Classifying Forest Cover Types

Eyal Stolov

June 2025

Introduction

In the following problem, we are asked to predict the different forest cover type based on cartographic variables. This dataset area includes four wilderness areas located in the Roosevelt National Forest of northern Colorado. These areas represent forests with minimal human-caused disturbances, so that existing forest cover types are more a result of ecological processes rather than forest management practices.

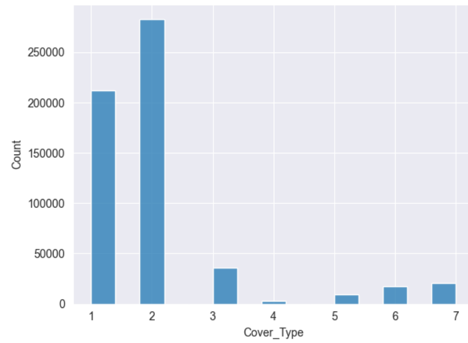
EDA

Features

Target Feature

Our target feature is called "Cover_Type", and is a categorial feature, with 7 different cover types:

1. Spruce/Fir
2. Lodgepole Pine
3. Ponderosa Pine
4. Cottonwood/Willow
5. Aspen
6. Douglas-fir
7. Krummholz



Unfortunately, our target feature is very imbalanced... We have 283,301 samples of class 2, whereas we have only 2747 samples of class 4! What does this mean for us? When splitting to train and test sets, we need to make sure there is a proportion of our mionority classes in both sets.

Predictor Features

The dataset contains 54 different predictor features:

1. Elevation - Elevation in meters
2. Aspect - Aspect in degrees azimuth
3. Slope - Slope in degrees
4. Horizontal_Distance_To_Hydrology - Horz Dist to nearest surface water features
5. Vertical_Distance_To_Hydrology - Vert Dist to nearest surface water features
6. Horizontal_Distance_To_Roadways - Horz Dist to nearest roadway
7. Hillshade_9am (0 to 255 index) - Hillshade index at 9am, summer solstice
8. Hillshade_Noon (0 to 255 index) - Hillshade index at noon, summer solstice
9. Hillshade_3pm (0 to 255 index) - Hillshade index at 3pm, summer solstice

10. Horizontal_Distance_To_Fire_Points - Horz Dist to nearest wildfire ignition points

11. Wilderness_Area (4 binary columns, 0 = absence or 1 = presence) - Wilderness area designation

12. Soil_Type (40 binary columns, 0 = absence or 1 = presence) - Soil Type designation

13. 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. athedral family - Rock outcrop complex, extremely stony.
2. anet - Ratake families complex, very stony.
3. aploborolis - Rock outcrop complex, rubbly.
4. atake family - Rock outcrop complex, rubbly.
5. anet family - Rock outcrop complex complex, rubbly.
6. anet - Wetmore families - Rock outcrop complex, stony.
7. othic family.
8. upervisor - Limber families complex.
9. routville 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.

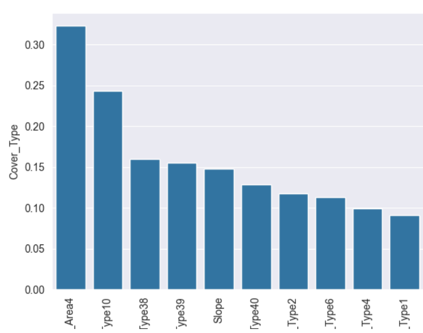
39. Moran family - Cryorthents - Leighcan family complex, extremely stony.

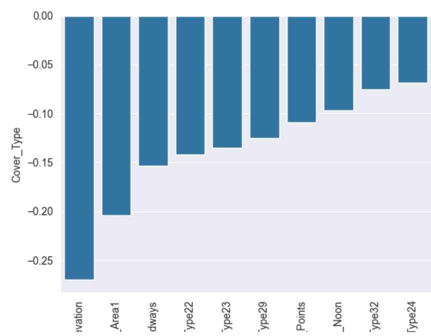
40. Moran family - Cryorthents - Rock land complex, extremely stony.

In summery, we have 44 binary features (0 or 1), and 10 numerical features.

Correlations

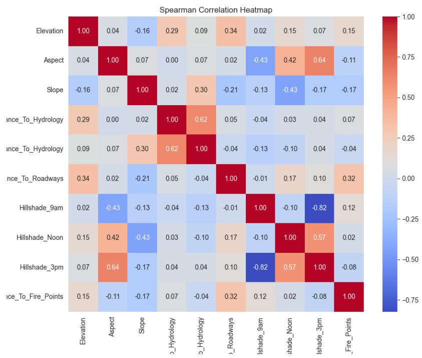
We start by checking what features have the biggest correlation with our target feature:

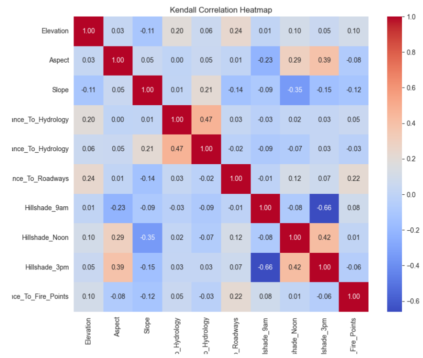




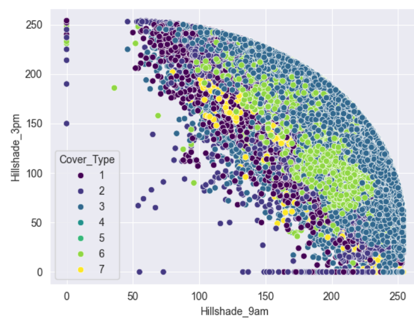
None of our features have a strong linear correlation with our target feature, suggesting linear methods might struggle. This indicates we should try using a non-linear model, like KNN.

