# A Machine Learning-based Approach to Better Understand Wildfire-Climate Relationships in the Arctic Tundra (2001-2015)

Arif Masrur
IST 597: Final Project

May 4, 2018

## 1 Introduction

In recent years different parts of Arctic tundra ecosystems have experienced an increased frequency of wildfire events. Although contemporary studies illuminated on climate-wildfire linkages for some regional and historic wildfire events (i.e. in Boreal forests), a detailed understanding is lacking regarding the nature of relationship between climatic parameters and wildfire characteristics (i.e. occurrence and intensity) in the Arctic tundra ecosystems. It should be noted that, wildfires exhibit characteristics of complex system dynamics. Recent studies have found [1] [2] complex, non-linear, intertwined relationships among bio-climatic factors, human influences, and wildfire activity around the planet. Wildfires are generally driven by bottom-up (fuel, weather, and topography) and top-down (climate) controls across a range of spatial and temporal scales. This complex interplay among different climatic and biophysical controls that drive Arctic tundra wildfires can be better understood with recently made available (big) climate and wildfire datasets by the National Aeronautics and Space Administration (NASA). Hence, in this project I utilized these datasets to conduct a statistical machine learning-based analyses to elucidate on poorly-understood Pan-Arctic tundra wildfire-climate dynamics.

***Objective:***

The key question answered in this project is: what is the nature of relationship between wildfire occurrence and climate parameters in the Arctic tundra at seasonal scale. Exploratory data analyses found that Arctic tundra wildfire season generally spans over May-October months - a typical summer period in the circumpolar Arctic.

***Data:***

In order to elucidate on explicit climate associations of recent Tundra wildfires, I have used *fire occurrence instances (yes/no fire)* variable found in NASA MODIS Wildfire dataset (2001-2015) as outcome variable, and six climatic parameters in MERRA (Modern-Era Retrospective analysis for Research and Applications) dataset as predictor variables (Table 1).

NASA TerraâĂŹs MODIS (Moderate Resolution Imaging Spectroradiometer) daily active wildfire products (available from 2000 to date) provide a new generation of moderate resolution (1 km) remotely-sensed global

fire datasets. On the other hand, the MERRA-Land dataset contains hourly averages of land surface fields (i.e., surface temperature) from January 1st, 1980 at a horizontal resolution of 2/3° longitude by 1/2° latitude.

| ***Predictor Variables (MERRA)*** | **Description** | **Units** |
|---|---|---|
| | ***Explanatory variables (MERRA-Land)*** | |
| TSURF | Mean soil **surface temperature** (including snow) | K |
| PRECTOT | Total **surface precipitation** | kg $m^{-2}$ $s^{-1}$ |
| SFMC | Top **soil layer moisture** content | kg $m^{-2}$ $s^{-1}$ |
| EVLAND | **Evaporation** from land | kg $m^{-2}$ $s^{-1}$ |
| *(For detail information on MERRA-Land climate variables, see Gelaro et al, 2017; Reichle, 2012)* | | |
| Temperature and Precipitation **Anomalies** | 2001-2015 average monthly **TSURF anomaly** | K |
| | 2001-2015 average monthly **PRECTOT anomaly** | kg $m^{-2}$ $s^{-1}$ |
| ***Response Variable (MODIS Active Fires)*** | | |
| Fire instances (yes/no fire) | *Yes* = at least one fire occurred per spatial cell, *No* = no fire occurred per spatial cell (over the study period) | |

Table 1. MODIS active Wildfire and MERRA climate varibales used in this study.

***Methods:*** To investigate wildfire-climate relationships, response variable (i.e. wildfire occurrence) was aggregated into 2/3° longitude by 1/2° latitude grid cells (conforming to MERRA's spatial scale). All the variables used in this project are documented in Table 1.

This seasonal-scale analysis was aimed at identifying if monthly climatic conditions of individual winter (November-February), spring (March-April), and summer (May-October) months had been influential in determining the likelihood of summer season tundra wildfire occurrences between 2001 and 2015. Since wildfire occurrence is not determined by climatic conditions alone, hence I don't expect that climate variables will be able to explain much variability in the outcome. However, with the rapid climate warming in the Arctic and associated vegetation change, it is reasonable to believe that climatic conditions may have played significant role in wildfire activity in the Pan-Arctic tundra ecosystems over the past 15 years.

To answer the overarching question, in this project I utilized five different statistical machine learning techniques. These are:

1. Naive Bayes,
2. Logistic Regression,
3. Support Vector Machines (SVM),
4. Random Forest, and
5. Adaptive Boosting (AdaBoost) and Extreme Gradient Boosting (XGBoost)

## 2 Exploratory Data Analysis (EDA)

Due to the time-series and spatial nature of wildfire-climate dataset, it's been essential to explore data distribution of predictor variables and multi-collinearity diagnostics. The distributions of some monthly predictor variables were found skewed. These variables were scaled before an during fitting classification models. The issues with multi-collinearity were dealt with different variable selection methods, discussed later.

