MACHINE LEARNING PROJECT

Forest Cover Type Prediction

By : Darshan Dalvi , Deepak Udyavar, Prashant Kasalkar, Sakshi Kalani, Saptak Dalvi

Source : [https://www.kaggle.com/c/forest-cover-type-prediction/data (https://www.kaggle.com/c/forest-cover-type-prediction/data)](https://www.kaggle.com/c/forest-cover-type-prediction/data)

Motivation :

The data is cartographic variables (as opposed to remotely sensed data). The actual forest cover type for a given 30 x 30 meter cell was determined from US Forest Service (USFS) Region 2 Resource Information System data. Independent variables were then derived from data obtained from the US Geological Survey and USFS. The data is in raw form (not scaled) and contains binary columns of data for qualitative independent variables such as wilderness areas and soil type. The areas of forest under study represent forests with minimal human-caused disturbances, so that existing forest cover types are more a result of ecological processes rather than forest management practices.

Data Overview :

The study area includes four wilderness areas located in the Roosevelt National Forest of northern Colorado which are as follows:
1 - Rawah Wilderness Area
2 - Neota Wilderness Area
3 - Comanche Peak Wilderness Area
4 - Cache la Poudre Wilderness Area

Our goal is to predict the forest cover type. The seven types of forest cover are :
1 - Spruce/Fir
2 - Lodgepole Pine
3 - Ponderosa Pine
4 - Cottonwood/Willow
5 - Aspen
6 - Douglas-fir
7 - Krummholz

- This is a classification problem where the target could belong to any of the seven classes.

Data Fields:

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 (0 to 255 index) - Hillshade index at 9am, summer solstice
Hillshade_Noon (0 to 255 index) - Hillshade index at noon, summer solstice
Hillshade_3pm (0 to 255 index) - Hillshade index at 3pm, summer solstice
Horizontal_Distance_To_Fire_Points - Horz Dist to nearest wildfire ignition points
Wilderness_Area (4 binary columns, 0 = absence or 1 = presence) - Wilderness area designation
Soil_Type (40 binary columns, 0 = absence or 1 = presence) - Soil Type designation
Cover_Type (7 types, integers 1 to 7) - Forest Cover Type designation

Question: Given the other attributes, what will be the 'Cover_Type'?

The training set (15120 observations) contains both features and the Cover_Type. The test set (565892 observations) contains only the features.

Assumptions:

1. Ecology of the areas across which the data is collected is similar.
2. Seasonal changes are constant across all the observations.
3. The dataset is recently collected.

Limitations:

1. Environmental factors affecting the growth of any cover type is not taken into consideration.
2. Human error while collecting data is not accounted for.
3. Management practices that might have affected the growth is not accounted for.

About the data :

In [49]:
```
%matplotlib inline
import warnings
import seaborn as sns
warnings.filterwarnings("ignore")
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
dataset = pd.read_csv("train.csv")
```

In [50]:
```
dataset = dataset.iloc[:,1:]
print(dataset.shape)
```

```
(15120, 55)
```

In [51]:
```
print(dataset.dtypes)
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
Elevation                               int64
Aspect                                  int64
Slope                                   int64
Horizontal_Distance_To_Hydrology        int64
Vertical_Distance_To_Hydrology          int64
Horizontal_Distance_To_Roadways         int64
Hillshade_9am                           int64
Hillshade_Noon                          int64
Hillshade_3pm                           int64
Horizontal_Distance_To_Fire_Points      int64
Wilderness_Area1                        int64
Wilderness_Area2                        int64
Wilderness_Area3                        int64
Wilderness_Area4                        int64
Soil_Type1                              int64
Soil_Type2                              int64
Soil_Type3                              int64
Soil_Type4                              int64
Soil_Type5                              int64
Soil_Type6                              int64
Soil_Type7                              int64
Soil_Type8                              int64
Soil_Type9                              int64
Soil_Type10                             int64
Soil_Type11                             int64
Soil_Type12                             int64
Soil_Type13                             int64
Soil_Type14                             int64
Soil_Type15                             int64
Soil_Type16                             int64
Soil_Type17                             int64
Soil_Type18                             int64
Soil_Type19                             int64
Soil_Type20                             int64
Soil_Type21                             int64
Soil_Type22                             int64
Soil_Type23                             int64
Soil_Type24                             int64
Soil_Type25                             int64
Soil_Type26                             int64
Soil_Type27                             int64
Soil_Type28                             int64
Soil_Type29                             int64
Soil_Type30                             int64
Soil_Type31                             int64
Soil_Type32                             int64
Soil_Type33                             int64
Soil_Type34                             int64
Soil_Type35                             int64
Soil_Type36                             int64
Soil_Type37                             int64
Soil_Type38                             int64
Soil_Type39                             int64
Soil_Type40                             int64
Cover_Type                              int64
dtype: object
```

Data Pre-processing

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [52]:
```
pd.set_option('display.max_columns', None)
print(dataset.describe())
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
         Elevation       Aspect        Slope  \
count  15120.000000  15120.000000  15120.000000
mean    2749.322553    156.676653     16.501587
std      417.678187    110.085801      8.453927
min     1863.000000      0.000000      0.000000
25%     2376.000000     65.000000     10.000000
50%     2752.000000    126.000000     15.000000
75%     3104.000000    261.000000     22.000000
max     3849.000000    360.000000     52.000000

       Horizontal_Distance_To_Hydrology  Vertical_Distance_To_Hydrology
  \
count                      15120.000000                    15120.000000

mean                         227.195701                       51.076521

std                          210.075296                       61.239406

min                            0.000000                     -146.000000

25%                           67.000000                        5.000000

50%                          180.000000                       32.000000

75%                          330.000000                       79.000000

max                         1343.000000                      554.000000


       Horizontal_Distance_To_Roadways  Hillshade_9am  Hillshade_Noon
\
count                     15120.000000   15120.000000    15120.000000

mean                       1714.023214     212.704299      218.965608

std                        1325.066358      30.561287       22.801966

min                           0.000000       0.000000       99.000000

25%                         764.000000     196.000000      207.000000

50%                        1316.000000     220.000000      223.000000

75%                        2270.000000     235.000000      235.000000

max                        6890.000000     254.000000      254.000000


       Hillshade_3pm  Horizontal_Distance_To_Fire_Points  Wilderness_Ar
ea1  \
count   15120.000000                        15120.000000      15120.000
000
mean      135.091997                         1511.147288          0.237
897
std        45.895189                         1099.936493          0.425
810
min         0.000000                            0.000000          0.000
```

```
000
25%       106.000000                          730.000000          0.000
000
50%       138.000000                         1256.000000          0.000
000
75%       167.000000                         1988.250000          0.000
000
max       248.000000                         6993.000000          1.000
000

       Wilderness_Area2  Wilderness_Area3  Wilderness_Area4    Soil_Typ
e1  \
count      15120.000000      15120.000000      15120.000000  15120.0000
00
mean           0.033003          0.419907          0.309193      0.0234
79
std            0.178649          0.493560          0.462176      0.1514
24
min            0.000000          0.000000          0.000000      0.0000
00
25%            0.000000          0.000000          0.000000      0.0000
00
50%            0.000000          0.000000          0.000000      0.0000
00
75%            0.000000          1.000000          1.000000      0.0000
00
max            1.000000          1.000000          1.000000      1.0000
00

         Soil_Type2    Soil_Type3    Soil_Type4    Soil_Type5    Soil_T
ype6  \
count  15120.000000  15120.000000  15120.000000  15120.000000  15120.00
0000
mean       0.041204      0.063624      0.055754      0.010913      0.04
2989
std        0.198768      0.244091      0.229454      0.103896      0.20
2840
min        0.000000      0.000000      0.000000      0.000000      0.00
0000
25%        0.000000      0.000000      0.000000      0.000000      0.00
0000
50%        0.000000      0.000000      0.000000      0.000000      0.00
0000
75%        0.000000      0.000000      0.000000      0.000000      0.00
0000
max        1.000000      1.000000      1.000000      1.000000      1.00
0000

       Soil_Type7    Soil_Type8    Soil_Type9   Soil_Type10   Soil_Type
11  \
count     15120.0  15120.000000  15120.000000  15120.000000  15120.0000
00
mean          0.0      0.000066      0.000661      0.141667      0.0268
52
std           0.0      0.008133      0.025710      0.348719      0.1616
56
min           0.0      0.000000      0.000000      0.000000      0.0000
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
00
25%          0.0      0.000000      0.000000      0.000000      0.0000
00
50%          0.0      0.000000      0.000000      0.000000      0.0000
00
75%          0.0      0.000000      0.000000      0.000000      0.0000
00
max          0.0      1.000000      1.000000      1.000000      1.0000
00

        Soil_Type12   Soil_Type13   Soil_Type14  Soil_Type15   Soil_Typ
e16  \
count  15120.000000  15120.000000  15120.000000      15120.0  15120.000
000
mean       0.015013      0.031481      0.011177          0.0      0.007
540
std        0.121609      0.174621      0.105133          0.0      0.086
506
min        0.000000      0.000000      0.000000          0.0      0.000
000
25%        0.000000      0.000000      0.000000          0.0      0.000
000
50%        0.000000      0.000000      0.000000          0.0      0.000
000
75%        0.000000      0.000000      0.000000          0.0      0.000
000
max        1.000000      1.000000      1.000000          0.0      1.000
000

        Soil_Type17   Soil_Type18   Soil_Type19   Soil_Type20   Soil_Ty
pe21  \
count  15120.000000  15120.000000  15120.000000  15120.000000  15120.00
0000
mean       0.040476      0.003968      0.003042      0.009193      0.00
1058
std        0.197080      0.062871      0.055075      0.095442      0.03
2514
min        0.000000      0.000000      0.000000      0.000000      0.00
0000
25%        0.000000      0.000000      0.000000      0.000000      0.00
0000
50%        0.000000      0.000000      0.000000      0.000000      0.00
0000
75%        0.000000      0.000000      0.000000      0.000000      0.00
0000
max        1.000000      1.000000      1.000000      1.000000      1.00
0000

        Soil_Type22   Soil_Type23   Soil_Type24   Soil_Type25   Soil_Ty
pe26  \
count  15120.000000  15120.000000  15120.000000  15120.000000  15120.00
0000
mean       0.022817      0.050066      0.016997      0.000066      0.00
3571
std        0.149326      0.218089      0.129265      0.008133      0.05
9657
min        0.000000      0.000000      0.000000      0.000000      0.00
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
0000
25%        0.000000     0.000000     0.000000     0.000000     0.00
0000
50%        0.000000     0.000000     0.000000     0.000000     0.00
0000
75%        0.000000     0.000000     0.000000     0.000000     0.00
0000
max        1.000000     1.000000     1.000000     1.000000     1.00
0000

        Soil_Type27  Soil_Type28  Soil_Type29  Soil_Type30  Soil_Ty
pe31  \
count  15120.000000 15120.000000 15120.000000 15120.000000 15120.00
0000
mean       0.000992     0.000595     0.085384     0.047950     0.02
1958
std        0.031482     0.024391     0.279461     0.213667     0.14
6550
min        0.000000     0.000000     0.000000     0.000000     0.00
0000
25%        0.000000     0.000000     0.000000     0.000000     0.00
0000
50%        0.000000     0.000000     0.000000     0.000000     0.00
0000
75%        0.000000     0.000000     0.000000     0.000000     0.00
0000
max        1.000000     1.000000     1.000000     1.000000     1.00
0000

        Soil_Type32  Soil_Type33  Soil_Type34  Soil_Type35  Soil_Ty
pe36  \
count  15120.000000 15120.000000 15120.000000 15120.000000 15120.00
0000
mean       0.045635     0.040741     0.001455     0.006746     0.00
0661
std        0.208699     0.197696     0.038118     0.081859     0.02
5710
min        0.000000     0.000000     0.000000     0.000000     0.00
0000
25%        0.000000     0.000000     0.000000     0.000000     0.00
0000
50%        0.000000     0.000000     0.000000     0.000000     0.00
0000
75%        0.000000     0.000000     0.000000     0.000000     0.00
0000
max        1.000000     1.000000     1.000000     1.000000     1.00
0000

        Soil_Type37  Soil_Type38  Soil_Type39  Soil_Type40   Cover_
Type
count  15120.000000 15120.000000 15120.000000 15120.000000 15120.00
0000
mean       0.002249     0.048148     0.043452     0.030357     4.00
0000
std        0.047368     0.214086     0.203880     0.171574     2.00
0066
min        0.000000     0.000000     0.000000     0.000000     1.00
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
0000
25%        0.000000     0.000000     0.000000     0.000000     2.00
0000
50%        0.000000     0.000000     0.000000     0.000000     4.00
0000
75%        0.000000     0.000000     0.000000     0.000000     6.00
0000
max        1.000000     1.000000     1.000000     1.000000     7.00
0000
```

- No attribute is missing as count is 15120 for all attributes. So, no deletion is required.
- Attributes Soil_Type7 and Soil_Type15 can be removed as they are constant.
- Wilderness_Area and Soil_Type are one hot encoded. Hence, they could be converted back for some analysis.
- Scales are not the same for all. Hence, rescaling and standardization may be necessary for some algorithms.
- Negative values are present in Vertical_Distance_To_Hydrology. Hence, some tests such as chi-sq cant be used.

Descriptive Statistics

In [53]:
```
print(dataset.skew())
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
Elevation                                 0.075640
Aspect                                    0.450935
Slope                                     0.523658
Horizontal_Distance_To_Hydrology          1.488052
Vertical_Distance_To_Hydrology            1.537776
Horizontal_Distance_To_Roadways           1.247811
Hillshade_9am                            -1.093681
Hillshade_Noon                           -0.953232
Hillshade_3pm                            -0.340827
Horizontal_Distance_To_Fire_Points        1.617099
Wilderness_Area1                          1.231244
Wilderness_Area2                          5.228781
Wilderness_Area3                          0.324594
Wilderness_Area4                          0.825798
Soil_Type1                                6.294716
Soil_Type2                                4.617019
Soil_Type3                                3.575995
Soil_Type4                                3.872721
Soil_Type5                                9.416209
Soil_Type6                                4.506716
Soil_Type7                                0.000000
Soil_Type8                              122.963409
Soil_Type9                               38.849712
Soil_Type10                               2.055410
Soil_Type11                               5.854551
Soil_Type12                               7.977205
Soil_Type13                               5.366836
Soil_Type14                               9.300318
Soil_Type15                               0.000000
Soil_Type16                              11.387050
Soil_Type17                               4.663945
Soil_Type18                              15.781426
Soil_Type19                              18.048915
Soil_Type20                              10.286265
Soil_Type21                              30.695081
Soil_Type22                               6.391991
Soil_Type23                               4.126701
Soil_Type24                               7.474026
Soil_Type25                             122.963409
Soil_Type26                              16.645076
Soil_Type27                              31.704896
Soil_Type28                              40.955261
Soil_Type29                               2.967651
Soil_Type30                               4.231913
Soil_Type31                               6.524804
Soil_Type32                               4.354839
Soil_Type33                               4.646742
Soil_Type34                              26.161230
Soil_Type35                              12.052838
Soil_Type36                              38.849712
Soil_Type37                              21.018939
Soil_Type38                               4.221771
Soil_Type39                               4.479186
Soil_Type40                               5.475256
Cover_Type                                0.000000
dtype: float64
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

- Values close to 0 show less skew
- Several attributes in Soil_Type show a large skew. Hence, some algos may benefit if skew is corrected

In [54]:
```
rem = []

#Add constant columns as they don't help in prediction process
for c in dataset.columns:
    if dataset[c].std() == 0: #standard deviation is zero
        rem.append(c)

