bayes method is a supervised learning algorithm that relies on treating all attributes independently to classify the categorical class attribute.

Boosted Classification Tree (‘ada’):

The Adaptive Boosting algorithm is a supervised boosting method which allows users to use weaker classifiers iteratively to yield a stronger final classifying attribute by adjusting weights throughout each iteration of the algorithm to result in a final optimized weighting.

Recursive Partitioning and Regression Trees (‘rpart’):

The Recursive Partitioning and Regression Trees classification algorithm, otherwise known as ‘rpart’, is a method of supervised learning classification which builds a recursive decision tree to predict class attributes.

Generalized Linear Model (‘glm’):

The Generalized Linear Model is a classification algorithm that builds off of the standard linear regression model by considering the possible presence of non-standard variables and distributions of error.

Random Forest (‘rf’):

The Random Forest classification method approaches classification problems by combining multiple decision trees to optimize prediction accuracy of the class attribute. This classification method is often used in overfitting scenarios like we have in this scenario.

Attribute Selection Methods

The following provides a brief description of the five attribute selection methods we applied to our source dataset.

Information Gain (information.gain()):

The Information Gain is a metric that is used in attribution selection which helps assign a value or weight of importance to how much an attribute contributes to the outcome of the target variable. These weights are then used to determine which attributes have the greatest contribution.

Boruta (Boruta()):

The Boruta algorithm is used for attribute selection and is a wrapper algorithm that fits around the Random Forest classifier. It achieves optimal feature selection by weighing the importance of attributes during different stages of classification to see which attributes have the highest significance at various intervals.

Genetic Algorithm:

The Genetic Algorithm method is a population based approach to optimizing for attribute selection. The algorithm can handle large quantities of attributes by using fitness functions and crossover techniques to extract the most important features from a set of attributes.

Simulated Annealing:

The Simulated Annealing approach to attribute selection parses a number of subsets of the data to determine which attributes retain importance across various subsections of the whole data set. This is accomplished by implementing various changes to the data set and analyzing which features remain significant in determining the target variable.

Recursive Feature Elimination:

Recursive Feature Elimination is an iterative approach to feature selection which accomplishes attribute elimination through fitting a model to the data and iteratively removing the weakest attribute in terms of its significance to determine the target variable.

Selected Attributes

The following is a list of the attributes selected by the various methods mentioned above.

Information Gain (information.gain()):

The information gain attribute selection method narrowed down the attributes to "PRCP", "PRCP_ATTRIBUTES", "SNOW_ATTRIBUTES", "TAVG", "TMAX", "TMIN", "WT01", "weather_condition".

Boruta (Boruta()):

The Boruta attribute selection method narrowed down the attributes to "PRCP", "TMAX", "WT01", "TAVG", "TMIN", "SNOW_ATTRIBUTES", and "weather_condition".

Genetic Algorithm:

The genetic algorithm attribute selection method narrowed down the attributes to "PRCP_ATTRIBUTES", "SNOW_ATTRIBUTES", "TAVG", "WT01", "WT03", "weather_condition".

Simulated Annealing:

The simulated annealing attribute selection method narrowed down the attributes to "SNOW_ATTRIBUTES", "TMAX", "WT08", "PRCP", "TAVG", and "weather_condition".

Recursive Feature Elimination:

The recursive feature elimination attribute selection method narrowed down the attributes to "PRCP","WT01","TMAX","TMIN","TAVG", and "weather_condition".

Data Mining Process

Data Preprocessing

The process of data sourcing, selection, and preprocessing began on the National Centers for Environmental Information (National Oceanic and Atmospheric Administration) website, where we were able to source our raw data and download it as a '.csv' file.

Then began the processing of the data via R which we accomplished by first importing the file and running basic commands to get a sense of the data, missing values, and the quantity of columns. We narrowed down the attributes which could be used for classification, namely 'AWND', 'PRCP', 'PRCP_ATTRIBUTES', 'SNOW_ATTRIBUTES', 'TAVG', 'TMAX', 'TMAX_ATTRIBUTES', 'TMIN', 'WDF2', 'WDF5', 'WSF2', 'WSF5', 'WT01', 'WT02', 'WT03', 'WT04', 'WT05', 'WT06', 'WT08', 'WT09', and 'weather_condition', and created a subset of the data with these feature attributes.

