   
   FINAL PROJECT REPORT

**FOREST COVER TYPE PREDICTION**

**Post Graduate Program in Data Science Engineering**

Location: **Hyderabad** Batch: **PGPDSE-FT-H-July23**

**Submitted by**

Batchu Heam chandu

S Esha Yadav

Yash Vardhan Khetawat

Mamidi Reshma

V.Sriram

Vishnu Vamsi

**Mentored by**

Mr. Animesh Tiwari

Table Of Content

1. Industry review
   1. Overview
   2. Current practice
   3. Background research
2. Dataset and domain
   1. Domain
   2. Data characteristics
   3. Problem statement
   4. Details of wilderness areas
   5. Details of forest cover type classes
   6. Details of soil types
3. Data exploration
   1. Univariate analysis
   2. Bivariate analysis
4. Data pre- processing
   1. Data cleaning
   2. Scaling the Features for Model Optimization
   3. Outlier treatment approach
5. Model building
   1. Base Model
   2. Models using Power Transformers
   3. Performing Hyper Parameter Tuning on the models
   4. Pruning the models
   5. Using Boosting techniques
   6. Stacking
6. Conclusion
7. References

1. **INDUSTRY REVIEW**

**1.1 Overview:**

The project focuses on predicting forest cover types using a dataset with 55 features, including 11 quantitative variables, 4 binary variables for Wilderness Area, and 40 binary variables for Soil Type. The dataset encompasses diverse aspects such as elevation, slope, distance to hydrology, and more. The target variable is the forest cover type, categorized into seven classes. The objective is to develop a predictive model that accurately classifies the forest cover type based on the given features. The project involves exploratory data analysis, statistical parameterization, and the application of machine learning algorithms to achieve accurate and efficient predictions. The categorical features, including Wilderness Area and Soil Type, offer additional complexity to the prediction task. The project aims to contribute insights into forest cover dynamics, aiding in ecological studies and sustainable forest management.

**1.2 Current Practices:**

In the field of remote sensing for forest monitoring, current practices involve the utilization of various Earth observation satellites and technologies. Satellite programs like Landsat, Copernicus, and MODIS are widely employed to provide spatial and temporal observations of forest characteristics at landscape and regional scales. Instruments such as light detection and ranging (LiDAR) and hyperspectral sensors are frequently used to quantify forest characteristics at stand to landscape levels.

**1.3 Background Research:**

The background research in remote sensing for forest monitoring has evolved with innovations in technology and computing methods. Over the last few decades, there has been a continuous improvement in forest monitoring efforts, driven by the need for effective management of forest resources. The research includes the development and application of statistical and machine-learning models derived from plot-level field observations, which are extrapolated to larger areas using remote sensing data.

1. **Dataset and Domain**

**Domain:**

The goal of this project is to predict the forest cover type, specifically the predominant kind of tree cover, using cartographic variables. This data is obtained from the US Geological Survey (USGS) and the US Forest Service (USFS) which is in open domain and includes four wilderness areas located in Roosevelt National Forest of northern Colorado and provided by Machine Learning Laboratory of University of California Irvine.

**Data Characteristics:**

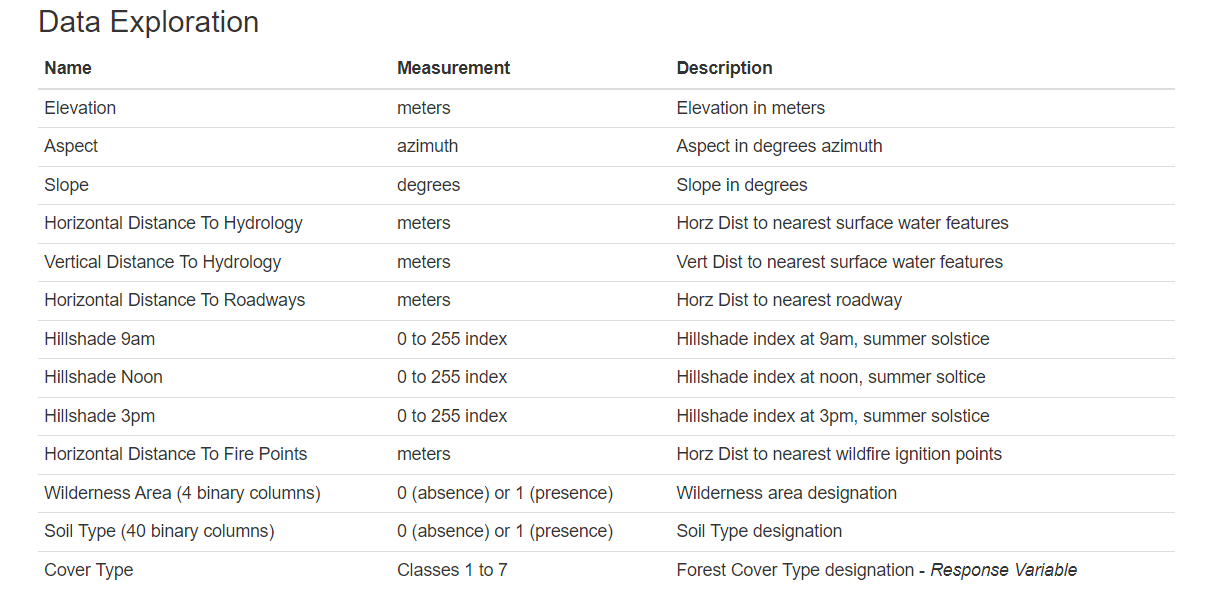
* **Raw Form:** The data is in its raw form and has not been scaled. This implies that feature scaling may be necessary during the preprocessing stage.
* **Qualitative Independent Variables:** The dataset contains binary columns representing qualitative independent variables, such as wilderness areas and soil types.

**Study Area:**

* **Location:** The study area is the Roosevelt National Forest in northern Colorado.
* **Wilderness Areas:** There are four wilderness areas in the study, representing forests with minimal human-caused disturbances. The existing forest cover types are more a result of ecological processes than forest management practices.

**Problem Statement:**

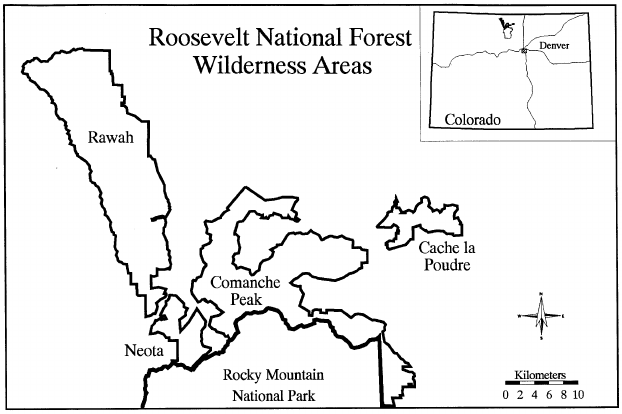
Develop an accurate predictive model for classifying seven different land cover types in the Roosevelt National Forest's four wilderness areas in northern Colorado. Each observation corresponds to a 30m x 30m patch. The goal is to enhance understanding of the region's ecological dynamics for sustainable forest management and environmental conservation.



**Name Data Type Measurement Description**

* Elevation quantitative meters Elevation in meters
* Aspect quantitative azimuth Aspect in degrees azimuth
* Slope quantitative degrees Slope in degrees
* Horizontal\_Distance\_To\_Hydrology quantitative meters Horz Dist to nearest surface water features
* Vertical\_Distance\_To\_Hydrology quantitative meters Vert Dist to nearest surface water features
* Horizontal\_Distance\_To\_Roadways quantitative meters Horz Dist to nearest roadway
* Hillshade\_9am quantitative 0 to 255 index Hillshade index at 9am, summer solstice
* Hillshade\_Noon quantitative 0 to 255 index Hillshade index at noon, summer soltice
* Hillshade\_3pm quantitative 0 to 255 index Hillshade index at 3pm, summer solstice
* Horizontal\_Distance\_To\_Fire\_Points quantitative meters Horz Dist to nearest wildfire ignition points
* Wilderness\_Area (4 binary columns) qualitative 0 (absence) or 1 (presence) Wilderness area designation
* Soil\_Type (40 binary columns) qualitative 0 (absence) or 1 (presence) Soil Type designation
* Cover\_Type (7 types) integer 1 to 7 Forest Cover Type designation

**Details of Wilderness Areas:**



|  |  |
| --- | --- |
| Wilderness\_Area1 | Rawah Wilderness Area |
| Wilderness\_Area2 | Neota Wilderness Area |
| Wilderness\_Area3 | Comanche Wilderness Area |
| Wilderness\_Area4 | Cache La Poudre Wilderness Area |

**Background Information on the Four Wilderness Areas:**

Neota (Area 2) likely has the highest mean elevation, primarily featuring spruce/fir. Rawah (Area 1) and Comanche Peak (Area 3) have a lower mean elevation, with lodgepole pine as the primary species. Cache la Poudre (Area 4) has the lowest mean elevation, characterized by Ponderosa pine, Douglas-fir, and cottonwood/willow. Rawah and Comanche Peak represent the overall dataset, while Neota and Cache la Poudre stand out due to unique features like elevation range and species composition. 

**Details of Forest Cover Type Classes:**



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

The study area has different types of forests, each with its own kinds of trees and plants. Some areas have spruce and fir trees, others have lodgepole pine, and some have a mix of spruce, fir, and aspen. There are also places with Ponderosa pine, Douglas-fir, and cottonwood/willow. The differences in elevation and the types of trees make each forest area unique and affect the plants and animals that live there.

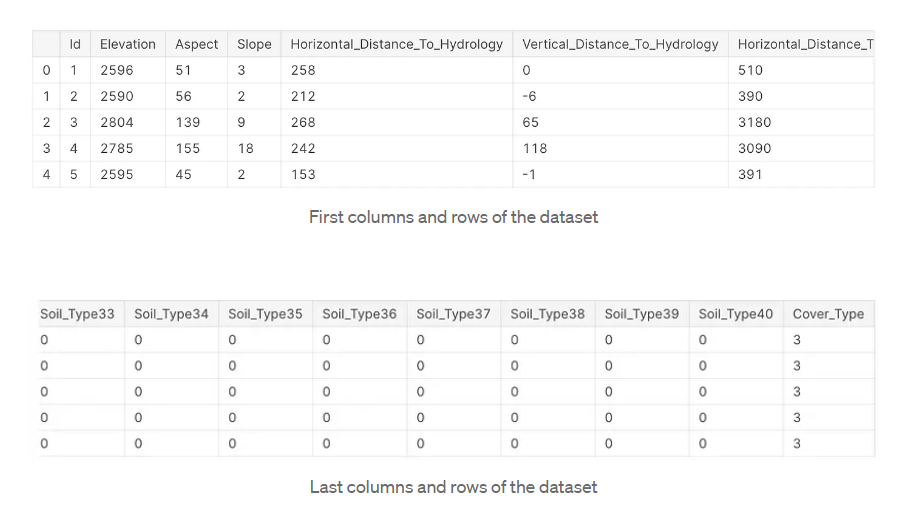
**Details of Soil Types:**

|  |  |
| --- | --- |
| 1 | Cathedral family - Rock outcrop complex, extremely stony |
| 2 | Vanet - Ratake families complex, very stony |
| 3 | Haploborolis - Rock outcrop complex, rubbly |
| 4 | Ratake family - Rock outcrop complex, rubbly |
| 5 | Vanet family - Rock outcrop complex, rubbly |
| 6 | Vanet - Wetmore families - Rock outcrop complex, stony |
| 7 | Gothic family |
| 8 | Supervisor - Limber families complex |
| 9 | Troutville 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 |

The study area features diverse soil types, from extremely stony rock outcrop complexes (Cathedral, Vanet) to rubbly ones (Haploborolis, Ratake). Complexes like Gothic and Supervisor-Limber contribute to the soil diversity, ranging from extremely stony (Leighcan, Granile-Catamount) to very stony (Rogert). The warm and extremely stony Leighcan family stands out. Various complexes like Cryaquolis and Cryumbrepts add to the intricate soil landscape. Understanding these soil variations is crucial for effective ecological and land management in the area.

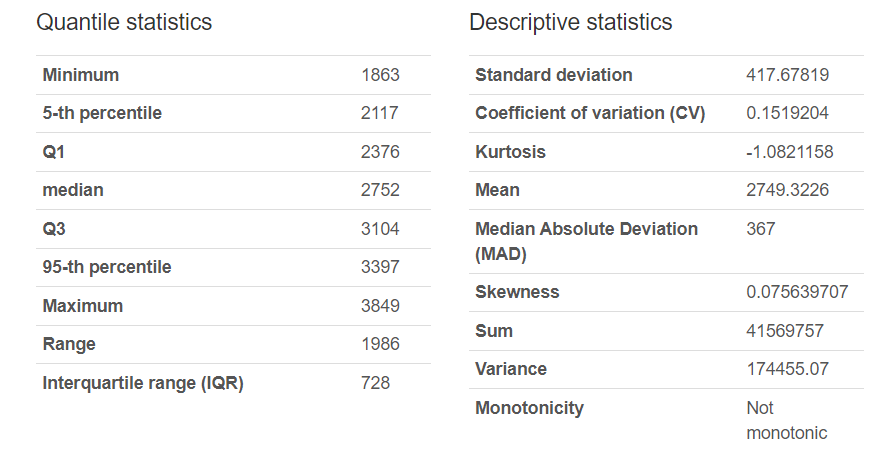
1. **DATA EXPLORATION**

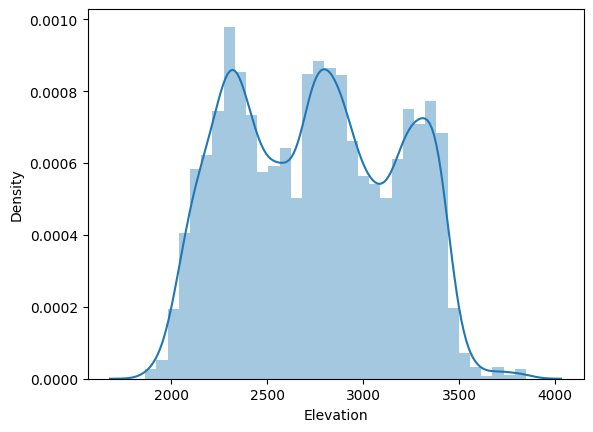
**Preview of Dataset: Top 5 Rows**



* 1. **UNIVARIATE ANALYSIS:** Numerical variables

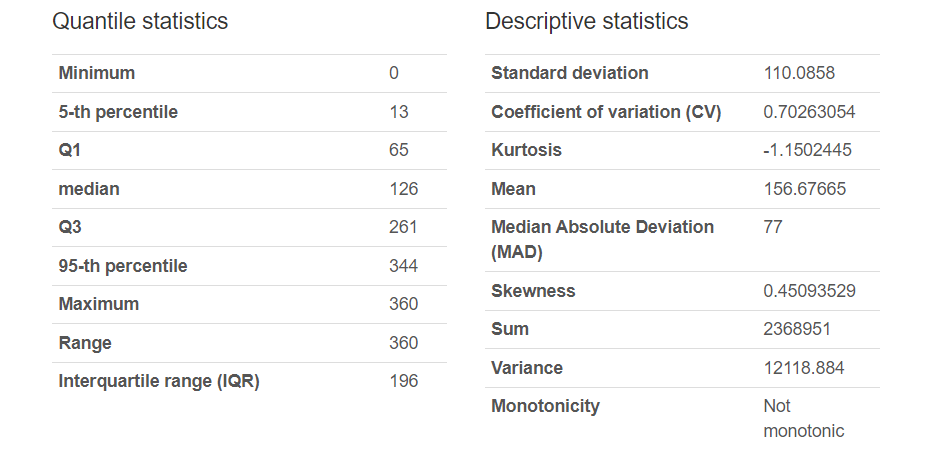
1. **ELEVATION:**

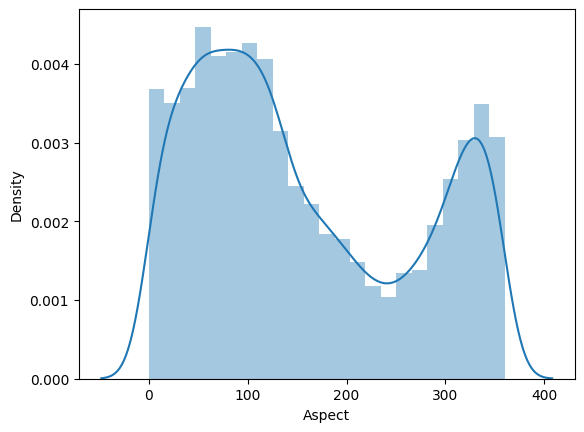




Here we say that from descriptive statistics, the variable ‘Elevation’ column has normal distribution with minimal skewness is +0.08 and kurtosis is -1.08. Therefore, we can say that it is normally distributed and platykurtic in nature.

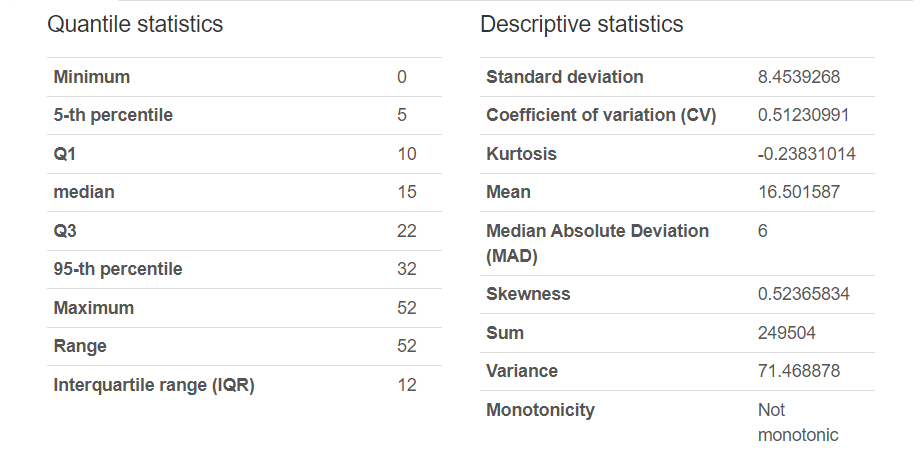
1. **ASPECT:**

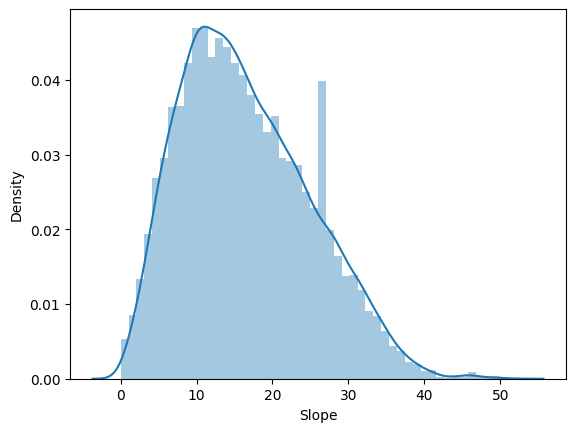




As we can see from descriptive statistics, the variable ‘Slope’ has a skew value of +0.45 and kurtosis value of -1.15. Therefore, we can say that it is normally distributed and platykurtic in nature.