#drop the columns
dataset.drop(rem,axis=1,inplace=True)

print(rem)
```

```
['Soil_Type7', 'Soil_Type15']
```

In [55]:
```
dataset.groupby('Cover_Type').size()
# We see that all classes have an equal presence. No class re-balancing
 is required.
```

Out[55]:
```
Cover_Type
1    2160
2    2160
3    2160
4    2160
5    2160
6    2160
7    2160
dtype: int64
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [56]:
```
dataset.corr()
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

Out[56]:

	Elevation	Aspect	Slope	Horizontal_Distance
Elevation	1.000000	-0.011096	-0.312640	0.412712
Aspect	-0.011096	1.000000	0.028148	0.040732
Slope	-0.312640	0.028148	1.000000	-0.055976
Horizontal_Distance_To_Hydrology	0.412712	0.040732	-0.055976	1.000000
Vertical_Distance_To_Hydrology	0.122092	0.056412	0.265314	0.652142
Horizontal_Distance_To_Roadways	0.578659	0.066184	-0.277049	0.203397
Hillshade_9am	0.097900	-0.593997	-0.200072	-0.033803
Hillshade_Noon	0.215782	0.324912	-0.612613	0.080047
Hillshade_3pm	0.089518	0.635022	-0.326887	0.080833
Horizontal_Distance_To_Fire_Points	0.443563	-0.052169	-0.239527	0.158817
Wilderness_Area1	0.330417	-0.131262	-0.152820	-0.009402
Wilderness_Area2	0.261729	0.028238	-0.065923	0.087484
Wilderness_Area3	0.354025	0.032578	-0.113033	0.200532
Wilderness_Area4	-0.783651	0.075228	0.286985	-0.239303
Soil_Type1	-0.218818	-0.024538	0.099355	-0.084766
Soil_Type2	-0.147947	-0.020970	-0.081498	0.024234
Soil_Type3	-0.307523	-0.069120	0.265541	-0.089578
Soil_Type4	-0.125342	0.018019	0.087841	-0.059398
Soil_Type5	-0.141478	0.000343	0.074720	-0.025247
Soil_Type6	-0.187354	-0.006066	-0.047868	0.021203
Soil_Type8	0.002934	0.001723	-0.012989	0.002819
Soil_Type9	-0.010571	-0.019391	-0.022220	-0.005523
Soil_Type10	-0.357816	0.111959	0.255804	-0.112852
Soil_Type11	-0.037906	-0.034549	-0.109798	0.026150
Soil_Type12	0.017432	-0.044142	-0.115088	0.034306
Soil_Type13	0.039304	0.024312	0.119863	0.026595
Soil_Type14	-0.140619	0.001181	-0.054085	-0.111878
Soil_Type16	-0.066252	0.027121	-0.064321	-0.084804
Soil_Type17	-0.200663	0.029870	-0.124375	-0.159717
Soil_Type18	-0.035173	-0.042140	-0.069326	-0.018282
Soil_Type19	0.029808	0.007570	-0.047742	-0.033946
Soil_Type20	0.008548	-0.023330	-0.068508	-0.062873

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

	Elevation	Aspect	Slope	Horizontal_Distance
Soil_Type21	0.032509	0.018815	-0.033935	-0.025207
Soil_Type22	0.146236	0.022301	-0.076393	-0.007840
Soil_Type23	0.159872	0.041880	-0.184528	-0.087963
Soil_Type24	0.097647	0.005480	0.017982	0.046915
Soil_Type25	0.010586	-0.002340	0.011062	-0.004615
Soil_Type26	0.020669	-0.009775	-0.030700	0.027879
Soil_Type27	0.040019	0.018986	0.012295	0.064616
Soil_Type28	-0.001077	0.026330	0.036082	0.019663
Soil_Type29	0.165304	-0.063428	-0.083108	0.033854
Soil_Type30	0.048204	-0.086897	0.118725	-0.032540
Soil_Type31	0.093191	0.008160	-0.076851	0.060886
Soil_Type32	0.172349	0.003700	-0.147258	0.138275
Soil_Type33	0.123821	0.018719	0.072027	0.062121
Soil_Type34	0.021876	0.012927	-0.030590	0.072485
Soil_Type35	0.120157	-0.004235	-0.048855	-0.015446
Soil_Type36	0.040571	0.003160	-0.004570	0.077251
Soil_Type37	0.073825	-0.046309	0.003129	-0.009549
Soil_Type38	0.323440	0.043860	-0.148342	0.131444
Soil_Type39	0.296405	-0.031342	0.051900	0.066284
Soil_Type40	0.306755	0.007208	-0.043513	0.242304
Cover_Type	0.016090	0.008015	0.087722	-0.010515

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

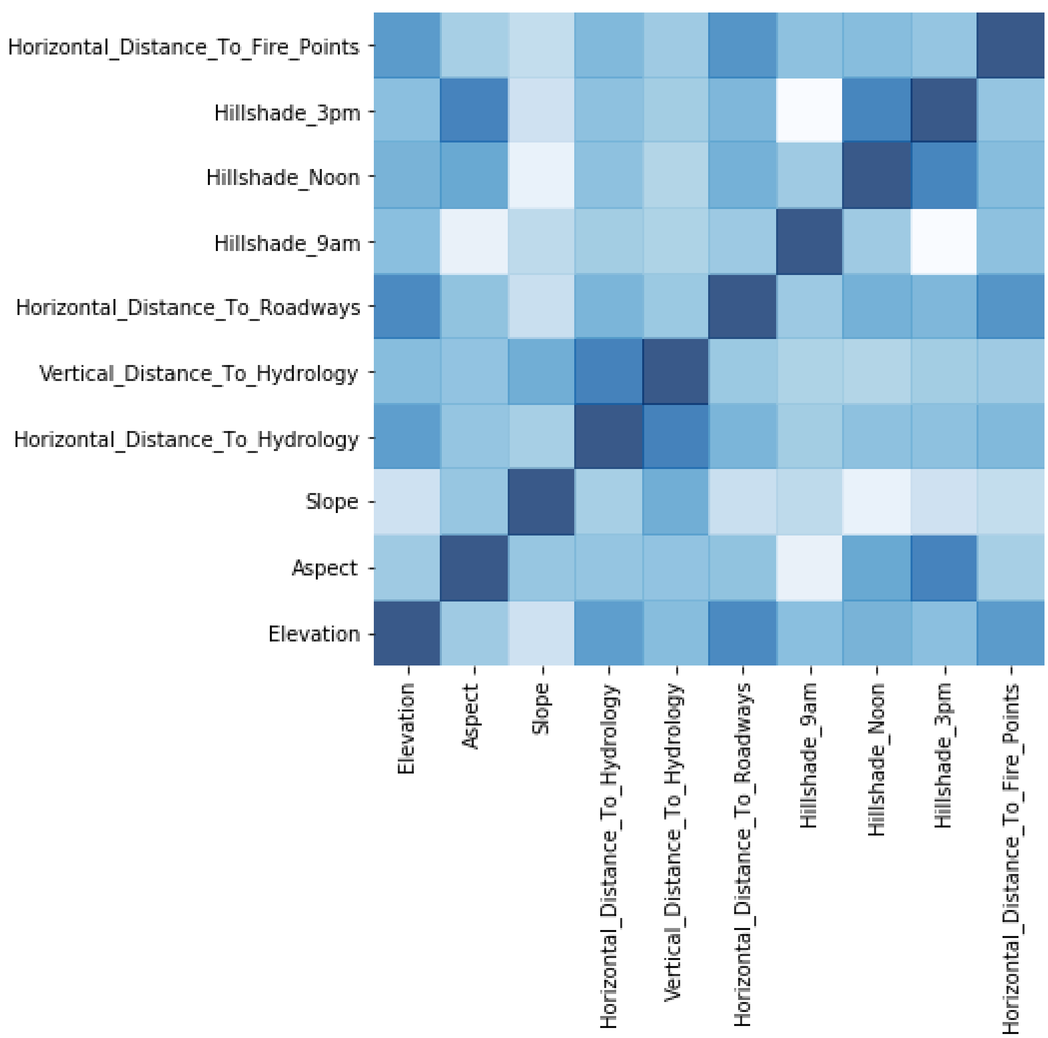
In [57]:
```
dataset_corr = dataset.corr()

# Set the threshold to select only only highly correlated attributes
threshold = 0.5

# List of pairs along with correlation above threshold
corr_list = []
cols=dataset.columns
#len(dataset)
size = len(dataset_corr)
#Search for the highly correlated pairs
for i in range(0,size): #for 'size' features
    for j in range(i+1,size): #avoid repetition
        if (dataset_corr.iloc[i,j] >= threshold and dataset_corr.iloc[i,
j] < 1) or (dataset_corr.iloc[i,j] < 0 and dataset_corr.iloc[i,j] <= -th
reshold):
            corr_list.append([dataset_corr.iloc[i,j],i,j]) #store correl
ation and columns index

#Sort to show higher ones first
s_corr_list = sorted(corr_list,key=lambda x: -abs(x[0]))

#Print correlations and column names
for v,i,j in s_corr_list:
    print ("%s and %s = %.2f" % (cols[i],cols[j],v))
```

```
Elevation and Wilderness_Area4 = -0.78
Hillshade_9am and Hillshade_3pm = -0.78
Horizontal_Distance_To_Hydrology and Vertical_Distance_To_Hydrology =
0.65
Aspect and Hillshade_3pm = 0.64
Hillshade_Noon and Hillshade_3pm = 0.61
Slope and Hillshade_Noon = -0.61
Aspect and Hillshade_9am = -0.59
Elevation and Horizontal_Distance_To_Roadways = 0.58
Wilderness_Area3 and Wilderness_Area4 = -0.57
Wilderness_Area1 and Soil_Type29 = 0.55
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [58]:
```
con = ['Elevation' , 'Aspect' , 'Slope', 'Horizontal_Distance_To_Hydrolo
gy' , 'Vertical_Distance_To_Hydrology' ,'Horizontal_Distance_To_Roadway
s','Hillshade_9am','Hillshade_Noon','Hillshade_3pm','Horizontal_Distance
_To_Fire_Points']
con_variables = dataset[con]

#Cor = con_variables.iloc[:,0:10]
Cor_matrix = con_variables.corr(method='pearson', min_periods=1)
#print(Cor_matrxi)

fig, ax = plt.subplots()
heatmap = ax.pcolor(Cor_matrix, cmap=plt.cm.Blues, alpha=0.8)
fig = plt.gcf()
fig.set_size_inches(6, 6)
ax.set_frame_on(False)
ax.set_yticks(np.arange(10) + 0.5, minor=False)
ax.set_xticks(np.arange(10) + 0.5, minor=False)
ax.set_xticklabels(con[0:10], minor=False)
ax.set_yticklabels(con[0:10], minor=False)
plt.xticks(rotation=90)
```

Out[58]:
```
(array([ 0.5,  1.5,  2.5,  3.5,  4.5,  5.5,  6.5,  7.5,  8.5,  9.5]),
 <a list of 10 Text xticklabel objects>)
```



File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [59]:
```
# define the data/predictors as the pre-set feature names
df = pd.DataFrame(dataset, columns=["Elevation"])

# Put the target  in another DataFrame
target = pd.DataFrame(dataset, columns=["Cover_Type"])

import statsmodels.api as sm

X = df["Elevation"]
y = target["Cover_Type"]

# Note the difference in argument order
model = sm.OLS(y, X).fit()
# make the predictions by the model
predictions = model.predict(X)

# Print out the statistics
model.summary()
```

Out[59]:

OLS Regression Results

Dep. Variable:	Cover_Type	**R-squared:**	0.784
Model:	OLS	**Adj. R-squared:**	0.784
Method:	Least Squares	**F-statistic:**	5.483e+04
Date:	Wed, 18 Apr 2018	**Prob (F-statistic):**	0.00
Time:	06:14:19	**Log-Likelihood:**	-32521.
No. Observations:	15120	**AIC:**	6.504e+04
Df Residuals:	15119	**BIC:**	6.505e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Elevation	0.0014	6.08e-06	234.165	0.000	0.001	0.001

Omnibus:	5998.494	**Durbin-Watson:**	1.144
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1320.370
Skew:	-0.482	**Prob(JB):**	1.93e-287
Kurtosis:	1.920	**Cond. No.**	1.00

- We can see here that this model has a much higher R-squared value of 0.784, meaning that this model explains 78.4% of the variance in our dependent variable.

In [60]:
```
# define the data/predictors as the pre-set feature names
df = pd.DataFrame(dataset, columns=["Slope"])

# Put the target  in another DataFrame
target = pd.DataFrame(dataset, columns=["Cover_Type"])

import statsmodels.api as sm

X = df["Slope"]
y = target["Cover_Type"]

# Note the difference in argument order
model = sm.OLS(y, X).fit()
# make the predictions by the model
predictions = model.predict(X)

# Print out the statistics
model.summary()
```

Out[60]:

OLS Regression Results

Dep. Variable:	Cover_Type	**R-squared:**	0.662
Model:	OLS	**Adj. R-squared:**	0.662
Method:	Least Squares	**F-statistic:**	2.968e+04
Date:	Wed, 18 Apr 2018	**Prob (F-statistic):**	0.00
Time:	06:14:22	**Log-Likelihood:**	-35891.
No. Observations:	15120	**AIC:**	7.178e+04
Df Residuals:	15119	**BIC:**	7.179e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Slope	0.1963	0.001	172.269	0.000	0.194	0.199

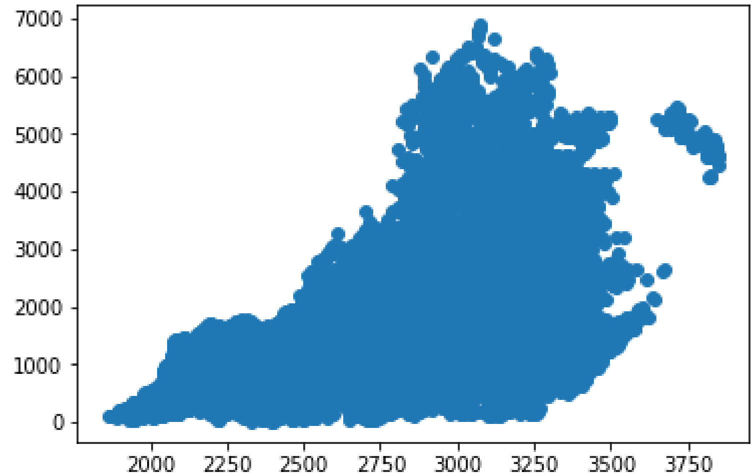
Omnibus:	517.060	**Durbin-Watson:**	1.205
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	255.416
Skew:	0.107	**Prob(JB):**	3.44e-56
Kurtosis:	2.400	**Cond. No.**	1.00

- We can see here that this model has a much higher R-squared value of 0.662.

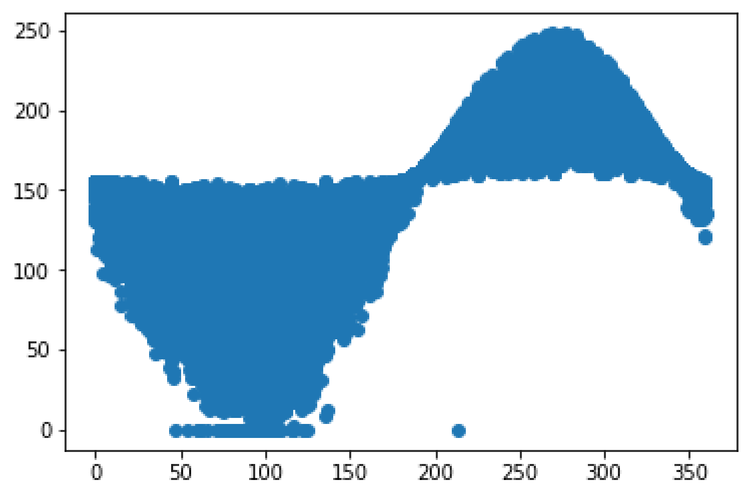
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

Data Visualization

In [61]:
```
#Scatter Plot for Elevation and Horizontal_Distance_To_Roadways
x = dataset['Elevation']
y = dataset['Horizontal_Distance_To_Roadways']

plt.scatter(x,y)
plt.show()
```

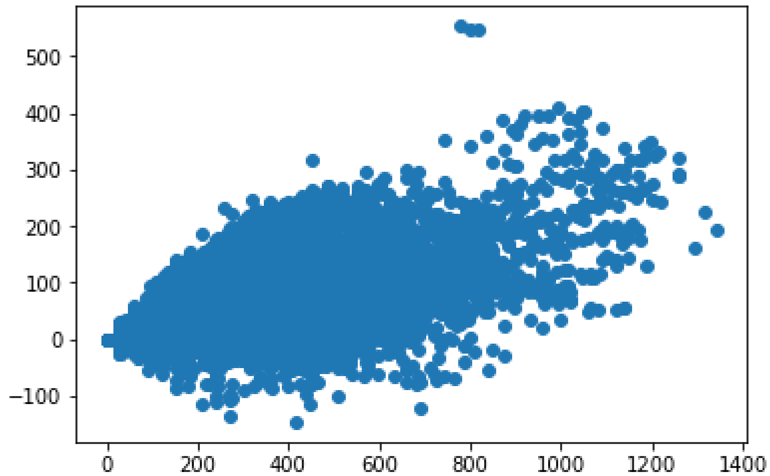


In [62]:
```
#Scatter Plot for Aspect vs Hillshade_3pm
x = dataset['Aspect']
y = dataset['Hillshade_3pm']

plt.scatter(x,y)
plt.show()
```

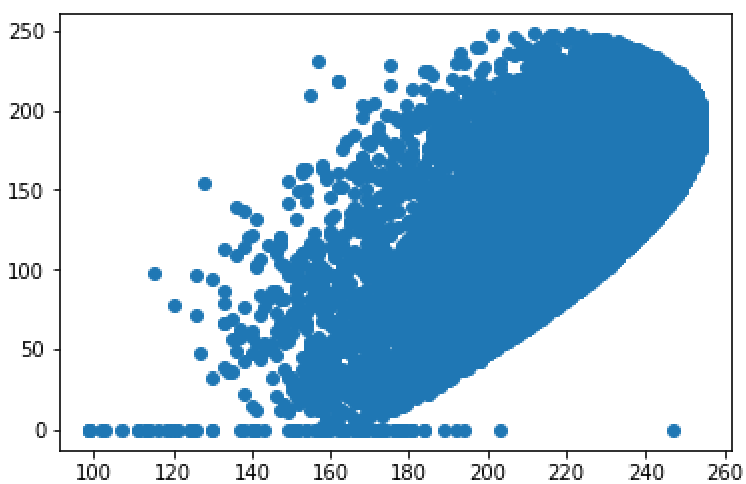


File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [63]:
```
#Scatter Plot for Horizontal_Distance_To_Hydrology vs the Vertical Dista
nce
x = dataset['Horizontal_Distance_To_Hydrology']
y = dataset['Vertical_Distance_To_Hydrology']

plt.scatter(x,y)
plt.show()
```



In [64]:
```
#ScatterPlot for Hillshade_Noon vs Hillshade_3pm
x = dataset['Hillshade_Noon']
y = dataset['Hillshade_3pm']

plt.scatter(x,y)
plt.show()
```

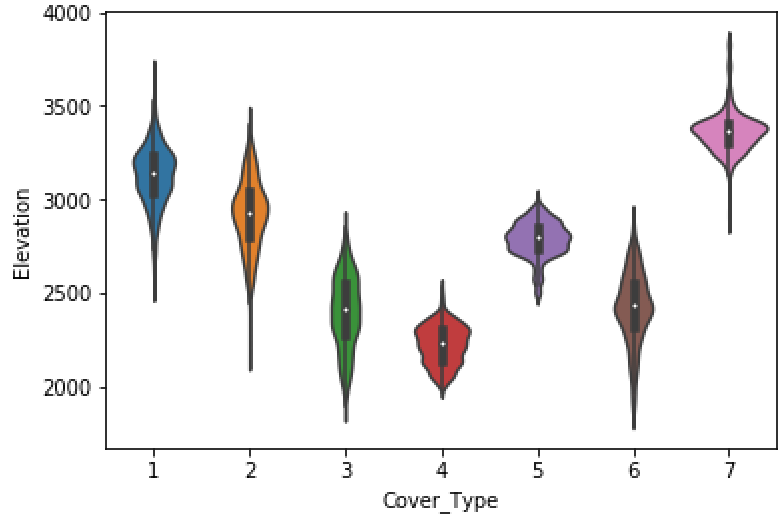


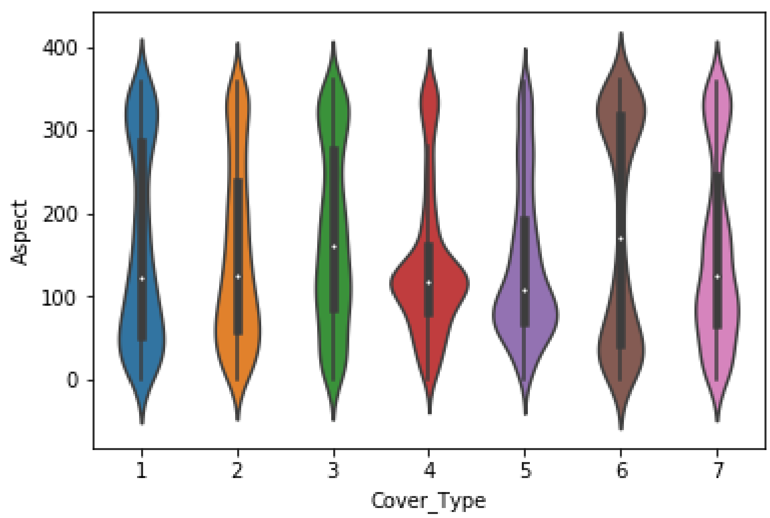
In [65]:
```
cols = dataset.columns
size = len(cols)-1
x = cols[size]
y = cols[0:size]
```

