**MACHINE**

**LEARNING**

**PROJECT**

**REPORT**

Diagram

Description automatically generated

Prepared By – Dishit Vasani – N01440553

**Data Set Information**

* **Context**

This dataset contains tree observations from four areas of the Roosevelt National Forest in Colorado. All observations are cartographic variables (no remote sensing) from 30-meter x 30-meter sections of the forest. There are over half a million measurements in total.

* **Content**

This dataset includes information on tree type, shadow coverage, distance to nearby landmarks (roads etcetera), soil type, and local topography.

* **Acknowledgment**

This dataset is part of the UCI Machine Learning Repository, and the original source can be found at https://archive.ics.uci.edu/ml/datasets/Covertype . The original database owners are Jock A. Blackard, Dr. Denis J. Dean, and Dr. Charles W. Anderson of the Remote Sensing and GIS Program at Colorado State University.

* **Inspiration**
* Can you build a model that predicts what types of trees grow in an area based on the surrounding characteristics?
* What kinds of trees are most common in the Roosevelt National Forest?
* Which tree types can grow in more diverse environments? Are there certain tree types that are sensitive to an environmental factor, such as elevation or soil type?



**Unsupervised Learning**

Unsupervised learning is a type of machine learning where the users don't have to watch over the model. Instead, it enables the model to operate independently and find previously overlooked patterns and information. It mostly addresses unlabeled data.

**Unsupervised Learning Algorithms**

Compared to supervised learning, unsupervised learning algorithms enable users to carry out more complicated processing tasks. Unsupervised learning, however, can be more unpredictable than other types of natural learning. Algorithms for unsupervised learning include neural networks, anomaly detection, and grouping.

**Clustering**

In terms of unsupervised learning, clustering is a key idea. The primary focus is on identifying a structure or pattern in a set of uncategorized data. Unsupervised Education If there are any natural clusters or groupings in your data, clustering algorithms will process them and locate them. You can alter the number of clusters your algorithms should find as well. You can change the level of granularity for these categories.

**K-Means Clustering**

It is an iterative clustering algorithm, denoted by the letter K, that aids in locating the highest value for each iteration. The desired number of clusters is initially chosen. You must divide the data points into k groups to use this clustering technique. In the same way, a bigger k results in smaller groups with more granularity. Less granularity and larger groups are the results of a lower k.

The algorithm produces a collection of "labels" as its result. A data point is assigned to one of the k groups. Each group in k-means clustering is identified by the creation of a centroid for that group. The centroids act as the cluster's "heart," capturing and incorporating the nearby points into the whole.

**Principal Components Analysis**

In case you want a higher-dimensional space. You need to select a basis for that space and only the 200 most important scores of that basis. This base is known as a principal component. The subset you select constitute is a new space which is small in size compared to original space. It maintains as much of the complexity of data as possible.

**Notebook**

In this notebook we will be exploring method that can be used to visualize clusters that were formed on high-dimensional data (data with more than three dimensions).

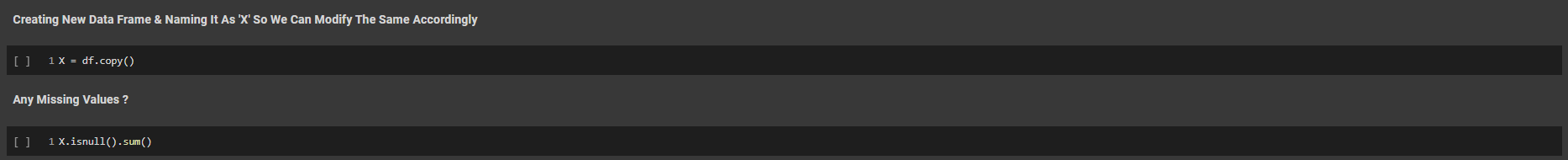
First, we will clean our data so that it's in a proper format for clustering, then, we will divide the data into three different clusters using K-Means Clustering. After that, we will go ahead and visualize our three clusters using method: Principal Component Analysis (PCA).

