Forest Fire Project
Shi Fan Jin, Esther Law, Changning Liu
December 7, 2018

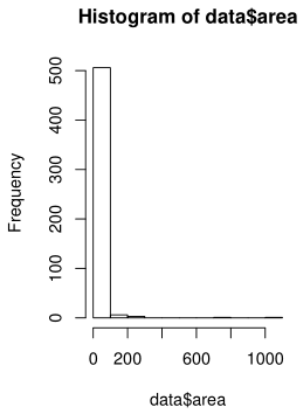
Introduction of Data

The data set that we are using is “ForestFires” which contains information about fires in the Montesinho Park, which is in northeastern Portugal. There are 517 observations in 13 variables. These variables include the location of the fire, total area burned, and other meteorological information. Below is the list and description of each variable as it is found on Kaggle.com (also found in Cortez and Morais, 2007):

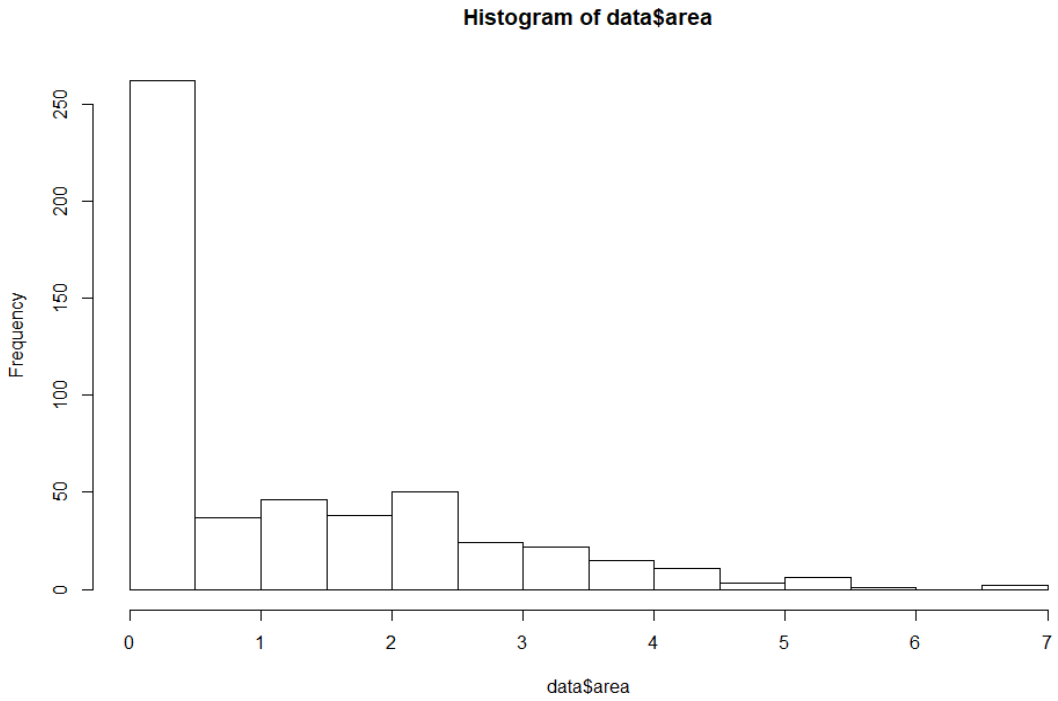
1. X - x-axis spatial coordinate within the Montesinho park map: 1 to 9
2. Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9
3. month - month of the year: 'jan' to 'dec'
4. day - day of the week: 'mon' to 'sun'
5. FFMC - FFMC index from the FWI system: 18.7 to 96.20
6. DMC - DMC index from the FWI system: 1.1 to 291.3
7. DC - DC index from the FWI system: 7.9 to 860.6
8. ISI - ISI index from the FWI system: 0.0 to 56.10
9. temp - temperature in Celsius degrees: 2.2 to 33.30
10. RH - relative humidity in %: 15.0 to 100
11. wind - wind speed in km/h: 0.40 to 9.40
12. rain - outside rain in mm/m2 : 0.0 to 6.4
13. area - the burned area of the forest (in ha): 0.00 to 1090.84

We want to be able to predict the amount of area that a fire in Montesinho Park would burn given the meteorological information. An important piece of information to remember about this data is that all 517 observations represent fires that had happened. Therefore, observations that have area value equal to zero does not mean that there was no fire. Instead, it means that the area burned was less than 1 hectare, or less than roughly 2.5 acres.

