QTM 7571 – Project Group 4 Project Phase 1
Dilo, Kenneth, Javier, and Priyanka

Executive Summary

The goal of this project was to build a machine learning model that could accurately predict forest cover types in the Roosevelt National Forest in Northern Colorado. For this project, we downloaded the training dataset from Kaggle and loaded it into R for exploration. We verified the data quality, confirming there were no missing values, identified the structure, and explored the content of the variables. After exploring the data, we transformed and cleaned it and had it ready for modeling. We then built and evaluated two models: K-Nearest Neighbors (KNN) and Logistic Regression (more on these models later).

Despite removing some potentially important categorical features from the KNN model (since KNN cannot directly handle categorical inputs) it achieved a 28.5% error rate. Logistic Regression followed closely, with a 30.3% error rate. These results were a significant improvement from our baseline model, which had an error rate of 86.5% by simply predicting the most common forest cover type for all observations. These findings not only helped us better understand the data, but also highlighted the strengths and weaknesses of each model, as we will discuss by diving deeper into each section.

Data Preparation & Visualization

Data Understanding & Transformation

Our dataset consisted of 15,120 rows and 55 columns with a combination of numerical and categorical features. Table 1 shows the numerical and categorical columns of the data frame.

Column Name	Data Type		Column Name	Data Type
Elevation	Numerical		Hillshade_Noon	Numerical
Aspect	Numerical		Hillshade_3pm	Numerical
Slope	Numerical		Horizontal_Distance_To_Fire_Points	Numerical
Horizontal_Distance_To_Hydrology	Numerical		Wilderness_Area(1–4)	Cat (logical)
Vertical_Distance_To_Hydrology	Numerical		Soil_Type(1–40)	Cat (logical)
Horizontal_Distance_To_Roadways	Numerical		Cover_Type (target)	Cat (factor)
Hillshade_9am	Numerical			

Table 1: Summary of Dataset Columns

We began by confirming that the dataset was clean (i.e. there were no missing or null values across any column), which saved us significant preprocessing time. We also ensured that each row of data had exactly one wilderness area and one soil type by checking the row-wise sums across the binary columns. This allowed us to transform the 40 binary soil type columns into a single column labeled 'Soil_Type.' Similarly, we combined the 4 binary wilderness area columns into one 'Wilderness_Area' column. These changes allowed us to treat these variables as categorical (factor) data rather than as many independent logical (true/false) variables, which simplifies the modeling process and makes the output easier to interpret.

Next, we dropped the 'ID' column, which is simply a row identifier and does not contain any predictive value. We also converted the 'Cover_Type' variable, the variable we are trying to predict (often called the Target Variable), into a factor, since we are working with a classification problem with seven distinct forest types.

We then created a training and test split: 60% of the data was used to train the models and the remaining 40% was reserved for testing their performance. This approach helps us evaluate how well our models perform on unseen data.

Visual Insights

To help us further understand this dataset, we decided to visualize the data in a number of different ways to give us some understanding of the data before we modeled. First we observed that the distribution of forest cover types in the dataset was balanced across all seven categories, resulting in 2,160 instances of each cover type, meaning no single cover type dominated the data, and ensuring that the model is not biased toward the most common class. We also showed the distribution of each continuous variable. For example, elevation was somewhat normally distributed, while distances to hydrology or roadways were skewed toward smaller values, indicating that most forest areas were closer to water sources and roads.