*- Imports*

Background pattern

Description automatically generated

*- Creating New Data frame to work on & checking Null Values*



- *If we look at the columns: X["Horizontal\_Distance\_To\_Hydrology"] and X[Vertical\_Distance\_To\_Hydrology"], we see that we can create from them, a new column X[Distance\_To\_Hydrology], which measures the shortest distance to Hydrology. We can calculate the values of this column through using the equation from the Pythagorean Theorem.*

*-Now that we have X["Distance\_To\_Hydrology"], and because there's nothing extra special about Vertical or Horizontal Distances to Hydrology, we can drop the original two columns:*

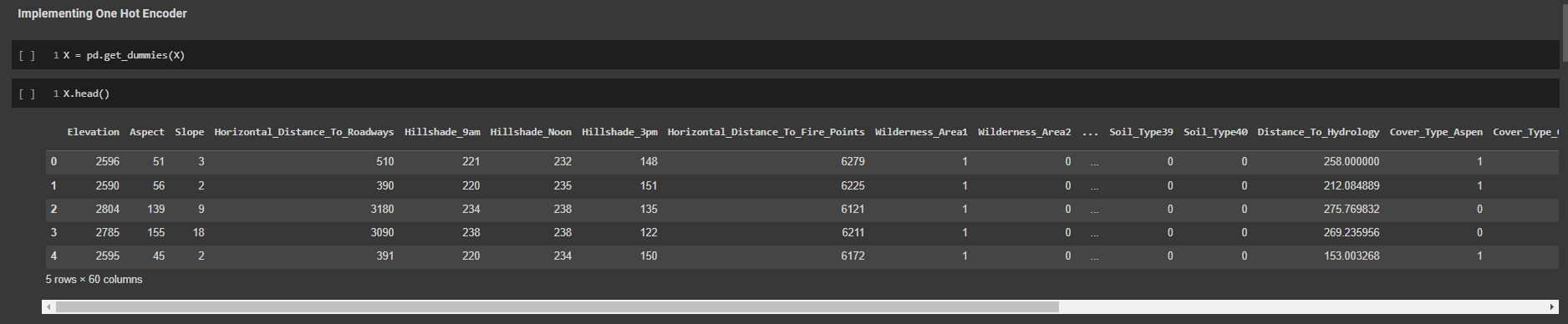
A screenshot of a computer screen

Description automatically generated with medium confidence

*-if we look at the values contained within X['Cover\_Type'], you'll notice that it contains numerically encoded categorical data. If we head over to the column descriptions on the Forest Cover Type Dataset page, it says that:*

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

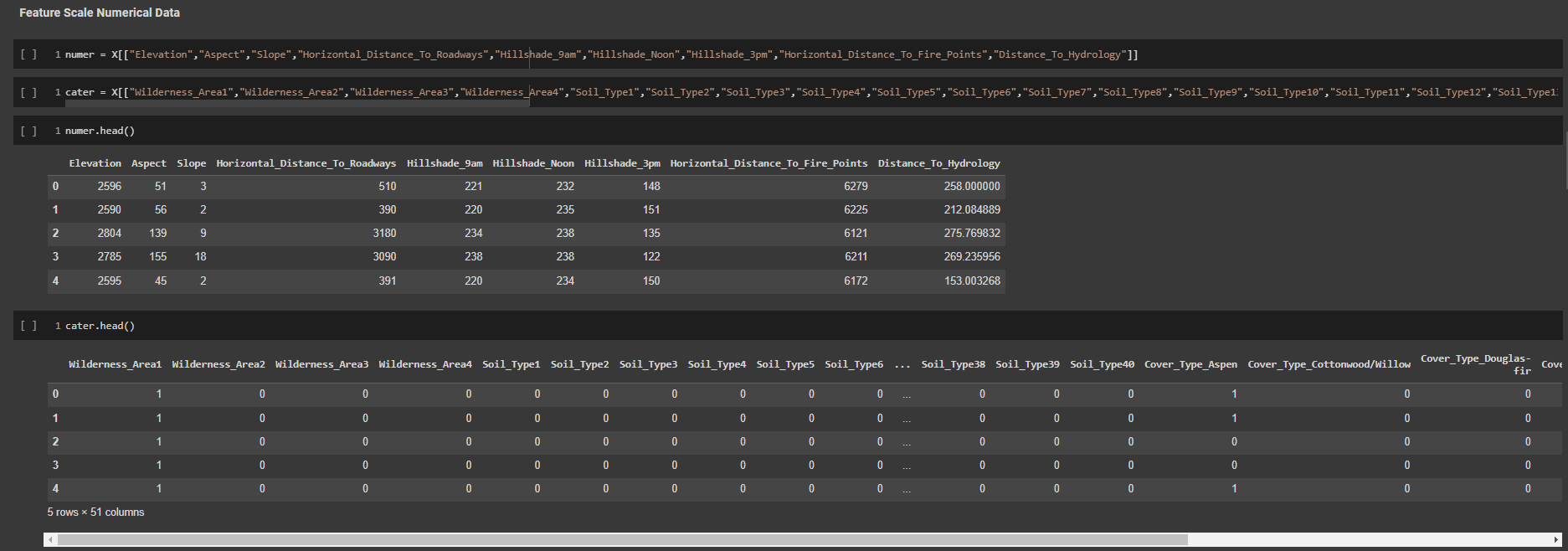
*-We'll relabel our data so that the values in X['Cover\_Type'] are more descriptive of what's really contained within it. We'll also do it so that we can easily apply a one-hot-encoding to it, afterwards - so that X['Cover\_Type'] will be properly encoded along with the rest of the categorical data in X.*



*- We need to do this because, while each of our categorical variables hold values of either 0 or 1, some of our numerical variables hold values like 2596 and 2785. If we were to leave our data like this, then K-Means Clustering would not give us such a nice result, since K-Means Clustering measures the Euclidean distance between data-points. This means that, if I were to leave our numerical variables un-scaled, then most of the distance measured between points would be attributed to the larger numerical variables, rather than any of the categorical variables.*

*-To fix this problem we will scale all our numerical variables using sklearn's StandardScaler tool. This tool allows us to scale each numerical variable such that each numerical variable's mean becomes 0, and its variance becomes 1. This is a good way to make sure that all the numerical variables are on roughly the same scale that the categorical (binary) variables are on.*

*-But, to make sure we scale only our numerical variables -- and not our categorical variables --, we'll split our current Data Frame, X, into two other Data Frames: numer and cater; feature-scale. numer, then recombine the two Data Frames together again into a Data Frame that is suitable for clustering.*

**

*-Feature Scale*

*Graphical user interface

Description automatically generated with medium confidence*

*-Time to build Clusters, I will be visualizing only three different clusters on our data. I chose three because I found it to be a good number of clusters to help us visualize our data in a non-complicated way.*

*Background pattern

Description automatically generated with medium confidence*

*-* *PCA is an algorithm that is used for dimensionality reduction - meaning, informally, that it can take in a Data Frame with many columns and return a Data Frame with a reduced number of columns that still retains much of the information from the columns of the original Data Frame. The columns of the Data Frame produced from the PCA procedure are called Principal Components. I would use these principal components to help visualize the clusters in 1-D, 2-D, and 3-D space, since I cannot easily visualize the data, we have in higher dimensions. For example, we can use two principal components to visualize the clusters in 2-D space, or three principal components to visualize the clusters in 3-D space.*

**

*-* *1-D Visualization*

*A picture containing application

Description automatically generated*

*-2-D Visualization*

*Chart, text, scatter chart

Description automatically generated*

*-3-D Visualization*

*Chart

Description automatically generated*

*-Weka*

*Graphical user interface, application

Description automatically generated*

*Text

Description automatically generated*

*Text

Description automatically generated*

*Graphical user interface, application

Description automatically generated*

**Conclusion**

*-As we can see from the plots above: if you have data that is highly cluster able, then PCA is a pretty good way to view the clusters formed on the original data. Also, visualizing the clusters is more effective when the clusters are visualized using more principal components, rather than less. For example, the 2-D plot did a better job of providing a clear visual representation of the clusters than the 1-D plot; and the 3-D plot did a better job than the 2-D plot!*

*-Comparing the result set with WEKA we would observe a great variation as the Data we selected for WEKA contained only around 5000 Instances due to Machine limitations.*

**Supervised Machine Learning**

Supervised Machine Learning is an algorithm that learns from labeled training data to help you predict outcomes for unforeseen data. In Supervised learning, you train the machine using data that is well “labeled.” It means some data is already tagged with correct answers. It can be compared to learning in the presence of a supervisor or a teacher.

**Random Forest**

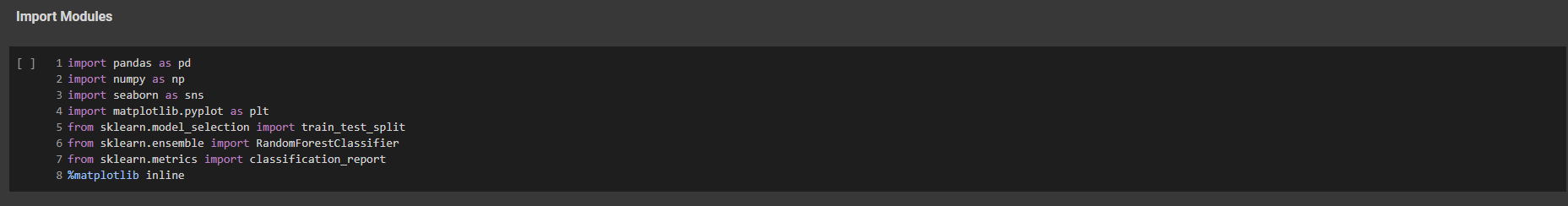
Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

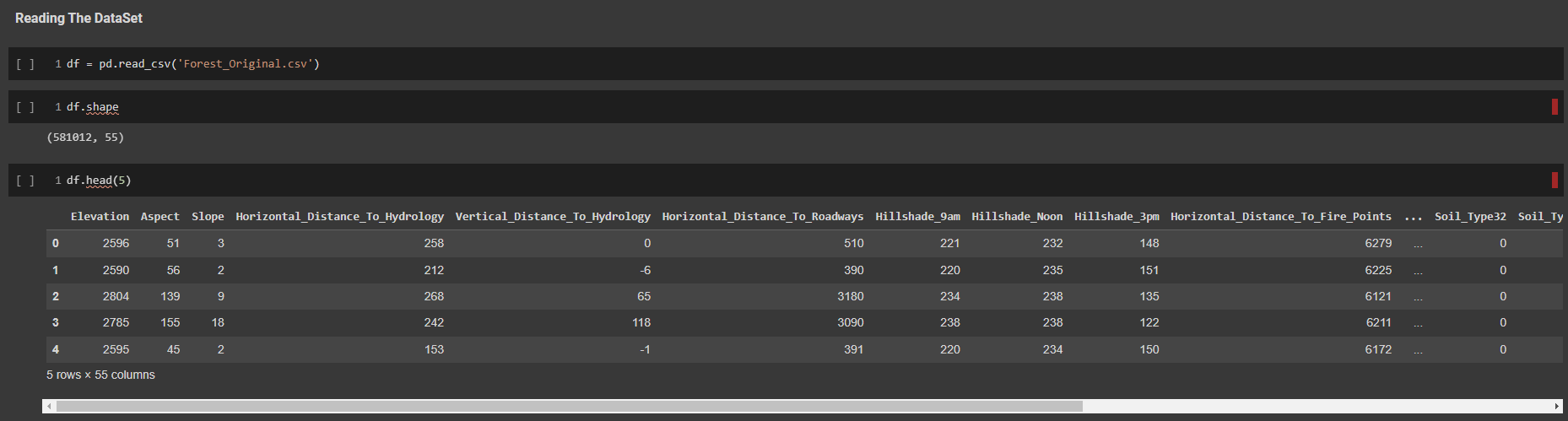
**Notebook**

In this Notebook we are going to Implement Random Forest Algorithm and compare the same result with the output produced by WEKA.

*- Imports*



-*Reading the Dataset*

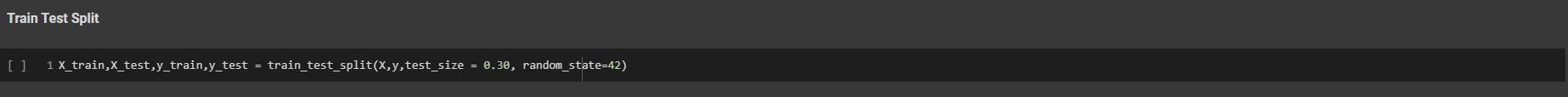


*-Model*

Background pattern

Description automatically generated

*-Train Test Split*



-*Random Forest Classifier*

A screenshot of a computer

Description automatically generated with medium confidence

*-Weka*

*A picture containing graphical user interface

Description automatically generated*

*A picture containing chart

Description automatically generated*

**Conclusion**

* *As we can see both the outputs Running Random Forest on Python gave us accuracy of 95.21% here, we took the whole Dataset for better evaluation. On the other hand running the Random Forest on WEKA fetched us with accuracy of 93.69%, in here we ran the algorithm on fewer instances due to machine incompatibility. Then to there was not much difference in their accuracy score.*

**References**

* <https://www.guru99.com/r-random-forest-tutorial.html>
* <https://www.kaggle.com>
* <https://www.geeksforgeeks.org/k-means-clustering-using-weka/>
* <https://ikuz.eu/csv2arff/>
* <https://www.guru99.com/unsupervised-machine-learning.html>
* <https://archive.ics.uci.edu/ml/datasets.php>