Geological_Log_Data_ML_final

April 8, 2018

The following work has been performed under the supervision of Dr. Ekarit Panacharoensawad, Assistant Professor, Department of Petroleum Engineering, Texas Tech University.

1 Notebook Setup

```
In [1]: # Common imports
        import numpy as np
        import pandas as pd
        # to make this notebook's output stable across runs
        np.random.seed(42)

        # To plot pretty figures
        %matplotlib inline
        import matplotlib
        import matplotlib.pyplot as plt
        plt.rcParams['axes.labelsize'] = 14
        plt.rcParams['xtick.labelsize'] = 12
        plt.rcParams['ytick.labelsize'] = 12
```

```
In [2]: LogDat = pd.read_excel(r"C:/Users/Amir's/Desktop/Amir_Adjusted.xlsx")
```

2 Getting an overview of the data

```
In [3]: LogDat.head()
```

```
Out[3]:     DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
        0  3600.0           29.6276   7.8359           29.0183    26.4565
        1  3600.5           28.5671   7.8418           28.4555    28.7921
        2  3601.0           27.1170   7.8434           27.3459    27.4413
        3  3601.5           24.8582   7.8558           25.3447    25.6896
        4  3602.0           23.1241   7.8720           22.7096    27.0588

           Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
        0         4.0462        2.2138             -0.0356              0.9126
        1         4.1226        2.2234             -0.0395              0.8803
        2         4.2350        2.2424             -0.0362              0.8754
```

```
3        4.3685       2.2766            -0.0289             0.9005
4        4.5133       2.3217            -0.0250             0.9582

   Resistivity (Medium)     ...      Micro-inverse resistivity (micro log)  \
0                1.0719     ...                                     5.1127
1                1.0008     ...                                     5.0602
2                0.9679     ...                                     4.9294
3                0.9813     ...                                     5.2303
4                1.0502     ...                                     5.2853

   Micro-normal resistivity (micro log)  \
0                                7.7722
1                                7.4297
2                                7.0917
3                                7.0816
4                                7.2144

   Delta-t (interval transit time, or slowness)  Sonic porosity  \
0                                       78.7252         22.0122
1                                       78.2474         21.6743
2                                       77.6106         21.2239
3                                       76.7257         20.5981
4                                       75.4503         19.6961

   Type of Formation  Unnamed: 18  NPOR  \
0    shaly limestone          NaN  CALI
1    shaly limestone          NaN  DPOR
2    shaly limestone          NaN    GR
3    shaly limestone          NaN    PE
4    shaly limestone          NaN  RHOB

  Neutron porosity, calculated assuming a limestone matrix  Unnamed: 21  \
0                                            Caliper                NaN
1  Density porosity, calculated assuming a limest...                NaN
2                                              Gamma                ray
3                                      Photoelectric             factor
4                                               Bulk            density

  Unnamed: 22
0         NaN
1         NaN
2         NaN
3         NaN
4         NaN

[5 rows x 23 columns]
```

```
In [4]: LogDat_adj = LogDat.drop(LogDat.columns[[18,19,20,21,22]], axis=1)  # df.columns is ze
```

In [5]:
```python
LogDat_adj.head()
```

Out[5]:
```
    DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
0  3600.0           29.6276   7.8359           29.0183    26.4565
1  3600.5           28.5671   7.8418           28.4555    28.7921
2  3601.0           27.1170   7.8434           27.3459    27.4413
3  3601.5           24.8582   7.8558           25.3447    25.6896
4  3602.0           23.1241   7.8720           22.7096    27.0588

   Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
0         4.0462        2.2138             -0.0356              0.9126
1         4.1226        2.2234             -0.0395              0.8803
2         4.2350        2.2424             -0.0362              0.8754
3         4.3685        2.2766             -0.0289              0.9005
4         4.5133        2.3217             -0.0250              0.9582

   Resistivity (Medium)  Resistivity (Shallow)  \
0                1.0719                 5.2530
1                1.0008                 4.6464
2                0.9679                 4.3056
3                0.9813                 4.1801
4                1.0502                 4.1355

   Ratio (shallow/deep resistivity)  Spontaneous Potential  \
0                          -68.4118               -20.3987
1                          -65.0243               -19.9382
2                          -62.2656               -19.4078
3                          -60.0036               -18.7673
4                          -57.1585               -17.8640

   Micro-inverse resistivity (micro log)  \
0                                 5.1127
1                                 5.0602
2                                 4.9294
3                                 5.2303
4                                 5.2853

   Micro-normal resistivity (micro log)  \
0                                7.7722
1                                7.4297
2                                7.0917
3                                7.0816
4                                7.2144

   Delta-t (interval transit time, or slowness)  Sonic porosity  \
0                                       78.7252         22.0122
1                                       78.2474         21.6743
2                                       77.6106         21.2239
```

```
3                                     76.7257         20.5981
4                                     75.4503         19.6961

  Type of Formation
0   shaly limestone
1   shaly limestone
2   shaly limestone
3   shaly limestone
4   shaly limestone
```

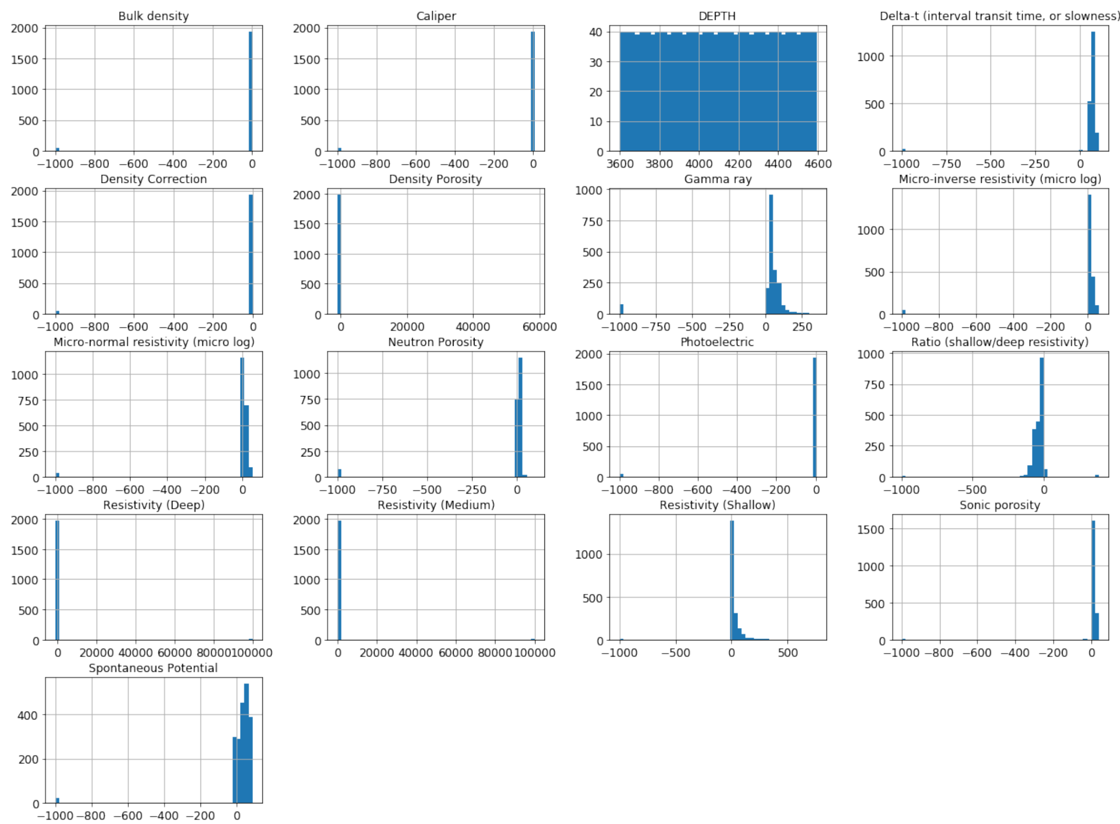
In [6]:
```
LogDat_adj.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1989 entries, 0 to 1988
Data columns (total 18 columns):
DEPTH                                            1989 non-null float64
Neutron Porosity                                 1989 non-null float64
Caliper                                          1989 non-null float64
Density Porosity                                 1989 non-null float64
Gamma ray                                        1989 non-null float64
Photoelectric                                    1989 non-null float64
Bulk density                                     1989 non-null float64
Density Correction                               1989 non-null float64
Resistivity (Deep)                               1989 non-null float64
Resistivity (Medium)                             1989 non-null float64
Resistivity (Shallow)                            1989 non-null float64
Ratio (shallow/deep resistivity)                 1989 non-null float64
Spontaneous Potential                            1989 non-null float64
Micro-inverse resistivity (micro log)            1989 non-null float64
Micro-normal resistivity (micro log)             1989 non-null float64
Delta-t (interval transit time, or slowness)     1989 non-null float64
Sonic porosity                                   1989 non-null float64
Type of Formation                                1936 non-null object
dtypes: float64(17), object(1)
memory usage: 272.0+ KB
```

In [7]:
```
LogDat_adj.hist(bins=50, figsize=(20,15))
```

Out[7]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x097BF5D0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x09888090>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0B588E90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x098E1E70>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x097FB1B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x097FB630>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0984C9D0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0B605D90>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0B63BEB0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0B687110>,
```

```
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B6BEA30>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B706BD0>],
[<matplotlib.axes._subplots.AxesSubplot object at 0x0B755970>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B79C2D0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B7ED970>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B7F5CB0>],
[<matplotlib.axes._subplots.AxesSubplot object at 0x0B867CB0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B89DD30>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B8E2F50>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x0B9218D0>]], dtype=object)
```




```
In [8]: LogDat_adj["Type of Formation"].value_counts()

Out[8]: limestone          956
        shaly limestone    456
        shale              258
        dolomite           163
        sandstone           59
        sandy limestone     38
        shaly sandstone      6
        Name: Type of Formation, dtype: int64
```

```
In [9]: LogDat_adj.describe()
```

```
Out[9]:             DEPTH  Neutron Porosity      Caliper  Density Porosity  \
        count  1989.000000       1989.000000  1989.000000       1989.000000
        mean   4097.000000        -20.183288   -17.790037         15.149971
        std     287.159581        185.811695   159.254436       1323.847169
        min    3600.000000       -999.250000  -999.250000       -999.250000
        25%    3848.500000          9.126500     7.824600          5.306900
        50%    4097.000000         14.206000     7.963200         10.149200
        75%    4345.500000         19.510700     8.116000         16.641000
        max    4594.000000        100.000000    10.674400      58594.152300

                Gamma ray  Photoelectric  Bulk density  Density Correction  \
        count  1989.000000    1989.000000   1989.000000         1989.000000
        mean     16.863204     -21.745637    -23.678348          -26.110950
        std     210.172649     158.613934    159.884135          159.485422
        min    -999.250000    -999.250000   -999.250000         -999.250000
        25%      33.252500       3.435300      2.411800           -0.027600
        50%      45.027200       4.119000      2.525400           -0.004600
        75%      72.440300       4.509200      2.611900            0.032700
        max     351.118300       5.908800      3.760500            0.250300

               Resistivity (Deep)  Resistivity (Medium)  Resistivity (Shallow)  \
        count         1989.000000           1989.000000            1989.000000
        mean           611.082785            668.888914              26.364424
        std           7746.001211           8058.989422              80.716629
        min           -999.250000              0.933400            -999.250000
        25%              3.207600              3.463900               6.482900
        50%              6.791700              7.057600              12.195300
        75%             11.986000             13.988200              26.418100
        max         100000.000000         100000.000000             761.316400

               Ratio (shallow/deep resistivity)  Spontaneous Potential  \
        count                       1989.000000            1989.000000
        mean                         -32.485838              28.171579
        std                           68.841062             110.035585
        min                         -999.250000            -999.250000
        25%                          -52.762100              20.228000
        50%                          -22.155900              41.508600
        75%                          -12.660200              63.621300
        max                          392.674500              88.764900

               Micro-inverse resistivity (micro log)  \
        count                            1989.000000
        mean                               -5.352822
        std                               141.076323
        min                              -999.250000
        25%                                 6.013800
```

```
50%                              9.454500
75%                             19.562200
max                             61.073700

       Micro-normal resistivity (micro log)  \
count                             1989.000000
mean                                -5.786598
std                                138.991134
min                               -999.250000
25%                                  7.326300
50%                                 10.860500
75%                                 16.472200
max                                 55.081500

       Delta-t (interval transit time, or slowness)  Sonic porosity
count                                   1989.000000     1989.000000
mean                                      60.550176        7.206027
std                                       93.067429       88.104178
min                                     -999.250000     -999.250000
25%                                       61.623200        9.917400
50%                                       67.248700       13.895900
75%                                       73.485100       18.306300
max                                      106.495900       41.652000
```

3 Data Cleaning

In [10]:
```
inds = pd.isnull(LogDat_adj['Type of Formation']).nonzero()[0]
inds
```

Out[10]:
```
array([1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946,
       1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957,
       1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968,
       1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979,
       1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988], dtype=int32)
```

In [11]:
```
LogDat_prob_GR = LogDat_adj[(LogDat_adj['Gamma ray'] <= -700)]
LogDat_prob_GR.head()
```

Out[11]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
1910  4555.0           26.5237   7.6225           20.0142    -999.25
1911  4555.5           26.6968   7.6137           19.7698    -999.25
1912  4556.0           27.3526   7.5680           19.5087    -999.25
1913  4556.5           28.3447   7.5731           19.3305    -999.25
1914  4557.0           33.4406   7.6147           19.8108    -999.25

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
1910         2.6090        2.3678              0.0404              2.5194
1911         2.5529        2.3719              0.0520              2.4917
```

```
1912        2.5357      2.3764             0.0522              2.4647
1913        2.5532      2.3794             0.0462              2.4469
1914        2.5530      2.3712             0.0368              2.4500

      Resistivity (Medium)  Resistivity (Shallow)  \
1910                2.4510                 4.3735
1911                2.4320                 4.1189
1912                2.4350                 3.1196
1913                2.4490                 3.5166
1914                2.4477                 5.0658

      Ratio (shallow/deep resistivity)  Spontaneous Potential  \
1910                          -21.5584                19.6388
1911                          -19.6457                19.3252
1912                           -9.2108                19.2341
1913                          -14.1763                19.4474
1914                          -28.3930                19.8138

      Micro-inverse resistivity (micro log)  \
1910                                 4.5649
1911                                 5.3605
1912                                 4.1063
1913                                 4.5767
1914                                 6.5113

      Micro-normal resistivity (micro log)  \
1910                                5.7073
1911                                5.4725
1912                                4.6215
1913                                5.4416
1914                                6.7475

      Delta-t (interval transit time, or slowness)  Sonic porosity  \
1910                                       69.8108         15.7078
1911                                       69.0965         15.2026
1912                                       68.9684         15.1120
1913                                       69.5271         15.5071
1914                                       70.4744         16.1771

     Type of Formation
1910          dolomite
1911          dolomite
1912          dolomite
1913          dolomite
1914          dolomite
```