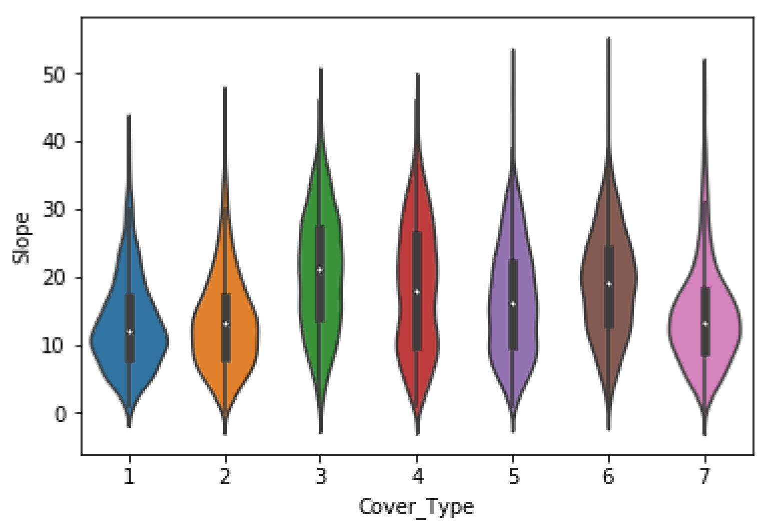
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [66]:
```
for i in range(0,size):
    sns.violinplot(data=dataset,x=x,y=y[i])
    plt.show()
```

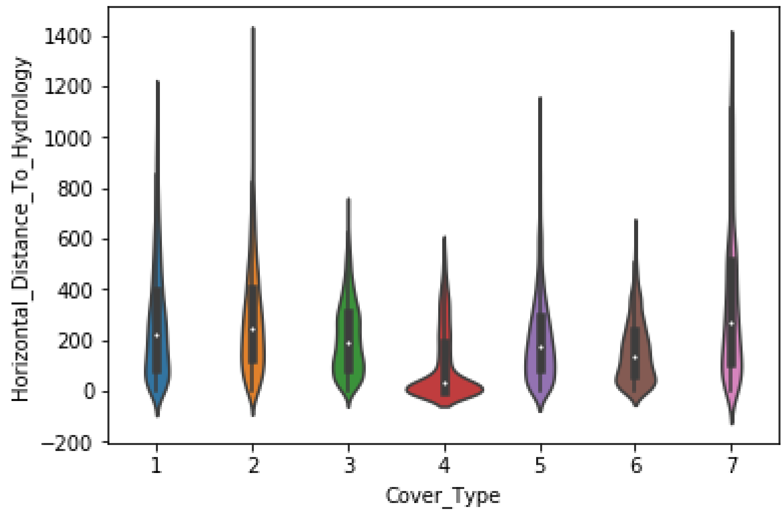
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

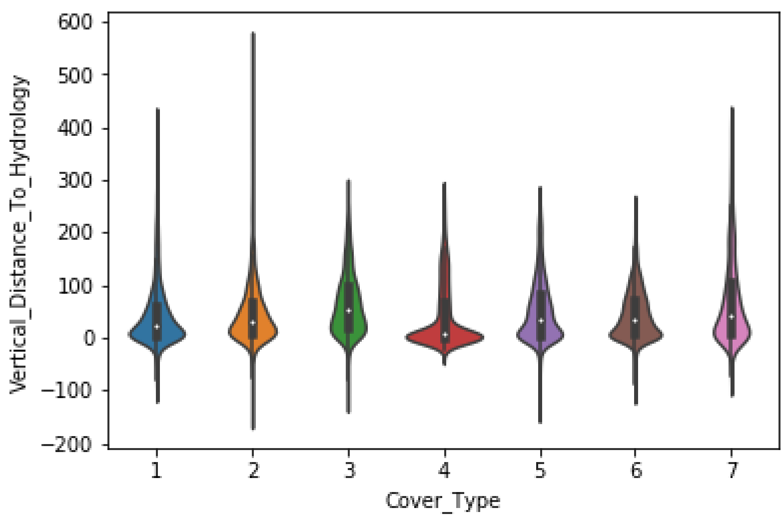


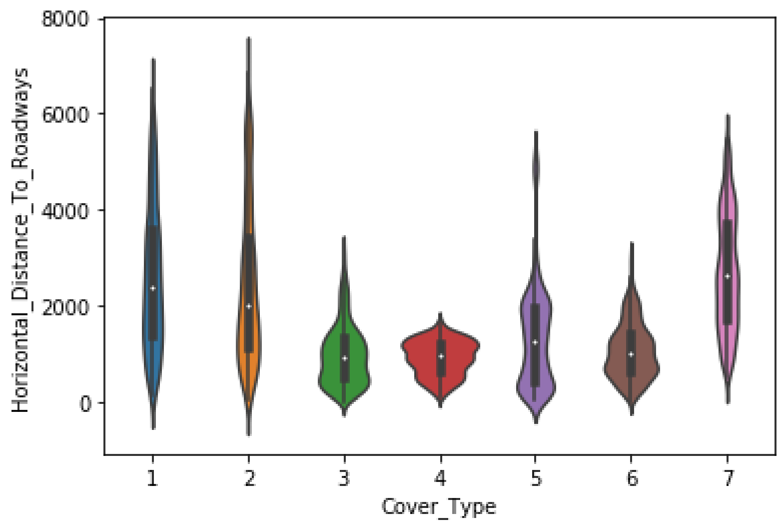




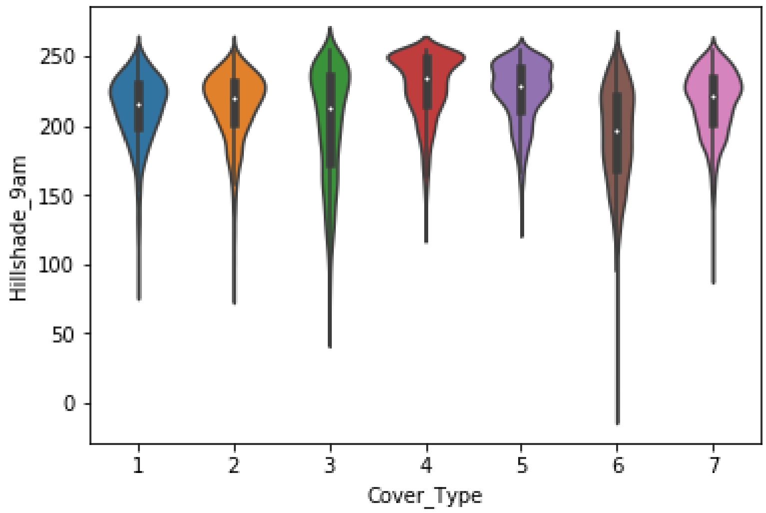
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

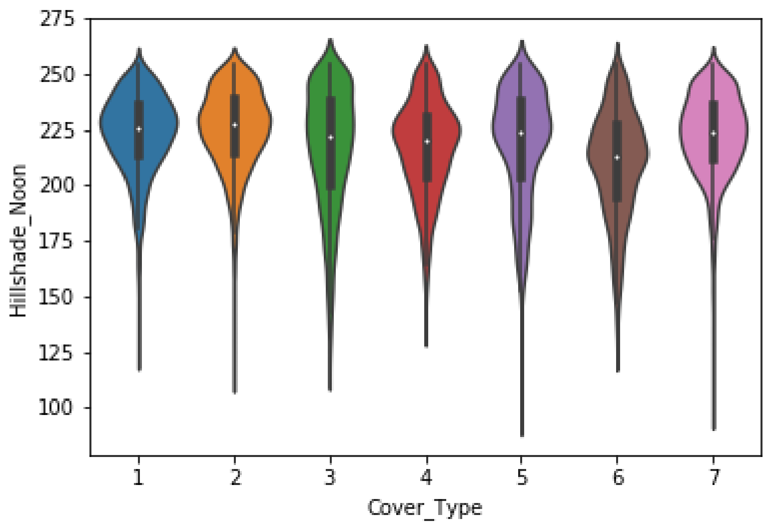


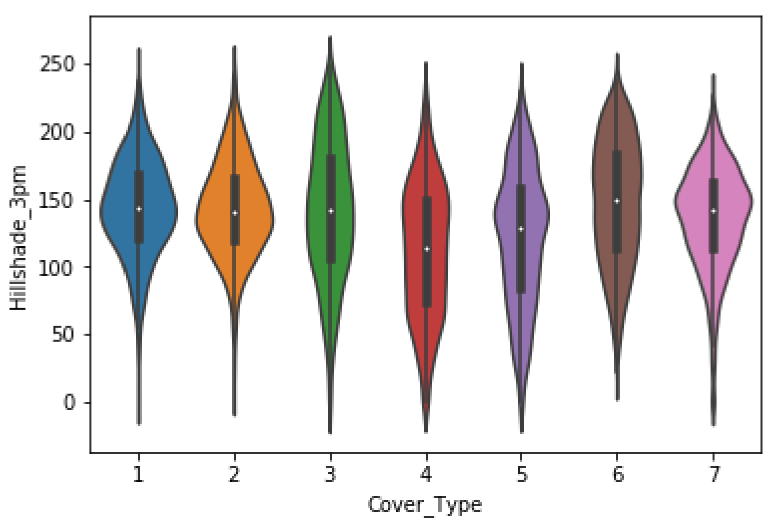




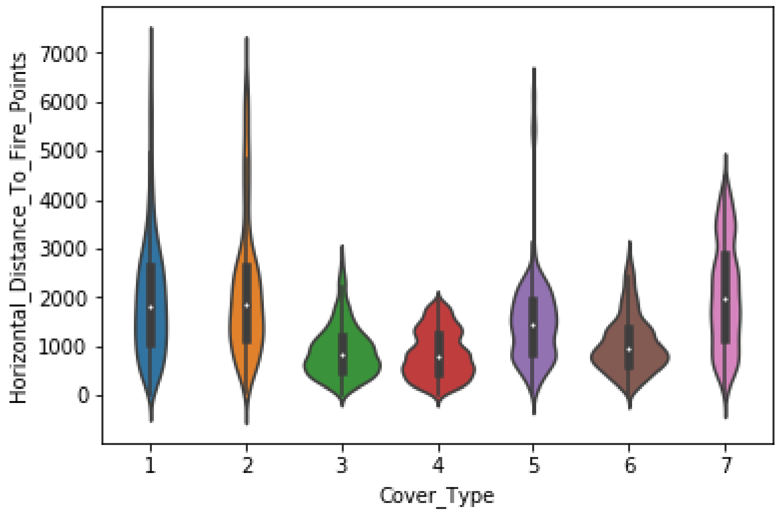
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

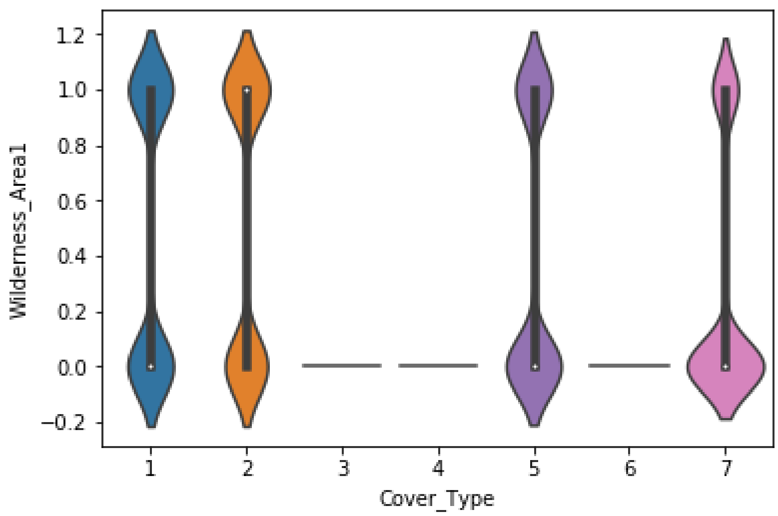


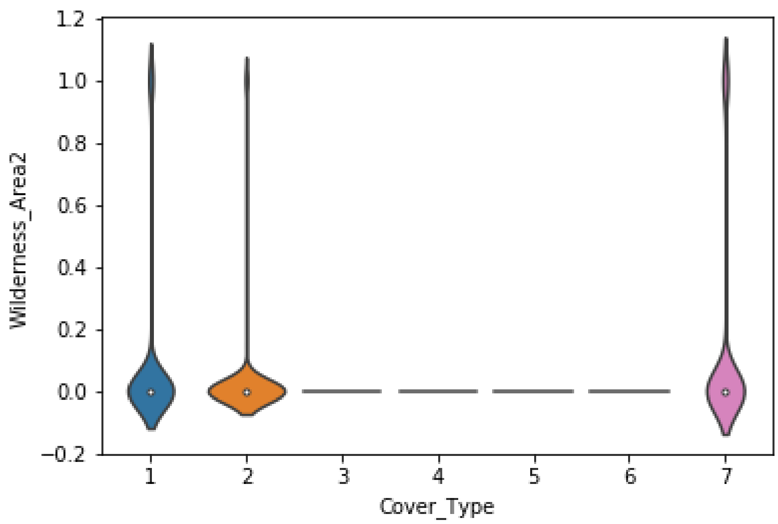


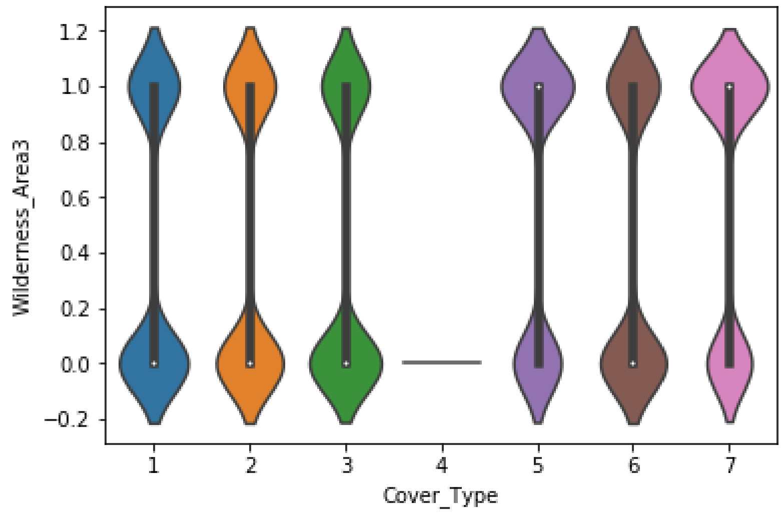


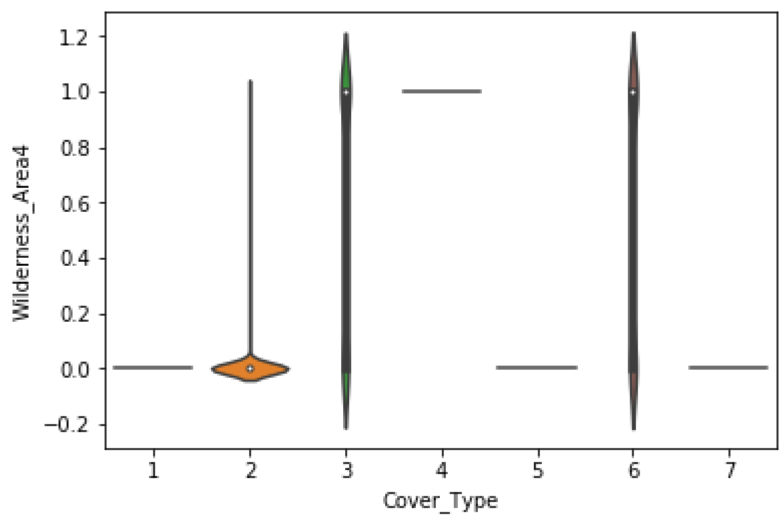
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

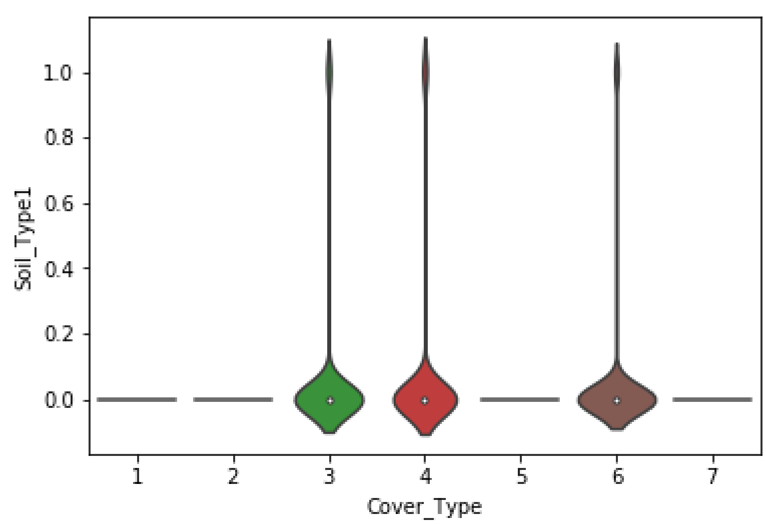

Horizontal_Distance_To_Fire_Points
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
6
7
Cover_Type