We converted the target variable column 'weather_condition' to a factor, and also created a mapping to convert all categorical variables to a numeric equivalent. Next, we replaced all NA/NULL values with 0 to be able to account for those values in our classifications and scaled the numeric column to an average of zero and a standard deviation of one to standardize. The final step was to split the data into a training set and a test set which we decided on a 70/30 split for. This concluded our data preprocessing.

Data Mining Procedure

Before beginning the construction of our classification algorithms, we focused on building out our five attribute selection methods: Information Gain, Boruta, Genetic Algorithm, Simulated Annealing, and Recursive Feature Elimination. The following briefly describes how we achieved each of the attribute selection methods.

Information Gain

After installing the FSelector package, we used the information.gain() method to calculate the information gain of each attribute and stored those values as weights. We then created a subset of those non-zero weights to get a list of the optimal attributes.

Boruta

After installing the Boruta package, we used the Boruta algorithm for the feature selection and stored the selected weights in the boruta_output element. We addressed the tentative attributes and excluded all unnecessary attributes via 'imps' to get the final list of selected attributes.

Genetic Algorithm

We started by writing the ga_ctrl object which essentially fits a random forest algorithm as a fitness function for a Genetic Algorithm, and then performed 3 fold repeated cross validation. We then run the genetic algorithm feature selection on the data set and extract the optimal variables.

Simulated Annealing

We started by writing the sa_ctrl object which essentially fits a random forest algorithm as a fitness function for a Genetic Algorithm, and then performed 3 fold repeated cross validation, but set 'improve' to 5 which stipulates that there will be 5 iterations without improvement before a reset. We then printed the optimal set of attributes to use for our simulated annealing training and test sets.

Recursive Feature Elimination

For the RFE, we began by creating a subset of integers to represent the number of features. We then used 'rfFuncs' to fit the random forest to the attribute selector and performed the same cross validation. Using this we were able to extract a list of the optimal attributes to use in the RFE training and test sets.

Following creating our attribute selection algorithms and obtaining our 10 total new training and test sets, we set the control object, 'ctrl', for our training sets to define our cross validation procedure. Based on examples followed throughout the course, we opted for 10 fold repeated cross validation, repeated 5 times. We also chose in the case of our project to implement oversampling. We made this determination because our dataset was unbalanced in terms of 'Snow' days to 'No Snow' days, which is expected as during a calendar year in Boston, there are significantly less snow days than there are days without snow. To address this imbalance, we added the sample 'up' method to our control which essentially arbitrarily duplicates various lines of data with small differences to create a more balanced sample space.

All that remained at this point was to write the code for out classification algorithms. The process is nearly identical for all five of the datasets resulting from the attribute selection methods, therefore we will only describe it in detail for Naive Bayes information gain example.

Naive Bayes

The first step was to reset the grid for the Naive Bayes classifier, we did this using the expand.grid method:

expand.grid(usekernel = c(TRUE), fL = 0:5, adjust = seq(0, 5, by = 1))

We then created the model with the following code segment, making sure to use our optimized attributes from the information gain training set, setting the method to ‘nb’, scaling and centering, and accounting for our control and optimized naive bayesian grid.

```
ig_nb_model <- train(weather_condition ~ .,
                     data = ig_train,
                     method = "nb",
                     preProcess = c("scale", "center"),
                     trControl = ctrl,
                     tuneGrid = nb_grid)
```

After which, we were able to run a simple predict() on the model with the corresponding information gain test set, and create the resulting confusion matrix.

This process was repeated 25 times total, five times each with each classifier for each attribute selection method.

Data Mining Result

After conducting all of our classifications, we came to the conclusion that the Random Forest classifier on the data set optimized to the attributes selected via the Information Gain method had the highest accuracy in classifying a 'Snow' day versus a 'No Show' day. In this section, we will present our findings and walk through how we came to his conclusion, starting with the confusion matrices, and other relevant metrics, for all 25 of our classifications, including visuals and descriptions.