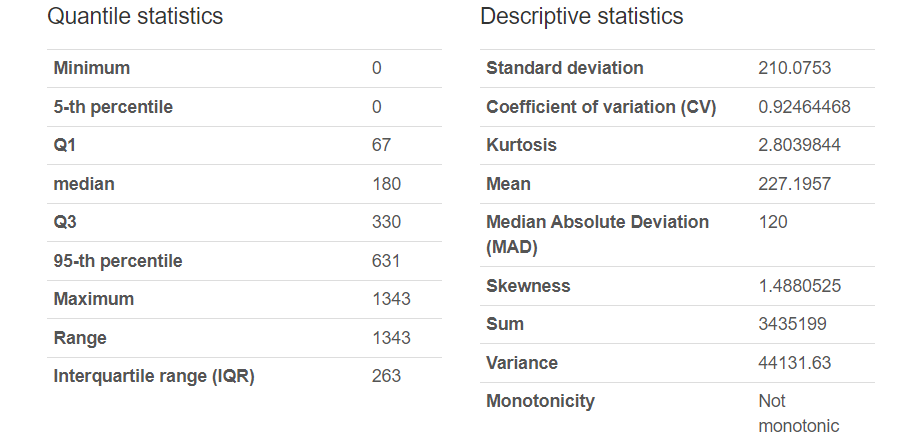
1. **SLOPE:**

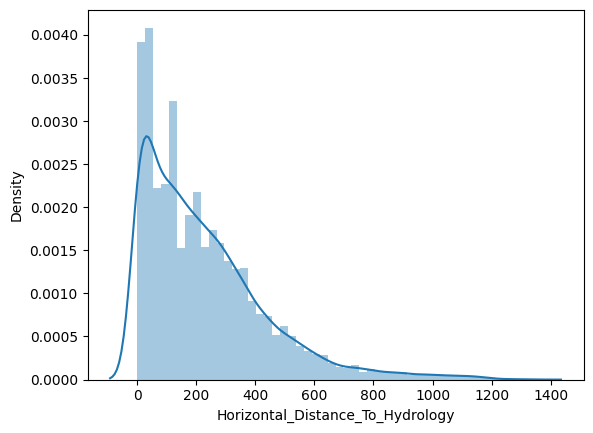




As we can see from descriptive statistics, the variable ‘Slope’ has a skew value of +0.52 and kurtosis value of -0.238. Therefore, we can say that it is normally distributed and platykurtic in nature.

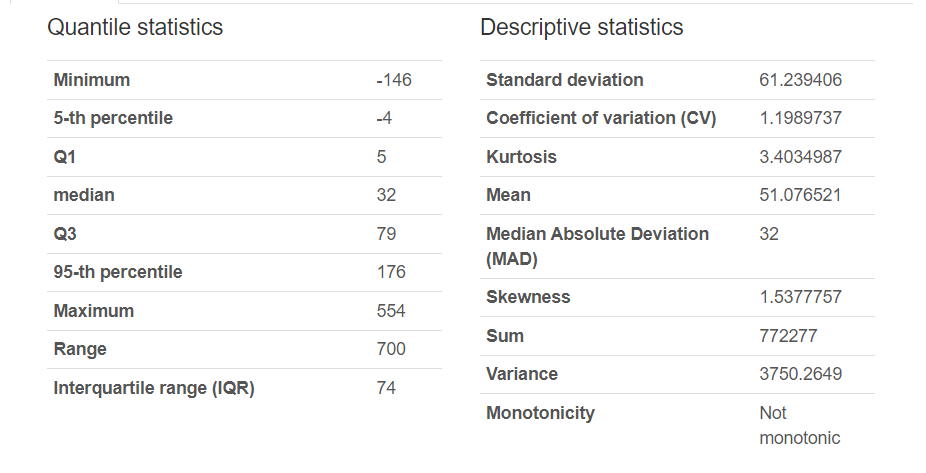
1. **HORIZONTAL DISTANCE TO HYDROLOGY:**

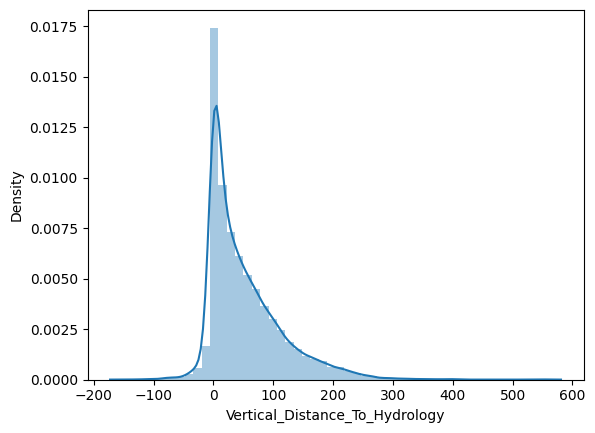




As we can see from descriptive statistics, the variable ‘Slope’ has a skew value of +1.488 and kurtosis value of +2.803. Therefore, we can say that it is positively skewed and leptokurtic in nature.

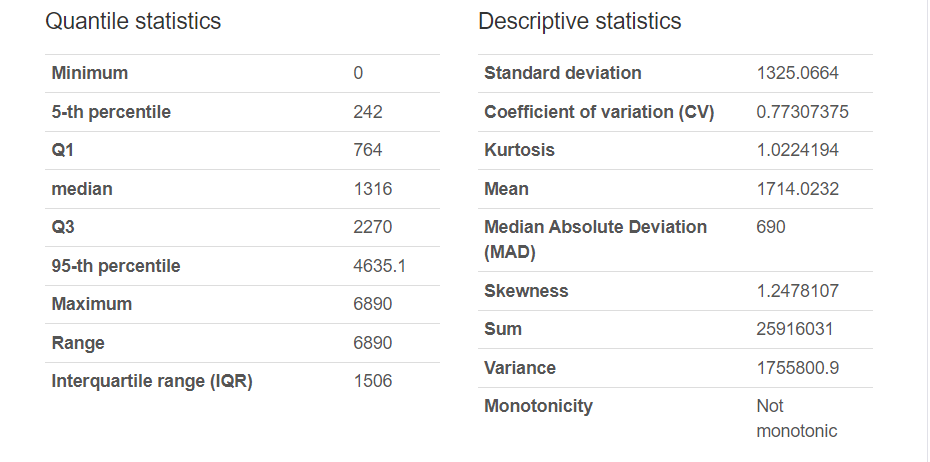
1. **VERTICAL DISTANCE TO HYDROLOGY:**

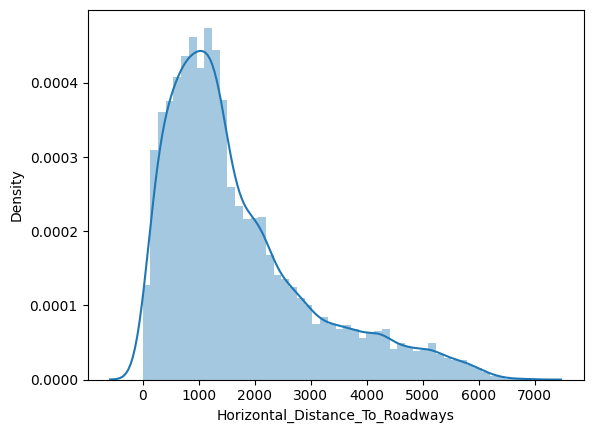




As we can see from descriptive statistics, the variable ‘Vertical\_Distance\_To\_Hydrology’ has a skew value of +1.53 and kurtosis value of +3.40. Therefore, we can say that it is positively skewed and leptokurtic in nature.

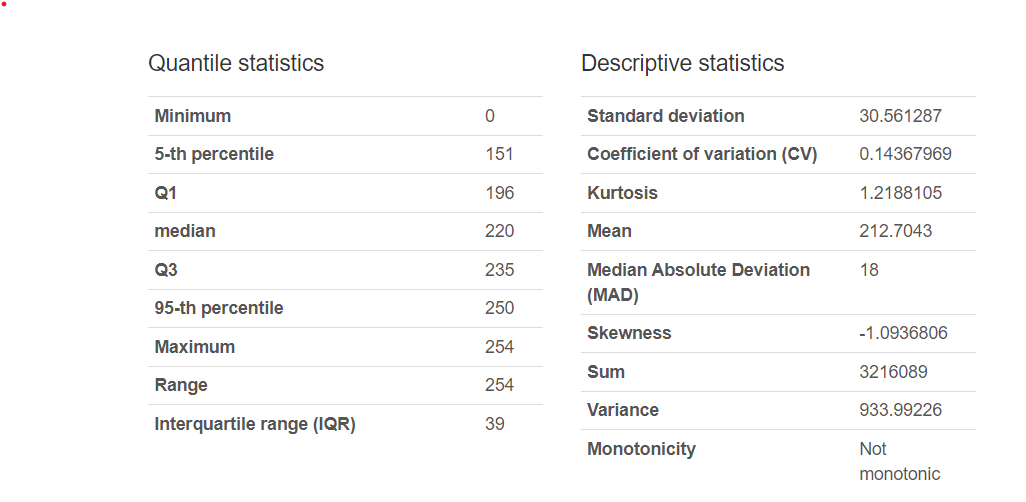
1. **HORIZONTAL DISTANCE TO ROADWAYS:**

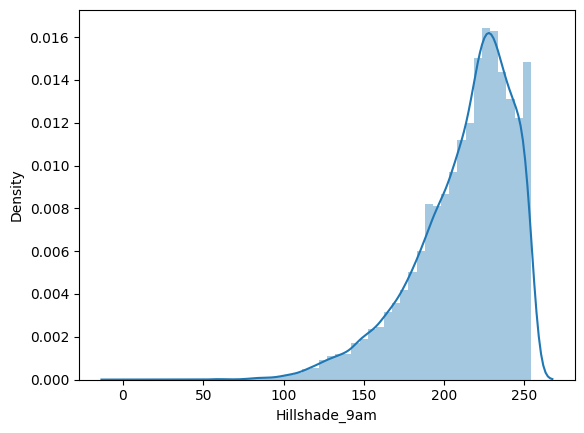




As we can see from descriptive statistics, the variable 'Horizontal\_Distance\_To\_Roadways' has a skew value of +1.24 and kurtosis value of +1.022. Therefore, we can say that it is positively skewed and leptokurtic in nature.

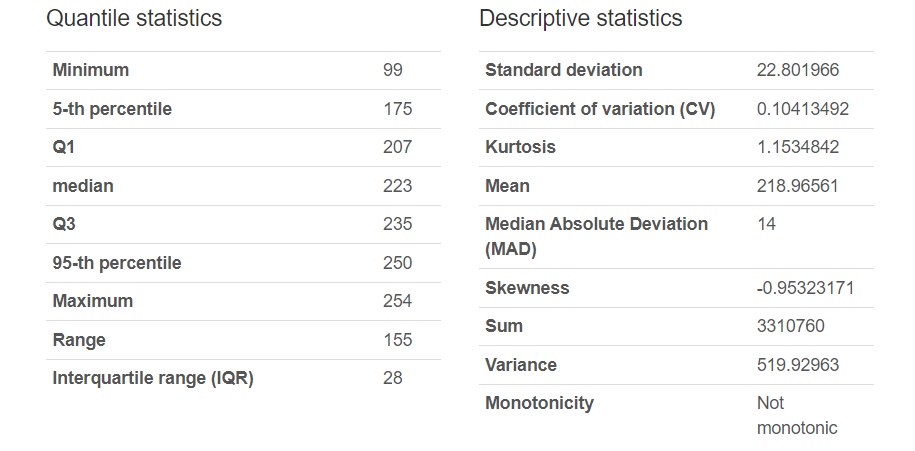
1. **HILLSHADE\_9AM:**

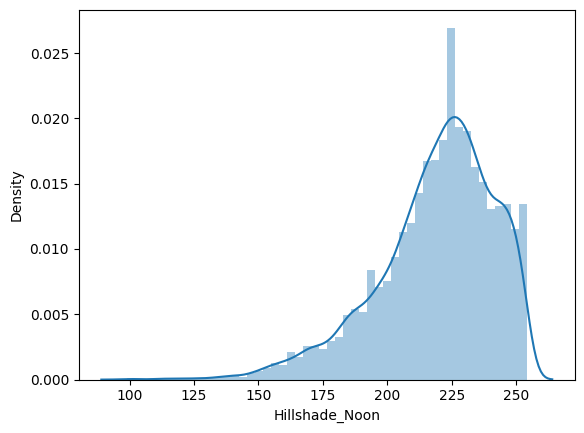




As we can see from descriptive statistics, the variable ‘Hillshade\_9am’, has a skew value of   -1.093 and kurtosis value of +1.21. Therefore, we can say that it is negatively skewed and leptokurtic in nature.

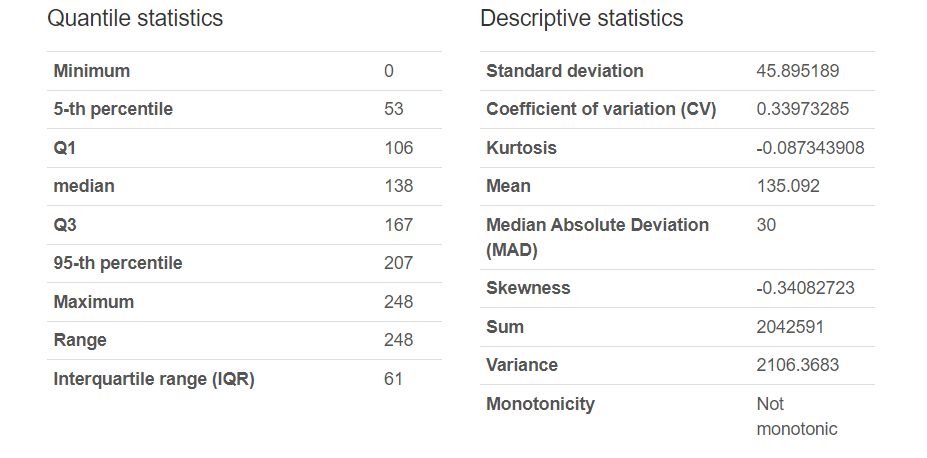
1. **HILLSHADE NOON:**

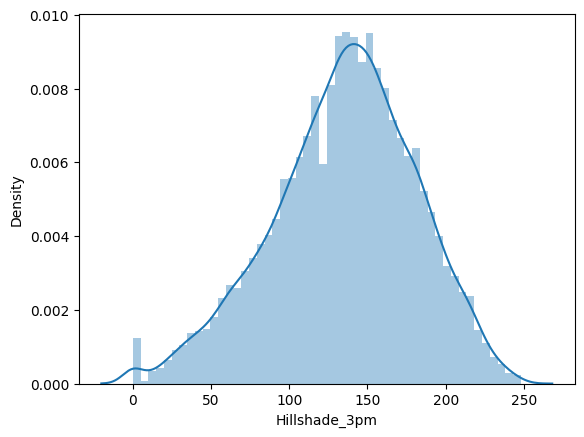




As we can see from descriptive statistics, the variable ‘Hillshade\_Noon',has a skew value of  -1.095 and kurtosis value of +1.15. Therefore, we can say that it is negatively skewed and leptokurtic in nature.

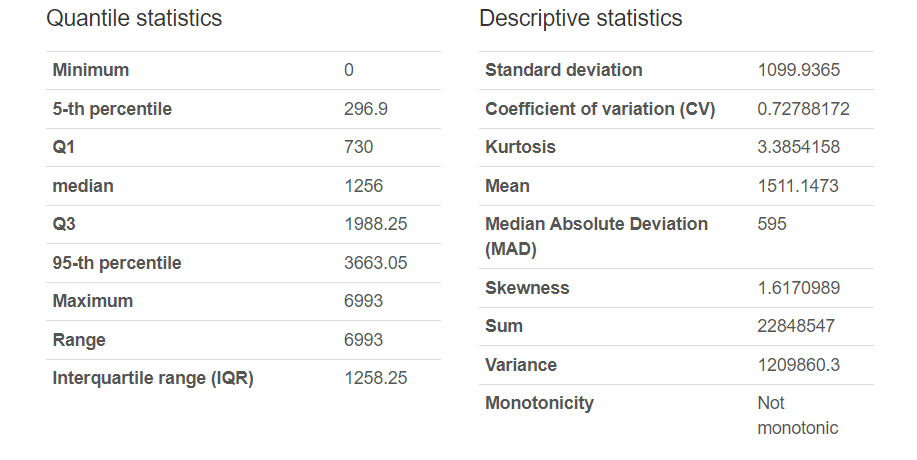
1. **HILLSHADE 3PM:**

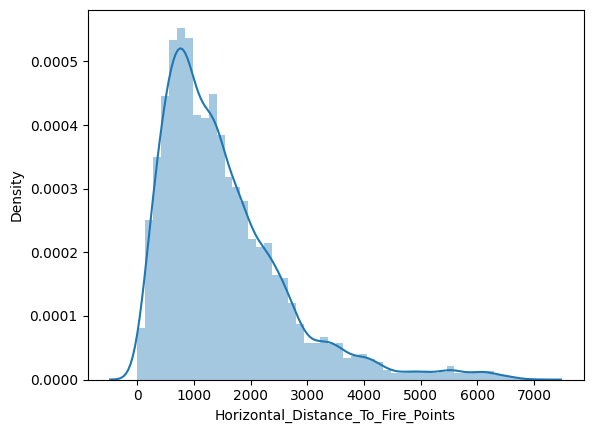




As we can see from descriptive statistics, the variable ‘Hillshade\_3pm', 'Hillshade\_Noon',has a skew value of -0.34 and kurtosis value of -0.08. Therefore, we can say that it is negatively skewed and mesokurtic in nature.