First, we looked at the distribution of the data for “area”. This is plotted on the right. We noticed that the data is incredibly skewed right. In fact, about half of the observations have value for area equal to zero. So we implement a transformation of the data by using log(area+1) instead of area. Even though this will not take care of the zero area problem, it will at least spread out the data a little bit.



After taking the log of the area, we end up with the graph below. As mentioned, we cannot completely remove the right skew since half of the observations have area equals to zero. However, the values are more spread out than in the previous histogram.



Similarly, we transformed the "FFMC" variable by raising its values to the 13th power. The “ISI” variable contained one outlier that made its values skewed. So we removed the row containing this outlier. This made our visualization cleaner and easier to see.

Finally, when we looked at the different values for the “rain” variable, we saw that almost 99.99% of the fires happened when there is no rain (“rain” = 0). We decided to remove the variable “rain” because this variable will not give us any information for a majority of our analysis. The table below shows the “rain” values and how often those values appeared. However, we note that rain is actually important in containing or putting out fires. Therefore, we are not removing the “rain” variable because rain is not important, but rather because it will not affect our modeling methods.

Table of “rain” frequency:

0	0.2	0.4	0.8	1	1.4	6.4
509	2	1	2	1	1	1

The summary statistics for the variables we worked with after these changes to the data set are:

```
> summary(data)
       X               Y             month              day
 Min.   :1.000   Min.   :2.0   Length:516         Length:516
 1st Qu.:3.000   1st Qu.:4.0   Class :character   Class :character
 Median :4.000   Median :4.0   Mode  :character   Mode  :character
 Mean   :4.665   Mean   :4.3
 3rd Qu.:7.000   3rd Qu.:5.0
 Max.   :9.000   Max.   :9.0
      FFMC                DMC               DC              ISI
 Min.   :3.419e+16   Min.   :  1.10   Min.   :  7.9   Min.   : 0.000
 1st Qu.:2.616e+25   1st Qu.: 67.03   1st Qu.:440.1   1st Qu.: 6.475
 Median :3.196e+25   Median :108.30   Median :664.2   Median : 8.400
 Mean   :3.178e+25   Mean   :110.90   Mean   :548.6   Mean   : 8.930
 3rd Qu.:3.839e+25   3rd Qu.:142.40   3rd Qu.:713.9   3rd Qu.:10.725
 Max.   :6.043e+25   Max.   :291.30   Max.   :860.6   Max.   :22.700
      temp             RH              wind             area
 Min.   : 2.20   Min.   : 15.00   Min.   :0.400   Min.   :0.0000
 1st Qu.:15.50   1st Qu.: 32.75   1st Qu.:2.700   1st Qu.:0.0000
 Median :19.30   Median : 41.50   Median :4.000   Median :0.4252
 Mean   :18.89   Mean   : 44.29   Mean   :4.017   Mean   :1.1132
 3rd Qu.:22.80   3rd Qu.: 53.00   3rd Qu.:4.900   3rd Qu.:2.0245
 Max.   :33.30   Max.   :100.00   Max.   :9.400   Max.   :6.9956
>
```

What we are trying to find:

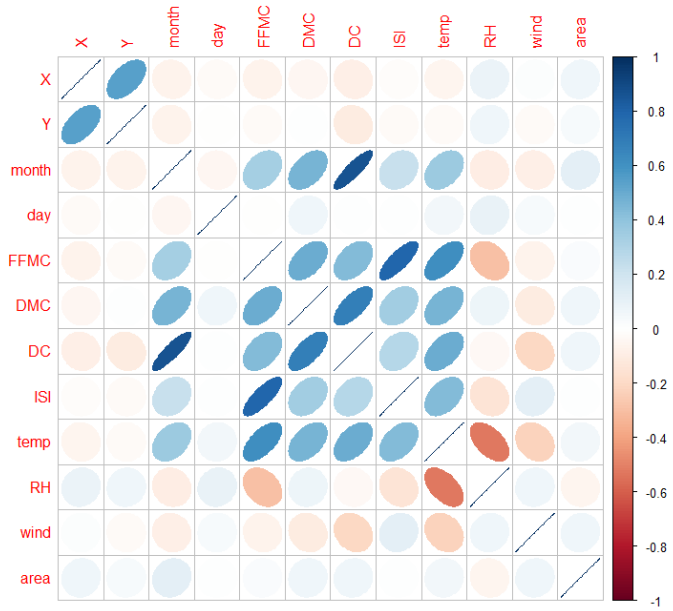
In this project, we will try to build a model to predict the area burned by fire given the other meteorological information.

