Phase 2

Forest Cover Type Prediction

Group 2 - Kristin Wendell and Erica Calvi

Karst: QTM6300-01

Due December 7th, 2021

Background

The goal of our project was to predict the forest cover type in four different wilderness areas in the Roosevelt National Forest of northern Colorado. The independent cartographic variables considered included numerical variables such as elevation and aspect in azimuth as well as categorical variables such as soil type. For a description of each variable in our dataset, please see Appendix A. Each cover type was represented equally in our dataset with each representing approximately 14% of the data.

We chose to model our data with three different model types: KNN analysis, Classification Tree, and Naïve Bayesian Classification. Before analysis, we split the data into two sets. 60% of the data was used for training the models and 40% was reserved for testing the models. The larger training dataset is used for the models to learn how to predict cover types using the predictor variables present in the data. Once the learning process is over, the reserved smaller test dataset is fed to the models to determine how well the models predict cover type with new data. The quality of the model is evaluated based on the accuracy of the predictions.

Specifically, model performance is evaluated using error rates. The overall error rate is the percentage of correct predictions out of all predictions made on the test set. In order to evaluate how well a model predicts specific cover types, we look at two performance metrics: precision and recall. Precision looks at all instances where the model predicted a particular cover type and calculates the percentage of them that are correct. Recall looks at all the actual observations of a particular cover type and calculates the percentage of them that were correctly predicted.

Preprocessing

With any dataset, it's always standard practice to check the data for missing values or abnormalities. For our training data set, we found that there weren't any missing values present. We checked for outliers (unusually high or low values) within each of our variables and noticed that Horizontal Distance to Firepoints had many data points very far from the mean value. However upon closer inspection we determined that these outliers were not erroneous and kept them as is. When managing the data set in RStudio, we made sure to adjust all variables to their correct data types. During this process, we removed the ID column of the data set since it wasn't useful for our prediction model. We also removed two soil types (Gothic family and unspecified) because there weren't any data points with those soil types present in our training data set. Lastly, we noticed two sets of variables that seemed highly related to each other: the hillshade indexes, aspect and slope as the first set and distance to hydrology, both

vertical and horizontal as the second set. Given that strong relationships among variables can affect the performance of certain model types, we decided to perform another pre-processing step referred to as principal component analysis (PCA). PCA combines specified groupings of related variables, removing the relationship issue and ultimately reducing the number of variables while maintaining the level of predictive power they provide. This PCA preprocessing was applied to KNN and Naïve Bayes.

KNN Analysis

The first model we ran was KNN (k-nearest-neighbor). KNN works under the theory that "you look like your neighbors" - meaning similar objects tend to be grouped together. The KNN model estimates how likely a data point is to be a member of one group or another depending on what group the data points (neighbors) nearest to it are in. The "neighbors" then "vote" to determine the most likely classification for the data point in question.

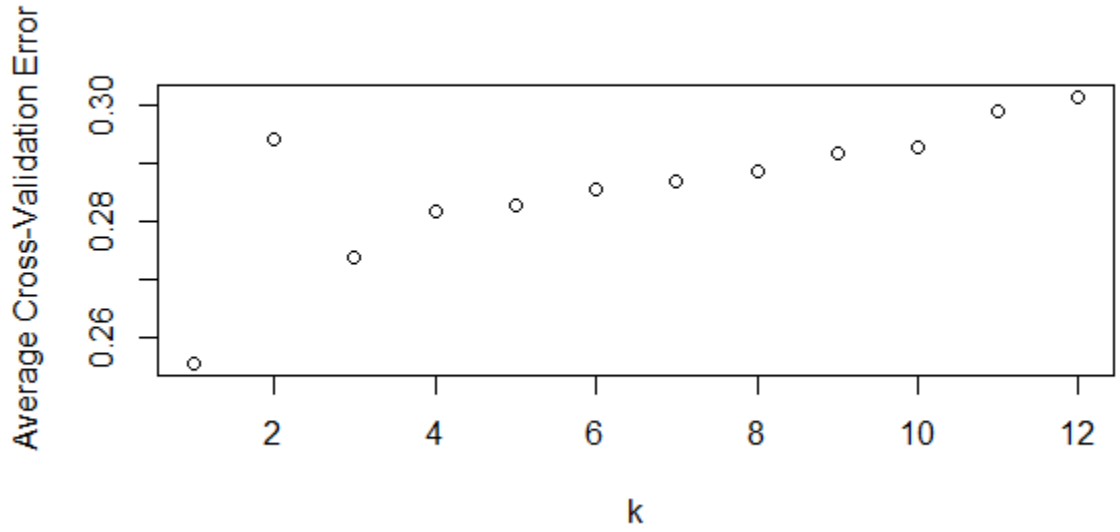
Before creating a KNN model we went through a series of pre-processing steps. An important limitation of the KNN model type is that it only accepts numerical variables as inputs given its usage of distance to classify. Given this limitation, we removed all non-numeric variables, such as wilderness areas and soil types. This allowed the model to ignore these pieces of the dataset for the time being.

Lastly, we had to perform an additional preprocessing step to account for the differing scales of our predictor variables. We put all of the variables on equal footing by defining an observation's attributes not in terms each variable's units (ie. elevation is 2,749 meters) but in terms of a common unit. The common unit used is a distance from the average (ie. elevation is equal to average elevation in the dataset). This is referred to as standardization. This is necessary since KNN uses a distance calculation to classify and variables with large scales would have an oversized influence on the model.

For KNN analysis, one must decide the best value of "k" to use. For example, if k = 3, it would mean that the 3 closest points (neighbors) to the observation in question will "vote" to determine its classification. When creating our KNN model, we used cross-validation to help determine what the optimal value of k should be for our data set. Cross validation allows us to try out different values of k to see which one will give us the lowest average error. Our cross validation method is 10-fold, which means we split the data into 10 groups and sequentially hold out one section as our test. This means that all parts of the data get a turn at being the test data set and the training data set. The errors at each level of complexity are then averaged across these groups. The

whole process helps us reduce variability in error and become more confident about how our model may respond with the true test dataset.

Please see below for a visualization of number of k values vs average error:



As you can see in the graph above, the average error rate for the knn model is lowest when using k = 1. This means that the single closest point to a point in question will determine the label of the given point.The error rate for this model is 24%, which means that the model can correctly predict the cover type of a data point 76% of the time. This is a great improvement over the benchmark error rate of 86% - which means if we were not to create a model at all, and we were simply to use the most common cover type per prediction (mode), we would be incorrect 86% of the time, or correct only 14% of the time. Benchmark error is useful for determining whether or not the error rate of a model is “good”.

When evaluating the performance of cover type predictions, we used recall as our metric in determining the success of the prediction. In the chart below, we can see that the model was able to successfully predict certain cover types more accurately than others. For example, Cottonwood/Willow, Aspen, and Krummholz have high recall rates of 93%, 91%, 93% , respectively. While cover types such as Spruce/Fir, Lodgepole Pine, and Ponderosa Pine have significantly lower recall rates of 60%, 56%, and 63% respectively. The cover types with lower recall rates tend to get confused with other cover types. For example, the predictions misclassify Spruce/Fir as Lodgepole Pine for 22% of the observations and Krummholz for about 13% of the observations.