## 3 Wildfire Occurrence and Climate

This part focused on analyzing the influence of seasonally aggregated climate parameters for tundra wildfire *occurrence*. Since the response variable is categorical ('yes' or 'no' fire), its a classification problem. Therefore, I started with widely-used classification measure - Naive Bayes and then logistic regression - to model the tundra wildfire occurrence and climate relationship.

$$\Pr(\text{fire} = \text{Yes} \mid 72 \text{ climate parameters})$$

The dataset consists of one response (class values are 1 and 0; 'yes' as 1 and 'no' fire as 0) and 72 climatic predictor variables (6 monthly climate variables for 12 months âĂŞ see Table 1 and 2). There are 940 observations (both fire and non-fire cells) among which 704 cells contained wildfire events for each season between 2001 and 2015. The remained 470 non-fire cell observations were randomly chosen around neighboring wildfire cells.

Before fitting every classification model, I randomly partitioned the wildfire-climate dataset into training (75%) and test (25%) datasets. Therefore, out of total 940 'yes' fire and 'no' fire observations (spatial cells), 705 were randomly designated as training and 235 as test observations.

### 3.1 Naive Bayes

The first method applied here is Naive Bayes (NB) classification. Naive Bayes classifier is widely used for its simplicity and thus oftentimes outperforming even highly sophisticated classification methods. This motivated me to utilize Naive Bayes classifier on the wildfire-climate dataset. The classification accuracy is 69% and the AUC is reported in Figure 1. This poor performance may have resulted from NB classifier's assumption of independence among predictors - probably an unlikely scenario in a dataset of time-series nature. Additionally, presumably the existence of a non-linear class boundary in the dataset also contributed to considerably high classification error rate.

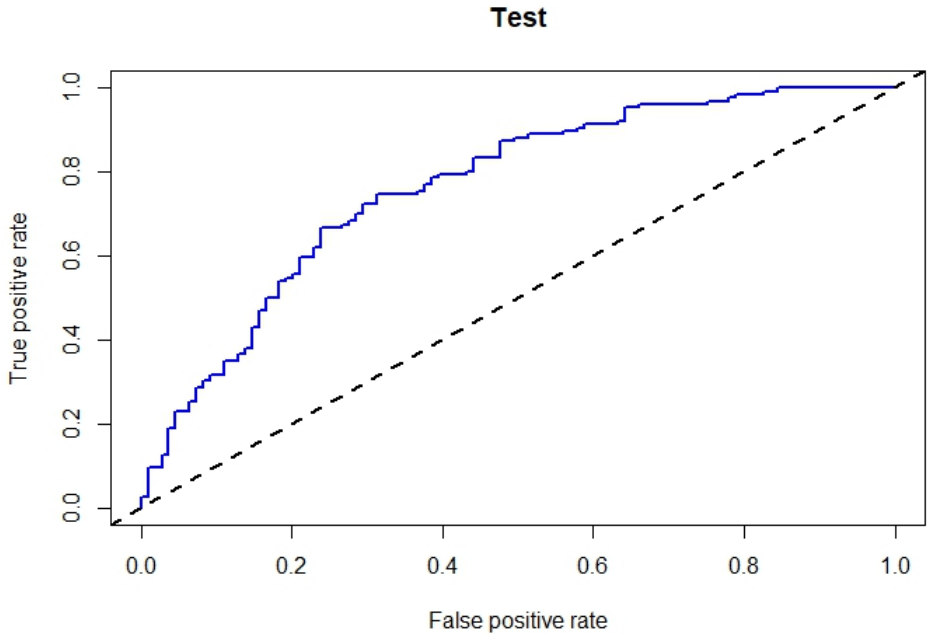


Figure 1. Performance of Naive Bayes Classification.

## 3.2 Logistic Regression

Next, I fitted a logistic regression model using glm() function in R (a class of generalized linear model that includes logistic regression) with binomial link. The output showed only 24 (out of 72) predictors were statistically significant. To address this issue, I performed *over-dispersion* and *collinearity* tests. Pearson residuals were used to check for over-dispersion which allowed analyzing whether the actual variance of response was greater than that suggested by the model.

Pearson residuals appeared to have mean close to zero and variance close to one - an evidence suggestive of no over-dispersion. Additionally, no influential outlier(s) was detected (see R output). In Looking for collinearity, 92 pairs of predictors had correlations greater or equal to 0.7 and 30 pairs had correlation greater or equal to 0.9. To find most relevant variables that are associated with the response ('yes' fire vs. 'no' fire), I used Lasso Regression - a popularly used regularization method that sets coefficient estimates of irrelevant predictors to zero and thereby enhances *model interpretability* and *prediction accuracy* . Additionally, for the purpose of enhancing model's prediction power I used 10-fold cross validation. I randomly partitioned into training (75% of observations) and test (25% of observations) sets.