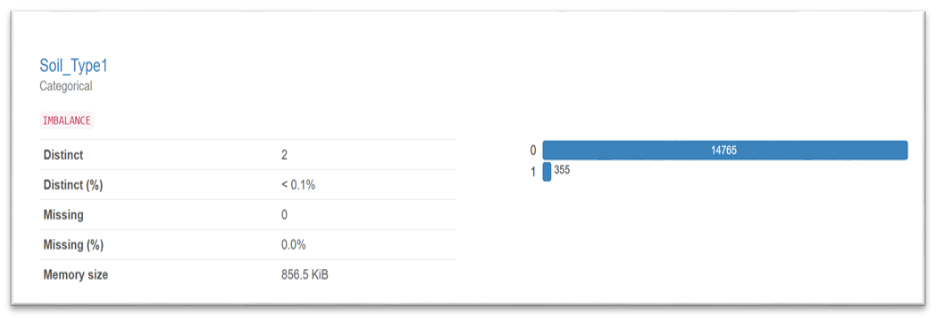
1. **HORIZONTAL DISTANCE TO FIRE POINTS:**





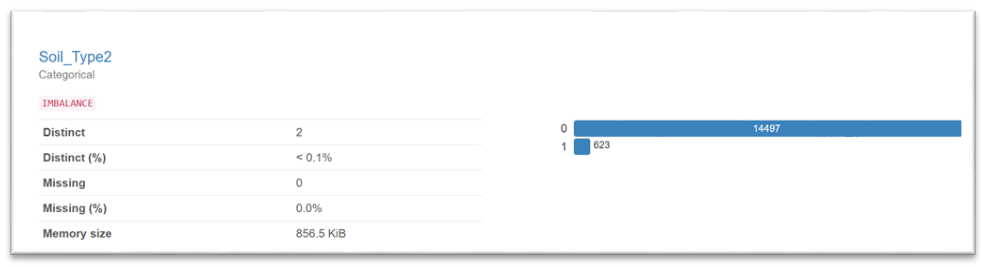
As we can see from descriptive statistics, the variable ‘Hillshade\_Noon',has a skew value of  +1.617 and kurtosis value of +3.38. Therefore, we can say that it is negatively skewed and leptokurtic in nature.

1. **Soil Type1:**



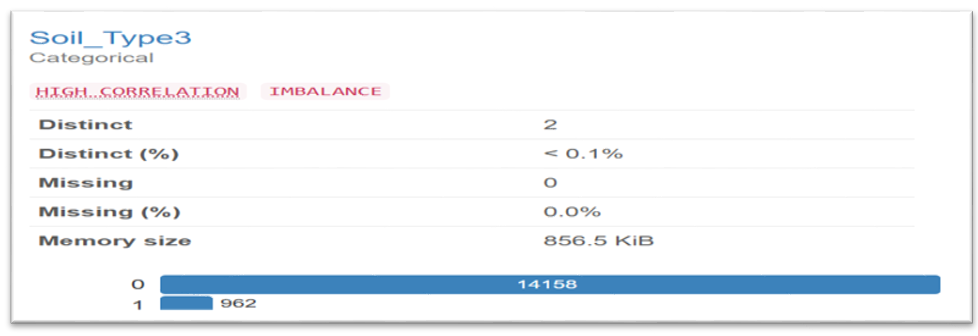
Here the variable ‘Soil\_Type1’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 97.7% of data is 0 along with 2.3% as 1 which means 2.3% of data contains from ‘Soil\_Type1’ and remaining from rest of the soils.

1. **Soil Type 2:**



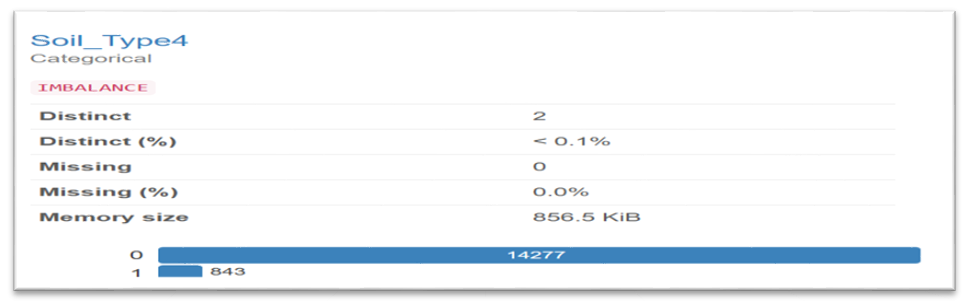
Here the variable ‘Soil\_Type2’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.9% of data is 0 along with 4.1% as 1 which means 4.1% of data contains from ‘Soil\_Type2’ and remaining from rest of the soils.

1. **Soil Type 3:**



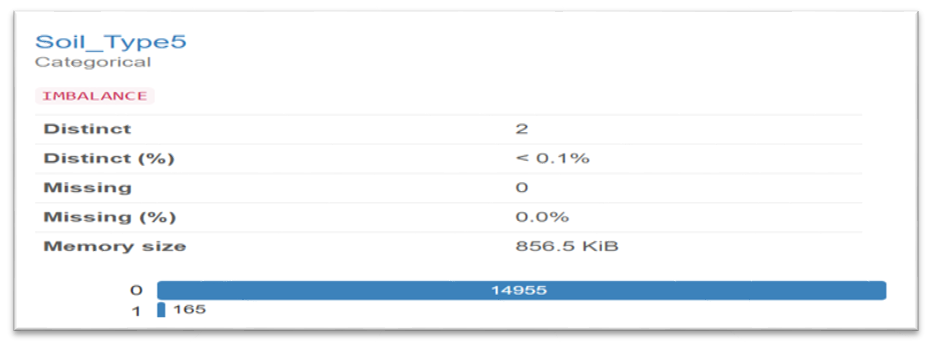
Here the variable ‘Soil\_Type3’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 93.6% of data is 0 along with 6.4% as 1 which means 6.4% of data contains from ‘Soil\_Type3’ and remaining from rest of the soil.

1. **Soil Type 4:**



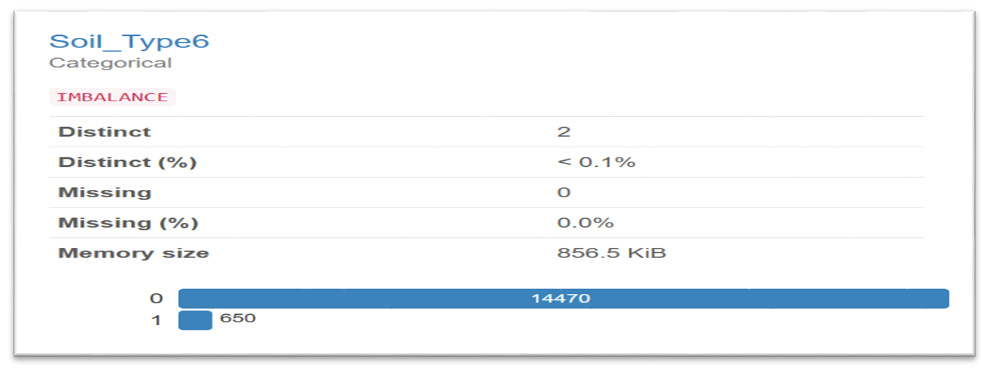
Here the variable ‘Soil\_Type4’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 94.4% of data is 0 along with 5.6% as 1 which means 5.6% of data contains from ‘Soil\_Type4’ and remaining from rest of the soils

1. **Soil Type 5:**



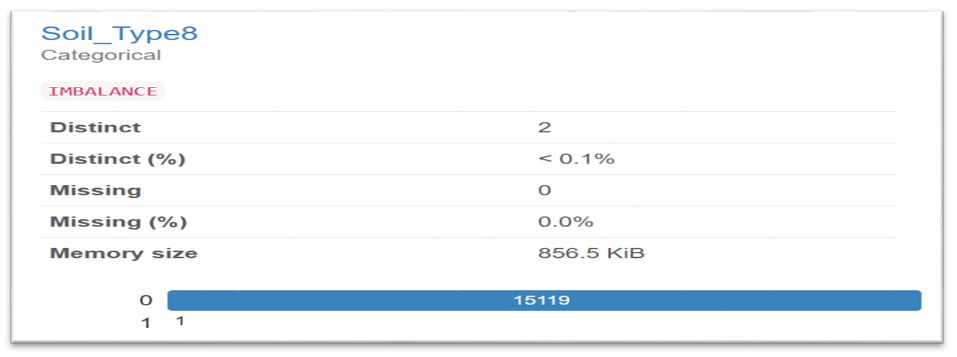
Here the variable ‘Soil\_Type5’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 98.9% of data is 0 along with 1.1% as 1 which means 1.1% of data contains from ‘Soil\_Type5’ and remaining from rest of the soils.

1. **Soil Type 6:**



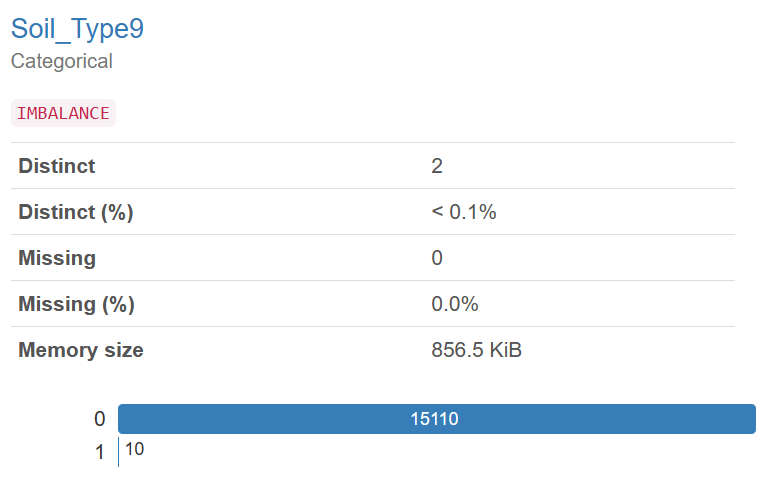
Here the variable ‘Soil\_Type6’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.7% of data is 0 along with 4.3% as 1 which means 4.3% of data contains from ‘Soil\_Type6’ and remaining from rest of the soils.

1. **Soil Type 8:**



Here the variable ‘Soil\_Type8’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.9% of data is 0 along with 0.1% as 1 which means 0.1% of data contains from ‘Soil\_Type8’ and remaining from rest of the soils.

1. **Soil Type 9:**



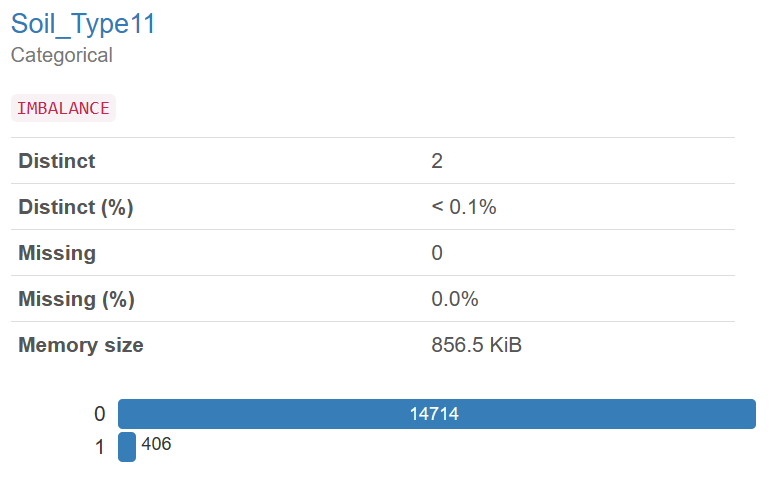
Here the variable ‘Soil\_Type9’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.9% of data is 0 along with 0.1% as 1 which means 0.1% of data contains from ‘Soil\_Type9’ and remaining from rest of the soils.

1. **Soil Type10:**



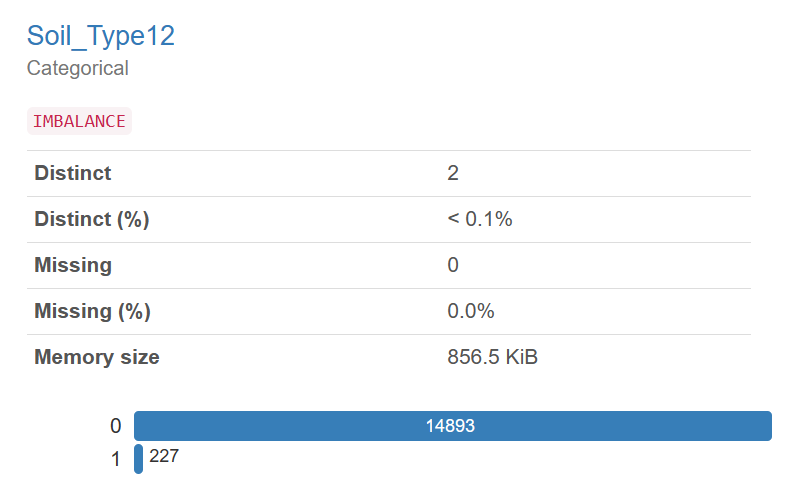
Here the variable ‘Soil\_Type10’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 85.8% of data is 0 along with 14.2% as 1 which means 14.2% of data contains from ‘Soil\_Type10’ and remaining from rest of the soils.

1. **Soil Type 11:**

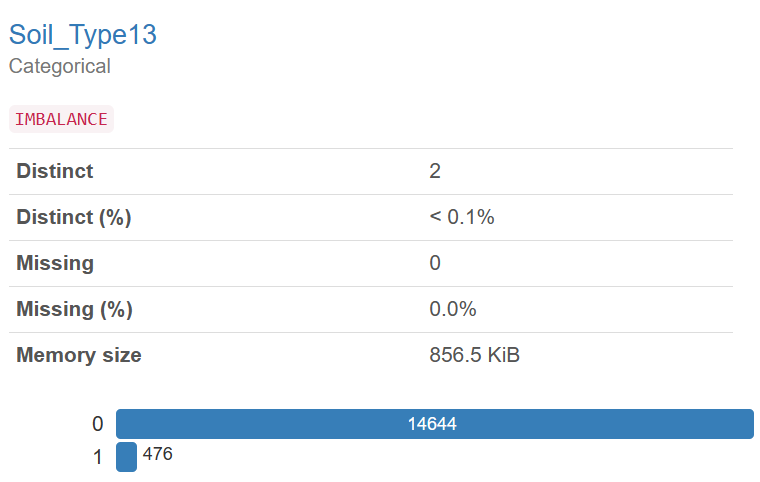


Here the variable ‘Soil\_Type11’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 97.3% of data is 0 along with 2.7% as 1 which means 2.7% of data contains from ‘Soil\_Type11’ and remaining from rest of the soils.

1. **Soil Type12:**

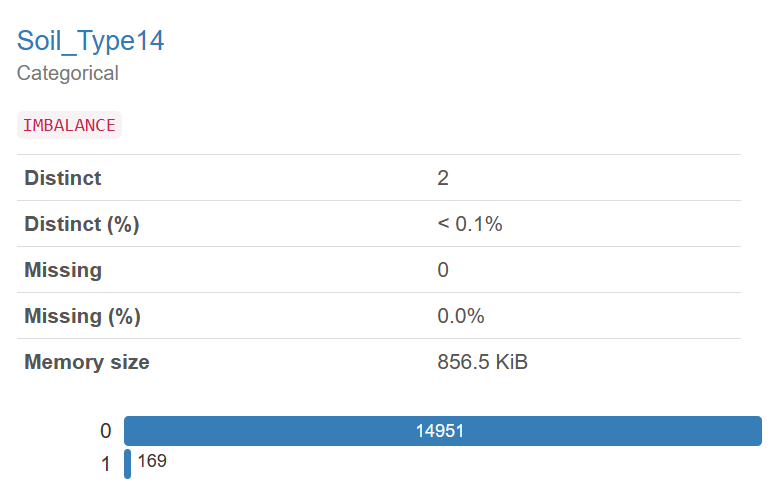


Here the variable ‘Soil\_Type12’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 98.5% of data is 0 along with 1.5% as 1 which means 1.5% of data contains from ‘Soil\_Type12’ and remaining from rest of the soils.

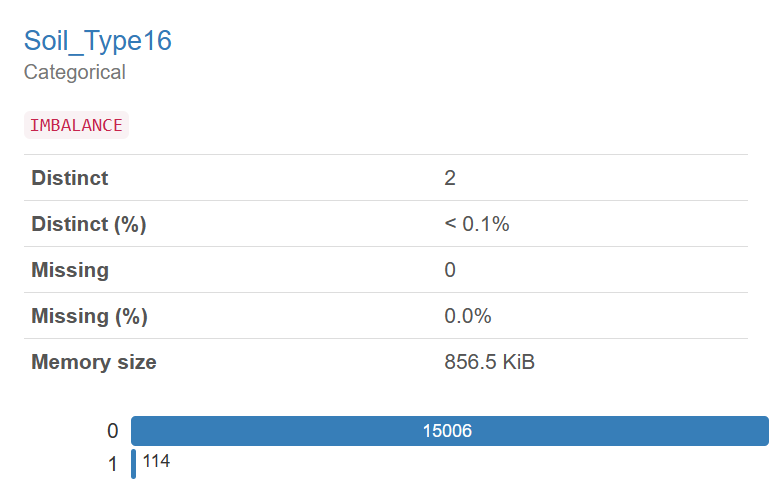
1. **Soil Type13:**

Here the variable ‘Soil\_Type13’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 96.9% of data is 0 along with 3.1% as 1 which means 3.1% of data contains from ‘Soil\_Type13’ and remaining from rest of the soils.