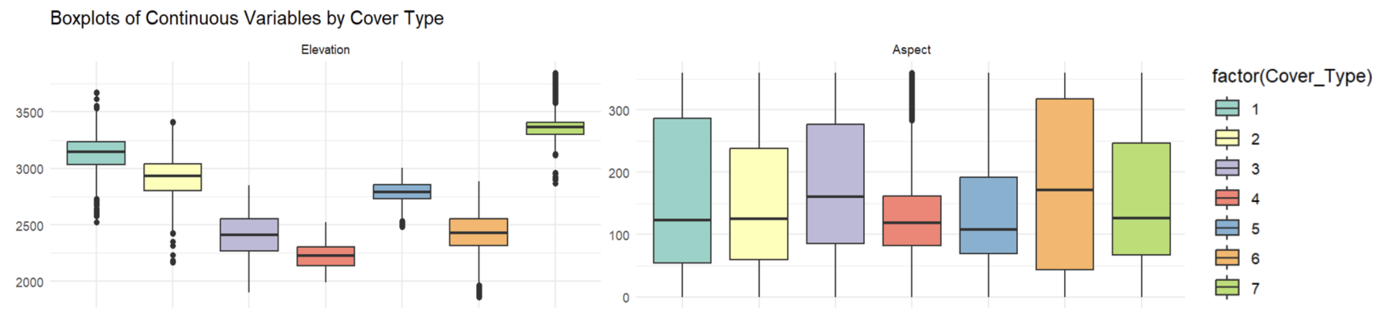

Figure 1: Box plots of Elevation and Aspect for each Cover Type in the dataset

Box plots helped us see how these variables varied across different forest cover types. As seen in Figure 1, elevation appeared to be a strong differentiator between forest types, as some types were clearly found only at higher or lower elevations. On the other hand, variables like Aspect (the compass direction a slope faces) showed less variation across cover types, suggesting they might not be as informative for prediction.These visualizations provided helpful early insights into which variables were most likely to be useful predictors such as Elevation, and which ones might be less informative such as Aspect.

Performance Measurement

After building our models, we needed a way to understand how well they performed on our Test data. We used performance metrics to evaluate how accurate and reliable each model's predictions were when tested on data it hadn't seen before.

To have something to compare to, we also established a baseline error rate. This represents the error rate we would get if we made a very simple prediction by always guessing the most common forest cover type in the training set for every observation in the test set. In our case, the baseline error rate was 86.5%, meaning this basic approach was wrong most of the time. Our goal was to build models that could improve upon this error rate and analyze the following metrics:

- **Error Rate:** This is the percentage of predictions that were incorrect. For example, an error rate of 30% means the model got 3 out of 10 predictions wrong. The lower the error rate, the better the model.
- **Accuracy:** This is the opposite of the error rate. It tells us how often the model was right. An accuracy of 70% means the model correctly predicted the cover type 7 out of 10 times.
- **Confusion Matrix:** This is a table that breaks down the model's predictions compared to the actual values. It shows which forest types the model tends to mix up. For example, if the model often predicts Type 2 when the true type is actually Type 3, the confusion matrix will highlight that.
- **Precision:** Precision tells us how often the model's prediction for a certain cover type is actually correct. For example, if the model predicts Type 4 twenty times, but only fifteen of those are right, the precision for Type 4 is 75%.
- **Recall:** Recall looks at it from the other direction than precision. Out of all the actual cases of a certain cover type, how many did the model catch correctly? If there were 100 real cases of Type 1 and the model correctly predicted 80 of them, the recall is 80%.

These metrics give us a more complete picture of model performance than just looking at accuracy alone. They help reveal not just whether the model is right, but also how it tends to be wrong.

Model 1: K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) model is an intuitive classification algorithm. It predicts the class of a new data point by looking at the "K" most similar observations in the training data. In our case, we chose K = 5 as a reasonable default to start with, and we plan to explore fine-tuning this value in our future work. The model looked at the five closest observations and assigned the most common forest cover type among them. Because KNN is sensitive to the scale of the data, we standardized all numeric variables before training the model to ensure that no variable would dominate due to its scale. We also removed the categorical variables (Soil_Type and Wilderness_Area), since KNN cannot directly handle non-numeric inputs. The KNN model performed significantly better than the benchmark error rate of 86.5%, achieving an error rate of 28.5%.

Model 2: Logistic Regression

Logistic Regression is a statistical modeling technique that estimates the probability that a given observation belongs to a particular category. In our case, we used multinomial logistic regression, which is suited for situations where the target variable has more than two categories, like our forest cover types, which range from 1 to 7. The model uses the input variables to calculate the likelihood of each possible cover type and then assigns the one with the highest probability. To improve the model's efficiency and reduce complexity, we used a technique called stepwise selection. This method removes variables of insignificant value from the model and therefore simplifying the model while maintaining strong performance. The logistic regression model had an error rate of 30.3%, meaning it was slightly less accurate than our KNN model but still a major improvement over the baseline.

