Final Project

Alex Hyman

11/19/2018

## Introduction

The United States Forest Service (USFS) administers the forest and grasslands of the United States, which spans more than 193 million acres. One data point the USFS regularly maintains is the type of forest cover of a particular parcel of land. One way to classify the forest cover type of a particular parcel would be to visit every parcel of land, and manually determine what kind of cover type the forest maintains. However, this method is not feasible, considering the enormous span of the USFS responsibilities.

Another way the USFS has explored characterizing forest cover types is with machine learning algorithms using cartographic data acquired from satellites and historical United States Geological Survey (USGS) data. The USFS has already explored and labeled the forest cover type of 30 meter x 30 meter parcels of land in the Roosevelt National Forest in northern Colorado, and have now provided the cartographic data for each of these parcels. Using the provided cartographic data along with the labeled forest cover types, I will experiment to see if it is possible to train a model to correctly classify a parcel of land based on the cartographical and geological attributes provided. Three different styles of models will be used to conduct the supervised classification: tree-based models, distance based models, and probabilistic models.

## Initial Preprocessing

The USFS dataset provided contained cartographic and geological data that included the soil type (40 different types), elevation, slope, slope aspect, horizontal distance to roadways, horizontal distance to fire points, horizontal and vertical distance to hydrology, hillshade, and which wilderness area of the parcel for 15,120 different parcels.

Along with the data provded by the USFS, I decided to engineer three different features to the dataset. The first two features created were the sine and the cosine of the slope aspect. These features will provide a more intuitive range for the direction north/south, east/west the slope faces on a scale of -1 to 1.

The second feature I created utilized the vertical distance of the parcel to hydrology and the horizontal distance of the parcel to hydrology to calculate the euclidean distance of the parcel from the hydrology. This feature provides the machine learning algorithms the Euclidean distance of the parcel from a hydorlogical feature, which could provide insight into how wet or dry is a parcel of land.

## Number of rows missing data: 0

## [1] "Training set dimensions: "

## [1] 12096 58

## [1] "Test set dimensions: "

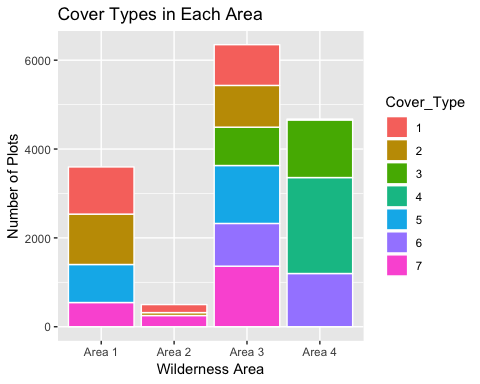
## [1] 3024 58

## [1] "Distribution of Train Cover Types "  
## 1 2 3 4 5 6 7   
## 1728 1728 1728 1728 1728 1728 1728

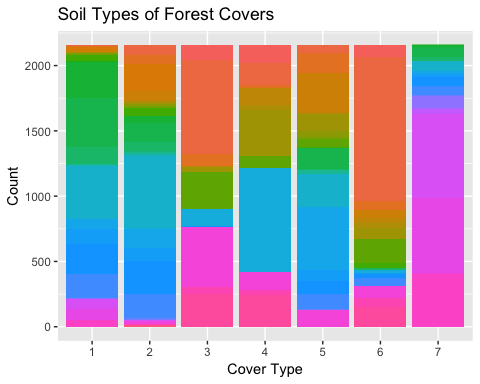
## [1] "Distribution of Test Cover Types"  
## 1 2 3 4 5 6 7   
## 432 432 432 432 432 432 432

# Exploratory Data Analysis

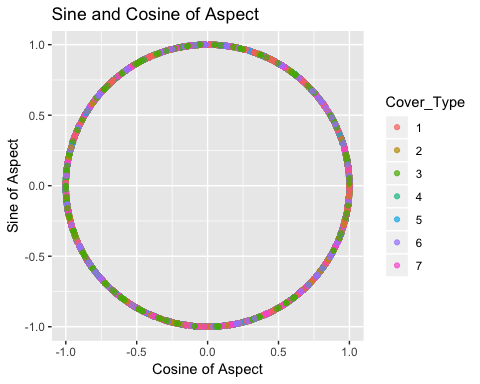
Before classification models were trained to predict the forest cover type, I wanted to see how some of the data was distributed, and to see if there were any trends or interesting patterns in the labeled data. The first pattern I wanted to explore was: how was the cover types distributed between the four wilderness areas? A bar chart was created to show the number of parcels included in the training data from each wilderness area, and the forest cover type of each parcel was separated by color. The plot showed that most of the parcels came from wilderness areas 1, 3, and 4, and that each of the forest cover types were generally spread mostly between two of the four wilderness areas. A distrubution matrix further showed that no one of the wilderness areas has all seven of the forest cover types. It is important to note that if this distribution is not representative of the actual distrubution of cover types in the different wilderness areas, some modeling algorithms would perform significantly worse on new instances if the algorithm finds that wilderness area is an important variable in determining forest cover type.