KNN Analysis

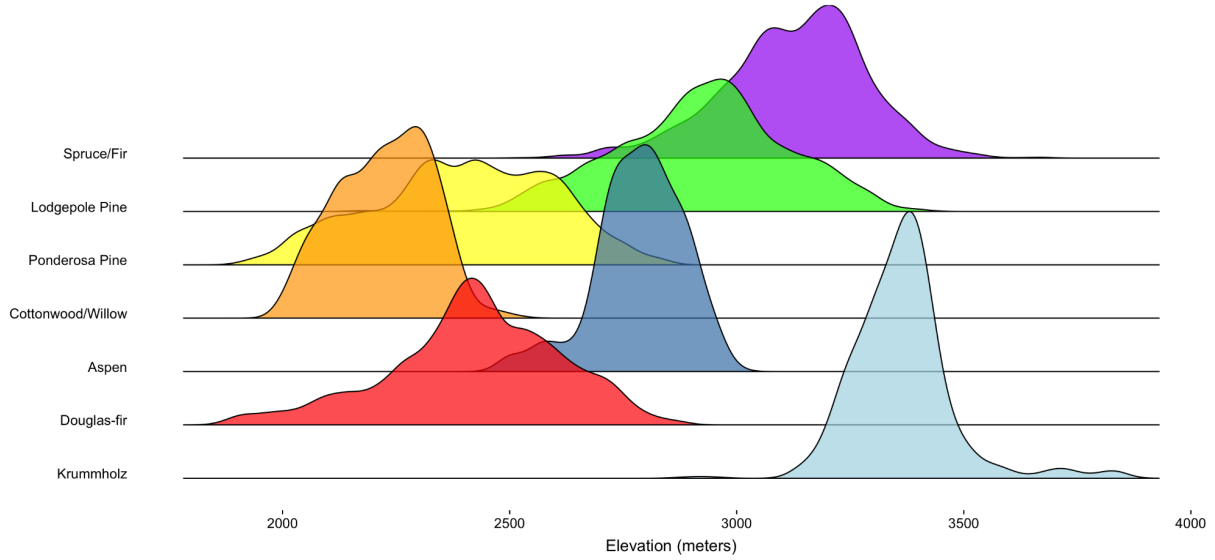
Observations

Predictions	Spruce/Fir	Lodgepole	Ponderosa	Cottonwood/ Willow	Aspen	Douglas-fir	Krummholz	*Total Predictions*	Precision
Spruce/Fir	515	175	1	0	11	2	47	*751*	**69%**
Lodgepole	185	473	23	0	39	20	13	*753*	**63%**
Ponderosa	2	34	545	28	15	136	0	*760*	**72%**
Cottonwood/Willow	0	1	70	808	0	52	0	*931*	**87%**
Aspen	37	100	18	0	782	19	0	*956*	**82%**
Douglas-fir	3	48	205	31	11	647	0	*945*	**68%**
Krummholz	110	20	0	0	0	0	822	*952*	**86%**
Total Observations	*852*	*851*	*862*	*867*	*858*	*876*	*882*		
Recall	**60%**	**56%**	**63%**	**93%**	**91%**	**74%**	**93%**		

Note: Recall represents % of observations that were correctly predicted

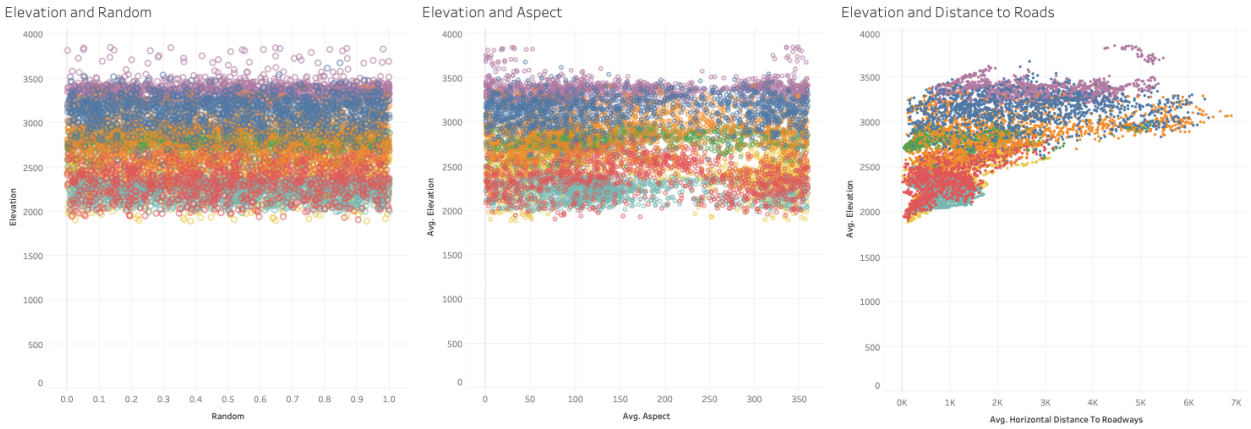
Note: Precision represents % predictions that were correct

Given the confusion between cover types, we chose to explore our variables more thoroughly using visualization. Looking at a density plot below of elevation colored by cover type, we can see that there is a clear separation between our cover types at specific elevations. This signals to us that elevation is an important variable in our data set. However, we also see a link between where our model is confused and where cover types overlap by elevation. This led us to investigate the explanatory power of the other variables in our model.



We chose to create a series of scatter plots with each individual variable (prior to PCA preprocessing) and elevation. We kept elevation throughout the graphs because it allows us to see how the scatter plot changes based on the explanatory power of the variable we are investigating in a consistent way. For example, in the chart on the left we see elevation plotted with a randomly generated number. We see a consistent band

of cover types spanning the range of the randomly generated number. We would expect the variables we are testing to look like this if they were not meaningful. In the middle, we see the variable aspect. While there may be some meaning in the variable, we question how much this variable is adding to KNN's predictive power in the model. Conversely, when looking at the variable horizontal distance to roadways, we see this is likely a variable that is contributing to the model in a meaningful way.



Using this methodology on all the variables in our model (see Appendix B), we concluded that slope and aspect are likely not providing a lot of meaningful connections in our model while distances to hydrology, roadways and fire points seem to be helpful. Hillshade and 9am and Noon seem to provide some explanatory power at the extremes. Overall, we conclude that elevation is largely driving the KNN model with varying levels of helpfulness coming from the other variables. Cover types in elevation ranges with lots of overlap are simply more likely to have higher error rates.

Naïve Bayesian Classification

The next model we implemented was Naïve Bayes. Naïve Bayes starts by considering the probability of encountering each of our predictor values given a particular cover type. We can refer back to the elevation density chart with a probability lens to understand this further. Naïve Bayes will see that an observation classified as Krummholz has a 54% probability of being between elevation 3,320 - 3,440 meters. The model calculates these probabilities for each of the predictor values then uses them as an input to classify new observations.