Model Comparison

Both models strongly improved upon our baseline error rate of 86.5%. Surprisingly, KNN performed slightly better despite losing potentially valuable categorical information. This could mean that the categorical variables (soil and wilderness area) were less critical to predicting forest type than we originally thought, or it could indicate a mild overfitting issue. A model is overfit when it "memorizes" the training data instead of learning the patterns in the data, thus performing very well on the test dataset, but not performing as well on new data. Logistic Regression, while not performing as well as KNN, still provided good insights by clearly showing which variables mattered most for predicting forest cover.

Criteria	K-Nearest Neighbors (KNN)	Logistic Regression
Error Rate	28.5% — better overall performance	30.3% — slightly higher error rate
Categorical Variable Handling	Could not use soil or wilderness variables (excluded from model)	Included all variables, including soil type and wilderness area
Interpretability	Not easily interpretable — difficult to know which features influenced predictions	Highly interpretable — shows which variables contribute most to predictions
Overfitting Control	Used K = 5 and standardized all numeric features to prevent bias	Used stepwise selection to keep only the most useful variables
Preprocessing Required	Required additional steps — removed categorical variables and standardized data	Minimal preprocessing — able to use all features directly
Strengths	Strong accuracy; captures similarities in data effectively	Provides clear insights into which features matter most
Limitations	Can't handle categorical variables directly; performance drops on large datasets	Slightly less accurate; assumes relationships between variables are linear

Table 2: Criteria comparison between KNN and Logistic Regression models

To further evaluate model performance, we compared their predictive accuracy using precision, recall, and accuracy for each forest cover type. These metrics help us understand not just how often the model was correct overall, but how reliable it was for each specific category. The table below summarizes the results for both K-Nearest Neighbors (KNN) and Logistic Regression.

Cover Type	KNN			Logistic Regression		
	Precision	**Recall**	**Accuracy**	**Precision**	**Recall**	**Accuracy**
1	0.49	0.53	0.53	0.58	0.65	0.65
2	0.45	0.50	0.50	0.47	0.56	0.56
3	0.54	0.47	0.47	0.52	0.43	0.43
4	0.90	0.76	0.76	0.79	0.68	0.68
5	0.86	0.75	0.75	0.74	0.70	0.70
6	0.64	0.56	0.56	0.61	0.52	0.52
7	0.92	0.82	0.82	0.88	0.75	0.75

Table 3: Criteria comparison between KNN and Logistic Regression models

Both models performed similarly when predicting forest cover types 3, 4, and 7. For these types, both precision and recall were relatively high, indicating that the models were not only good at identifying them when they occurred but also accurate when they made those predictions (high recall & accuracy at the cover type level). This suggests that these forest types have more distinct or consistent feature patterns. In contrast, predictions for forest types 1, 2, 5, and 6 showed more inconsistency across both models. These types were more frequently misclassified, which could mean that these forest types share similar environmental conditions in the dataset, making them harder to separate using the variables available.

Summary & Future Work

This project allowed us to practice our machine learning skills to a real-world dataset by creating machine learning models to predict forest cover types based on environmental features. After exploring and preparing the data, we built and evaluated two models: K-Nearest Neighbors (KNN) and Logistic Regression. Both models performed significantly better than our baseline. KNN achieved a slightly better error rate of 28.5%, while Logistic Regression achieved an error rate of 30.3%, though offered greater interpretability. We also observed that some cover types (3, 4, and 7) were easier to predict due to clearer patterns, while the other variables seemed to have overlapping patterns for both models. KNN performed well even without categorical variables, though we must be conscious that better performance on the test dataset does not always indicate a better model, as it can be an indication of an overfit model.

For future work, we look forward to the following:

- Exploring more advanced models (e.g., classification trees, ensembling different models)
- Creating new columns or bucketing existing ones
- Fine-tuning model parameters (e.g. K in KNN such be sqrt(N) where N=Training Observations)

Overall, this project Phase 1 has helped us understand and practice model building, data preparation, and performance evaluation using real-world data. We look forward to learning different models and more ways on how we can improve our data cleaning and management.