1. **Soil Type14:**

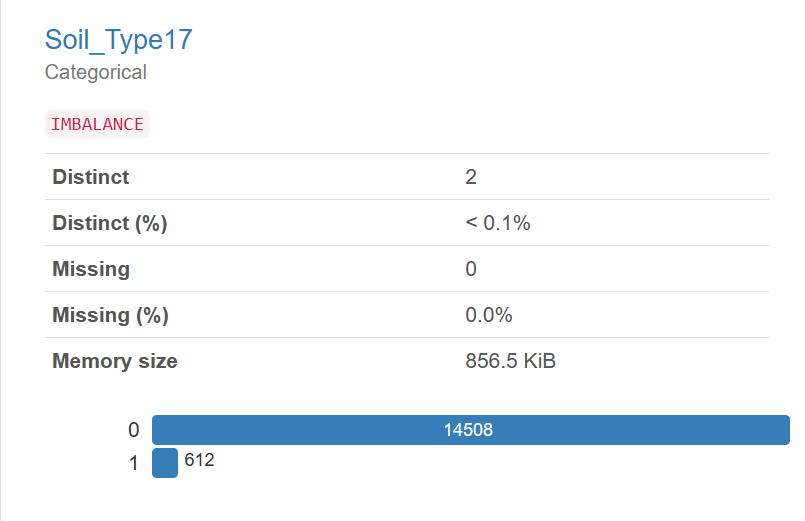


Here the variable ‘Soil\_Type14’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 98.9% of data is 0 along with 1.1% as 1 which means 1.1% of data contains from ‘Soil\_Type14’ and remaining from rest of the soils.

1. **Soil Type16:**

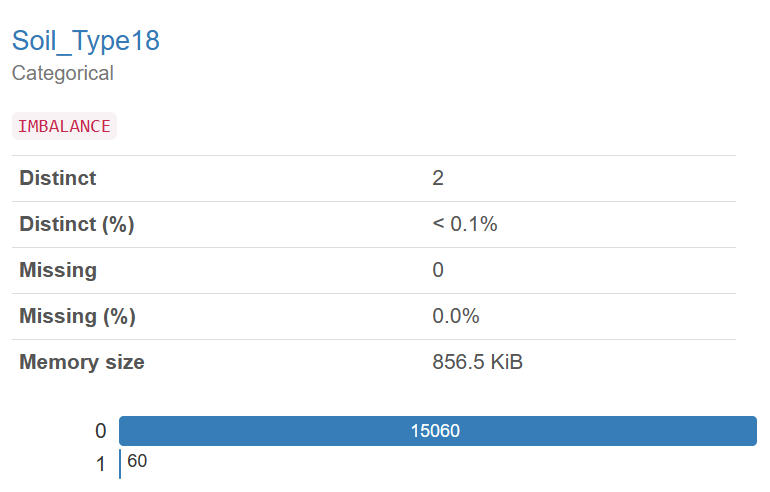
Here the variable ‘Soil\_Type16’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.2% of data is 0 along with 0.8% as 1 which means 0.8% of data contains from ‘Soil\_Type16’ and remaining from rest of the soils.

1. **Soil Type17:**

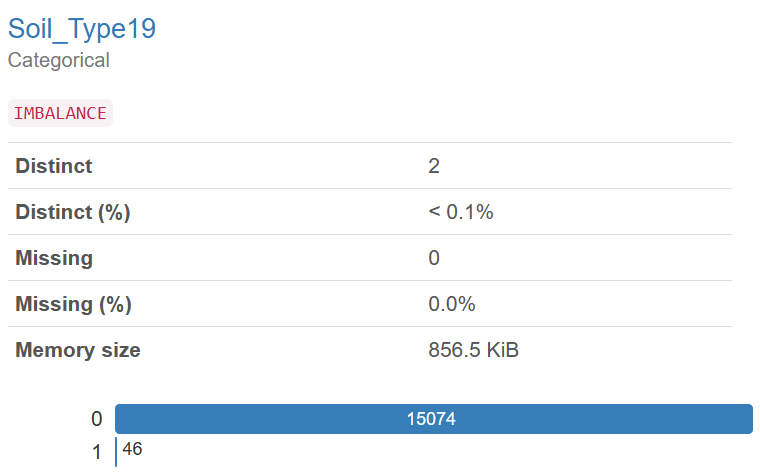


Here the variable ‘Soil\_Type17’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 96.0% of data is 0 along with 4% as 1 which means 4% of data contains from ‘Soil\_Type17’ and remaining from rest of the soils.

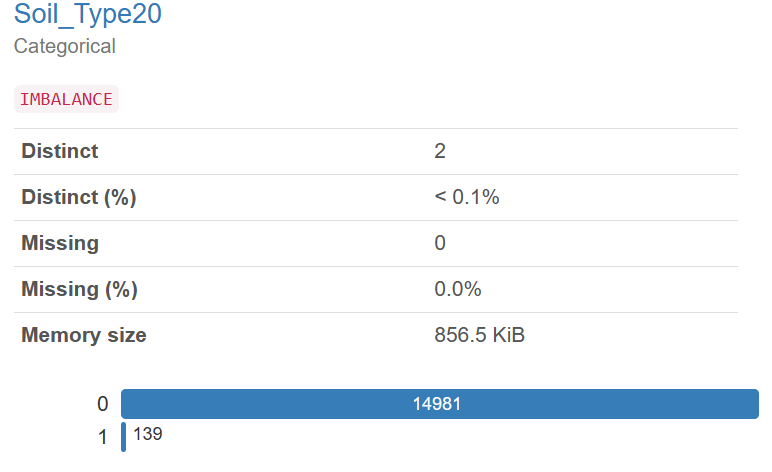
1. **Soil Type18:**



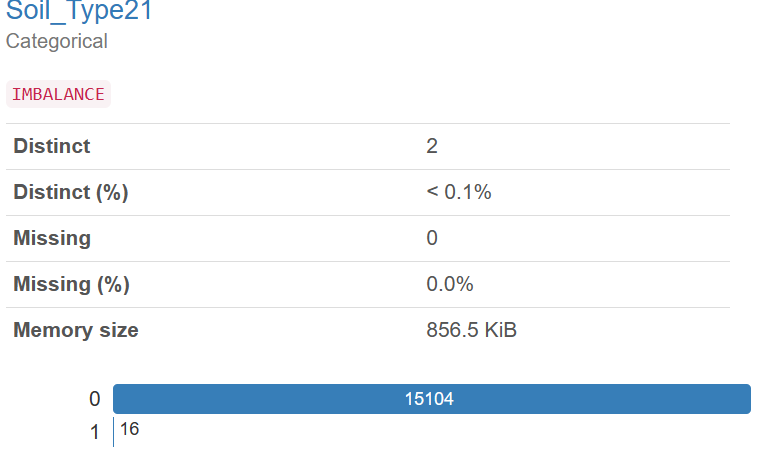
Here the variable ‘Soil\_type18’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.6% of data is 0 along with 0.4% as 1 which means 0.4% of data contains from ‘Soil\_Type18’ and remaining from rest of the soils.

1. **Soil Type19:**

Here the variable ‘Soil\_type19’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.7% of data is 0 along with 0.3% as 1 which means 0.32% of data contains from ‘Soil\_Type19’ and remaining from rest of the soils.

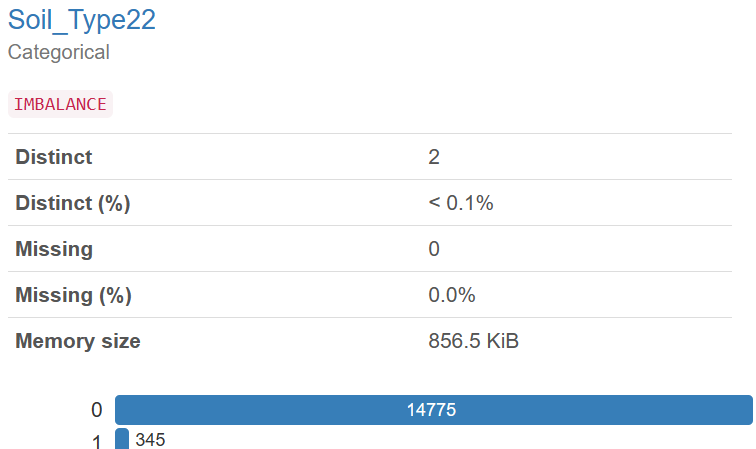
1. **Soil Type20:**

Here the variable ‘Soil\_Type20’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.1% of data is 0 along with 0.9% as 1 which means 0.9% of data contains from ‘Soil\_Type20’ and remaining from rest of the soils.

1. **Soil Type21:**

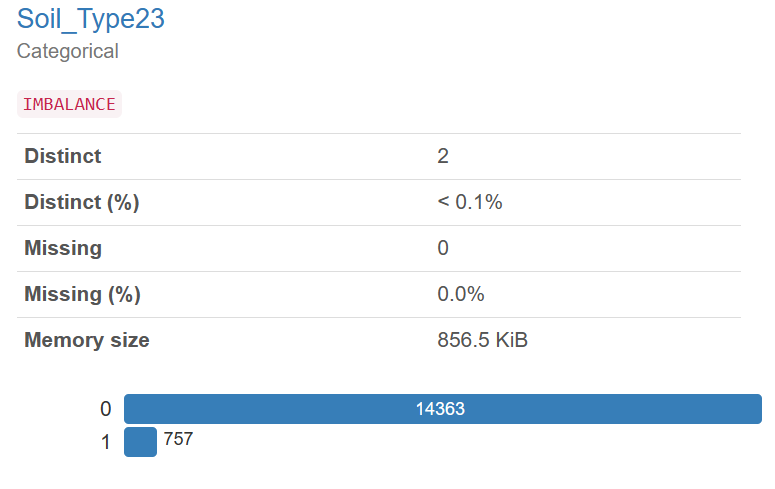
Here the variable ‘Soil\_Type21’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.9% of data is 0 along with 0.1% as 1 which means 0.1% of data contains from ‘Soil\_Type21’ and remaining from rest of the soils.

1. **Soil Type22:**



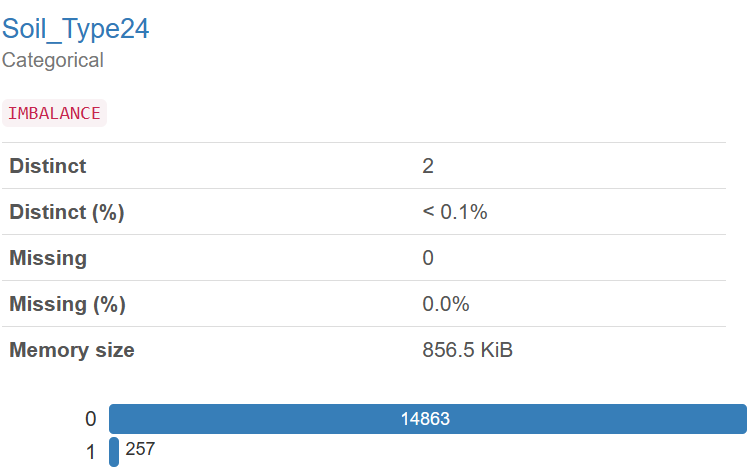
Here the variable ‘Soil\_Type22’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 97.7% of data is 0 along with 2.3% as 1 which means 2.3% of data contains from ‘Soil\_Type22’ and remaining from rest of the soils.

1. **Soil Type23:**

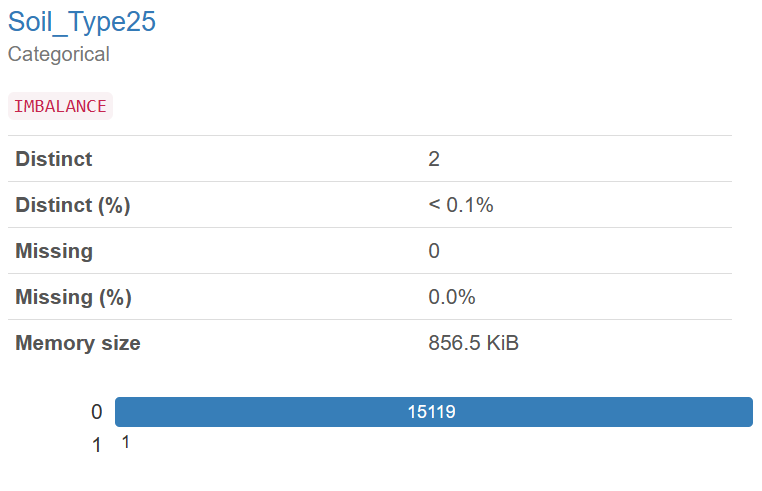


Here the variable ‘Soil\_Type23’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.0% of data is 0 along with 5% as 1 which means 5% of data contains from ‘Soil\_Type23’ and remaining from rest of the soils.

1. **Soil Type24:**

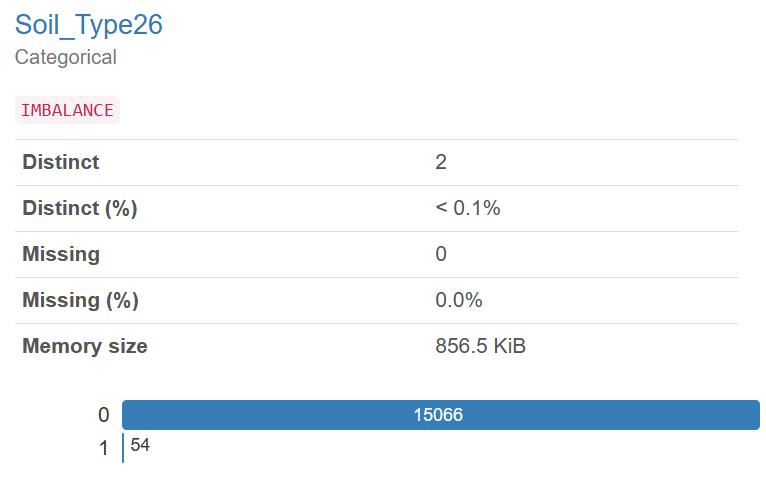


Here the variable ‘Soil\_Type24’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 98.3% of data is 0 along with 1.7% as 1 which means 1.7% of data contains from ‘Soil\_Type24’ and remaining from rest of the soils.

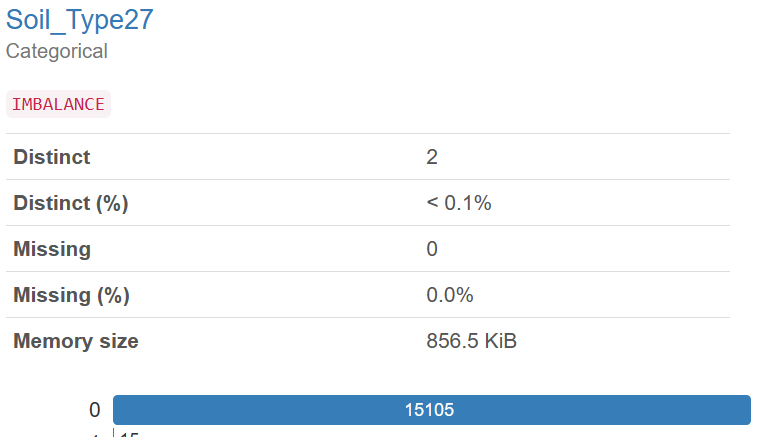
1. **Soil Type25:**

Here the variable ‘Soil\_Type25’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that greater than 99.9% of data is 0 along with less than 0.1% as 1 which means less than 0.1% of data contains from ‘Soil\_Type25’ and remaining from rest of the soils.

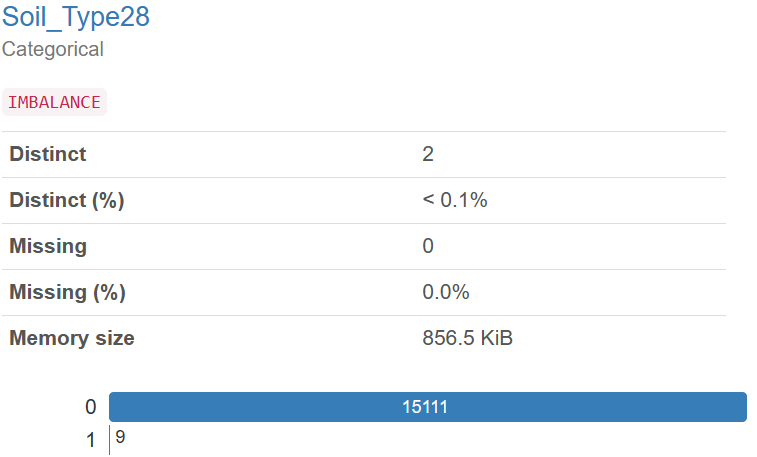
1. **Soil Type26:**



Here the variable ‘Soil\_Type26’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.6% of data is 0 along with 0.4% as 1 which means 0.4% of data contains from ‘Soil\_Type26’ and remaining from rest of the soils.

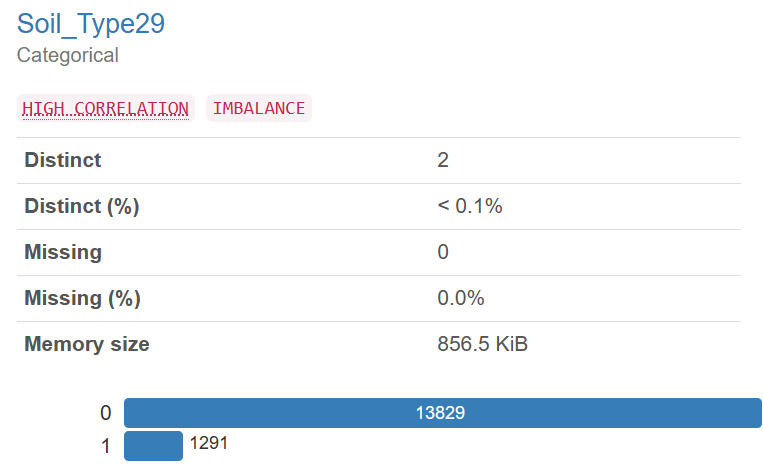
1. **Soil Type 27:**

Here the variable ‘Soil\_Type27’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.9% of data is 0 along with 0.1% as 1 which means 0.1% of data contains from ‘Soil\_Type27’ and remaining from rest of the soils.