Possible trends and patterns:

We looked at the values of each variable plotted against each other first. Half of the area data has a value of zero, and they occur at every value of the other variables except for two variables. However, for both ISI and FFMC^13, the values for log(area+1) is still very diverse. So we predict that there is not a strong model that will predict log(area+1).

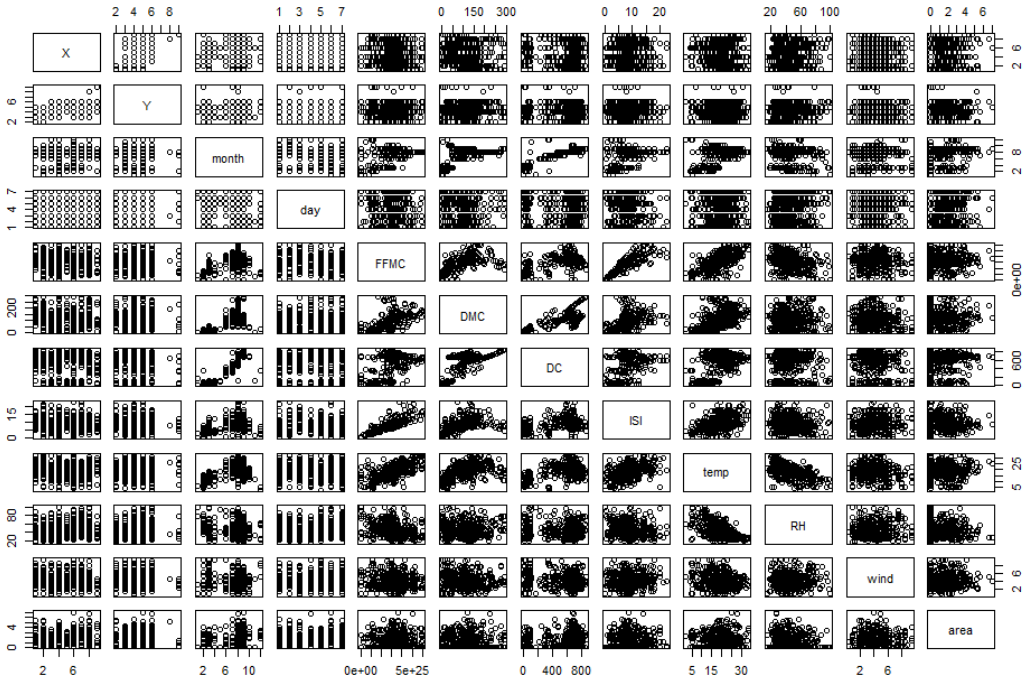
We also looked at the correlation matrix, which is attached below. What we found was that some of the variables are moderately correlated with each other. We will keep this in mind if these variables come up during our model selection, so we will not discuss them in detail here. Also, we notice that area is not too correlated with any of the variables. This again suggest that it will be difficult to build a model with high test accuracy.

Correlation Matrix of the Variables



Another observation from this plot is that “DC” is strongly correlated with “month” and “DMC”. These relationships will prompt the use of random forests later in this report. This relationship is also seen in the pairs plot below.

Plot of pairs of variables



The data in the plots above are all spread out instead of being skewed. Finally, we made dummy variables for all of the categorical values for month and day. This is

because later months are no greater than earlier months, so using just numbers 1 to 12 for the months do not make sense.

For the following methods, we randomly selected a training set of 344 observations from the original data. This is about two-thirds of the observations. The remaining 172 observations are part of our testing set.

Model Selection and LOOCV

Forward/Backward Selection

We want to start with finding a model to predict the area burned using the values for the other variables. Since there are twelve other variables, our model would be too complicated if we used all of the variables. So we need to find out which variables are good to put into a linear model, and which ones are not good. To do this, we use forward and backward selection and choose the model that gives us the lowest AIC value.