(Naive Bayes) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	301	0
Snow	121	17
	Accuracy	0.7244
	Sensitivity	0.7133
	Specificity	1.0000
	F1	0.8326
	MCC	0.2964
	AUC	0.8566
	Precision	1.0000
	Recall	0.7133

(RPart) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	422	17
Snow	0	0
	Accuracy	0.9613
	Sensitivity	1.0000
	Specificity	0.0000
	F1	0.9803
	MCC	0
	AUC	0.5
	Precision	0.9613
	Recall	1.0000

(AdaBoost) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	417	4
Snow	5	13
	Accuracy	0.9795
	Sensitivity	0.9882
	Specificity	0.7647
	F1	0.9893
	MCC	0.7325
	AUC	0.8764
	Precision	0.9905
	Recall	0.9882

(GLM) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	396	0
Snow	26	17
	Accuracy	0.9408
	Sensitivity	0.9384
	Specificity	1.0000
	F1	0.9682
	MCC	0.6090902
	AUC	0.9692
	Precision	1.0000
	Recall	0.9384

(Rand Forest) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	416	3
Snow	6	14
	Accuracy	0.9795
	Sensitivity	0.9858
	Specificity	0.8235
	F1	0.9893
	MCC	0.7488147
	AUC	0.9047
	Precision	0.9928
	Recall	0.9858

(Naive Bayes) - Boruta	Reference	
Prediction	No Snow	Snow
No Snow	295	0
Snow	127	17
	Accuracy	0.7107
	Sensitivity	0.6991
	Specificity	1.0000
	F1	0.8229
	MCC	0.2872751
	AUC	0.8495
	Precision	1.0000
	Recall	0.6991

(AdaBoost) - Boruta	Reference	
Prediction	No Snow	Snow
No Snow	412	5
Snow	10	12
	Accuracy	0.9658
	Sensitivity	0.9763
	Specificity	0.7059
	F1	0.9821
	MCC	0.6032591
	AUC	0.8411
	Precision	0.9880
	Recall	0.9763

(RPart) - Boruta	Reference	
Prediction	No Snow	Snow
No Snow	422	17
Snow	0	0
	Accuracy	0.9613
	Sensitivity	1.0000
	Specificity	0.0000
	F1	0.9803
	MCC	0
	AUC	0.5
	Precision	0.9613
	Recall	1.0000

(GLM) - Boruta	Reference	
Prediction	No Snow	Snow
No Snow	385	0
Snow	37	17
	Accuracy	0.9157
	Sensitivity	0.9123
	Specificity	1.0000
	F1	0.9542
	MCC	0.5359222
	AUC	0.9562
	Precision	1.0000
	Recall	0.9123

(Rand Forest) - Boruta	Reference	
Prediction	No Snow	Snow
No Snow	413	3
Snow	9	14
	Accuracy	0.9727
	Sensitivity	0.9787
	Specificity	0.8235
	F1	0.985
	MCC	0.6946308
	AUC	0.9011
	Precision	0.9928
	Recall	0.9787

(Naive Bayes) - GA	Reference	
Prediction	No Snow	Snow
No Snow	7	0
Snow	415	17
	Accuracy	0.0547
	Sensitivity	0.01659
	Specificity	1.00000
	F1	0.03263
	MCC	0.02554909
	AUC	0.5083
	Precision	1.00000
	Recall	0.01659

(AdaBoost) - GA	Reference	
Prediction	No Snow	Snow
No Snow	360	1
Snow	62	16
	Accuracy	0.8565
	Sensitivity	0.8531
	Specificity	0.9412
	F1	0.9195
	MCC	0.4009044
	AUC	0.8971
	Precision	0.9972
	Recall	0.8531

(RPart) - GA	Reference	
Prediction	No Snow	Snow
No Snow	422	17
Snow	0	0
	Accuracy	0.9613
	Sensitivity	1.0000
	Specificity	0.0000
	F1	0.9803
	MCC	0
	AUC	0.5
	Precision	0.9613
	Recall	1.0000

(GLM) - GA	Reference	
Prediction	No Snow	Snow
No Snow	390	0
Snow	32	17
	Accuracy	0.9271
	Sensitivity	0.9242
	Specificity	1.0000
	F1	0.9606
	MCC	0.5662425
	AUC	0.9621
	Precision	1.0000
	Recall	0.9242

