**Forest Cover Type**

Jardin Dantzler, [jdantzler@bellarmine.edu](about:blank)

Caleb Puckett, [cpuckett@bellarmine.edu](about:blank)

**ABSTRACT**

We will conduct some exploratory data analysis to gain some insight into the data provided by the set from UCI. Then, we will employ different machine learning algorithms and attempt to figure out the classification of the Cover\_Type feature. Since we don't have the all the labels of the data in the covtype.data file, as stated earlier we will add them to the header later when conducting more comparisons. To evaluate the performance of our classification models, we will split our data into training and testing/validation subsets. We can use several statistics to judge how well our models predict the testing data by examining confusion matrices, accuracy, precision, recall, and F1 scores.

1. **INTRODUCTION**

The dataset used involves seven “cover types,” or types of trees, within a forest. The main idea is to predict the placement of these trees through machine learning based on many other factors. Some of these factors are elevation, slope, and soil type. This data comes from Roosevelt National Forest in Colorado and is divided into four wilderness areas. The reason for use of these four wilderness areas is based on their minimal exposure to human change. Because of this, the cover type locations and frequency result more from natural ecology than human intervention, giving us some insight into the natural process happening in forests. The goal was to predict how the information we were given related to cover type. We did this through the use of Decision Tree and Random Forest, and we were able to gather some interesting predictions. We found Random Forest to be a better predictor than Decision Tree based on its higher f1-score.

1. **BACKGROUND**

A cover type is referencing what type of tree covers the area of a chosen sample of a mostly unchanged forest environment. This data was originally collected to gain insight on the way trees are naturally placed. Because of the astounding efficiency we find everywhere in nature, it makes sense that we would want to observe all the natural observations we can. Learning from the natural world can help us to better manage our manmade world, in this case helping us with forest management practices.

1. **EXPLORATORY ANALYSIS**

This dataset contains 15,120 samples with 56 columns. For clarification, “Horizontal\_Distance\_to\_Hydrology” means the horizontal distance to the nearest surface water features. Subsequent distance column names are similarly defined. “Hillshade\_9am,” as well as for noon and 3pm, tells us the shade of the area based on geographical surroundings at the time noted in the name. It should be noted that these data points are all taken on the summer solstice for consistency. A brief note on the data types: There are forty different columns for “Soil\_Type,” labeled “Soil\_Type1” through “Soil\_Type40,” as well as four columns for “Wilderness\_Area,” labeled “Wilderness\_Area1” through “Wilderness\_Area4.” These were generalized int the table above for the sake of avoiding redundancy. All of these contain the same data type: Ratio/int64.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| Elevation | float64 |
| Aspect | float64 |
| Slope | float64 |
| R\_Hydrology | float64 |
| Z\_Hydrology | float64 |
| R\_Roadways | float64 |
| Hillshade\_9am | float64 |
| Hillshade\_Noon | float64 |
| Hillshade\_3pm | float64 |
| R\_Fire\_Points | float64 |

*When initially running through our data there were no missing values but when during our distribution.* *We found that we had 7 classes out of which 2 classes (cover type 1 and 2) form 85% of the data. There are 2 classes (cover type 4 and 5) with less than 2% representation. Therefore, the dataset is heavily skewed, and this would be accounted for when creating a model (thus the support vector machine and sampling).*

**METHODS**

1. *Data Preparation*

After calling in the dataset, we decided to implement the categorical variables to pass them in our comparisons. From there we normalized the data and began the setup for the SVM. We ran through a sampleto train the Support Vector Machine Classifier and validate that the sample is representative of the data. Then of course after receiving our accuracy we proceed with the Random Forest Classifier and the Decision Tree.

1. *Experimental Design*

Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | All four (4) raw features with 80/10/10 split for train, validate, and test |
| 2 | All four (4) normalized features with 80/10/10 split for train, validate, and test |
| 3 | All four (4) normalized features with 70/15/15 split for train, validate, and test |
| 4 | Data with normalized features, 80/10/10 split for train, validate, and test |

1. *Tools Used*

This project was done using supervised machine learning techniques from sklearn to work on the classification and prediction of the forest coverage type based on data given. Along with a .docx for a more in-depth breakdown of our project and lines of code. The dataset exists on the UCI machine learning repository: [[https://archive.ics.uci.edu/ml/datasets/covertype](about:blank)] Machine learning techniques used: Anaconda for Mac ver.2.4.1, Sklearn ver.2.2.2 Support Vector Machine, Decision Tree, and Random Forest Classifier. Primary python ver.3.9.7 libraries used were Visualizations including matplotlib, and seaborn. Data analysis including NumPy, and pandas. The Machine learning algorithm used was ran through sklearn and our model performance analysis was running through sklearn as well.

1. **RESULTS**
   1. *Classification Measures*

Provide the classification measures for each experiment. For example, you could provide a contingency table for each model to measure how well it classifies data. You could also do an ROC curve (using SciKit Learn). You need to demonstrate how you are measuring the success/failure of the models.

* 1. *Discussion of Results*

We think the decision was our worst model and not just due to it being the lowest. We understand that decision tree doesn’t usually won’t effectively run well with imbalanced. Even from our SVM by passing the class\_weight='balanced'. It would still possibly pass through the bits of what was part of the data that was originally skewed. For our best model. It would be the Random Forest. Since we used a smaller sample of the data it was able to run more effectively instead of passing well everything. So, with that model just running the estimated amount it was able to showcase an area that was more balanced.

* 1. *Comparison of Models*

**Figure 1: Histogram**

Chart

Description automatically generated

**Figure 2: Heatmap**

Chart

Description automatically generated

**Figure 3: Scatterplot**

A picture containing table

Description automatically generated

**Figure 4: Wilderness Area Distribution**

Chart, waterfall chart

Description automatically generated

**Figure 5: Soil Type Distribution**

**Chart

Description automatically generated**

Text

Description automatically generated with low confidence

1. **RESULTS**
   1. *Classification Measures*

**Decision Tree Result 1:**

Chart

Description automatically generated

**Decision Tree Results 2:**

**Chart

Description automatically generated**

**Random Forest Results 1:**

**Chart

Description automatically generated with medium confidence**

**Random Forest Results 2:**

A picture containing graphical user interface

Description automatically generated

* 1. *Discussion of Results*

Discuss which of your models provided the best classification (or some other outcome if not classification). Explain why you think your best model was the best and why your worst model was the worst.

We believe the Random Forest model was the best because it had the highest f1 score. Alternatively, because of its low f1 score, we found Decision Tree to be worse than Random Forest. We used f1-score as a measure of success because this is the combination of precision and recall. For this reason, it is a good general measure of success.

* 1. *Comparison of Models*

We think the decision was our worst model and not just due to it being the lowest. We understand that decision tree doesn’t usually won’t effectively run well with imbalanced. Even from our SVM by passing the class\_weight='balanced'. It would still possibly pass through the bits of what was part of the data that was originally skewed. For our best model. It would be the Random Forest. Since we used a smaller sample of the data it was able to run more effectively instead of passing well everything. So, with that model just running the estimated amount it was able to showcase an area that was more balanced.

* 1. *Problems Encountered*

Our first issue happened when starting the distribution comparisons as the wild\_areas and \_soil\_types were categorical variables in comparison the continuous variables initially. So, we had to go through and figure out how we wanted to implement those without over-skewing our data even more. Another issue we ran into was we were struggling to figure out how to balance our data. Even, when Jardin did a run through without including the wild\_areas and soil\_types there was a bit of a difference between the accuracies, but it still seemed a bit low even with the data being cleaner for the decision tree.

* 1. *Limitations of Implementation*

With our data being as imbalanced as it was, we thought the decision tree would be able to provide a more accurate showing however, with us adding in the wild\_areas and soil\_types it was added on to the skewed dataset.

* 1. *Improvements/Future Work*

We would like to further pursue correcting the imbalanced data and then rerunning it through different samplings. More than likely using xgboost or logistic regression to balance out those classes.

1. **CONCLUSION**

Our goal in this project was to implement forms of machine learning to a dataset about forest cover type. We decided it would be best to use Random Forest and Decision Tree as our two processes of machine learning. To begin this process, we had to clean the data. This was important for making sure we could operate smoothly on the data as needed. The next step was finding distribution of the data based on the columns we were given. This was a helpful step to better understand the dataset before attempting to go further. Then, we normalized and trained the data in preparation for running the Random Forest and Decision Tree models. These were fairly successful, Random Forest being a better predictor than Decision Tree, though there is definitely room for improvement in both. In the future it could be good to experiment with use of other models.

**REFERENCES**

Basic Inspiration for notebook outline - [https://towardsdatascience.com/predicting-forest-cover-types-with-the-machine-learning-workflow-1f6f049bf4df](about:blank)

Imbalanced Data - [https://github.com/scikit-learn-contrib/imbalanced-learn](about:blank)

Min/Max Normalization - [https://towardsdatascience.com/everything-you-need-to-know-about-min-max-normalization-in-python-b79592732b79](about:blank)

Support Vector Machine - [https://scikit-learn.org/stable/modules/svm.html](about:blank)

Grid Search CV for SVM - [https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/](about:blank)

*Division of Labor*

Jardin:

Cleaning Data

Training Data

Markdowns

Outlining distribution

Part of word doc

Caleb:

Distribution graphs

Decision Tree

Random Forest

Part of Word document

In this section after you references, provide a paragraph or two outlining what each team member did on this project. Please don’t tell you both did everything! Be honest about who did what.

**Other directions:**

1. 10-pt, Times New Roman, 1” margins all around (if you use this template you are already set).
2. **Ensure all tables and figures are numbered appropriately and referenced in the text.** See examples above and below.

|  |  |
| --- | --- |
| **Figure 1: Comparison of X/Y from dataset (single plot) (8 pt.)** | **Figure 2: (a) Function Output (b) A against B (multiple plots) (8 pt.)** |