Probability of Elevation Range Given Cover Type

	Elevation (in meters)							
	1800 - 2000	2000 - 2400	2400 - 2640	2640 - 2929	2929 - 3160	3160 - 3320	3320 - 3440	3440 +
Spruce/Fir	-	-	-	9%	44%	38%	7%	
Lodgepole Pine	-	-	8%	40%	41%	10%	1%	-
Ponderosa Pine		45%	42%	11%	-	-	-	-
Cottonwood/Willow	-	98%	2%	-	-	-	-	-
Aspen	-	-	7%	87%	6%	-	-	-
Douglas-fir	3%	39%	46%	12%		-	-	-
Krummholz	-	-	-	-	1%	31%	54%	13%

Naïve Bayes only accepts categorical predictor values; therefore, the first step in building the model was to convert the numerical predictors into binned values. For example, the elevation value 3,325 meters was transformed into an elevation "3,320 - 3,440 meters".

In terms of performance, the overall error rate was 32%, reducing the error by 63% as compared to the benchmark but representing the worst of our models. Cottonwood/Willow and Krummholz were still accurately classified in over 90% of their observations. However, Lodgepole Pine, Ponderosa Pine and Douglas Fir observations were only correctly classified approximately 50% of the time.

Naïve Bayes

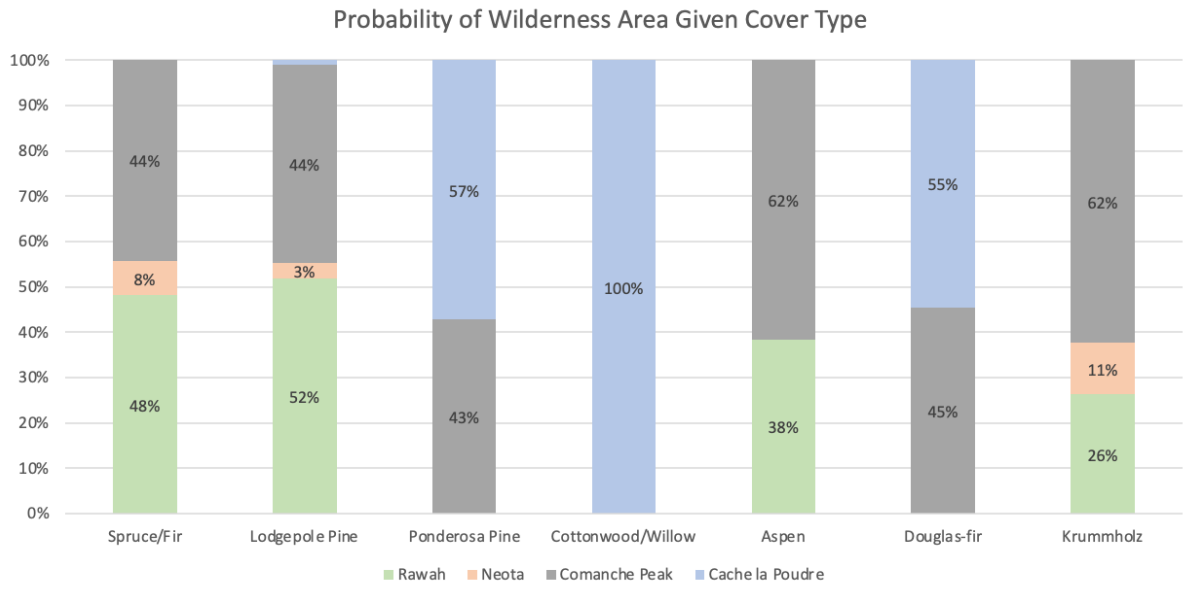
Observations

Predictions	Spruce/Fir	Lodgepole	Ponderosa	Cottonwood/ Willow	Aspen	Douglas-fir	Krummholz	*Total Predictions*	Precision
Spruce/Fir	511	231	0	0	22	0	77	*841*	**61%**
Lodgepole	180	415	1	0	119	11	5	*731*	**57%**
Ponderosa	0	15	431	47	18	184	0	*695*	**62%**
Cottonwood/Willow	0	0	149	805	0	97	0	*1051*	**77%**
Aspen	41	154	70	0	670	104	4	*1043*	**64%**
Douglas-fir	1	25	211	15	29	480	0	*761*	**63%**
Krummholz	119	11	0	0	0	0	796	*926*	**86%**
Total Observations	*852*	*851*	*862*	*867*	*858*	*876*	*882*		
Recall	**60%**	**49%**	**50%**	**93%**	**78%**	**55%**	**90%**		

Note: Recall represents % of observations that were correctly predicted

Note: Precision represents % predictions that were correct

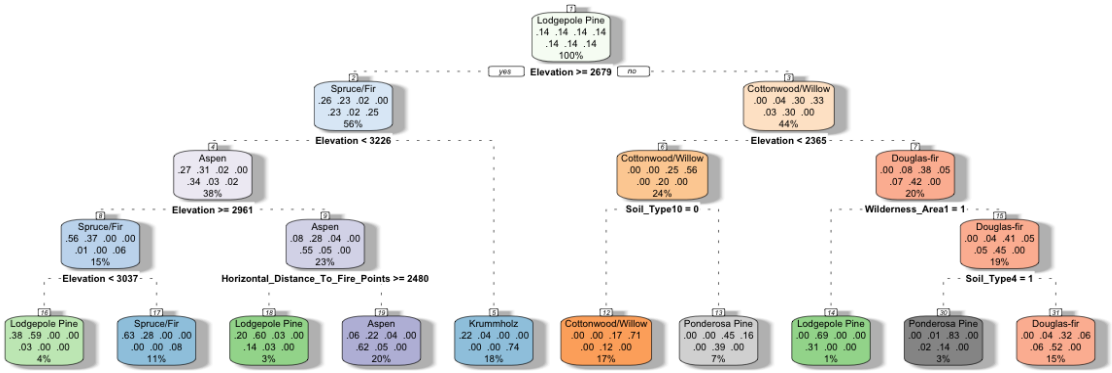
Since Naïve Bayes uses basic frequencies and probabilities, we can easily extract valuable descriptive insights about the dataset. Based on the chart below, we know Cottonwood/Willow only grows in Cache la Poudre Wilderness Area and Ponderosa Pine only grows in Cache la Poudre or Comanche Peak. Additionally, we see Spruce/Fir and Lodgepole Pine have similar probabilities in the same wilderness areas. The same can be said for Ponderosa Pine and Douglas-fir.



Other interesting insights were the wide ranges of distance to fire points and roadways for Spruce/Fir and Lodgepole Pine, particularly in comparison to Cottonwood/Willow (Appendix C) and the breakdown of soil type probabilities (Appendix D). As with wilderness areas, we see Spruce/Fir and Lodgepole Pine have similar composition and probabilities among soil types. We also see Ponderosa Pine and Douglas-Fir are likely to be in the Bullwak soil type (30% and 50%, respectively). Lastly, we see Krummholz is primarily in soils not common in other cover types.

Classification Tree

The next model we ran was a classification tree. Classification trees break down the population of data into subpopulations, or nodes, by a sequence of decisions with the goal of creating nodes with only one type of classification represented. This is similar to a decision tree. For example, we start with our entire dataset of cover types. The classification tree decided the best way to split the data is based on an elevation value of 2,679 meters. The observations where elevation is greater than this value move to the group to the left (blue) while those below it move to the right (orange). A few other decisions based on soil types, wilderness areas, and distances to fire points are made to split the data into nodes with increasing homogeneity. The final prediction nodes are found in the bottom row.



