RandomForest_Part01

March 7, 2022

[1]:

```
import pandas as pd
import seaborn as sns; sns.set()
import matplotlib.pyplot as plt
import numpy as np

pd.options.display.max_columns = None
pd.set_option('display.max_rows', 200)
pd.set_option('display.float_format', lambda x: '%.3f' % x)

#import dataset
df = pd.read_csv ('WinnipegDataset.txt',sep = ",")
```

0.1 Exploratory Data Analysis

[2]:

```
# determine data types in dataset
df.dtypes
```

[2]:

```
label      int64
f1       float64
f2       float64
f3       float64
f4       float64
f5       float64
f6       float64
f7       float64
f8       float64
f9       float64
f10      float64
f11      float64
f12      float64
f13      float64
f14      float64
f15      float64
f16      float64
f17      float64
f18      float64
f19      float64
```

```
f20     float64
f21     float64
f22     float64
f23     float64
f24     float64
f25     float64
f26     float64
f27     float64
f28     float64
f29     float64
f30     float64
f31     float64
f32     float64
f33     float64
f34     float64
f35     float64
f36     float64
f37     float64
f38     float64
f39     float64
f40     float64
f41     float64
f42     float64
f43     float64
f44     float64
f45     float64
f46     float64
f47     float64
f48     float64
f49     float64
f50     float64
f51     float64
f52     float64
f53     float64
f54     float64
f55     float64
f56     float64
f57     float64
f58     float64
f59     float64
f60     float64
f61     float64
f62     float64
f63     float64
f64     float64
f65     float64
f66     float64
```

```
f67      float64
f68      float64
f69      float64
f70      float64
f71      float64
f72      float64
f73      float64
f74      float64
f75      float64
f76      float64
f77      float64
f78      float64
f79      float64
f80      float64
f81      float64
f82      float64
f83      float64
f84      float64
f85      float64
f86      float64
f87      float64
f88      float64
f89      float64
f90      float64
f91      float64
f92      float64
f93      float64
f94      float64
f95      float64
f96      float64
f97      float64
f98      float64
f99      float64
f100     float64
f101     float64
f102     float64
f103     float64
f104     float64
f105     float64
f106     float64
f107     float64
f108     float64
f109     float64
f110     float64
f111     float64
f112     float64
f113     float64
```

```
f114    float64
f115    float64
f116    float64
f117    float64
f118    float64
f119    float64
f120    float64
f121    float64
f122    float64
f123    float64
f124    float64
f125    float64
f126    float64
f127    float64
f128    float64
f129    float64
f130    float64
f131    float64
f132    float64
f133    float64
f134    float64
f135    float64
f136    float64
f137      int64
f138      int64
f139      int64
f140      int64
f141      int64
f142    float64
f143    float64
f144    float64
f145    float64
f146    float64
f147    float64
f148    float64
f149    float64
f150    float64
f151    float64
f152    float64
f153    float64
f154    float64
f155    float64
f156    float64
f157    float64
f158    float64
f159    float64
f160    float64
```

```
f161      float64
f162      float64
f163      float64
f164      float64
f165      float64
f166      float64
f167      float64
f168      float64
f169      float64
f170      float64
f171      float64
f172      float64
f173      float64
f174      float64
dtype: object
```