In [12]:
```
LogDat_prob_NP = LogDat_adj[(LogDat_adj['Neutron Porosity'] <= -700)]
LogDat_prob_NP.head()
```

Out[12]:

```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
1920  4560.0           -999.25   7.2890           20.0320    -999.25
1921  4560.5           -999.25   7.0684           18.4702    -999.25
1922  4561.0           -999.25   7.0343           17.4400    -999.25
1923  4561.5           -999.25   7.0373           17.0232    -999.25
1924  4562.0           -999.25   7.0323           17.0760    -999.25

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
1920         2.3696        2.3675              0.1415              2.7566
1921         2.1840        2.3942              0.1957              2.8312
1922         2.0050        2.4118              0.2296              2.9054
1923         1.8791        2.4189              0.2441              2.9639
1924         1.8574        2.4180              0.2482              2.9982

      Resistivity (Medium)  Resistivity (Shallow)  \
1920                2.8015                 4.6931
1921                2.8722                 4.6887
1922                2.9531                 4.6327
1923                3.0340                 4.5897
1924                3.0811                 4.4398

      Ratio (shallow/deep resistivity)  Spontaneous Potential  \
1920                          -20.7969                20.6447
1921                          -19.7168                20.7250
1922                          -18.2357                20.7377
1923                          -17.0929                20.7974
1924                          -15.3455                20.9353

      Micro-inverse resistivity (micro log)  \
1920                                 5.0779
1921                                 5.8167
1922                                 5.3251
1923                                 5.0409
1924                                 4.7036

      Micro-normal resistivity (micro log)  \
1920                                6.6270
1921                                6.6950
1922                                6.2995
1923                                5.8876
1924                                5.7652

      Delta-t (interval transit time, or slowness)  Sonic porosity  \
1920                                       72.8398         17.8499
1921                                       72.4746         17.5916
1922                                       72.2784         17.4529
1923                                       72.3258         17.4864
1924                                       72.4687         17.5875
```

```
     Type of Formation
1920          dolomite
1921          dolomite
1922          dolomite
1923          dolomite
1924          dolomite
```

```
In [13]: LogDat_drop = LogDat_adj.drop(LogDat_adj.index[1905:1989])
```

```
In [14]: LogDat_drop.head()
```

```
Out[14]:     DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
         0  3600.0           29.6276   7.8359           29.0183    26.4565
         1  3600.5           28.5671   7.8418           28.4555    28.7921
         2  3601.0           27.1170   7.8434           27.3459    27.4413
         3  3601.5           24.8582   7.8558           25.3447    25.6896
         4  3602.0           23.1241   7.8720           22.7096    27.0588

            Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
         0         4.0462        2.2138             -0.0356              0.9126
         1         4.1226        2.2234             -0.0395              0.8803
         2         4.2350        2.2424             -0.0362              0.8754
         3         4.3685        2.2766             -0.0289              0.9005
         4         4.5133        2.3217             -0.0250              0.9582

            Resistivity (Medium)  Resistivity (Shallow)  \
         0                1.0719                 5.2530
         1                1.0008                 4.6464
         2                0.9679                 4.3056
         3                0.9813                 4.1801
         4                1.0502                 4.1355

            Ratio (shallow/deep resistivity)  Spontaneous Potential  \
         0                          -68.4118               -20.3987
         1                          -65.0243               -19.9382
         2                          -62.2656               -19.4078
         3                          -60.0036               -18.7673
         4                          -57.1585               -17.8640

            Micro-inverse resistivity (micro log)  \
         0                                 5.1127
         1                                 5.0602
         2                                 4.9294
         3                                 5.2303
         4                                 5.2853

            Micro-normal resistivity (micro log)  \
```

```
0                                 7.7722
1                                 7.4297
2                                 7.0917
3                                 7.0816
4                                 7.2144

   Delta-t (interval transit time, or slowness)  Sonic porosity  \
0                                       78.7252         22.0122
1                                       78.2474         21.6743
2                                       77.6106         21.2239
3                                       76.7257         20.5981
4                                       75.4503         19.6961

  Type of Formation
0   shaly limestone
1   shaly limestone
2   shaly limestone
3   shaly limestone
4   shaly limestone
```

4 Creating a Test : Setting it aside as in never looking at it

```
In [15]: from sklearn.model_selection import StratifiedShuffleSplit

         split = StratifiedShuffleSplit(test_size=0.27,  random_state=42)
         for train_index, test_index in split.split(LogDat_drop, LogDat_drop["Type of Formatio
             strat_train_set = LogDat_drop.loc[train_index]
             strat_test_set = LogDat_drop.loc[test_index]

In [16]: strat_train_set

Out[16]:        DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
         364   3782.0           14.9090   8.0845           10.8138    37.9302
         1119  4159.5            5.4242   7.9526            2.0286    87.9380
         974   4087.0           26.8415   7.6343           30.5027    30.2045
         481   3840.5           29.6743  10.0896           17.5728   155.6782
         828   4014.0            3.9354   8.1300            3.0607    38.1546
         361   3780.5           15.7576   7.8321           13.9503    34.3738
         1628  4414.0           24.1064   8.7090           18.4338   180.3199
         536   3868.0           25.6040   9.1893           19.6051    99.0748
         1548  4374.0           14.6525   7.9889            8.3089    85.6620
         1001  4100.5            5.7411   8.0615            4.0265    32.5423
         1615  4407.5            2.4681   7.8820            2.0416    28.3417
         349   3774.5           19.5342   7.7914           18.8532    30.3607
         628   3914.0            4.5268   8.0204            0.7654    27.8219
         1061  4130.5            8.6487   7.9255            4.7786    45.1159
         1375  4287.5            6.9581   7.9081            4.7630    30.0639
         1840  4520.0           11.9362   7.7080            9.8908    21.3981
```

1464	4332.0	9.6910	7.9502	6.8707	40.7099
1138	4169.0	18.6090	8.4035	10.4345	133.9904
1815	4507.5	18.3457	7.6868	8.3953	26.0199
314	3757.0	17.6590	8.1546	9.3416	109.3162
1760	4480.0	15.0935	7.7758	21.1614	26.6931
333	3766.5	15.8041	7.8049	15.2636	44.2266
826	4013.0	6.2868	8.0844	4.2857	41.5511
1800	4500.0	21.9464	7.7996	14.3607	34.5017
299	3749.5	18.8596	8.5540	30.6595	186.4201
237	3718.5	14.8584	8.1228	7.9773	65.7383
1360	4280.0	20.6957	8.1821	11.0452	114.6199
796	3998.0	1.0332	8.1156	0.4334	32.6536
706	3953.0	8.3378	7.9880	7.9189	36.4187
1717	4458.5	12.2060	7.7407	22.1095	22.4116
...	...	...	...	...	...
232	3716.0	13.8051	7.8854	11.3098	34.3235
186	3693.0	17.0633	8.5065	11.3280	81.0988
1560	4380.0	6.9080	7.6101	3.2134	38.0405
751	3975.5	10.8680	7.6476	8.6837	22.3086
1385	4292.5	10.1502	7.8432	11.3457	35.0013
415	3807.5	17.1766	7.7538	16.5338	45.6701
1736	4468.0	11.1897	7.7834	21.3862	20.8727
357	3778.5	17.7639	7.7329	15.4558	37.7116
891	4045.5	3.9459	8.1124	1.1336	31.4204
123	3661.5	17.6154	7.8193	17.0787	33.2959
1241	4220.5	22.8152	8.7390	13.8283	89.4555
503	3851.5	17.5635	7.7449	12.7779	18.4678
466	3833.0	36.3161	8.0519	29.7182	346.2676
132	3666.0	16.1130	7.8843	12.5566	73.5115
1765	4482.5	20.7226	7.8193	15.9306	54.7018
1175	4187.5	21.5565	8.2872	12.4448	92.0803
853	4026.5	3.9004	7.9546	4.2983	38.2186
1153	4176.5	14.3046	8.0579	9.6978	104.0814
996	4098.0	21.0242	7.9388	20.0702	30.2634
1474	4337.0	13.9711	7.9963	6.5198	51.9808
1835	4517.5	15.1400	7.7164	4.0195	24.8957
1075	4137.5	3.4587	7.9160	2.5060	45.6947
1874	4537.0	16.0821	7.7419	13.6395	24.2429
1508	4354.0	15.1167	7.8675	9.3258	74.3979
1144	4172.0	25.9765	8.9085	18.5633	127.9412
387	3793.5	17.3002	7.8865	15.5588	69.4515
506	3853.0	20.6176	7.6915	14.2105	24.1259
1865	4532.5	18.9917	7.7979	11.8264	37.7947
1813	4506.5	17.2243	7.5689	9.6989	25.5495
1301	4250.5	9.3415	7.9737	8.4069	77.3434

	Photoelectric	Bulk density	Density Correction	Resistivity (Deep)	\
364	4.3361	2.5251	-0.0403	2.3521	

1119	4.9899	2.6753	-0.0504	12.3318
974	4.2036	2.1884	-0.0213	23.6848
481	3.4528	2.4095	0.1149	2.1401
828	4.9711	2.6577	-0.0481	28.2380
361	4.1760	2.4714	-0.0261	1.7617
1628	3.1915	2.3948	0.1004	5.1816
536	3.1478	2.3748	0.0610	2.2041
1548	3.4109	2.5679	0.0194	7.9924
1001	4.8936	2.6411	-0.0136	33.3160
1615	5.1281	2.6751	-0.0288	38.6341
349	4.1685	2.3876	-0.0232	1.4476
628	5.2657	2.6969	-0.0194	15.9217
1061	4.2026	2.6283	-0.0357	17.6943
1375	4.4764	2.6286	0.0072	15.4157
1840	3.3314	2.5409	0.0266	11.5785
1464	4.3495	2.5925	-0.0183	9.6100
1138	3.4125	2.5316	0.1662	5.0319
1815	3.4103	2.5664	0.0175	9.2287
314	4.0526	2.5503	0.0385	3.9673
1760	2.2249	2.3481	0.0659	6.5753
333	4.0356	2.4490	0.0153	1.3428
826	4.2560	2.6367	-0.0572	21.3249
1800	3.4895	2.4644	0.0018	5.7760
299	2.4448	2.1857	0.1635	4.4770
237	5.1485	2.5736	-0.0440	3.5664
1360	3.0824	2.5211	0.1409	6.6349
796	4.9437	2.7026	-0.0385	68.4639
706	4.5253	2.5746	-0.0005	10.7131
1717	1.8925	2.3319	-0.0076	18.5201
...	...	...	...	...
232	4.1277	2.5166	-0.0168	2.7369
186	3.6400	2.5163	0.0104	3.0754
1560	4.2755	2.6551	0.0761	13.3308
751	4.4641	2.5615	0.0337	2.4198
1385	3.9068	2.5160	-0.0390	13.0313
415	3.9646	2.4273	-0.0142	1.3881
1736	2.0216	2.3443	0.0160	10.3836
357	3.8462	2.4457	-0.0125	1.3343
891	4.3833	2.6906	-0.0291	28.5309
123	3.6268	2.4180	-0.0114	1.9938
1241	3.9401	2.4735	0.0636	4.6421
503	3.5776	2.4915	0.0142	1.7516
466	3.2481	2.2018	-0.0379	3.6664
132	4.2511	2.4953	-0.0010	2.0966
1765	2.7254	2.4376	0.0668	7.5877
1175	3.1311	2.4972	0.0638	4.6428
853	4.5709	2.6365	-0.0122	42.9910
1153	3.7230	2.5442	0.1327	4.4839

996	4.0347	2.3668	-0.0352	42.3146
1474	4.0585	2.5985	-0.0208	8.8093
1835	4.0344	2.6413	0.0006	9.4100
1075	5.1801	2.6671	-0.0198	34.2759
1874	2.7631	2.4768	0.0676	6.6375
1508	3.8703	2.5505	-0.0114	9.3526
1144	3.0451	2.3926	0.1272	3.1704
387	4.3497	2.4439	-0.0140	1.1539
506	3.2985	2.4670	0.0098	1.5080
1865	3.0219	2.5078	0.0128	7.0517
1813	3.4470	2.5441	0.0159	9.3731
1301	4.4373	2.5662	0.0640	14.9452

	Resistivity (Medium)	Resistivity (Shallow)	\
364	2.4766	7.0462	
1119	15.4253	34.3674	
974	24.9051	67.3733	
481	2.1240	2.8021	
828	39.9660	162.5017	
361	1.9908	7.2125	
1628	4.3753	7.3549	
536	2.1413	3.0275	
1548	7.9791	10.0815	
1001	42.2157	48.1978	
1615	66.8716	357.9112	
349	1.6294	4.6045	
628	25.3511	42.2548	
1061	20.2806	23.3243	
1375	20.8038	33.2025	
1840	12.9634	39.5160	
1464	11.1224	19.6170	
1138	4.7733	6.3826	
1815	8.2077	9.1339	
314	3.8026	4.3180	
1760	6.5251	9.6687	
333	1.5702	8.1056	
826	26.2664	84.0958	
1800	4.9936	8.5803	
299	3.4048	4.4362	
237	3.5575	3.9146	
1360	6.8426	10.5845	
796	242.2576	303.4082	
706	13.0670	63.7553	
1717	20.6141	25.4153	
...	...	...	
232	3.3762	12.6250	
186	3.2935	4.9207	
1560	17.8225	43.1671	

751	3.0052	32.2290
1385	14.5992	22.8447
415	1.6599	6.5847
1736	10.4896	15.7468
357	1.5675	7.6728
891	52.1829	93.5368
123	2.3950	8.5235
1241	4.9506	5.5887
503	2.0816	7.3872
466	4.6517	6.1567
132	2.4217	10.4500
1765	7.8840	9.5834
1175	4.5267	5.5382
853	62.9819	104.2278
1153	4.3708	8.1251
996	69.0348	57.3894
1474	9.6402	13.6412
1835	10.5373	30.9698
1075	39.1070	119.2575
1874	6.9025	11.9798
1508	9.0291	10.6468
1144	3.0564	5.0676
387	1.4112	8.1776
506	1.6822	5.2301
1865	7.2568	9.2269
1813	9.5325	14.6245
1301	16.1329	35.8731