(Rand Forest) - GA	Reference	
Prediction	No Snow	Snow
No Snow	400	2
Snow	22	15
	Accuracy	0.9453
	Sensitivity	0.9479
	Specificity	0.8824
	F1	0.9709
	MCC	0.5765807
	AUC	0.9151
	Precision	0.9950
	Recall	0.9479

(Naive Bayes) - SA	Reference	
Prediction	No Snow	Snow
No Snow	287	0
Snow	135	17
	Accuracy	0.6925
	Sensitivity	0.6801
	Specificity	1.0000
	F1	0.8096
	MCC	0.2757956
	AUC	0.84
	Precision	1.0000
	Recall	0.6801

(AdaBoost) - SA	Reference	
Prediction	No Snow	Snow
No Snow	412	3
Snow	10	14
	Accuracy	0.9704
	Sensitivity	0.9763
	Specificity	0.8235
	F1	0.9845
	MCC	0.6788131
	AUC	0.8999
	Precision	0.9928
	Recall	0.9763

(Rand Forest) - SA	Reference	
Prediction	No Snow	Snow
No Snow	417	4
Snow	5	13
	Accuracy	0.9795
	Sensitivity	0.9882
	Specificity	0.7647
	F1	0.9893
	MCC	0.7325144
	AUC	0.8764
	Precision	0.9905
	Recall	0.9882

(RParts) - SA	Reference	
Prediction	No Snow	Snow
No Snow	422	17
Snow	0	0
	Accuracy	0.9613
	Sensitivity	1.0000
	Specificity	0.0000
	F1	0.9803
	MCC	0
	AUC	0.5
	Precision	0.9613
	Recall	1.0000

(Naive Bayes) - RFE	Reference	
Prediction	No Snow	Snow
No Snow	360	0
Snow	62	17
	Accuracy	0.8588
	Sensitivity	0.8581
	Specificity	1.0000
	F1	0.9207
	MCC	0.4284557
	AUC	0.9265
	Precision	1.0000
	Recall	0.8531

(GLM) - SA	Reference	
Prediction	No Snow	Snow
No Snow	368	0
Snow	54	17
	Accuracy	0.877
	Sensitivity	0.8720
	Specificity	1.0000
	F1	0.9316
	MCC	0.456944
	AUC	0.936
	Precision	1.0000
	Recall	0.8720

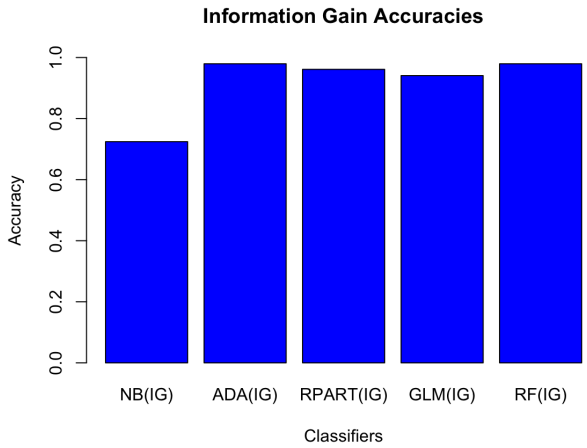
(AdaBoost) - RFE	Reference	
Prediction	No Snow	Snow
No Snow	409	4
Snow	13	13
	Accuracy	0.9613
	Sensitivity	0.9692
	Specificity	0.7647
	F1	0.9796
	MCC	0.5998684
	AUC	0.867
	Precision	0.9903
	Recall	0.9692

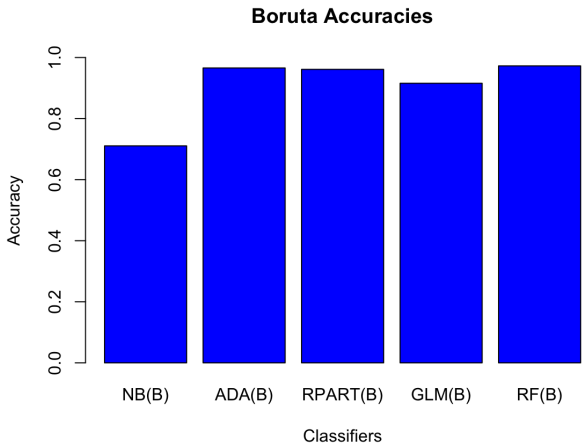
(RPart) - RFE	Reference	
Prediction	No Snow	Snow
No Snow	422	17
Snow	0	0
	Accuracy	0.9613
	Sensitivity	1.0000
	Specificity	0.0000
	F1	0.9803
	MCC	0
	AUC	0.5
	Precision	0.9613
	Recall	1.0000