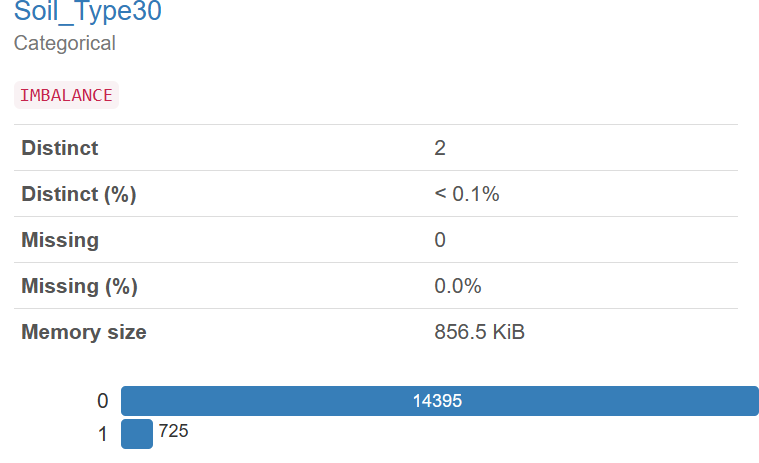
1. **Soil Type28:**

Here the variable ‘Soil\_Type28’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.9% of data is 0 along with 0.1% as 1 which means 0.1% of data contains from ‘Soil\_Type28’ and remaining from rest of the soils.

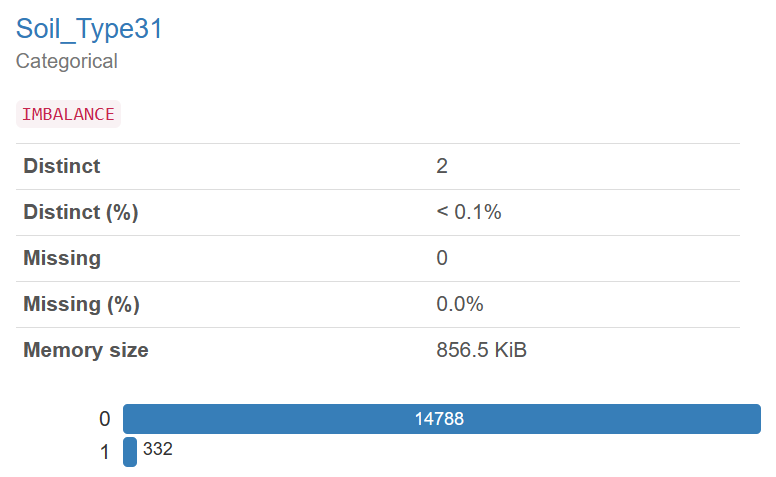
1. **Soil Type 29:**



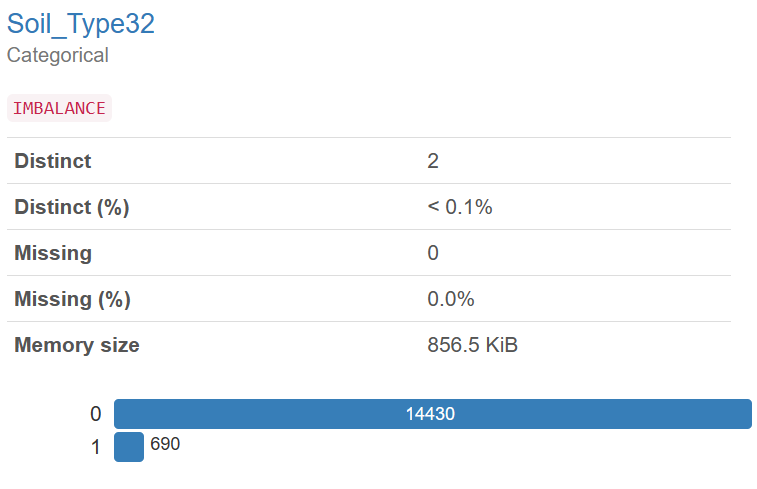
Here the variable ‘Soil\_Type29’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 91.5% of data is 0 along with 8.5% as 1 which means 8.5% of data contains from ‘Soil\_Type29’ and remaining from rest of the soils.

1. **Soil Type 30:**

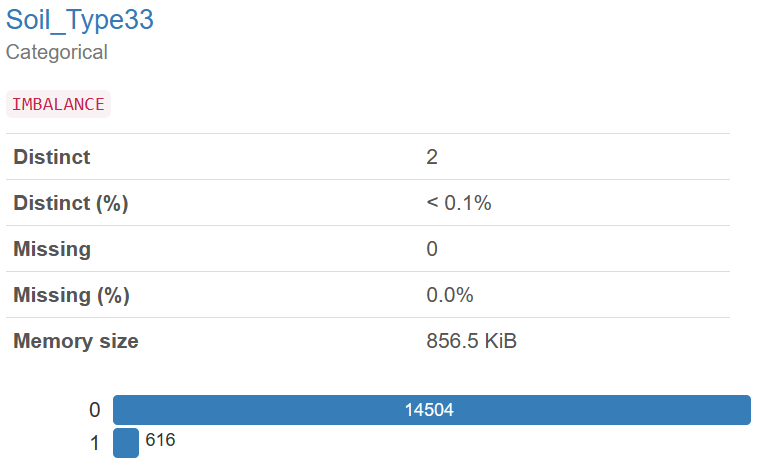
Here the variable ‘Soil\_Type30’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.2% of data is 0 along with 4.8% as 1 which means 4.8% of data contains from ‘Soil\_Type30’ and remaining from rest of the soils.

1. **Soil Type 31:**

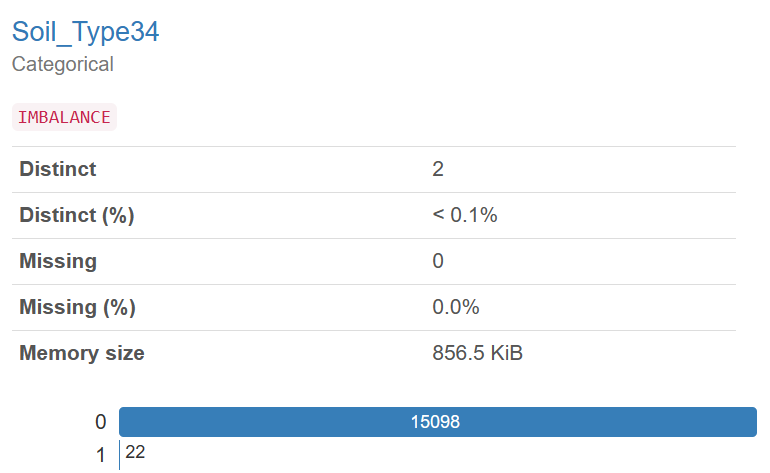
Here the variable ‘Soil\_Type31’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 97.8% of data is 0 along with 2.2% as 1 which means 2.2% of data contains from ‘Soil\_Type31’ and remaining from rest of the soils.

1. **Soil Type32:**

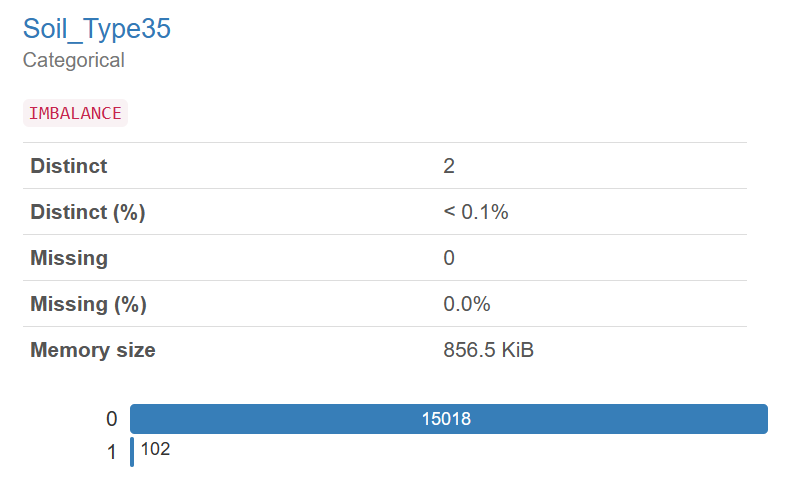
Here the variable ‘Soil\_Type32’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.4% of data is 0 along with 4.6% as 1 which means 4.6% of data contains from ‘Soil\_Type32’ and remaining from rest of the soils.

1. **Soil Type 33:**

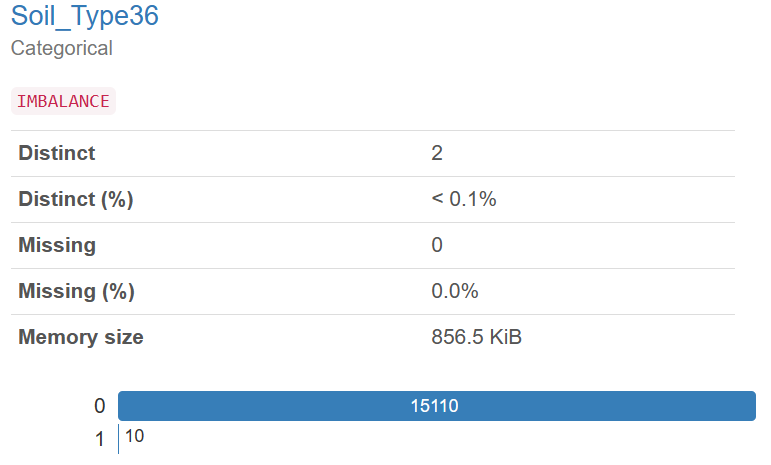
Here the variable ‘Soil\_Type33’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.9% of data is 0 along with 4.1% as 1 which means 4.1% of data contains from ‘Soil\_Type33’ and remaining from rest of the soils.

1. **Soil Type 34:**

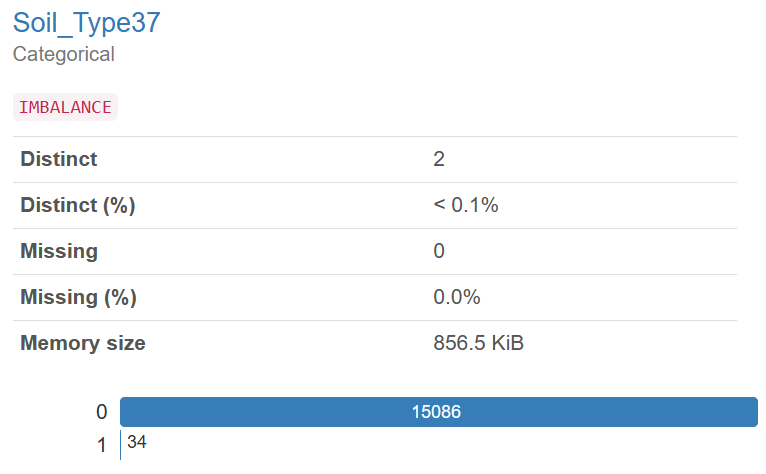
Here the variable ‘Soil\_Type34’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.9% of data is 0 along with 0.1% as 1 which means 0.1% of data contains from ‘Soil\_Type34’ and remaining from rest of the soils.

1. **Soil Type35:**  

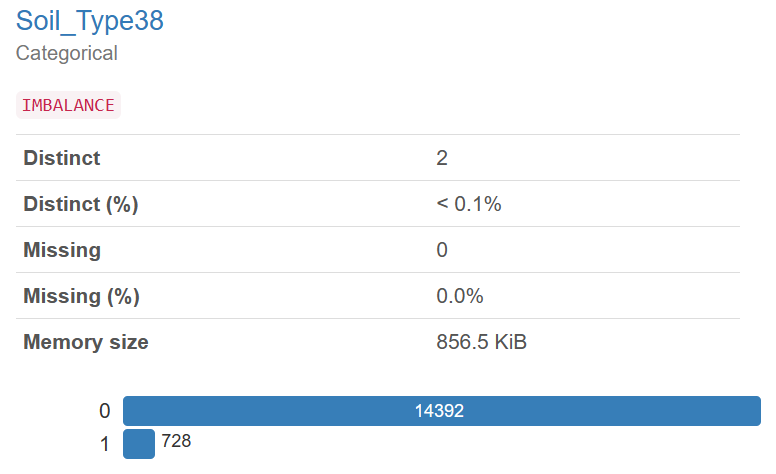
Here the variable ‘Soil\_Type35’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.3% of data is 0 along with 0.7% as 1 which means 0.7% of data contains from ‘Soil\_Type35’ and remaining from rest of the soils.

1. **Soil Type36:**

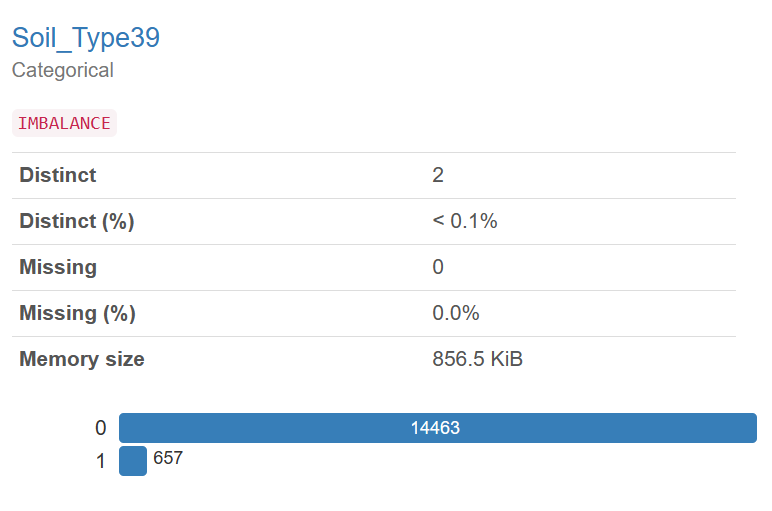
Here the variable ‘Soil\_Type36’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.9% of data is 0 along with 0.1% as 1 which means 0.1% of data contains from ‘Soil\_Tpe36’ and remaining from rest of the soils.

1. **Soil Type37:**

Here the variable ‘Soil\_Type37 has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 99.8% of data is 0 along with 0.2% as 1 which means 0.2% of data contains from ‘Soil\_Type37 and remaining from rest of the soils.

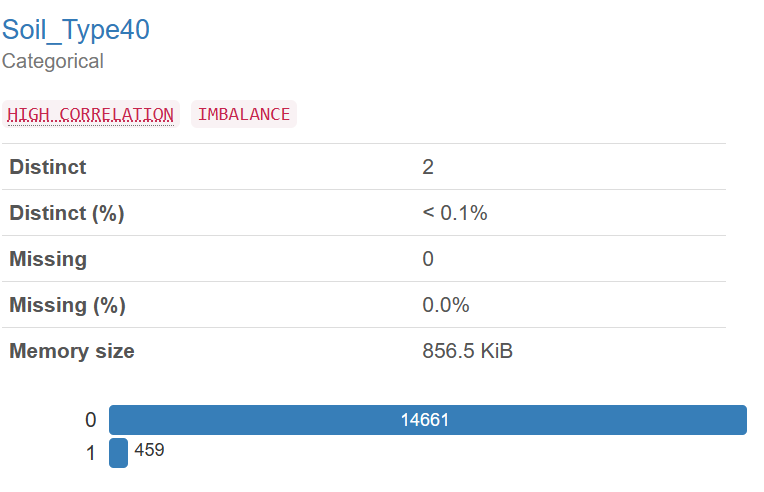
1. **Soil Type38:**

Here the variable ‘Soil\_Type38’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.2% of data is 0 along with 4.8% as 1 which means 4.8% of data contains from ‘Soil\_Type38’ and remaining from rest of the soils.

1. **Soil Type39:**

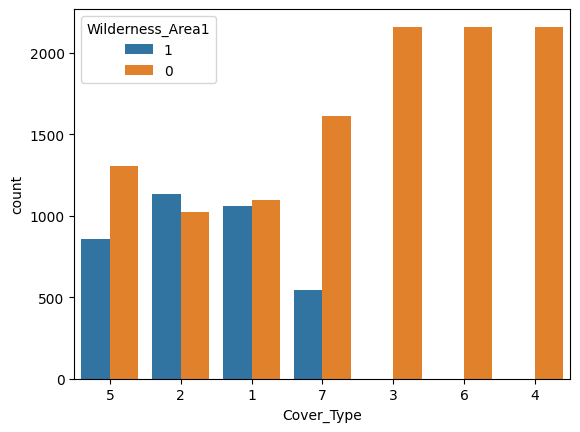
Here the variable ‘Soil\_Type39’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 95.7% of data is 0 along with 4.3% as 1 which means 4.3% of data contains from ‘Soil\_Type39’and remaining from rest of the soils.

1. **Soil Type 40:**



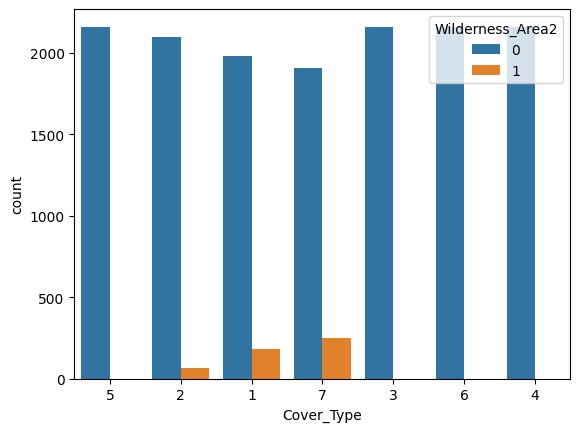
Here the variable ‘Soil\_Type40’ has two categories yes and no in its record which is encoded to 1 and 0 respectively. we can see from the chart that 97.0% of data is 0 along with 3.0% as 1 which means 3.0% of data contains from ‘Soil\_Type40’ and remaining from rest of the soils.

1. **Wilderness Area 1:**



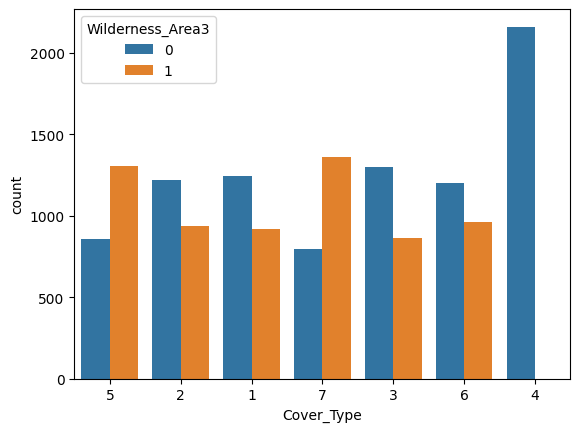
As we can see the plot from wilderness area1 has most forest cover types are Spruce/Fir, Lodgepole Pine, Krummholz and Aspen and remaining forests present in other areas. Here we conclude that moderate amount of data available from Wilderness area 1.