[3]:
```
#display attributes
display(df)
```

```
        label      f1      f2      f3      f4      f5      f6     f7      f8  \
0           1 -13.559 -21.407 -11.404 -15.248 -11.923 -15.291 -2.155  -7.847
1           1 -12.802 -20.335 -10.399 -14.132 -11.096 -14.361 -2.404  -7.533
2           1 -12.431 -19.902 -10.074 -13.598 -10.829 -14.048 -2.357  -7.472
3           1 -12.689 -19.529 -10.028 -13.350 -11.056 -14.014 -2.661  -6.840
4           1 -12.686 -19.278  -9.819 -13.108 -10.932 -13.939 -2.868  -6.592
…       …     …     …     …     …     …     …     …     …     …
325829      7   2.482  -7.687   1.079   0.743  -0.941   0.905  1.403 -10.169
325830      7   2.523  -7.675   1.087   0.762  -0.912   0.937  1.437 -10.198
325831      7  -1.927 -11.416  -2.435  -3.454  -4.151  -3.481  0.508  -9.489
325832      7   0.125 -10.144  -0.622  -1.542  -2.310  -1.525  0.747 -10.269
325833      7   0.201 -10.050  -0.599  -1.501  -2.251  -1.463  0.800 -10.251

            f9    f10    f11    f12   f13   f14   f15   f16   f17   f18  \
0      -10.002  0.042  3.325  3.368 0.356 0.058 0.585 0.241 0.519 0.239
1       -9.937  0.228  3.036  3.264 0.343 0.061 0.597 0.252 0.508 0.240
2       -9.828  0.450  2.769  3.219 0.345 0.062 0.593 0.264 0.499 0.238
3       -9.501  0.664  2.294  2.958 0.328 0.068 0.605 0.281 0.477 0.241
4       -9.459  0.831  2.176  3.007 0.317 0.069 0.614 0.288 0.475 0.238
…          …    …    …    …    …    …    …    …    …    …
325829  -8.767 -0.162 -1.684 -1.846 0.549 0.053 0.398 0.368 0.250 0.382
325830  -8.761 -0.175 -1.674 -1.848 0.551 0.053 0.396 0.367 0.250 0.383
325831  -8.980  0.027 -0.698 -0.670 0.499 0.056 0.444 0.351 0.299 0.349
325832  -9.522 -0.017 -0.768 -0.785 0.517 0.049 0.435 0.352 0.295 0.353
325833  -9.451 -0.038 -0.749 -0.787 0.519 0.049 0.432 0.351 0.295 0.354

           f19    f20    f21    f22    f23    f24   f25   f26   f27   f28  \
0       -0.624 -0.815 -0.708 -0.656 -0.196 -0.632 0.092 0.027 0.005 0.620
```

```
1      -0.572 -0.749 -0.648 -0.628 -0.188 -0.554 0.116 0.031 0.006 0.604
2      -0.533 -0.703 -0.588 -0.618 -0.178 -0.486 0.127 0.032 0.006 0.587
3      -0.504 -0.670 -0.525 -0.612 -0.175 -0.436 0.127 0.031 0.006 0.578
4      -0.503 -0.682 -0.507 -0.591 -0.175 -0.452 0.132 0.032 0.006 0.579
…        …     …      …      …      …      …     …     …     …     …
325829 -1.581 -1.683 -0.174 -1.650 -0.040 -1.534 2.421 0.755 0.047 0.561
325830 -1.558 -1.654 -0.170 -1.624 -0.039 -1.513 2.439 0.758 0.046 0.560
325831 -2.196 -2.467 -0.167 -2.232 -0.044 -2.194 0.889 0.376 0.019 0.617
325832 -2.000 -2.368 -0.218 -2.068 -0.047 -1.974 1.391 0.570 0.032 0.615
325833 -1.952 -2.295 -0.220 -2.023 -0.048 -1.921 1.408 0.575 0.033 0.616

         f29    f30   f31   f32   f33   f34   f35   f36     f37     f38  \
0      0.684 44.369 0.424 0.196 0.260 0.120 0.055 0.162 -11.923 -12.823
1      0.675 44.992 0.407 0.196 0.268 0.129 0.052 0.158 -11.096 -11.805
2      0.680 45.466 0.399 0.188 0.281 0.132 0.048 0.149 -10.829 -11.377
3      0.675 46.654 0.390 0.188 0.285 0.137 0.047 0.147 -11.056 -11.263