The tree above is a simple classification tree - additional splits are necessary in order to obtain the full prediction potential of our model. However, we will reach a point where adding additional splits actually decreases the quality of our model. At this point, the model would simply be describing the training data used to build the model and not finding real insights that could be generalized to predict accurately given new data. This is referred to as “overfitting”. The goal is to construct a tree that is large enough to incorporate all the valuable insights but not so large that it is overfit.

In order to accomplish this, we purposefully modeled an overfit tree then estimated the error expected with new data at varying numbers of splits. We removed the splits that were not adding predictive value in our tests, also known as pruning. This process helps us be confident that the insights discovered can be generalized to make strong predictions given new data.

In terms of performance, the overall error rate of our pruned model was 23%, reducing the error by 73% as compared to the benchmark. As we saw above, the first and most impactful split in our tree is based on an elevation value of 2,679 meters which divides our population into two groups, representing a 60:40 split of those at the higher elevations to the lower. Within the higher elevation node we can be highly confident the cover type will be either Lodgepole Pine, Spruce, Aspen and Krummholz with similar representation among them. Alternatively, we can be highly confident if the elevation is less than 2,679 meters, the cover type will be either Ponderosa Pine, Cottonwood and Douglas Fir. Please refer to the density plot of elevation in the KNN section to visualize the logic behind this split.

Below we can see the recall and precision by cover type. We see a clear performance spectrum: at the high performance end there is Krummholz, Aspen and Cottonwood/Willow (recall above 90%), at the low end there is Spruce/Fir, Lodgepole Pine and Ponderosa Pine (recall less than 66%) and in the middle there is Douglas-fir at

76% recall. Generally, precision seems in line with recall, indicating strong classification performance is not being achieved by over-classifying some cover types at the expense of others. However, it's worth noting those at the lower end of the performance spectrum have precision slightly higher than recall while at the higher end of the performance spectrum have slightly lower precision than recall.

Classification Trees

Observations

Predictions	Spruce/Fir	Lodgepole	Ponderosa	Cottonwood /Willow	Aspen	Douglas-fir	Krummholz	*Total Predictions*	Precision
Spruce/Fir	549	199	0	0	9	0	66	*823*	**67%**
Lodgepole	180	496	5	0	47	13	3	*744*	**67%**
Ponderosa	0	20	567	52	20	130	1	*790*	**72%**
Cottonwood/Willow	0	0	57	800	0	51	0	*908*	**88%**
Aspen	17	93	33	0	771	17	1	*932*	**83%**
Douglas-fir	5	31	200	15	11	665	0	*927*	**72%**
Krummholz	101	12	0	0	0	0	811	*924*	**88%**
TOTAL	*852*	*851*	*862*	*867*	*858*	*876*	*882*		
Recall	**64%**	**58%**	**66%**	**92%**	**90%**	**76%**	**92%**		

Note: Recall represents % of observations that were correctly predicted

Note: Precision represents % predictions that were correct

The model accurately classified over 90% of observations having cover types Cottonwood, Aspen and Krummholz. The high recall for Krummholz is largely because it makes up the majority of cover types represented in its elevation band. After only three splits on elevation we see a node where 87% of the subpopulation is Krummholz, representing 12% of the total data. Conversely, after the initial elevation splits, both Cottonwood and Aspen are in nodes where we can only be 50% sure of the classification. At that point, Cottonwood is largely distinguished from Ponderosa Pine and Douglas-fir by being more likely to have a higher hillshade index value at 9am, being further from roads at certain elevations and being less likely to grow in the Bullwark soil type. As for Aspen, it is distinguished from Lodgepole Pine by being less likely to be far from roadways and fire points. It is also less likely to be far from water as compared to Douglas-Fir.

The model accurately classified less than 66% of observations having cover types Spruce/Fir, Lodgepole Pine and Ponderosa Pine. Of Spruce/Fir observations, 34% are incorrectly classified. Specifically, over 50% of those misclassified were classified as Lodgepole Pine. On the lower end of Spruce/Fir's elevation range where it overlaps with Lodgepole Pine, we see nodes where representation between the two is approximately a 50:50 split. Various splits on distances improve homogeneity incrementally, but not much. The difficulty in distinguishing between the two is consistent with the attribute similarity we found in the Naive Bayes analysis. On the higher end of Spruce/Fir's

elevation range, we see nodes with a high representation of both Spruce/Fir and Krummholz; however, after splitting on soil types, particularly the Moran soil type where Krummholz is far more likely to grow (Appendix D), the model is better able to distinguish between the two.

Ponderosa Pine and Douglas-fir also get confused with each other often. Approximately one out of every 5 Douglas-Fir or Ponderosa Pine predictions made by the model are incorrectly classified as each other. As we can see from the density plot above, there is a lot of overlap in their elevation range. Other variables, such as soil type, provide additional classification power. For example, when Douglas-fir and Ponderosa Pine are above elevation 2,370 meters, soil types help provide additional classification. Ponderosa Pine grows more often in Ratake family soil type. Additionally, Douglas-fir is more likely to grow in Bullwark soil type.

Lastly, the model accurately classified 58% of Lodgepole Pine observations. It is largely misclassified as Spruce/Fir on the higher end of its elevation range and Aspen on the lower end of its elevation range. After elevation, distances are the most impactful attribute to distinguish between Lodgepole Pine and Aspen. Aspen is more likely to be closer to both fire points and roads. Again, this is consistent with the findings from Naive Bayes (Appendix C). Even though Lodgepole Pine has the lowest prediction accuracy among cover types, it is still significantly better than the benchmark.

Random Forest

The analysis above was based on a single classification tree. One drawback to classification trees is that they are susceptible to variability. Each split in the tree is highly dependent on the previous splits that were made. For example, a change in the order of splits could produce a dramatically different tree. In order to counteract this, we chose to produce a series of classification trees and aggregate their results. This is known as a random forest.

The theory behind random forests is that obtaining the collective opinion of a group is superior to the opinion of a single person who may have their own biases. The error rate using random forest was 19%, the lowest of all the models we attempted, supporting the effectiveness of the random forest methodology. Not surprisingly, the model identified the most impactful variable by far in terms of node purity as elevation. Other important variables identified were distance to roads, fire and hydrology and the Cache la Poudre Wilderness Area. This fits with our previous findings. See appendix E for full breakdown of variable importance.

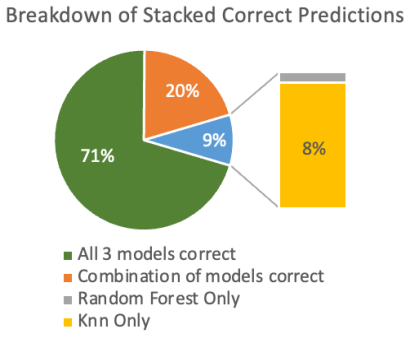
Ensemble

Our final task was to combine our models through a method called “stacking”. Stacking works on the theory that the whole is greater than the sum of its parts. Specifically, combining the models should leverage individual strengths and compensate for weaknesses resulting in greater overall predictive power.

In order to accomplish this, we added the individual predictions from KNN, Random Forests, and Naïve Bayes to our existing dataset and fed this “supplemented” training data to a new model referred to as the “manager model”. The manager model’s job is to learn how to best combine the predictions of the base models as well as its own findings in order to increase model performance. While several models were considered for the role of manager model, random forest produced the lowest error rate.

In terms of performance, the overall error rate was 22%. This performs better than all models with the exception of random forest which has an error rate of 19%. We have two theories to explain why the stacking model did not perform as well as expected based on the theory behind the model.

The first is the similarity between the input models in terms of predicting certain cover types well and others relatively poorly. In order to investigate this theory, we looked at the number of instances where only one model correctly classified cover type while the remaining models were incorrect. Out of all correct classifications from the manager model using the test dataset, only 9% were correctly made by only one model type. This means 91% of correct classifications were predicted by all models or a combination of models. While this is valuable information, it’s important to place it in the context of the stacking model. The manager model ultimately predicts the cover type so it is possible that a correct classification was made by a base model but it was not ultimately what the manager model predicted. This leads into our next theory as to why random forest performed better on its own than stacked.

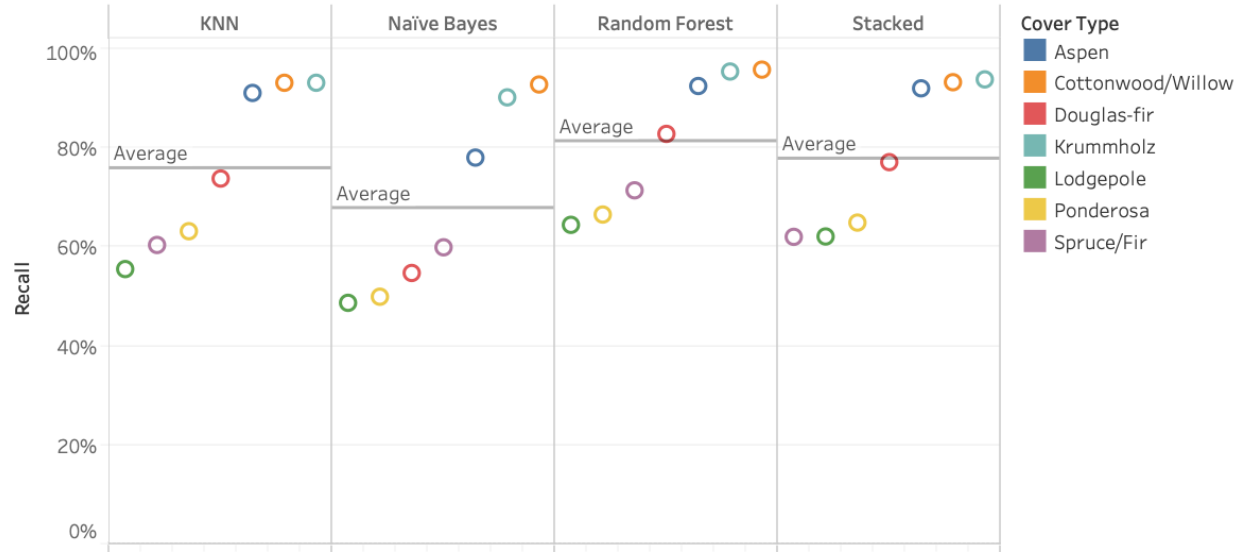


There were 679 instances in the test data set where random forest correctly predicted the cover type but the manager model predicted a different (incorrect) cover type. We theorize this is because KNN correctly classified similar observations in training which didn't generalize as well to the test set. The manager model learned KNN was a strong classifier with the training set and weighed it's predictions higher with the test set. This resulted in the predictions of random forest not being utilized by the manager model to their fullest extent, resulting in an error rate higher than random forest on its own. To test this, we switched to k=3 which resulted in a 17% error rate for the stacked model, the best we have been able to achieve. The error rate for KNN on its own increased to 26%.

Before changing k, the most impactful variable in terms of node purity in the stacked model were the predictions made by KNN. This changed to be the predictions made by random forest after the change in k was made (Appendix F).

Comparison

Generally, all models performed well at predicting cover type - particularly in comparison to the benchmark. Below we see the breakdown of recall by cover type:



As mentioned in the stacking discussion, the different models tend to classify the same cover types well and others relatively poorly. Cottonwood/Willow, Aspen and Krummholz are predicted well by all models while Spruce/Fir, Lodgepole Pine and Ponderosa Pine

have the most errors. The quality of Douglas-Fir classifications seems to vary the most amongst models, but is generally about average.

When comparing between the two models, we see benefits and drawbacks to both of them. A benefit of Knn is that it is a simple model with strong prediction power that is easy to understand and equally easy to implement. Additionally, it does particularly well with observations that other models have trouble classifying. As previously discussed, one of the drawbacks of KNN is that it is only able to use numerical data. KNN's ability to accurately classify a cover type will partly depend on the dependency of a specific cover type to numerical predictors. For example, the use of soil type to help classify Ponderosa Pine likely explains its reduced error rate for the classification tree as compared to KNN, where categorical data was excluded. Additionally, we theorize that KNN will have particular trouble with cover types that have extremely similar attributes due to the nature of the KNN model. For example, Spruce/Fir and Lodgepole Pine observations will mostly be plotted close together. When searching for neighbors, the model is likely to encounter a mix of both classes. This explains why the best value for "k" was found to be 1, which leaves us with a lower error rate but generalization issues.

Classification trees are a strong classifier for cover types. They can accept both numerical and categorical predictors and their flexibility allows them to split subpopulations based on multiple, specific data points that differentiate between similar cover types. Additionally, single trees and random forest can work together to compensate for each other's weaknesses. Classification trees are easy to understand and explain to end users of the model; however, they can be prone to overfitting and are susceptible to variability. Random forest increases accuracy over classification trees and controls for overfitting; however, they are not as easy to view as classification trees.

Naïve Bayes is easy to understand and build; however, it does not do as well as other models in predicting cover types. Since the model only accepts categorical data, we lose some of the prediction power found in the numeric details versus binned values. This is particularly relevant for Lodgepole Pine, Ponderosa Pine and Douglas Fir. As we see from the classification tree, they are distinguishable from each other by detailed, specific cut points. Additionally, Naïve Bayes assumes the predictor probabilities are not impacted by each other. For example, the probability of Spruce/Fir being in Rawah Wilderness Area remains 48% despite the elevation level. Given the cartographical nature of our dataset, this assumption is likely not realistic and could be an additional factor as to why the prediction power is not as strong as other models.

As previously discussed, the benefit to a stacked model is that it leverages the strengths and weaknesses of several models, resulting in stronger overall performance. However,

it comes with an added complexity in having to build and run several different models. Additionally, it is difficult to explain to end users. Given this, we would not choose this as our preferred model. When predicting cover types in the Roosevelt National Forest of northern Colorado we would use Classification Trees, specifically Random Forest.

Conclusion

The purpose of our project was to predict the forest cover type in four different wilderness areas in the Roosevelt National Forest of northern Colorado. We explored four different predictive model types based on the composition of the data and our intended target, cover type. The input data was both categorical and numerical, with a target output that was categorical, so we opted to use KNN analysis, Naïve Bayes Classification, Classification Trees, and Random Forest as our predictive models. KNN analysis can only use numerical inputs, so categorical variables such as Wilderness Area and Soil Type were removed. For Naïve Bayes, the model only accepts categorical variables, so all numeric variables such as Elevation were converted to binned values. However Classification Trees and Random Forest can use both categorical and numeric variables in its analysis, so they were used in the training of their models.

Overall, all models did well with predicting cover types, particularly when compared to benchmark error rate of 86%. The error rates for each of the models are shown below:

Model	Error Rate
KNN Analysis	24%
Naïve Bayes	32%
Classification Tree	23%
Random Forest	19%
Stacked Model - k = 1	22%
Stacked Model - k = 3	17%

We believe the stacked model didn't perform as strongly initially because the predictions of random forest weren't being utilized by the manager model to their fullest extent. Once k was increased to 3, the error rate on the stacked model reduced to 17%. The Random Forest model produced the next lowest error rate at 19%.

We believe Random Forest is able to provide us with a low error rate because of its ability to incorporate both numeric and categorical variables, which is most fitting for our data set. In addition to being able to utilize both types of variables, Random Forest incorporates the results of a large number of classification trees, helping reduce the biases of individual trees alone, and decreasing the error rate of the total model as a

result. Additionally, Random Forest was best able to identify our most important variables, elevation, as well as other important variables such as distance to roads, fire, and hydrology, and the Cache la Poudre Wilderness Area.

Functionally, all models have their pros and cons in terms of data inputs, preprocessing efforts, time and resources needed to train the model, and how intuitive and easy it is to explain the model to stakeholders. While the training of Random Forest can be costly in terms of time and resources, it can use both categorical and numeric inputs which allows it to use more of the dataset when training. Random Forest provides us with the ability to easily understand what is occurring within the model, however due its complexity, it's not as easy to show others visually. However, the transparency that classification trees provide is paramount when trying to explain results to a non-technical audience. Due to its optimal predictive power, we would recommend using a Random Forest model to predict the cover types in four specific wilderness areas within Roosevelt National Forest of northern Colorado.

Appendix A:

Elevation - Elevation in meters
Aspect - Aspect in degrees azimuth (The azimuth angle is the compass direction from which the sunlight is coming)
Slope - Slope in degrees
Horizontal_Distance_To_Hydrology - Horz Dist to nearest surface water features
Vertical_Distance_To_Hydrology - Vert Dist to nearest surface water features
Horizontal_Distance_To_Roadways - Horz Dist to nearest roadway
Hillshade_9am (0 to 255 index) - Hillshade index at 9am, summer solstice
Hillshade_Noon (0 to 255 index) - Hillshade index at noon, summer solstice
Hillshade_3pm (0 to 255 index) - Hillshade index at 3pm, summer solstice
Horizontal_Distance_To_Fire_Points - Horz Dist to nearest wildfire ignition points
Wilderness_Area (4 binary columns, 0 = absence or 1 = presence) - Wilderness area designation
Soil_Type (40 binary columns, 0 = absence or 1 = presence) - Soil Type designation
Cover_Type (7 types, integers 1 to 7) - Forest Cover Type designation

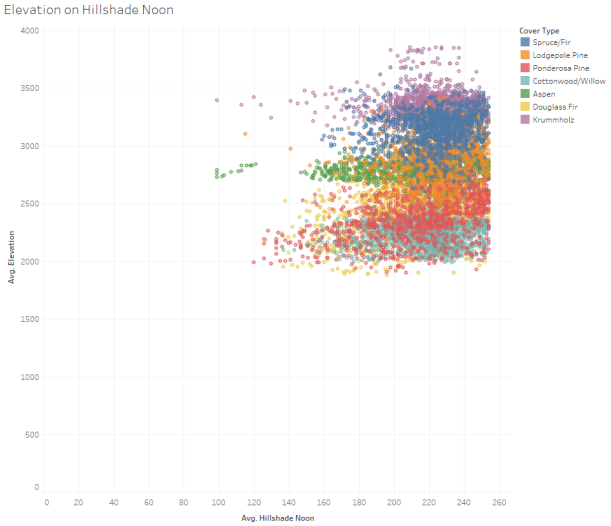
The wilderness areas are:
1 - Rawah Wilderness Area
2 - Neota Wilderness Area
3 - Comanche Peak Wilderness Area
4 - Cache la Poudre Wilderness Area

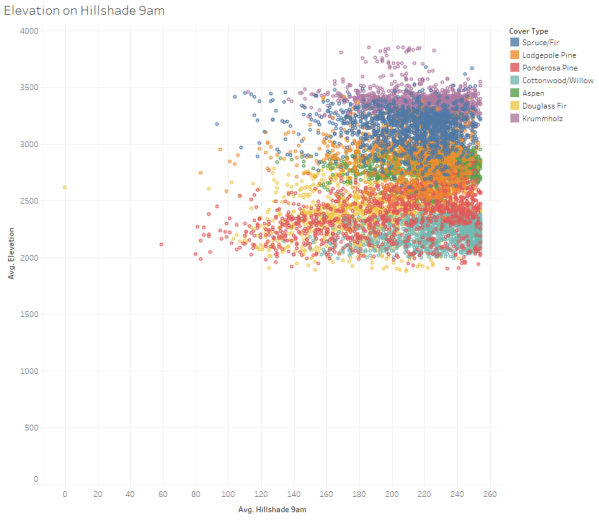
The soil types are:

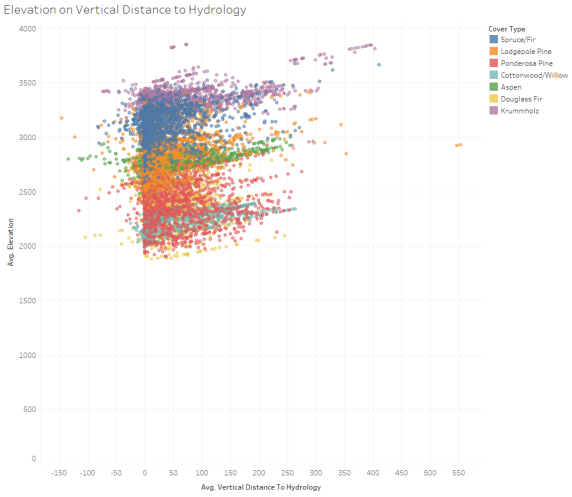
1 Cathedral family - Rock outcrop complex, extremely stony.
2 Vanet - Ratake families complex, very stony.
3 Haploborolis - Rock outcrop complex, rubbly.
4 Ratake family - Rock outcrop complex, rubbly.
5 Vanet family - Rock outcrop complex complex, rubbly.
6 Vanet - Wetmore families - Rock outcrop complex, stony.
7 Gothic family.
8 Supervisor - Limber families complex.
9 Troutville family, very stony.
10 Bullwark - Catamount families - Rock outcrop complex, rubbly.
11 Bullwark - Catamount families - Rock land complex, rubbly.
12 Legault family - Rock land complex, stony.
13 Catamount family - Rock land - Bullwark family complex, rubbly.
14 Pachic Argiborolis - Aquolis complex.
15 unspecified in the USFS Soil and ELU Survey.
16 Cryaquolis - Cryoborolis complex.
17 Gateview family - Cryaquolis complex.
18 Rogert family, very stony.
19 Typic Cryaquolis - Borohemists complex.
20 Typic Cryaquepts - Typic Cryaquolls complex.
21 Typic Cryaquolls - Leighcan family, till substratum complex.
22 Leighcan family, till substratum, extremely bouldery.
23 Leighcan family, till substratum - Typic Cryaquolls complex.
24 Leighcan family, extremely stony.
25 Leighcan family, warm, extremely stony.
26 Granile - Catamount families complex, very stony.
27 Leighcan family, warm - Rock outcrop complex, extremely stony.
28 Leighcan family - Rock outcrop complex, extremely stony.
29 Como - Legault families complex, extremely stony.
30 Como family - Rock land - Legault family complex, extremely stony.
31 Leighcan - Catamount families complex, extremely stony.
32 Catamount family - Rock outcrop - Leighcan family complex, extremely stony.
33 Leighcan - Catamount families - Rock outcrop complex, extremely stony.
34 Cryorthents - Rock land complex, extremely stony.
35 Cryumbrepts - Rock outcrop - Cryaquepts complex.
36 Bross family - Rock land - Cryumbrepts complex, extremely stony.
37 Rock outcrop - Cryumbrepts - Cryorthents complex, extremely stony.
38 Leighcan - Moran families - Cryaquolls complex, extremely stony.
39 Moran family - Cryorthents - Leighcan family complex, extremely stony.

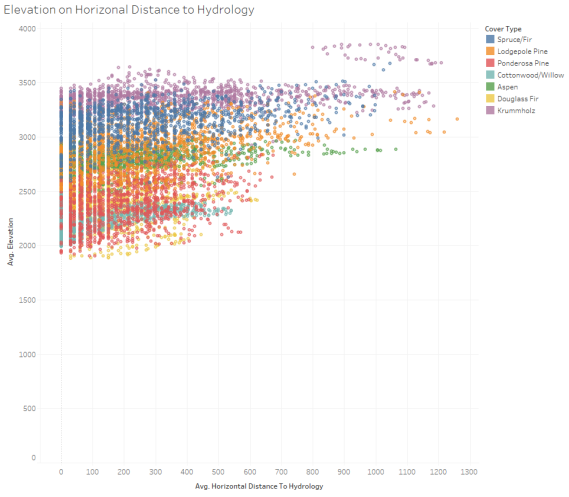
40 Moran family - Cryorthents - Rock land complex, extremely stony.

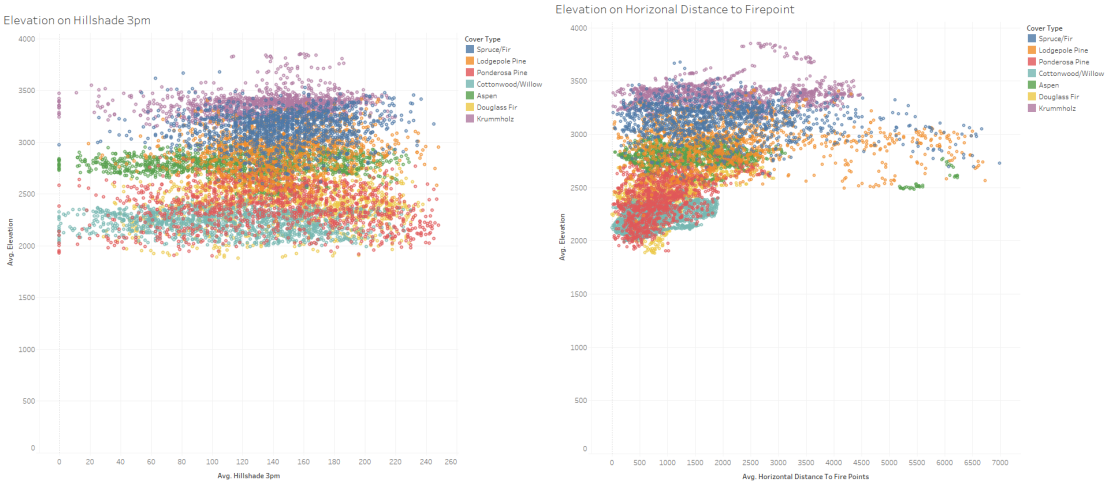
Appendix B: Scatterplots of variables (referenced in KNN section)











Appendix C: Probability of distance from roadways and fire points given cover type

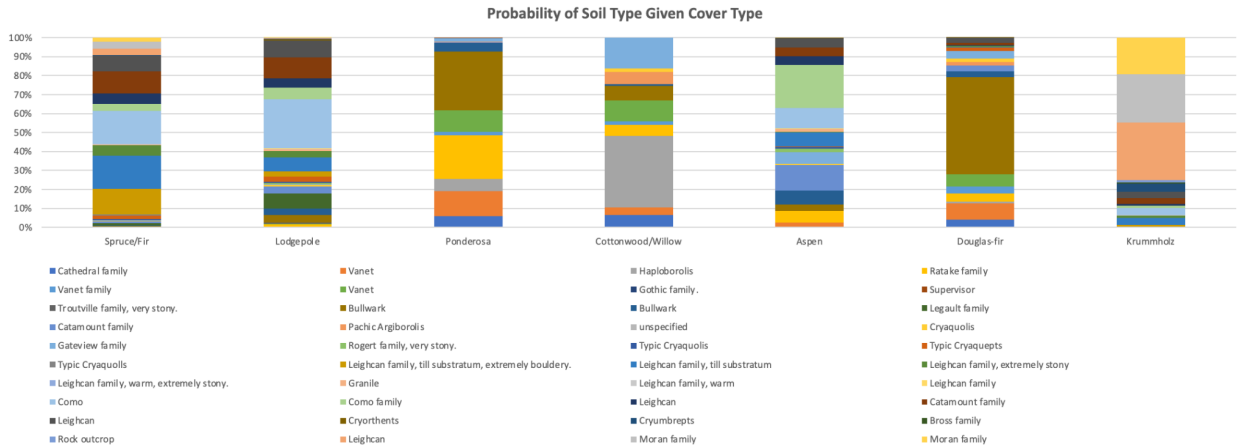
Probability of Distance from Roadways (meters) Given Cover Type

	0 - 689	689 - 1380	1380 - 2070	2070 - 2760]	2760 - 3440	3440 - 4130	4130 - 4820	4820 - 5510]	5510 - 6200	6200 - 6900
Spruce/Fir	9%	16%	18%	16%	14%	10%	8%	6%	4%	0%
Lodgepole Pine	13%	22%	18%	13%	11%	6%	5%	5%	6%	1%
Ponderosa Pine	38%	39%	15%	6%	1%	0%	0%	0%	0%	0%
Cottonwood/Willow	28%	62%	10%	0%	0%	0%	0%	0%	0%	0%
Aspen	34%	21%	26%	14%	2%	0%	1%	2%	0%	0%
Douglas-fir	28%	43%	22%	5%	1%	0%	0%	0%	0%	0%
Krummholz	1%	14%	20%	21%	15%	15%	11%	4%	0%	0%