I used a threshold probability of 0.5 to classify into class 0 or 1. The prediction power of the fitted model on test dataset showed that 73% of times the model correctly predicted the fire occurrences.

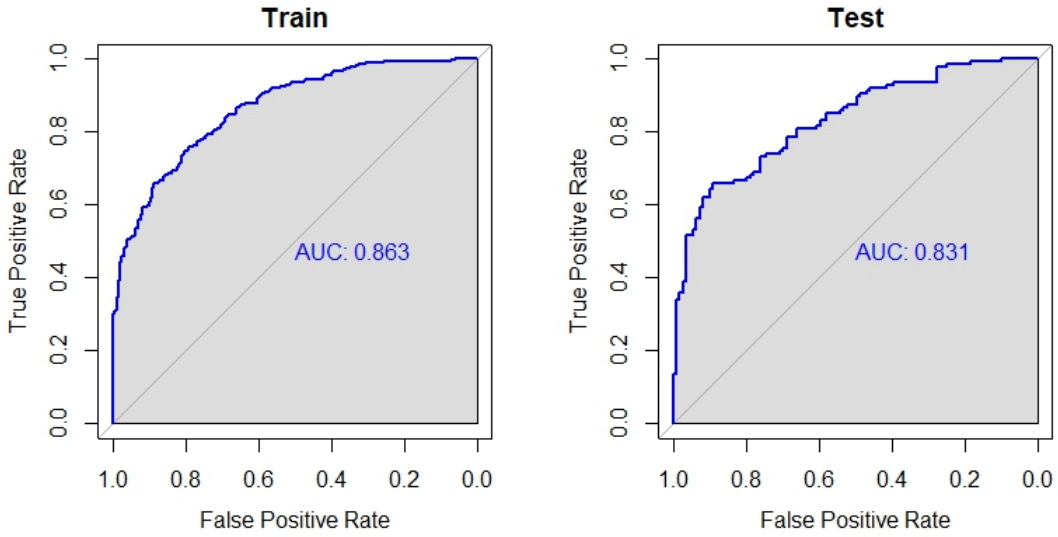


Figure 2. Performance of Logistic Regression Classification.

It should be noted that I aimed at finding classification model for prediction of fire occurrences. While logistic model has reasonably well performed in terms of prediction than the Naive Bayes (See AUC values in Figure 1 and 2), this may not hold for unknown future observations. Moreover, in order to be able to make interpretations about coefficients and infer a causal relation between response and predictors further analysis is necessary. Using other methods of regularization, aggregating observations to get only pre-fire 3-month average instead of monthly observations, or using $\acute{\lambda} = \lambda + SD(\lambda)$ as best lambda in lasso regression analysis and Principal Component Analysis are possible approaches towards a model with sensible interpretation.

### 3.3 Decision Trees: Random Forests

To supplement Logistic Regression, next I focused on employing decision tree-based classification method. I chose widely-used (in wildfire research) Random Forest classification - a non-linear, tree-based ensemble learning method, to examine the nature of tundra wildfire-climate relationship, since this method usually yields relatively better classification results among all tree-based methods. As opposed to growing single decision tree (as in CART), random forest grows multiple trees, with having each split to consider only a subset of all predictors.Then it takes average of all trees to make final tree. In this way, random forest can reduce amount of potential correlation between trees and thereby helps reduce the variance of the final tree. In addition, while tree-based method don't have same level of predictive accuracy as classical methods (e.g. logistic regression), Random Forest can actually yield improved prediction accuracy through a "consensus" prediction based on many aggregated decision trees [3].

Before fitting Random Forest model, I used tuneRF function to find the optimal numbers of variables to try (known as *mtry*) splitting on at each node. I found mtry = 8 produces least out of the box (OBB) error (17.87%), that means, 8 out of 72 predictors should be considered for each split. Then I fitted random forest model on training data using mtry = 8 and 1000 trees (known as *ntree*). Its worth mentioning that,

I also tried other ntree values, e.g. ntree = 100, 1000, 1500, and 2000, which didn't cause any significant differences in model performance. From plotted model (Appendix 1: Figure 1) I can see that, albeit having little fluctuations after about 600 trees there wasn't much changes in terms of prediction error. Black solid line is for overall OOB error, green line represents classification error rate for 'yes' fire, whereas red line represents classification error rate for 'no' fire cases.

I found that the training model fitted the training data fairly well. The prediction accuracy was surprisingly 100%. Next, I used this fitted model on training dataset to predict 'yes' fire and 'no' fire instances in the test dataset. The prediction accuracy dropped to 89.8%. The Receiver operating characteristic (ROC) curve for the prediction in test dataset is included in Figure 3.