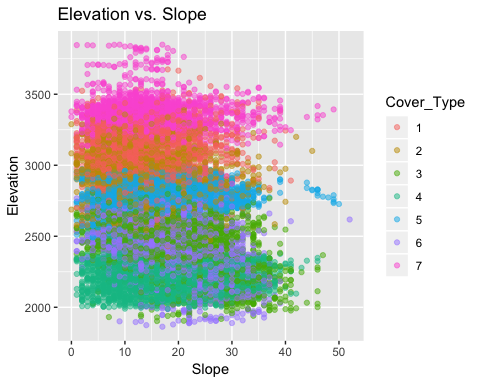
The next variable that I wanted to explore to see if there was a relationship with the forest cover of the parcel was the soil type of the parcel. If a cover type could only grow in a few soil types, it could make sense to group or eliminate certain soil types from the analysis. When looking at the graphic, it appears that a lot of soils are common across all of the forest cover types, but there are some forest cover types that have larger chunks of certain soils in their forest cover type than others. While there are no obvious trends in the soil types of the different forest covers, there could be some information that is quite deciperable by an algorithm, but not the human eye. This graphic also shows that there is an equal amount of each forest cover type in the dataset, meaning the labels are balanced.



A third figure was created to see if there were any trends between the aspect of the parcel and the forest cover type; or, if certain forest cover types can only grow in certain directions. Because the the sine and the cosine of the aspect has a range from 0 to 360 degrees, the plot is circular with a magnitude of 1. The different cover types were shown in different color around the circle. The plot did not show any significant patterns, but it could be an important variable along with other variables.



The final variables that I explored to see if they had a significant effect on the forest cover type of a particular parcel of forest were elevation and slope. To explore this relationship, I plotted a slope vs. elevation scatterplot with the color of the points showing the forest cover type. The graphic showed that for the most part, the different forest cover types can occur at any range in the slopes; however, the graphic does show that elevation has a fairly significant effect in which forest cover types can occur. Specifically, the graphic shows that Ponderosa Pine, Douglas-fir, and Cottonwood/Willow cover types occur in the lower elevations, and Krummholz, Lodgepole Pine, and Spruce/Fir cover types occur in the higher elevations.



# Modeling

While the EDA provided some insight into the behavior of some variables and how they can affect the forest cover type of a given parcel, a supervised learning model trained to classify the forest type from the provided and created featured can prove to be more helpful.

Three different styles of learning algorithms test to for performance were tree-based, distance-based, and probability-based. Each of the algorithms used the same features when training the model, but certain preocessing steps were taken for certain models that were not required for other models. To train the models, 80% of the labeled data was used as the training set, and the remaining 20% of the labeled data was used as the test set. The models were evaluated for performance by the accuracy of the model on the test set.

## Tree-based Algorithms

The first of the model styles used to predict the forest cover types of the parcels of land were the tree-based models. Of the tree-based models, I chose to train a simple decision tree and a random forest model.

### Decision Tree

The first tree-based model trained to classify the different cover types was a decision tree. The caret package was used to train multiple decision trees at complexities of 0.01 through 0.00005 to find the best tuned model with a three-fold cross validation. The package chose the decision tree with a complexity of 0.0005 as the best tuned model with a three-fold cross validation accuracy of 0.753. With this best tuned model, the decision tree algorithm was able to score an accuracy 0.751 on the test set. This wold suggest that decision tree model did not overfit the data used to train the model, as the test accuracy was relatively close to the test accuracy.

A glance at the confusion matrix of the decision tree algorithm shows that the model was able to classify Cottonwood/Willow, Aspen, and Krummholz quite well, but often misclassified the other four forest cover types.

## Decision tree has an accuracy of 0.7678571 on hold out set

## Reference  
## Prediction 1 2 3 4 5 6 7  
## 1 298 92 0 0 3 0 39  
## 2 84 242 12 0 36 6 4  
## 3 0 13 307 17 14 80 0  
## 4 0 0 19 398 0 23 0  
## 5 12 55 8 0 376 11 0  
## 6 3 22 86 17 3 312 0  
## 7 35 8 0 0 0 0 389

### Random Forest

A random forest was the second tree-based model trained to classify the forest cover types. The random forest method within the caret package only has on tuning parameter, which is the number of features to sample at each split. The optimal mtry parameter chosen by the model was picked to be 29, which had a threefold cross-validation accuracy of 0.842. The best-tuned random forest model was then used to evaluate on the test set, and an accuracy of 0.857 was achieved.