1. **Wilderness Area 2:**



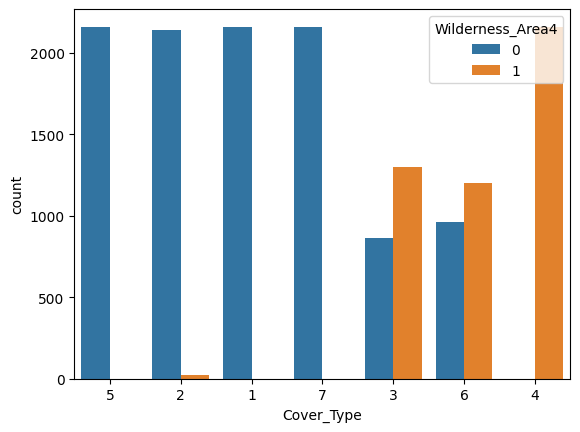
As we can see the plot from wilderness area 2 has most forest cover types are Spruce/Fir, Lodgepole Pine, Krummholz and remaining forests present other areas. Here we can conclude that data contains very less types of forest cover type available from the wilderness Area 2.

1. **Wilderness Area 3:**



As we can see the plot from wilderness area 3 most cover types are Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Aspen, Douglas-fir, Krummholz and there is no data available from Cottonwood/Willow. Here we conclude that most of the data available from forest cover type contains wilderness Area 3.

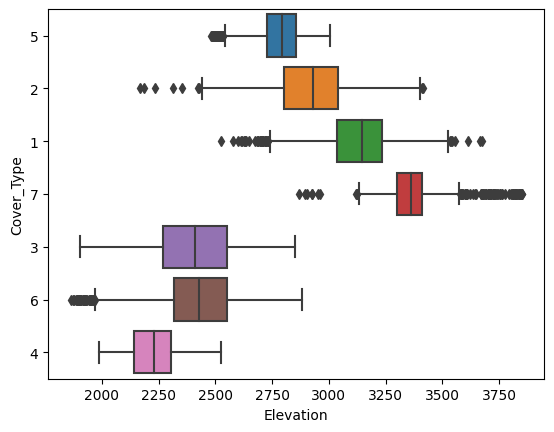
1. **Wilderness Area 4:**



As we can see the plot from wilderness area 4 has most forest cover type are Lodgepole/Fir, Ponderosa Pine, Douglas-fir, Cottonwood/willow and remaining forests present other areas. Here we conclude that cottonwood/willow forest cover type is very high in Wilderness area 4.

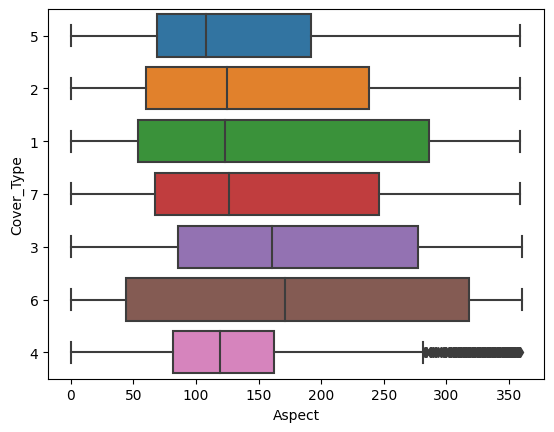
* 1. **BIVARIATE ANALYSIS**

1. **Bivariate Analysis of Elevation and Cover Type:**



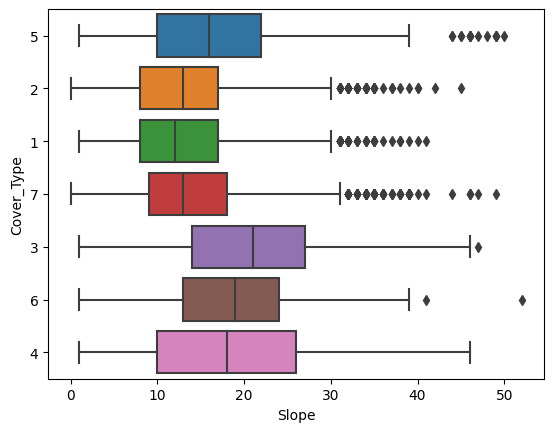
The box plot indicates that for Cover Type 7, Cover Type 2, Cover Type 5, and Cover Type 1, are having high median values when compared to Cover Type 3, Cover Type 6, and Cover Type 4.

1. **Bivariate Analysis of Aspect and Cover Type:**



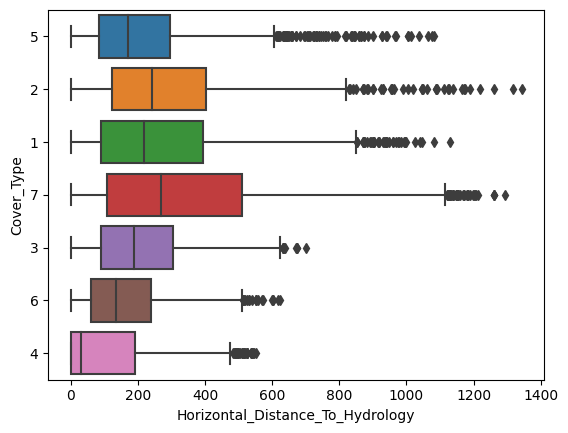
The box plot indicates that for Cover Type 1 and Cover Type 6, Aspect values are widely dispersed, as evidenced by the box size. Additionally, some Cover Type median values overlap, suggesting that Aspect might not be a strong variable for accurate predictions.

1. **Bivariate Analysis of Slope and Cover Type:**



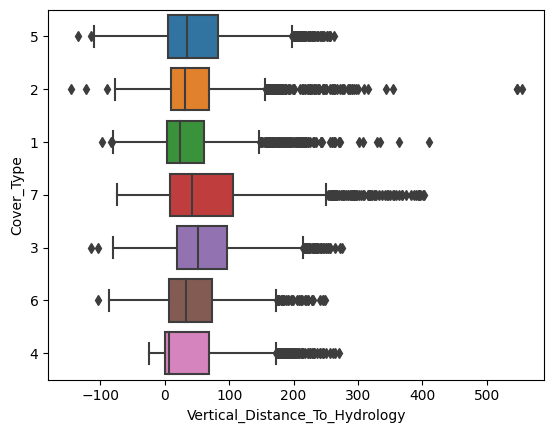
The boxplot reveals that median values are the same for certain slope values across different cover types. Further statistical analysis will be carried out to determine the significance of this observation.

1. **Horizontal Distance to Hydrology and Cover Type:**



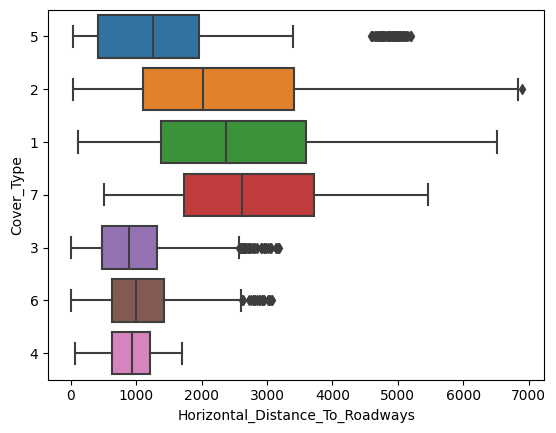
The box plot illustrates the distribution of values for each cover type in horizontal distance to hydrology. The median values show slight differences among cover types, and there are some extreme values presents.

1. **Vertical Distance to Hydrology and Cover Type:**



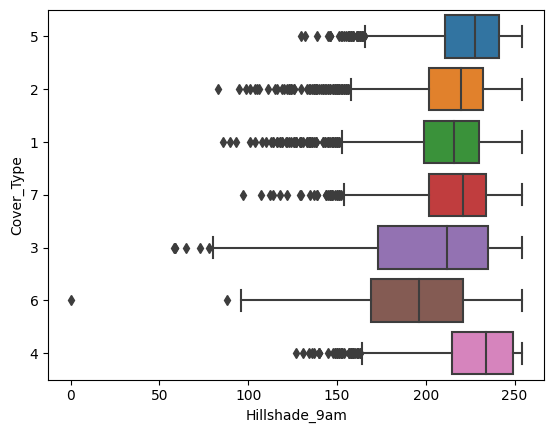
The box plot indicates nearly equal median values for vertical distance to hydrology across cover types, suggesting this may not be a significant feature for model building.

1. **Horizontal Distance to Roadways and Cover Type:**



The boxplot clearly shows differences in how far the forest areas are from roadways. Cover Type 1 and Cover Type 2, especially Type 2, tend to have higher distances. The middle line (median) also varies, indicating distinctions between cover types.

1. **Hillshade 9AM and Cover Type:**



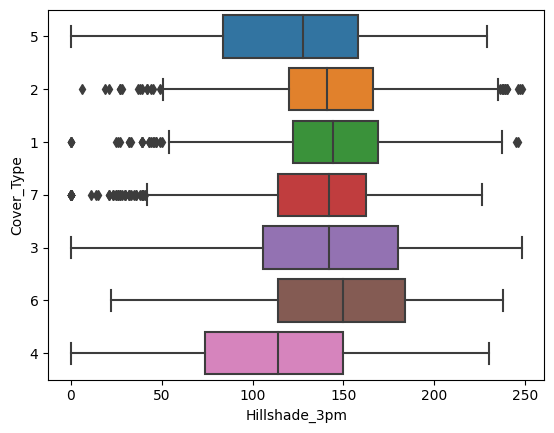
The boxplot indicates how the data is spread for the Hillshade at 9 am. The median values are relatively close, suggesting some similarity in the distribution across different categories.

1. **Hillshade Noon and Cover Type:**



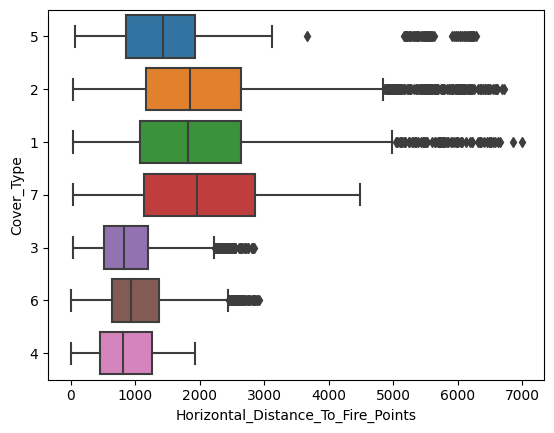
The boxplot indicates that median values are close for Hillshade at Noon, implying similarity. Further statistical analysis is needed to determine its significance as a feature in relation to cover types.

1. **Hillshade 3PM and Cover Type:**



The boxplot illustrates that median values are relatively close for Hillshade at 3 PM. A detailed statistical analysis will be conducted to assess its significance in relation to different cover types.

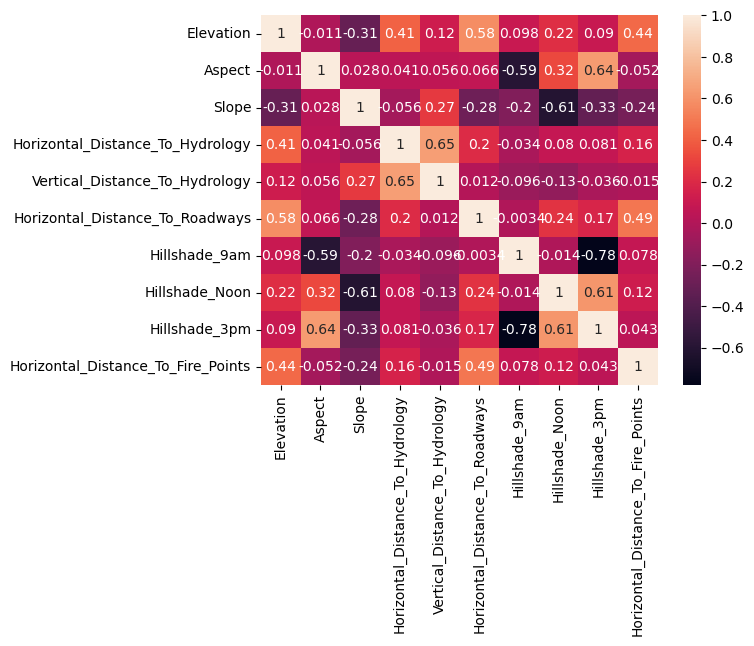
1. **Horizontal Distance to Fire Points and Cover Type:**



The boxplot indicates noticeable differences in median values among various cover types. However, cover types 1 and 2 show relatively similar median values, suggesting a potential overlap in their distributions.

* 1. **Numeric Feature correlation:**



To check the correlation between the variables, Spearman’s correlation is used. The Spearman's rank correlation coefficient (ρ) is a measure of monotonic correlation between two variables, and is therefore better in catching nonlinear monotonic correlations than Pearson's r. Its value lies between -1 and +1, -1 indicating total negative monotonic correlation, 0 indicating no monotonic correlation and 1 indicating total positive monotonic correlation.

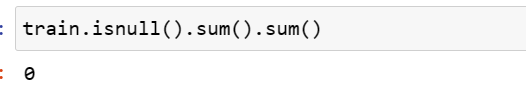
1. **DATA PREPROCESSING**

* 1. **DATA CLEANING:**

**Checking Missing Values:**

To understand trees data frame, let us look at the data types and descriptive

statistics. With pandas **isnull().sum().sum()**method, we can list the non-null values and data types:



There are no missing attribute values in the dataset.

* 1. **Scaling the Features for Model Optimization:**

Here we are using robust scaling to scale the independent variable.

 If there are too many outliers in the data, they will influence the mean and the max value or the min value. Thus, even if we scale this data using the above methods, we cannot guarantee a balanced data with a normal distribution.

The Robust Scaler, as the name suggests is not sensitive to outliers. This scaler-

1. removes the median from the data.
2. scales the data by the Interquartile Range (IQR)

It is the difference between the first and third quartile of the variable. The interquartile range can be defined as-

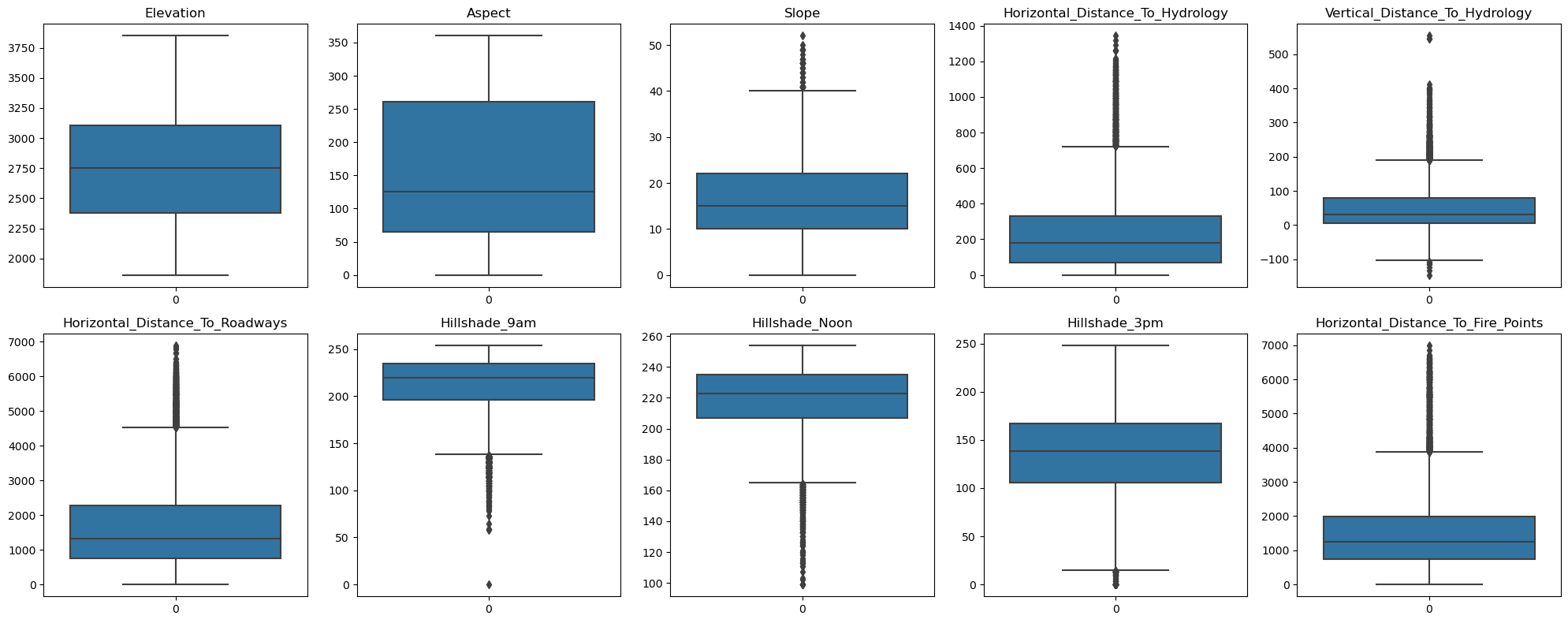
IQR = Q3 – Q1

Thus, the formula would be:

**x\_scaled = (x – Q1)/(Q3 – Q1)**

* 1. **Outlier Treatment Approach:**

**Exploring Numerical Feature Distributions and Outliers**



The boxplots reveal the presence of outliers in specific columns, namely 'Horizontal\_Distance\_To\_Hydrology’,

'Vertical\_Distance\_To\_Hydrology,'

'Horizontal\_Distance\_To\_Roadways,'

'Hillshade\_9am,'

'Hillshade\_Noon,'

'Horizontal\_Distance\_To\_Fire\_Points.'

On observation Among these ‘Horizontal\_Distance\_To\_Roadways’, ‘Horizontal\_Distance\_To\_Fire\_Points’ exhibit a higher frequency of outliers compared to the other variables with a lower frequency of outliers.

Box plots were generated to assess the relationship between independent features and the target variable, revealing the presence of outliers. The boxplots illustrate the existence of outliers in certain columns, including 'Horizontal\_Distance\_To\_Hydrology,' 'Vertical\_Distance\_To\_Hydrology,' 'Horizontal\_Distance\_To\_Roadways,' 'Hillshade 9am,' 'Hillshade\_Noon,' and 'Horizontal Distance\_To\_Fire Points.'

Here we can say that mean and median almost same values are same for the columns - Slope, hillshade\_3pm, hillshade\_noon, hillshade\_9am.we cannot doing any outlier treatment for this columns.