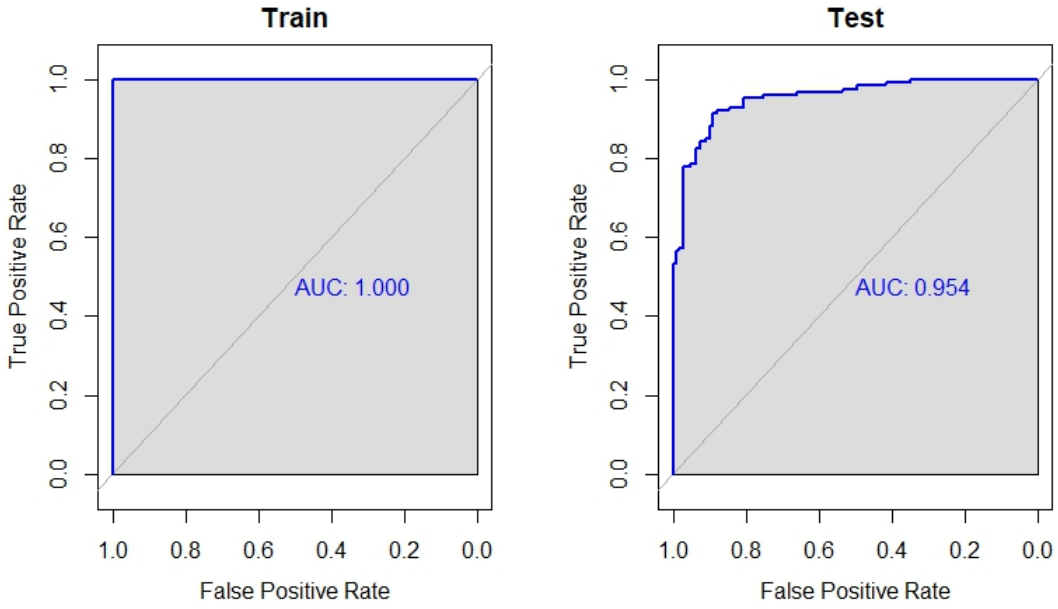


Figure 3. Performance of Random Forest Classification.

In the Variable Importance Plot (Figure 4), predictor variables that resulted in nodes with higher purity have a higher decrease in Gini coefficient. The June surface temperature anomalies variable were by far the most important variable in determining a 'yes' fire vs. 'no' fire instance. It seems overall surface temperature values in the pre-fire and on-fire months have been very important drivers for tundra wildfire occurrence in 2001-2015 period.

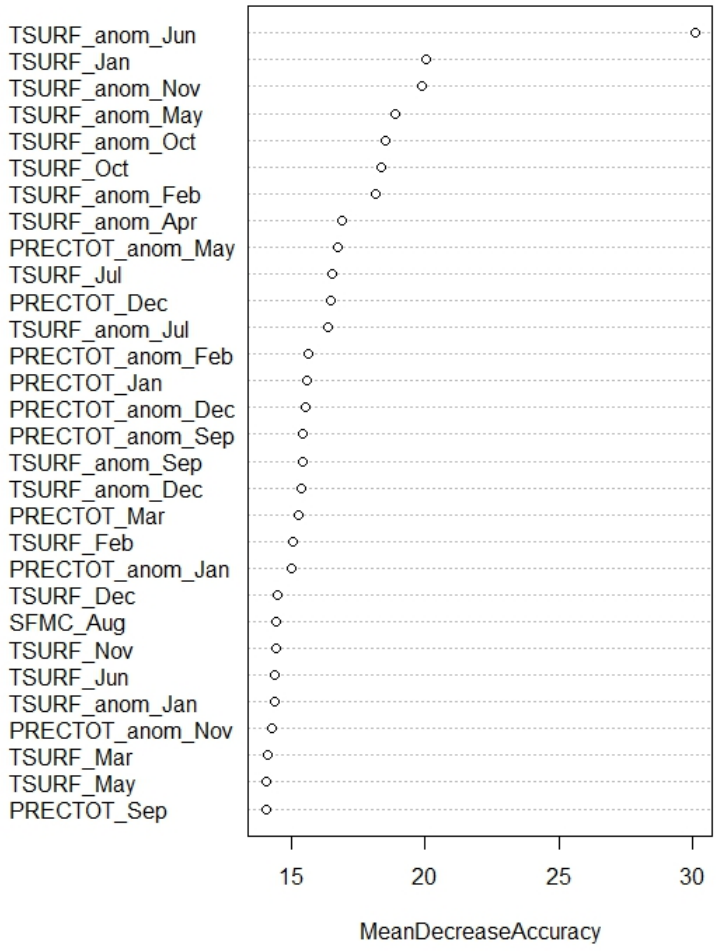


Figure 4. Variable Importance Plot from Random Forest Classification.

### 3.4 Adaptive Boosting (AdaBoost)

Next, I implemented algorithm of AdaBoost method which is usually best used to boost the performance of decision trees on binary classification problems. Here I found classification accuracy on the training dataset being 94%, while test accuracy dropped to 83%. All parameters used in the AdaBoost models are documented in Appendix 2.

### 3.5 Extreme Gradient Boosting (XGBoost)

The implemented algorithm for Extreme Gradient Boosting (XGBoost) resulted in following variable importance plot which somewhat matches with the one from Random Forest classification.

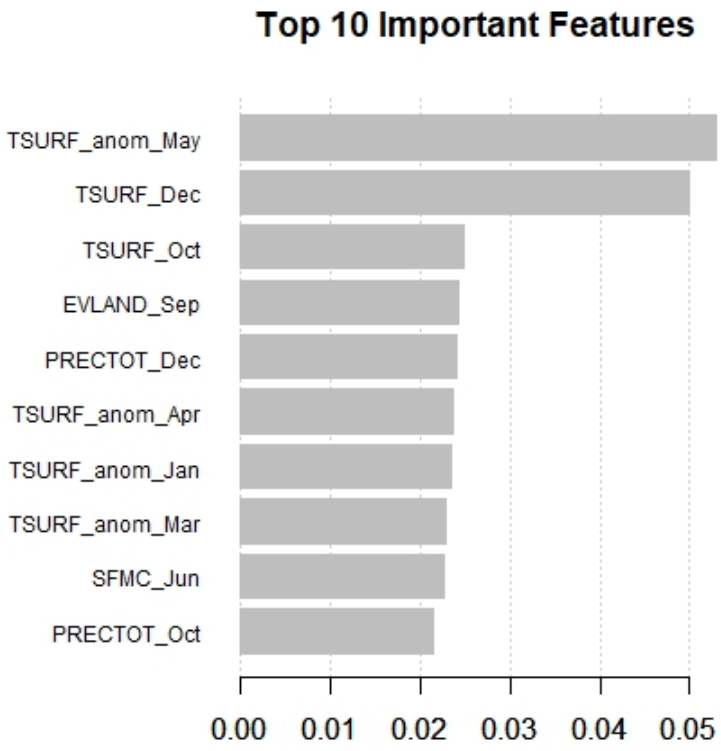


Figure 5. Variable Importance Plot from XGBoost.

# 4 Support Vector Machine (SVM)

Next, I implemented support vector machine algorithm that allows for the classification of data with non-linear boundaries. The idea is to project the data into a space of higher dimension, determine decision boundaries, and then project back into the original space. I attempted to fit this approach to the wildfire-climate dataset, using a linear, radial, and polynomial kernel.The AuC's are reported in Figure 6, 7, and 8. It can be seen that SVM classifier with Polynomial Kernel outperformed classifiers with both linear and radial kernels. Relevant R codes are provided in Appendix 2.

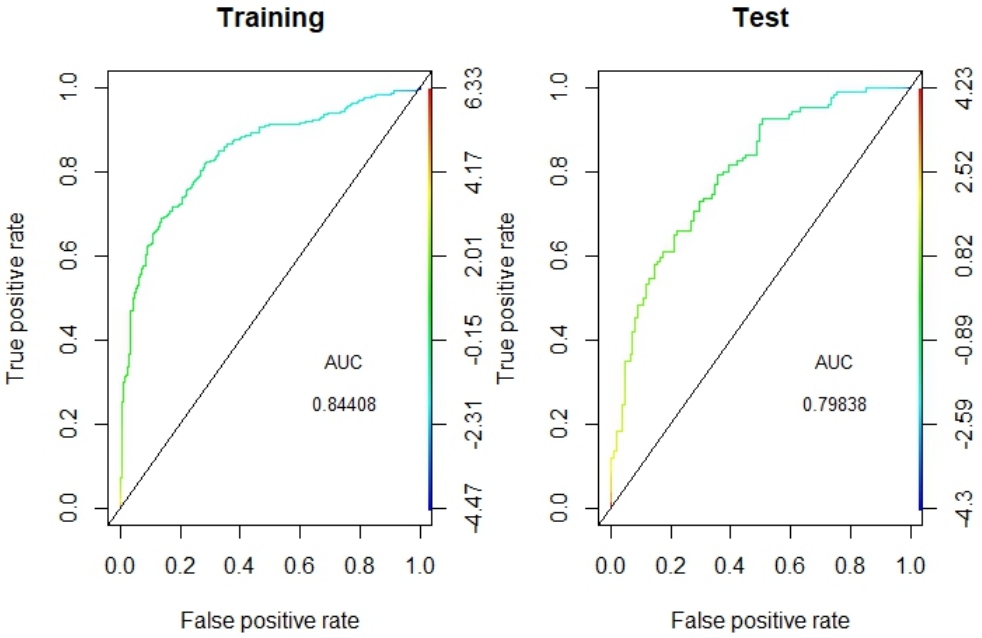


Figure 6. Performance of Support Vector Machine with Linear Kernel.

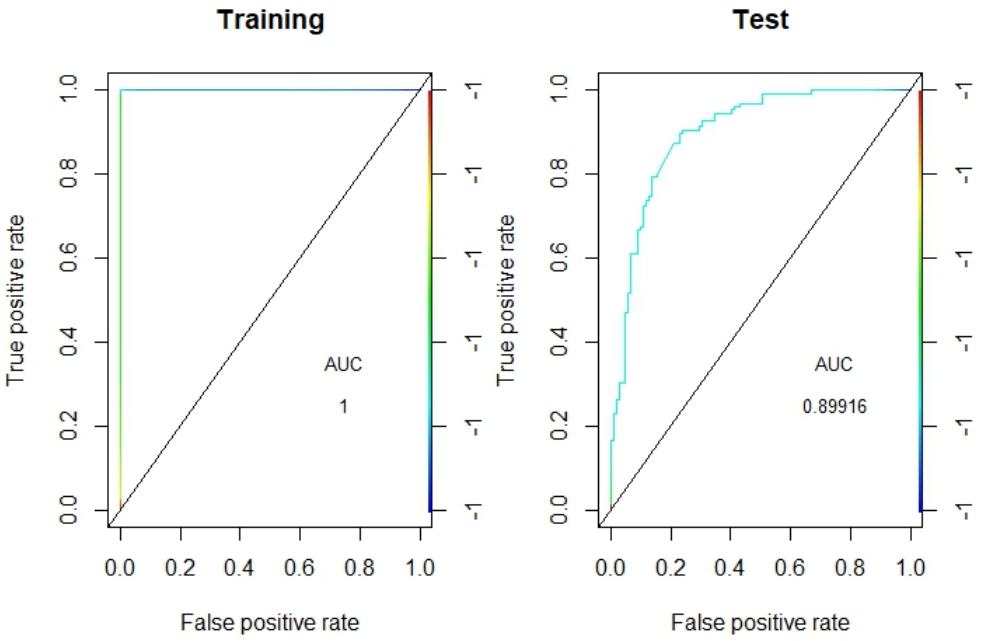


Figure 7. Performance of Support Vector Machine with Radial Kernel.

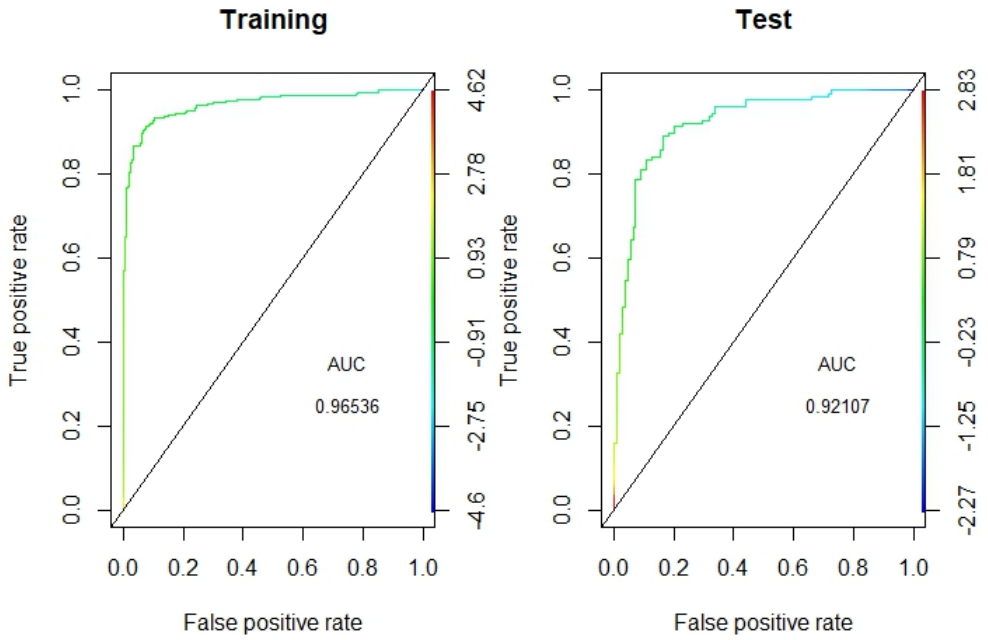


Figure 8. Performance of Support Vector Machine with Polynomial Kernel.

# 5 Discussion

The classification methods utilized in this analyses aimed at predicting a binary response - fire or no-fire. Among all methods, the tree-based methods seem to have outperformed Naive Bayes, Logistic Regression, and Support Vector Machines (SVMs). However, SVMs with radial and polynomial kernels have resulted in better prediction accuracies than the Naive Bayes and Logistic Regression. This may bolster the understanding to the existence of a complex, non-linear relationships between wildfire occurrence and climate variables. More important predictors to wildfire, i.e. fuel level and fire trigger (i.e. lightning by thunderstorms or ignition by human activity) could have contributed to improved prediction accuracies of classification models, as well as more refined understanding of the Arctic wildfire-climate dynamics. Future studies should incorporate these datasets for better prediction of wildfire occurrences in the circumpolar Arctic tundra.

# 6 Appendix 1

## 6.1 Figures

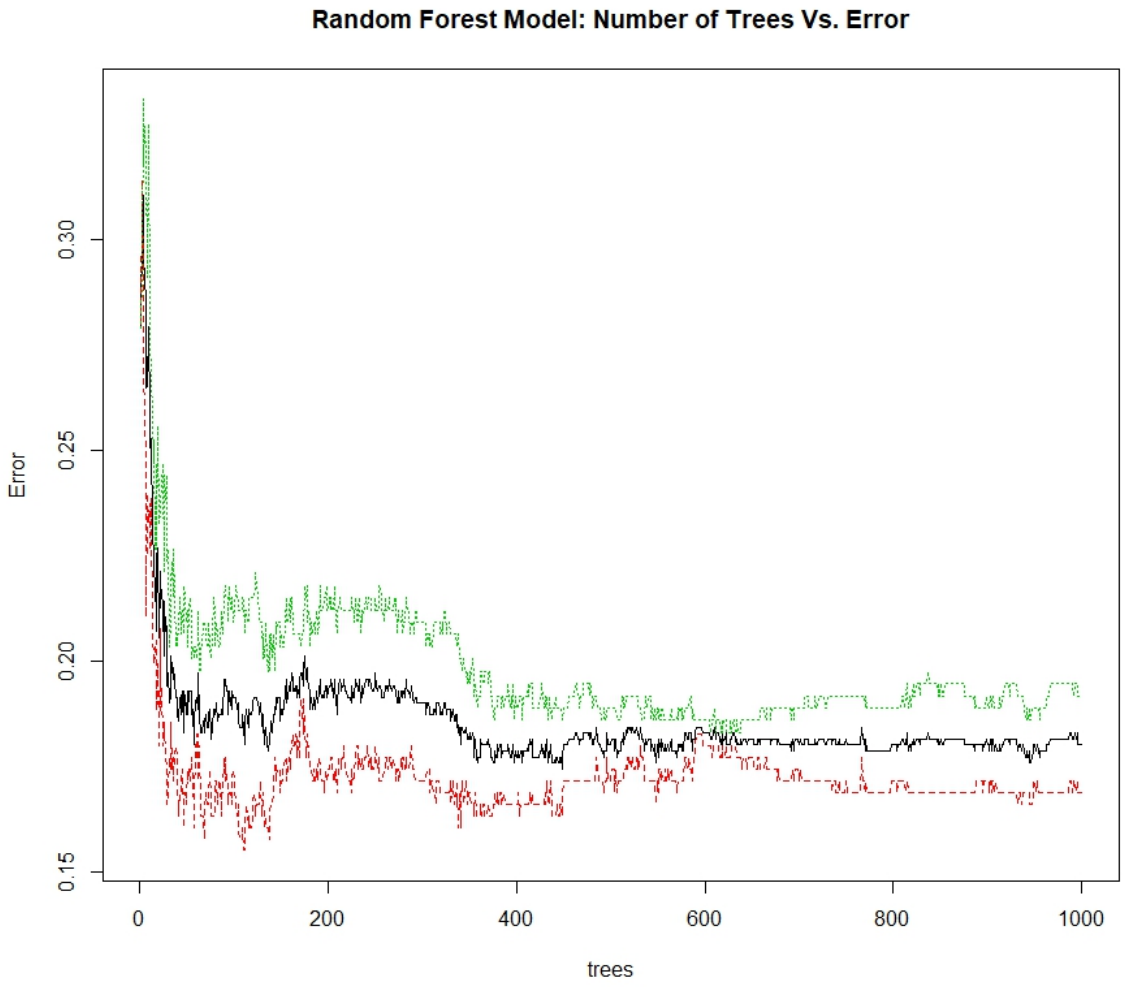


Figure 1: Random Forest Model: Tress vs. Prediction Errors

# 7 Appendix 2: Commented R code

## 7.1 AdaBoost

```
# Build best ada boost model
model = ada(x = fire.train[,-1],
            y = fire.train$Y,
            iter=20, loss="logistic", verbose=TRUE) # 20 Iterations
```

```
# Look at the model summary
model
summary(model)

# Predict on train data
pred_Train  =  predict(model, fire.train[,-1])

# Build confusion matrix and find accuracy
cm_Train = table(fire.train$Y, pred_Train)
accu_Train= sum(diag(cm_Train))/sum(cm_Train)
#rm(pred_Train, cm_Train)

# Predict on test data
pred_Test = predict(model, fire.test[,-1])

# Build confusion matrix and find accuracy
cm_Test = table(fire.test$Y, pred_Test)
accu_Test= sum(diag(cm_Test))/sum(cm_Test)
#rm(pred_Test, cm_Test)

accu_Train
accu_Test
```