4      0.665 46.828 0.385 0.194 0.280 0.141 0.049 0.152 -10.932 -11.109
…        …     …     …     …     …     …     …     …       …       …
325829 0.882 63.745 0.496 0.066 0.387 0.052 0.019 0.059  -0.941   3.518
325830 0.885 63.669 0.495 0.064 0.390 0.051 0.019 0.057  -0.912   3.544
325831 0.903 60.734 0.557 0.060 0.346 0.037 0.022 0.060  -4.151  -0.820
325832 0.892 60.757 0.549 0.066 0.343 0.041 0.023 0.065  -2.310   1.167
325833 0.890 60.623 0.548 0.068 0.342 0.042 0.024 0.066  -2.251   1.215

           f39   f40   f41   f42   f43   f44   f45   f46   f47   f48   f49  \
0      -21.407 0.176 0.168 0.003 0.066 0.025 0.028 0.057 0.037 0.006 0.025
1      -20.335 0.189 0.182 0.006 0.079 0.027 0.034 0.067 0.043 0.006 0.028
2      -19.902 0.196 0.191 0.010 0.027 0.086 0.040 0.073 0.047 0.006 0.030
3      -19.529 0.198 0.194 0.014 0.025 0.091 0.043 0.075 0.049 0.006 0.030
4      -19.278 0.199 0.198 0.016 0.023 0.095 0.046 0.076 0.051 0.007 0.031
…          …     …     …     …     …     …     …     …     …     …     …
325829  -7.687 0.565 0.940 0.014 0.355 1.554 0.523 0.390 1.545 0.026 0.471
325830  -7.675 0.606 1.014 0.016 0.412 1.827 0.587 0.457 1.815 0.034 0.520
325831 -11.416 0.432 0.748 0.005 0.195 0.971 0.335 0.207 0.967 0.008 0.319
325832 -10.144 0.496 0.834 0.009 0.275 1.222 0.401 0.296 1.216 0.015 0.371
325833 -10.050 0.534 0.897 0.011 0.325 1.424 0.449 0.352 1.416 0.019 0.410

           f50     f51     f52     f53     f54     f55   f56     f57     f58  \
0      -12.564 -23.436 -13.858 -18.209 -11.556 -17.934 1.294 -10.873  -9.578
1      -11.559 -22.730 -12.844 -17.334 -10.500 -17.095 1.285 -11.172  -9.886
2      -11.040 -22.566 -12.458 -16.903  -9.997 -16.873 1.418 -11.526 -10.108
3      -10.910 -22.520 -12.126 -16.840  -9.699 -16.965 1.216 -11.610 -10.395
4      -10.581 -22.387 -11.858 -16.623  -9.384 -16.690 1.277 -11.805 -10.529
…          …       …       …       …       …       …     …       …       …
325829   2.600 -14.506  -2.713  -1.836   0.133  -1.522 5.313 -17.106 -11.793
325830   2.964 -14.247  -2.550  -1.510   0.414  -1.159 5.514 -17.211 -11.697
325831  -0.670 -18.389  -6.103  -6.081  -2.222  -5.675 5.433 -17.718 -12.286
325832   1.641 -16.098  -3.370  -2.929  -0.490  -2.667 5.011 -17.739 -12.728
```

```
325833   2.070 -15.409  -3.201  -2.534  -0.169  -2.231 5.271 -17.479 -12.208

           f59   f60   f61   f62   f63   f64   f65   f66   f67    f68    f69  \
0       -0.275 6.652 6.378 0.548 0.045 0.407 0.149 0.691 0.159 -1.873 -0.785
1       -0.239 6.834 6.595 0.549 0.042 0.409 0.145 0.701 0.154 -1.955 -0.753
2       -0.031 6.907 6.876 0.558 0.039 0.403 0.145 0.710 0.146 -2.361 -0.724
3        0.125 7.141 7.265 0.548 0.038 0.414 0.140 0.724 0.136 -2.684 -0.671
4        0.067 7.239 7.306 0.552 0.036 0.411 0.137 0.727 0.135 -2.442 -0.662
…          …     …     …     …     …     …     …     …     …      …      …
325829 -0.314 1.969 1.655 0.761 0.015 0.224 0.274 0.431 0.295 -0.957 -1.298
325830 -0.351 1.924 1.573 0.769 0.015 0.216 0.275 0.428 0.298 -0.887 -1.322
325831 -0.406 3.859 3.453 0.767 0.013 0.220 0.221 0.537 0.242 -0.689 -0.996