The forward selection works by starting with the simplest linear model for the training data, which is just using an intercept. Then we add variables one-by-one to the model and choose the variable that gives the lowest AIC values. We continue adding variables in this way until the AIC value will not drop anymore. The backward selection works by starting with all of the variables in the model. Then we take out a variable one-by-one and choose to take out the variable that would lower the AIC value. We continue taking out variables until none of the variables would lower the AIC when taken out.

Using Forward Selection, it gave us the output of log(area+1) ~ dec+DMC+X+sep, with the lowest AIC = 1186.4. On the other hand, using Backward Selection, it gave us log(area+1) ~ X+DMC+temp+jan+feb+mar+apr+jun+jul+aug+sep+oct, with the lowest AIC = 1196.18.

The summary of Forward:

We noticed that sep is not significant whereas dec is very significant in this forward model. This is not expected since December in Portugal actually has mild, cool weather. It is not very hot or dry. Another unexpected variable that is significant is X, or the x coordinate of where the fires occurred. This suggests that certain areas of the park are more prone to fire than others. Finally, DMC is a significant variable which is not too surprising since this is an index of moisture. We should assume that more moisture would result in smaller fires, even though we cannot infer this from the R output. However, this is some explanation as to why DMC is a significant variable.

```
> summary(fwd.model)

Call:
glm(formula = area ~ dec + DMC + X + sep, data = data[train,
    ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7275  -1.0296  -0.6187   0.7914   5.2285

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.463222   0.217028   2.134   0.0335 *
dec         1.880864   0.614836   3.059   0.0024 **
DMC         0.002302   0.001147   2.007   0.0456 *
X           0.051585   0.031172   1.655   0.0989 .
sep         0.241941   0.156250   1.548   0.1225
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.805297)

    Null deviance: 642.2  on 343  degrees of freedom
Residual deviance: 612.0  on 339  degrees of freedom
AIC: 1186.4

Number of Fisher Scoring iterations: 2
```

The summary of Backward:

We noticed that jun and aug are both very significant in the Backward model compare to other variables. This makes sense since Portugal has high temperature in summer seasons. In addition, we can see that actually, all of the months here are significant (jan, feb, mar, apr, jun, jul, aug, sep, oct). However, in the backward selection, DMC is not a significant variable. This is unexpected because it is the index of moisture which should be significant intuitively, and it contradicts with our forward selection. What is worrisome about this model is that most of the predictor variables are dummy variables. Only three of the twelve variables are not dichotomous. We know that meteorological factors can greatly affect the size of fires, and that those factors can be affected by the month.

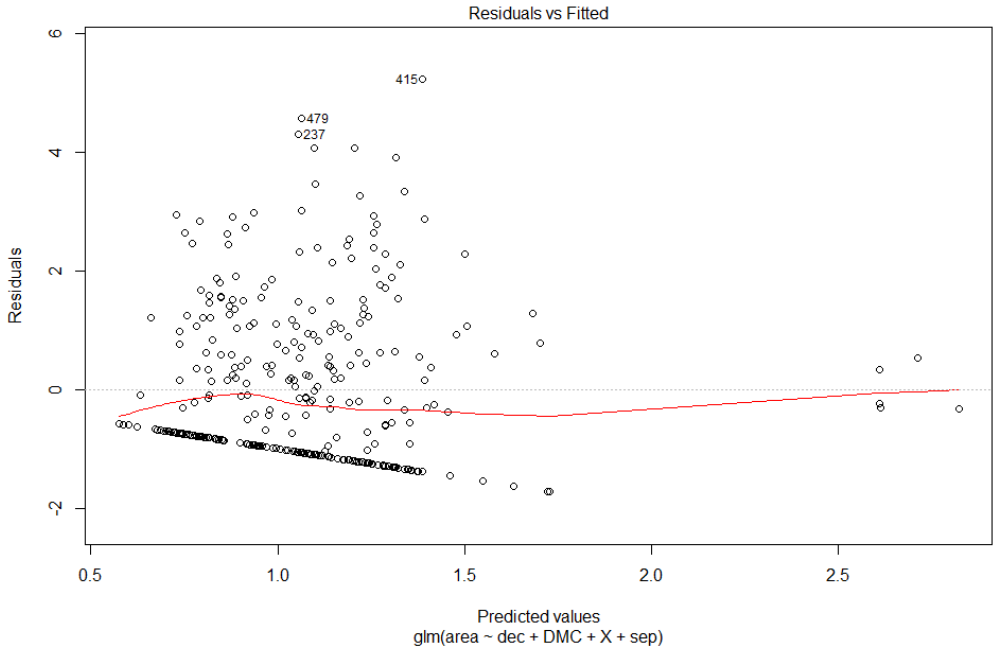
```
> summary(back.model)

Call:
glm(formula = area ~ X + DMC + temp + jan + feb + mar + apr +
    jun + jul + aug + sep + oct, data = data[train, ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5710  -1.0110  -0.5401   0.8511   5.0487

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.899020   0.553211   3.433 0.000673 ***
X            0.054504   0.031870   1.710 0.088166 .
DMC          0.002904   0.001616   1.797 0.073318 .
temp         0.023329   0.016461   1.417 0.157368
jan         -2.191979   1.084102  -2.022 0.043988 *
feb         -1.580064   0.654024  -2.416 0.016237 *
mar         -1.757156   0.562279  -3.125 0.001935 **
apr         -1.927354   0.850269  -2.267 0.024050 *
jun         -2.450821   0.682641  -3.590 0.000380 ***
jul         -2.018529   0.637968  -3.164 0.001700 **
aug         -2.054782   0.609288  -3.372 0.000833 ***
sep         -1.745756   0.585178  -2.983 0.003064 **
oct         -1.417793   0.700605  -2.024 0.043807 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.815776)

    Null deviance: 642.20  on 343  degrees of freedom
Residual deviance: 601.02  on 331  degrees of freedom
AIC: 1196.2
```

