Forest Cover Type Prediction

- Diana Chacon
- Jyoti Kumari
- Malachy Moran

Summer, 2021

Berkeley
UNIVERSITY OF CALIFORNIA

Agenda

Introduction
Data
Data Loading, Processing and Exploratory Data Analysis
Models
Results
Conclusion

Introduction

- Study area is located in the Roosevelt National Forest, CO.
- Each observation corresponds to 30m by 30m patch determined from US Forest Service (USFS) Region 2 Resource Information System data.
- Goal is to predict the forest cover type:

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

Data

Training set – 15,120 observations (Features and Cover Type)
Test set – 565,892 observations (Features) ⇨ **Predict Cover Type**
Training Data Fields (56):

- ID
- Elevation
- Aspect
- Slope
- Horizontal_Distance_To_Hydrology
- Vertical_Distance_To_Hydrology
- Horizontal_Distance_To_Roadways
- Hillshade_9am (0 to 255 index)
- Hillshade_Noon (0 to 255 index)
- Hillshade_3pm (0 to 255 index)
- Horizontal_Distance_To_Fire_Points
- Wilderness_Area (4 binary columns, 0 = absence or 1 = presence)
- Soil_Type (40 binary columns, 0 = absence or 1 = presence)
- Cover_Type (7 types, integers 1 to 7)

Data Loading, Processing and EDA

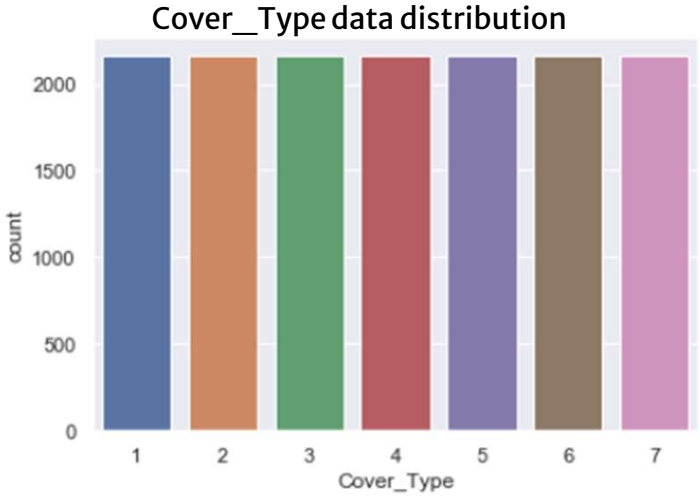
The train and test dataset were loaded

Only train dataset was used

Shuffle data

Split into training data (67%) and development data (33%)

- train_data (13608, 54)
- train_label (13608,)
- dev_data (1512, 54)
- dev_label (1512,)