The mean and median values of the columns 'Horizontal\_Distance\_To\_Hydrology', 'Vertical\_Distance\_To\_Hydrology', 'Horizontal\_Distance\_To\_Roadways', and 'Horizontal\_Distance\_To\_Fire\_Points' are different, indicating a lack of symmetry in their distributions. To address this issue specifically for these columns, the decision was made to utilize the KNN (K-Nearest Neighbours) imputer.

The KNN imputer is a robust technique employed to handle missing or outlier values by leveraging distances to the k nearest neighbours. By using the characteristics and values of neighboring data points, this method predicts and replaces missing or outlier values more accurately. It helps ensure a more representative depiction of the dataset's distribution, effectively maintaining its integrity while mitigating the impact of outliers on subsequent analysis.

* 1. **Addressing Multicollinearity in the Feature:**

To mitigate multicollinearity within the feature set, we employ the Variance Inflation Factor (VIF) method. This technique helps us identify and eliminate highly correlated columns to ensure that the predictive variables are independent and do not introduce redundancy into the model. If two columns exhibit similar significance and demonstrate high correlation, we opt to retain only one of them. Following the application of the VIF method, we have identified and removed specific columns, including 'Wilderness\_Area1,' 'Soil\_Type10,' 'Hillshade\_Noon,' 'Hillshade\_9am,' 'Hillshade\_3pm,' 'Wilderness\_Area4,' and 'Slope,' which displayed elevated correlation levels. This process enhances the robustness and accuracy of the model by ensuring that the selected features contribute unique information without introducing collinearity-related issues.

1. **MODEL BUILDING**

For model building we are Splitting the dataset into an 80-20 ratio for training and testing.

**Base Model:**

In the base model, we employed Logistic Regression, Decision Tree, and Random Forest to predict the target variable based on cover types. The target variable comprises 7 classes, where each class corresponds to a specific forest cover type:

1. Class 1: Spruce/Fir

2. Class 2: Lodgepole Pine

3. Class 3: Ponderosa Pine

4. Class 4: Cottonwood/Willow

5. Class 5: Aspen

6. Class 6: Douglas-fir

7. Class 7: Krummholz

**Decision Tree:**

A decision tree is a flow chart structure which consist of internal node which represents a test in attribute and that comes with each branch out i.e. the result of the test and each leaf represents a category label (a decision taken after testing all attributes in the path from the beginning to the leaf). Each path from the source to a leaf can also be interpreted as a sorting rule.

When establishing a supervised classification model, the frequency distribution of attribute values is a potentially significant component in deciding the proportional importance of each attribute at various levels in the model construction procedure.

In data modeling, we can use frequency distributions to compute entropy. We calculate the entropy of multiplying the proportion of cases with each category label by the log of that proportion, and then getting the negative essence of those conditions.

Entropy (S) =−p1log2 (p1) − p2log2 (p2)

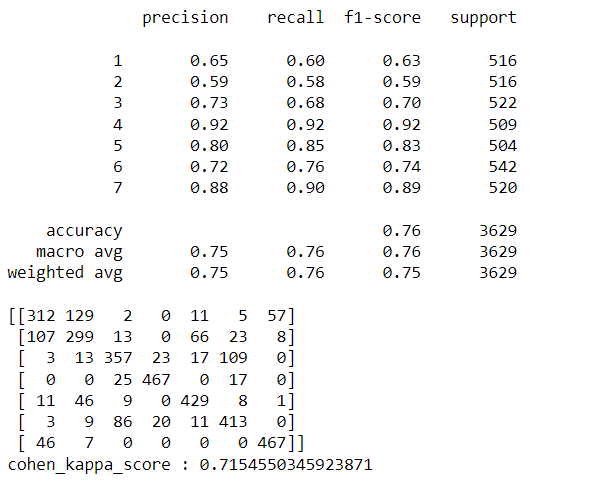
Where pi is proportion (relative frequency) of class i within the set S.

A decision tree is constructed algorithm that selects the best attribute, splits the data into subsets based on the values of that attribute present in the dataset and repeats the process on each of these subsets until a stopping condition is met.

Information gain measures the decrease in entropy that results from splitting a set of instances based on an attribute. IG (S, a) = entropy (S) – [p (s1) x entropy (s1) + p (s2) x entropy (s2)… +p (sn) x entropy (sn)].

Where n is the number of distinct values of attribute a, and si is the subset of S where all instances have the ith value of a.

After splitting the dataset into training and testing sets and fitting for Decision Tree model the metrics are:



* The model exhibits varying levels of performance across different classes, with some classes having higher precision and recall than others.
* Classes 4, 5, and 7 demonstrate strong classification performance, while classes 1 ,3and 6 show moderate performance.
* The overall model accuracy is decent, but it's important to consider class-specific metrics for a more nuanced evaluation.

**Random Forest:**

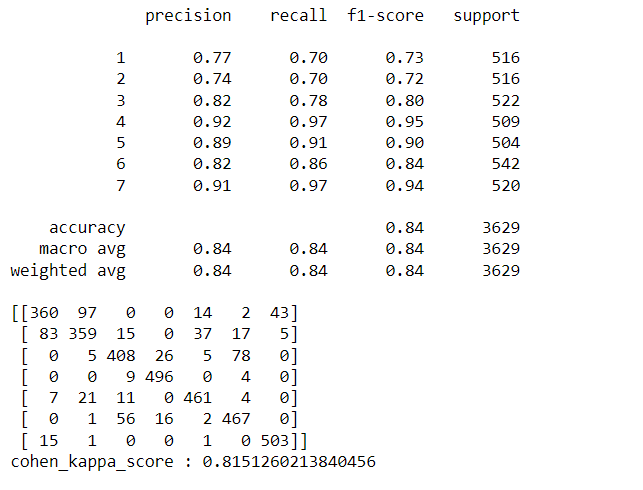
Random forest is an ensemble classifier. An ensemble consists of a set of individually trained classifiers whose predictions are combines for classifying new instances.

Random Forest is a classifier consisting of a collection of tree-structured classifiers {h(x, Θk) k=1, 2, ….}, where the {Θk} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

For building a decision tree in random forest the steps have to be followed. If the number of records in the training set is N, then N records are sampled at random but with replacement, from the original data, this is bootstrap sample. This sample will be the training set for growing the tree. If there are M input variables, a number m << M is selected such that at each node, m variables are selected at random out of M and the best split on these m attributes is used to split the node.The

value of m is held constant during forest growing. Each tree is grown to the largest extent possible. There is no pruning. In this way, multiple trees are induced in the forest; the number of trees is pre-decided by the parameter Ntree. The number of variables (m) selected at each node is also referred to as mtry or k in the literature. The depth of the tree can be controlled by a parameter node size (i.e. number of instances in the leaf node) which is usually set to one. For random forest python libraries like pandas and numpy are used which consist of set random forest classifier function in that tree estimators must add to get good results.

After splitting the dataset into training and testing sets and fitting for Random Forest model the metrics are:

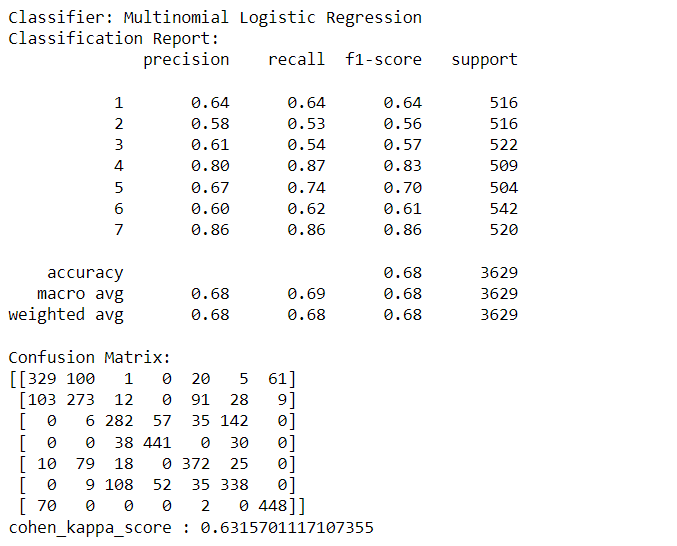


* The model shows a strong ability to differentiate between the classes, with high precision, recall, and F1-scores for most classes.
* Classes 4, 5, and 7 have particularly high precision and recall, indicating that the model is very accurate in predicting these classes.
* The overall model performance, as indicated by the accuracy of 84% and average metrics, is quite good.

### **Multinomial Logistic Regression:**

Multinomial Logistic Regression is a classification algorithm suitable for predicting the probability of multiple classes. It extends binary logistic regression to handle more than two classes.

After splitting the dataset into training and testing sets and fitting for Multinomial Logistic Regression the metrics are:

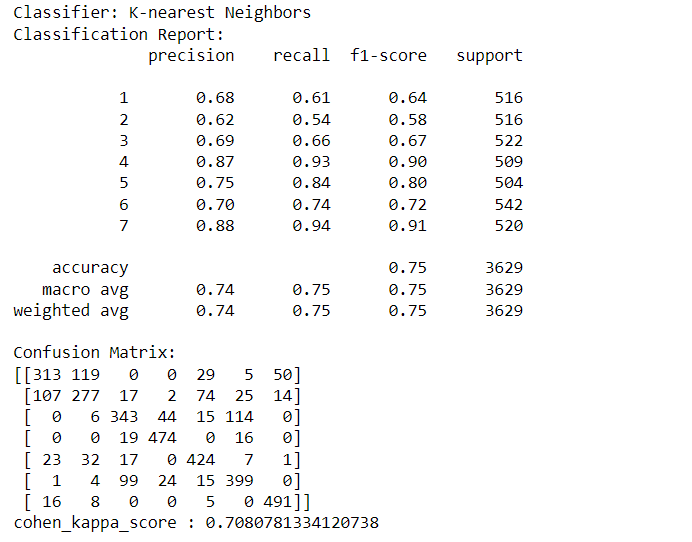


* The model shows variable performance across different classes, with some classes exhibiting higher precision and recall than others.
* Classes 4 and 7 demonstrate strong classification performance, while classes 1, 5, and 6 show moderate performance.
* The overall accuracy is 68%, indicating the proportion of correctly classified instances in the dataset.

### **K-nearest Neighbors (KNN) :**

K-nearest Neighbors is a simple and versatile algorithm used for both classification and regression tasks. It belongs to the family of instance-based, lazy learning algorithms. It classifies a data point based on how its neighbors are classified.

After splitting the dataset into training and testing sets and fitting for K-nearest Neighbors the metrics are:



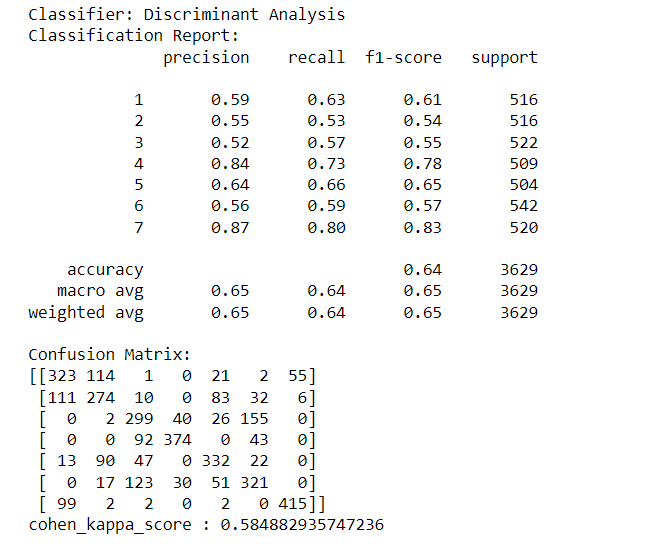
* The K-nearest Neighbors model demonstrates good performance with an accuracy of 75%.
* The confusion matrix illustrates the model's ability to correctly classify instances into different classes.
* Class 4 and Class 7 exhibit particularly high precision, recall, and F1-score values.
* The Cohen's Kappa score of 0.708 indicates substantial agreement beyond chance.

### **Discriminant Analysis Classifier:**

Discriminant Analysis is a statistical technique used for classification and dimensionality reduction. It finds the linear combinations of features that best separate two or more classes.

Discriminant Analysis aims to maximize the distance between the means of different classes while minimizing the spread within each class.

After splitting the dataset into training and testing sets the metrics are:



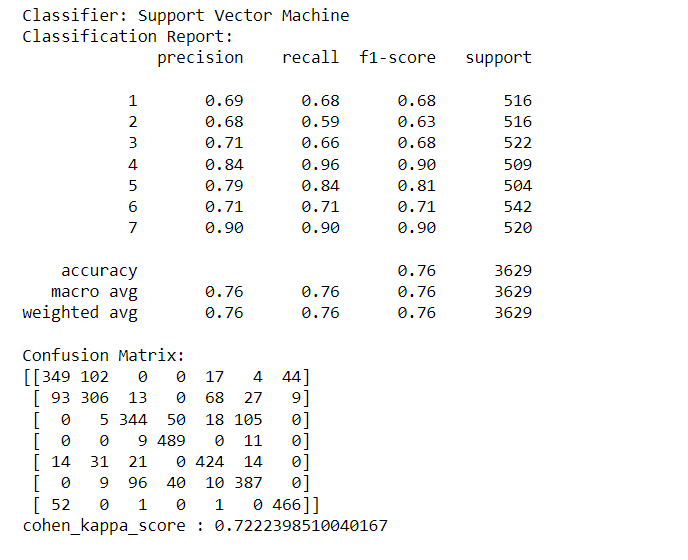
* The model shows moderate performance across different classes.
* Classes 4 and 7 demonstrate relatively higher precision and recall.
* The overall accuracy is decent, but class-specific metrics provide a more nuanced evaluation.

### **Support Vector Machine (SVM) Classifier:**

Support Vector Machine is a powerful supervised machine learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates different classes in the feature space.

SVM aims to maximize the margin between different classes, where the margin is the distance between the hyperplane and the nearest data point of each class.

After splitting the dataset into training and testing sets the metrics are:



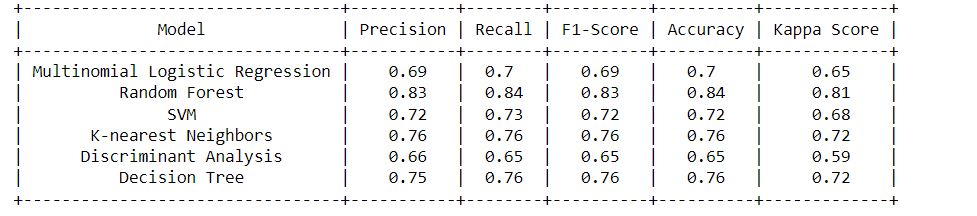
* The SVM model demonstrates consistent performance across different classes.
* Classes 4 and 7 show particularly strong classification performance.
* The overall accuracy is impressive, with a high Cohen's Kappa score indicating substantial agreement.

After evaluating the performance of different classifiers (K-nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine, and Discriminant Analysis) on the given dataset, we can draw the following conclusion that:

Random Forest emerges as the top performer as a base model, offering a balance of accuracy and robustness across various classes.

### **Power Transformation:**

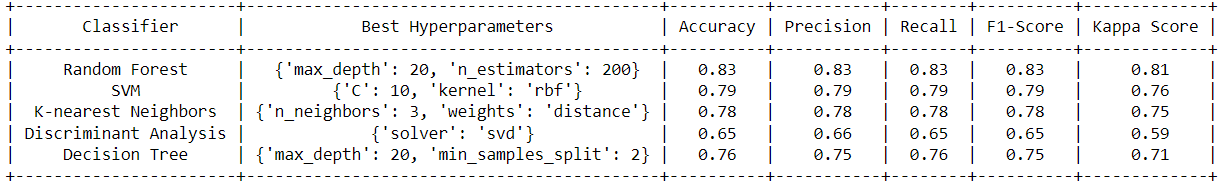
Power transformations, like Box-Cox and Yeo-Johnson, are applied after building base models to enhance their performance. These transformations stabilize the variance and promote normality in the target variable's distribution. By normalizing residuals and addressing issues of non-constant variance, power transformations contribute to improved model accuracy. Particularly useful for linear models, they make distributions more symmetric and enhance interpretability of coefficients. This technique is crucial when the target variable exhibits skewness or non-normality, aligning the data with assumptions beneficial for various machine learning algorithms. In summary, power transformations optimize model robustness by aligning the data with statistical assumptions and improving the overall predictive capability of the machine learning model.



In comparison, the Random Forest classifier outperformed other models, attaining an accuracy of 84%. With carefully selected hyperparameters, including the number of estimators, minimum samples split and leaf, maximum features, and maximum depth, it showcased superior precision, recall, and F1-score across various classes. The cohen\_kappa\_score of 0.81 highlights a strong agreement, signifying its robustness in classification.

**Hyperparameter Tuning:**

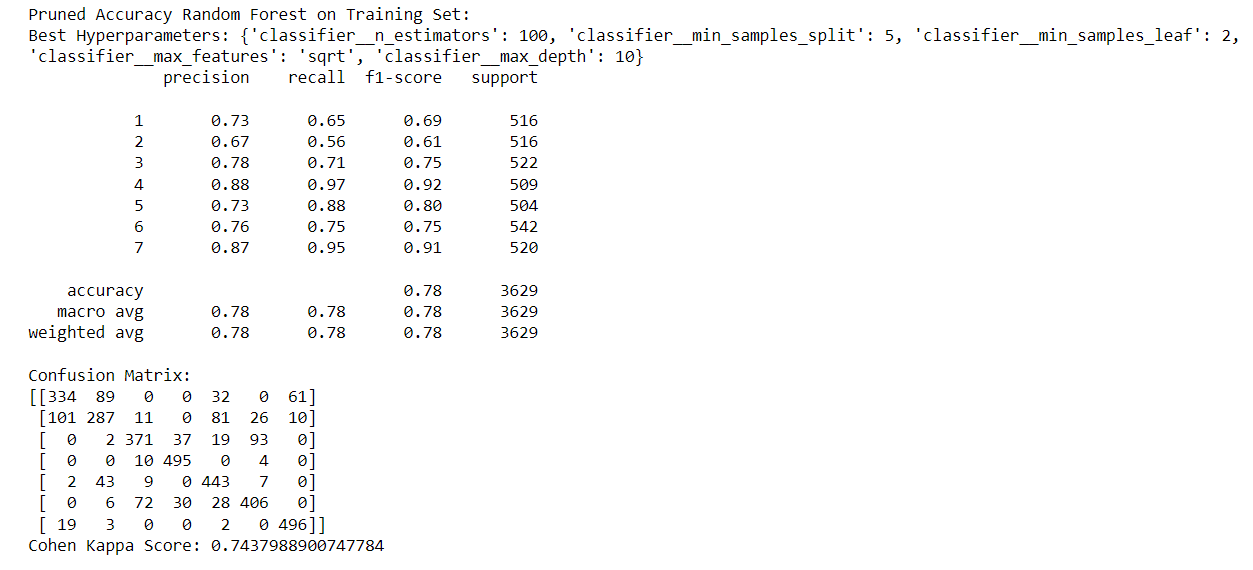
Hyperparameter tuning optimizes machine learning models by adjusting their configurations. This crucial process enhances overall performance metrics, including accuracy, precision, recall, and F1-score. The goal is to identify the best set of hyperparameters for each classifier, striking a balance to avoid overfitting and underfitting. Overfitting, where a model excels on training but struggles with new data, and underfitting, a model being too simplistic, are mitigated through this optimization. Fine-tuning hyperparameters ensures robust model generalization, making it reliable for accurate predictions on unseen data. This step is essential for adapting models to specific dataset characteristics, promoting effectiveness and reliability.