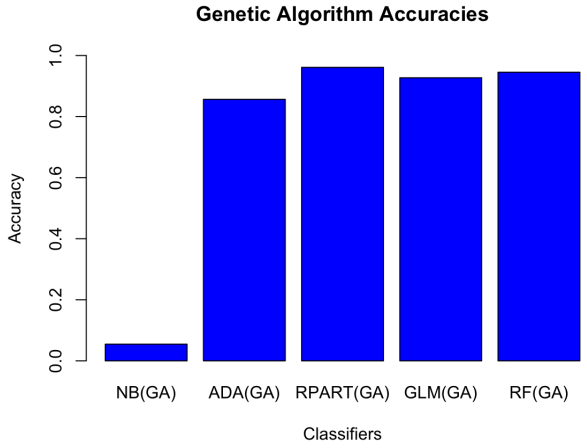
(GLM) - RFE	Reference	
Prediction	No Snow	Snow
No Snow	380	0
Snow	42	17
	Accuracy	0.9043
	Sensitivity	0.9005
	Specificity	1.0000
	F1	0.9476
	MCC	0.5093708
	AUC	0.9502
	Precision	1.0000
	Recall	0.9005

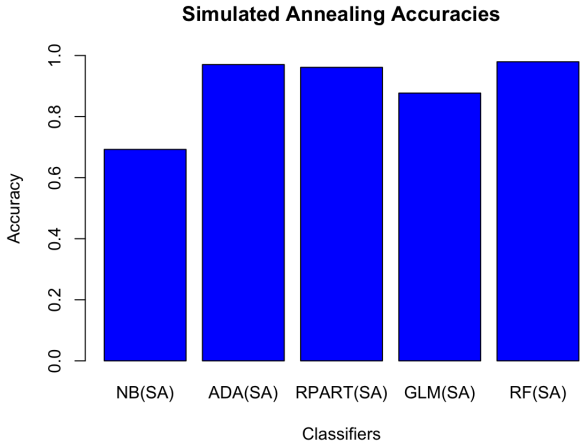
(Rand Forest) - RFE	Reference	
Prediction	No Snow	Snow
No Snow	410	4
Snow	12	13
	Accuracy	0.9636
	Sensitivity	0.9716
	Specificity	0.7647
	F1	0.9809
	MCC	0.6129817
	AUC	0.8681
	Precision	0.9903
	Recall	0.9716

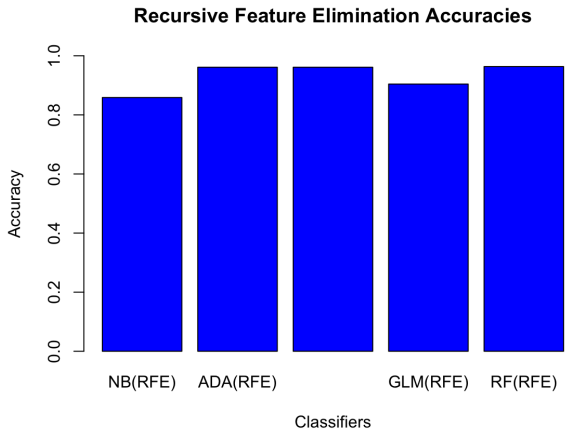
From the confusion matrices, we were able to obtain the accuracy of each classification algorithm. We plotted each of the accuracies within their attribute selection method first to see how they compared horizontally across the same set of column attributes.