	Ratio (shallow/deep resistivity)	Spontaneous Potential \
364	-42.8840	6.5761
1119	-40.0607	19.4222
974	-40.8617	-5.8343
481	-10.5339	63.1526
828	-68.4021	40.2980
361	-55.0948	-8.5466
1628	-13.6900	71.6885
536	-12.4074	60.8314
1548	-9.0765	80.6624
1001	-14.4337	26.1104
1615	-87.0124	68.3274
349	-45.2269	-18.9447
628	-38.1497	28.9515
1061	-10.7978	38.4503
1375	-29.9887	66.7202
1840	-47.9807	31.9374
1464	-27.8920	82.1785
1138	-9.2935	69.8563
1815	0.4036	30.1418

314	-3.3106	56.1944
1760	-15.0705	30.9727
333	-70.2686	-15.3677
826	-53.6299	46.1137
1800	-15.4685	31.5184
299	0.3582	54.6334
237	-3.6415	25.3382
1360	-18.2550	76.0086
796	-58.1909	9.1828
706	-69.7141	29.5343
1717	-12.3706	15.0475
...	...	...
232	-59.7571	0.4599
186	-18.3715	53.9348
1560	-45.9266	42.5611
751	-101.2018	-4.1495
1385	-21.9419	59.1830
415	-60.8492	-14.3916
1736	-16.2759	22.9677
357	-68.3730	-15.2831
891	-46.4100	40.3977
123	-56.7847	-8.6133
1241	-7.2533	72.4865
503	-56.2550	-13.4675
466	-20.2597	56.9320
132	-62.7833	-4.1719
1765	-9.1269	46.7169
1175	-6.8934	62.1995
853	-34.6145	40.3073
1153	-23.2356	71.7611
996	-11.9107	9.2151
1474	-17.0920	80.4019
1835	-46.5613	30.1678
1075	-48.7348	18.4674
1874	-23.0802	21.5909
1508	-5.0660	84.5664
1144	-18.3324	68.5238
387	-76.5420	-18.4378
506	-48.6103	-17.6296
1865	-10.5084	36.4206
1813	-17.3880	30.1516
1301	-34.2241	73.4843

	Micro-inverse resistivity (micro log) \
364	18.2895
1119	36.6666
974	8.5468
481	2.5362

828	46.6361
361	7.0766
1628	6.5378
536	2.0710
1548	16.1564
1001	26.5922
1615	31.2269
349	5.7734
628	28.7685
1061	19.0196
1375	29.3510
1840	10.5515
1464	21.3167
1138	4.3026
1815	7.5583
314	5.6050
1760	6.9288
333	6.9830
826	40.9135
1800	7.8121
299	6.2303
237	13.2950
1360	8.7710
796	46.3579
706	13.0928
1717	8.3810
...	...
232	6.9938
186	8.3021
1560	11.1532
751	6.8663
1385	26.9042
415	6.5178
1736	8.1029
357	6.5906
891	43.1060
123	6.5181
1241	8.0625
503	7.0037
466	12.4429
132	6.6320
1765	9.0714
1175	2.4682
853	16.7749
1153	11.9123
996	9.2681
1474	15.3013
1835	7.9669

```
1075                                  45.3930
1874                                   5.9033
1508                                  12.5307
1144                                   2.3328
387                                    7.5977
506                                    6.1828
1865                                   9.9524
1813                                   8.6686
1301                                  23.8282

      Micro-normal resistivity (micro log)  \
364                                 15.8936
1119                                27.7671
974                                 14.4710
481                                  2.0344
828                                 28.2133
361                                  9.8464
1628                                 4.3007
536                                  2.0939
1548                                 9.9267
1001                                26.8800
1615                                33.8984
349                                  8.6306
628                                 25.2013
1061                                15.9674
1375                                18.1148
1840                                13.6926
1464                                15.1712
1138                                 3.9592
1815                                 9.6560
314                                  5.1371
1760                                 8.5544
333                                  9.7513
826                                 26.4104
1800                                 8.4219
299                                  3.9414
237                                 10.3351
1360                                 9.8405
796                                 36.8581
706                                 21.0197
1717                                12.3823
...                                     ...
232                                 10.8588
186                                  6.5024
1560                                18.4944
751                                 11.8451
1385                                17.5907
415                                  9.8126
```

1736	10.4732
357	10.0096
891	24.9425
123	9.8438
1241	6.6304
503	10.8309
466	4.9666
132	10.2774
1765	7.6158
1175	3.2066
853	20.5747
1153	7.4445
996	15.0629
1474	12.8145
1835	14.5692
1075	38.9610
1874	7.9031
1508	9.3901
1144	2.3470
387	10.2225
506	8.4563
1865	9.8431
1813	11.2534
1301	18.0074

	Delta-t (interval transit time, or slowness)	Sonic porosity \
364	68.4875	14.7719
1119	53.3764	4.0851
974	70.9938	16.5444
481	91.0397	30.7212
828	54.2995	4.7380
361	69.4630	15.4618
1628	90.8278	30.5713
536	90.5111	30.3473
1548	68.7938	14.9886
1001	57.1257	6.7367
1615	53.7795	4.3702
349	68.2359	14.5940
628	54.0743	4.5787
1061	61.6668	9.9483
1375	61.5905	9.8943
1840	60.8066	9.3399
1464	63.7141	11.3961
1138	88.7486	29.1009
1815	63.9358	11.5529
314	75.8742	19.9959
1760	74.8143	19.2463
333	70.2990	16.0530

826	57.6426	7.1023
1800	71.8308	17.1364
299	82.5548	24.7205
237	66.3715	13.2755
1360	74.3790	18.9385
796	50.9018	2.3351
706	57.0577	6.6886
1717	67.8137	14.2954
...	...	...
232	65.5922	12.7243
186	75.2197	19.5330
1560	55.4824	5.5745
751	61.1390	9.5750
1385	64.2034	11.7421
415	69.0536	15.1723
1736	67.1670	13.8381
357	70.0429	15.8719
891	53.8982	4.4542
123	66.9476	13.6829
1241	78.9483	22.1700
503	61.1229	9.5636
466	99.1068	36.4263
132	69.9868	15.8322
1765	71.6997	17.0437
1175	81.1048	23.6950
853	53.3848	4.0911
1153	78.5368	21.8790
996	64.3077	11.8159
1474	67.8789	14.3415
1835	59.9817	8.7565
1075	52.3719	3.3747
1874	65.8447	12.9029
1508	72.6184	17.6934
1144	102.7420	38.9971
387	69.8499	15.7354
506	62.8798	10.8060
1865	70.0454	15.8737
1813	62.7647	10.7247
1301	62.6937	10.6744

	Type of Formation
364	limestone
1119	limestone
974	sandy limestone
481	shale
828	limestone
361	limestone
1628	shale

```
536    shaly limestone
1548   shaly limestone
1001         limestone
1615         limestone
349          limestone
628          limestone
1061         limestone
1375         limestone
1840          dolomite
1464         limestone
1138             shale
1815          dolomite
314              shale
1760         sandstone
333          limestone
826          limestone
1800          dolomite
299              shale
237          limestone
1360             shale
796          limestone
706          limestone
1717         sandstone
...                ...
232          limestone
186    shaly limestone
1560         limestone
751          limestone
1385         limestone
415          limestone
1736         sandstone
357          limestone
891          limestone
123    shaly limestone
1241   shaly limestone
503          limestone
466              shale
132    shaly limestone
1765   shaly sandstone
1175   shaly limestone
853    shaly limestone
1153   shaly limestone
996    sandy limestone
1474         limestone
1835          dolomite
1075         limestone
1874          dolomite
1508         limestone
```

```
1144             shale
387          limestone
506          limestone
1865          dolomite
1813          dolomite
1301             shale

[1390 rows x 18 columns]
```