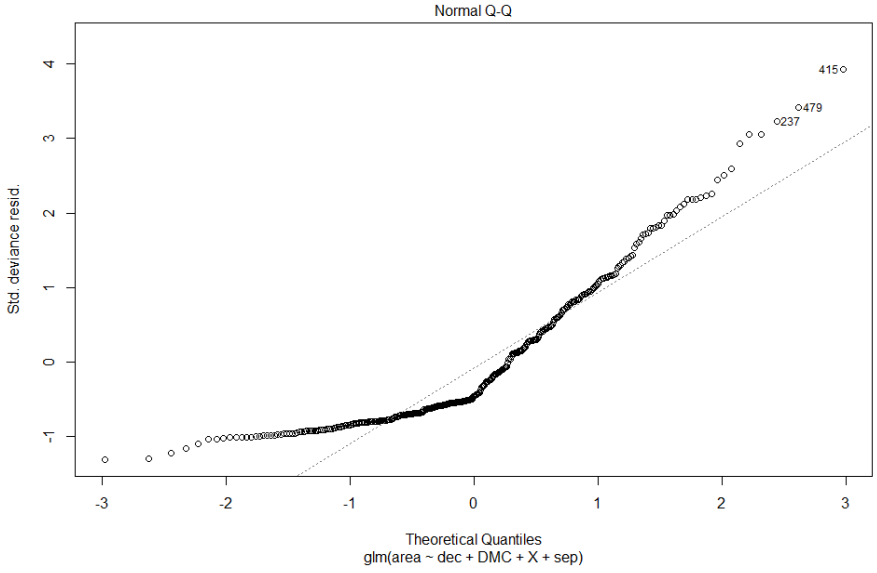
The MSE for the forward model is 2.067513, which is large for the magnitude of area values. The MSE for the backward model is 2.134971, which is also large. In this case, we want to lower the variance of our model. To do this, we use leave-one-out cross validation. This cross-validation uses the entire original dataset. One of the observations is not included in building the linear model, and is tested using the model made by the remaining observations. The final MSE is the average of the MSE’s found using this method.

We implement this cross validation on the whole data set to see which model is better. We try cross validation on both backwards and forwards models. What we get for the MSE is 1.904671 for the forward, and 1.946604 for the backward. In this case, the MSE of the forward model is slightly better. The MSE value is still rather large, as they are not too different from the MSE of the raw forwards and backwards models. This suggests

that either our variables are not good at predicting the area, or a linear model is not good for estimating area.

First, we look at why the linear model is not good for fitting the data. Recall that half of our observations have value for area equal to zero. This already suggests that the data can be separated into two groups instead of fitting under a single linear model. In fact, we saw previously that area is greatly skewed right. When we look at the residual plot for the forward model, we can see a distinct negative linear pattern. This means that the error in our model is not randomly distributed and does not have an expected value of zero. From the QQ plot, the error is also not normally distributed.