325832 -0.262 2.438 2.176 0.751 0.013 0.237 0.262 0.460 0.278 -0.933 -1.263
325833 -0.303 2.365 2.062 0.760 0.014 0.226 0.263 0.454 0.282 -0.909 -1.321

           f70    f71    f72    f73   f74   f75   f76   f77   f78    f79   f80  \
0       -1.990 -2.997 -1.774 -1.733 0.071 0.025 0.004 0.664 0.705 42.797 0.468
1       -1.978 -2.940 -1.828 -1.755 0.091 0.031 0.005 0.651 0.715 42.706 0.466
2       -2.128 -3.689 -2.106 -1.668 0.102 0.034 0.005 0.638 0.721 42.195 0.460
3       -2.298 -3.803 -2.348 -1.802 0.109 0.034 0.006 0.626 0.719 42.803 0.450
4       -2.098 -3.860 -2.130 -1.765 0.117 0.036 0.006 0.619 0.727 42.535 0.450
…          …      …      …      …     …     …     …     …     …      …     …
325829 -1.357 -0.989 -1.072 -0.540 1.880 0.482 0.029 0.515 0.886 26.197 0.456
325830 -1.373 -0.913 -1.030 -0.521 2.039 0.503 0.030 0.506 0.887 25.437 0.449
325831 -0.950 -0.703 -0.766 -0.461 0.905 0.202 0.010 0.475 0.906 26.574 0.430
325832 -1.235 -0.982 -1.001 -0.561 1.514 0.410 0.020 0.518 0.908 27.211 0.470
325833 -1.299 -0.946 -1.012 -0.540 1.662 0.432 0.023 0.513 0.898 26.204 0.461

          f81   f82   f83   f84   f85     f86     f87     f88   f89   f90  \
0       0.196 0.237 0.099 0.062 0.174 -11.556 -15.740 -23.436 0.182 0.119
1       0.185 0.249 0.099 0.057 0.163 -10.500 -14.859 -22.730 0.201 0.130
2       0.178 0.261 0.101 0.053 0.154  -9.997 -14.509 -22.566 0.216 0.138
3       0.176 0.269 0.105 0.051 0.150  -9.699 -14.532 -22.520 0.226 0.141
4       0.169 0.277 0.104 0.049 0.143  -9.384 -14.269 -22.387 0.235 0.146
…         …     …     …     …     …       …       …       …     …     …
325829 0.059 0.430 0.055 0.015 0.049   0.133   1.219 -14.506 0.669 0.717
325830 0.057 0.438 0.056 0.015 0.047   0.414   1.567 -14.247 0.719 0.802
325831 0.045 0.475 0.050 0.011 0.036  -2.222  -2.986 -18.389 0.552 0.559
325832 0.048 0.437 0.044 0.013 0.041  -0.490   0.112 -16.098 0.614 0.655
325833 0.052 0.437 0.049 0.014 0.044  -0.169   0.521 -15.409 0.664 0.736

          f91   f92   f93   f94   f95   f96   f97   f98   f99  f100  f101  \
0       0.003 0.058 0.020 0.018 0.059 0.019 0.004 0.016 0.009 0.057 0.048
1       0.002 0.072 0.023 0.020 0.073 0.023 0.004 0.019 0.001 0.044 0.037
2       0.000 0.084 0.026 0.021 0.084 0.026 0.004 0.021 0.001 0.046 0.046
3       0.001 0.093 0.028 0.022 0.093 0.028 0.004 0.021 0.018 0.071 0.074
4       0.002 0.100 0.030 0.023 0.101 0.030 0.004 0.022 0.023 0.066 0.068
…         …     …     …     …     …     …     …     …     …     …     …
```

```
325829 0.027 0.517 1.324 0.121 0.614 1.269 0.040 0.038 0.001 0.039 0.031
325830 0.031 0.597 1.636 0.137 0.709 1.577 0.050 0.033 0.046 0.088 0.069
325831 0.021 0.359 0.828 0.071 0.425 0.787 0.024 0.022 0.001 0.026 0.005
325832 0.024 0.456 1.097 0.093 0.533 1.052 0.031 0.028 0.001 0.041 0.024
325833 0.026 0.532 1.363 0.110 0.621 1.315 0.039 0.030 0.005 0.045 0.045

        f102  f103  f104   f105  f106  f107  f108  f109   f110  f111  f112  \
0      0.165 0.369 0.770  7.688 1.188 0.505 0.618 0.525  0.086 0.732 0.503