Wilderness_Area1
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
1
2
3
4
5
6
7
Cover_Type

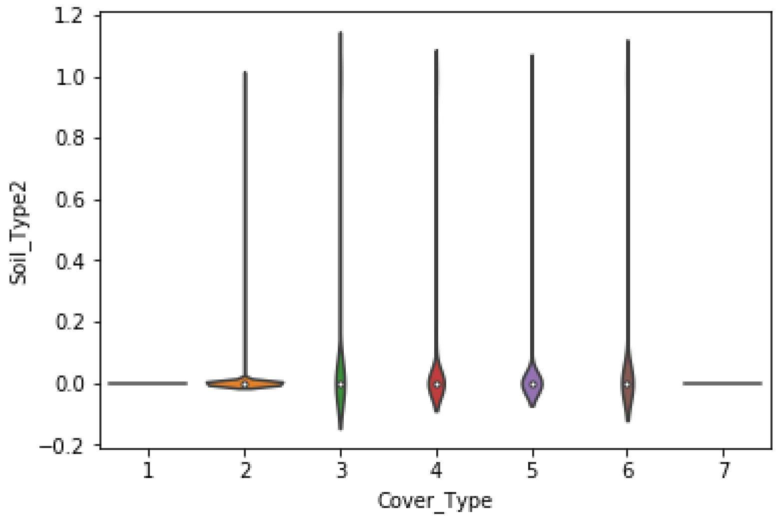

Wilderness_Area2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
1
2
3
4
5
6
7
Cover_Type

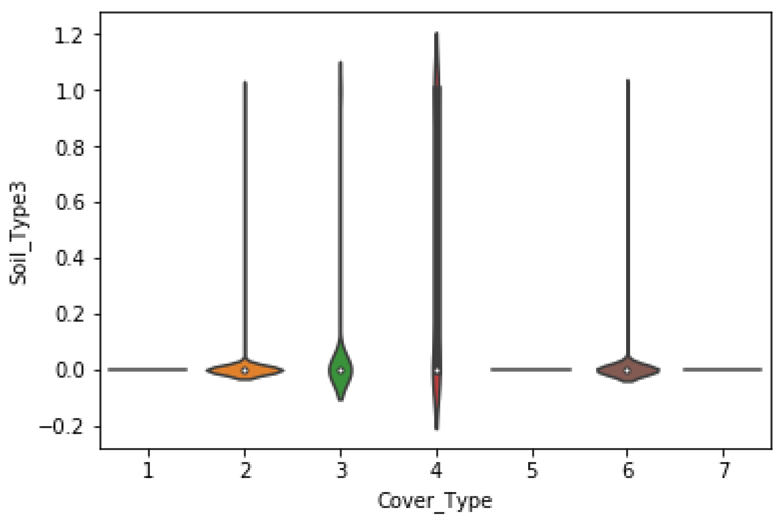


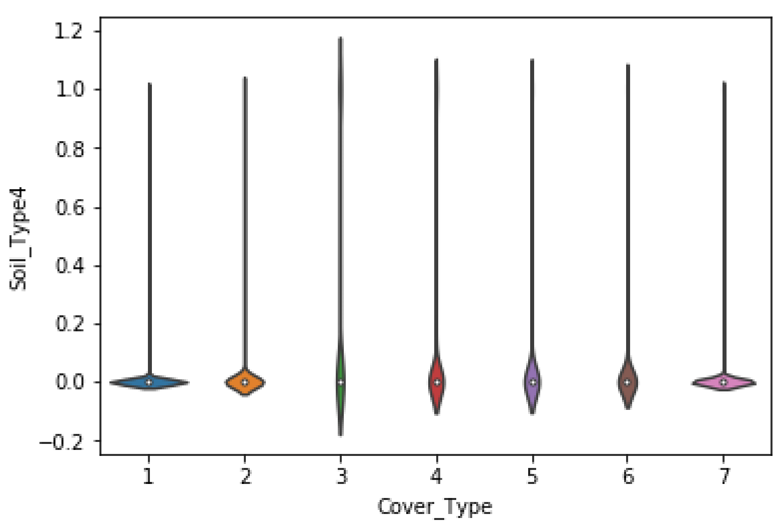




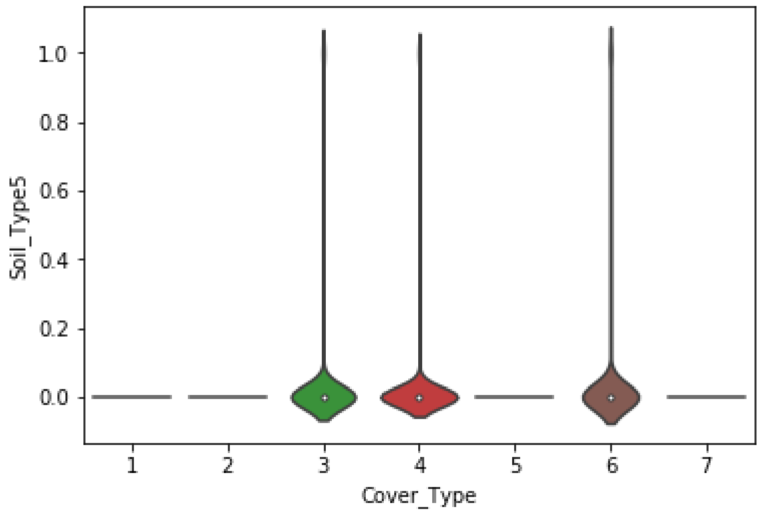
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

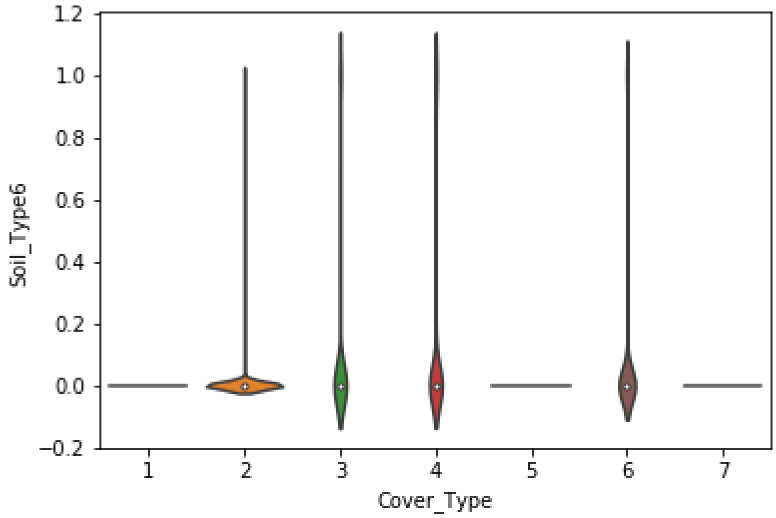


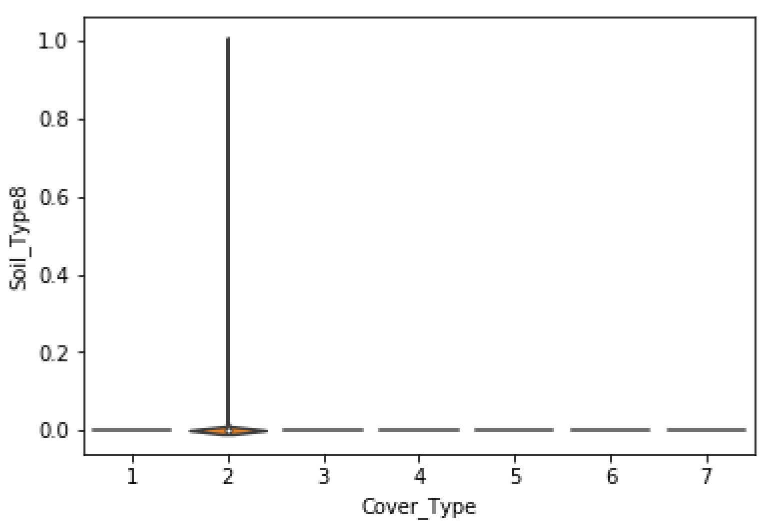




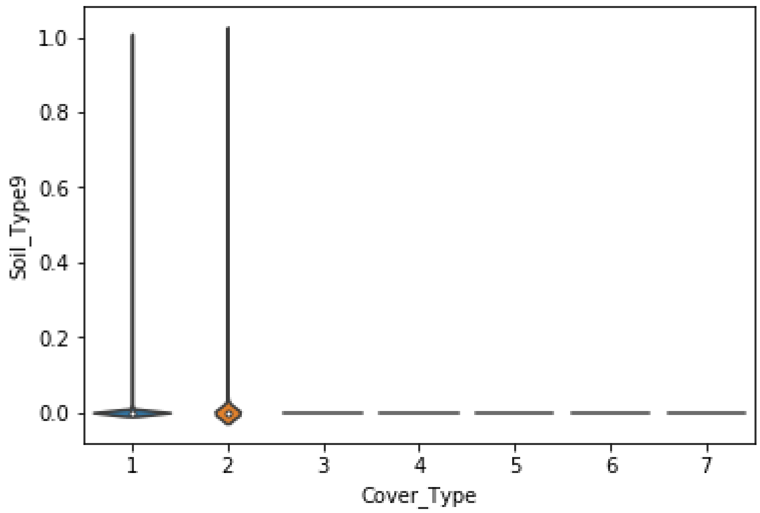
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

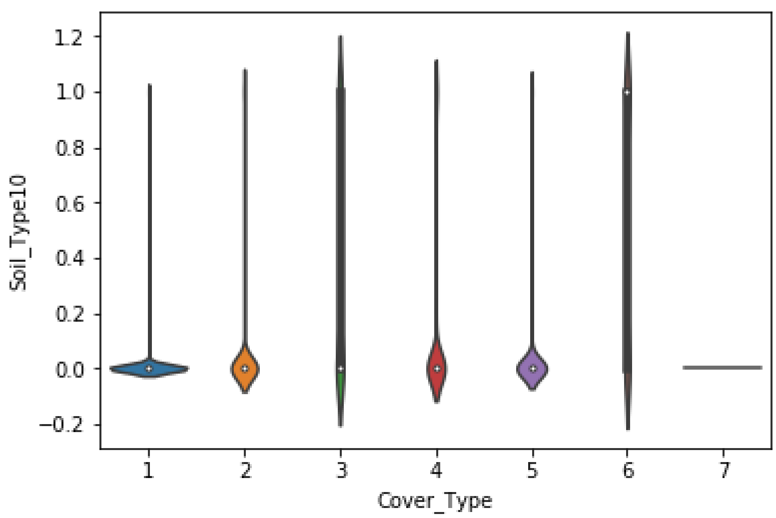


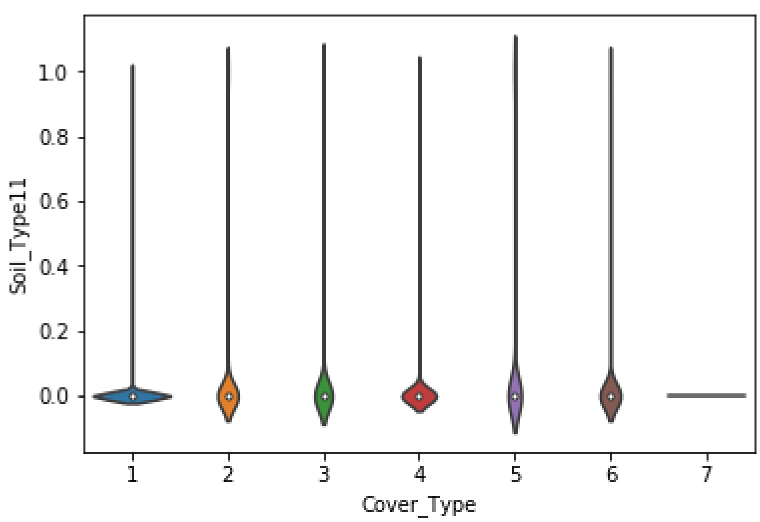




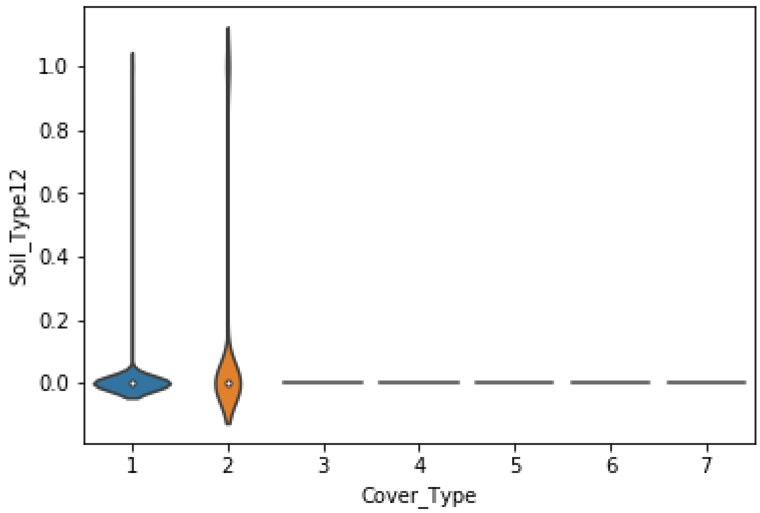
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

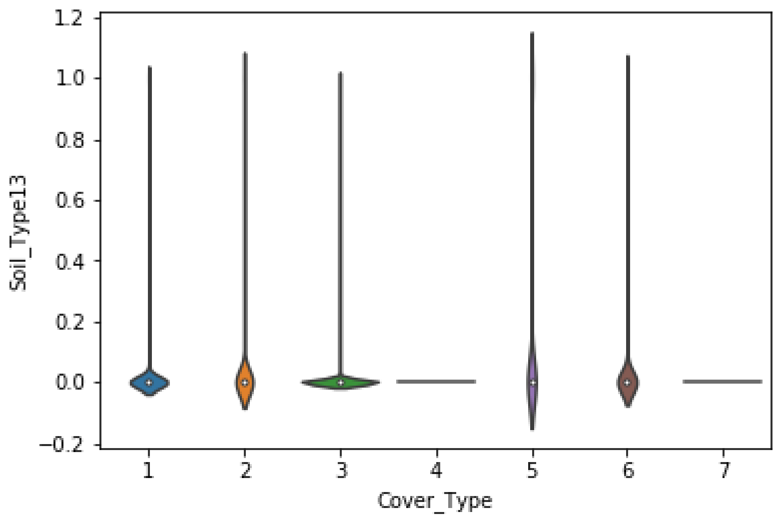


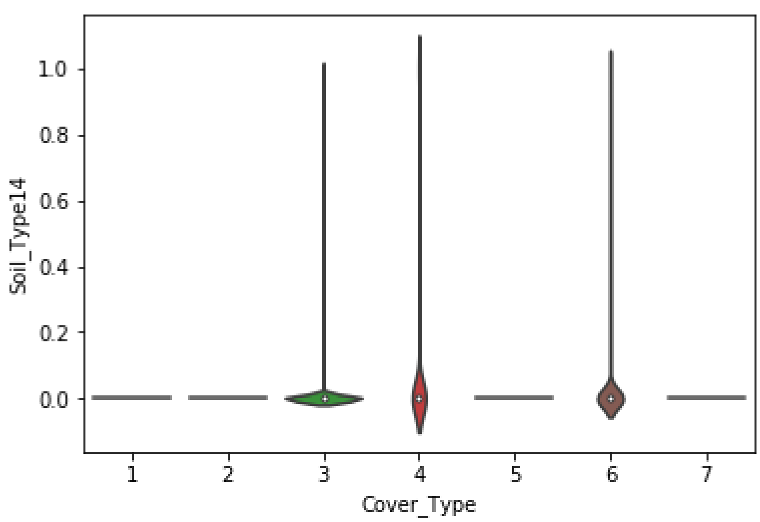




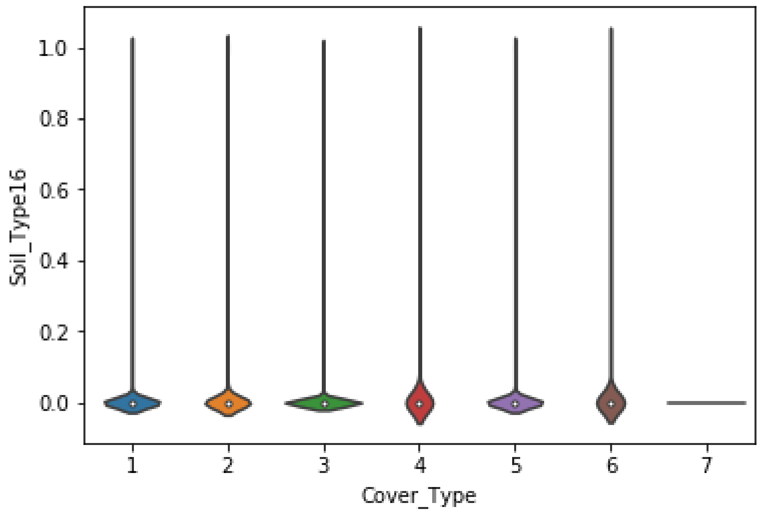
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