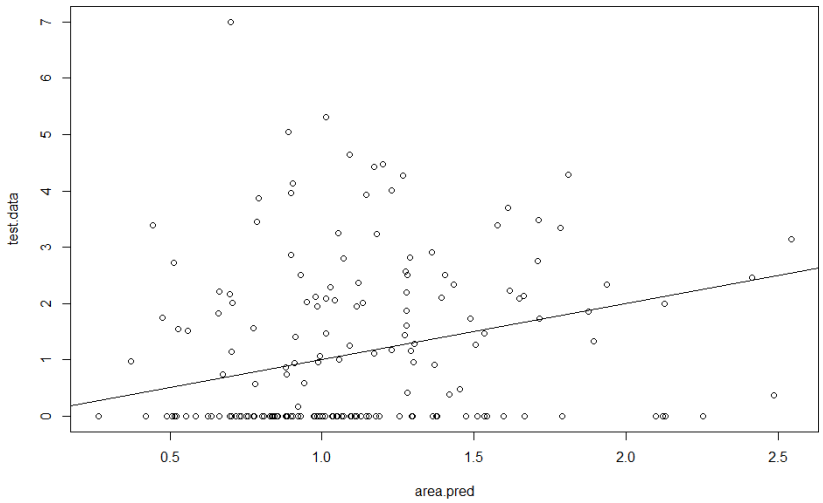


From this analysis, we decided that it is best not to use a linear model to fit our data. However, this process helped us to see which variables may be significant in the next methods that we use. For example, dec, DMC, and X showed up in both the forwards and backwards model. Therefore, we expect to see these variables be used again in later methods. To check this, we implement some methods to try and find out which variables are important in predicting area. We will use this information to build models that do not require linearity.

Variable Importance

A way of checking which variables are important in the model is to look at variable importance using random forest on the entire data set. The reason we want to use random forest is because some of the predictor variables may be correlated by context. For example, temperature may be higher in the summer months, so temp may be correlated with jun or jul. Therefore, using random forest can create a tree model that reduces correlation between variables. This will give us a better model to predict area.

Random forest happens by randomly choosing a number of predictors (6, which is the square root of the number of all our predictors, including dummy variables) to build a regression tree on. A way of splitting the tree to achieve more nodal purity in one of these trees can be added onto the previous tree as a way of growing the tree model. By choosing a small number of random predictors, this method decreases the chance of using predictors that are correlated. The predictions made by using the tree is compared with the actual values of area burned in the graph below. There is also a y=x line to show how closely the predicted values match with the actual values of area. If the tree model is good at predicting the area values, then the plotted points should lie along the y=x line.