Once again, a glance at the confusion matrix shows that the random forest model performed significantly better on the Spruce/Fir, Ponderosa Pine, and Douglas-fir classes than the decision tree was able to, but also continue to classify the Cottonwood/Willow, Aspen, and Krummholz classes well.

Additionally, the variable importance of the features in the training set were checked to see which variables the random forest model found to be the most important when classifying which forest cover type. The variable importance function showed that the model found elevation to be the most important variable when choosing a class for an instance. This variable makes sense as the most important variable in the classification task, given the forest cover type layers that appeared on the slope vs. elevation graph.

The most important variable function also seemed to show that generally, continuous variables were more important than categorical or binary variables, with Widerness Area 4 being the only categorical variable ranked in the top 20 of the most important variables. The likely reason for Wilderness Area 4 being ranked as an important variable is because Cottonwood/Willow only occur in Wilderness Area 4, which could mean the importance of Wilderness Area 4 is only important when trying to classify the Cottonwood/Willow forest cover types.

## Random Forest has an accuracy of 0.8578042 on hold out set

## Reference  
## Prediction 1 2 3 4 5 6 7  
## 1 327 78 0 0 3 0 14  
## 2 66 297 3 0 10 3 0  
## 3 1 10 354 8 4 42 0  
## 4 0 0 16 420 0 15 0  
## 5 9 30 6 0 412 6 0  
## 6 1 13 53 4 3 366 0  
## 7 28 4 0 0 0 0 418

## rf variable importance

Overall  
## Elevation 100.000  
## Horizontal\_Distance\_To\_Roadways 26.046  
## Horizontal\_Distance\_To\_Fire\_Points 19.880  
## Euclidean\_Distance\_to\_Hydrology 16.190  
## Hillshade\_9am 13.457  
## Wilderness\_Area4 12.009  
## Vertical\_Distance\_To\_Hydrology 10.934  
## Horizontal\_Distance\_To\_Hydrology 10.777  
## Aspect 10.750  
## Hillshade\_Noon 10.048

## Distance-Based Algorithms

The second class of supervised learning algorithms used were distance-based supervised learning algorithms. The distance-based leaning algorithms used in this training included Support Vector Machines (SVM) and K-Nearest Neightbor (KNN).

Because the evaluations of these algorithms utilize distance (in this case euclidean distance) from either another point or from a support vector, the continous variables in the training set needed to be scaled so features that have higher magnitudes (elevation) are on the same scale as variables with a small scale (sinAspect).

Additionally, the test data needed to be scaled, but because these variables are meant to represent unseen data, the continuous testing data was centered and scaled with the mean and standard deviations found from the continuous variables in the training set.

### SVM

SVMs were the second distance based learning algorithm used to classify the forest cover types. The first SVM modles that were trained were SVMs with a linear kernel within the caret package. The only tuning parameter for SVMs with a linear kernel was cost. The linear SVM was trained with cost values ranging from 1 to 0.1 and once again, three-fold cross validation was used to determine which hyper-parameter produced the most accutate model. After all the models were evaluated, caret determined that the best SVM with a linear kernel was the model with the cost value of 1, which had a three-fold cross validation accuracy of 0.714.

After the SVMs with a linear kernel were trained, more SVM models were trained with a polynomial kernel. The SVMs with polynomial kernels have three different hyperparameter: cost, scale, and degree. To choose the best parameters for the SVM with the polynomial kernel, created a grid to test the best cost, scale, and shape parameters. After the gid search was complete, the caret package chose the SVM with a polynomal kernel to have a cost of 0.8, a scale of 0.2, and a degree of 3. This model resulted in a thee-fold cross validation accuracy of 0.783.

Because the SVM model with the polynomial kernel resulted in the higher accuracy, it was used to predict the test set. The model resulted in a testing accuracy of 0.810, and performed fairly high in predicting all of the forest cover types except for the Lodgepole Pine cover, which none of the classifier have performed too well on. While the 0.810 testing accuracy scored fairly high, the random forest model scored a bit higher with a testing accuracy of 0.857.