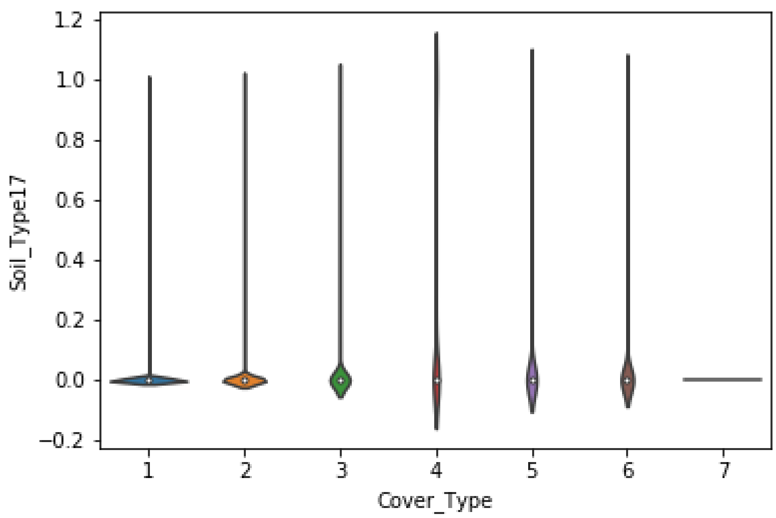


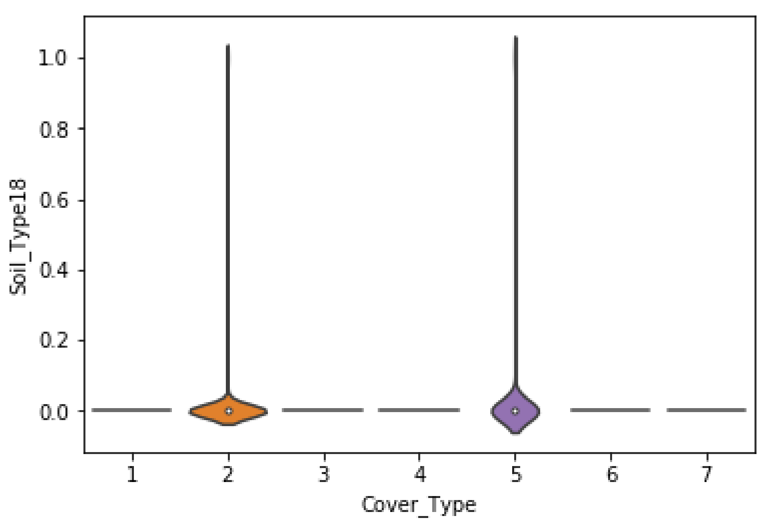




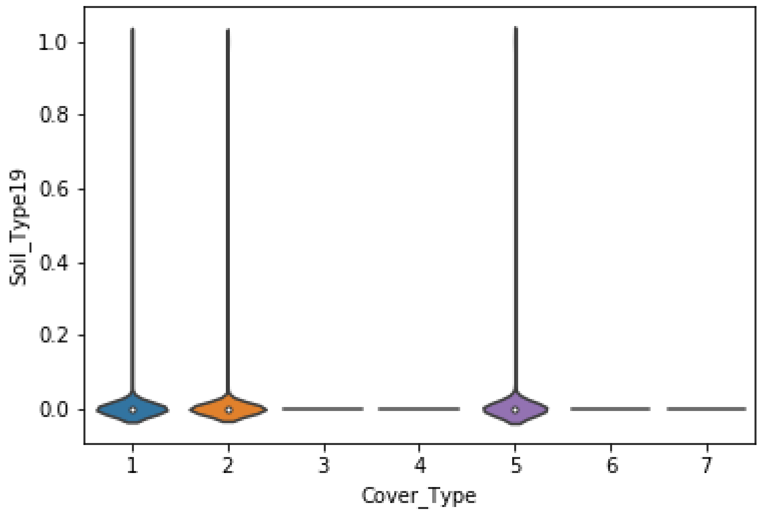
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

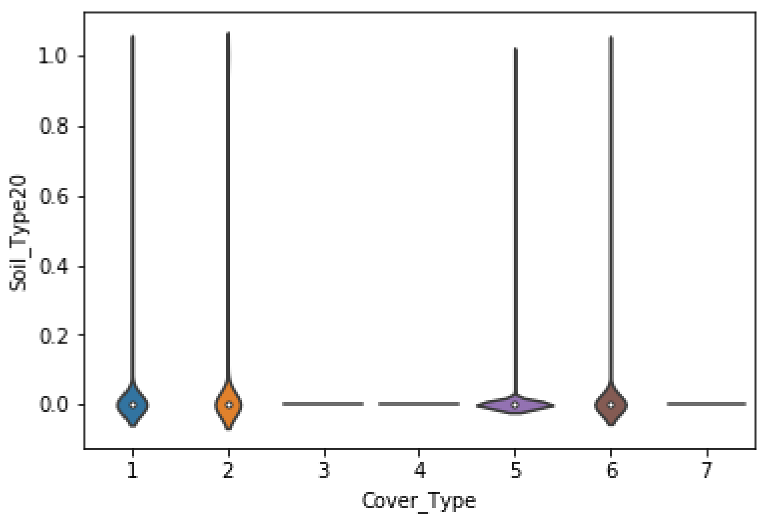


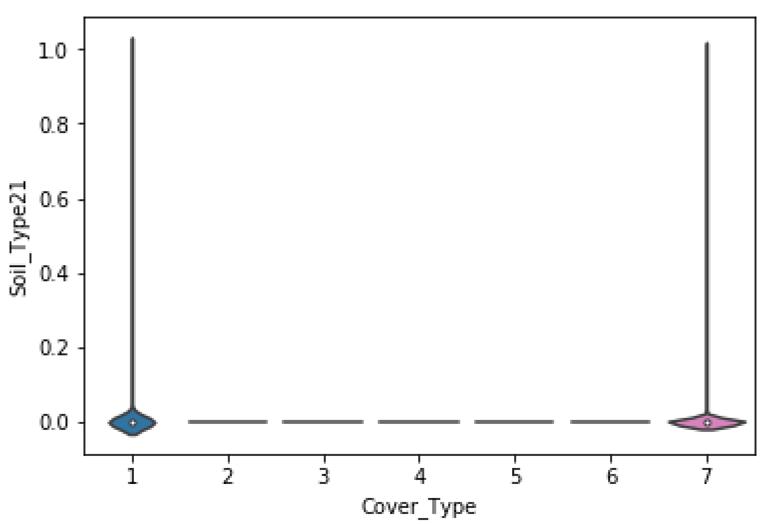




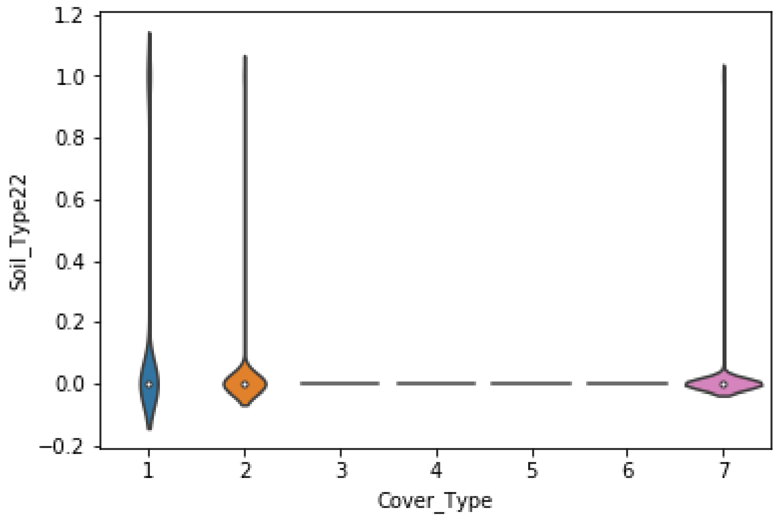
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

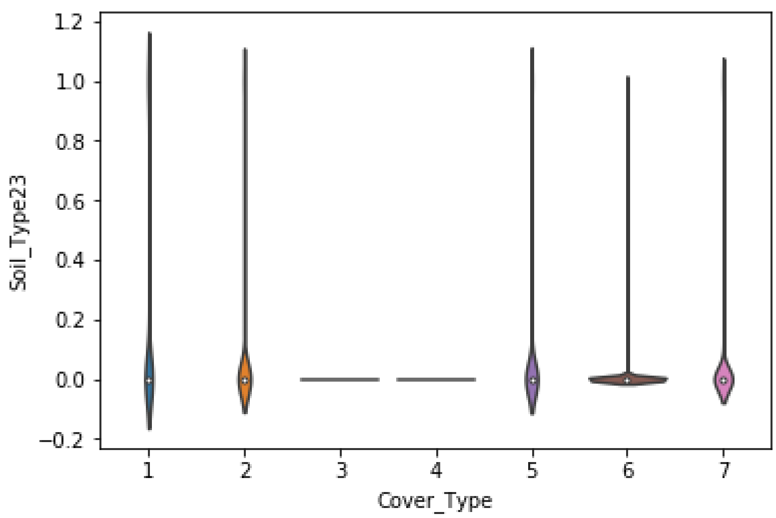


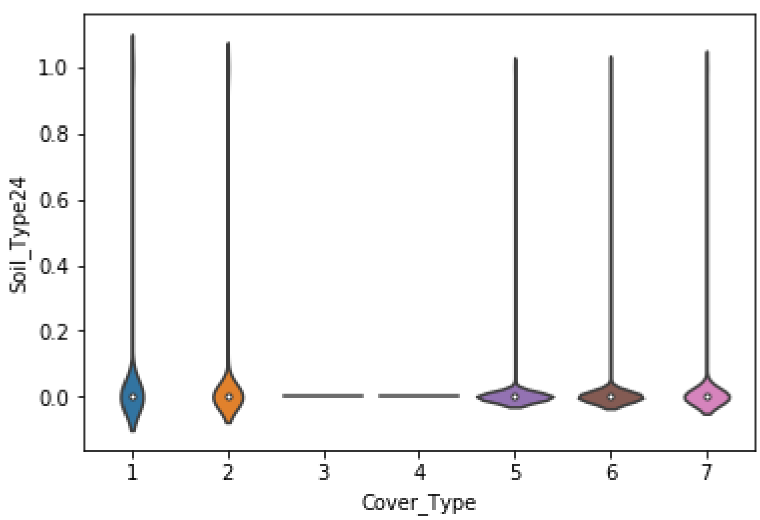




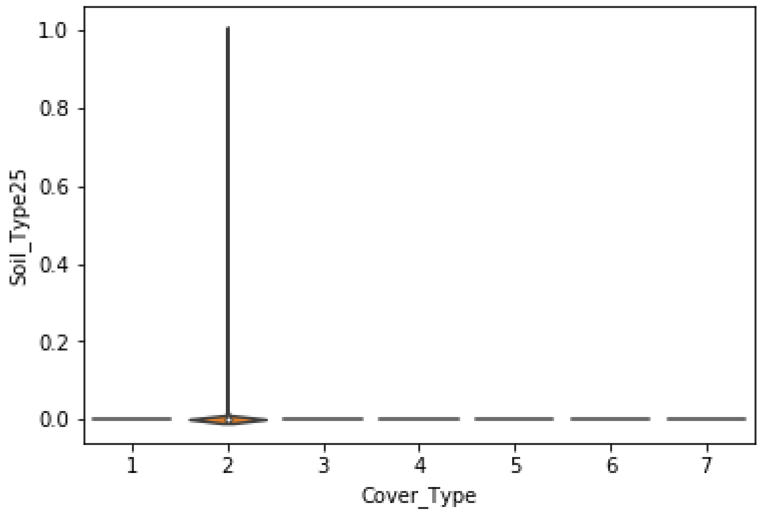
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

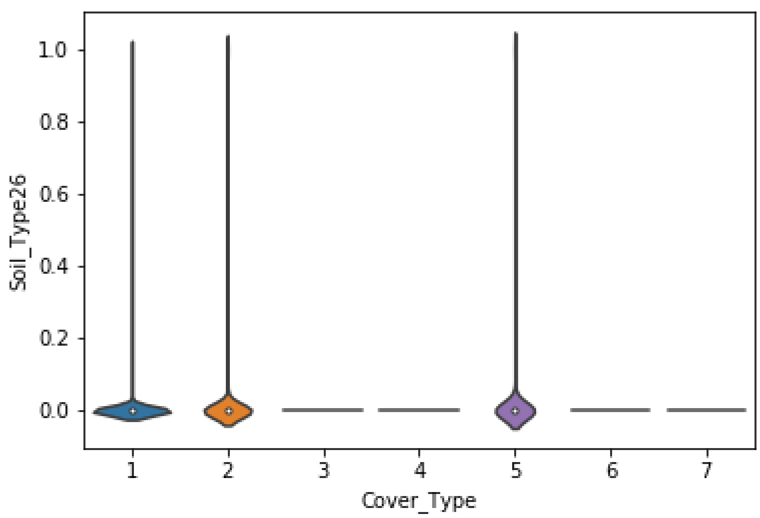


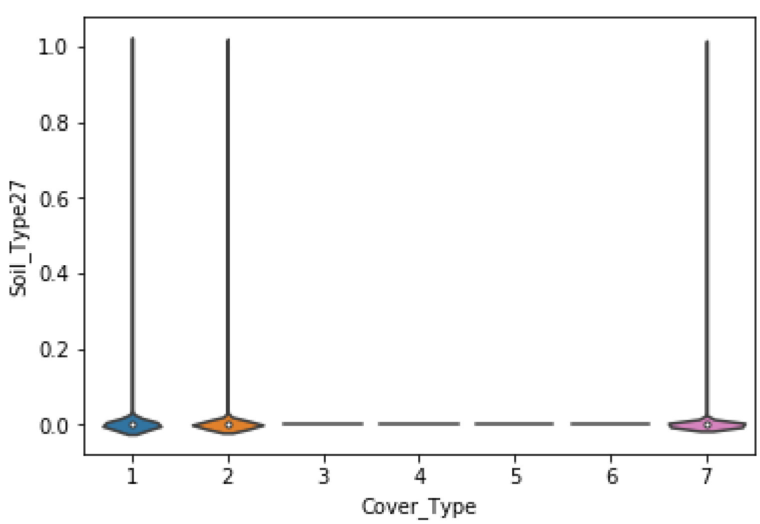




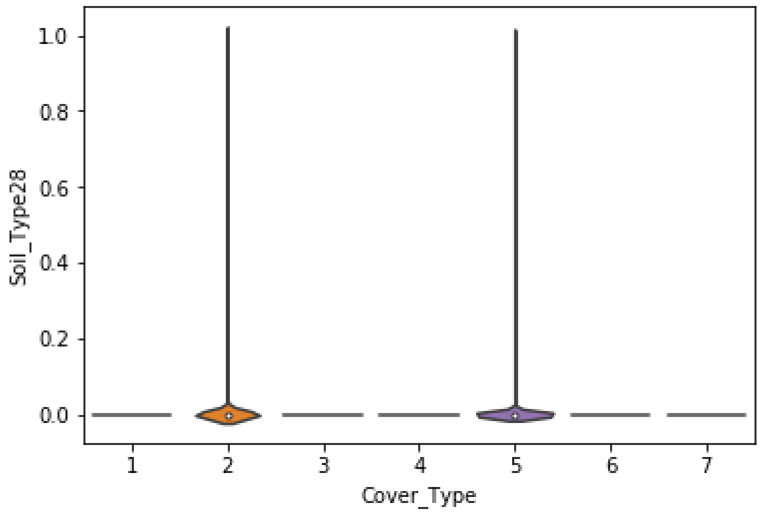
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

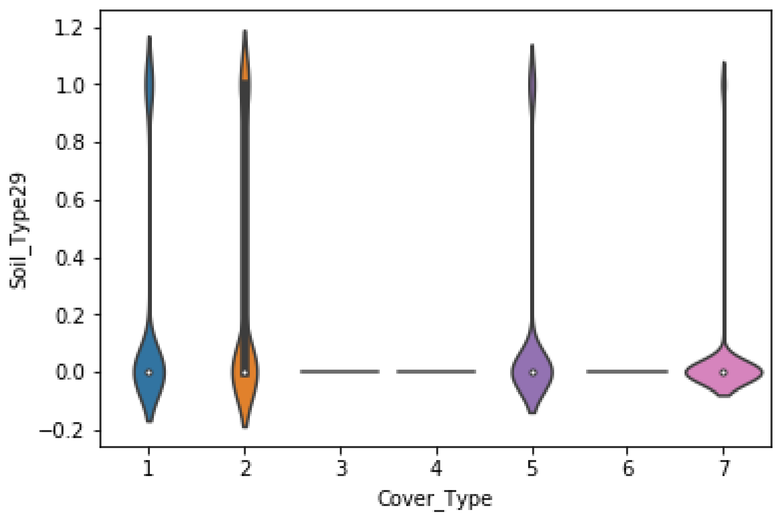


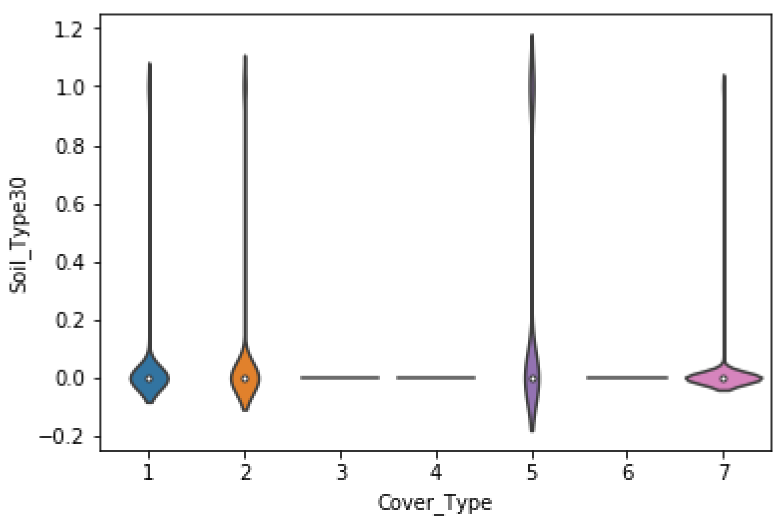




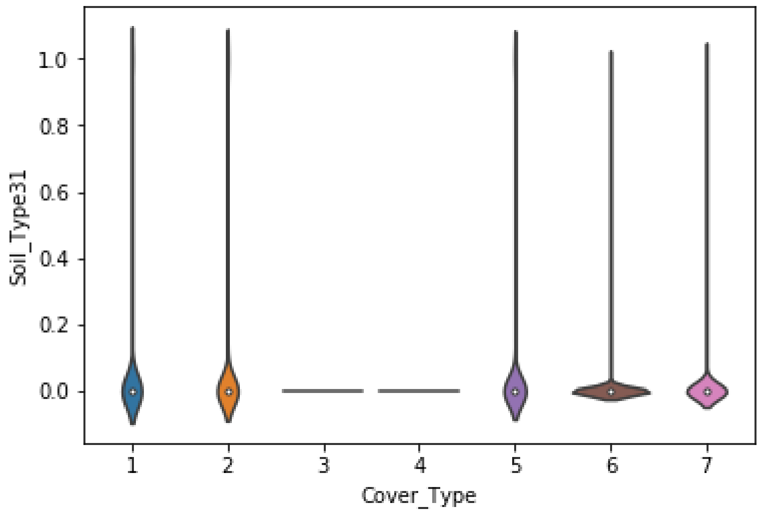
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