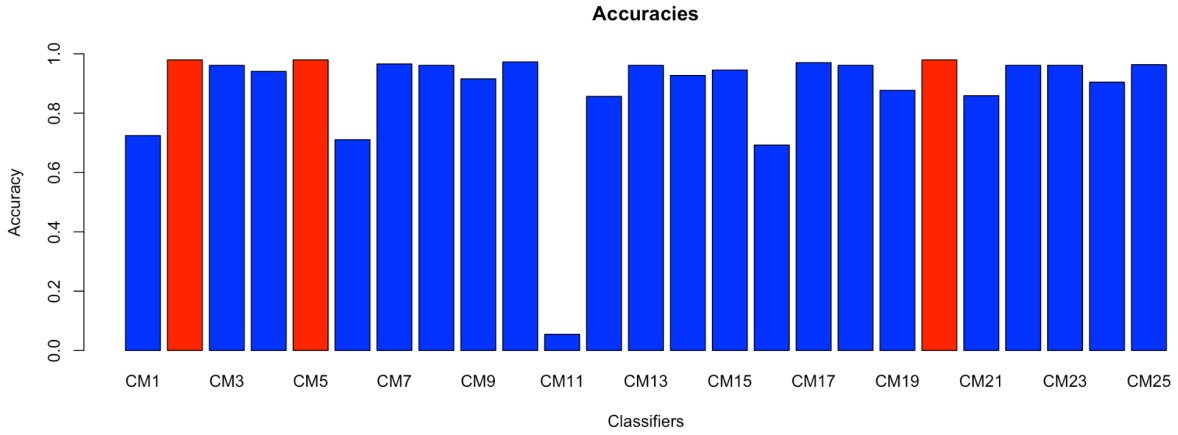








As seen in the individual bar plots of classifier accuracies, the Naive Bayes classifier proved consistently to be the worst classifier for accurately predicting ‘Snow’ versus ‘No Snow’ days. While the differences in accuracy are not always striking, we can better see the clear winner when considering all 25 classifiers side by side.



The bar plot above shows all of the accuracies plotted side by side, and the red bars indicates the maximum accuracy value, 0.979, which was achieved by the Confusion Matrices 2, 5, and 20 (highlighted in green above), corresponding to the Information Gain optimized AdaBoost, the Information Gain optimized Random Forest, and the Simulated Annealing optimized Random Forest.

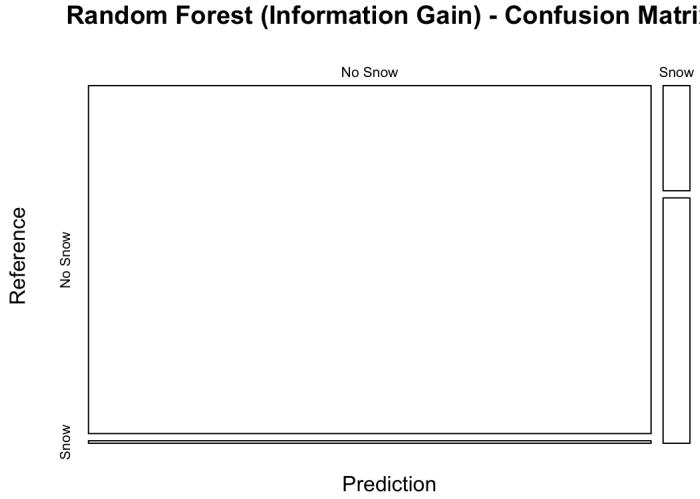
To determine which of these three classifiers is objectively the best in the scope of our project, we will revisit the confusion matrices and take a deeper look into the various metrics we can use to come to this determination.

(AdaBoost) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	417	4
Snow	5	13
	Accuracy	0.9795
	Sensitivity	0.9882
	Specificity	0.7647
	F1	0.9893
	Precision	0.9905
	Recall	0.9882

(RF) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	416	3
Snow	6	14
	Accuracy	0.9795
	Sensitivity	0.9858
	Specificity	0.8235
	F1	0.9893
	Precision	0.9928
	Recall	0.9858

(RF) - SA	Reference	
Prediction	No Snow	Snow
No Snow	417	4
Snow	5	13
	Accuracy	0.9795
	Sensitivity	0.9882
	Specificity	0.7647
	F1	0.9893
	Precision	0.9905
	Recall	0.9882

What we can determine from the extended confusion matrices shown above follows from comparing the precisions, recalls, sensitivities, and specificities of each of the classifiers. Precision, which is the measure of the true positive rate, calculated as TP / (TP + FP), tells us how many of the predicted positives are actually so. By the precision metric, the Information Gain optimized AdaBoost classifier is superior. Similarly, Recall is the measure of how many of the actual positive instances were reported as so, it is calculated as TP / (TP + FN). When considering the recall metric, the two AdaBoost classifiers are superior.