In [17]:
```
strat_test_set
```

Out[17]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
1501  4350.5           14.3377   7.9358            8.1210    53.0320
377   3788.5           21.5996   7.7935           20.2126    31.8369
1025  4112.5            9.2560   8.0716            6.8427    63.5457
819   4009.5            9.8961   8.1299            6.8641    45.8450
1364  4282.0           25.5932   9.6504           32.0067   179.7232
652   3926.0           12.0566   7.6617            7.8241    21.1828
196   3698.0           14.6286   8.5282            6.4527    43.7186
931   4065.5            2.4583   8.0370            2.4487    33.2769
737   3968.5           13.5330   8.0725            6.2675    36.5269
460   3830.0           17.7343   7.9395           14.6574    84.5551
254   3727.0           15.0385   7.9101           14.6463    30.5299
1421  4310.5           10.1931   7.9195            5.5717    36.9000
1221  4210.5           30.0377   8.4605           23.5010   101.8768
1311  4255.5           31.3822   8.3199           16.5961    79.2960
1349  4274.5           11.8592   7.8710            6.4113    51.5693
1065  4132.5            5.7913   7.9125            3.9936    43.8034
884   4042.0            5.6622   7.9475            3.2034    31.8415
745   3972.5           25.2338   7.4773           17.9663    32.5635
901   4050.5           11.6410   8.0270            4.5262   141.9610
675   3937.5            1.5691   8.0647            0.0196    28.3934
19    3609.5           12.3970   7.9007           11.8429    30.9422
160   3680.0           20.8786   7.7864           21.3394    44.4679
294   3747.0            7.6151   8.0187            4.7423    44.7744
1395  4297.5            9.1259   7.8725            5.2877    32.6802
1381  4290.5           10.3767   7.9642            9.6205    34.4115
1467  4333.5           11.0900   7.9757            9.8634    40.7693
1633  4416.5            8.7986   7.9703            3.5815    98.2391
346   3773.0           17.5967   7.7756           16.1116    28.5177
486   3843.0           22.7174   9.0800           16.2860   106.3980
1768  4484.0           19.9964   7.7094           20.5075    47.2963
...      ...               ...      ...               ...        ...
10    3605.0           13.1647   7.8476            8.6937    35.1545
266   3733.0           15.1945   7.9337           11.4821    29.7505
638   3919.0            9.6472   8.4362            3.2826    49.4102
957   4078.5           34.0657  10.5603           23.0875    80.2997
1749  4474.5           14.6817   7.5154           25.8923    26.8586
```

1270	4235.0	14.6026	7.9551	8.5329	50.1326
21	3610.5	12.0639	8.0506	8.1908	51.7473
1758	4479.0	14.3895	7.7148	22.4755	23.4189
283	3741.5	10.0940	8.0915	3.4155	42.8376
712	3956.0	5.9888	8.1760	4.2916	41.7935
1051	4125.5	9.9211	7.9341	5.6138	40.3269
47	3623.5	11.1560	8.2353	3.0855	53.5208
971	4085.5	25.7429	7.6236	27.2005	36.4536
1609	4404.5	7.9618	7.8377	2.7822	46.5948
1754	4477.0	14.2597	7.5532	24.8003	22.2541
1580	4390.0	20.2663	8.6059	17.6999	142.4401
1002	4101.0	6.0306	8.0268	3.9722	34.4836
1342	4271.0	10.7720	7.7700	7.9436	42.0334
683	3941.5	5.0685	7.9936	2.6732	39.6291
432	3816.0	15.5509	8.0978	12.3106	66.4121
1706	4453.0	14.8605	8.1353	15.8127	70.4406
394	3797.0	23.8342	7.7421	21.1272	55.2595
633	3916.5	8.0507	7.8160	7.4358	25.4477
1623	4411.5	11.3751	8.1142	9.3672	102.6024
1742	4471.0	13.1567	7.8342	22.5157	23.3592
1078	4139.0	5.2280	7.8194	7.9286	44.3974
108	3654.0	20.3017	7.8869	16.2622	36.4418
1839	4519.5	10.7933	7.7783	6.2375	19.3418
1642	4421.0	28.1864	7.7532	20.7235	110.4910
368	3784.0	14.4520	8.0910	10.9635	57.6896

	Photoelectric	Bulk density	Density Correction	Resistivity (Deep) \
1501	4.1425	2.5711	-0.0126	8.2467
377	3.9984	2.3644	-0.0032	1.3436
1025	3.9832	2.5930	0.1195	8.1046
819	4.8488	2.5926	-0.0653	18.2810
1364	2.5904	2.1627	0.1948	4.4241
652	4.3524	2.5762	0.0395	2.9945
196	4.7996	2.5997	-0.0175	4.7205
931	5.2405	2.6681	-0.0654	29.3896
737	4.3165	2.6028	0.0100	4.9044
460	3.7342	2.4594	-0.0003	2.0027
254	4.3720	2.4595	-0.0413	1.9171
1421	4.2040	2.6147	-0.0181	12.7388
1221	3.0548	2.3081	0.0527	2.8815
1311	3.3812	2.4262	0.1414	3.1820
1349	4.2451	2.6004	-0.0044	13.0474
1065	4.6680	2.6417	-0.0336	22.1030
884	4.7946	2.6552	-0.0197	17.9695
745	2.9640	2.4028	0.0043	1.4256
901	4.4666	2.6326	-0.0330	17.0283
675	5.3264	2.7407	-0.0243	20.4190
19	4.3645	2.5075	-0.0134	4.0230

160	4.6043	2.3451	-0.0437	1.1113
294	4.9023	2.6289	-0.0234	8.9137
1395	4.2531	2.6196	0.0109	14.2731
1381	4.1795	2.5455	-0.0123	13.4846
1467	3.9629	2.5413	-0.0323	9.2973
1633	4.7660	2.6488	0.0541	7.1312
346	4.1382	2.4345	0.0210	1.8731
486	3.4400	2.4315	0.1344	2.1254
1768	2.1726	2.3593	0.0865	7.9016
...	...	...	...	...
10	4.2130	2.5613	-0.0287	2.5309
266	5.1702	2.5137	-0.0180	3.1269
638	4.3171	2.6539	0.0520	9.9862
957	3.5195	2.3152	0.1179	4.0925
1749	1.7140	2.2672	0.0786	6.6162
1270	3.7550	2.5641	0.0965	11.8545
21	4.7104	2.5699	-0.0005	3.9582
1758	2.1543	2.3257	0.0540	6.4255
283	5.2038	2.6516	-0.0187	5.2362
712	5.2929	2.6366	-0.0473	18.3724
1051	4.5502	2.6140	-0.0182	14.3166
47	4.5919	2.6572	-0.0250	6.6012
971	3.9306	2.2449	0.0077	23.5393
1609	4.6767	2.6624	-0.0378	10.1454
1754	1.6769	2.2859	0.0690	6.2596
1580	3.3534	2.4073	0.1903	4.0420
1002	4.8197	2.6421	-0.0070	31.7826
1342	3.8711	2.5742	0.0284	11.4063
683	4.5907	2.6643	-0.0061	24.2134
432	4.3195	2.4995	-0.0545	6.5663
1706	2.4431	2.4396	0.0606	10.5557
394	4.1621	2.3487	0.0007	0.9333
633	3.6743	2.5828	0.0405	16.4093
1623	3.6038	2.5498	0.1297	7.7633
1742	1.9073	2.3250	-0.0025	8.3858
1078	4.6249	2.5744	-0.0191	33.6172
108	3.9400	2.4319	-0.0170	1.1335
1839	3.7269	2.6033	0.0327	12.1791
1642	2.8518	2.3556	0.1074	4.8622
368	3.7277	2.5225	-0.0093	3.1108

	Resistivity (Medium)	Resistivity (Shallow)	\
1501	8.7727	13.0996	
377	1.4921	5.5935	
1025	7.6471	26.1792	
819	18.2213	26.1286	
1364	3.5417	4.1200	
652	4.0550	19.9778	

196	5.3410	6.5978
931	93.2097	171.9868
737	5.7088	7.4236
460	2.3486	7.5432
254	2.2322	9.7655
1421	14.1036	18.3011
1221	2.8808	3.8963
1311	3.2132	4.7288
1349	11.9505	13.1527
1065	36.1147	42.9430
884	24.3283	42.6080
745	1.3194	3.3252
901	14.8293	40.6426
675	67.2340	226.3224
19	4.4854	11.3663
160	1.3050	5.1712
294	15.1400	27.8127
1395	15.0177	22.8830
1381	14.2135	17.9459
1467	10.4436	13.8357
1633	10.1490	63.2648
346	2.0974	5.2837
486	2.1327	2.9900
1768	9.0045	11.8906
...	...	...
10	3.7152	20.9967
266	3.4862	13.8435
638	12.1101	22.4591
957	3.6348	5.7601
1749	6.4532	8.2469
1270	12.6029	28.5582
21	4.3331	12.3555
1758	6.6867	11.1896
283	6.1783	17.4660
712	17.5236	41.9868
1051	14.9045	21.3636
47	7.4551	10.1474
971	28.4299	52.9245
1609	12.6937	84.8631
1754	6.3875	10.1342
1580	3.5888	5.7052
1002	38.4689	43.5237
1342	12.6349	20.6063
683	55.8310	108.9264
432	7.4698	14.3176
1706	11.0783	11.8203
394	1.0068	3.7800
633	19.1752	37.9988

1623	6.6962	7.5727
1742	8.4141	12.1470
1078	39.7794	63.8913
108	1.2682	4.7878
1839	15.8295	94.1977
1642	4.0943	6.2705
368	3.2071	4.9261

	Ratio (shallow/deep resistivity)	Spontaneous Potential \
1501	-18.0879	83.4069
377	-55.7457	-7.8881
1025	-45.8304	59.9619
819	-13.9605	52.6570
1364	2.7839	71.7486
652	-74.1805	13.4047
196	-13.0871	46.7625
931	-69.0572	26.5819
737	-16.2027	32.0076
460	-51.8342	30.4475
254	-63.6349	-7.6460
1421	-14.1614	75.8793
1221	-11.7924	65.2676
1311	-15.4844	73.6335
1349	-0.3144	74.6296
1065	-25.9597	33.1183
884	-33.7461	36.7995
745	-33.1045	-7.8571
901	-34.0028	55.0595
675	-94.0224	31.1388
19	-40.5962	14.5853
160	-60.0974	-12.8783
294	-44.4767	50.5661
1395	-18.4496	62.9638
1381	-11.1713	60.6715
1467	-15.5381	80.9590
1633	-85.3197	68.8934
346	-40.5333	-13.0543
486	-13.3404	65.7432
1768	-15.9741	45.3423
...	...	...
10	-82.6992	-5.8998
266	-58.1514	8.2170
638	-31.6792	43.3761
957	-13.3604	56.5730
1749	-8.6115	26.3589
1270	-34.3663	63.5996
21	-44.4930	25.6328
1758	-21.6816	26.6388

283	-47.0864	42.7228
712	-32.3053	21.1977
1051	-15.6449	49.3887
47	-16.8060	60.0656
971	-31.6677	3.8536
1609	-83.0204	72.3693
1754	-18.8318	25.0445
1580	-13.4703	77.1694
1002	-12.2883	29.6639
1342	-23.1172	73.1189
683	-58.7769	30.2756
432	-30.4698	11.3494
1706	-4.4228	51.7541
394	-54.6718	-17.6313
633	-32.8212	24.3960
1623	0.9715	69.0275
1742	-14.4835	23.9438
1078	-25.0992	21.2974
108	-56.3148	-14.0653
1839	-79.9583	31.7944
1642	-9.9425	60.9551
368	-17.9671	28.9817

	Micro-inverse resistivity (micro log) \
1501	14.2095
377	6.3832
1025	19.7475
819	23.2709
1364	2.5252
652	6.9916
196	9.9384
931	49.7344
737	13.0180
460	13.1563
254	7.1033
1421	16.9813
1221	2.2601
1311	2.6640
1349	11.4939
1065	28.1719
884	33.0845
745	4.7577
901	23.6919
675	38.5655
19	5.8787
160	5.0597
294	25.3110
1395	21.9365

1381	19.0563
1467	14.6088
1633	22.1897
346	6.8134
486	3.3482
1768	6.4592
...	...
10	7.1417
266	6.6174
638	16.6218
957	1.7424
1749	5.8258
1270	31.7403
21	12.5952
1758	6.2642
283	11.5749
712	43.0720
1051	20.0936
47	12.2094
971	7.9487
1609	11.1796
1754	6.5826
1580	3.5175
1002	24.6422
1342	18.5275
683	27.3422
432	28.2723
1706	7.5985
394	5.8927
633	11.8541
1623	9.7125
1742	6.6405
1078	39.5627
108	5.6442
1839	13.1841
1642	2.3440
368	17.3865

	Micro-normal resistivity (micro log) \
1501	11.5789
377	9.2743
1025	15.6692
819	15.0720
1364	1.8596
652	11.1765
196	9.0382
931	43.0881
737	9.4111

460	12.1822
254	10.4282
1421	13.7167
1221	2.0336
1311	2.3696
1349	9.9000
1065	24.5998
884	20.8163
745	5.9738
901	25.0663
675	38.8312
19	9.9533
160	8.0840
294	21.0285
1395	18.3494
1381	14.1501
1467	9.2011
1633	25.5828
346	9.5236
486	2.7988
1768	7.0856
...	...
10	11.4769
266	10.4764
638	19.3192
957	1.9408
1749	7.8035
1270	16.5051
21	12.8044
1758	8.2761
283	14.0470
712	36.3040
1051	18.0599
47	7.6260
971	13.9102
1609	16.2093
1754	8.5389
1580	3.2629
1002	23.7191
1342	14.8896
683	37.9857
432	22.0969
1706	7.9203
394	7.5896
633	14.8861
1623	9.8969
1742	9.6395
1078	32.5814

108	7.7562
1839	15.7351
1642	2.0319
368	9.5538

	Delta-t (interval transit time, or slowness)	Sonic porosity	\
1501	70.6689	16.3146	
377	73.7622	18.5023	
1025	68.4833	14.7690	
819	61.6092	9.9075	
1364	92.2882	31.6041	
652	61.9317	10.1356	
196	65.8218	12.8867	
931	52.0303	3.1332	
737	61.5920	9.8953	
460	78.1573	21.6106	
254	67.6642	14.1897	
1421	63.8359	11.4823	
1221	95.7481	34.0510	
1311	93.8746	32.7260	
1349	67.9489	14.3910	
1065	57.7861	7.2038	
884	56.0489	5.9752	
745	68.2564	14.6085	
901	64.0371	11.6245	
675	53.2927	4.0260	
19	65.4587	12.6299	
160	75.2694	19.5682	
294	59.8692	8.6770	
1395	62.0381	10.2108	
1381	64.0097	11.6052	
1467	61.5716	9.8809	
1633	68.6599	14.8938	
346	67.0902	13.7837	
486	87.2250	28.0233	
1768	68.4128	14.7191	
...	...	...	
10	62.7989	10.7489	
266	66.7525	13.5449	
638	61.2897	9.6815	
957	90.1738	30.1088	
1749	79.8667	22.8195	
1270	64.1010	11.6697	
21	66.9407	13.6780	
1758	74.1527	18.7785	
283	63.3638	11.1484	
712	56.6092	6.3714	
1051	63.4965	11.2422	

47	66.6999	13.5077
971	69.6821	15.6167
1609	62.7396	10.7070
1754	77.7316	21.3095
1580	84.3280	25.9745
1002	57.4356	6.9558
1342	65.0109	12.3132
683	54.0801	4.5828
432	66.5213	13.3814
1706	74.7736	19.2175
394	73.8185	18.5421
633	57.4261	6.9492
1623	75.6819	19.8599
1742	70.4121	16.1330
1078	53.9093	4.4620
108	70.0447	15.8732
1839	58.1464	7.4586
1642	102.9218	39.1243
368	70.6703	16.3156

	Type of Formation
1501	limestone
377	limestone
1025	shale
819	limestone
1364	shale
652	limestone
196	limestone
931	limestone
737	limestone
460	shale
254	limestone
1421	limestone
1221	shaly limestone
1311	shaly limestone
1349	limestone
1065	limestone
884	limestone
745	limestone
901	limestone
675	limestone
19	shaly limestone
160	shaly limestone
294	shaly limestone
1395	limestone
1381	limestone
1467	limestone
1633	shale

```
346           limestone
486               shale
1768    shaly sandstone
...                 ...
10      shaly limestone
266           limestone
638           limestone
957     shaly limestone
1749          sandstone
1270    shaly limestone
21      shaly limestone
1758          sandstone
283     shaly limestone
712           limestone
1051          limestone
47      shaly limestone
971     sandy limestone
1609          limestone
1754          sandstone
1580              shale
1002          limestone
1342          limestone
683           limestone
432           limestone
1706    shaly limestone
394           limestone
633           limestone
1623              shale
1742          sandstone
1078          limestone
108     shaly limestone
1839           dolomite
1642              shale
368           limestone

[515 rows x 18 columns]
```

In [18]:
```
train_objs_num = len(strat_train_set)
train_objs_num
```

Out[18]: 1390

In [19]:
```
dataset = pd.concat(objs=[strat_train_set, strat_test_set], axis=0)
dataset.head()
```

Out[19]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
364   3782.0           14.9090   8.0845           10.8138    37.9302
1119  4159.5            5.4242   7.9526            2.0286    87.9380
974   4087.0           26.8415   7.6343           30.5027    30.2045
```

481	3840.5	29.6743	10.0896	17.5728	155.6782
828	4014.0	3.9354	8.1300	3.0607	38.1546

	Photoelectric	Bulk density	Density Correction	Resistivity (Deep) \
364	4.3361	2.5251	-0.0403	2.3521
1119	4.9899	2.6753	-0.0504	12.3318
974	4.2036	2.1884	-0.0213	23.6848
481	3.4528	2.4095	0.1149	2.1401
828	4.9711	2.6577	-0.0481	28.2380

	Resistivity (Medium)	Resistivity (Shallow) \
364	2.4766	7.0462
1119	15.4253	34.3674
974	24.9051	67.3733
481	2.1240	2.8021
828	39.9660	162.5017

	Ratio (shallow/deep resistivity)	Spontaneous Potential \
364	-42.8840	6.5761
1119	-40.0607	19.4222
974	-40.8617	-5.8343
481	-10.5339	63.1526
828	-68.4021	40.2980

	Micro-inverse resistivity (micro log) \
364	18.2895
1119	36.6666
974	8.5468
481	2.5362
828	46.6361

	Micro-normal resistivity (micro log) \
364	15.8936
1119	27.7671
974	14.4710
481	2.0344
828	28.2133

	Delta-t (interval transit time, or slowness)	Sonic porosity \
364	68.4875	14.7719
1119	53.3764	4.0851
974	70.9938	16.5444
481	91.0397	30.7212
828	54.2995	4.7380

	Type of Formation
364	limestone
1119	limestone

```
974     sandy limestone
481               shale
828           limestone
```

In [20]:
```python
dataset_preprocessed = pd.get_dummies(dataset)
dataset_preprocessed.head()
```

Out[20]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
364   3782.0           14.9090   8.0845           10.8138    37.9302
1119  4159.5            5.4242   7.9526            2.0286    87.9380
974   4087.0           26.8415   7.6343           30.5027    30.2045
481   3840.5           29.6743  10.0896           17.5728   155.6782
828   4014.0            3.9354   8.1300            3.0607    38.1546

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
364          4.3361        2.5251             -0.0403              2.3521
1119         4.9899        2.6753             -0.0504             12.3318
974          4.2036        2.1884             -0.0213             23.6848
481          3.4528        2.4095              0.1149              2.1401
828          4.9711        2.6577             -0.0481             28.2380

      Resistivity (Medium)                ...                 \
364                 2.4766                ...
1119               15.4253                ...
974                24.9051                ...
481                 2.1240                ...
828                39.9660                ...