1      0.103 0.290 0.774  7.838 1.189 0.420 0.598 0.459  0.086 0.737 0.425
2      0.107 0.298 0.733  6.478 1.000 0.402 0.532 0.448  0.000 0.733 0.393
3      0.120 0.284 0.587  3.838 0.959 0.330 0.372 0.367 -0.021 0.600 0.293
4      0.131 0.264 0.590  3.882 0.971 0.327 0.401 0.353 -0.015 0.600 0.280
…        …     …     …      …     …     …     …     …      …     …     …
325829 0.113 0.309 0.818  9.968 1.258 0.467 0.670 0.496  0.114 0.776 0.482
325830 0.117 0.222 0.526  3.217 1.275 0.296 0.414 0.290  0.121 0.432 0.250
325831 0.096 0.408 0.976 81.600 5.200 0.704 0.957 0.662  0.677 0.880 0.914
325832 0.107 0.278 0.841 11.583 1.708 0.449 0.711 0.475  0.262 0.743 0.484
325833 0.097 0.229 0.672  5.089 1.000 0.315 0.459 0.357  0.000 0.672 0.289

        f113  f114   f115   f116  f117   f118  f119   f120   f121   f122  \
0      0.382 2.236 -0.486 17.280 0.549  0.128 7.380  3.438 22.667  8.222
1      0.476 2.816 -0.401 16.240 0.471  0.099 4.240  2.784 18.556  0.469
2      0.472 2.785 -0.399 16.580 0.399  0.098 3.660  2.326 19.000  0.667
3      0.406 2.367 -0.257 14.270 0.237  0.090 2.640  1.622 17.333  1.556
4      0.337 2.015 -0.330 11.320 0.317  0.114 3.700  1.927 16.667  0.444
…        …     …      …      …     …      …     …      …      …      …
325829 0.464 2.735 -0.487 16.900 0.569  0.084 5.240  3.645 22.222 24.395
325830 0.310 1.897 -0.141  9.160 0.258  0.115 3.640  1.696 15.444 25.136
325831 0.619 4.250 -0.574 27.380 0.901 -0.533 6.300 19.200 23.111  3.877
325832 0.444 2.598 -0.446 14.730 0.634  0.072 5.660  4.458 19.111  4.543
325833 0.405 2.361 -0.366 11.360 0.366  0.089 3.120  2.156 14.667 10.222

        f123   f124  f125  f126  f127   f128  f129   f130  f131   f132  f133  \
0      0.275 13.444 3.000 2.197 0.111  0.453 7.778 17.284 0.276 24.556 3.667
1      0.600  1.333 0.889 1.523 0.235 -0.132 3.889  0.321 0.722  0.556 0.556
2      0.578  2.444 1.111 1.677 0.210  0.317 3.889  0.321 0.667  0.667 0.667
3      0.400  2.000 1.333 1.523 0.235  0.081 6.222  1.506 0.533  2.000 1.111
4      0.611  0.778 0.778 1.273 0.309  0.811 6.556  0.914 0.600  1.333 0.889
…        …      …     …     …     …      …     …      …     …      …     …
325829 0.156 40.667 5.333 1.677 0.210  0.380 3.778  0.173 0.833  0.333 0.333
325830 0.116 32.000 4.444 1.889 0.160  0.015 6.556  6.469 0.242 16.667 3.333
325831 0.429  3.778 1.556 2.043 0.136  0.791 2.222  0.395 0.722  0.556 0.556
325832 0.320  7.667 2.333 1.889 0.160  0.039 3.222  0.617 0.667  0.667 0.667
325833 0.144 18.444 3.778 1.889 0.160 -0.300 6.111 11.210 0.199 29.667 4.556

        f134  f135   f136  f137  f138  f139  f140  f141  f142  f143  f144  \
0      2.043 0.136  0.651  6711  6143  4570  5064  8212 0.285 1.797 1.344
1      1.273 0.309  0.105  6274  5084  3297  3777  8214 0.427 2.491 1.542
```

```
2      1.273 0.309  0.614  6215  5035  3033  3837  8588 0.478 2.832 1.660
3      1.889 0.160  0.064  6836  5745  4212  4534  7691 0.292 1.826 1.364
4      1.581 0.210  0.658  6682  5883  4434  4627  7072 0.229 1.595 1.327
…          …     …      …     …     …     …     …     …     …     …     …
