MDS Final Project

4/24/2022

Understanding Dataset:

Let's first import all the required libraries -

```
library(tidyverse) # for data exploration and manipulation
library(randomForest)
library(corrplot)
library(RColorBrewer)
library(expss)
library(ggplot2)
library(gbm)
library(e1071)
library(ROCR)
library(caret)
```

Now that we have imported libraries, our first step is to read the data.

```
# Reading the data set:
forestdata = read.csv('data/covtype.csv', header=TRUE)

# Displaying first 10 rows(observation) of dataset
head(forestdata,10)
```

```
##    Elevation Aspect Slope Horizontal_Distance_To_Hydrology
## 1       2596     51     3                               258
## 2       2590     56     2                               212
## 3       2804    139     9                               268
## 4       2785    155    18                               242
## 5       2595     45     2                               153
## 6       2579    132     6                               300
## 7       2606     45     7                               270
## 8       2605     49     4                               234
## 9       2617     45     9                               240
## 10      2612     59    10                               247
##    Vertical_Distance_To_Hydrology Horizontal_Distance_To_Roadways Hillshade_9am
## 1                               0                             510           221
## 2                              -6                             390           220
## 3                              65                            3180           234
## 4                             118                            3090           238
## 5                              -1                             391           220
## 6                             -15                              67           230
## 7                               5                             633           222
## 8                               7                             573           222
```

```
## 9                            56                             666           223
## 10                           11                             636           228
##    Hillshade_Noon Hillshade_3pm Horizontal_Distance_To_Fire_Points
## 1             232           148                              6279
## 2             235           151                              6225
## 3             238           135                              6121
## 4             238           122                              6211
## 5             234           150                              6172
## 6             237           140                              6031
## 7             225           138                              6256
## 8             230           144                              6228
## 9             221           133                              6244
## 10            219           124                              6230
##    Wilderness_Area1 Wilderness_Area2 Wilderness_Area3 Wilderness_Area4
## 1                 1                0                0                0
## 2                 1                0                0                0
## 3                 1                0                0                0
## 4                 1                0                0                0
## 5                 1                0                0                0
## 6                 1                0                0                0
## 7                 1                0                0                0
## 8                 1                0                0                0
## 9                 1                0                0                0
## 10                1                0                0                0
##    Soil_Type1 Soil_Type2 Soil_Type3 Soil_Type4 Soil_Type5 Soil_Type6 Soil_Type7
## 1           0          0          0          0          0          0          0
## 2           0          0          0          0          0          0          0
## 3           0          0          0          0          0          0          0
## 4           0          0          0          0          0          0          0
## 5           0          0          0          0          0          0          0
## 6           0          0          0          0          0          0          0
## 7           0          0          0          0          0          0          0
## 8           0          0          0          0          0          0          0
## 9           0          0          0          0          0          0          0
## 10          0          0          0          0          0          0          0
##    Soil_Type8 Soil_Type9 Soil_Type10 Soil_Type11 Soil_Type12 Soil_Type13
## 1           0          0           0           0           0           0
## 2           0          0           0           0           0           0
## 3           0          0           0           0           1           0
## 4           0          0           0           0           0           0
## 5           0          0           0           0           0           0
## 6           0          0           0           0           0           0
## 7           0          0           0           0           0           0
## 8           0          0           0           0           0           0
## 9           0          0           0           0           0           0
## 10          0          0           0           0           0           0
##    Soil_Type14 Soil_Type15 Soil_Type16 Soil_Type17 Soil_Type18 Soil_Type19
## 1            0           0           0           0           0           0
## 2            0           0           0           0           0           0
## 3            0           0           0           0           0           0
## 4            0           0           0           0           0           0
## 5            0           0           0           0           0           0
## 6            0           0           0           0           0           0
## 7            0           0           0           0           0           0
```

```
## 8            0            0            0            0            0            0
## 9            0            0            0            0            0            0
## 10           0            0            0            0            0            0
##    Soil_Type20 Soil_Type21 Soil_Type22 Soil_Type23 Soil_Type24 Soil_Type25
## 1            0            0            0            0            0            0
## 2            0            0            0            0            0            0
## 3            0            0            0            0            0            0
## 4            0            0            0            0            0            0
## 5            0            0            0            0            0            0
## 6            0            0            0            0            0            0
## 7            0            0            0            0            0            0
## 8            0            0            0            0            0            0
## 9            0            0            0            0            0            0
## 10           0            0            0            0            0            0
##    Soil_Type26 Soil_Type27 Soil_Type28 Soil_Type29 Soil_Type30 Soil_Type31
## 1            0            0            0            1            0            0
## 2            0            0            0            1            0            0
## 3            0            0            0            0            0            0
## 4            0            0            0            0            1            0
## 5            0            0            0            1            0            0
## 6            0            0            0            1            0            0
## 7            0            0            0            1            0            0
## 8            0            0            0            1            0            0
## 9            0            0            0            1            0            0
## 10           0            0            0            1            0            0
##    Soil_Type32 Soil_Type33 Soil_Type34 Soil_Type35 Soil_Type36 Soil_Type37
## 1            0            0            0            0            0            0
## 2            0            0            0            0            0            0
## 3            0            0            0            0            0            0
## 4            0            0            0            0            0            0
## 5            0            0            0            0            0            0
## 6            0            0            0            0            0            0
## 7            0            0            0            0            0            0
## 8            0            0            0            0            0            0
## 9            0            0            0            0            0            0
## 10           0            0            0            0            0            0
##    Soil_Type38 Soil_Type39 Soil_Type40 Cover_Type
## 1            0            0            0          5
## 2            0            0            0          5
## 3            0            0            0          2
## 4            0            0            0          2
## 5            0            0            0          5
## 6            0            0            0          2
## 7            0            0            0          5
## 8            0            0            0          5
## 9            0            0            0          5
## 10           0            0            0          5
```

Next, we note down the dimensions of our dataframe using dim().

```
dim(forestdata)
```

```
## [1] 581012     55
```

Above values tell us that currently there are **581012** rows and **55** columns in our dataframe.

Moving ahead, We check the structure of our dataframe to find what data type each column is -

```
str(forestdata)
```

```
## 'data.frame':    581012 obs. of  55 variables:
##  $ Elevation                         : int  2596 2590 2804 2785 2595 2579 2606 2605 2617 2612 ...
##  $ Aspect                            : int  51 56 139 155 45 132 45 49 45 59 ...
##  $ Slope                             : int  3 2 9 18 2 6 7 4 9 10 ...
##  $ Horizontal_Distance_To_Hydrology  : int  258 212 268 242 153 300 270 234 240 247 ...
##  $ Vertical_Distance_To_Hydrology    : int  0 -6 65 118 -1 -15 5 7 56 11 ...
##  $ Horizontal_Distance_To_Roadways   : int  510 390 3180 3090 391 67 633 573 666 636 ...
##  $ Hillshade_9am                     : int  221 220 234 238 220 230 222 222 223 228 ...
##  $ Hillshade_Noon                    : int  232 235 238 238 234 237 225 230 221 219 ...
##  $ Hillshade_3pm                     : int  148 151 135 122 150 140 138 144 133 124 ...
##  $ Horizontal_Distance_To_Fire_Points: int  6279 6225 6121 6211 6172 6031 6256 6228 6244 6230 ...
##  $ Wilderness_Area1                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Wilderness_Area2                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Wilderness_Area3                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Wilderness_Area4                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type1                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type2                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type3                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type4                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type5                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type6                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type7                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type8                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type9                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type10                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type11                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type12                       : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Soil_Type13                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type14                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type15                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type16                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type17                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type18                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type19                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type20                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type21                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type22                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type23                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type24                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type25                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type26                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type27                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type28                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type29                       : int  1 1 0 0 1 1 1 1 1 1 ...
##  $ Soil_Type30                       : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Soil_Type31                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type32                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type33                       : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Soil_Type34                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type35                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type36                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type37                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type38                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type39                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Soil_Type40                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cover_Type                        : int  5 5 2 2 5 2 5 5 5 5 ...
```

As seen above, every column is a **numeric** value. There is not a single character or factor datatype present in this dataset.