      Micro-normal resistivity (micro log)  \
364                                15.8936
1119                               27.7671
974                                14.4710
481                                 2.0344
828                                28.2133

      Delta-t (interval transit time, or slowness)  Sonic porosity  \
364                                        68.4875         14.7719
1119                                       53.3764          4.0851
974                                        70.9938         16.5444
481                                        91.0397         30.7212
828                                        54.2995          4.7380

      Type of Formation_dolomite  Type of Formation_limestone  \
364                            0                            1
1119                           0                            1
974                            0                            0
481                            0                            0
828                            0                            1
```

```
      Type of Formation_sandstone  Type of Formation_sandy limestone  \
364                             0                                  0
1119                            0                                  0
974                             0                                  1
481                             0                                  0
828                             0                                  0

      Type of Formation_shale  Type of Formation_shaly limestone  \
364                         0                                  0
1119                        0                                  0
974                         0                                  0
481                         1                                  0
828                         0                                  0

      Type of Formation_shaly sandstone
364                                   0
1119                                  0
974                                   0
481                                   0
828                                   0

[5 rows x 24 columns]
```

In [21]:
```
train_preprocessed = dataset_preprocessed[:train_objs_num]
train_preprocessed.head()
```

Out[21]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
364   3782.0           14.9090   8.0845           10.8138    37.9302
1119  4159.5            5.4242   7.9526            2.0286    87.9380
974   4087.0           26.8415   7.6343           30.5027    30.2045
481   3840.5           29.6743  10.0896           17.5728   155.6782
828   4014.0            3.9354   8.1300            3.0607    38.1546

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
364          4.3361        2.5251             -0.0403              2.3521
1119         4.9899        2.6753             -0.0504             12.3318
974          4.2036        2.1884             -0.0213             23.6848
481          3.4528        2.4095              0.1149              2.1401
828          4.9711        2.6577             -0.0481             28.2380

      Resistivity (Medium)                ...                 \
364                 2.4766                ...
1119               15.4253                ...
974                24.9051                ...
481                 2.1240                ...
828                39.9660                ...
```

```
     Micro-normal resistivity (micro log)  \
364                               15.8936
1119                              27.7671
974                               14.4710
481                                2.0344
828                               28.2133

     Delta-t (interval transit time, or slowness)  Sonic porosity  \
364                                        68.4875         14.7719
1119                                       53.3764          4.0851
974                                        70.9938         16.5444
481                                        91.0397         30.7212
828                                        54.2995          4.7380

     Type of Formation_dolomite  Type of Formation_limestone  \
364                           0                            1
1119                          0                            1
974                           0                            0
481                           0                            0
828                           0                            1

     Type of Formation_sandstone  Type of Formation_sandy limestone  \
364                            0                                  0
1119                           0                                  0
974                            0                                  1
481                            0                                  0
828                            0                                  0

     Type of Formation_shale  Type of Formation_shaly limestone  \
364                        0                                  0
1119                       0                                  0
974                        0                                  0
481                        1                                  0
828                        0                                  0

     Type of Formation_shaly sandstone
364                                  0
1119                                 0
974                                  0
481                                  0
828                                  0

[5 rows x 24 columns]
```

In [22]:
```
test_preprocessed = dataset_preprocessed[train_objs_num:]
test_preprocessed.head()
```

Out[22]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
1501  4350.5           14.3377   7.9358            8.1210    53.0320
```

```
377   3788.5           21.5996   7.7935           20.2126    31.8369
1025  4112.5            9.2560   8.0716            6.8427    63.5457
819   4009.5            9.8961   8.1299            6.8641    45.8450
1364  4282.0           25.5932   9.6504           32.0067   179.7232

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
1501         4.1425        2.5711             -0.0126              8.2467
377          3.9984        2.3644             -0.0032              1.3436
1025         3.9832        2.5930              0.1195              8.1046
819          4.8488        2.5926             -0.0653             18.2810
1364         2.5904        2.1627              0.1948              4.4241

      Resistivity (Medium)                ...                  \
1501                8.7727                ...
377                 1.4921                ...
1025                7.6471                ...
819                18.2213                ...
1364                3.5417                ...

      Micro-normal resistivity (micro log)  \
1501                               11.5789
377                                 9.2743
1025                               15.6692
819                                15.0720
1364                                1.8596

      Delta-t (interval transit time, or slowness)  Sonic porosity  \
1501                                       70.6689         16.3146
377                                        73.7622         18.5023
1025                                       68.4833         14.7690
819                                        61.6092          9.9075
1364                                       92.2882         31.6041

      Type of Formation_dolomite  Type of Formation_limestone  \
1501                           0                            1
377                            0                            1
1025                           0                            0
819                            0                            1
1364                           0                            0

      Type of Formation_sandstone  Type of Formation_sandy limestone  \
1501                            0                                  0
377                             0                                  0
1025                            0                                  0
819                             0                                  0
1364                            0                                  0

      Type of Formation_shale  Type of Formation_shaly limestone  \
```

```
1501                    0                            0
377                     0                            0
1025                    1                            0
819                     0                            0
1364                    1                            0

      Type of Formation_shaly sandstone
1501                                0
377                                 0
1025                                0
819                                 0
1364                                0

[5 rows x 24 columns]
```

5 Preparing the data for Machine Learning algorithms

Creating a copy of training set so that no harm is done to original training set

In [23]:
```python
LogDat_train_new = strat_train_set.copy()
```

In [24]:
```python
train_preprocessed.head()
```

Out[24]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
364   3782.0           14.9090   8.0845           10.8138    37.9302
1119  4159.5            5.4242   7.9526            2.0286    87.9380
974   4087.0           26.8415   7.6343           30.5027    30.2045
481   3840.5           29.6743  10.0896           17.5728   155.6782
828   4014.0            3.9354   8.1300            3.0607    38.1546

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
364          4.3361        2.5251             -0.0403              2.3521
1119         4.9899        2.6753             -0.0504             12.3318
974          4.2036        2.1884             -0.0213             23.6848
481          3.4528        2.4095              0.1149              2.1401
828          4.9711        2.6577             -0.0481             28.2380

      Resistivity (Medium)                ...                   \
364                 2.4766                ...
1119               15.4253                ...
974                24.9051                ...
481                 2.1240                ...
828                39.9660                ...

      Micro-normal resistivity (micro log)  \
364                                15.8936
1119                               27.7671
974                                14.4710
```

```
481                                2.0344
828                               28.2133

      Delta-t (interval transit time, or slowness)  Sonic porosity  \
364                                        68.4875         14.7719
1119                                       53.3764          4.0851
974                                        70.9938         16.5444
481                                        91.0397         30.7212
828                                        54.2995          4.7380

      Type of Formation_dolomite  Type of Formation_limestone  \
364                            0                            1
1119                           0                            1
974                            0                            0
481                            0                            0
828                            0                            1

      Type of Formation_sandstone  Type of Formation_sandy limestone  \
364                             0                                  0
1119                            0                                  0
974                             0                                  1
481                             0                                  0
828                             0                                  0

      Type of Formation_shale  Type of Formation_shaly limestone  \
364                         0                                  0
1119                        0                                  0
974                         0                                  0
481                         1                                  0
828                         0                                  0

      Type of Formation_shaly sandstone
364                                   0
1119                                  0
974                                   0
481                                   0
828                                   0

[5 rows x 24 columns]
```

In [25]:
```
LogDat_train_num = train_preprocessed.drop(train_preprocessed.loc[:,'Type of Formatio
                                                                 'Type of Formation_sl
LogDat_train_num.head()
```

Out[25]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
364   3782.0           14.9090   8.0845           10.8138    37.9302
1119  4159.5            5.4242   7.9526            2.0286    87.9380
974   4087.0           26.8415   7.6343           30.5027    30.2045
```

```
481   3840.5           29.6743  10.0896           17.5728   155.6782
828   4014.0            3.9354   8.1300            3.0607    38.1546

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
364          4.3361        2.5251             -0.0403              2.3521
1119         4.9899        2.6753             -0.0504             12.3318
974          4.2036        2.1884             -0.0213             23.6848
481          3.4528        2.4095              0.1149              2.1401
828          4.9711        2.6577             -0.0481             28.2380

      Resistivity (Medium)  Resistivity (Shallow)  \
364                 2.4766                 7.0462
1119               15.4253                34.3674
974                24.9051                67.3733
481                 2.1240                 2.8021
828                39.9660               162.5017

      Ratio (shallow/deep resistivity)  Spontaneous Potential  \
364                           -42.8840                 6.5761
1119                          -40.0607                19.4222
974                           -40.8617                -5.8343
481                           -10.5339                63.1526
828                           -68.4021                40.2980

      Micro-inverse resistivity (micro log)  \
364                                 18.2895
1119                                36.6666
974                                  8.5468
481                                  2.5362
828                                 46.6361

      Micro-normal resistivity (micro log)  \
364                                15.8936
1119                               27.7671
974                                14.4710
481                                 2.0344
828                                28.2133

      Delta-t (interval transit time, or slowness)  Sonic porosity
364                                        68.4875         14.7719
1119                                       53.3764          4.0851
974                                        70.9938         16.5444
481                                        91.0397         30.7212
828                                        54.2995          4.7380
```

In [26]:
```
LogDat_train_labels = train_preprocessed.drop(train_preprocessed.loc[:,'DEPTH':
                                                                     'Sonic porosity'].hea
LogDat_train_labels.head()
```

Out[26]:
```
       Type of Formation_dolomite  Type of Formation_limestone  \
364                             0                            1
1119                            0                            1
974                             0                            0
481                             0                            0
828                             0                            1

      Type of Formation_sandstone  Type of Formation_sandy limestone  \
364                             0                                  0
1119                            0                                  0
974                             0                                  1
481                             0                                  0
828                             0                                  0

      Type of Formation_shale  Type of Formation_shaly limestone  \
364                         0                                  0
1119                        0                                  0
974                         0                                  0
481                         1                                  0
828                         0                                  0

      Type of Formation_shaly sandstone
364                                   0
1119                                  0
974                                   0
481                                   0
828                                   0
```

In [27]:
```
test_preprocessed.head()
```

Out[27]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
1501  4350.5           14.3377   7.9358            8.1210    53.0320
377   3788.5           21.5996   7.7935           20.2126    31.8369
1025  4112.5            9.2560   8.0716            6.8427    63.5457
819   4009.5            9.8961   8.1299            6.8641    45.8450
1364  4282.0           25.5932   9.6504           32.0067   179.7232

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
1501         4.1425        2.5711             -0.0126              8.2467
377          3.9984        2.3644             -0.0032              1.3436
1025         3.9832        2.5930              0.1195              8.1046
819          4.8488        2.5926             -0.0653             18.2810
1364         2.5904        2.1627              0.1948              4.4241

      Resistivity (Medium)                ...                    \
1501                8.7727                ...
377                 1.4921                ...
1025                7.6471                ...
```

```
819              18.2213              ...
1364              3.5417              ...

      Micro-normal resistivity (micro log)  \
1501                               11.5789
377                                 9.2743
1025                               15.6692
819                                15.0720
1364                                1.8596

      Delta-t (interval transit time, or slowness)  Sonic porosity  \
1501                                       70.6689         16.3146
377                                        73.7622         18.5023
1025                                       68.4833         14.7690
819                                        61.6092          9.9075
1364                                       92.2882         31.6041

      Type of Formation_dolomite  Type of Formation_limestone  \
1501                           0                            1
377                            0                            1
1025                           0                            0
819                            0                            1
1364                           0                            0

      Type of Formation_sandstone  Type of Formation_sandy limestone  \
1501                            0                                  0
377                             0                                  0
1025                            0                                  0
819                             0                                  0
1364                            0                                  0

      Type of Formation_shale  Type of Formation_shaly limestone  \
1501                        0                                  0
377                         0                                  0
1025                        1                                  0
819                         0                                  0
1364                        1                                  0

      Type of Formation_shaly sandstone
1501                                  0
377                                   0
1025                                  0
819                                   0
1364                                  0

[5 rows x 24 columns]
```

```
In [28]: LogDat_test_num = test_preprocessed.drop(test_preprocessed.loc[:,'Type of Formation_d
```

'Type of Formation_s

```
LogDat_test_num.head()
```

Out[28]:

```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
1501  4350.5           14.3377   7.9358            8.1210    53.0320
377   3788.5           21.5996   7.7935           20.2126    31.8369
1025  4112.5            9.2560   8.0716            6.8427    63.5457
819   4009.5            9.8961   8.1299            6.8641    45.8450
1364  4282.0           25.5932   9.6504           32.0067   179.7232

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
1501         4.1425        2.5711             -0.0126              8.2467
377          3.9984        2.3644             -0.0032              1.3436
1025         3.9832        2.5930              0.1195              8.1046
819          4.8488        2.5926             -0.0653             18.2810
1364         2.5904        2.1627              0.1948              4.4241

      Resistivity (Medium)  Resistivity (Shallow)  \
1501                8.7727                13.0996
377                 1.4921                 5.5935
1025                7.6471                26.1792
819                18.2213                26.1286
1364                3.5417                 4.1200

      Ratio (shallow/deep resistivity)  Spontaneous Potential  \
1501                          -18.0879                83.4069
377                           -55.7457                -7.8881
1025                          -45.8304                59.9619
819                           -13.9605                52.6570
1364                            2.7839                71.7486

      Micro-inverse resistivity (micro log)  \
1501                                14.2095
377                                  6.3832
1025                                19.7475
819                                 23.2709
1364                                 2.5252

      Micro-normal resistivity (micro log)  \
1501                               11.5789
377                                 9.2743
1025                               15.6692
819                                15.0720
1364                                1.8596

      Delta-t (interval transit time, or slowness)  Sonic porosity
1501                                       70.6689         16.3146
377                                        73.7622         18.5023
```

```
1025                                            68.4833         14.7690
819                                             61.6092          9.9075
1364                                            92.2882         31.6041
```

```
In [29]: LogDat_test_labels = test_preprocessed.drop(test_preprocessed.loc[:,'DEPTH':
                                                                   'Sonic porosity'].hea
         LogDat_test_labels.head()
```

```
Out[29]:       Type of Formation_dolomite  Type of Formation_limestone  \
         1501                           0                            1
         377                            0                            1
         1025                           0                            0
         819                            0                            1
         1364                           0                            0

               Type of Formation_sandstone  Type of Formation_sandy limestone  \
         1501                            0                                  0
         377                             0                                  0
         1025                            0                                  0
         819                             0                                  0
         1364                            0                                  0

               Type of Formation_shale  Type of Formation_shaly limestone  \
         1501                        0                                  0
         377                         0                                  0
         1025                        1                                  0
         819                         0                                  0
         1364                        1                                  0

               Type of Formation_shaly sandstone
         1501                                  0
         377                                   0
         1025                                  0
         819                                   0
         1364                                  0
```

```
In [30]: LogDat_test_labels.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 515 entries, 1501 to 368
Data columns (total 7 columns):
Type of Formation_dolomite           515 non-null uint8
Type of Formation_limestone          515 non-null uint8
Type of Formation_sandstone          515 non-null uint8
Type of Formation_sandy limestone    515 non-null uint8
Type of Formation_shale              515 non-null uint8
Type of Formation_shaly limestone    515 non-null uint8
Type of Formation_shaly sandstone    515 non-null uint8
dtypes: uint8(7)
```

```
memory usage: 7.5 KB
```

```
In [31]: LogDat_test_labels.sum()
```

```
Out[31]: Type of Formation_dolomite            36
         Type of Formation_limestone          258
         Type of Formation_sandstone           16
         Type of Formation_sandy limestone     10
         Type of Formation_shale               70
         Type of Formation_shaly limestone    123
         Type of Formation_shaly sandstone      2
         dtype: int64
```

```
In [32]: LogDat_train_labels.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1390 entries, 364 to 1301
Data columns (total 7 columns):
Type of Formation_dolomite           1390 non-null uint8
Type of Formation_limestone          1390 non-null uint8
Type of Formation_sandstone          1390 non-null uint8
Type of Formation_sandy limestone    1390 non-null uint8
Type of Formation_shale              1390 non-null uint8
Type of Formation_shaly limestone    1390 non-null uint8
Type of Formation_shaly sandstone    1390 non-null uint8
dtypes: uint8(7)
memory usage: 20.4 KB
```

```
In [33]: LogDat_train_labels.sum()
```