325829 0.849 0.506 -0.189  5960  5510  3023  4711  9327 0.510 3.085 1.823
325830 2.043 0.136  0.823  7649  6544  4690  4823  7104 0.205 1.515 1.395
325831 1.677 0.210  0.561  6004  5343  3444  4255  8071 0.402 2.344 1.551
325832 1.523 0.235  0.620  6234  5469  3442  4472  6766 0.326 1.966 1.589
325833 2.197 0.111  0.704  7980  6950  5297  5004  6458 0.099 1.219 1.312

         f145  f146  f147  f148   f149  f150  f151  f152  f153       f154  \
0      -0.619 0.543 0.427 0.147  0.144 0.586 0.237 1.622 0.237 294110.000
1      -0.645 0.925 0.641 0.213  0.235 0.751 0.370 2.175 0.370 412400.000
2      -0.700 1.035 0.717 0.248  0.261 0.809 0.382 2.238 0.382 439570.000
3      -0.475 0.658 0.438 0.154  0.145 0.601 0.258 1.696 0.258 296240.000
4      -0.401 0.528 0.344 0.140  0.092 0.536 0.209 1.528 0.209 232610.000
…           …     …     …     …      …     …     …     …     …          …
325829 -0.914 0.982 0.766 0.291  0.257 0.868 0.329 1.980 0.329 423430.000
325830 -0.273 0.608 0.307 0.165  0.041 0.560 0.191 1.473 0.191 222500.000
325831 -0.710 0.803 0.603 0.216  0.203 0.745 0.310 1.897 0.310 354320.000
325832 -0.430 0.825 0.488 0.227  0.106 0.734 0.204 1.513 0.204 216430.000
325833 -0.134 0.424 0.148 0.135 -0.037 0.411 0.127 1.291 0.127 150320.000

         f155     f156       f157  f158   f159  f160  f161   f162  f163  f164  \
0       0.051 2199.400  92560.000 1.108 48.444 3.136 0.578  2.444 1.111 1.831
1       0.068 2338.400 100280.000 1.146 49.778 0.395 0.467  1.333 1.111 1.215
2       0.117 3321.300 128320.000 1.265 49.444 0.247 0.611  0.778 0.778 1.369
3       0.037 1748.100  80640.000 1.076 50.667 0.889 0.611  0.778 0.778 1.889
4       0.021 1365.400  69540.000 1.044 51.222 0.395 0.567  1.667 1.000 1.735
…           …        …          …     …      …     …     …      …     …     …
325829  0.218 5811.100 200760.000 1.558 47.000 9.556 0.367 10.556 2.556 2.043
325830  0.014 1460.900  82140.000 1.028 49.667 2.222 0.371  4.889 1.778 2.043
325831  0.105 3239.500 124620.000 1.236 48.778 6.617 0.341  9.000 2.333 2.043
325832  0.130 3867.200 142880.000 1.299 51.556 0.247 0.778  0.444 0.444 1.215
325833 -0.028  224.020  48540.000 0.945 51.333 1.333 0.495  3.111 1.333 2.197

        f165   f166  f167   f168  f169   f170  f171  f172  f173   f174
0      0.185  0.726 5.333  6.000 0.295  9.778 2.444 1.677 0.210  0.654
1      0.333 -0.488 2.111  0.099 0.833  0.333 0.333 0.849 0.506 -0.189
2      0.259  0.253 2.222  0.173 0.689  0.889 0.667 1.273 0.309  0.105
3      0.160  0.438 4.111  0.321 0.833  0.333 0.333 1.149 0.383  0.416
4      0.185  0.350 4.000  0.444 0.689  0.889 0.667 1.581 0.210  0.500
…          …      …     …      …     …      …     …     …     …      …
325829 0.136  0.761 2.333  0.222 0.889  0.222 0.222 1.061 0.358  0.632
325830 0.136 -0.384 6.000 10.444 0.127 19.556 4.000 2.197 0.111  0.812
325831 0.136  0.755 2.333  0.222 0.833  0.333 0.333 1.273 0.309  0.316
325832 0.333  0.060 3.000  0.000 0.944  0.111 0.111 0.349 0.802  0.189
325833 0.111  0.168 6.667 11.111 0.356 14.333 3.000 2.043 0.136  0.727
```

```
[325834 rows x 175 columns]
```