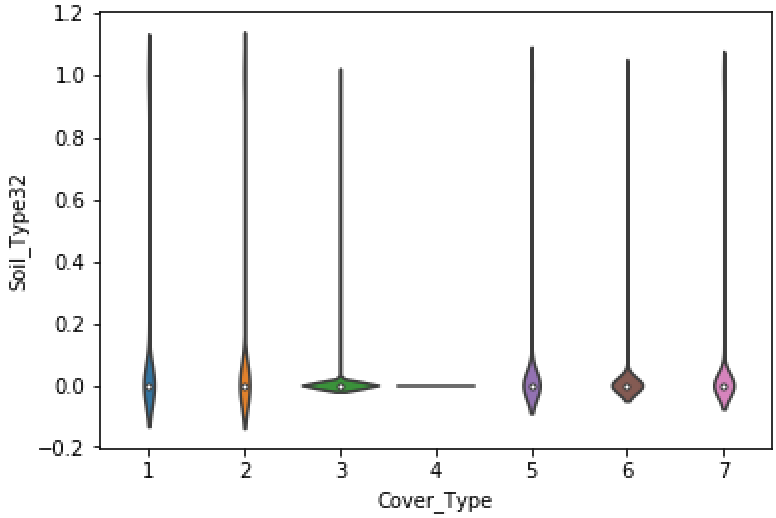


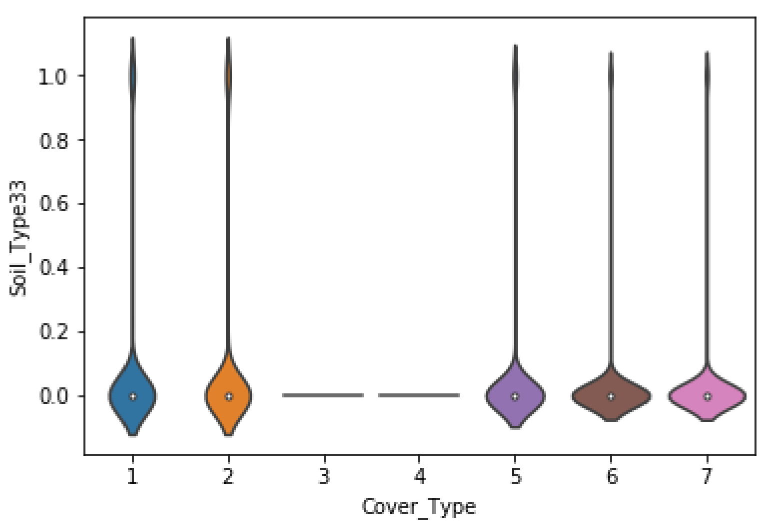




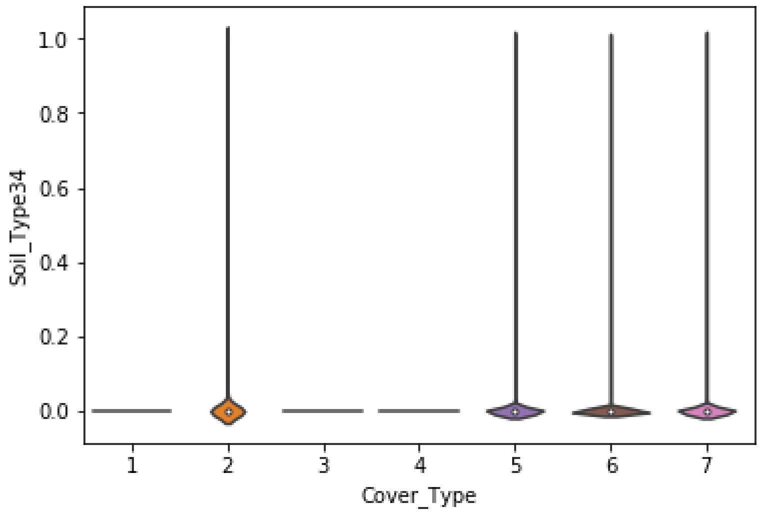
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

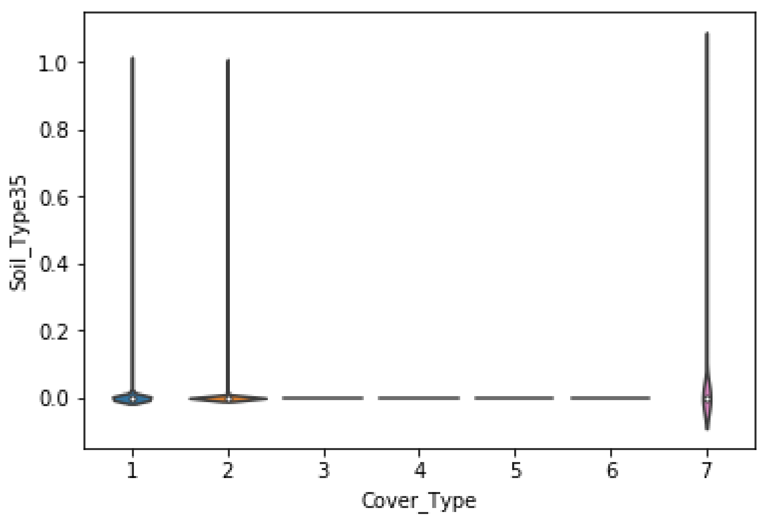


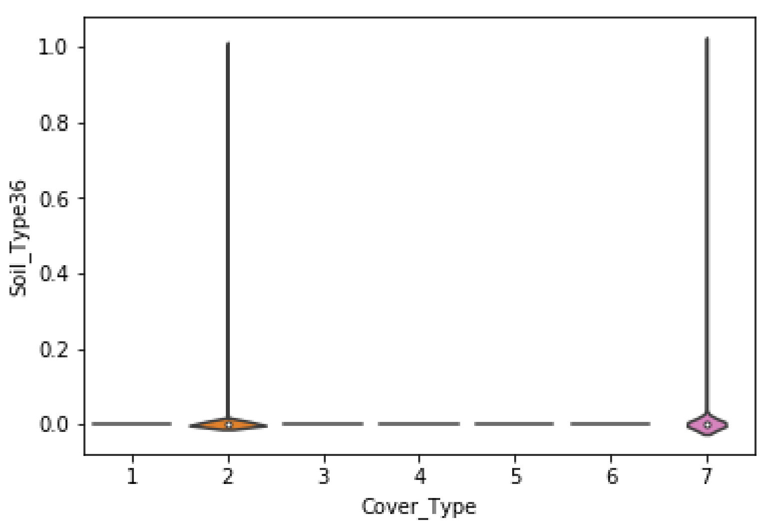




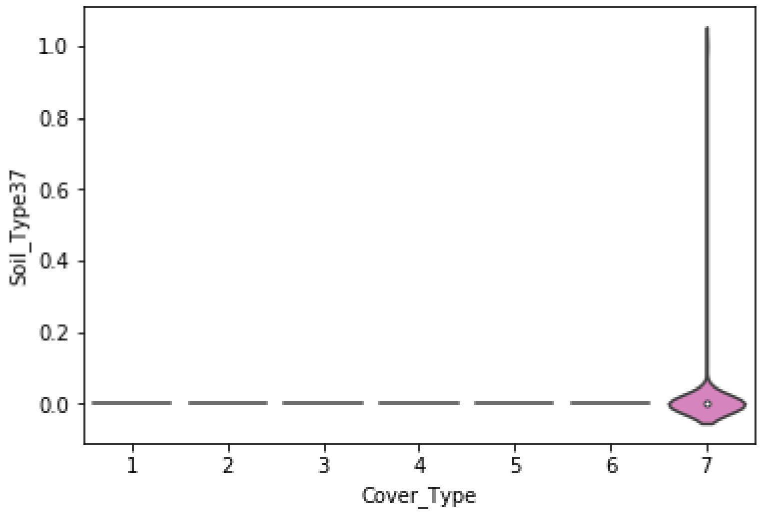
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

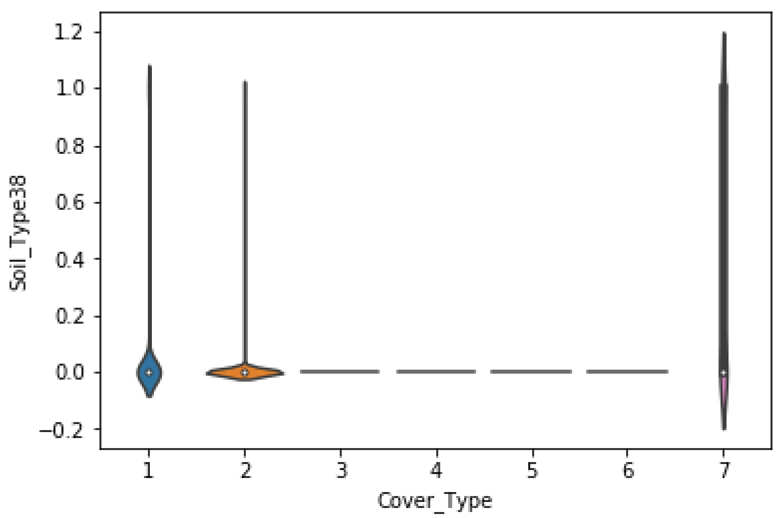


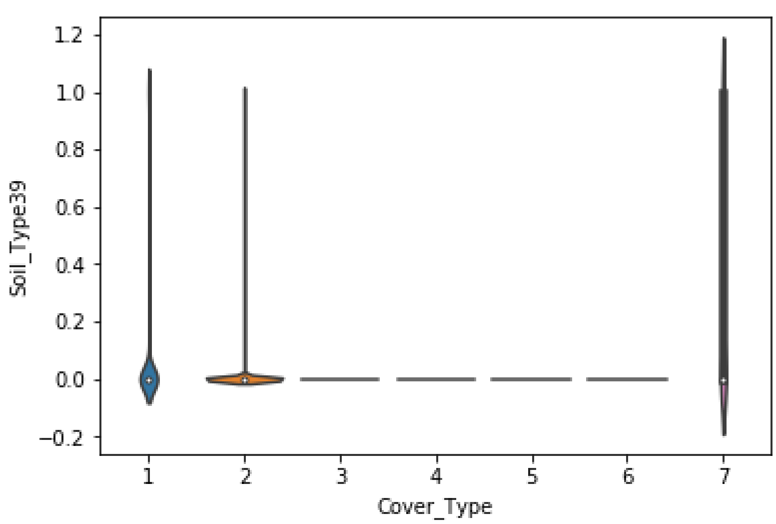




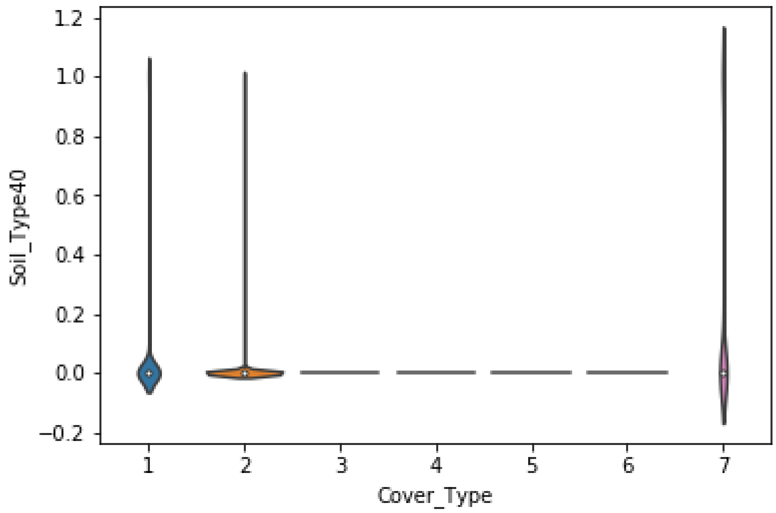
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js







File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js



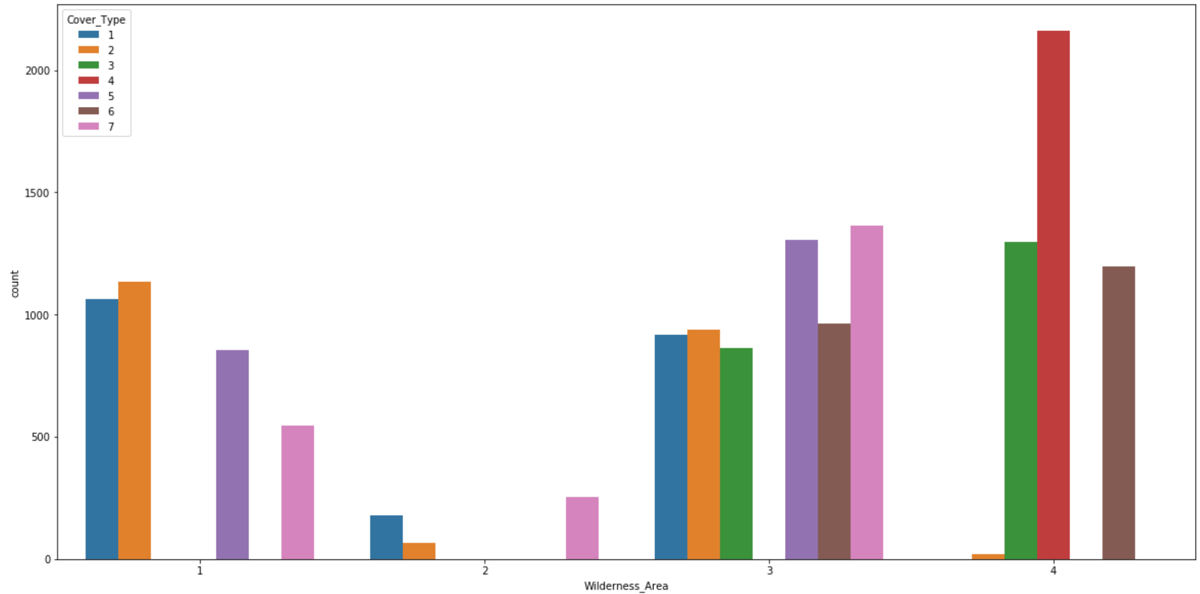
Data Visualization

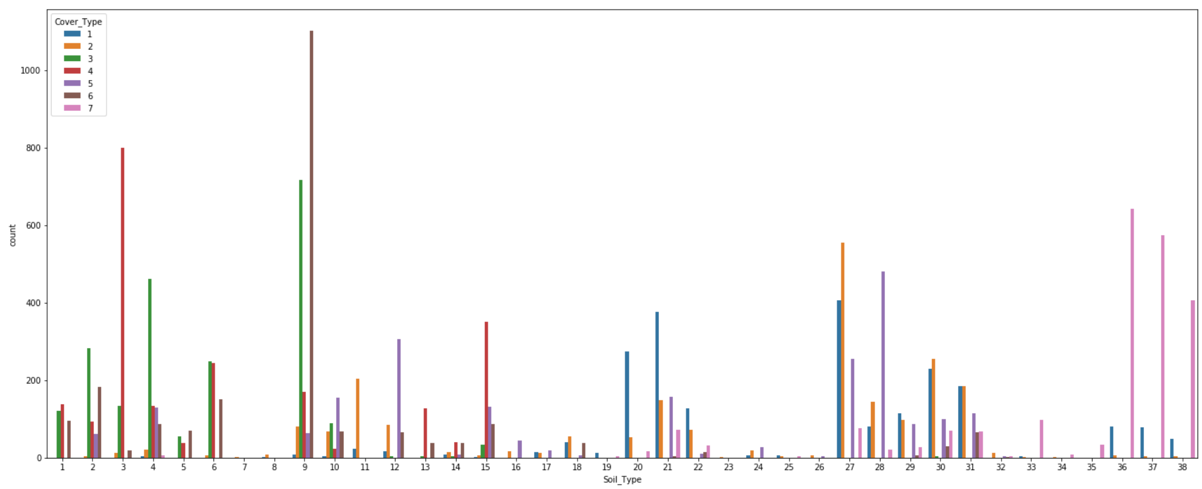
In [211]:
```
# Group one-hot encoded variables of a category into one single variable

#names of all the columns
cols = dataset.columns
#number of rows=r , number of columns=c
r,c = dataset.shape
#Create a new dataframe with r rows, one column for each encoded categor
y, and target in the end
dataS = pd.DataFrame(index=np.arange(0, r),columns=['Wilderness_Area','S
oil_Type','Cover_Type'])
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [212]:
```
#Make an entry in 'data' for each r as category_id, target value
for i in range(0,r):
    w=0;
    s=0;
   # Category1 range
    for j in range(10,14):
        if (dataset.iloc[i,j] == 1):
            w=j-9
            break
    # Category2 range
    for k in range(14,54):
        if (dataset.iloc[i,k] == 1):
            s=k-13
            break
     #Make an entry in 'data' for each r as category_id, target value

    dataS.iloc[i]=[w,s,dataset.iloc[i,c-1]]

#Plot for Category1
sns.countplot(x="Wilderness_Area", hue="Cover_Type", data=dataS)
plt.show()
#Plot for Category2
plt.rc("figure", figsize=(25, 10))
sns.countplot(x="Soil_Type", hue="Cover_Type", data=dataS)
plt.show()

#WildernessArea_4 has a lot of presence for cover_type 4. Good class dis
tinction
#WildernessArea_3 has not much class distinction
#SoilType 1-6,10-14,17, 22-23, 29-33,35,38-40 offer lot of class distinc
tion as counts for some are very high
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js