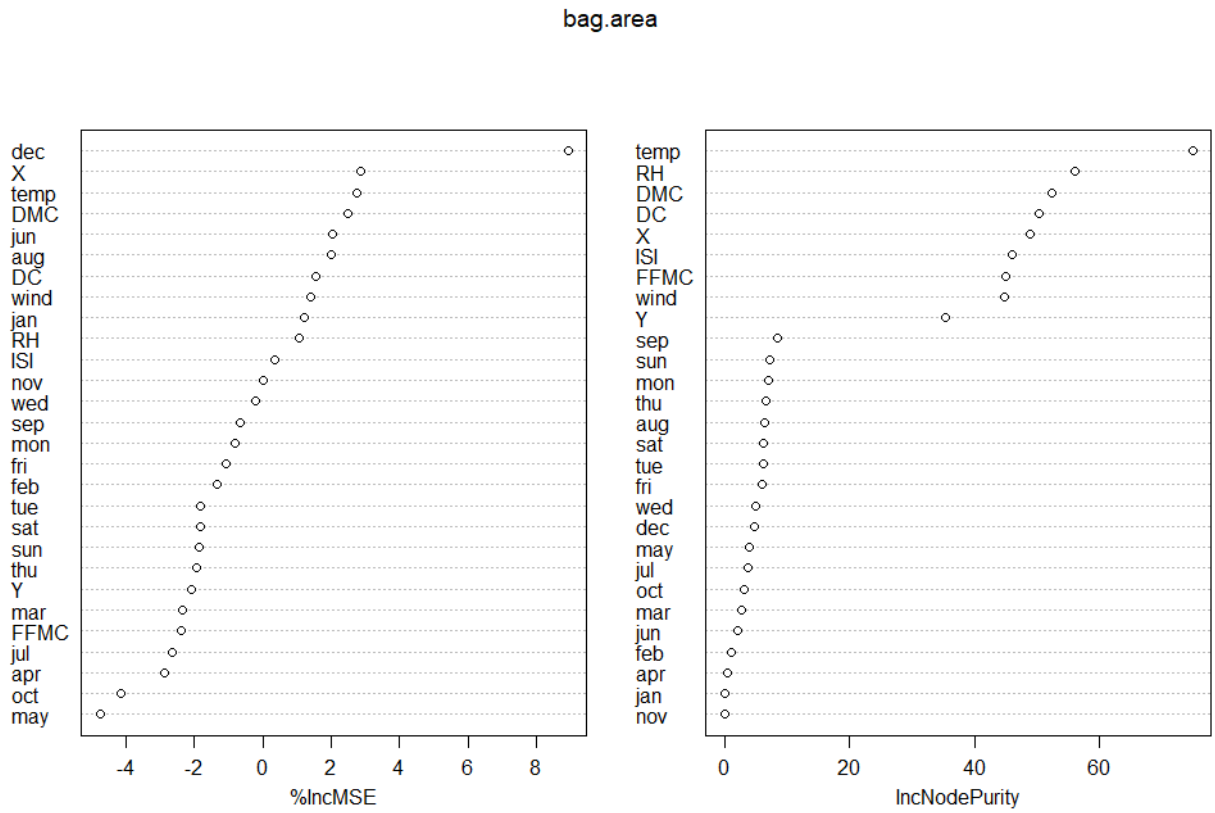


We see that the points are not resembling the y=x line. This regression tree is probably not good for predicting the actual values of area burned. In fact, the MSE found using

this regression tree was 2.122, which is not any better than what we found using the linear models. We also notice a straight line of points for all the observations that have area=0. Whereas these points do not follow the y=x line, the other points are actually dispersed (not closely) around the line. So, the many area=0 observations may have something to do with the bad random forest model.

The next thing we can do with this random forest model is to see which variables are important. We can measure the change in MSE when we take a predictor variable and randomly permute the values for that variable only. The magnitude of how much the MSE changes because of these permutations will show how important that variable is in splitting the regression tree. If the magnitude is not too big, then the variable is not that important. Similarly, we can measure the Gini index (nodal purity) of the tree after splitting by a variable. If the index drops a lot, then the variable is important for splitting the tree. We will use these measures of variable importance since our previous methods of finding significant variables seemed to yield the same results. This method will hopefully give us more ideas as to what variables to include in our final model.

In the graphs below, permuting the values for dec lead to the biggest changes in MSE. As for nodal purity, temp is the most important variable. RH, DMC, DC, X, ISI, FFMC^13, wind, and Y are other important variables when splitting the regression tree. It may be useful to build another model based only on these variables. It is also interesting that only permutation importance lists a dummy variable as a important variable while nodal purity uses all meteorological variables.



The ten variables that we chose as important from these plots are X, Y, FFMC, DMC, DC, ISI, temp, RH, wind, and dec. Notice that only one of these ten predictor variables is a dummy variable. This is already different from the previous linear models that used a lot of dummy variables. Fire sizes should not be largely dependent on which month the fire occurred in, but rather the environmental factors at the moment of the fire. Therefore, choosing these variables as important makes sense in the context of what we are trying to find. These ten variables will be used later in K-nearest-neighbor analysis.

Clustering Methods

Since our linear models did not work very well above, we will try to classify the area burned with clustering methods instead. To do this, we first reverse the transformations done to the variables in order to get the raw data. We then group the area values into 2 categories: “small” for fires that burned zero area, and “big” for fires that burned more than zero area. Recall from the earlier discussion that zero area does not mean that no area was burned.