[4]:

```
# identifies columns having null values as a list
df.columns[df.isna().any()].tolist()
```

[4]:

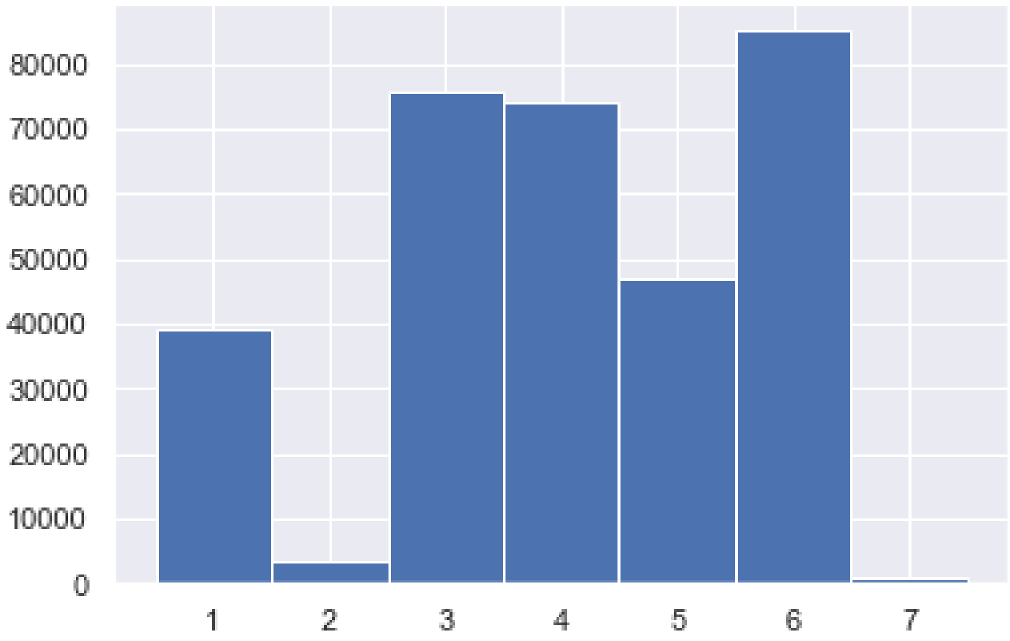
```
[]
```

[5]:

```
# plot histogram of label column
plt.hist(df['label'], bins=np.arange(df['label'].min(), df['label'].max()+2)-0.
 ↪5)
```

[5]:

```
(array([39162.,  3598., 75673., 74067., 47117., 85074.,  1143.]),
 array([0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5]),
 <BarContainer object of 7 artists>)
```



The histograms, boxplots and descriptive stats of the attributes were not run because when the output pdf is created the 175 columns when plotted in the pdf take up a lot of redundant space.

[ ]:

```
# histograms of dataset attributes
for column in df:
    plt.figure()
    df.hist([column])
```

[ ]:

```
# boxplots of dataset attributes
for column in df:
    plt.figure()
    df.boxplot([column])
```

[ ]:

```
# descriptive statistics of attributes
print(df.describe(include='all'))
```

It was decided not to drop any features or remove any outliers. Outliers were not removed because in remote sensing each value of attribute corresponds to a pixel in the output remote sensing image. In essesnce by removing that value you would not be able to classify that pixel.

0.2 Split Dataset

[6]:

```
# prepare data for train/test split
X = df.drop("label", 1)
y = df['label']
```

[7]:

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=0)
```

[8]:

```
# standardize the X dataset
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

0.3 Reduce Dimensions Using Principal Component Analysis

[9]:

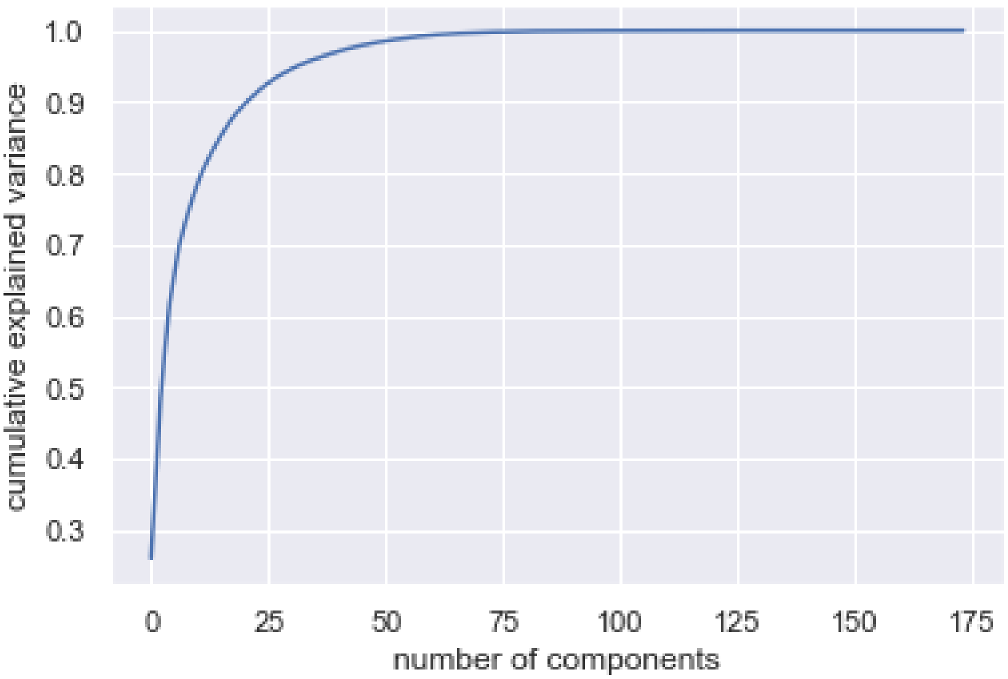
```
# run base PCA using all components
from sklearn.decomposition import PCA

pca = PCA()
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
```