We continue by checking the *spearman* and *kendall* correlations of all our numerical features:





Again, we don't see any strong correlations, except for the features $Hillshade\_3pm$ and $Hillshade\_9am$, lets have a look at there combined scatterplot:



This shape of the scatterplot makes a lot of sense, as this features are affected by the Earthś rotation. Nonetheless, by also coloring the dots by each $Cover\_type$, we can see class 6 (Douglas-fir), has quite a nice cluster inside. This further indicates KNN might be a more suitable algorithm for this exercise.

Training

For this problem, as suggested from the EDA, we will use KNN for our algorithm.

Potential Issues

Computationally Expensive

KNN is a non-parametric, this means we do not need to make assumptions about the relationship between the predictors and the target feature. KNN is also an instance-based algorithm means that our algorithm doesn't explicitly learn a model. Instead, it chooses to memorize the training instances which are subsequently used as "knowledge" for the prediction phase. The downside of this is it's both computationally and memory expensive. To help with those downsides, we use a special version of KNN, called $KDtreeKNN$. In this version KNN doesn't calculate distances with points that are far away and for sure won't be in our K closest neighbors.

Minority Classes

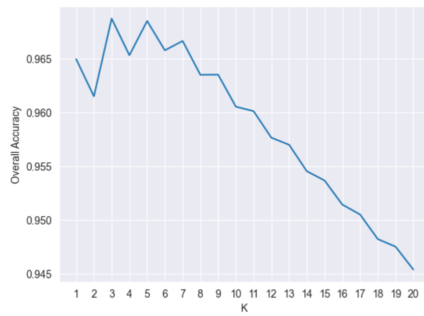
We have some classes that dont have a lot of samples compared to others. For this reason we will "stratify" our dataset, which simply means when splitting to test and train, we will by ration of classes. We will be using KFold, so at each fold we will do a stratified split.

Choosing K

How do we choose K? Let's try 20 different K's and then decide:

Overall Accuracy

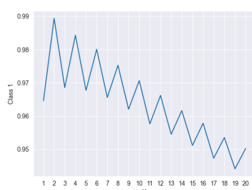
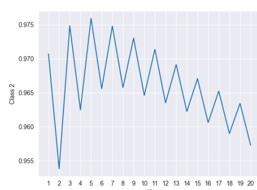
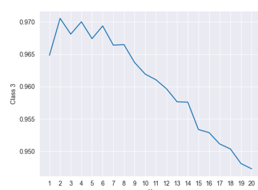
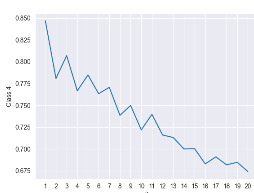
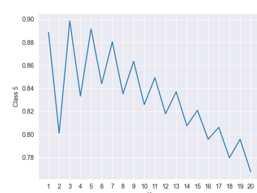
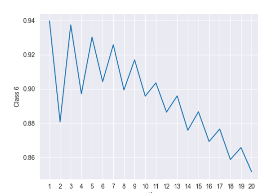
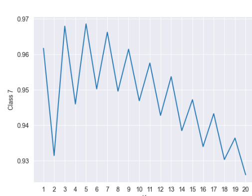
First, we check the overall accuracy of 20 different K's:



If we check only the overall accuracy, its clear the best K by overall accuracy is 3 with an impressive accuracy of 0.9687!

Accuracy Per Class

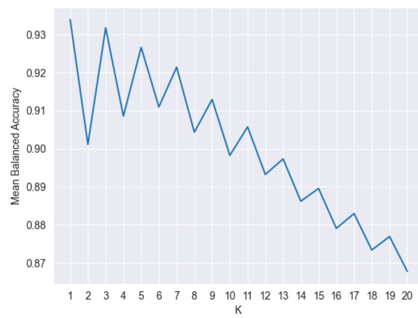
We can't check only the overall accuracy, as there are minority classes that need special attention. Lets see what is the best K per class:

Now, if we for instance want to minimize the recall for class 4 (one of our minority classes), we should actually choose $K = 1$!

Mean Balanced Accuracy

Another interesting metric is $MeanBalancedAccuracy$, which just means we take each class accuracy at each k, and calculate their combined mean:



Looking at this, it's clear that if we want to choose a K that gives fair accuracy in all the classes, we should choose $K = 1$ (accuracy = 0.9339).

Conclusions

In conclusion, choosing the right K will vary based on the priorities of the client. If we want to get overall good accuracy for all different classes, we will choose $K = 1$, but if we only care about overall accuracy between all classes, a better fit will be $K = 3$. Furthermore, it's very clear that choosing an odd K will almost always return better accuracy than an even one.