Summary function helps us identify mean, median and inter-quantile range values along with null vlaues present by each column.

```
summary(forestdata)
```

```
##    Elevation        Aspect          Slope      Horizontal_Distance_To_Hydrology
##  Min.   :1859   Min.   :  0.0   Min.   : 0.0   Min.   :   0.0
##  1st Qu.:2809   1st Qu.: 58.0   1st Qu.: 9.0   1st Qu.: 108.0
##  Median :2996   Median :127.0   Median :13.0   Median : 218.0
##  Mean   :2959   Mean   :155.7   Mean   :14.1   Mean   : 269.4
##  3rd Qu.:3163   3rd Qu.:260.0   3rd Qu.:18.0   3rd Qu.: 384.0
##  Max.   :3858   Max.   :360.0   Max.   :66.0   Max.   :1397.0
##  Vertical_Distance_To_Hydrology Horizontal_Distance_To_Roadways Hillshade_9am
##  Min.   :-173.00                Min.   :   0                    Min.   :  0.0
##  1st Qu.:   7.00                1st Qu.:1106                    1st Qu.:198.0
##  Median :  30.00                Median :1997                    Median :218.0
##  Mean   :  46.42                Mean   :2350                    Mean   :212.1
##  3rd Qu.:  69.00                3rd Qu.:3328                    3rd Qu.:231.0
##  Max.   : 601.00                Max.   :7117                    Max.   :254.0
##  Hillshade_Noon  Hillshade_3pm   Horizontal_Distance_To_Fire_Points
##  Min.   :  0.0   Min.   :  0.0   Min.   :   0
##  1st Qu.:213.0   1st Qu.:119.0   1st Qu.:1024
##  Median :226.0   Median :143.0   Median :1710
##  Mean   :223.3   Mean   :142.5   Mean   :1980
##  3rd Qu.:237.0   3rd Qu.:168.0   3rd Qu.:2550
##  Max.   :254.0   Max.   :254.0   Max.   :7173
##  Wilderness_Area1 Wilderness_Area2  Wilderness_Area3 Wilderness_Area4
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.4489   Mean   :0.05143   Mean   :0.4361   Mean   :0.06363
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##    Soil_Type1         Soil_Type2        Soil_Type3         Soil_Type4
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.000000   Median :0.00000
##  Mean   :0.005217   Mean   :0.01295   Mean   :0.008301   Mean   :0.02134
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.000000   Max.   :1.00000
##    Soil_Type5         Soil_Type6        Soil_Type7          Soil_Type8
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.0000000   Min.   :0.0000000
```

```
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.0000000   1st Qu.:0.0000000
##  Median :0.000000   Median :0.00000   Median :0.0000000   Median :0.0000000
##  Mean   :0.002749   Mean   :0.01132   Mean   :0.0001807   Mean   :0.0003081
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.0000000   3rd Qu.:0.0000000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.0000000   Max.   :1.0000000
##    Soil_Type9        Soil_Type10       Soil_Type11       Soil_Type12
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.001974   Mean   :0.05617   Mean   :0.02136   Mean   :0.05158
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##   Soil_Type13    Soil_Type14        Soil_Type15       Soil_Type16
##  Min.   :0.00   Min.   :0.000000   Min.   :0.0e+00   Min.   :0.000000
##  1st Qu.:0.00   1st Qu.:0.000000   1st Qu.:0.0e+00   1st Qu.:0.000000
##  Median :0.00   Median :0.000000   Median :0.0e+00   Median :0.000000
##  Mean   :0.03   Mean   :0.001031   Mean   :5.2e-06   Mean   :0.004897
##  3rd Qu.:0.00   3rd Qu.:0.000000   3rd Qu.:0.0e+00   3rd Qu.:0.000000
##  Max.   :1.00   Max.   :1.000000   Max.   :1.0e+00   Max.   :1.000000
##   Soil_Type17       Soil_Type18        Soil_Type19        Soil_Type20
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :0.00000   Median :0.000000   Median :0.000000   Median :0.00000
##  Mean   :0.00589   Mean   :0.003268   Mean   :0.006921   Mean   :0.01594
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.000000   Max.   :1.00000
##   Soil_Type21        Soil_Type22       Soil_Type23      Soil_Type24
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.001442   Mean   :0.05744   Mean   :0.0994   Mean   :0.03662
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##   Soil_Type25         Soil_Type26        Soil_Type27        Soil_Type28
##  Min.   :0.0000000   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.000000
##  Median :0.0000000   Median :0.000000   Median :0.000000   Median :0.000000
##  Mean   :0.0008158   Mean   :0.004456   Mean   :0.001869   Mean   :0.001628
##  3rd Qu.:0.0000000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.000000
##  Max.   :1.0000000   Max.   :1.000000   Max.   :1.000000   Max.   :1.000000
##   Soil_Type29       Soil_Type30       Soil_Type31       Soil_Type32
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.1984   Mean   :0.05193   Mean   :0.04417   Mean   :0.09039
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##   Soil_Type33       Soil_Type34        Soil_Type35        Soil_Type36
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.0000000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.0000000
##  Median :0.00000   Median :0.000000   Median :0.000000   Median :0.0000000
##  Mean   :0.07772   Mean   :0.002773   Mean   :0.003255   Mean   :0.0002048
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.0000000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.000000   Max.   :1.0000000
```

```
##   Soil_Type37         Soil_Type38      Soil_Type39       Soil_Type40
##  Min.   :0.0000000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.0005129   Mean   :0.0268   Mean   :0.02376   Mean   :0.01506
##  3rd Qu.:0.0000000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.0000000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##    Cover_Type
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :2.051
##  3rd Qu.:2.000
##  Max.   :7.000
```

Data Cleaning & Transformation:

After understanding the dataset, we can start working on cleaning if required.

Summary function helps us identify mean, median and inter-quantile range values along with null vlaues present by each column.

```
summary(forestdata)
```

```
##    Elevation        Aspect          Slope      Horizontal_Distance_To_Hydrology
##  Min.   :1859   Min.   :  0.0   Min.   : 0.0   Min.   :   0.0
##  1st Qu.:2809   1st Qu.: 58.0   1st Qu.: 9.0   1st Qu.: 108.0
##  Median :2996   Median :127.0   Median :13.0   Median : 218.0
##  Mean   :2959   Mean   :155.7   Mean   :14.1   Mean   : 269.4
##  3rd Qu.:3163   3rd Qu.:260.0   3rd Qu.:18.0   3rd Qu.: 384.0
##  Max.   :3858   Max.   :360.0   Max.   :66.0   Max.   :1397.0
##  Vertical_Distance_To_Hydrology Horizontal_Distance_To_Roadways Hillshade_9am
##  Min.   :-173.00                Min.   :   0                    Min.   :  0.0
##  1st Qu.:   7.00                1st Qu.:1106                    1st Qu.:198.0
##  Median :  30.00                Median :1997                    Median :218.0
##  Mean   :  46.42                Mean   :2350                    Mean   :212.1
##  3rd Qu.:  69.00                3rd Qu.:3328                    3rd Qu.:231.0
##  Max.   : 601.00                Max.   :7117                    Max.   :254.0
##  Hillshade_Noon  Hillshade_3pm   Horizontal_Distance_To_Fire_Points
##  Min.   :  0.0   Min.   :  0.0   Min.   :   0
##  1st Qu.:213.0   1st Qu.:119.0   1st Qu.:1024
##  Median :226.0   Median :143.0   Median :1710
##  Mean   :223.3   Mean   :142.5   Mean   :1980
##  3rd Qu.:237.0   3rd Qu.:168.0   3rd Qu.:2550
##  Max.   :254.0   Max.   :254.0   Max.   :7173
##  Wilderness_Area1 Wilderness_Area2  Wilderness_Area3 Wilderness_Area4
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.4489   Mean   :0.05143   Mean   :0.4361   Mean   :0.06363
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##    Soil_Type1         Soil_Type2        Soil_Type3         Soil_Type4
```

```
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.000000   Median :0.00000
##  Mean   :0.005217   Mean   :0.01295   Mean   :0.008301   Mean   :0.02134
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.000000   Max.   :1.00000
##    Soil_Type5         Soil_Type6         Soil_Type7          Soil_Type8
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.0000000   Min.   :0.0000000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.0000000   1st Qu.:0.0000000
##  Median :0.000000   Median :0.00000   Median :0.0000000   Median :0.0000000
##  Mean   :0.002749   Mean   :0.01132   Mean   :0.0001807   Mean   :0.0003081
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.0000000   3rd Qu.:0.0000000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.0000000   Max.   :1.0000000
##    Soil_Type9        Soil_Type10       Soil_Type11       Soil_Type12
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.001974   Mean   :0.05617   Mean   :0.02136   Mean   :0.05158
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##   Soil_Type13    Soil_Type14        Soil_Type15       Soil_Type16
##  Min.   :0.00   Min.   :0.000000   Min.   :0.0e+00   Min.   :0.000000
##  1st Qu.:0.00   1st Qu.:0.000000   1st Qu.:0.0e+00   1st Qu.:0.000000
##  Median :0.00   Median :0.000000   Median :0.0e+00   Median :0.000000
##  Mean   :0.03   Mean   :0.001031   Mean   :5.2e-06   Mean   :0.004897
##  3rd Qu.:0.00   3rd Qu.:0.000000   3rd Qu.:0.0e+00   3rd Qu.:0.000000
##  Max.   :1.00   Max.   :1.000000   Max.   :1.0e+00   Max.   :1.000000
##   Soil_Type17       Soil_Type18        Soil_Type19        Soil_Type20
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :0.00000   Median :0.000000   Median :0.000000   Median :0.00000
##  Mean   :0.00589   Mean   :0.003268   Mean   :0.006921   Mean   :0.01594
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.000000   Max.   :1.00000
##   Soil_Type21        Soil_Type22       Soil_Type23      Soil_Type24
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.001442   Mean   :0.05744   Mean   :0.0994   Mean   :0.03662
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##   Soil_Type25         Soil_Type26        Soil_Type27        Soil_Type28
##  Min.   :0.0000000   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.000000
##  Median :0.0000000   Median :0.000000   Median :0.000000   Median :0.000000
##  Mean   :0.0008158   Mean   :0.004456   Mean   :0.001869   Mean   :0.001628
##  3rd Qu.:0.0000000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.000000
##  Max.   :1.0000000   Max.   :1.000000   Max.   :1.000000   Max.   :1.000000
##   Soil_Type29       Soil_Type30        Soil_Type31       Soil_Type32
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.1984   Mean   :0.05193   Mean   :0.04417   Mean   :0.09039
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
```

```
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##   Soil_Type33       Soil_Type34        Soil_Type35        Soil_Type36
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.0000000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.0000000
##  Median :0.00000   Median :0.000000   Median :0.000000   Median :0.0000000
##  Mean   :0.07772   Mean   :0.002773   Mean   :0.003255   Mean   :0.0002048
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.0000000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.000000   Max.   :1.0000000
##   Soil_Type37         Soil_Type38      Soil_Type39       Soil_Type40
##  Min.   :0.0000000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.0005129   Mean   :0.0268   Mean   :0.02376   Mean   :0.01506
##  3rd Qu.:0.0000000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.0000000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##    Cover_Type
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :2.051
##  3rd Qu.:2.000
##  Max.   :7.000
```

Since none of the column returned NA's as output from summary(), there are no missing values in this dataset.

Next, we work on transforming dataset(Feature Engineering). We will focus on following columns -

1. Hillshade_9am,Hillshade_Noon and Hillshade_3pm.
2. Horizontal_Distance_To_Hydrology and Vertical_Distance_To_Hydrology.
3. Wilderness_Area1, Wilderness_Area2, Wilderness_Area3 and Wilderness_Area4
4. Soil_Type1, Soil_Type2, Soil_Type3 ...Soil_Type40.

For 1st part we will be combining 3 variables of hillshade by calculating mean:

```
# Calculating mean of hillshade
forestdata$Hillshade_mean = (forestdata$Hillshade_9am + forestdata$Hillshade_3pm
                             + forestdata$Hillshade_Noon) / 3
```

For 2nd part we will be combining 2 variables of hydrology using Pythagorus theorem. Following diagram helps us see how two distances form a hypotenuse and Pythagorus theorem is the best way to combine these 2 columns.

```
# Calculating Euclidean distance(hypotenuse) using pythagorus theorem
forestdata$Distance_To_Hydrology = (forestdata$Horizontal_Distance_To_Hydrology^2 + forestdata$Vertical
```

Wilderness columns range from Wilderness_Area1 to Wilderness_Area4 with values 0 and 1 where 1 indicates existence of tree in that wilderness area while 0 indicates absence.

```
# Create single Wilderness_Area column
forestdata$Wilderness_Area = 0
for (i in 11:14) {
  forestdata$Wilderness_Area[forestdata[,i] == 1] = i-10
}
```

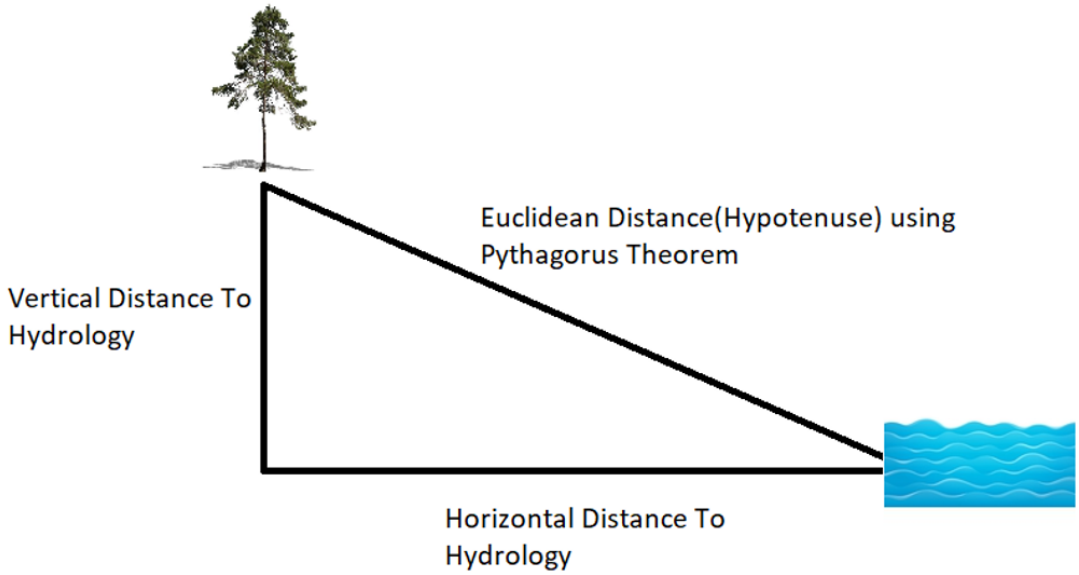


Figure 1: Euclidean Distance between tree and hydrology

Wilderness columns range from Soil_Type1 to Soil_Type40 with values 0 and 1 where 1 indicates existence of that soil type below our particular tree(observation).

```
# Create single Soil_Type column
forestdata$Soil_Type = 0
for (i in 15:54) {
  forestdata$Soil_Type[forestdata[,i] == 1] = i-14
}
```

Finally, we have all the required columns ready for exploratory data analysis and model fitting. All we have to do is subset those from original dataframe

Column Reduction:

```
# Selecting subset from all columns to form our new dataframe
new_forestdata <- forestdata[, c("Elevation","Aspect","Slope","Distance_To_Hydrology",
                            "Horizontal_Distance_To_Roadways",
                            "Horizontal_Distance_To_Fire_Points",
                            "Wilderness_Area",
                            "Hillshade_mean","Soil_Type","Cover_Type")]
```

We will label the wilderness area and cover type for better understanding of our exploratory plots.

```
new_forestdata = apply_labels(new_forestdata,
                              Wilderness_Area = c("Rawah" = 1,
                                                  "Neota" = 2,
                                                  "Comanche Peak" = 3,
                                                  "Cache la Poudre" = 4),
                              Cover_Type = c("Spruce/Fir" = 1,
```

```
                                                "Lodgepole Pine" = 2,
                                                "Ponderosa Pine" = 3,
                                                "Cottonwood/Willow" = 4,
                                                "Aspen" = 5,
                                                "Douglas Fir" = 6,
                                                "Krummholz" = 7)
)
```

While fitting our model, we encounter a difficulty due to technical restrictions of laptop. Tuning a model comes at a cost of high performance our laptops couldn't handle. As a solution to this, we decided to sample our dataset and run models on less number of observations. 1000 samples from each cover type were picked to ensure minimum bias towards particular cover type.

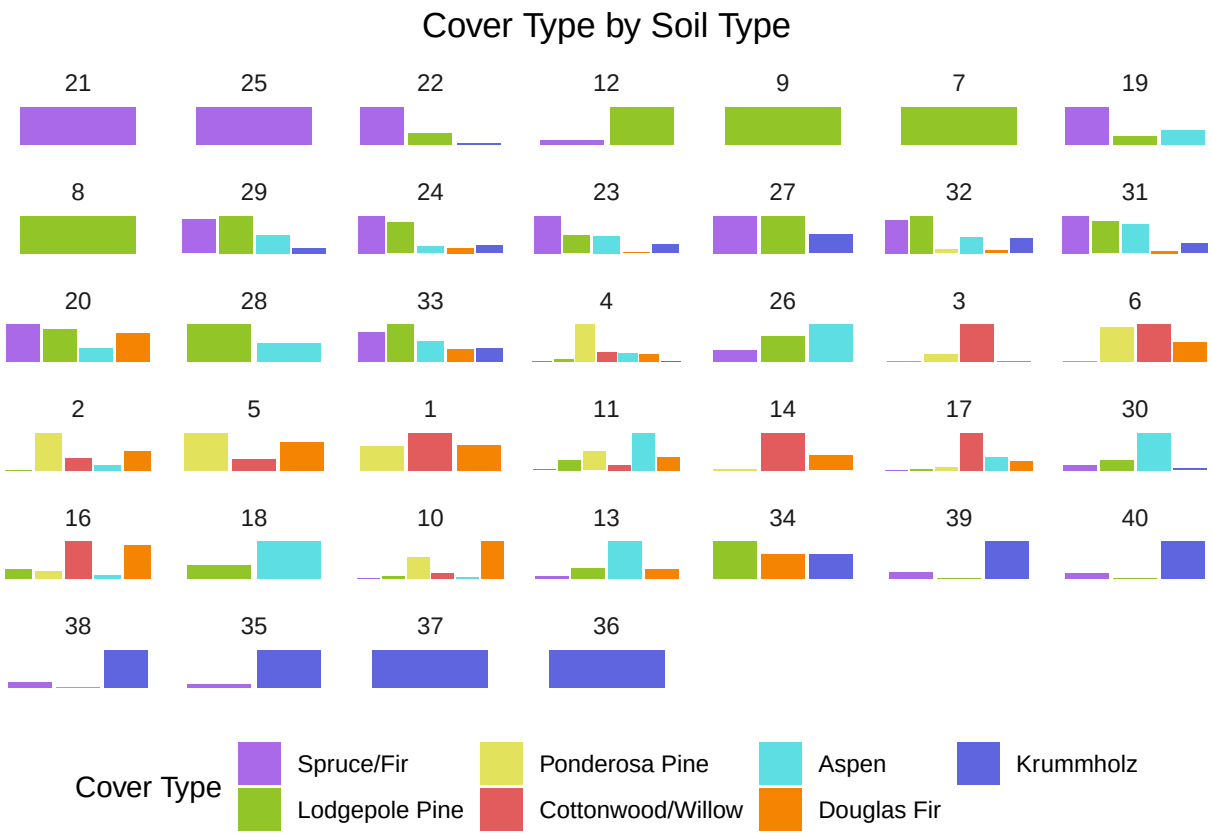
```
sampled_forestdata <- (new_forestdata %>%
                         group_by(Cover_Type) %>%
                         sample_n(size = 1000))

dim(sampled_forestdata)
```

```
## [1] 7000   10
```

Exploratory Data Analysis:

Cover x Soil Type

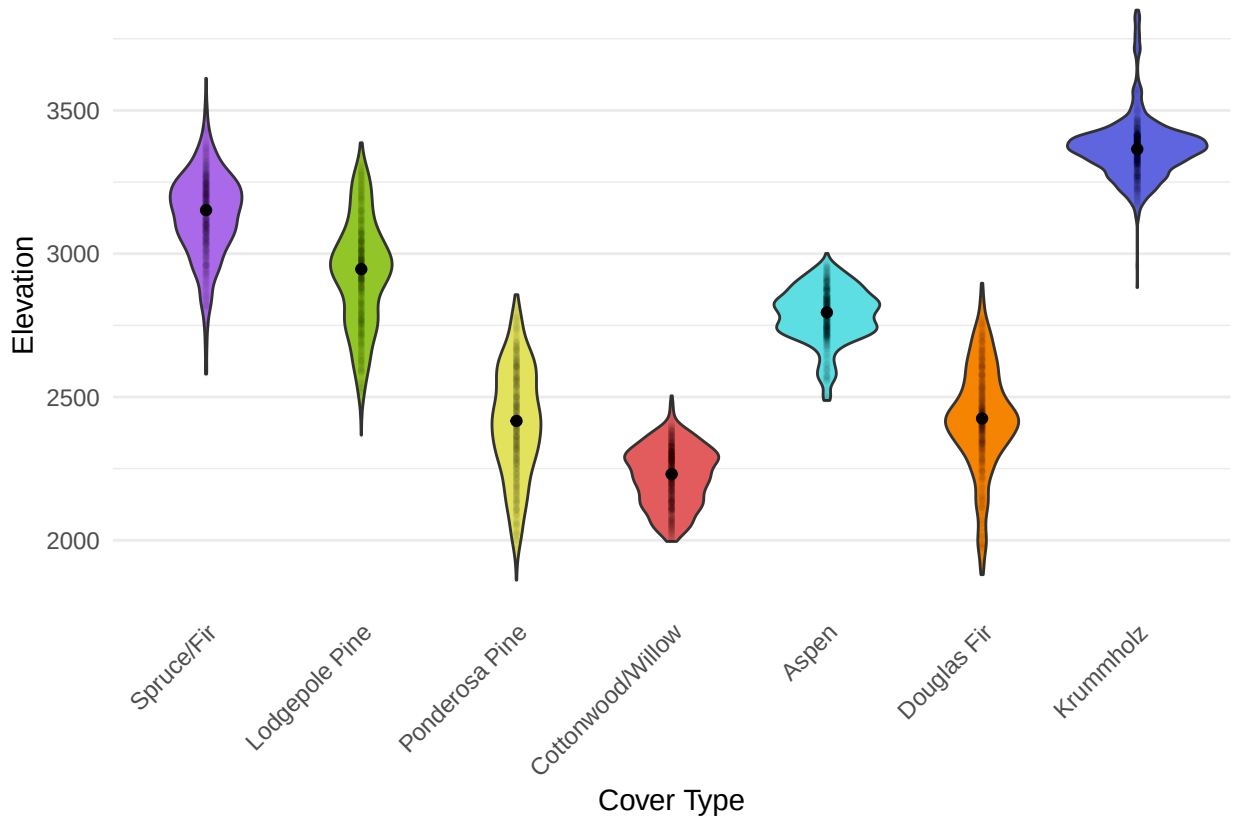
```
palette <- c('#A969E9', '#92C628', '#E2E25C', '#E25C5E', '#5CDEE2', '#F58403', '#5F65DE')


ggplot(sampled_forestdata, aes(x = as.factor(Cover_Type), fill = as.factor(Cover_Type))) +
  geom_bar() +
  facet_wrap(~reorder(as.factor(Soil_Type), sort.int(as.integer(Soil_Type), decreasing = FALSE)), scales
  labs(fill = 'Cover Type', title = 'Cover Type by Soil Type') +
  scale_fill_manual(values = palette) +
  theme_minimal() +
  theme(legend.position = 'bottom',
        plot.title = element_text(hjust = 0.5),
        axis.title = element_blank(),
        axis.text = element_blank(),
        panel.grid = element_blank())
```



Observations by Cover and Soil Type: 1. Krummholz grow predominantly in soil types 35-40 specifically 36 and 37 has only Krumholz growing. There are significant amount of K growing in soil type 21,34,32 2. Lodgepole Pine are found in majority of soil types. Soil type 25 is seen to have only Lodgepole Pines growing while 9,12,20,28,34 has visible amount of Lodgepoles. We also noticed that Krummholz and Cottonwood/Willow covers share only one soil type: 4 & 10.

Cover x Elevation

```
ggplot(sampled_forestdata, aes(x = as.factor(Cover_Type), y = Elevation)) +
  geom_violin(aes(fill = as.factor(Cover_Type))) +
  geom_point(alpha = 0.01, size = 0.5) +
  stat_summary(fun = 'median', geom = 'point') +
  labs(x = 'Cover Type') +
  scale_fill_manual(values = palette) +
  theme_minimal() +
  theme(legend.position = 'none',
        axis.text.x = element_text(angle = 45,
                                   hjust = 1),
        panel.grid.major.x = element_blank())
```



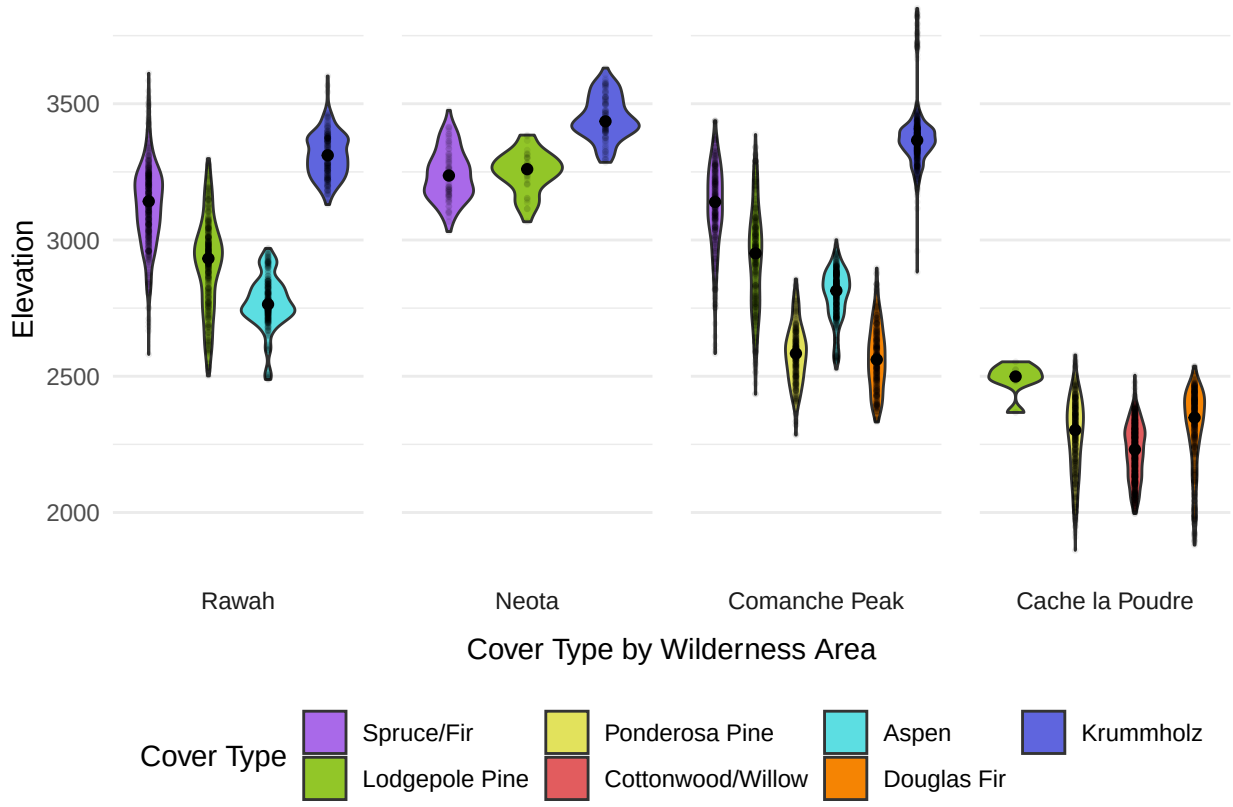
Observations by Cover Type and Elevations:

1. Krummholz cover may be found at extremely high elevations when other types of cover are uncommon. Not only do the Krummholz and Cottonwood/Willow covers lack many of the same soil types, but they are also separated by about 500 meters in elevations.
2. Cottonwood/Willow trees are found in low elevation areas and as Cache la Poudre is one of the low elevation areas and this tree can be found only here.
3. Commanche peak is one of the average elevation areas and most of the cover types except Cottonwood/Willow can be found here.

Cover x Elevation x Wilderness Area

```
ggplot(sampled_forestdata, aes(x = as.factor(Cover_Type), y = Elevation)) +
  geom_violin(aes(fill = as.factor(Cover_Type))) +
  geom_point(alpha = 0.08, size = 0.5) +
  stat_summary(aes(group = as.factor(Cover_Type)),
               fun = 'median',
               geom = 'point',
               show.legend = FALSE) +
  labs(x = 'Cover Type by Wilderness Area') +
  scale_fill_manual(name = 'Cover Type',
                    values = palette) +
  facet_grid(. ~ as.factor(Wilderness_Area), scales = 'free_x', switch = 'x') +
  theme_minimal() +
  theme(legend.position = 'bottom',
```

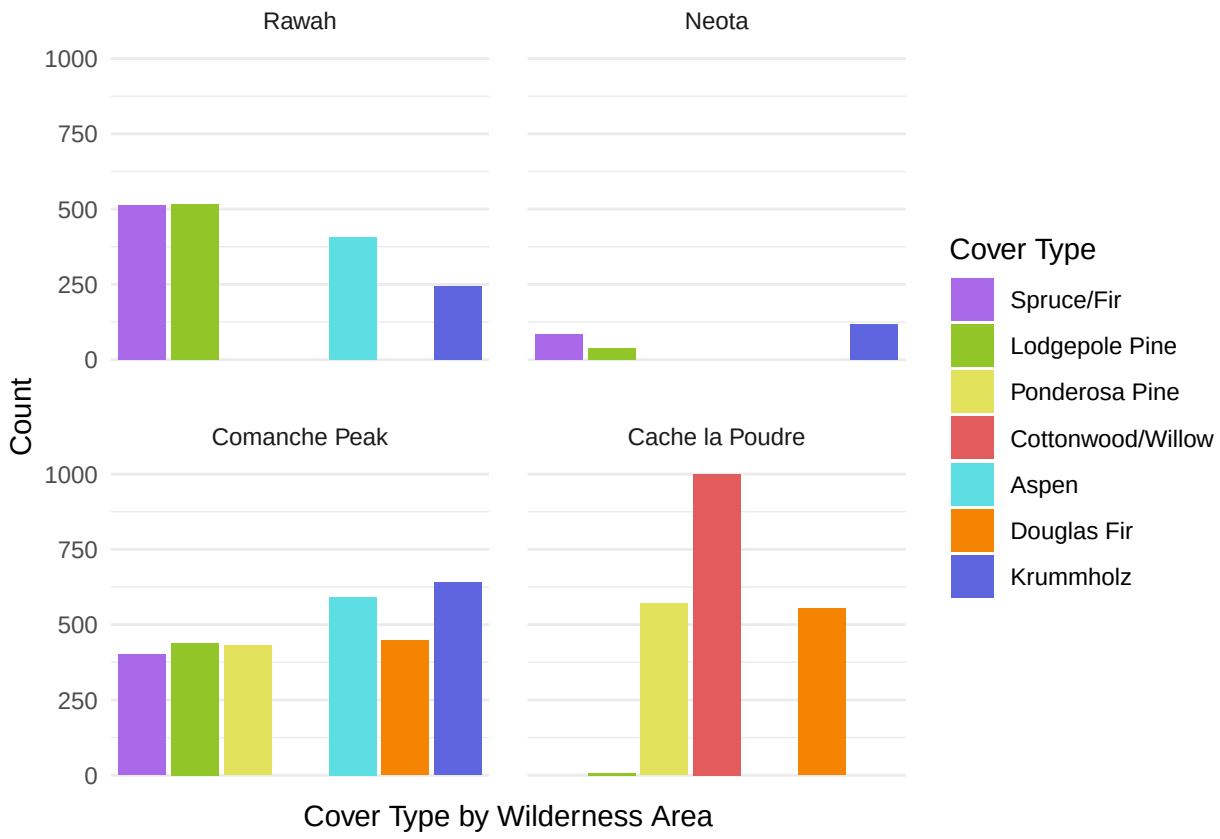
```
        axis.text.x = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.spacing = unit(1, 'lines'))
```



We observe that in the wilderness area named Neota, we can see a minimum growth of trees and only three types of trees which are Krummholz, Spruce/Fir, Lodgepole pine with a maximum count of 250 for any given cover type.

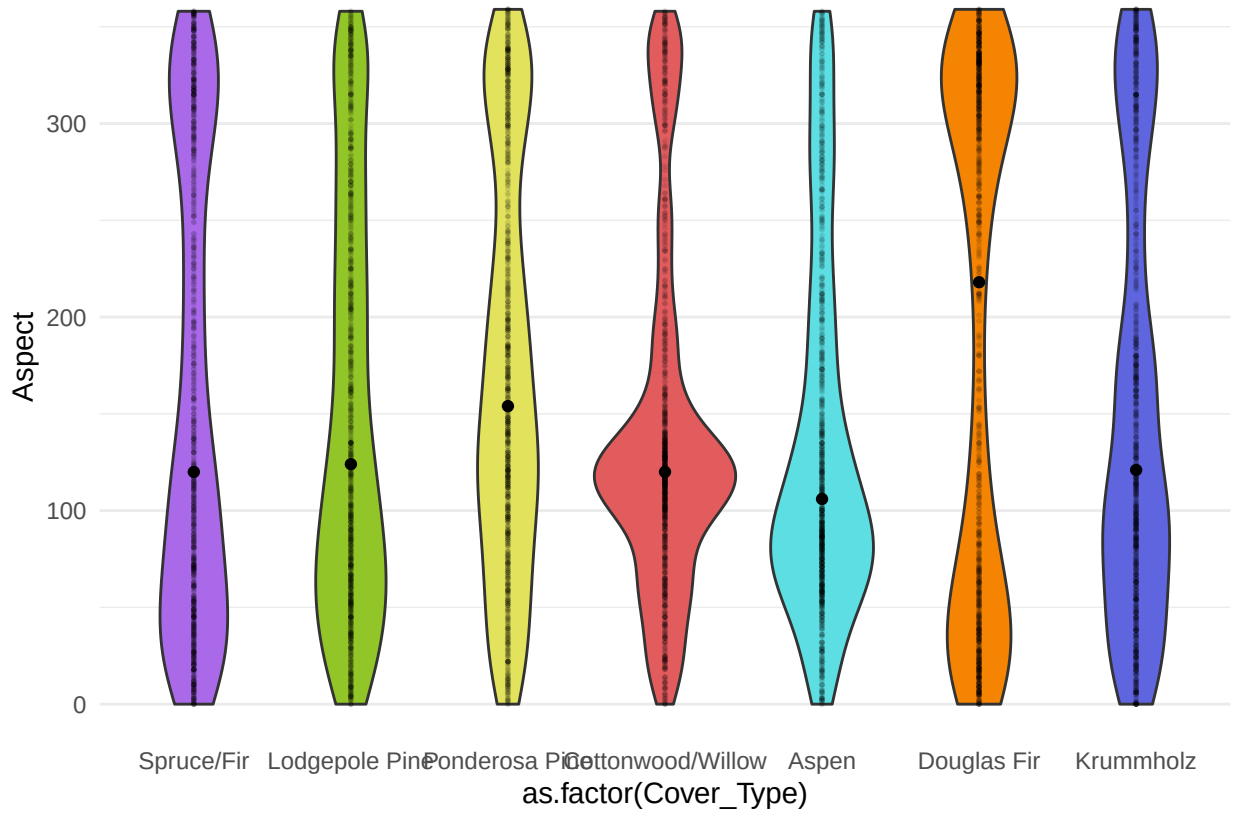
Cover x Wilderness area x count

```
ggplot(sampled_forestdata, aes(x = as.factor(Cover_Type), fill = as.factor(Cover_Type))) +
  geom_bar() +
  facet_wrap(~as.factor(Wilderness_Area)) +
  labs(x = 'Cover Type by Wilderness Area', y = 'Count') +
  scale_fill_manual(name = 'Cover Type',
                    values = palette) +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.spacing = unit(1, 'line'))
```



cover x Aspect

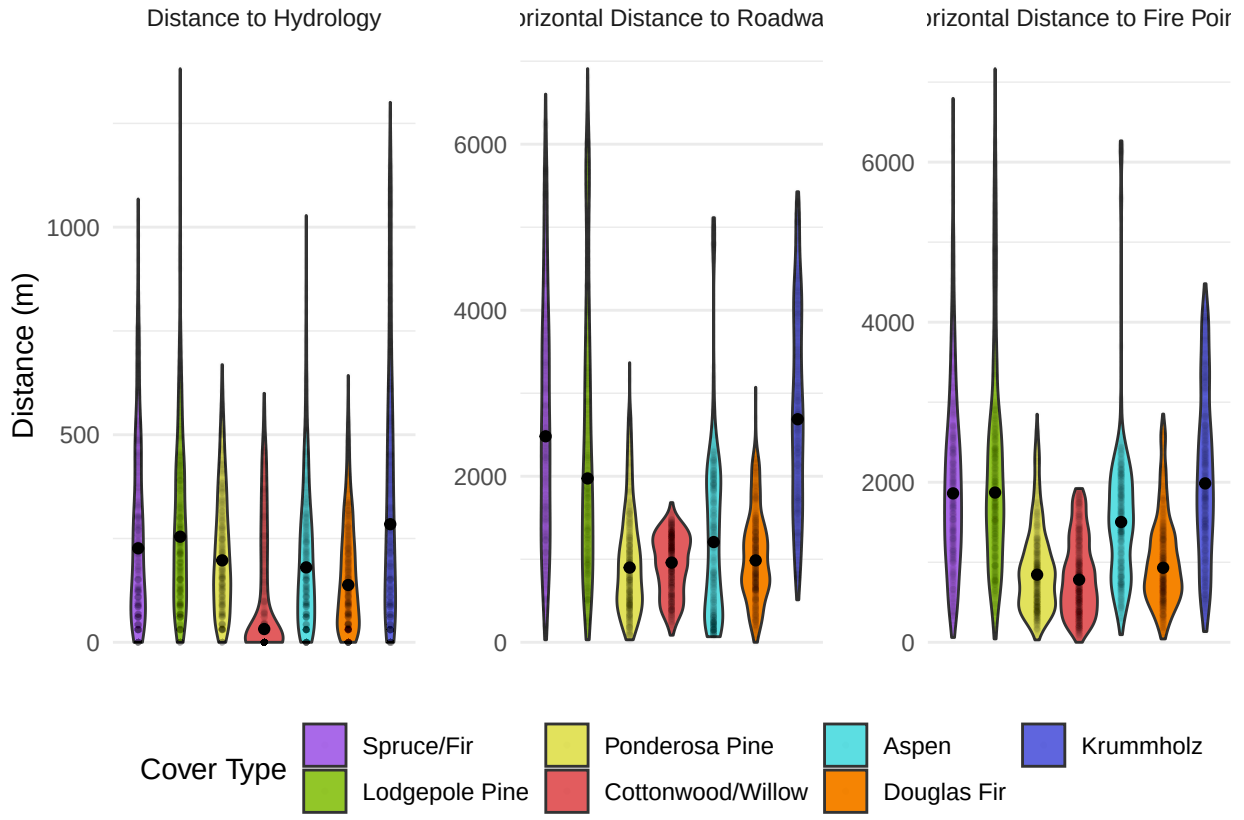
```
ggplot(sampled_forestdata, aes(x = as.factor(Cover_Type), y = Aspect)) +
  geom_violin(aes(fill = as.factor(Cover_Type))) +
  stat_summary(fun = 'median', geom = 'point') +
  geom_point(alpha = 0.1, size = 0.3) +
  scale_fill_manual(values = palette) +
  theme_minimal() +
  theme(legend.position = 'none',
        panel.grid.major.x = element_blank())
```



cover x Distance(Hydrology, roadway, firepoints)

```
sampled_forestdata %>%
  gather(Measure, Distance,
         Distance_To_Hydrology:Horizontal_Distance_To_Fire_Points) %>%
  mutate(Measure = factor(Measure,
                          levels = c('Distance_To_Hydrology',
                                     'Horizontal_Distance_To_Roadways',
                                     'Horizontal_Distance_To_Fire_Points'),
                          labels = c('Distance to Hydrology',
                                     'Horizontal Distance to Roadways',
                                     'Horizontal Distance to Fire Points'))) %>%


  ggplot(aes(x = as.factor(Cover_Type), y = Distance, fill = as.factor(Cover_Type))) +
  geom_violin() +
  geom_point(alpha = 0.01, size = 0.5) +
  stat_summary(fun = 'median', geom = 'point',
               show.legend = FALSE) +
  facet_wrap(~Measure, scales = 'free_y') +
  labs(x = NULL, y = 'Distance (m)') +
  scale_fill_manual(name = 'Cover Type',
                    values = palette) +
  theme_minimal() +
```

```
theme(legend.position = 'bottom',
      axis.text.x = element_blank(),
      panel.spacing = unit(1, 'line'),
      panel.grid.major.x = element_blank())
```



Observations: 1. Distance to hydrology + Cottonwood/Willow is found closest to the water body and the growth Krummholz is not dependent on the distance from hydrology as they can be seen growing in areas farther away from the water bodies hence proving our previous observation which stated that Krummholz is seen in higher elevation areas.

2. Distance to roadways

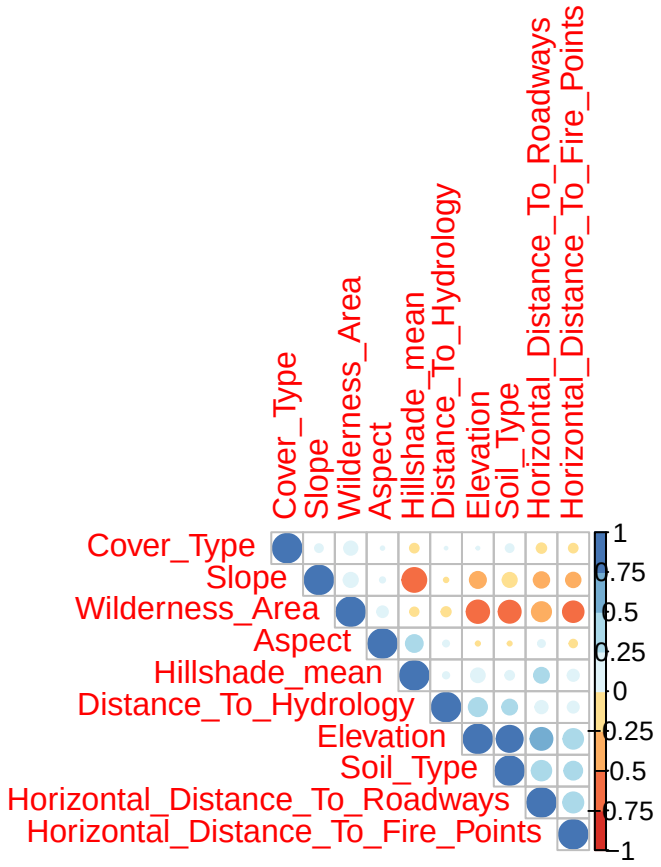
- The distance to roadways indicates that how well a cover type deals with human interference and hence we can see that Cottonwood/Willow grows in areas closer to roadways and Krummholz are found away from the roadways. While the other tree types don't seem to be affected by this factor.

3. Horizontal distance to fire points:

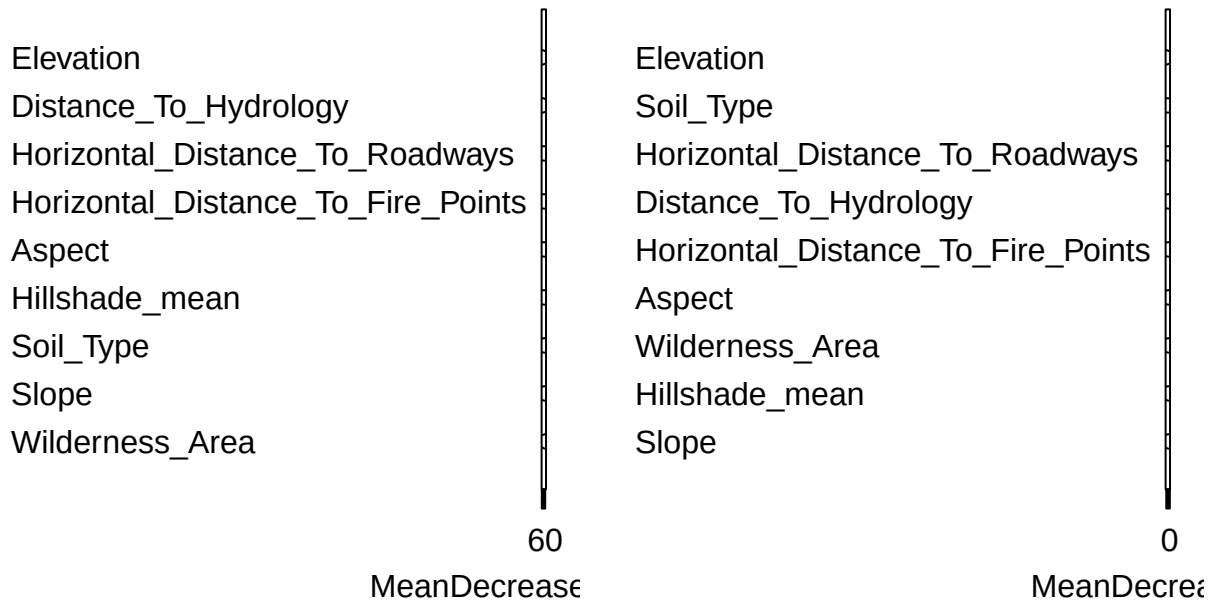
- This factor tells us about the ability of cover type to regrow after a forest fire. Looking at the graph we can see that Ponderosa Pine, Cottonwood/Willow and Douglas Fir seem to grow in areas close to fire point while this factor doesn't seem to have a great effect on the other tree types.

correlation among variables

```
M <-cor(sampled_forestdata)
corrplot(M, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```




```
set.seed(1808)

vars <- randomForest::randomForest(as.factor(Cover_Type) ~ .,
                                   data = sampled_forestdata,
                                   importance = TRUE)

varImpPlot(vars)
```



From both the correlation graphs we can see that Wilderness area, Slope and Hill shade correlate directly with Cover Type but as Wilderness area along with distance to Hydrology, distance to Fire points and elevation act as strong factors, we chose all the factors while fitting the models.

Model Fitting:

Let's start fitting our models now. But before that, we shall split outr dataset into 70-30 split for Training and Test dataset.

```
# Setting seed to reproduce same dataset
set.seed(1)

# Generating training dataset
training_forestdata = sampled_forestdata[ sample(nrow(sampled_forestdata), round(0.7*nrow(sampled_fores

# Generating test dataset
test_forestdata = sampled_forestdata[-(sample(nrow(sampled_forestdata), round(0.7*nrow(sampled_forestda
```

Bagging:

We will try our classification using Bagging first.

Bagging is often called as bootstrap aggregation. It is type of ensemble learning method and reduces variance in the dataset.

In this method, several random samples are chosen with replacement and then the model is trained using weak learners. Highest majority class of all the models gives us accurate prediction. We must note that since samples are chosen with replacement, sample maybe repeated multiple times in sample.

Two major advantages of bagging are

1. Variance Reduction - Bagging reduces variance in dataset and hence can be useful in cases of high dimensional data with missing values. Missing values in high dimensional data can lead to overfitting.
2. Bagging is easy to implement as it uses weak learners and associated math is relatively complex compared to other models.

Let's move to actually fitting the model -

```
set.seed(1)
bagging.forestdata <- randomForest(Cover_Type ~ ., data = training_forestdata, mtry = 9, importance = TR
bagging.forestdata
```

```
##
## Call:
##  randomForest(formula = Cover_Type ~ ., data = training_forestdata,      mtry = 9, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##           Mean of squared residuals: 1.327155
##                     % Var explained: 67.28
```

Evaluating performance on test data set & plotting confusion matrix:

```
yhat.bag <- round(predict(bagging.forestdata, newdata = test_forestdata))
table(yhat.bag, test_forestdata$Cover_Type)
```

```
##
## yhat.bag Spruce/Fir Lodgepole Pine Ponderosa Pine Cottonwood/Willow Aspen
##        1        198              5              0                 0     0
##        2         75            227              0                 0     2
##        3         13             58            170                 0     3
##        4         11             10             87               286    43
##        5          2             11             12                 0   269
##        6          2              2              1                 0     0
##        7          0              0              0                 0     0
##
## yhat.bag Douglas Fir Krummholz
##        1           0         1
##        2           0         4
##        3           4         4
##        4          18         2
##        5         115        19
##        6         173        42
##        7           0       231
```

Accuracy:

```
bagging_accuracy = mean(yhat.bag == test_forestdata$Cover_Type)
bagging_accuracy
```

```
## [1] 0.74
```

Boosting:

Boosting is an extension of bagging which utilizes weak learners and strong learners reducing training errors. A random sample is selected and the model is fitted on it after which weak learners are dropped or combined into stronger learners. This compensates weaker learners from Bagging where they are trained parallely in contrast to sequential in boosting.

The biggest advantage of Boosting algorithm is the computational efficiency due to selective features(strong learners) which reduces dimensions. Less dimensions means increase in computational speed.

This in turn is also the biggest disadvantage as sequential feature selection means decreased flexibility and scalability.

We will now fit gradient boosting model for our classsification problem -

```
training_forestdata$Cover_Type <- as.factor(training_forestdata$Cover_Type)

training_forestdata$Cover_Type <- as.numeric(training_forestdata$Cover_Type)

test_forestdata$Cover_Type <- as.factor(test_forestdata$Cover_Type)

test_forestdata$Cover_Type <- as.numeric(test_forestdata$Cover_Type)

test_forestdata$Cover_Type <- factor(test_forestdata$Cover_Type, levels=c(1:7), labels = c(1:7))

# Fitting the model
boosting = gbm(Cover_Type ~ ., data = training_forestdata,
                          distribution = "multinomial",
                          n.trees = 1000,
                          interaction.depth = 3)
```

```
# Setting seed for reproducibility
set.seed(999)

# Making predictions
boost_pred = predict(boosting, test_forestdata, n.trees=1000, type='response')
boost_pred = apply(boost_pred, 1, which.max)

boost_pred = as.factor(boost_pred)

# Printing confusion matrix
confusionMatrix(boost_pred, test_forestdata$Cover_Type, positive='1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3   4   5   6   7
##          1 276  14   0   0   2   0   7
```

```
##          2  14 274   2   0   3   2   0
##          3   0   2 241   1   1  15   1
##          4   0   0   9 282   0   1   0
##          5   3  17   1   0 308   0   0
##          6   0   2  17   3   3 292   0
##          7   8   4   0   0   0   0 295
##
## Overall Statistics
##
##                Accuracy : 0.9371
##                  95% CI : (0.9259, 0.9471)
##     No Information Rate : 0.151
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9266
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity            0.9169   0.8754   0.8926   0.9860   0.9716   0.9419
## Specificity            0.9872   0.9882   0.9891   0.9945   0.9882   0.9860
## Pos Pred Value         0.9231   0.9288   0.9234   0.9658   0.9362   0.9211
## Neg Pred Value         0.9861   0.9784   0.9842   0.9978   0.9949   0.9899
## Prevalence             0.1433   0.1490   0.1286   0.1362   0.1510   0.1476
## Detection Rate         0.1314   0.1305   0.1148   0.1343   0.1467   0.1390
## Detection Prevalence   0.1424   0.1405   0.1243   0.1390   0.1567   0.1510
## Balanced Accuracy      0.9521   0.9318   0.9408   0.9903   0.9799   0.9640
##                      Class: 7
## Sensitivity            0.9736
## Specificity            0.9933
## Pos Pred Value         0.9609
## Neg Pred Value         0.9955
## Prevalence             0.1443
## Detection Rate         0.1405
## Detection Prevalence   0.1462
## Balanced Accuracy      0.9835
```

Accuracy:

```
boosting_accuracy <- mean(boost_pred == test_forestdata$Cover_Type)
boosting_accuracy
```

```
## [1] 0.9371429
```

Random Forest:

Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. It contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Higher number of trees produces higher accuracy and thwarts overfitting.

Advantages of Random Forest classifier -

1. Less training time with high accuracy(even for larger datasets).
2. It gives amazing accuracy even when huge amount of data is missing(null values or NaNs)

The only disadvantage with random forest is high prediction time with large number of trees making it ineffective for real time applications. Not to be confused with training time. Even with less training time, predictions can take time.

```
set.seed (1)
random.forest <- randomForest (Cover_Type~ ., data = training_forestdata ,
                               ntree=500, importance = TRUE)
random.forest
```

```
##
## Call:
##  randomForest(formula = Cover_Type ~ ., data = training_forestdata,      ntree = 500, importance = TR
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 1.27232
##                     % Var explained: 68.63
```

```
yhat_bag <- round(predict(random.forest , newdata = test_forestdata, type = "class"))
table(yhat_bag, test_forestdata$Cover_Type)
```

```
##
## yhat_bag   1   2   3   4   5   6   7
##        1 174   1   0   0   0   0   0
##        2  95 228   0   0   1   0   3
##        3  21  63 163   0   4   3   6
##        4   8  14  97 285  60  18   3
##        5   1   6   9   1 252 146  17
##        6   2   1   1   0   0 143  61
##        7   0   0   0   0   0   0 213
```

Accuracy:

```
random_accuracy = mean(yhat_bag == test_forestdata$Cover_Type)
random_accuracy
```

```
## [1] 0.6942857
```

Support Vector Classifier:

While training using support vector classifier, sometimes called a soft margin classifier, rather than seeking the largest possible margin so that every observation is not only on the correct side of the hyperplane but also on the correct side of the margin, we instead allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane. This compensates for any additional points that maybe added to the dataset making it robust to chanes.

For cost=0.01

```
fit_svm_linear <- svm(Cover_Type ~ ., data = training_forestdata, kernel='linear',
                      cost = 0.01, epsilon = 0.01)
summary(fit_svm_linear)
```

```
##
## Call:
## svm(formula = Cover_Type ~ ., data = training_forestdata, kernel = "linear",
##     cost = 0.01, epsilon = 0.01)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.01
##       gamma:  0.1111111
##     epsilon:  0.01
##
##
## Number of Support Vectors:  4850
```

```
# Predicting the Test values
test_svm_pred <- round(predict(fit_svm_linear, test_forestdata))
# Calculating Accuracy on Test dataset
svc_accuracy <- mean(test_svm_pred == test_forestdata$Cover_Type)
svc_accuracy
```

```
## [1] 0.2680952
```

```
svm_lin_tune <- tune(svm, Cover_Type~., data = training_forestdata, kernel = "linear", ranges = list(co
summary(svm_lin_tune)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  0.01
##
## - best performance: 3.974683
##
## - Detailed performance results:
##    cost    error dispersion
## 1  0.01 3.974683  0.2819991
## 2  0.05 4.012462  0.2801579
## 3  0.10 4.017353  0.2797664
## 4  0.50 4.022593  0.2799482
## 5  1.00 4.022889  0.2797861
## 6  5.00 4.023084  0.2795438
## 7 10.00 4.023028  0.2793778
```

As we see in above table, error increases when cost is increased. But just to be safe, we will try cost>10 to see if we get any optimum value.

```
svm_lin_tune <- tune(svm, Cover_Type~., data = training_forestdata, kernel = "linear", ranges = list(co
summary(svm_lin_tune)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##    60
##
## - best performance: 4.022656
##
## - Detailed performance results:
##   cost    error dispersion
## 1   20 4.023329  0.2776305
## 2   30 4.023128  0.2773580
## 3   40 4.022721  0.2771222
## 4   50 4.022683  0.2770606
## 5   60 4.022656  0.2770763
```

We can see that error remains more or less the same for values above 10.

Hence, our first calculation gives us the least error and most accuracy i.e. 26.47%. There is no point in further exploring this model.

Support Vector Machine Classifier:

The support vector classifier seeks a linear boundary, and consequently performs very poorly. Hence, we use non-linear boundary created by support vector machine classifier to fit the model. The support vector machine (SVM) is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using kernels.

For cost= 0.01

```
fit_svm <- svm(Cover_Type ~ ., data = training_forestdata, cost = 0.01, epsilon = 0.01)
summary(fit_svm)
```

```
##
## Call:
## svm(formula = Cover_Type ~ ., data = training_forestdata, cost = 0.01,
##     epsilon = 0.01)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  0.01
##       gamma:  0.1111111
```

```
##     epsilon:  0.01
##
##
## Number of Support Vectors:  4652
```

```
# Predicting the Test values
test_svm_pred <- round(predict(fit_svm, test_forestdata))
# Calculating Accuracy on Test dataset
mean(test_svm_pred == test_forestdata$Cover_Type)
```

```
## [1] 0.2128571
```

As seen above, for cost=0.01 we get only 20.8% accuracy.

```
svm_rad_tune <- tune(svm, Cover_Type~., data = training_forestdata, kernel = "radial", ranges = list(co
summary(svm_rad_tune)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##    10
##
## - best performance: 1.876008
##
## - Detailed performance results:
##    cost    error dispersion
## 1  0.01 3.550015  0.1614774
## 2  0.05 3.124133  0.2131717
## 3  0.10 2.852813  0.1767088
## 4  0.50 2.298714  0.1356609
## 5  1.00 2.168946  0.1317686
## 6  5.00 1.961073  0.1533073
## 7 10.00 1.876008  0.1478100
```

As the cost increases, our error decreases. Let's try fitting the model at cost=10 and check how accurate is the model.

For cost= 10

```
fit_svm <- svm(Cover_Type ~ ., data = training_forestdata, cost = 10, epsilon = 0.01)
summary(fit_svm)
```

```
##
## Call:
## svm(formula = Cover_Type ~ ., data = training_forestdata, cost = 10,
##     epsilon = 0.01)
##
##
```

```
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  0.1111111
##     epsilon:  0.01
##
##
## Number of Support Vectors:  4721
```

```
# Predicting the Test values
test_svm_pred <- round(predict(fit_svm, test_forestdata))
# Calculating Accuracy on Test dataset
mean(test_svm_pred == test_forestdata$Cover_Type)
```

```
## [1] 0.5647619
```

We get 54% accuracy. Let's try furthering the cost using tune() to find optimal values.

```
svm_rad_tune <- tune(svm, Cover_Type~., data = training_forestdata, kernel = "radial", ranges = list(co
summary(svm_rad_tune)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##    75
##
## - best performance: 1.696302
##
## - Detailed performance results:
##   cost    error dispersion
## 1   10 1.842149  0.2020806
## 2   25 1.745826  0.1982894
## 3   50 1.707472  0.1989333
## 4   75 1.696302  0.2007172
## 5  100 1.703938  0.1996110
```

As seen from the table, the error drops from 50 to 75 and then increases again. Hence, the optimal cost lies somewhere between.

With little error and trial we found, optimal value at 65

For cost= 65

```
fit_svm <- svm(Cover_Type ~ ., data = training_forestdata, cost = 65, epsilon = 0.01)
summary(fit_svm)
```

```
##
```

```
## Call:
## svm(formula = Cover_Type ~ ., data = training_forestdata, cost = 65,
##     epsilon = 0.01)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  65
##       gamma:  0.1111111
##     epsilon:  0.01
##
##
## Number of Support Vectors:  4714
```

```
# Predicting the Test values
test_svm_pred <- round(predict(fit_svm, test_forestdata))
# Calculating Accuracy on Test dataset
svmc_accuracy <- mean(test_svm_pred == test_forestdata$Cover_Type)
svmc_accuracy
```

```
## [1] 0.6228571
```

Thus, the accuracy is 61.57% for SVM classifier.

Summary Statistics:

1. Which model will be best suited to classify the type of predominant tree that will develop in each location based on the environment?

```
df = data.frame(
  "Model Name" = "Accuracy",
  "Bagging" = bagging_accuracy,
  "Random Forest" = random_accuracy,
  "Boosting" = boosting_accuracy,
  "Support Vector Classifier" = svc_accuracy,
  "SVM Classifier" = svmc_accuracy
)
Models <- c('Bagging', 'Random Forest', 'Boosting',
            'Support Vector Classifier', 'SVM Classifier')
Accuracy <- c(bagging_accuracy, random_accuracy, boosting_accuracy,
              svc_accuracy,svmc_accuracy)
accuracy_table <- data.frame(Models,Accuracy)
accuracy_table
```

```
##                      Models  Accuracy
## 1                   Bagging 0.7400000
## 2             Random Forest 0.6942857
## 3                  Boosting 0.9371429
## 4 Support Vector Classifier 0.2680952
## 5            SVM Classifier 0.6228571
```

2. What are the most prevalent tree species in the Roosevelt National Forest?

Counting the trees for every type in the Roosevelt National Forest:

```
new_forestdata %>%
  mutate(Cover_Type = str_replace_all(Cover_Type,
                                      c(`1` = 'Spruce/Fir',
                                        `2` = 'Lodgepole Pine',
                                        `3` = 'Ponderosa Pine',
                                        `4` = 'Cottonwood/Willow',
                                        `5` = 'Aspen',
                                        `6` = 'Douglas Fir',
                                        `7` = 'Krummholz')),
         Cover_Type = as.factor(Cover_Type)) %>%
  group_by(Cover_Type) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  knitr::kable(booktabs = TRUE)
```

Cover_Type	Count
Lodgepole Pine	283301
Spruce/Fir	211840
Ponderosa Pine	35754
Krummholz	20510
Douglas Fir	17367
Aspen	9493
Cottonwood/Willow	2747

3. Which tree types can grow in most diverse environments?

After looking at the EDA we can say that Krummholz seems to grow in much diverse environments like widespread elevation, distance to hydrology and soil type.

4. Are there any tree species which are susceptible to environmental factors?

Cottonwood/Willow has lowest count of trees in the Roosevelt National Forest and the EDA also confirms that this tree type is the most susceptible to all the factors.

References

1. https://www.kaggle.com/code/rsizem2/forest-cover-type-feature-engineering
2. https://rstudio-pubs-static.s3.amazonaws.com/160297_f7bcb8d140b74bd19b758eb328344908.html