The first clustering method that we used is the support vector machine. In particular, we used the support vector machine with radial kernel. The process for constructing this

SVM starts with plotting the observations from the training set and drawing a circle that separates a majority of the "small" fires from the "big" fires. Cost was set to one, meaning that any fire that was on the wrong side of the circle, or was inside the margins of the circle, would inflict a penalty of one on the overall model. Thus, the SVM created by this method has the lowest total penalty in the training set. Next, the model was used on the testing set and the predicted classifications were compared with the actual classifications. We used radial kernel instead of polynomial or linear kernel because the radial kernel gave us the lowest misclassification rate of about 40% instead of 47% and 41%, respectively. The difference between the radial kernel and the linear kernel is not very large, but we still choose to use the radial kernel because from our previous analyses, it appeared that the predictors were not very useful for classifying the fires. Therefore, "small" fires seem to be have a wide range of predictor values. So separating big and small fires by a line did not seem to fit this note. The details about the SVM are listed below:

```
Call:
svm(formula = area ~ ., data = clus[train, ], kernel = "radial",
    scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.03571429

Number of Support Vectors:  341
```

```
       pred
true    big small
  big    73    19
  small  50    30
```

A 40% misclassification rate means that 60% of the fires were correctly classified as "big" or "small" based on which side of the circle they were plotted on. This percentage is acceptable although not very high. In the worst case, it will predict 19 big fire as small one. Anyway, it is more than 50%, which is the expected classification rate for randomly guessing the size of a fire.

Another method of classification is the K-Nearest-Neighbors classification method.This plots an observation onto a space, and classifies it with the classification that is most prominent amongst the nearest k observations around it. The new observation is classified with a certain classification only if more than half of the k neighbors is that

classification. In this way, there is no need to split up the data with any lines or kernels. We use this method in hopes that not using an explicit model will help increase the rate of classifying a fire correctly.

Using the variables that we found were important due to permutation or nodal purity, we set up a knn function using two of ten variables. Then, we chose to look at 1 to 20 nearest neighbors (setting up K as 1 to 20). For each value of how many neighbors to choose, we looked at which combination of variables produced the greatest percentage of correct classifications. However, we know that KNN does not produce the same results every time that it is run. So we ran all of the combinations five times and got the following results:

```
[1] 0.622093
[1] 124
[1] 0.6337209
[1] 123
[1] 0.6162791
[1] 124
[1] 0.6162791
[1] 123
[1] 0.6162791
[1] 37
```

The decimal numbers are the rates for correct classifications whereas the integer number denotes which combination of variables resulted in that rate. For all simulations, the correct classification rate is about 60%, which is consistent with what we found in the SVM. The particular combination of predictors vary for each simulation. Therefore, we choose not to use the results from KNN to determine our classification model for two reasons: First, the rates are not so different from SVM, so KNN is not much better than SVM. Second, KNN uses only two predictor variables each time while SVM uses many more variables. In this sense, SVM can paint a better picture of forest fires than KNN. Therefore, the results from KNN serve to affirm the 60% correct classification rate and the idea that we should use clustering methods over linear or tree models.

Conclusion

In this project, we ran a variety of machine learning methods on our data including multiple linear regression, random forest, SVM, KNN, etc. We started with linear regression only because linear model is easy to interpret. Given the highly skewed distribution of the response variable we actually don't expect that a linear model will

lead to a good fit. The resulted high MSE made us conduct variable selection on the predictor variables. However, they all ended up with similar MSE. We followed up with a residual test that confirmed our previous expectation that the linear assumption is not satisfied by showing us a non-normally distributed residual. Hoping that a nonlinear model may have better prediction power, we then applied the method of regression tree in random forest because we thought that the MSE was affected by many correlated variables. That test again turned us down by giving us a MSE that is not any better than what we got from the linear model.

The above frustration may be unavoidable, it may be impossible to get a good numerical prediction of fire area based on the data. So we then turned more attention to those predictor variables to see if we can get a good classification of the responsible variable “area” using SVM or KNN. Before that we first ran an importance test to get a group of important variables and re-classified the area variable as big (area larger than 0) and small (area equal 0). We applied this criteria just because we would otherwise make our model too complex. Both SVM and KNN gave us similar correct classification rate at 60% which is acceptable although not very good since at least it is better than randomly guessing.

In sum, we know that linear models are not very good for our data. Other numerical methods gave us MSE that was similar to the MSE of linear models, so we decided to use classification methods instead. We ended up with similar models from SVM and KNN that were slightly better than randomly guessing the classifications of area burned by fire. We learned that building predictive models on real data is not easy or straightforward, and does not always work out nicely. However, there are always more methods to try to make the model better at predicting.

Examples of future work would be the use of polynomial regression to see if it can return a lower MSE rate. Or we may try to make our SVM model a little bit more complex to see if that will lead to a higher classification rate. Of course we also want to try the more advanced neural networks approach that we do not fully covered in this course.