Data Standardization

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [192]:
```
#get the number of rows and columns
r, c = dataset.shape

#get the list of columns
cols = dataset.columns
#create an array which has indexes of columns
i_cols = []
for i in range(0,c-1):
    i_cols.append(i)
#array of importance rank of all features
ranks = []

#Extract only the values
array = dataset.values

#Y is the target column, X has the rest
X_orig = array[:,0:(c-1)]
Y = array[:,(c-1):c]

#Validation chunk size
val_size = 0.1

#Use a common seed in all experiments so that same chunk is used for val
idation
seed = 0

#Split the data into chunks
from sklearn import cross_validation
X_train, X_val, Y_train, Y_val = cross_validation.train_test_split(X_ori
g, Y, test_size=val_size, random_state=seed)

#Import libraries for data transformations
from sklearn.preprocessing import Imputer
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import Normalizer

#All features
X_all = []
#Additionally we will make a list of subsets
X_all_add =[]

#columns to be dropped
rem_cols = []
#indexes of columns to be dropped
i_rem = []

#Add this version of X to the list
X_all.append(['Orig','All', X_train,X_val,1.0,cols[:c-1],rem_cols,ranks,
i_cols,i_rem])

#point where categorical data begins
size=10

import numpy
```

```
#Standardized
#Apply transform only for non-categorical data
X_temp = StandardScaler().fit_transform(X_train[:,0:size])
X_val_temp = StandardScaler().fit_transform(X_val[:,0:size])
#Concatenate non-categorical data and categorical
X_con = numpy.concatenate((X_temp,X_train[:,size:]),axis=1)
X_val_con = numpy.concatenate((X_val_temp,X_val[:,size:]),axis=1)
#Add this version of X to the list
X_all.append(['StdSca','All', X_con,X_val_con,1.0,cols,rem_cols,ranks,i_
cols,i_rem])

#MinMax
#Apply transform only for non-categorical data
X_temp = MinMaxScaler().fit_transform(X_train[:,0:size])
X_val_temp = MinMaxScaler().fit_transform(X_val[:,0:size])
#Concatenate non-categorical data and categorical
X_con = numpy.concatenate((X_temp,X_train[:,size:]),axis=1)
X_val_con = numpy.concatenate((X_val_temp,X_val[:,size:]),axis=1)
#Add this version of X to the list
X_all.append(['MinMax', 'All', X_con,X_val_con,1.0,cols,rem_cols,ranks,i
_cols,i_rem])

#Normalize
#Apply transform only for non-categorical data
X_temp = Normalizer().fit_transform(X_train[:,0:size])
X_val_temp = Normalizer().fit_transform(X_val[:,0:size])
#Concatenate non-categorical data and categorical
X_con = numpy.concatenate((X_temp,X_train[:,size:]),axis=1)
X_val_con = numpy.concatenate((X_val_temp,X_val[:,size:]),axis=1)
#Add this version of X to the list
X_all.append(['Norm', 'All', X_con,X_val_con,1.0,cols,rem_cols,ranks,i_c
ols,i_rem])

#Impute
#Imputer is not used as no data is missing

#List of transformations
trans_list = []

for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_rem_
list in X_all:
    trans_list.append(trans)
```

Feature Selection

In [215]:

```
#Select top 75%,50%,25%
ratio_list = [0.75,0.50,0.25]
```

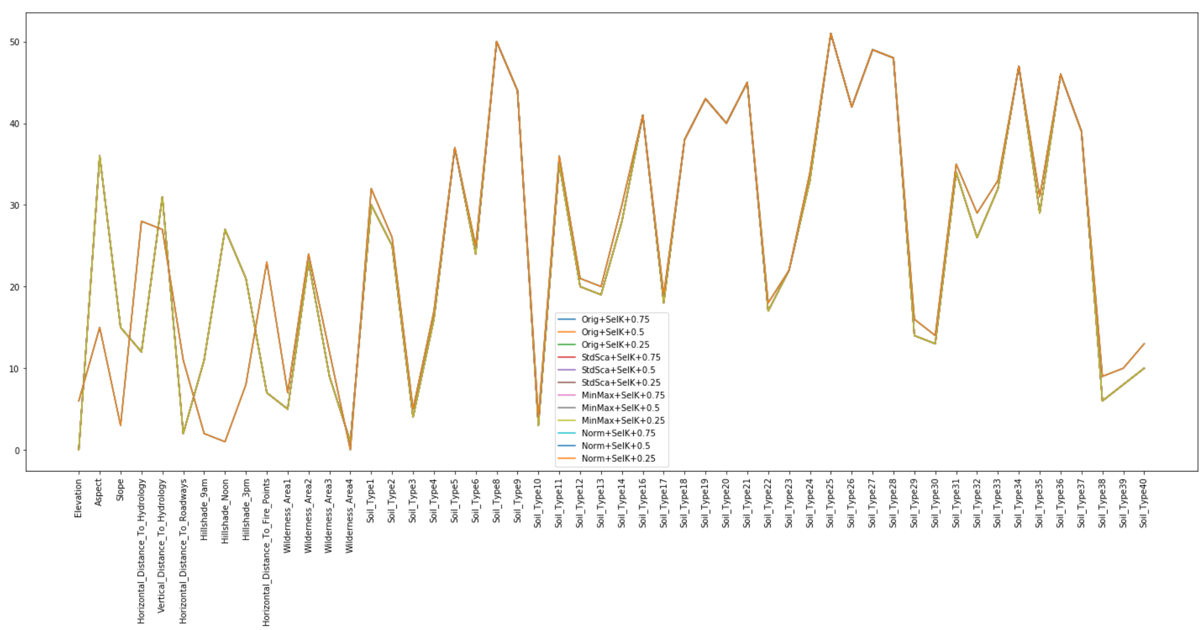
Feature Selection =- SelectPercentile

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [216]:
```
#List of feature selection models
feat = []

#List of names of feature selection models
feat_list =[]

#Libraries for SelectPercentile
from sklearn.feature_selection import SelectPercentile
from sklearn.feature_selection import f_classif

n = 'SelK'
feat_list.append(n)
for val in ratio_list:
    comb.append("%s+%s" % (n,val))
    feat.append([n,val,SelectPercentile(score_func=f_classif,percentile=
val*100)])


#For all transformations of X
for trans,s, X, X_val, d, cols, rem, ra, i_cols, i_rem in X_all:
    #For all feature selection models
    for name,v, model in feat:
        #Train the model against Y
        model.fit(X,Y_train)
        #Combine importance and index of the column in the array joined
        joined = []
        for i, pred in enumerate(list(model.scores_)):
            joined.append([i,cols[i],pred])
        #Sort in descending order
        joined_sorted = sorted(joined, key=lambda x: -x[2])
        #Starting point of the columns to be dropped
        rem_start = int((v*(c-1)))
        #List of names of columns selected
        cols_list = []
        #Indexes of columns selected
        i_cols_list = []
        #Ranking of all the columns
        rank_list =[]
        #List of columns not selected
        rem_list = []
        #Indexes of columns not selected
        i_rem_list = []
        #Split the array. Store selected columns in cols_list and remove
d in rem_list
        for j, (i, col, x) in enumerate(list(joined_sorted)):
            #Store the rank
            rank_list.append([i,j])
            #Store selected columns in cols_list and indexes in i_cols_l
ist
            if(j < rem_start):
                cols_list.append(col)
                i_cols_list.append(i)
            #Store not selected columns in rem_list and indexes in i_rem
_list
            else:
                rem_list.append(col)
```

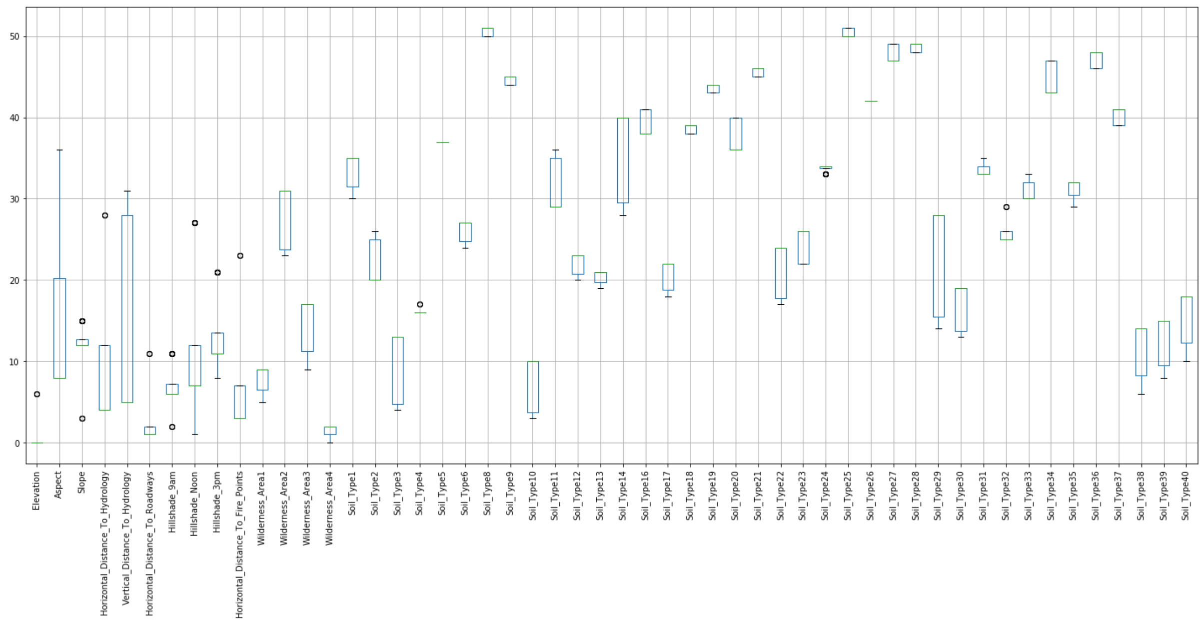
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
                i_rem_list.append(i)
        #Sort the rank_list and store only the ranks. Drop the index
        #Append model name, array, columns selected and columns to be re
moved to the additional list
        X_all_add.append([trans,name,X,X_val,v,cols_list,rem_list,[x[1]
for x in sorted(rank_list,key=lambda x:x[0])],i_cols_list,i_rem_list])



#Set figure size
plt.rc("figure", figsize=(25, 10))

#Plot a graph for different feature selectors
for f_name in feat_list:
    #Array to store the list of combinations
    leg=[]
    fig, ax = plt.subplots()
    #Plot each combination
    for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_
rem_list in X_all_add:
        if(name==f_name):
            plt.plot(rank_list)
            leg.append(trans+"+"+name+"+%s"% v)
    #Set the tick names to names of columns
    ax.set_xticks(range(c-1))
    ax.set_xticklabels(cols[:c-1],rotation='vertical')
    #Display the plot
    plt.legend(leg,loc='best')
    #Plot the rankings of all the features for all combinations
    plt.show()
```



File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [217]:
```
rank_df = pd.DataFrame(data=[x[7] for x in X_all_add],columns=cols[:c-1
])
_ = rank_df.boxplot(rot=90)
#Below plot summarizes the rankings according to the standard feature se
lection techniques
#Top ranked attributes are ... first 10 attributes, Wilderness_Area1,4
 ...Soil_Type 3,4,10,38-40
```



Rank Features based on Median

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [218]:
```
rank_df = pd.DataFrame(data=[x[7] for x in X_all_add],columns=cols[:c-1
])
med = rank_df.median()
print(med)
```

```
Elevation                              0.0
Aspect                                 8.0
Slope                                 12.0
Horizontal_Distance_To_Hydrology       4.0
Vertical_Distance_To_Hydrology         5.0
Horizontal_Distance_To_Roadways        1.0
Hillshade_9am                          6.0
Hillshade_Noon                         7.0
Hillshade_3pm                         11.0
Horizontal_Distance_To_Fire_Points     3.0
Wilderness_Area1                       9.0
Wilderness_Area2                      31.0
Wilderness_Area3                      17.0
Wilderness_Area4                       2.0
Soil_Type1                            35.0
Soil_Type2                            20.0
Soil_Type3                            13.0
Soil_Type4                            16.0
Soil_Type5                            37.0
Soil_Type6                            27.0
Soil_Type8                            51.0
Soil_Type9                            45.0
Soil_Type10                           10.0
Soil_Type11                           29.0
Soil_Type12                           23.0
Soil_Type13                           21.0
Soil_Type14                           40.0
Soil_Type16                           38.0
Soil_Type17                           22.0
Soil_Type18                           39.0
Soil_Type19                           44.0
Soil_Type20                           36.0
Soil_Type21                           46.0
Soil_Type22                           24.0
Soil_Type23                           26.0
Soil_Type24                           34.0
Soil_Type25                           50.0
Soil_Type26                           42.0
Soil_Type27                           47.0
Soil_Type28                           49.0
Soil_Type29                           28.0
Soil_Type30                           19.0
Soil_Type31                           33.0
Soil_Type32                           25.0
Soil_Type33                           30.0
Soil_Type34                           43.0
Soil_Type35                           32.0
Soil_Type36                           48.0
Soil_Type37                           41.0
Soil_Type38                           14.0
Soil_Type39                           15.0
Soil_Type40                           18.0
dtype: float64
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [219]:
```
#Select top 75%,50%,25%
ratio_list = [0.75,0.50,0.25]

#Median of rankings for each column
unsorted_rank = [0,8,11,4,5,2,5,7.5,9.5,3,8,28.5,14.5,2,35,19.5,12,14,37
,25.5,50,44,9,28,20.5,19.5,40,38,20,38,43,35,44,22,24,33,49,42,46,47,27.
5,19,31.5,23,28,42,30.5,46,40,12,13,18]

#List of feature selection models
feat = []

#Add Median to the list
n = 'Median'
for val in ratio_list:
    feat.append([n,val])

for trans,s, X, X_val, d, cols, rem_cols, ra, i_cols, i_rem in X_all:
    #Create subsets of feature list based on ranking and ratio_list
    for name, v in feat:
        #Combine importance and index of the column in the array joined
        joined = []
        for i, pred in enumerate(unsorted_rank):
            joined.append([i,cols[i],pred])
        #Sort in descending order
        joined_sorted = sorted(joined, key=lambda x: x[2])
        #Starting point of the columns to be dropped
        rem_start = int((v*(c-1)))
        #List of names of columns selected
        cols_list = []
        #Indexes of columns selected
        i_cols_list = []
        #Ranking of all the columns
        rank_list =[]
        #List of columns not selected
        rem_list = []
        #Indexes of columns not selected
        i_rem_list = []
        #Split the array. Store selected columns in cols_list and remove
d in rem_list
        for j, (i, col, x) in enumerate(list(joined_sorted)):
            #Store the rank
            rank_list.append([i,j])
            #Store selected columns in cols_list and indexes in i_cols_l
ist
            if(j < rem_start):
                cols_list.append(col)
                i_cols_list.append(i)
            #Store not selected columns in rem_list and indexes in i_rem
_list
            else:
                rem_list.append(col)
                i_rem_list.append(i)
        #Sort the rank_list and store only the ranks. Drop the index
        #Append model name, array, columns selected and columns to be re
moved to the additional list
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
        X_all_add.append([trans,name,X,X_val,v,cols_list,rem_list,[x[1]
for x in sorted(rank_list,key=lambda x:x[0])],i_cols_list,i_rem_list])
```

In [220]:
```
#Import plotting library
import matplotlib.pyplot as plt

#Dictionary to store the accuracies for all combinations
acc = {}

#List of combinations
comb = []

#Append name of transformation to trans_list
for trans in trans_list:
    acc[trans]=[]
```