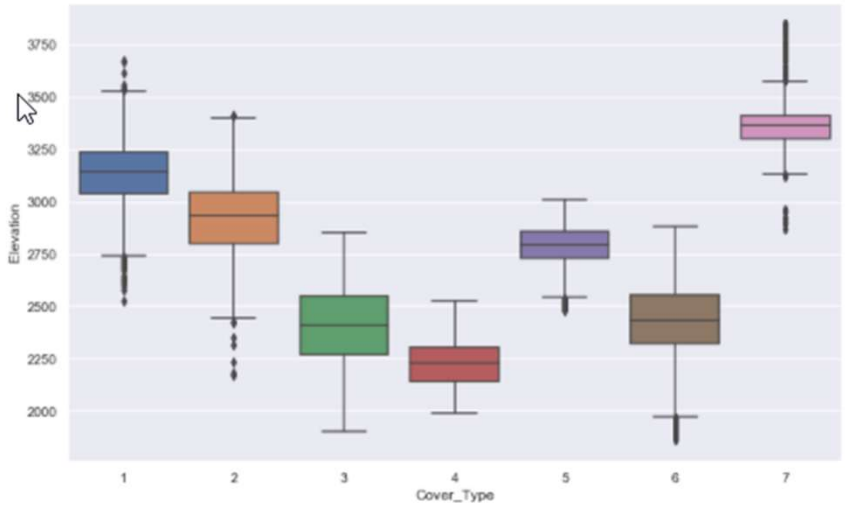


Check for unique values, na values, outliers or miscoded values

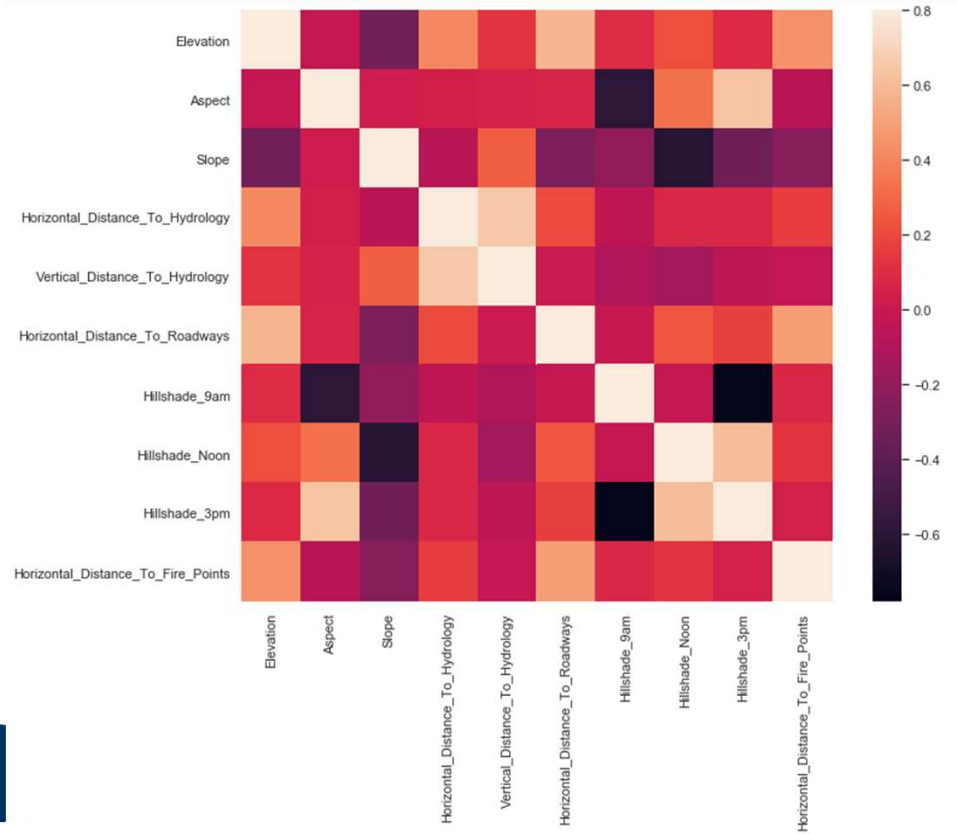
Distribution of 'Cover_Type Data'

Data Loading, Processing and EDA

Check the distribution of the numeric variables across cover types



Check the numeric predictor variables and see which are correlated



Berkeley
UNIVERSITY OF CALIFORNIA

Data Loading, Processing and EDA

Dataset is a combination of continuous and binary features

Data Preprocessing:

- preprocessing.MinMaxScaler
- preprocessing.StandardScaler
- binarize_data_for_tree() function

Models

Classifiers and algorithms for Supervised/unsupervised Learning

Model	Best Accuracy	Train Time(seconds)
K-Nearest Neighbors(kNN)	85%	0.27
Naive Bayes (Bernoulli)	61%	0.05
Logistic Regression	68%	12.98
Stochastic Gradient Descent	66%	0.23
Neural networks	59%	2362.66
Support Vector machine*	85%	658.90

*SVM with non-scaled data performs worst with an accuracy of just 16%

Models

Classifiers and algorithms for Supervised/Unsupervised Learning

Model	Accuracy	Train Time(in seconds)
Single Decision tree	69.0%	0.04
Random Forest	69.8%	9.54
Adaboost	52.2%	555.67
Unsupervised learning		
Gaussian Mixture Model	70.6%	98.72

Decision Tree and Random Forests

3 Models Fit:

- Single Decision Tree
- Random Forest
- Adaboost

Hyperparameter Tuning

Decision Tree:

- GridSearchCV used to find best:
  - Minimum Sample Split
  - Max Depth

Random Forest:

- Decision Tree parameters re-used
- RandomSearchCV used to find best:
  - Number of estimators
  - 15% of 100 options with 2 folds

Adaboost:

- Decision Tree and Random Forest parameters re-used
- RandomSearchCV used to find best:
  - Learning Rate
  - 15% of 100 options with 2 folds

Outcomes

3 Models Fit:

- Single Decision Tree – 68% Accuracy
- Random Forest – 69.6% Accuracy
- Adaboost – 60.9% Accuracy

KNN and Best Fit Model

- KNN Models were fit on three types of data
  - Standardized Data
  - Scaled to Range Data
  - Unaltered Data
- GridSearchCV was used to find the best value for K
- Results:
  - 1 Nearest Neighbor on the unaltered data returned 84.5% accuracy

Conclusion

The highest accuracy:
Model 1 – K Nearest Neighbors and Model 6 – SVM both with an accuracy of 85%.

Although SVM has a higher accuracy than Random Forest, we would still be going ahead and submitting Random Forest Classifier with an accuracy of ~ 70% as our best model.

- Random Forest is intrinsically suited for multiclass problems, while SVM is intrinsically two-class.
- Random Forest works well with a mixture of numerical and categorical features
- For a classification problem, Random Forest gives the probability of belonging to class whereas SVM gives the distance to the boundary


Thank you!