PHASE 3

Prof. Nathan Karst

Nishtha Gupta & Jacob Nyamu

Introduction:

Kaggle hosted a competition with a dataset that the US Forest Service surveyed from the Roosevelt National Forest in Colorado. In it, there are several data points, for example, the distance to fire points, slope, elevation, and even soil types. The challenge would be using these different variables to predict what the cover type might be. In this paper, we build upon our findings from Phase 2 analysis. 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. First, we pre-process the data using different strategies. After that, we build three machine learning models to predict the forest cover type. Each model has a unique approach and limitations. This is why we stack our different models using a manager model to smooth out the weaknesses and hopefully improve overall performance. We then compare the error rates produced by the constituent models to that of the stacked models.

Data Cleaning Strategies

As it is with any analysis, data hygiene is crucial to model performance. We implemented the following strategies to pre-process our dataset before creating the models.

Checking for null values

Null values can skew the average of a particular variable, which is why we looked up if we had any null values. Fortunately, our dataset didn't have any. If there were, we would use imputation techniques - filling with the mean of the values in that column, a sample, or creating a model that would predict what the missing value would be.

Removing unnecessary columns

We dropped the ID column/variable because it contributes very little to the overall prediction. Moreover, we found some columns, e.g., Soil Type 7, only had only one unique value of 0. In this case, it would mean that there were no instances of soil type 7 in the sample from the Roosevelt National Forest, which is why we dropped them as well.

Adding labels to Cover Type column

In the dataset, values for the cover type are represented as numbers, e.g., 5, 8, 4. However, the data dictionary, under the Data Fields section in the dataset, shows the names of the different cover types. For the classification tree model, we used the recode_factor function to assign the cover type name to the number in the dataset, e.g., Spruce/Fir and Aspen are cover types 1 and 5, respectively.

Binning certain columns

We chose to implement a Naive Bayes model as our third model. Its drawback is that it only accepts categorical input. So for some columns, such as elevation, we used binning in order to include them in the model. Binning is transforming numerical variables into categorical counterparts.

Principal Component Analysis

Strong relationships between the features drive the performance of the model. To deal with this, we perform PCA (Principal Component Analysis) in two models, i.e., Naive Bayes and Knn. 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.

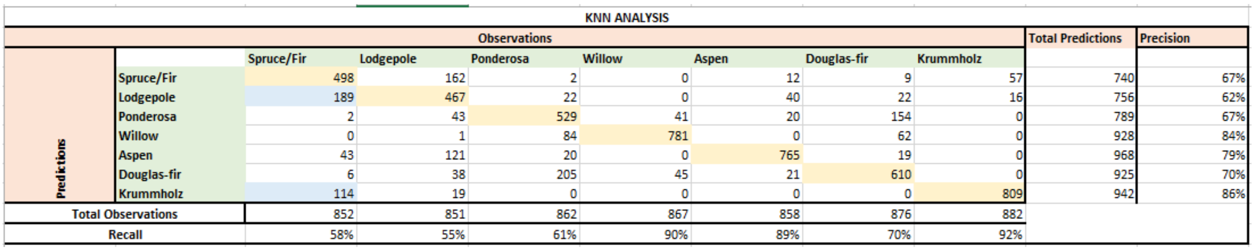
Performance before and after cleansing

Our dataset did not need much cleaning. However, we compared the performance before and after removing the unnecessary columns, such as Soil Type 7. Before, our Classification tree model's error rate was approximately 23% and was the same after dropping the columns and renaming the cover type names. Therefore, removing the columns had no effect on the overall model performance.

KNN Model Analysis

KNN (K - Nearest Neighbours) 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.

<u>Evaluating Model:</u>



KNN ANALYSIS

		Observations							Total Predictions	Precision
		Spruce/Fir	Lodgepole	Ponderosa	Willow	Aspen	Douglas-fir	Krummholz		
Predictions	Spruce/Fir	498	162	2	0	12	9	57	740	67%
	Lodgepole	189	467	22	0	40	22	16	756	62%
	Ponderosa	2	43	529	41	20	154	0	789	67%
	Willow	0	1	84	781	0	62	0	928	84%
	Aspen	43	121	20	0	765	19	0	968	79%
	Douglas-fir	6	38	205	45	21	610	0	925	70%
	Krummholz	114	19	0	0	0	0	809	942	86%
Total Observations		852	851	862	867	858	876	882		
Recall		58%	55%	61%	90%	89%	70%	92%		

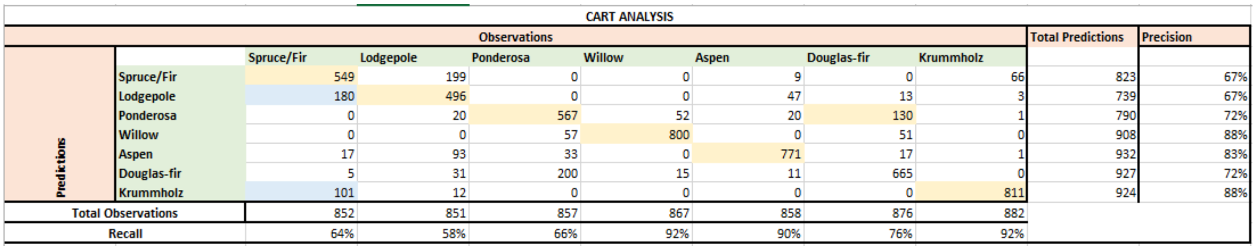
When evaluating the performance of cover-type predictions, we used recall as our metric in determining the success of the prediction. Recall refers to % of observations correctly predicted. In the chart above, we can see that the model was able to successfully predict certain cover types more accurately than others. For example, Cottonwood/Willow, Aspen, and Krummholz have high recall rates of 90%, 89%, and 92%, respectively. While cover types such as Spruce/Fir, Lodgepole Pine, and Ponderosa Pine have significantly lower recall rates of 58%, 55%, and 61%, 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 14% of the observations.

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.

Classification Model Analysis

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. In terms of performance, the overall error rate of our pruned model was 23%, reducing the error by 73% as compared to the benchmark.

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 Group 2 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 those at the higher end of the performance spectrum have slightly lower precision than recall.



CART ANALYSIS

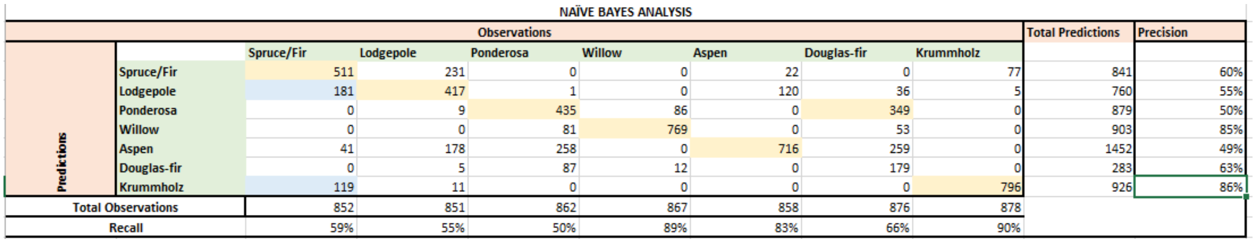
		Observations							Total Predictions	Precision
		Spruce/Fir	Lodgepole	Ponderosa	Willow	Aspen	Douglas-fir	Krummholz		
Predictions	Spruce/Fir	549	199	0	0	9	0	66	823	67%
	Lodgepole	180	496	0	0	47	13	3	739	67%
	Ponderosa	0	20	567	52	20	130	1	790	72%
	Willow	0	0	57	800	0	51	0	908	88%
	Aspen	17	93	33	0	771	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 Observations		852	851	857	867	858	876	882		
Recall		64%	58%	66%	92%	90%	76%	92%		

Naive Bayes' Model Analysis

The 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 of 3,325 meters was transformed into an elevation of "3,320 - 3,440 meters".

Evaluating Model:

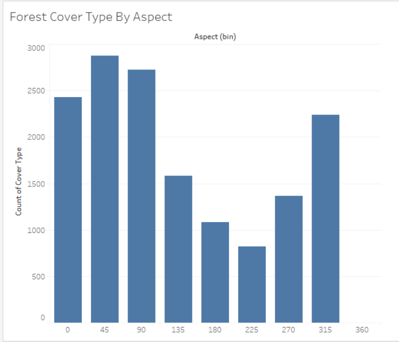
In terms of performance, the overall error rate was 36%, reducing the error by 64% as compared to the benchmark but representing the worst of our models. Cottonwood/Willow and Krummholz were still accurately classified in over 90% of their observations. However, Lodgepole Pine, Ponderosa Pine, and Douglas Fir observations were only correctly classified approximately 50% of the time.



NAÏVE BAYES ANALYSIS

		Observations							Total Predictions	Precision
		Spruce/Fir	Lodgepole	Ponderosa	Willow	Aspen	Douglas-fir	Krummholz		
Predictions	Spruce/Fir	511	231	0	0	22	0	77	841	60%
	Lodgepole	181	417	1	0	120	36	5	760	55%
	Ponderosa	0	9	435	86	0	349	0	879	50%
	Willow	0	0	81	769	0	53	0	903	85%
	Aspen	41	178	258	0	716	259	0	1452	49%
	Douglas-fir	0	5	87	12	0	179	0	283	63%
	Krummholz	119	11	0	0	0	0	796	926	86%
Total Observations		852	851	862	867	858	876	878		
Recall		59%	55%	50%	89%	83%	66%	90%		

The model accurately classified over 90% of observations having cover types Cottonwood, Aspen, and Krummholz. The high recall for Krummholz is large because it makes up the majority of cover types represented in its elevation band. Conversely, 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 9 am, being further from roads at certain elevations, and being less likely to grow in the Bullwark soil type.



The forest species are predominantly grouped in North East (0-90 Degrees) facing direction, transitioning decline in the middle aspect range of Southern (135 - 225 Degrees) facing direction. There is a gradual increase in trend in the range of North West (270-360 Degrees) facing direction. From this, it could be inferred that species are predominant in north-facing slopes as it receives a lesser amount of sunlight which helps them to retain moisture and are more humid. On the contrary, south-facing slopes receive direct sunlight and are warmer and drier, making it difficult for species to survive for long.

The model accurately classified less than 60% of observations having cover types Spruce/Fir, Lodgepole Pine, and Ponderosa Pine. Of Spruce/Fir observations, 40% 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.

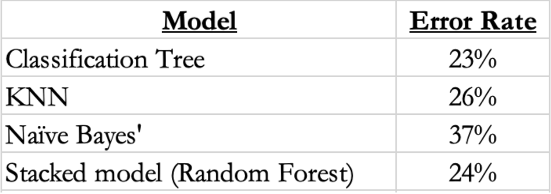
Lastly, the model accurately classified 55% of Lodgepole Pine observations. It is largely misclassified as Spruce/Fir on the higher end of its elevation range and Aspen on the lower end of its elevation range. After elevation, distances are the most impactful attribute to distinguish between Lodgepole 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. Even though Lodgepole Pine has the lowest prediction accuracy among cover types, it is still significantly better than the benchmark.

Ensemble Learning:

Different machine learning models have various weaknesses. The KNN model, for example, only uses numerical input, while our dataset has a lot of categorical inputs. To smooth out these individual weaknesses, we use an ensemble technique called stacking. This method uses the predictions from the other models as inputs alongside the ones from the dataset. The other models become helper models for the manager model. To do so, we bind the columns to the original data frame using the cbind function. To confirm we successfully bound the new columns, we check the number of columns of the data frame. The original one had 53 columns, and the new one has 56 columns.

After binding the predictions from the other models to create a new data frame, we use the same partitions of testing and training portions. We avoid sampling another set of training and testing partitions because it would cause leakage. We also choose random forests as our manager model and set it to have 500 trees.

Before ensembling, the classification tree model produced the best results. It lowered the benchmark error rate from 86% to 23% after pruning. After running the stacked model, the error rate came to 24%, a tad higher than our best performer - the classification tree. Below are the results for all models by error rates.



Model	Error Rate
Classification Tree	23%
KNN	26%
Naïve Bayes'	37%
Stacked model (Random Forest)	24%

Comparison:

Three 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.

On comparing the models, we observed Knn uses numerical data to predict. This implies for Knn to predict, it will have to have a dependency on a specific cover type to numerical predictors. 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 forests can work together to compensate for each other's weaknesses.

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.

The benefit of a stacked model is that it leverages the strengths and weaknesses of several models, resulting in stronger overall performance. However, it comes with the added complexity of 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:

To summarize, we implemented data cleaning strategies such as checking for null values and removing unnecessary variables/columns. Despite implementing these strategies, the model performance did not change significantly. We also chose Naive Bayes as our third machine learning model. This model only accepts categorical input, so we binned some columns, such as elevation, to run into the model. Despite lowering the error rate from a benchmark rate of 64% to 36%, the Naive Bayes model was the worst performer in comparison to our other models, KNN and Classification Tree. To blend all models together to hopefully achieve optimal results, we stacked the models. We implemented a random forests model as our manager model and the rest as helper models. Running it performed better than our Naive Bayes' and KNN models at 24% and just 1% higher than the classification tree model.