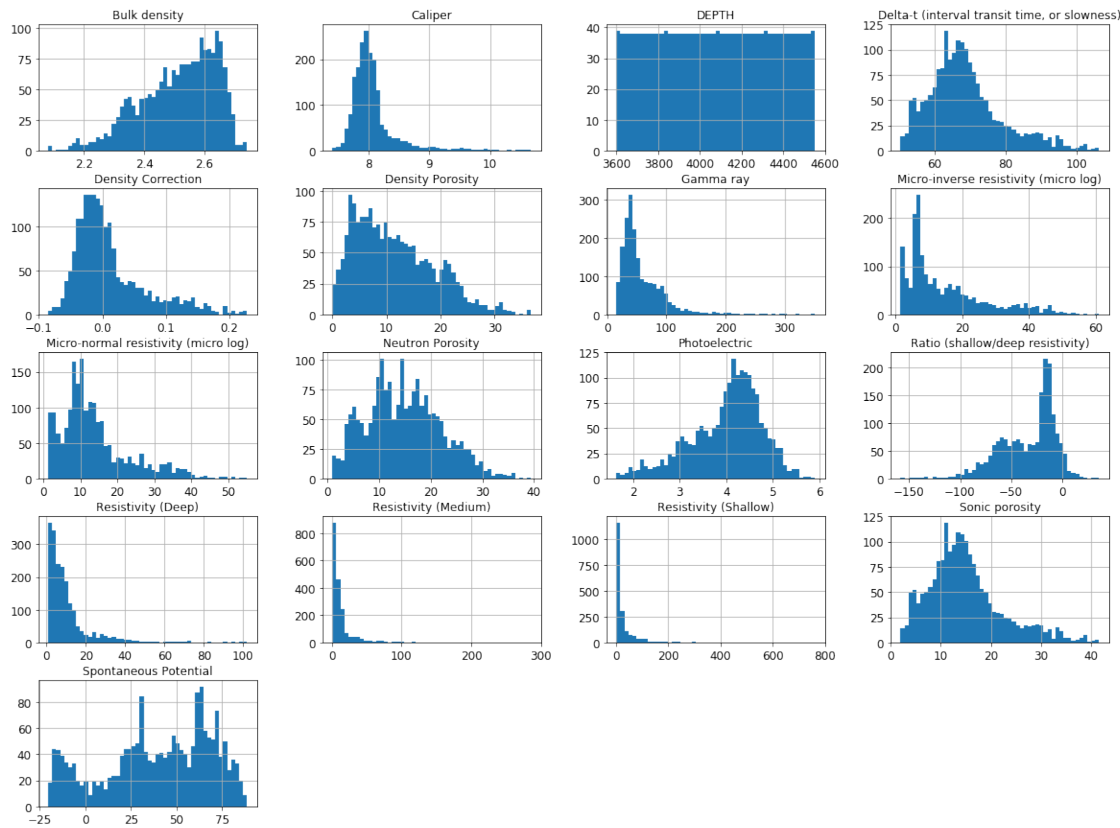
```
Out[33]: Type of Formation_dolomite            96
         Type of Formation_limestone          698
         Type of Formation_sandstone           43
         Type of Formation_sandy limestone     28
         Type of Formation_shale              188
         Type of Formation_shaly limestone    333
         Type of Formation_shaly sandstone      4
         dtype: int64
```

Data Visualization after cleaning the data

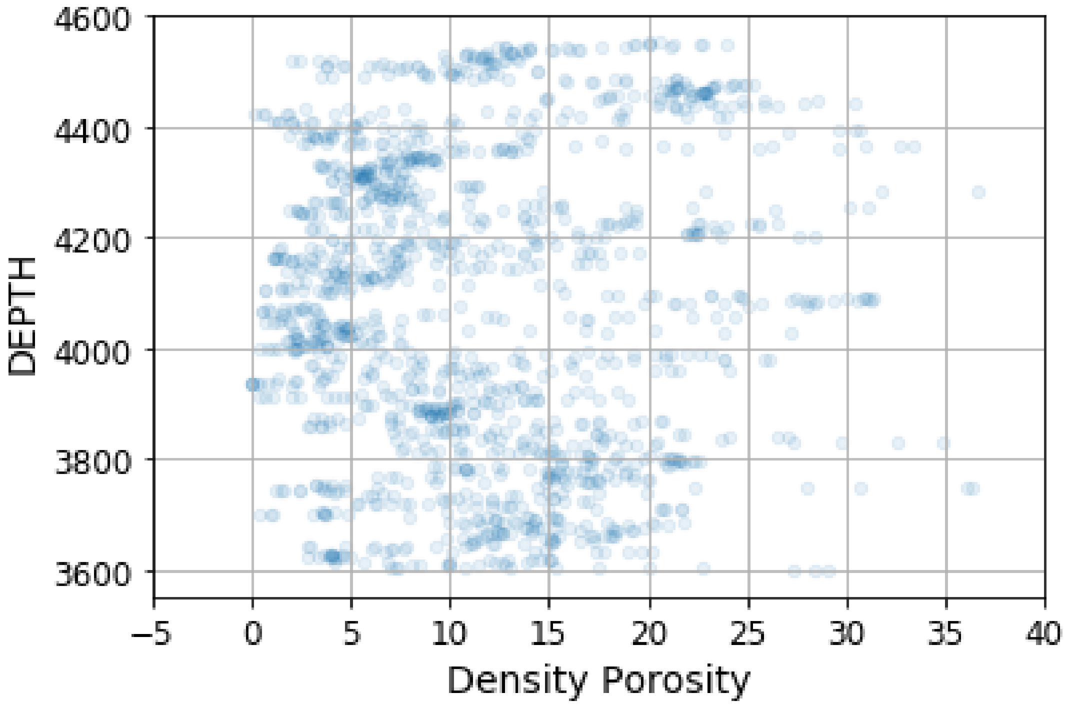
```
In [34]: LogDat_drop.hist(bins=50, figsize=(20,15))
```

```
Out[34]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x10D39490>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x10DC3670>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x10DFE250>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x10E3E470>],
```

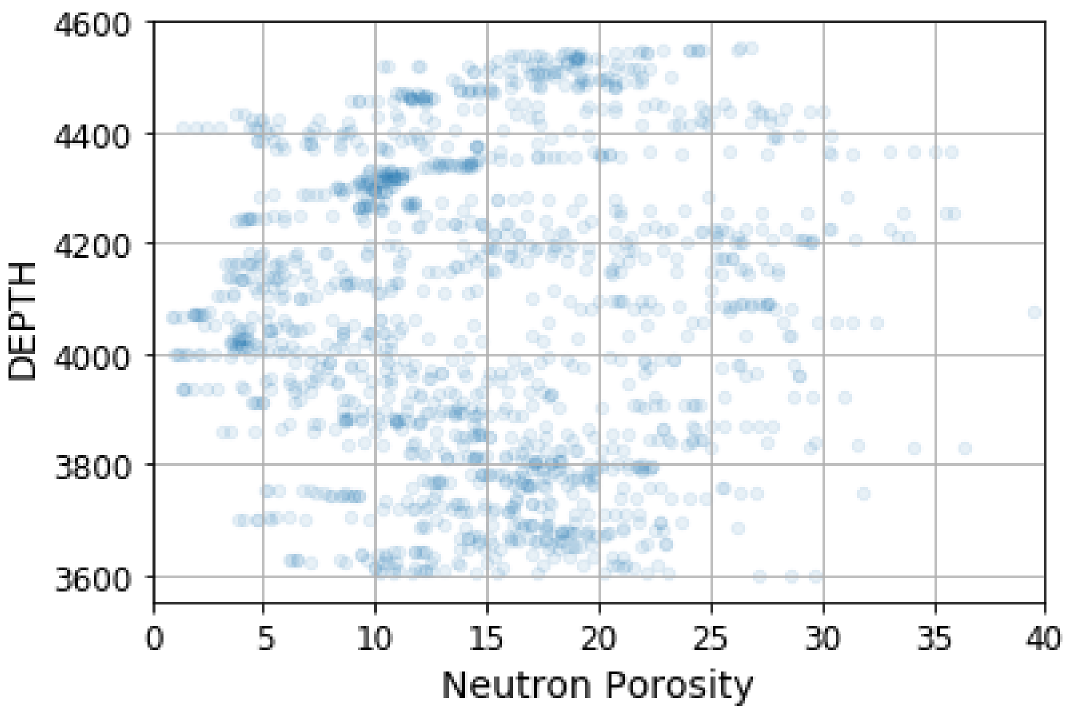
```
[<matplotlib.axes._subplots.AxesSubplot object at 0x10E767D0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x10E98E70>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x10E98CF0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x10F022F0>],
[<matplotlib.axes._subplots.AxesSubplot object at 0x10F712B0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x10FAD3F0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x10FEE5D0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x11025F30>],
[<matplotlib.axes._subplots.AxesSubplot object at 0x110760F0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x110ACE70>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x110D67D0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x11129E70>],
[<matplotlib.axes._subplots.AxesSubplot object at 0x111371D0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x111AA1D0>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x111E3250>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x11225470>]], dtype=object)
```




```
In [35]: LogDat_train_new.plot(kind = "scatter", x ="Density Porosity", y ="DEPTH",grid = True
                               alpha =0.1)

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x110f2db0>
```

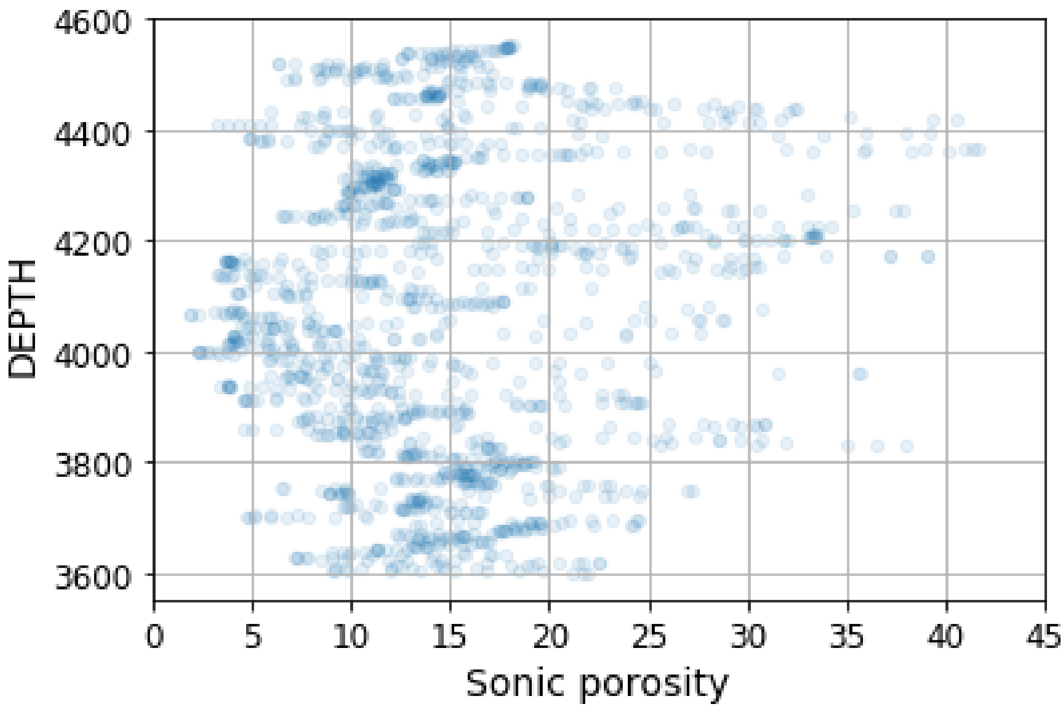



```
In [36]: LogDat_train_new.plot(kind = "scatter", x ="Neutron Porosity", y ="DEPTH",grid = True
         # A number of off-points on the right

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x11ac9730>
```



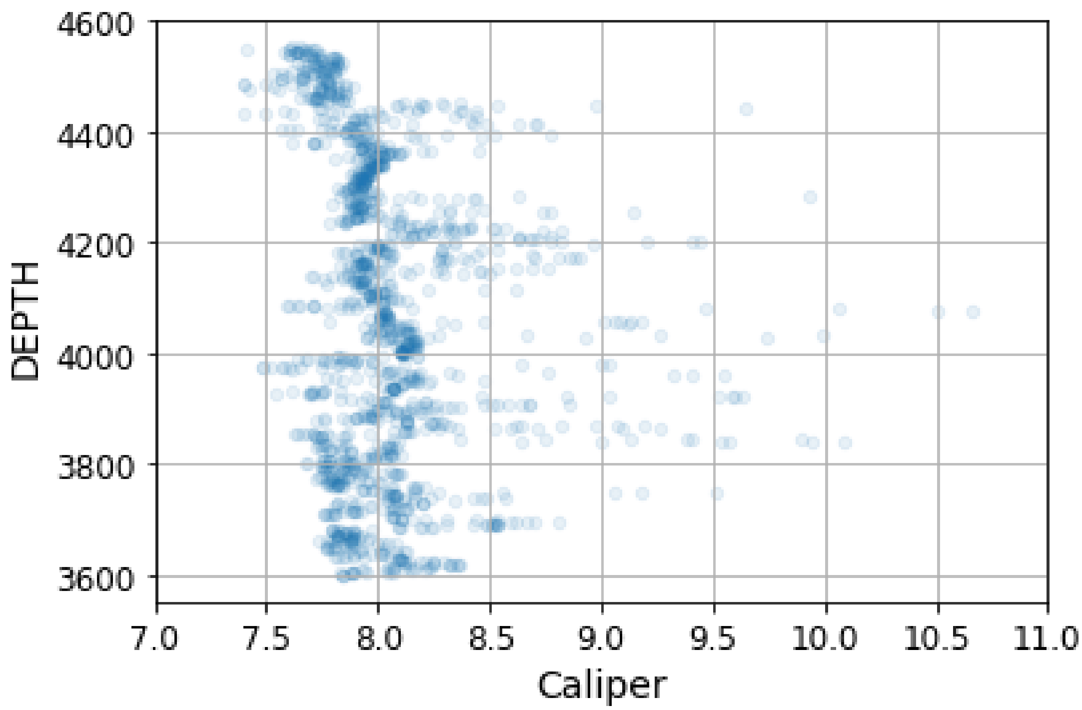
```
In [37]: LogDat_train_new.plot(kind = "scatter", x ="Sonic porosity", y ="DEPTH",grid = True, x
```

```
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x11833ed0>
```

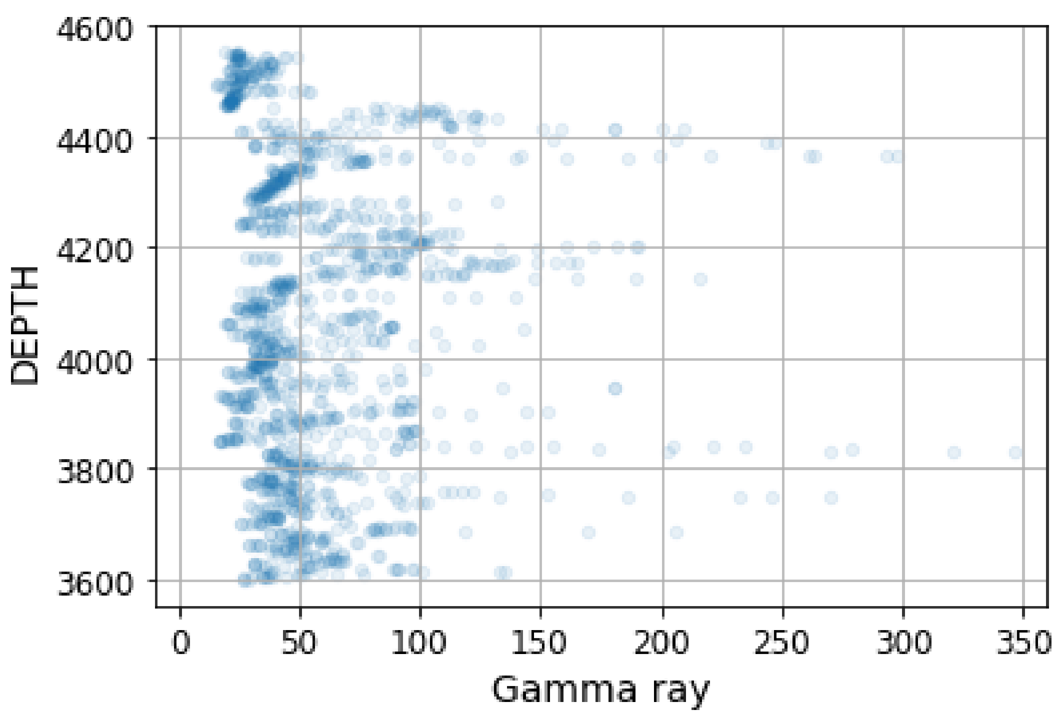



```
In [38]: LogDat_train_new.plot(kind = "scatter", x ="Caliper", y ="DEPTH",grid = True, xlim=[7
```

```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x1104f1f0>
```

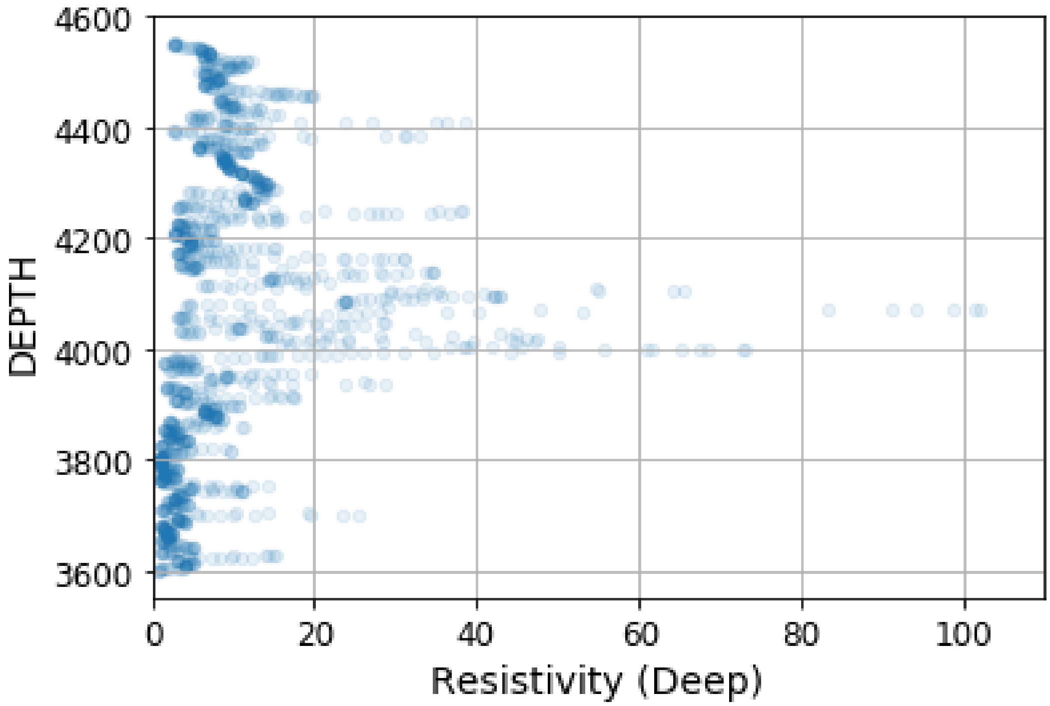



```
In [39]: LogDat_train_new.plot(kind = "scatter", x ="Gamma ray", y ="DEPTH",grid = True, xlim=

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x11603eb0>
```



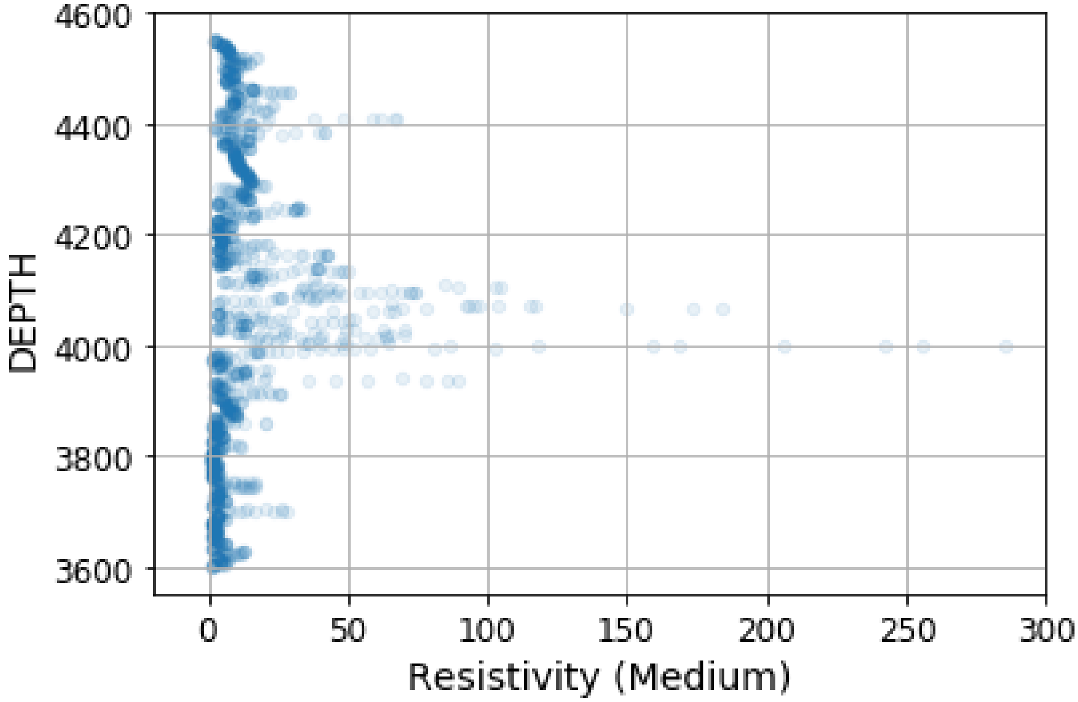
```
In [40]: LogDat_train_new.plot(kind = "scatter", x ="Resistivity (Deep)", y ="DEPTH",grid = Tru
                              ,alpha =0.1)
```

```
Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x11c24470>
```

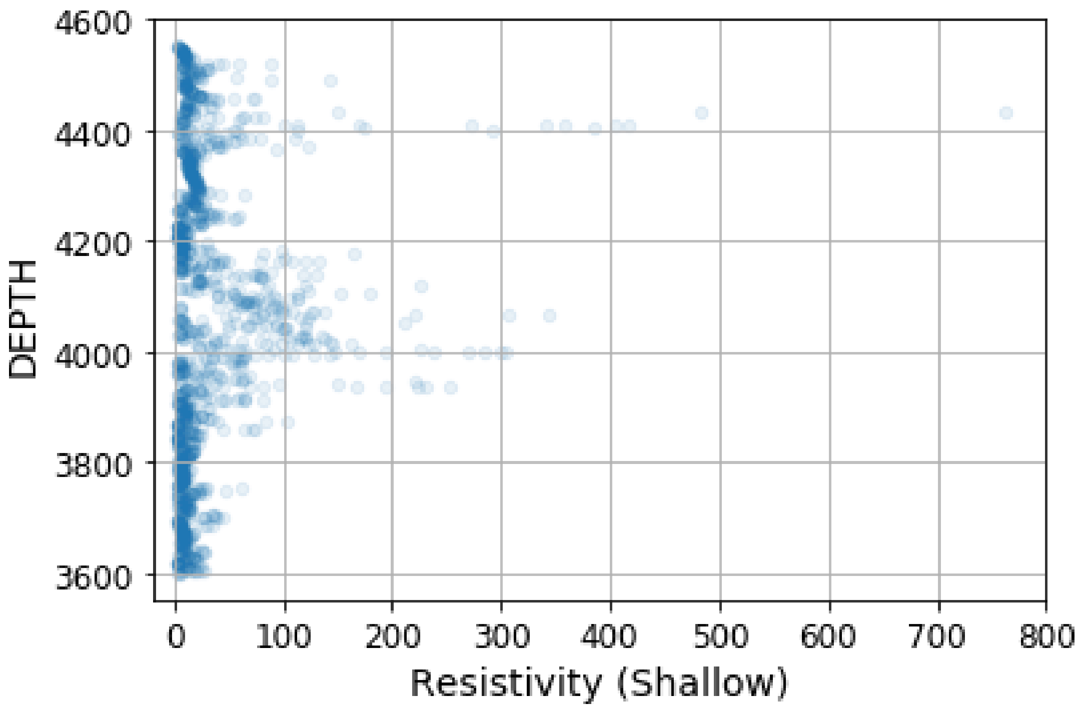



```
In [41]: LogDat_train_new.plot(kind = "scatter", x ="Resistivity (Medium)", y ="DEPTH",grid = 
                              ,alpha =0.1)
```

```
Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x11c748d0>
```

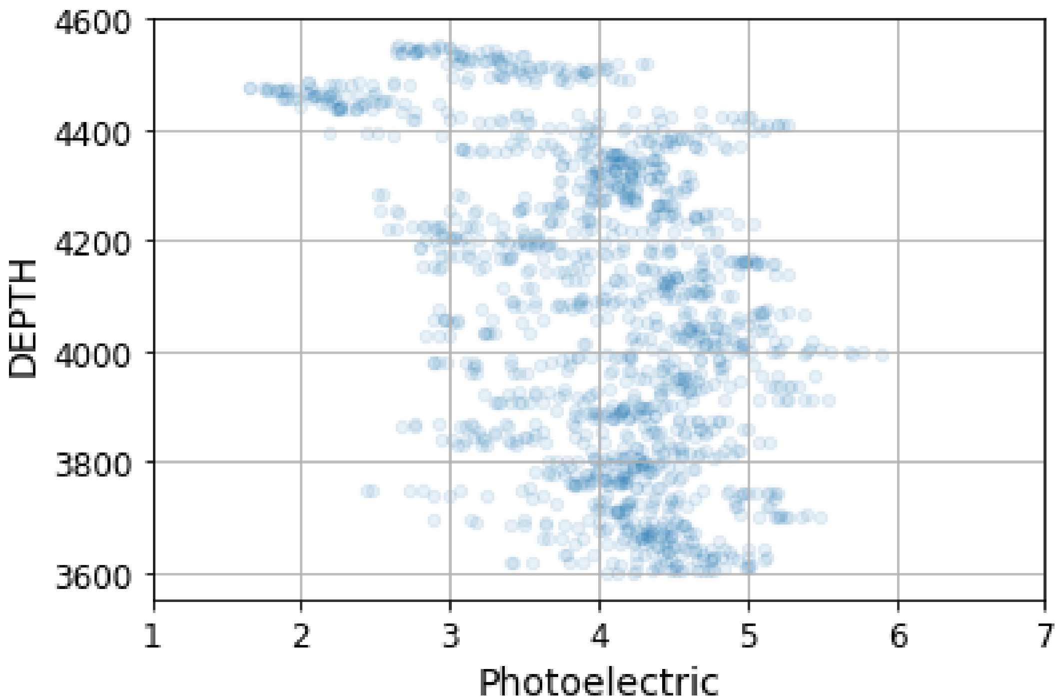



```
In [42]: LogDat_train_new.plot(kind = "scatter", x ="Resistivity (Shallow)", y ="DEPTH",grid =
                                   ,alpha =0.1)

Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x11c8cfd0>
```