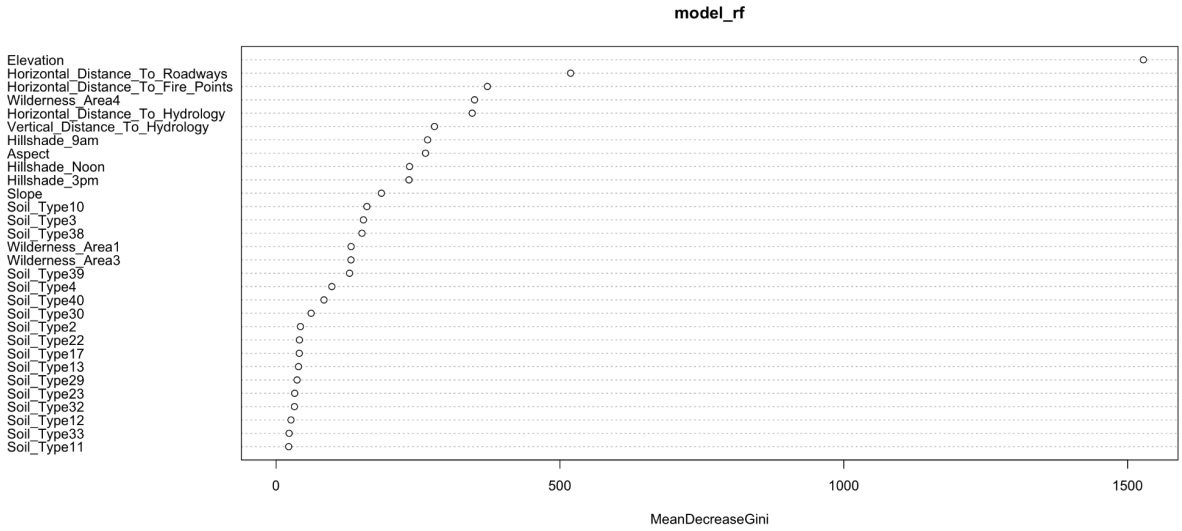
Probability of Distance from Fire Points (meters) Given Cover Type

	0 - 699	699 - 1400	1400 - 2100	2100 - 2800	2800 - 3500	3500 - 4200	4200 - 4900	4900 - 5590	5590 - 6290	6290 - 7000
Spruce/Fir	13%	23%	22%	21%	11%	5%	2%	1%	2%	1%
Lodgepole Pine	10%	25%	24%	20%	6%	3%	4%	4%	2%	1%
Ponderosa Pine	40%	43%	13%	4%	0%	0%	0%	0%	0%	0%
Cottonwood/Willow	44%	40%	16%	0%	0%	0%	0%	0%	0%	0%
Aspen	14%	35%	33%	14%	1%	0%	0%	2%	1%	0%
Douglas-fir	30%	48%	15%	6%	1%	0%	0%	0%	0%	0%
Krummholz	11%	20%	21%	21%	14%	12%	1%	0%	0%	0%

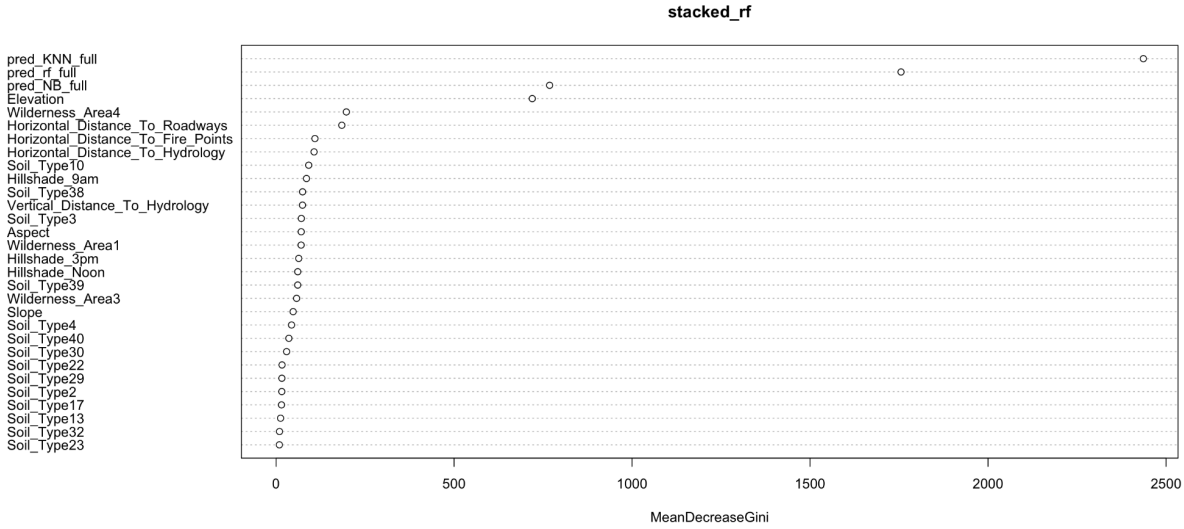
Appendix D: Probability of soil type given cover type

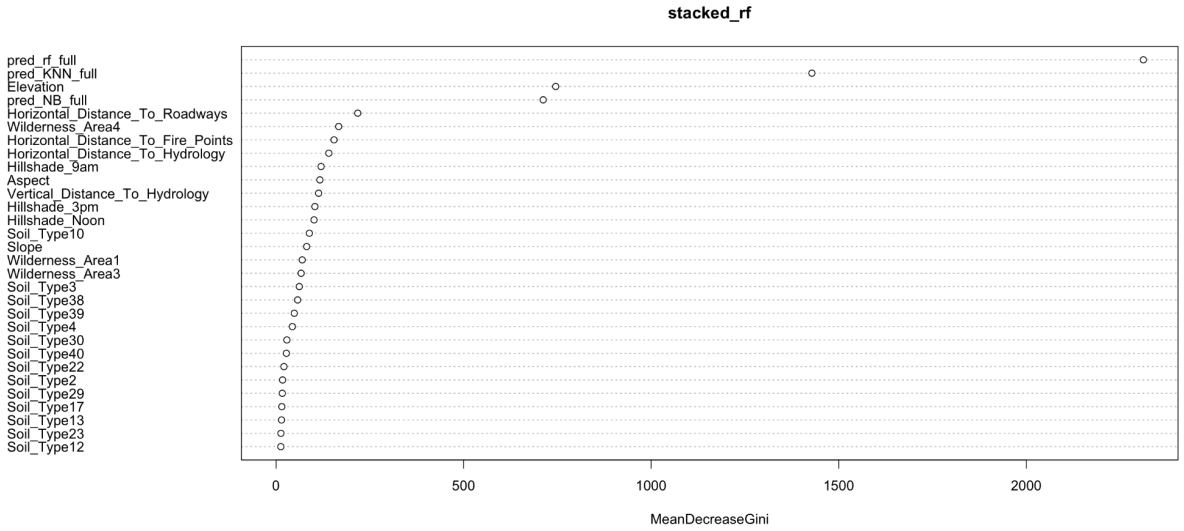


Appendix E: Random forest variable importance



Appendix F: Stacked Model - Random forest variable importance (before and after k change)





Appendix G: see attached file for tree (not full tree - minsplit=25,minbucket=25,cp=.0001)