## SVM with Polynomial Kernel has an accuracy of 0.8095238 on hold out set

## [1] "SVM (Polynomial Kernel) Confusion Matrix\n"

## Reference  
## Prediction 1 2 3 4 5 6 7  
## 1 321 96 0 0 4 0 19  
## 2 69 278 3 0 23 4 0  
## 3 1 8 310 15 8 79 0  
## 4 0 0 25 411 0 21 0  
## 5 11 38 8 0 394 7 0  
## 6 0 9 86 6 3 321 0  
## 7 30 3 0 0 0 0 413

### KNN

The next distance based learning algorithm trained to predict the the forest cover types was the K Nearest Neighbor algorithm. The most important hyperparameter when training the a KNN model is the k-value, or how many of the number of the closest points voting on which class the unlabeld point belongs to. Once again, the caret package was used to train the KNN models with K-values from 1 to 25, and three-fold cross validation was used to determine which model performed the best. After all the models were trained and evaluated, it was determined that the best model had a K-value of 1, which had a three-fold cross validation accuracy of 0.690. When the model was used to predict the class of the test set, it was able to acheive a test accuracy of 0.725.

## KNN w/ k = 1 has an accuracy of 0.7251984 on hold out set

## Reference  
## Prediction 1 2 3 4 5 6 7  
## 1 247 94 0 0 3 2 28  
## 2 98 240 13 0 32 8 10  
## 3 2 13 264 23 8 93 0  
## 4 0 1 46 380 0 26 0  
## 5 29 60 15 0 378 10 3  
## 6 5 11 94 29 10 293 0  
## 7 51 13 0 0 1 0 391

## Probabilistic Classifier

The final class of supervised learning algorithms used to predict forest cover types was probabilistic classifier, or classifiers that are able to predict the probability of an instance using historical probability distributions for a set or a class. The only probabilistic classifier used for this experiment was the Naïve Bayes algorithm. The algorithm used the scaled continuous features created for the distance algorithm, but the binary numerical data was converted into factors to so the algorithm would not attempt to confuse the 0 or 1 binary data into actual numerical data, where the distribution would be assumed to be normal.

### Naive Bayes

The klaR package was utilized to train a Naïve Bayes model to fit the training data. Two main hyper parameters were considered when determining the best tune for the model, the whether to usage of the kernel density estimate (KDE) and the Laplace correction factor to prevent 0 probability classes in the predictions. Two models were first trained with the same Laplace factor of 1, but one was trained with the kernel and one was trained without. The model that used the KDE achieved the higher accuracy of the two with a test accuracy of 0.698 versus a test accuracy of 0.651 in the model that did not use the KDE.

After the model with that used the KDE was determined to perform better than the model without it, I trained two more models that used the KDE, one with a Laplace correction factor of 2 and another with a Laplace correction factor of 5. After testing these models, it showed that the Naïve Bayes model that used the KDE and had a Laplace correction factor of 1 was the best scoring model still.

## Naive Bayes w/ Normal Distribution and fL = 1 has an accuracy of 0.651455

## Naive Bayes w/ KDE and fL = 1 has an accuracy of 0.698082 on hold out set

## Reference  
## Prediction 1 2 3 4 5 6 7  
## 1 276 104 0 0 13 0 38  
## 2 74 233 4 0 53 7 6  
## 3 1 4 232 30 8 87 0  
## 4 0 0 54 395 0 40 0  
## 5 17 64 40 0 329 40 0  
## 6 3 19 102 7 29 258 0  
## 7 61 8 0 0 0 0 388

## Naive Bayes w/ KDE and fL = 2 has an accuracy of 0.6974206 on hold out set

## Naive Bayes w/ KDE and fL = 5 has an accuracy of 0.6964286 on hold out set

## Conclusion

After using supervised learning to train models to classify forest cover types, I have determined that the USFS could use machine learning to accurately classify a forest cover type based on cartographic and geological features.

While a perfect classifier could not be created, a random forest classifier was able to classify 85% of the hold out set. If more features could be manufactured or collected from new technologies, it could be possible to create an even better performing classifier. Another benefit of the random forest classifier is that it was relatively quick to train and predict the forest cover types of the various parcels. Below is a summary of the performance for the best tuned model of all the algorithms tested.

|  |  |  |
| --- | --- | --- |
| **Model** | **Hyperparameters** | **Test Accuracy** |
| **Tree-Based Models** | | |
| Decision Tree | Complexity = 0.0005 | 0.768 |
| Random Forest | NONE | 0.858 |
| **Distance-Based Models** | | |
| SVM | Kernel = Polynomial, Scale = 0.2, Degree =3 , Cost = 0.8 | 0.810 |
| KNN | K = 1 | 0.725 |
| **Probabilistic Models** | | |
| Naïve Bayes | useKernel = True, fL = 1 | 0.698 |