## 7.2 XGBoost

```

xgb_fire = xgboost(data = data.matrix(fire.train[,-1]),
                   label = fire.train$Y,
                   eta = 0.1,
                   max_depth = 15,
                   nround = 25,
                   subsample = 0.5,
                   colsample_bytree = 0.5,
                   seed = 1,
                   eval_metric = "merror",
                   objective = "multi:softprob",
                   num_class = 12,
                   nthread = 3)

y_pred = predict(xgb_fire, data.matrix(fire.test[,-1]))

# Let's see what actaul tree looks like:
model = xgb.dump(xgb_fire, with_stats = T)
# Top 10 nodes of the model

model[1:10]

```

```
# Get the feature real names
names = dimnames(data.matrix(fire.train,-1))[[2]]



# Compute feature importance matrix
importance_matrix <- xgb.importance(names, model = xgb_fire)

# Graph
xgb.plot.importance(importance_matrix[1:10,])
```

## 7.3 Support Vector Machine with 3 Kernels

```
set.seed(2016)
test <- sample(n, round(n/4))
train <- (1:n)[-test]
fire.train <- firedata[train,]
fire.test <- firedata[test,]

#---------------------------------------------------
# Linear kernel
#---------------------------------------------------



tune.out <- tune(svm, Y ~., data = fire.train, kernel = "linear", ranges = list(cost = c
    (.001,.01,.1,1,5)))

# Look for a best model
summary(tune.out)
bestmod <- tune.out$best.model
summary(bestmod)

ypred <- predict(bestmod, fire.test)
table(predict = ypred, truth = fire.test$Y)

svmfit.best1 <- svm(Y~., data = fire.train, kernel = "linear", cost = bestmod$cost, decision.values = T
    )
fitted1 <- attributes(predict(svmfit.best1, fire.train, decision.values = T))$decision.values
fitted.test1 <- attributes(predict(svmfit.best1, fire.test, decision.values = T))$decision.values

rocplot(fitted1, fire.train$Y, main = "Training")
rocplot(fitted.test1, fire.test$Y, main = "Test")



decisionplot(svmfit.best1, fire.train, class = "Y", main = "SVD (linear)")



#---------------------------------------------------
```

```
# Radial kernel
#-------------------------------------------------

tune.out2 <- tune(svm, Y ~., data = fire.train, kernel = "radial", ranges = list(cost = c(.001,.01,.1),
      gamma = c(1,5,50)))

bestmod <- tune.out2$best.model
ypred <- predict(bestmod, fire.test)
table(predict = ypred, truth = fire.test$Y)

svmrad2 <- svm(Y~., data = fire.train, kernel = "radial", gamma = tune.out2$best.model$gamma, cost =
     tune.out2$best.model$cost, decision.values = T)
fitted2 <- attributes(predict(svmrad2, fire.train, decision.values = T))$decision.values
fitted.test2 <- attributes(predict(svmrad2, fire.test, decision.values = T))$decision.values

rocplot(fitted2, fire.train$Y, main = "Training")
rocplot(fitted.test2, fire.test$Y, main = "Test")

#-------------------------------------------------
# Polynomial kernel
#-------------------------------------------------

tune.out3 <- tune(svm, Y ~., data = fire.train, kernel = "polynomial", ranges = list(cost = c
     (.001,.01,.1,1), degree = c(2,3)))
bestmod <- tune.out3$best.model
ypred <- predict(bestmod, fire.test)
table(predict = ypred, truth = fire.test$Y)

svmrad3 <- svm(Y~., data = fire.train, kernel = "polynomial", cost = tune.out3$best.model$cost, degree
     = tune.out3$best.model$degree, decision.values = T)
fitted3 <- attributes(predict(svmrad3, fire.train, decision.values = T))$decision.values
fitted.test3 <- attributes(predict(svmrad3, fire.test, decision.values = T))$decision.values

rocplot(fitted3, fire.train$Y, main = "Training")
rocplot(fitted.test3, fire.test$Y, main = "Test")
```

# References


[1] Bowman, D.M., Balch, J.K., Artaxo, P., Bond, W.J., Carlson, J.M., Cochrane, M.A., DâĂŹAntonio, C.M., DeFries, R.S., Doyle, J.C., Harrison, S.P. and Johnston, F.H., 2009. Fire in the Earth system. science, 324(5926), pp.481-484.

[2] Masrur, A., Petrov, A.N. and DeGroote, J., 2018. Circumpolar spatio-temporal patterns and contributing climatic factors of wildfire activity in the Arctic tundra from 2001âĂŞ2015. Environmental Research Letters, 13(1), p.014019.

[3] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An introduction to statistical learning (Vol. 112). New York: springer.