Machine Learning Algorithms

K Nearest Neighbours

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

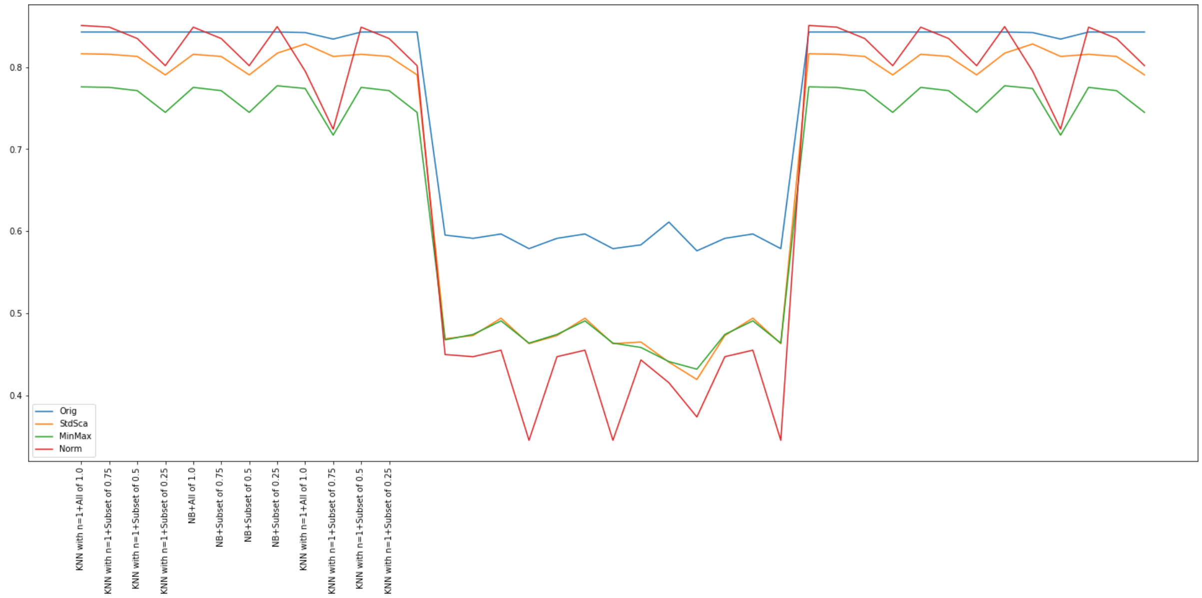
In [224]:
```
#Evaluation of various combinations of KNN Classifier using all the view
s

#Import the library
from sklearn.neighbors import KNeighborsClassifier

n_list = [1]

for n_neighbors in n_list:
    #Set the base model
    model = KNeighborsClassifier(n_jobs=-1,n_neighbors=n_neighbors)

    algo = "KNN"

    ##Set figure size
    #plt.rc("figure", figsize=(25, 10))

    #Accuracy of the model using all features
    for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_
rem_list in X_all:
        model.fit(X[:,i_cols_list],Y_train)
        result = model.score(X_val[:,i_cols_list], Y_val)
        acc[trans].append(result)
        print(trans+"+"+name+"+%d" % (v*(c-1)))
        print(result)
    comb.append("%s with n=%s+%s of %s" % (algo,n_neighbors,"All",1.0))

    #Accuracy of the model using a subset of features
    for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_
rem_list in X_all_add:
        model.fit(X[:,i_cols_list],Y_train)
        result = model.score(X_val[:,i_cols_list], Y_val)
        acc[trans].append(result)
        print(trans+"+"+name+"+%d" % (v*(c-1)))
        print(result)
    for v in ratio_list:
        comb.append("%s with n=%s+%s of %s" % (algo,n_neighbors,"Subset"
,v))

#print(acc)

##Plot the accuracies of all combinations
fig, ax = plt.subplots()
##Plot each transformation
for trans in trans_list:
        plt.plot(acc[trans])
##Set the tick names to names of combinations
ax.set_xticks(range(len(comb)))
ax.set_xticklabels(comb,rotation='vertical')
##Display the plot
plt.legend(trans_list,loc='best')
##Plot the accuracy for all combinations
plt.show()

#Best estimated performance is close to 85% when n_neighbors=1 and norma
lizer is used
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
Orig+All+52
0.842592592593
StdSca+All+52
0.816137566138
MinMax+All+52
0.775793650794
Norm+All+52
0.850529100529
Orig+Median+39
0.842592592593
Orig+Median+26
0.842592592593
Orig+Median+13
0.842592592593
StdSca+Median+39
0.815476190476
StdSca+Median+26
0.812830687831
StdSca+Median+13
0.790343915344
MinMax+Median+39
0.775132275132
MinMax+Median+26
0.771164021164
MinMax+Median+13
0.744708994709
Norm+Median+39
0.848544973545
Norm+Median+26
0.834656084656
Norm+Median+13
0.801587301587
Orig+Median+39
0.842592592593
Orig+Median+26
0.842592592593
Orig+Median+13
0.842592592593
StdSca+Median+39
0.815476190476
StdSca+Median+26
0.812830687831
StdSca+Median+13
0.790343915344
MinMax+Median+39
0.775132275132
MinMax+Median+26
0.771164021164
MinMax+Median+13
0.744708994709
Norm+Median+39
0.848544973545
Norm+Median+26
0.834656084656
Norm+Median+13
0.801587301587
Orig+SelK+39
```

```
0.842592592593
Orig+SelK+26
0.841931216931
Orig+SelK+13
0.833994708995
StdSca+SelK+39
0.816798941799
StdSca+SelK+26
0.828042328042
StdSca+SelK+13
0.812830687831
MinMax+SelK+39
0.777116402116
MinMax+SelK+26
0.77380952381
MinMax+SelK+13
0.716931216931
Norm+SelK+39
0.849206349206
Norm+SelK+26
0.794973544974
Norm+SelK+13
0.724206349206
Orig+Median+39
0.842592592593
Orig+Median+26
0.842592592593
Orig+Median+13
0.842592592593
StdSca+Median+39
0.815476190476
StdSca+Median+26
0.812830687831
StdSca+Median+13
0.790343915344
MinMax+Median+39
0.775132275132
MinMax+Median+26
0.771164021164
MinMax+Median+13
0.744708994709
Norm+Median+39
0.848544973545
Norm+Median+26
0.834656084656
Norm+Median+13
0.801587301587
```



Naive Bayes

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

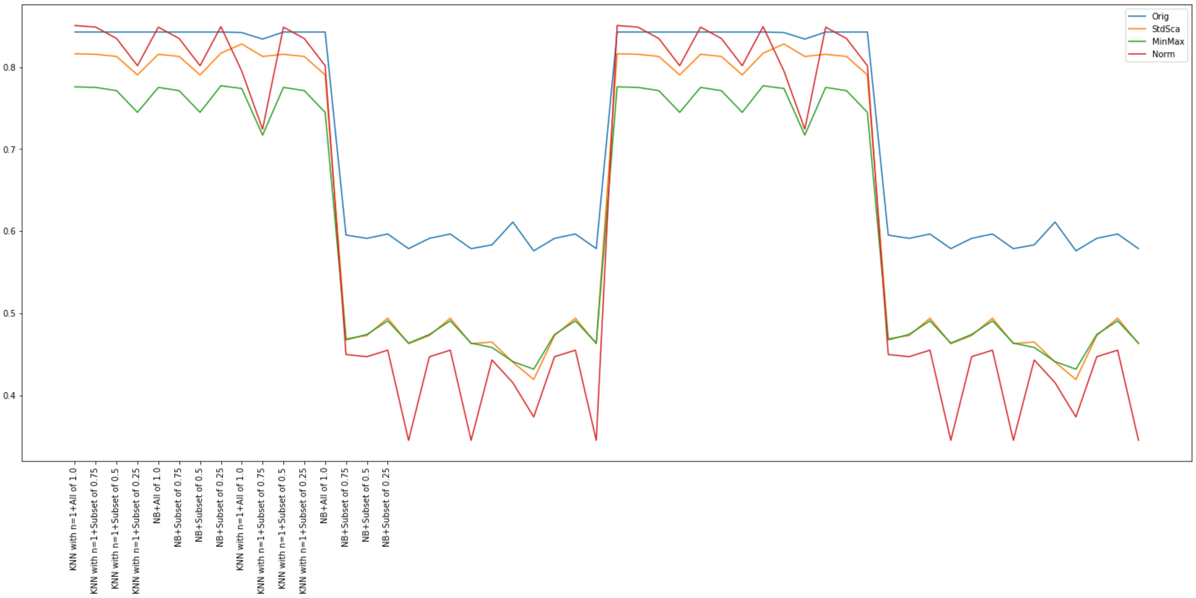
In [225]:
```
#Evaluation of various combinations of Naive Bayes using all the views

#Import the library
from sklearn.naive_bayes import GaussianNB

#Set the base model
model = GaussianNB()
algo = "NB"

##Set figure size
#plt.rc("figure", figsize=(25, 10))

#Accuracy of the model using all features
for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_rem_
list in X_all:
    model.fit(X[:,i_cols_list],Y_train)
    result = model.score(X_val[:,i_cols_list], Y_val)
    acc[trans].append(result)
    print(trans+"+"+name+"+%d" % (v*(c-1)))
    print(result)
comb.append("%s+%s of %s" % (algo,"All",1.0))

#Accuracy of the model using a subset of features
for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_rem_
list in X_all_add:
    model.fit(X[:,i_cols_list],Y_train)
    result = model.score(X_val[:,i_cols_list], Y_val)
    acc[trans].append(result)
    print(trans+"+"+name+"+%d" % (v*(c-1)))
    print(result)
for v in ratio_list:
    comb.append("%s+%s of %s" % (algo,"Subset",v))

##Plot the accuracies of all combinations
fig, ax = plt.subplots()
##Plot each transformation
for trans in trans_list:
        plt.plot(acc[trans])
##Set the tick names to names of combinations
ax.set_xticks(range(len(comb)))
ax.set_xticklabels(comb,rotation='vertical')
##Display the plot
plt.legend(trans_list,loc='best')
##Plot the accuracy for all combinations
plt.show()

#Best estimated performance is close to 61%. Original with 50% subset ou
tperfoms all transformations of NB
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
Orig+All+52
0.595238095238
StdSca+All+52
0.468915343915
MinMax+All+52
0.467592592593
Norm+All+52
0.449735449735
Orig+Median+39
0.59126984127
Orig+Median+26
0.596560846561
Orig+Median+13
0.578703703704
StdSca+Median+39
0.472883597884
StdSca+Median+26
0.494047619048
StdSca+Median+13
0.462962962963
MinMax+Median+39
0.474206349206
MinMax+Median+26
0.490740740741
MinMax+Median+13
0.463624338624
Norm+Median+39
0.44708994709
Norm+Median+26
0.455026455026
Norm+Median+13
0.345238095238
Orig+Median+39
0.59126984127
Orig+Median+26
0.596560846561
Orig+Median+13
0.578703703704
StdSca+Median+39
0.472883597884
StdSca+Median+26
0.494047619048
StdSca+Median+13
0.462962962963
MinMax+Median+39
0.474206349206
MinMax+Median+26
0.490740740741
MinMax+Median+13
0.463624338624
Norm+Median+39
0.44708994709
Norm+Median+26
0.455026455026
Norm+Median+13
0.345238095238
Orig+SelK+39
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
0.583333333333
Orig+SelK+26
0.611111111111
Orig+SelK+13
0.576058201058
StdSca+SelK+39
0.464947089947
StdSca+SelK+26
0.440476190476
StdSca+SelK+13
0.419312169312
MinMax+SelK+39
0.458333333333
MinMax+SelK+26
0.441137566138
MinMax+SelK+13
0.431878306878
Norm+SelK+39
0.443121693122
Norm+SelK+26
0.415343915344
Norm+SelK+13
0.373677248677
Orig+Median+39
0.59126984127
Orig+Median+26
0.596560846561
Orig+Median+13
0.578703703704
StdSca+Median+39
0.472883597884
StdSca+Median+26
0.494047619048
StdSca+Median+13
0.462962962963
MinMax+Median+39
0.474206349206
MinMax+Median+26
0.490740740741
MinMax+Median+13
0.463624338624
Norm+Median+39
0.44708994709
Norm+Median+26
0.455026455026
Norm+Median+13
0.345238095238
```



Random Forest

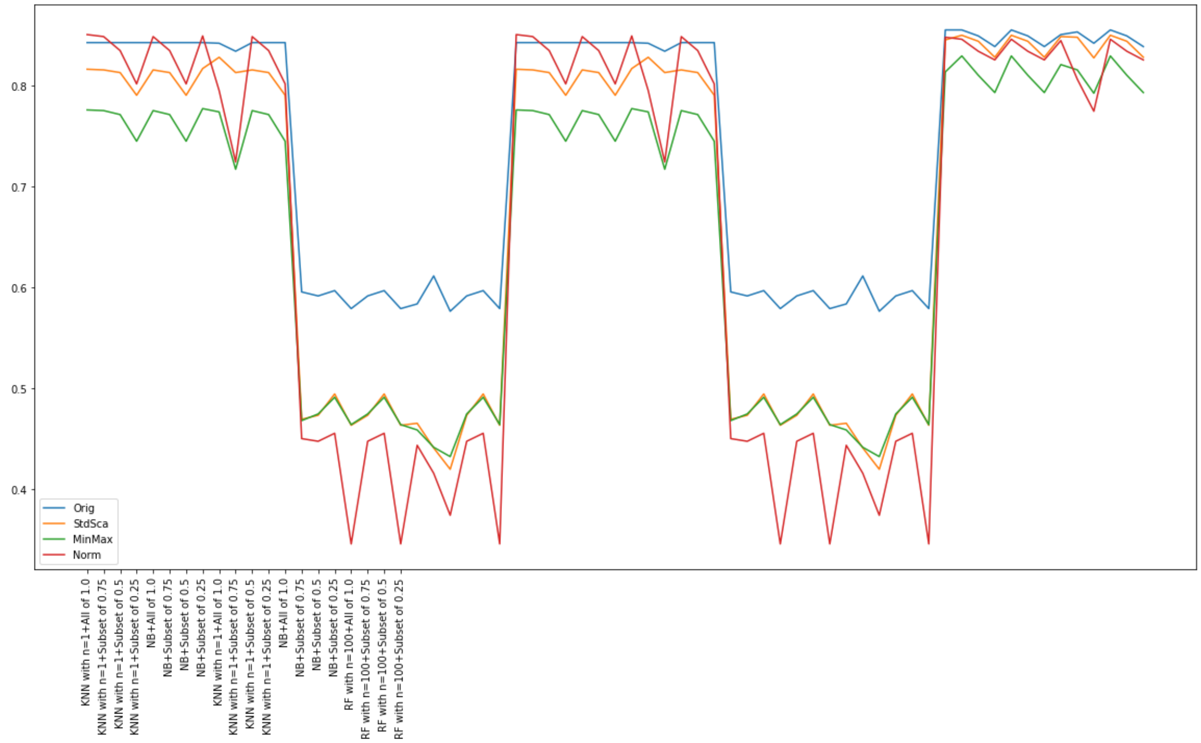
File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [226]:
```
#Evaluation of various combinations of Random Forest using all the views

#Import the library
from sklearn.ensemble import RandomForestClassifier

n_list = [100]

for n_estimators in n_list:
    #Set the base model
    model = RandomForestClassifier(n_jobs=-1,n_estimators=n_estimators,
random_state=seed)

    algo = "RF"

    #Set figure size
    plt.rc("figure", figsize=(20, 10))

    #Accuracy of the model using all features
    for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_
rem_list in X_all:
        model.fit(X[:,i_cols_list],Y_train)
        result = model.score(X_val[:,i_cols_list], Y_val)
        acc[trans].append(result)
        print(trans+"+"+name+"+%d" % (v*(c-1)))
        print(result)
    comb.append("%s with n=%s+%s of %s" % (algo,n_estimators,"All",1.0))

    #Accuracy of the model using a subset of features
    for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_
rem_list in X_all_add:
        model.fit(X[:,i_cols_list],Y_train)
        result = model.score(X_val[:,i_cols_list], Y_val)
        acc[trans].append(result)
        print(trans+"+"+name+"+%d" % (v*(c-1)))
        print(result)
    for v in ratio_list:
        comb.append("%s with n=%s+%s of %s" % (algo,n_estimators,"Subse
t",v))

##Plot the accuracies of all combinations
fig, ax = plt.subplots()
##Plot each transformation
for trans in trans_list:
        plt.plot(acc[trans])
##Set the tick names to names of combinations
ax.set_xticks(range(len(comb)))
ax.set_xticklabels(comb,rotation='vertical')
##Display the plot
plt.legend(trans_list,loc='best')
##Plot the accuracy for all combinations
plt.show()

#Best estimated performance is close to 86% when n_estimators is 100
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

```
Orig+All+52
0.855158730159
StdSca+All+52
0.845238095238
MinMax+All+52
0.813492063492
Norm+All+52
0.847883597884
Orig+Median+39
0.855158730159
Orig+Median+26
0.849206349206
Orig+Median+13
0.838624338624
StdSca+Median+39
0.849867724868
StdSca+Median+26
0.843915343915
StdSca+Median+13
0.828042328042
MinMax+Median+39
0.829365079365
MinMax+Median+26
0.810185185185
MinMax+Median+13
0.792989417989
Norm+Median+39
0.845899470899
Norm+Median+26
0.833994708995
Norm+Median+13
0.825396825397
Orig+Median+39
0.855158730159
Orig+Median+26
0.849206349206
Orig+Median+13
0.838624338624
StdSca+Median+39
0.849867724868
StdSca+Median+26
0.843915343915
StdSca+Median+13
0.828042328042
MinMax+Median+39
0.829365079365
MinMax+Median+26
0.810185185185
MinMax+Median+13
0.792989417989
Norm+Median+39
0.845899470899
Norm+Median+26
0.833994708995
Norm+Median+13
0.825396825397
Orig+SelK+39
```

```
0.850529100529
Orig+SelK+26
0.853174603175
Orig+SelK+13
0.841931216931
StdSca+SelK+39
0.848544973545
StdSca+SelK+26
0.847883597884
StdSca+SelK+13
0.827380952381
MinMax+SelK+39
0.820767195767
MinMax+SelK+26
0.815476190476
MinMax+SelK+13
0.792328042328
Norm+SelK+39
0.844576719577
Norm+SelK+26
0.806216931217
Norm+SelK+13
0.774470899471
Orig+Median+39
0.855158730159
Orig+Median+26
0.849206349206
Orig+Median+13
0.838624338624
StdSca+Median+39
0.849867724868
StdSca+Median+26
0.843915343915
StdSca+Median+13
0.828042328042
MinMax+Median+39
0.829365079365
MinMax+Median+26
0.810185185185
MinMax+Median+13
0.792989417989
Norm+Median+39
0.845899470899
Norm+Median+26
0.833994708995
Norm+Median+13
0.825396825397
```



Make Predictions

In [231]:
```
# Make predictions using Random Forest Classifier + 0.5 subset as it gav
e the best estimated performance

n_estimators = 100

#Obtain the list of indexes for the required model
indexes = []
for trans,name,X,X_val,v,cols_list,rem_list,rank_list,i_cols_list,i_rem_
list in X_all_add:
    if v == 0.5:
        if trans == 'Orig':
            indexes = i_cols_list
            break
```

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js

In [232]:
```
#Best model definition
best_model = RandomForestClassifier(n_jobs=-1,n_estimators=n_estimators)
best_model.fit(X_orig[:,indexes],Y)

#Read test dataset
dataset_test = pd.read_csv("test.csv")
#Drop unnecessary columns
ID = dataset_test['Id']
dataset_test.drop('Id',axis=1,inplace=True)
dataset_test.drop(rem,axis=1,inplace=True)
X_test = dataset_test.values

#Make predictions using the best model
predictions = best_model.predict(X_test[:,indexes])
# Write submissions to output file in the correct format
with open("submission.csv", "w") as subfile:
    subfile.write("Id,Cover_Type\n")
    for i, pred in enumerate(list(predictions)):
        subfile.write("%s,%s\n"%(ID[i],pred))
```

Algorithms we have used :

- KNN
- Naive Bayes
- Random Forest Classifier

Conclusion :

- We have made use of feature scaling and feature importance to identify the best features.
- To understand how well the model performs, we checked accuracy on the complete training data and with a subset of training data.
- The best model was obtained from using Random Forest Classifier with an accuracy of 86%.
- Other models like Naive Bayes and KNN render an accuracy of 64% and 85% respectively.
- We have predicted the values of testing data using Random Forest Classifier.

File failed to load: file:///Users/sakshikalani/Desktop/ML%20and%20Stats/Project_Group6_files/extensions/MathZoom.js