Ultimately, we considered the balanced accuracy of each of the three classifiers, which were AdaBoost (IG) = 0.8764, Random Forest (IG) = 0.9047, and Random Forest (SA) = 0.8764, to determine that the best classifier was the Information Gain optimized Random Forest classifier (CM5).

*All attributes model with best algorithm (Random Forest)

(All attributes) - Random Forest	Reference	
Prediction	No Snow	Snow
No Snow	418	5
Snow	4	12
	Accuracy	0.9795
	Sensitivity	0.9905
	Specificity	0.7059
	F1	0.9893
	MCC	0.7169875
	AUC	0.8482
	Precision	0.9882
	Recall	0.9905

(Rand Forest) - Info Gain	Reference	
Prediction	No Snow	Snow
No Snow	416	3
Snow	6	14
	Accuracy	0.9795
	Sensitivity	0.9858
	Specificity	0.8235
	F1	0.9893
	MCC	0.7488147
	AUC	0.9047
	Precision	0.9928
	Recall	0.9858

Conclusion

Throughout the course of this project, we were constantly challenged to create the best classification algorithms to determine if we could accurately classify a snow day versus a no snow day based on meteorological factors in the city of Boston, MA. By implementing various attribute selection methods, namely Information Gain, Boruta, Genetic Algorithms, Simulated Annealing, and Recursive Feature elimination, we were able to derive optimized datasets upon which to run our classifiers to help determine whether there was a best set of attributes to classify on. While the Recursive Feature Elimination optimized dataset provided the most consistently high accuracies for all five of our classification methods (Naive Bayes, AdaBoost, RPart, GLM, and Random Forest), it was the Information Gain optimized dataset which yielded the two highest accuracies across both the AdaBoost and Random Forest classifiers.

After further examining the outcomes and confusion matrices of the top 3 best classifiers, we determined that the Random Forest classifier fixed on top of the Information Gain optimized dataset was the best algorithm we were able to execute, coming in at 97.9% accuracy. The project taught us how important feature selection is, especially when dealing with large datasets which may include arguable extraneous attributes which add noise and confusion to the final outcome. When comparing our best model on the feature optimized data set versus the all attribute data set, we discovered that while the accuracy was the same, the optimized

data set resulted in greater specificity, precision, MCC, and AUC. These metrics are significant as they highlight how the feature optimized model outperformed the model on the regular dataset by more accurately identifying positive and negative outcomes, and also outperformed on overall performance as denoted by the higher AUC value. This difference can be explained by what we will refer to arguably insignificant features, details like the longitude and latitude of radar placement which may fluctuate marginally from day to day, or attributes like the elevation of the radar, which in this case should have no impact on the weather conditions caused by meteorological readings but may be mistakenly considered by the classification model if not careful.

We also learned the importance of oversampling versus undersampling. In the case of our project, had we chosen to undersample our data, it would have significantly biased our output. To make the number of snow and no snow days even in the training and test set artificially would make the classification too easy and therefore render the classifications useless, but by oversampling and arbitrarily duplicating entries to even the balance of the target outcome, we're able to construct a more robust model. Overall, this project was integral in helping advance our data processing and data mining skills around creating and testing classification algorithms, and presenting our findings in a straightforward way.

Contributions

Project Topic and Data Sourcing: Sarve Khorramshahi

Project Proposal: Sarve Khorramshahi

Data Preprocessing: Ayan Ashkenov

Feature Selection Algorithms: Ayan Ashkenov

Classification Algorithms: Ayan Ashkenov & Sarve Khorramshahi

- Naive Bayes: Ayan Ashkenov
- AdaBoost: Ayan Ashkenov
- RPart: Ayan Ashkenov
- GLM: Sarve Khorramshahi
- Random Forest: Sarve Khorramshahi

Data Visualizations: Sarve Khorramshahi

Project Report Write Up: Sarve Khorramshahi

Final Review: Ayan Ashkenov & Sarve Khorramshahi