[10]:

```
# plot variance explained by number of components in PCA
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance')
```

[10]:

```
Text(0, 0.5, 'cumulative explained variance')
```



```
[11]: # 11 components were determined to give the highest accuracy when running base␣
       ↪RandomForestClassifier
      pca = PCA(n_components = 11)
      X_train_pca11 = pca.fit_transform(X_train)
      X_test_pca11 = pca.transform(X_test)
```

0.4 Fine Tune Hyperparameters of Random Forest Classifier

```
[ ]: # next sections uses random grid and search to fine tune hyperparameters of␣
      ↪RandomForest classifier

     n_estimators = [5,20,50,100] # number of trees in the random forest
     max_features = ['auto', 'sqrt'] # number of features in consideration at every␣
      ↪split
     max_depth = [int(x) for x in np.linspace(10, 120, num = 12)] # maximum number␣
      ↪of levels allowed in each decision tree
     min_samples_split = [2, 6, 10] # minimum sample number to split a node
     min_samples_leaf = [1, 3, 4] # minimum sample number that can be stored in a␣
      ↪leaf node
     bootstrap = [True, False] # method used to sample data points

     random_grid = {'n_estimators': n_estimators,

     'max_features': max_features,
     'max_depth': max_depth,

     'min_samples_split': min_samples_split,

     'min_samples_leaf': min_samples_leaf,

     'bootstrap': bootstrap}
```

[12]:

```
# create RandomForest classifier
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier()
```

[ ]:

```
# create RandomizedSearch object with parameters
from sklearn.model_selection import RandomizedSearchCV
classifier_random = RandomizedSearchCV(estimator =␣
 ↪classifier,param_distributions = random_grid,
               n_iter = 100, cv = 5, verbose=2, random_state=35, n_jobs = -1)
```

[ ]:

```
# run the RandomizedSearch classifier on training data to determine best␣
 ↪parameters for RandomForest classifier run
classifier_random.fit(X_train_pca11, y_train)
```

[ ]:

```
# print random grid parameters
print ('Random grid: ', random_grid, '\n')
# print the best parameters
print ('Best Parameters: ', classifier_random.best_params_, ' \n')
```

0.5 Random Forest Classifier Model

[13]:

```
# run the RandomForest classifier using fine tuned settings
from sklearn.ensemble import RandomForestClassifier

classifier_finetune = RandomForestClassifier(n_estimators=50,␣
 ↪min_samples_split=6, min_samples_leaf=1, max_features='sqrt', max_depth=100,␣
 ↪bootstrap=False, random_state=0)
classifier_finetune.fit(X_train_pca11, y_train)

# Predicting the Test set results
y_pred = classifier_finetune.predict(X_test_pca11)
```

[14]:

```
# create confusion matrix and calculate accuracy and cohen's kappa
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import cohen_kappa_score

cm = confusion_matrix(y_test, y_pred)
print(cm, ' \n')
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Cohen's Kappa:", cohen_kappa_score(y_test, y_pred))
```

```
[[11803     3     5    27     4     6     8]
 [    2  1064     1     3     0     0     0]
 [    1     2 22791    24     6    10     0]
 [   29     1    24 22222    31    29     0]
 [   23     0    13    58 13500   373     6]
 [   18     0    11    33   172 25100     3]
 [   12     2     0     0     1     3   327]]

              precision    recall  f1-score   support

           1       0.99      1.00      0.99     11856
           2       0.99      0.99      0.99      1070
           3       1.00      1.00      1.00     22834
           4       0.99      0.99      0.99     22336
           5       0.98      0.97      0.98     13973
           6       0.98      0.99      0.99     25337
           7       0.95      0.95      0.95       345

    accuracy                           0.99     97751
   macro avg       0.99      0.98      0.98     97751
weighted avg       0.99      0.99      0.99     97751

Accuracy: 0.9903428097922272
Cohen's Kappa: 0.9877836562872402
```

[ ]: