Forest Cover Type Prediction

HarvardX Data Science Professional Certificate: PH125.9x

Data Science Initiative
The Palestinian Central Bureau of Statistics (PCBS)
Arab American University of Palestine (AAUP)

Mohammed K. M. Elhabbash

06 February, 2022

Contents

1. Introduction 3
   1.1 Investigation the dataset 3
   1.2 Wrangling and cleaning data 7
   1.3 Statistic summary and visualizing data after wrangling and cleaning 8
   1.4 Split data into training data set and validation data set 11
2. Algorithms 11
   2.1 Algorithm 1 (linear algorithm): Linear discriminant analysis (L.D.A.) 11
   2.2 Algorithm 2 (nonlinear algorithm): Classification And Regression Tree (C.A.R.T) 12
3. Results 12
   3.1 Result of (L.D.A.) model 12
   3.2 Result of (C.A.R.T) model 13
   3.2.1 The optimal complexity parameter "C.P." 13
   3.2.2 Redefine the model using the train_data and optimal cp 14
   3.3. Summarize accuracy of models 15
   3.4. Compare accuracy and inter-rater reliability of models 15
   3.5. Estimating skill of C.A.R.T. on the validation dataset 15
4. Conclusion 16

1. Introduction

As a type of artificial intelligence (AI), machine learning (ML) lets software applications predict outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Prediction and classification of things are the essential and famous applications of machine learning. There are various algorithms used to categorize things, such as Logistic Regression, Naive Bayes, K-Nearest Neighbors, Decision Tree, Support Vector Machines, and Linear discriminant analysis. Studying the Forest Cover Type, in this Paper, uses two algorithms to classify the cover type of forest depending on the data set from Kaggle. The first is a linear algorithm, Linear discriminant analysis (LDA), the second is a nonlinear algorithm, Classification And Regression Tree (CART). To clarify the purpose of this paper, first, we want to investigate specifications and show samples of the data set.

1.1 Investigation the dataset

The source of used data set in this paper is from a competition, Playground Prediction Competition, launched in Dec 2021 Kaggle. The data is artificially generated by a GAN that was trained on data from the Forest Cover Type Prediction. The goal here in this paper is to predict the Cover_Type class for each Id in the data set depending on the rest of data set. Data set is with dimension of 4000000, 56. A sample of data set is shown below

```
## Rows: 4,000,000
## Columns: 56
## $ Id                                 <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1~
## $ Elevation                          <int> 3189, 3026, 3106, 3022, 2906, 3115,~
## $ Aspect                             <int> 40, 182, 13, 276, 186, 144, 61, 94,~
## $ Slope                              <int> 8, 5, 7, 13, 13, 2, 5, 4, 12, 16, 1~
## $ Horizontal_Distance_To_Hydrology   <int> 30, 280, 351, 192, 266, 415, 312, 1~
## $ Vertical_Distance_To_Hydrology     <int> 13, 29, 37, 16, 22, 61, 32, 63, 22,~
## $ Horizontal_Distance_To_Roadways    <int> 3270, 3270, 2914, 3034, 2916, 3371,~
## $ Hillshade_9am                      <int> 206, 233, 208, 207, 231, 223, 225, ~
## $ Hillshade_Noon                     <int> 234, 240, 234, 238, 231, 231, 248, ~
## $ Hillshade_3pm                      <int> 193, 106, 137, 156, 154, 131, 163, ~
## $ Horizontal_Distance_To_Fire_Points <int> 4873, 5423, 5269, 2866, 2642, 2629,~
## $ Wilderness_Area1                   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ Wilderness_Area2                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Wilderness_Area3                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Wilderness_Area4                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type1                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type2                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type3                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type4                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type5                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type6                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type7                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type8                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type9                         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type10                        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type11                        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type12                        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type13                        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type14                        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type15                        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type16                        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

```
## $ Soil_Type17                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type18                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type19                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type20                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type21                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type22                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type23                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type24                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type25                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type26                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type27                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type28                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type29                      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ Soil_Type30                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type31                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type32                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type33                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type34                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type35                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type36                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type37                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type38                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type39                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Soil_Type40                      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Cover_Type                       <int> 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2,~
```

The details of data is described for each column as

Name	Describtion
Id	**A primary key (unique value for each record,row)**
Elevation	**Elevation in meters**
Aspect	**Aspect in degrees azimuth**
Slope	**Slope in degrees**
Horizontal_Distance_To_Hydrology	**Horz Dist to nearest surface water features**
Vertical_Distance_To_Hydrology	**Vert Dist to nearest surface water features**
Horizontal_Distance_To_Roadways	**Horz Dist to nearest roadway**
Hillshade_9am	**Hillshade index at 9am, summer solstice**
Hillshade_Noon	**Hillshade index at noon, summer solstice**
Hillshade_3pm	**Hillshade index at 3pm, summer solstice**
Horizontal_Distance_To_Fire_Points	**Horz Dist to nearest wildfire ignition points**

Name	Describtion
Wilderness_Area1	Rawah Wilderness Area
Wilderness_Area2	Neota Wilderness Area
Wilderness_Area3	Comanche Peak Wilderness Area
Wilderness_Area4	Cache la Poudre Wilderness Area
Soil_Type1	Cathedral family - Rock outcrop complex, extremely stony
Soil_Type2	Vanet - Ratake families complex, very stony
Soil_Type3	Haploborolis - Rock outcrop complex, rubbly
Soil_Type4	Ratake family - Rock outcrop complex, rubbly
Soil_Type5	Vanet family - Rock outcrop complex complex, rubbly
Soil_Type6	Vanet - Wetmore families - Rock outcrop complex, stony
Soil_Type7	Gothic family
Soil_Type8	Supervisor - Limber families complex
Soil_Type9	Troutville family, very stony
Soil_Type10	Bullwark - Catamount families - Rock outcrop complex, rubbly
Soil_Type11	Bullwark - Catamount families - Rock land complex, rubbly
Soil_Type12	Legault family - Rock land complex, stony
Soil_Type13	Catamount family -Rock land- Bullwark family complex, rubbly
Soil_Type14	Pachic Argiborolis - Aquolis complex
Soil_Type15	Unspecified in the USFS Soil and ELU Survey
Soil_Type16	Cryaquolis - Cryoborolis complex
Soil_Type17	Gateview family - Cryaquolis complex
Soil_Type18	Rogert family, very stony
Soil_Type19	Typic Cryaquolis - Borohemists complex
Soil_Type20	Typic Cryaquepts - Typic Cryaquolls complex
Soil_Type21	Typic Cryaquolls - Leighcan family, till substratum complex
Soil_Type22	Leighcan family, till substratum, extremely bouldery
Soil_Type23	Leighcan family, till substratum - Typic Cryaquolls complex
Soil_Type24	Leighcan family, extremely stony
Soil_Type25	Leighcan family, warm, extremely stony
Soil_Type26	Granile - Catamount families complex, very stony
Soil_Type27	Leighcan family, warm - Rock outcrop complex, extremely stony
Soil_Type28	Leighcan family - Rock outcrop complex, extremely stony
Soil_Type29	Como - Legault families complex, extremely stony
Soil_Type30	Como family -Rock land- Legault family complex, extremely stony
Soil_Type31	Leighcan - Catamount families complex, extremely stony
Soil_Type32	Catamount family - Rock outcrop - Leighcan family complex, extremely stony
Soil_Type33	Leighcan - Catamount families - Rock outcrop complex, extremely stony
Soil_Type34	Cryorthents - Rock land complex, extremely stony
Soil_Type35	Cryumbrepts - Rock outcrop - Cryaquepts complex
Soil_Type36	Bross family - Rock land - Cryumbrepts complex, extremely stony
Soil_Type37	Rock outcrop - Cryumbrepts - Cryorthents complex, extremely stony
Soil_Type38	Leighcan - Moran families - Cryaquolls complex, extremely stony
Soil_Type39	Moran family -Cryorthents- Leighcan family complex,extremely stony
Soil_Type40	Moran family -Cryorthents- Rock land complex, extremely stony
Cover_Type	7 types, integers 1 to 7- Forest Cover Type designation

In the following some statistics summary of the data, i.e. Min.,1st Qu., Median,Mean,3rd Qu., and Max. for Elevation, Aspect, Slope, Horizontal_Distance_To_Hydrology, Vertical_Distance_To_Hydrology, Horizontal_Distance_To_Roadways, Hillshade_9am, Hillshade_Noon, Hillshade_3pm, and Horizontal_Distance_To_Fire_Points

Elevation	Aspect	Slope
Min. :1773	**Min. :-33.0**	Min. :-3.0
1st Qu.:2760	**1st Qu.: 60.0**	1st Qu.: 9.0
Median :2966	**Median :123.0**	Median :14.0
Mean :2980	**Mean :151.6**	Mean :15.1
3rd Qu.:3217	**3rd Qu.:247.0**	3rd Qu.:20.0
Max. :4383	**Max. :407.0**	Max. :64.0

Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology
Min. : -92.0	Min. :-317.00
1st Qu.: 110.0	1st Qu.: 4.00
Median : 213.0	Median : 31.00
Mean : 271.3	Mean : 51.66
3rd Qu.: 361.0	3rd Qu.: 78.00
Max. :1602.0	Max. : 647.00

Hillshade_9am	Hillshade_Noon	Hillshade_3pm
Min. : -4.0	**Min. : 49.0**	Min. :-53.0
1st Qu.:198.0	**1st Qu.:210.0**	1st Qu.:115.0
Median :218.0	**Median :224.0**	Median :142.0
Mean :211.8	**Mean :221.1**	Mean :140.8
3rd Qu.:233.0	**3rd Qu.:237.0**	3rd Qu.:169.0
Max. :301.0	**Max. :279.0**	Max. :272.0

Horizontal_Distance_To_Roadways	Horizontal_Distance_To_Fire_Points
Min. :-287	Min. :-277
1st Qu.: 822	1st Qu.: 781
Median :1436	Median :1361
Mean :1767	Mean :1581
3rd Qu.:2365	3rd Qu.:2084
Max. :7666	Max. :8075

For wilderness area designation, the following table display how many records for each “wilderness area” variable

Wilderness area type	Number of records
Wilderness_Area1	**1044772**
Wilderness_Area2	**166644**
Wilderness_Area3	**2614293**
Wilderness_Area4	**87276**

For Soil Type designation, the following table display how many records for each “Soil Type” variable

Soil type	Number of records
Soil_Type1	**67366**
Soil_Type2	**123584**
Soil_Type3	**17102**
Soil_Type4	**151651**
Soil_Type5	**62861**
Soil_Type6	**31891**
Soil_Type7	**0**
Soil_Type8	**11599**
Soil_Type9	**43572**
Soil_Type10	**218163**
Soil_Type11	**111941**
Soil_Type12	**73160**
Soil_Type13	**125181**
Soil_Type14	**59906**
Soil_Type15	**0**
Soil_Type16	**63554**
Soil_Type17	**82687**
Soil_Type18	**53745**
Soil_Type19	**55245**
Soil_Type20	**69472**
Soil_Type21	**46156**
Soil_Type22	**125384**
Soil_Type23	**196683**
Soil_Type24	**100087**
Soil_Type25	**13033**
Soil_Type26	**54108**
Soil_Type27	**47063**
Soil_Type28	**42831**
Soil_Type29	**89094**
Soil_Type30	**115468**
Soil_Type31	**109973**
Soil_Type32	**149848**
Soil_Type33	**151283**
Soil_Type34	**47980**
Soil_Type35	**64214**
Soil_Type36	**42851**
Soil_Type37	**48830**
Soil_Type38	**163006**
Soil_Type39	**156957**
Soil_Type40	**126474**

1.2 Wrangling and cleaning data

1- “Horizontal_Distance_To_Hydrology”, ” Vertical_Distance_To_Hydrology”, “Horizontal_Distance_To_Roadways”, “Hillshade_9am”, “Hillshade_3pm”, and “Horizontal_Distance_To_Fire_Point” have negative values, but the distance should be only a positive value.

2- “Soil_Type7”, and “Soil_Type15” are with only 0 value. This means that we can drop these two columns.

3- “Cover_Type” has 1 value for level 5, this means that this single value will appear in the training data set or will appear in validation-data-set, and this will cause a problem in the “train” or in the “predict” function. the solution for this problem is to drop level 5 in Cover_Type, so the levels will be:1, 2, 3, 4, 6, 7 1,2,3,4,6, and 7.

1.3 Statistic summary and visualizing data after wrangling and cleaning

Elevation

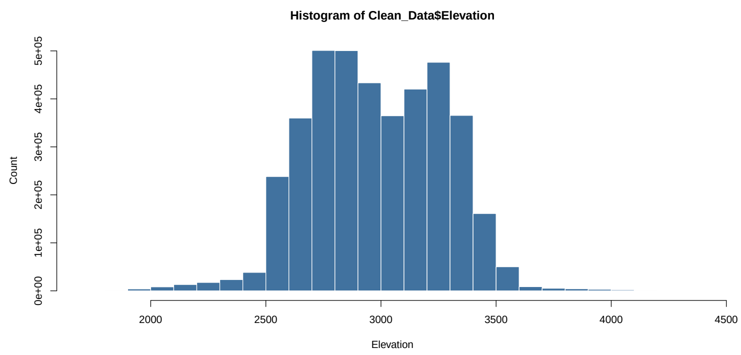


Figure 1: *Histogram of Elevation*

Min.	Q_1	Median	Mean	Q_3	Max
1773	2760	2966	2980.1916665	3217	4383

Aspect

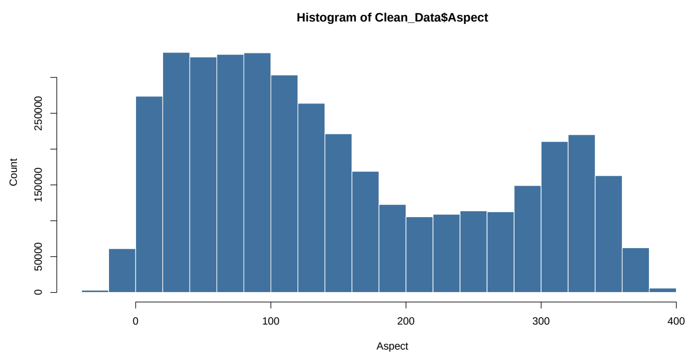


Figure 2: *Histogram of Aspect*

Min.	Q_1	Median	Mean	Q_3	Max
-33	60	123	151.5856804	247	407

Slope

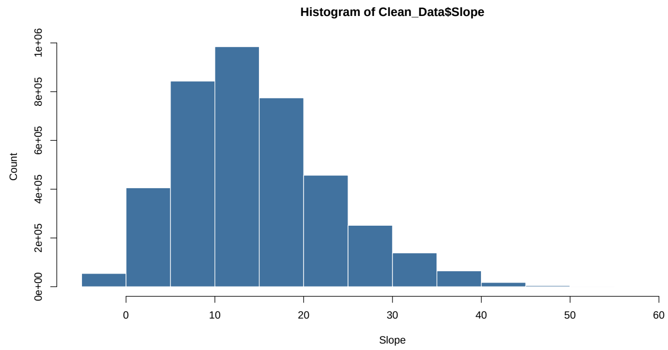


Figure 3: *Histogram of Slope*

Min.	Q_1	Median	Mean	Q_3	Max
-3	9	14	15.097531	20	64

Horizontal Distance To Hydrology

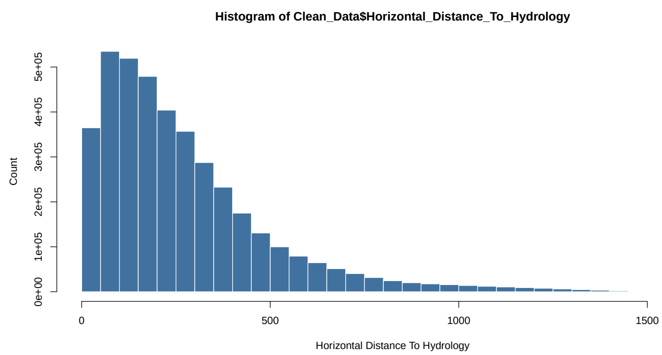


Figure 4: *Histogram of Horizontal Distance To Hydrology*

Min.	Q_1	Median	Mean	Q_3	Max
0	110	213	271.3392753	361	1602

Vertical Distance To Hydrology

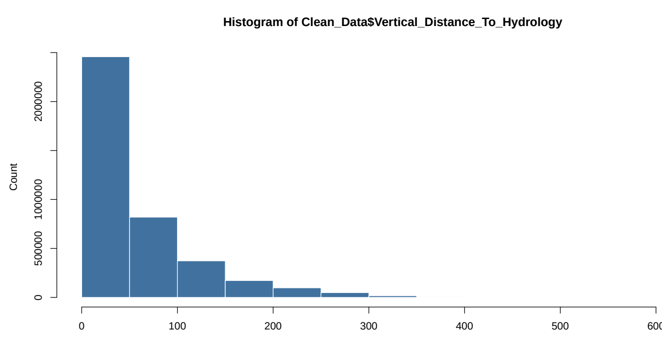


Figure 5: *Histogram of Vertical Distance To Hydrology*

Min.	Q_1	Median	Mean	Q_3	Max
0	7	34	55.6091704	79	647

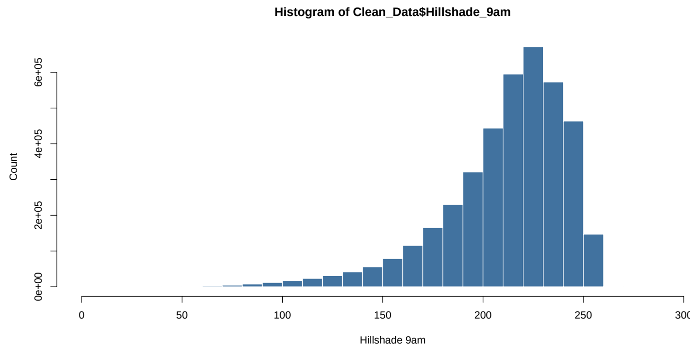


Figure 6: *Histogram of Hillshade 9am*

Min.	Q_1	Median	Mean	Q_3	Max
0	198	218	211.8375555	233	301

Hillshade Noon

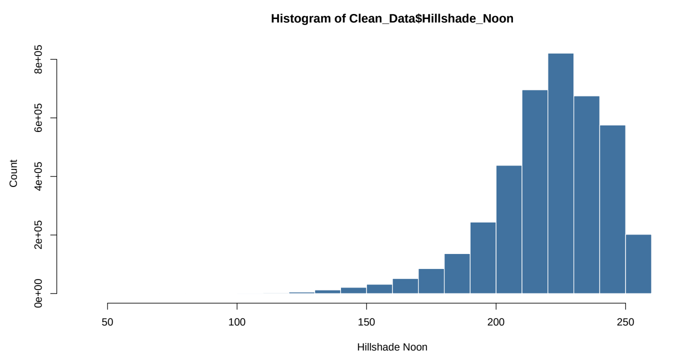


Figure 7: *Histogram of Hillshade Noon*

Min.	Q_1	Median	Mean	Q_3	Max
49	210	224	221.0614438	237	279

Hillshade 3pm

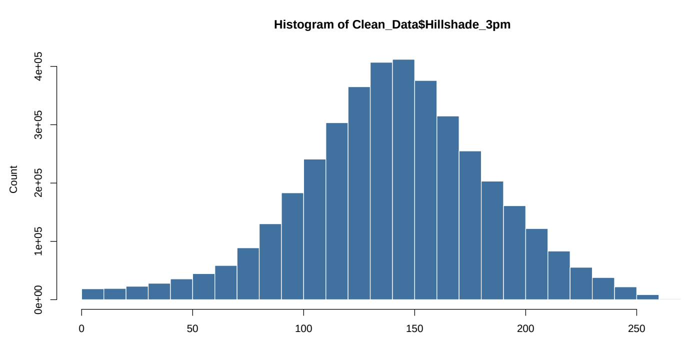


Figure 8: *Histogram of Hillshade 3pm*

Min.	Q_1	Median	Mean	Q_3	Max
0	115	142	140.908984	169	272

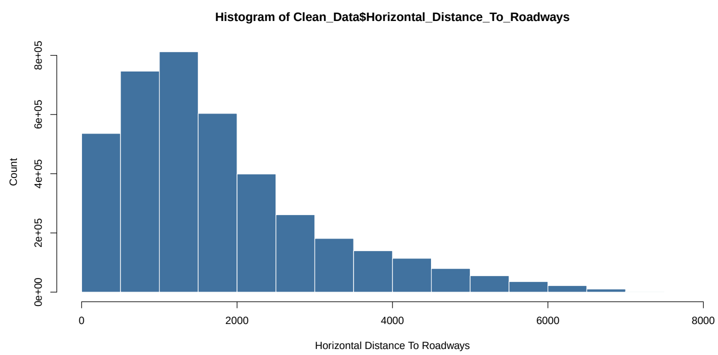


Figure 9: *Histogram of Horizontal Distance To Roadways*

Min.	Q_1	Median	Mean	Q_3	Max
0	822	1436	1767.6933582	2365	7666

Horizontal Distance To Fire Points

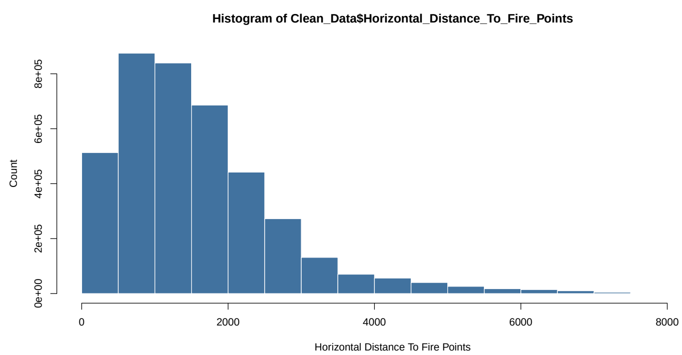


Figure 10: *Histogram of Horizontal Distance To Fire Points*

Min.	Q_1	Median	Mean	Q_3	Max
0	781	1361	1582.2932686	2084	8075

1.4 Split data into training data set and validation data set

For the purpose of machine learning we will split our data set into training data which is (80%) of the original data set, and validation data set which is (20%) of the original data set, so the training data set has 3199995 records for each 54 column, while validation data set has 800004 records for each 54 column.

The levels of variable Cover_Type is 1, 2, 3, 4, 6, 7 , and summary of the class distribution is shown below

```
##      freq percentage
## 1 1174508      36.70
## 2 1809669      56.55
## 3  156569       4.89
## 4     301       0.01
## 6    9140       0.29
## 7   49808       1.56
```


2. Algorithms

Machine learning in general uses training data set to train the model "Algorithm" to predict the results depending on the model. Piece of training data set plays as variables while the other piece is consedered as combination can be used as a linear classifier or, more typically, to reduce dimensionality before further classification. L.D.A. is closely connected to ANOVA and regression analysis, both of which aim to represent one dependent variable as a linear mixture of other traits or data. L.D.A. is a generalization of Fisher's linear discriminant. There is an option to extend the analysis used in the derivation of the Fisher discriminant to find a subspace that appears to contain all of the class variability when there are more than two classes. This generalization is due to C.R.Rao.

```
##             used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   2641394  141.1    7819660  417.7   7148267  381.8
## Vcells 363065041 2770.0  541516382 4131.5 450938507 3440.4
```

```
## [1] 1e+10
```

```
##             used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   2641415  141.1    7819660  417.7   7148267  381.8
## Vcells 363065093 2770.0  541516382 4131.5 450938507 3440.4
```

2.2 Algorithm 2 (nonlinear algorithm): Classification And Regression Tree (C.A.R.T)

The Classification and regression tree (C.A.R.T) approach is one of the most basic and oldest methods. It is used to forecast outcomes based on certain predictor factors. Because they need relatively minimal data pre-processing, they are ideal for data mining jobs. Decision tree models are simple to learn and apply, which provides them with a significant advantage over other analytical models. But trees can be very non-robust. A small change in the training data can result in a large change in the tree and consequently in the final predictions.

3. Results

3.1 Result of (L.D.A.) model

The following is the confusion matrix and statistics for L.D.A. model, with overall Accuracy of 0.8782869

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      1      2      3      4      6      7
##          1 272757  26316      0      0      0   7325
##          2  12582 423030  23436      0    553      0
##          3      0      0      3      0      0      0
##          4    213   1750   1417     33     49      2
##          6      0    560  14287     43   1684      0
##          7   8076    762      0      0      0   5126
##
## Overall Statistics
##
##                Accuracy : 0.8783
##                  95% CI : (0.8776, 0.879)
##     No Information Rate : 0.5655
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7722
```

```
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2  Class: 3  Class: 4 Class: 6 Class: 7
## Sensitivity            0.9289   0.9350 7.664e-05 4.342e-01 0.736658 0.411628
## Specificity            0.9336   0.8948 1.000e+00 9.957e-01 0.981334 0.988778
## Pos Pred Value         0.8902   0.9204 1.000e+00 9.527e-03 0.101605 0.367087
## Neg Pred Value         0.9577   0.9137 9.511e-01 9.999e-01 0.999232 0.990679
## Prevalence             0.3670   0.5655 4.893e-02 9.500e-05 0.002857 0.015566
## Detection Rate         0.3409   0.5288 3.750e-06 4.125e-05 0.002105 0.006407
## Detection Prevalence   0.3830   0.5745 3.750e-06 4.330e-03 0.020717 0.017455
## Balanced Accuracy      0.9312   0.9149 5.000e-01 7.150e-01 0.858996 0.700203
```

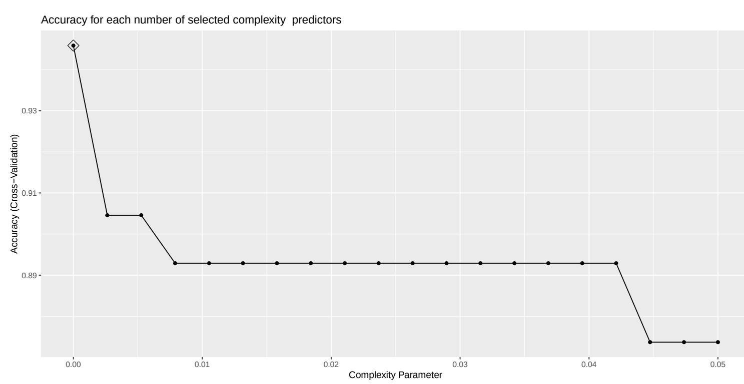
3.2 Result of (C.A.R.T) model

On the other hand, the C.A.R.T as a non-linear algorithm needs to be tuned for the complexity predictors "C.P.". In our model, we will try a C.P. array of length 20, with a minimum value of 0, and a maximum value of 0.05. The figure below shows that the best value of C.P which guarantees maximum accuracy is zero, i.e. Accuracy is max. when C.P.=0

```
##              used   (Mb) gc trigger    (Mb)   max used    (Mb)
## Ncells   2699903  144.2   10329330   551.7   12911662   689.6
## Vcells 381692735 2912.1 1603822936 12236.2 3132466669 23898.9
```

```
## [1] 1e+10
```

```
##              used   (Mb) gc trigger   (Mb)   max used    (Mb)
## Ncells   2699918  144.2    8263464  441.4   12911662   689.6
## Vcells 381692769 2912.1 1283058349 9789.0 3132466669 23898.9
```



3.2.1 The optimal complexity parameter "C.P."

The complexity parameter (cp) is used to control the size of the decision tree and to select the optimal tree size. If the cost of adding another variable to the decision tree from the current node is above the value of cp, then tree building does not continue. We could also say that tree construction does not continue unless it would decrease the overall lack of fit by a factor of cp.

```
## [1] "Optimal cp parameter = 0"
```

3.2.2 Redefine the model using the train_data and optimal cp

```
##              used   (Mb) gc trigger    (Mb)   max used    (Mb)
## Ncells    9197152  491.2   14965750   799.3   12911662   689.6
## Vcells 1117225931 8523.8 1847780021 14097.5 3132466669 23898.9
```

```
## [1] 1e+10
```

```
##              used   (Mb) gc trigger    (Mb)   max used    (Mb)
## Ncells    9197074  491.2   14965750   799.3   12911662   689.6
## Vcells 1117225810 8523.8 1847780021 14097.5 3132466669 23898.9
```

Now we will re-train the model with the same data set but using the optimized value of the complexity predictors C.P., so the confusing matrix for the second algorithm after optimization shows how (C.A.R.T) foretell each cover type of forest and compare the prediction with the real cover type reference for each cover type "Levels", with overall Accuracy of 0.9459565

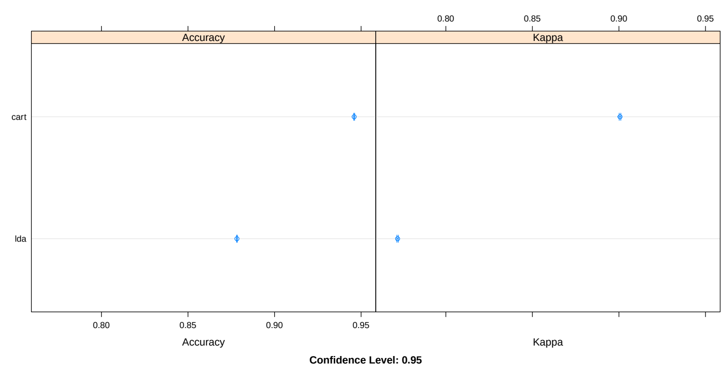
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      1      2      3      4      6      7
##          1 279116  11546      2      0      0   3595
##          2  12011 433917   4717      0    314     98
##          3      0   6686  33828     56    840      0
##          4      0      0     48     17      1      0
##          6      0    188    548      3   1131      0
##          7   2501     81      0      0      0   8760
##
## Overall Statistics
##
##                Accuracy : 0.946
##                  95% CI : (0.9455, 0.9465)
##     No Information Rate : 0.5655
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9005
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3  Class: 4 Class: 6 Class: 7
## Sensitivity            0.9506   0.9591  0.86422 2.237e-01 0.494751  0.70344
## Specificity            0.9701   0.9507  0.99003 9.999e-01 0.999074  0.99672
## Pos Pred Value         0.9485   0.9620  0.81690 2.576e-01 0.604813  0.77235
## Neg Pred Value         0.9713   0.9470  0.99299 9.999e-01 0.998553  0.99532
## Prevalence             0.3670   0.5655  0.04893 9.500e-05 0.002857  0.01557
## Detection Rate         0.3489   0.5424  0.04228 2.125e-05 0.001414  0.01095
## Detection Prevalence   0.3678   0.5638  0.05176 8.250e-05 0.002337  0.01418
## Balanced Accuracy      0.9603   0.9549  0.92713 6.118e-01 0.746912  0.85008
```

3.3. Summarize accuracy of models

According to statistics, accuracy is the degree to which the information accurately describes the phenomena it is designed to measure. While the Kappa coefficient measures inter-rater reliability (as well as intra-rater reliability) and is commonly used in qualitative (categorical) items. Basically, rater reliability is an indication of how well the data collected by the study reflect the variables measured. The accuracy and Kappa of the two algorithms are compared in the figure below. The figure shows more high levels of "Accuracy" and inter-rater reliability "Kappa" for the C.A.R.T algorithm over the L.D.A. algorithm.

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, cart
## Number of resamples: 10
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda  0.8778242 0.8780991 0.8782281 0.8782823 0.8784656 0.8788563    0
## cart 0.9455372 0.9457852 0.9459094 0.9459977 0.9460319 0.9468469    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda  0.7712989 0.7717935 0.7720088 0.7721268 0.7724681 0.7731645    0
## cart 0.8997872 0.9002797 0.9004448 0.9006121 0.9007083 0.9020944    0
```

3.4. Compare accuracy and inter-rater reliability of models



3.5. Estimating skill of C.A.R.T. on the validation dataset

```
##                used    (Mb) gc trigger    (Mb)   max used    (Mb)
## Ncells    9248192   494.0   28784568  1537.3   16084503   859.1
## Vcells 1848470965 14102.8 3211107786 24498.9 3211101327 24498.8
```

```
## [1] 1e+10
```

```
##                used    (Mb) gc trigger    (Mb)   max used    (Mb)
## Ncells    9248207   494.0   28784568  1537.3   16084503   859.1
## Vcells 1848471000 14102.8 3211107786 24498.9 3211101327 24498.8
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      1      2      3      4      6      7
##          1 279116  11546      2      0      0   3595
##          2  12011 433917   4717      0    314     98
##          3      0   6686  33828     56    840      0
##          4      0      0     48     17      1      0
##          6      0    188    548      3   1131      0
##          7   2501     81      0      0      0   8760
##
## Overall Statistics
##
##                Accuracy : 0.946
##                  95% CI : (0.9455, 0.9465)
##     No Information Rate : 0.5655
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9005
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3  Class: 4 Class: 6 Class: 7
## Sensitivity            0.9506   0.9591  0.86422 2.237e-01 0.494751  0.70344
## Specificity            0.9701   0.9507  0.99003 9.999e-01 0.999074  0.99672
## Pos Pred Value         0.9485   0.9620  0.81690 2.576e-01 0.604813  0.77235
## Neg Pred Value         0.9713   0.9470  0.99299 9.999e-01 0.998553  0.99532
## Prevalence             0.3670   0.5655  0.04893 9.500e-05 0.002857  0.01557
## Detection Rate         0.3489   0.5424  0.04228 2.125e-05 0.001414  0.01095
## Detection Prevalence   0.3678   0.5638  0.05176 8.250e-05 0.002337  0.01418
## Balanced Accuracy      0.9603   0.9549  0.92713 6.118e-01 0.746912  0.85008
```

One of the most important steps during machine learning is to test the chosen algorithm using another data set "validation data set". Validation data set is a sub data set that is split from the original data set. The overall "Accuracy" for the validation data set is 0.9459565. The following table depicts the statistical summary for the algorithm C.A.R.T. when using validation data set.

4. Conclusion

In this paper machine learning is used to predict the forest cover type using two different models; a linear model: Linear discriminant analysis (L.D.A.) and a nonlinear model: Classification And Regression Tree (C.A.R.T). Classification And Regression Tree (C.A.R.T) as a nonlinear algorithm shows its advantage over the Linear discriminant analysis (L.D.A.) as a linear algorithm when comparing Accuracy and intra-rater reliability for both.

The Accuracy of C.A.R.T model is 0.9459565, while the accuracy of L.D.A. is 0.8782869 The intra-rater reliability for C.A.R.T model is 0.9005432, while the intra-rater reliability for L.D.A. is 0.7721608.