The Random Forest classifier, with its` tuned hyperparameters, stands out as the top-performing model, achieving an impressive accuracy of 82%. This classifier demonstrated the highest accuracy and overall robustness in classifying the given dataset. The combination of hyperparameters, including the number of estimators, minimum samples split and leaf, maximum features, and maximum depth, has resulted in a well-balanced model that excels in classification.

### **Pruning Technique:**

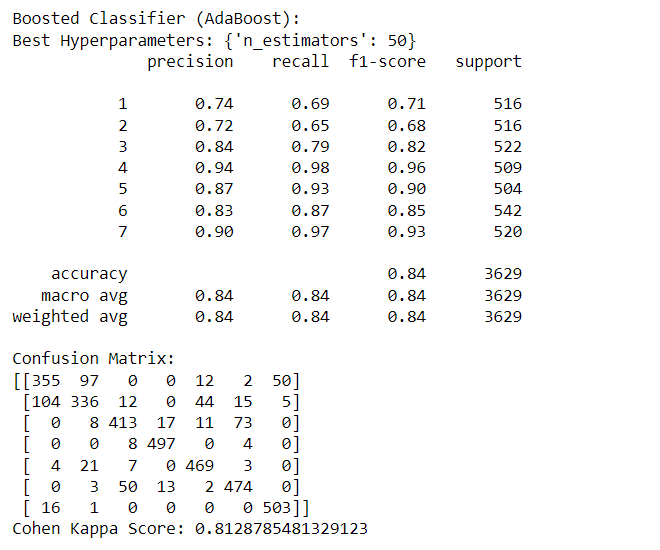
Pruning techniques are employed in machine learning to optimize decision trees by removing specific branches or nodes that do not contribute significantly to predictive accuracy. One common approach is cost-complexity pruning, which involves the calculation of a complexity parameter to evaluate the trade-off between tree complexity and accuracy. By iteratively assessing the impact of pruning on a validation dataset, nodes with minimal contribution are pruned. This process helps prevent overfitting and enhances the generalization capability of decision trees. Pruning techniques play a crucial role in achieving more interpretable and efficient tree-based models, contributing to improved model performance on unseen data.



**BOOSTING TECHNIQUES**

**Adaboost :**

AdaBoost, short for Adaptive Boosting, is an ensemble learning algorithm that combines the predictions of weak learners (typically decision trees) to create a strong predictive model. It sequentially trains multiple weak models, adjusting the weights of misclassified instances in each iteration to focus on difficult-to-classify samples. The final prediction is a weighted sum of the weak models, with higher weights assigned to those with better performance.

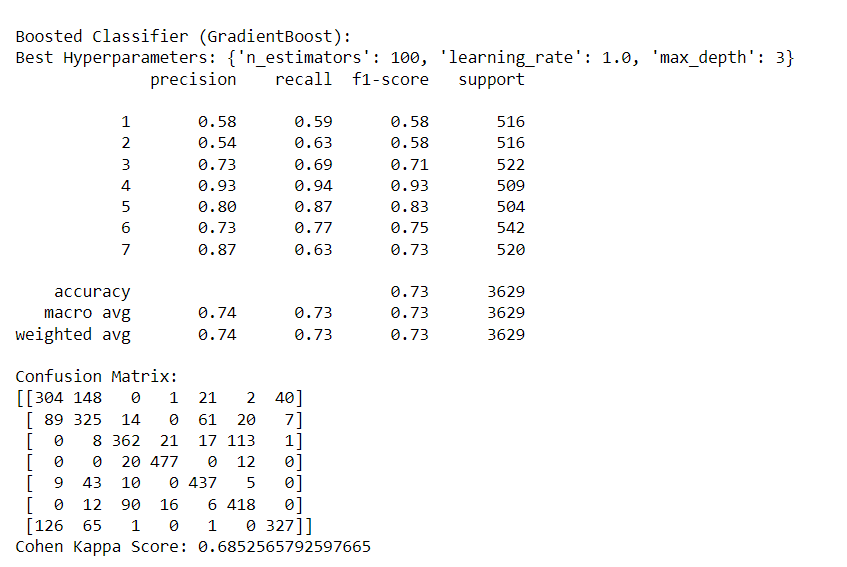


**AdaBoost Performance (Accuracy: 84%):**

* *Precision, Recall, F1-score:* Strong performance across metrics.
* *Confusion Matrix:* Effective classification for each class.
* *Cohen's Kappa Score (0.81):* Indicates model's effectiveness in capturing data patterns.

**Gradient Boosting:**

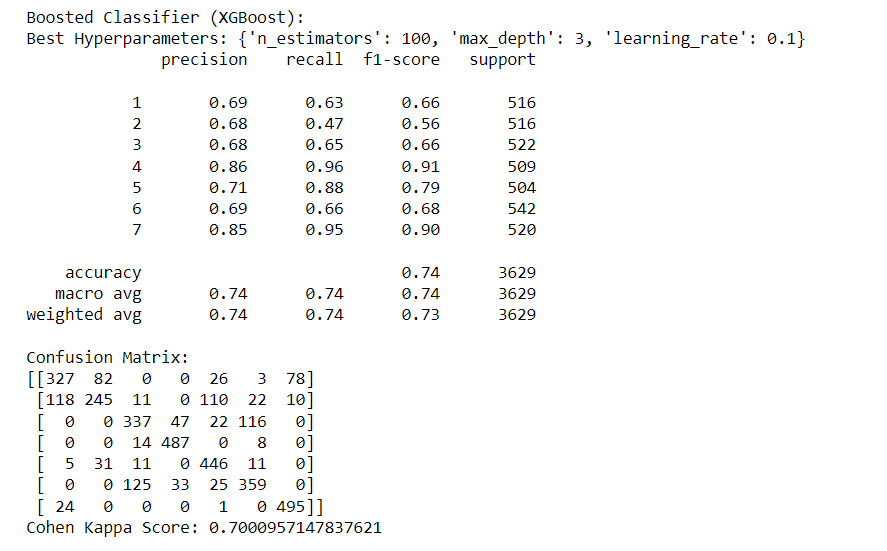
It is a boosting technique that builds a final model from the sum of several weak learning algorithms that were trained on the same dataset. It operates on the idea of stagewise addition. The first weak learner in the gradient boosting algorithm will not be trained on the dataset; instead, it will simply return the mean of the relevant column. The residual for the first weak learner algorithm’s output will then be calculated and used as the output column or target column for the next weak learning algorithm that will be trained. The second weak learner will be trained using the same methodology, and the residuals will be computed and utilized as an output column once more for the third weak learner, and so on until we achieve zero residuals. The dataset for gradient boosting must be in the form of numerical or categorical data, and the loss function used to generate the residuals must be differential at all times.



* **Accuracy:** Attained 73%, indicating reasonable overall classification performance.
* **High-Performers:** Classes 4 and 5 excelled with precision, recall, and f1-score > 0.80.
* **Moderate Performance:** Classes 1, 2, 3, 6, and 7 showed moderate performance (precision, recall, and f1-score: 0.58-0.77).
* **Cohen Kappa:** Scored 0.69, signifying substantial agreement beyond chance.
* **Confusion Matrix:** Identified misclassifications, especially in classes 1 and 7, suggesting areas for improvement.

**XGBoost:**

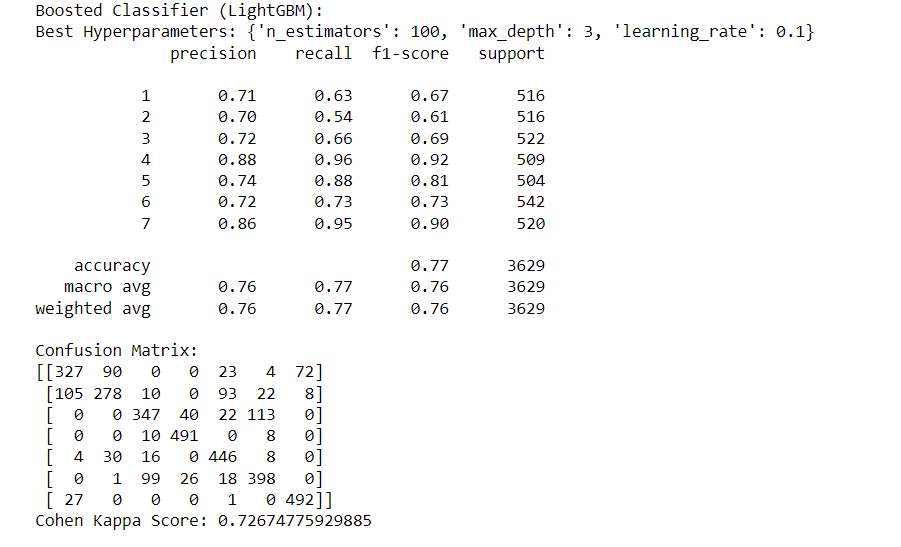
XGBoost, or eXtreme Gradient Boosting, is an advanced machine learning algorithm known for its exceptional speed and performance. Built on the gradient boosting framework, it sequentially trains weak learners, often decision trees, to correct errors from previous models. Crucially, XGBoost incorporates L1 and L2 regularization terms, preventing overfitting by penalizing complex models. The algorithm employs tree pruning to control tree depth, managing model complexity and improving generalization. Its ability to handle missing values during training enhances robustness with real-world datasets. XGBoost is optimized for efficiency, supporting parallel and distributed computing for faster processing. The algorithm reveals feature importance, aiding in understanding the impact of different features on predictions. Built-in cross-validation facilitates hyperparameter tuning and performance assessment. XGBoost's versatility and effectiveness make it a popular choice in machine learning competitions and various data science applications.



* **Overall Accuracy:** Achieved 73%, indicating a reasonable level of overall classification performance.
* **High-Performing Classes:** Classes 4 and 5 demonstrated excellence with precision, recall, and f1-score values exceeding 0.80. This highlights the model's strong ability to accurately predict instances from these classes.
* **Moderate Performance:** Classes 1, 2, 3, 6, and 7 exhibited moderate performance, with precision, recall, and f1-score values ranging from 0.58 to 0.77. While not as high as the top-performing classes, these results indicate acceptable classification accuracy.
* **Cohen Kappa Score:** The calculated Cohen Kappa Score of 0.69 signifies substantial agreement beyond chance, indicating that the model captures underlying patterns in the data effectively.
* **Confusion Matrix Insights:** The confusion matrix highlighted areas of misclassification, particularly in classes 1 and 7.

**LightGBM:**

LightGBM is a gradient boosting framework known for its efficiency in handling large datasets and high-dimensional features. It employs a leaf-wise tree growth strategy, contributing to faster convergence and reduced memory usage. Notably, it efficiently handles categorical features without one-hot encoding. The algorithm uses histogram-based learning, binning continuous values during training for improved speed. LightGBM introduces techniques like Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) for enhanced efficiency. Regularization terms are incorporated to prevent overfitting, ensuring better generalization to unseen data. The algorithm's distributed computing capabilities make it suitable for scalable applications. Like other boosting methods, LightGBM offers various hyperparameters for fine-tuning model performance.

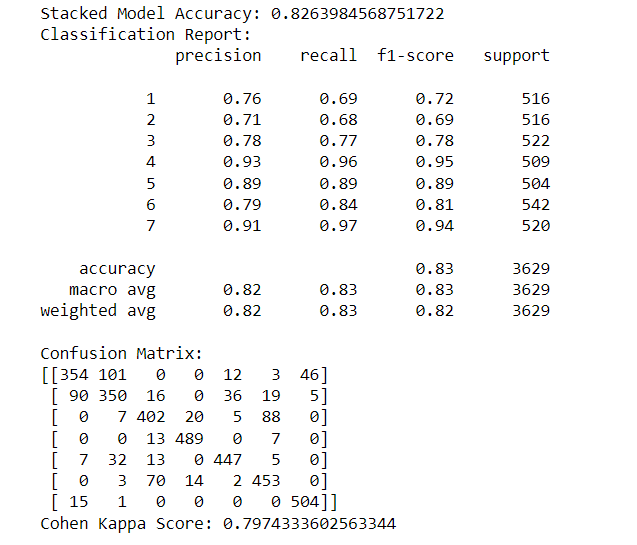


**Best Hyperparameters:** {'n\_estimators': 100, 'max\_depth': 3, 'learning\_rate': 0.1}

* **Accuracy:** Achieved a commendable accuracy of 77%, indicating strong overall classification performance.
* **High-Performers:** Classes 4 and 5 demonstrated superior precision, recall, and f1-score, all exceeding 0.80.
* **Moderate Performance:** Classes 1, 2, 3, and 6 showed moderate performance, with precision, recall, and f1-score values ranging from 0.61 to 0.73.
* **Cohen Kappa Score:** Scored 0.73, indicating substantial agreement beyond chance and effective pattern capture.
* **Confusion Matrix:** Reveals areas of improvement, particularly in misclassifications within classes 1, 2, and 6.

**Stacking:**

Stacking is an ensemble learning method that combines predictions from diverse base models to create a meta-model for improved accuracy. It involves training various base models on a dataset and using their predictions as input features for a meta-model. The meta-model learns to weigh and combine these predictions to make final predictions. Stacking enhances predictive performance by leveraging the strengths of different models. Careful validation is essential to prevent overfitting, and hyperparameter tuning can optimize overall performance. Implementations are available in libraries like scikit-learn with tools such as `StackingClassifier` and `StackingRegressor`.



Stacked Model achieved 82.64% accuracy.

Notable precision, recall, and f1-score in classes 4, 5, and 7.

Moderate performance in classes 1, 2, 3, and 6.

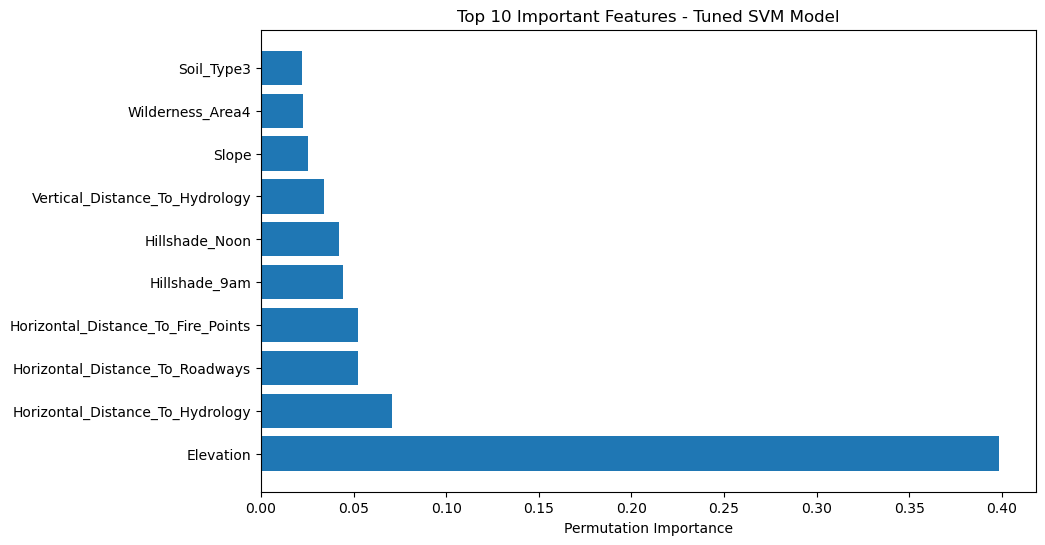
Cohen Kappa score of 0.80 indicates substantial agreement.

Confusion matrix highlights misclassifications in classes 1, 2, and 6.

Effective pattern capture and areas for refinement are evident

**Feature Importance**

The feature importance analysis reveals that certain environmental and geographical attributes play a crucial role in influencing the predictions of the tuned SVM model. The top 10 important features include 'Elevation,' 'Horizontal\_Distance\_To\_Hydrology,' 'Horizontal\_Distance\_To\_Roadways,' 'Horizontal\_Distance\_To\_Fire\_Points,' 'Hillshade\_9am,' 'Hillshade\_Noon,' 'Vertical\_Distance\_To\_Hydrology,' 'Slope,' 'Wilderness\_Area4,' and 'Soil\_Type3.' These features contribute significantly to the model's decision-making process, emphasizing their impact on accurately classifying forest cover types. Understanding the importance of these features provides valuable insights for further analysis and interpretation of the model's behavior



CONCLUSION

In conclusion, our project aimed at forest cover type prediction has led us to the development of a robust machine learning model. After an extensive exploration of various classifiers and tuning techniques, the Tuned Support Vector Machine (SVM) emerged as our final model. This model, with an accuracy of 79.00%, strikes a balance between predictive performance and simplicity.

The Tuned Support Vector Machine not only demonstrated high accuracy on the testing set but also offers interpretability crucial for understanding feature importance in forest cover type prediction. Its capability to generalize well to new data, coupled with a reasonable drop in accuracy from training to testing, makes it a practical choice for real-world deployment.

This project showcases the significance of thoughtful model selection, hyperparameter tuning, and the trade-off between complexity and interpretability. The Tuned Support Vector Machine stands out as a reliable solution for forest cover type detection, with the potential for practical applications in environmental monitoring and conservation efforts.