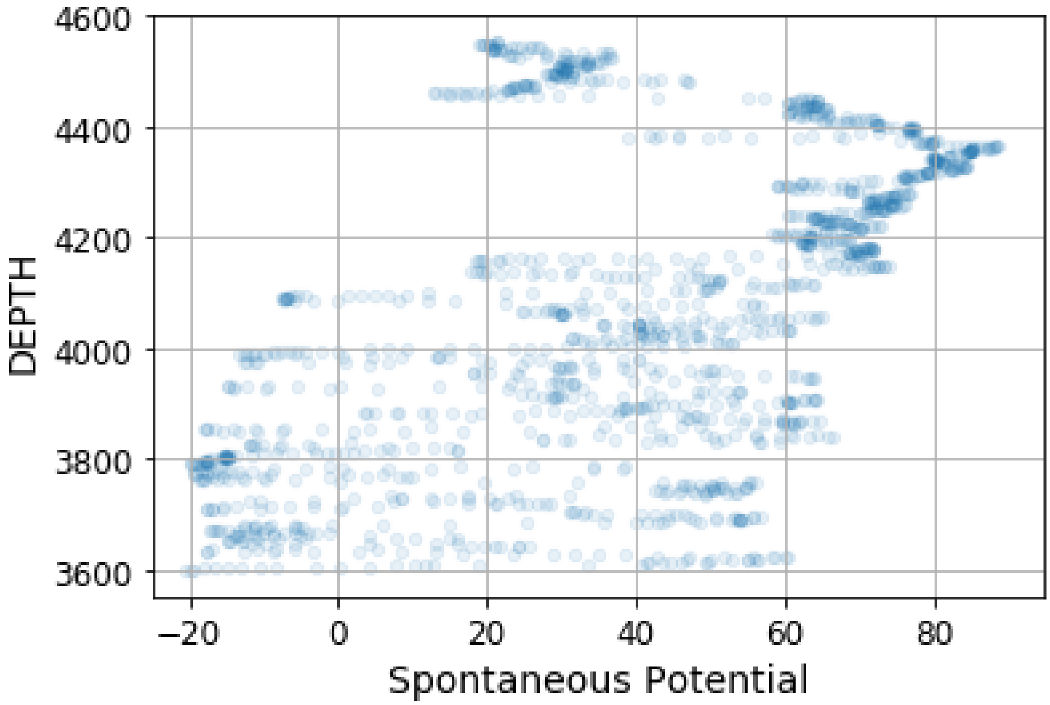
```
In [43]: LogDat_train_new.plot(kind = "scatter", x ="Photoelectric", y ="DEPTH",grid = True, xl
```

```
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x11d989f0>
```



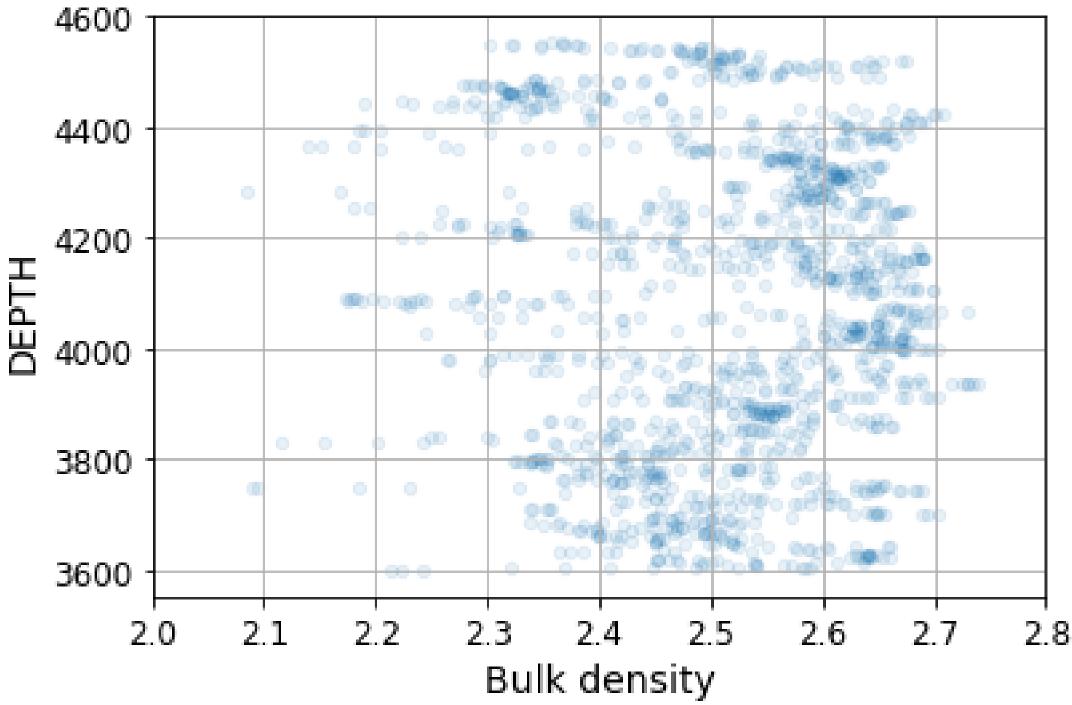

```
In [44]: LogDat_train_new.plot(kind = "scatter", x ="Spontaneous Potential", y ="DEPTH",grid =
```

```
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x11de2f70>
```



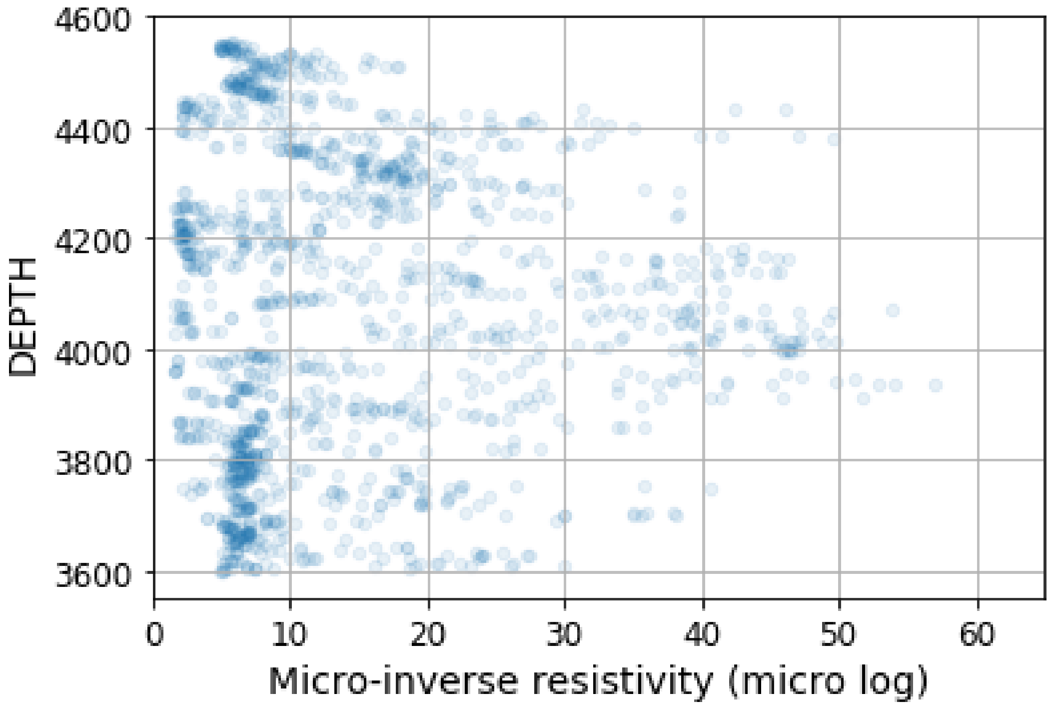

```
In [45]: LogDat_train_new.plot(kind = "scatter", x ="Bulk density", y ="DEPTH",grid = True, xli
```

```
Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x11dd5830>
```



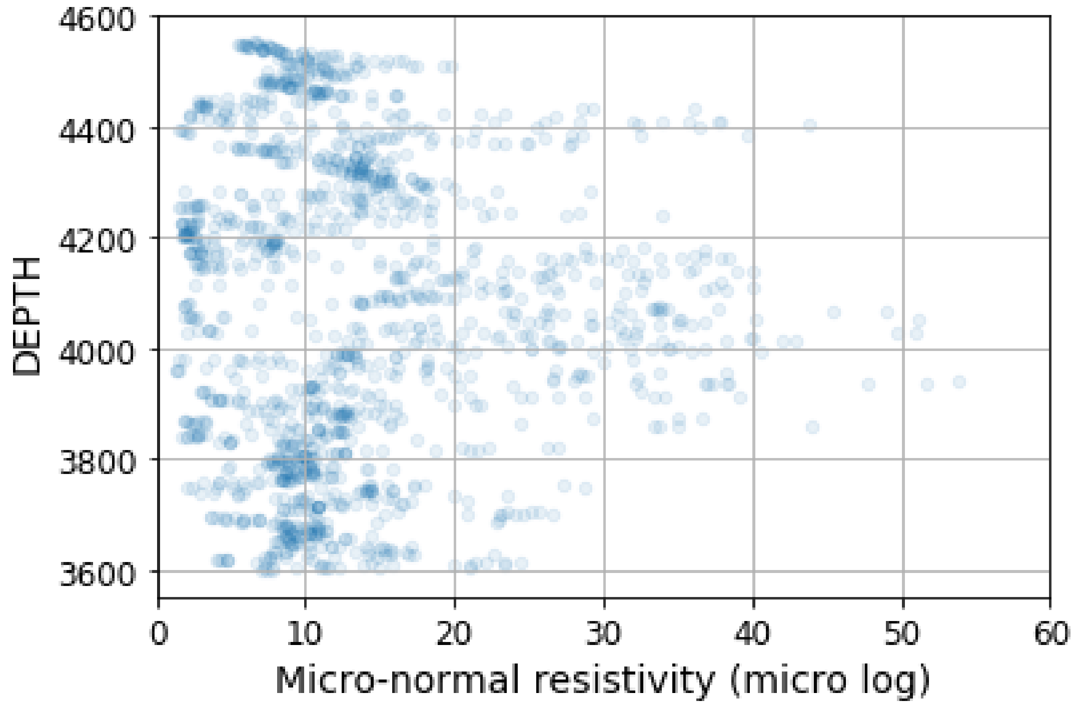
```
In [46]: LogDat_train_new.plot(kind = "scatter", x ="Micro-inverse resistivity (micro log)", y
                             ,alpha=0.1) #Off-point on the left again in this log as well
```

```
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x11f7d590>
```

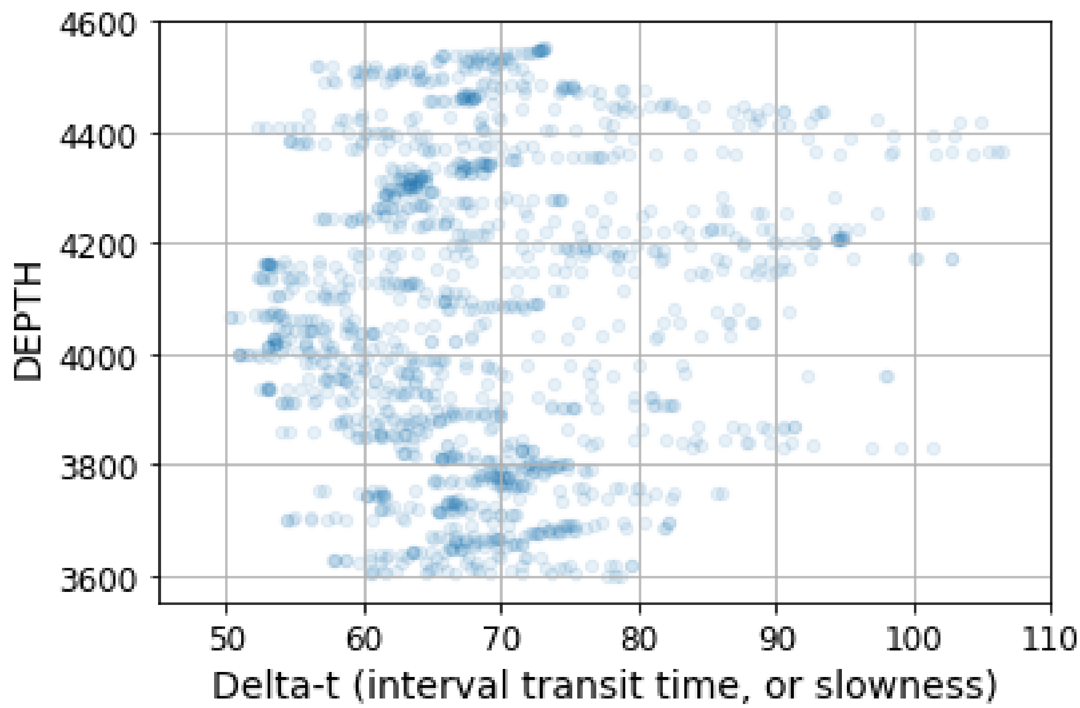



```
In [47]: LogDat_train_new.plot(kind = "scatter", x ="Micro-normal resistivity (micro log)", y =
                             ,alpha=0.1)
```

```
Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x11e65cd0>
```

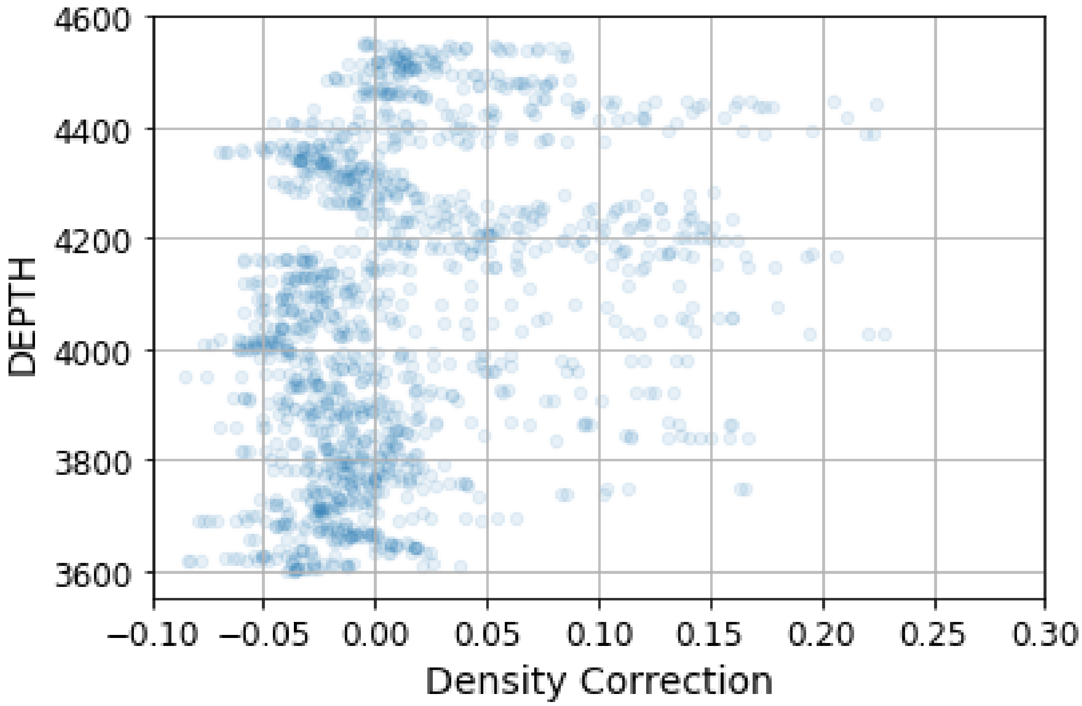



```
In [48]: LogDat_train_new.plot(kind = "scatter", x ="Delta-t (interval transit time, or slownes
                                     xlim=[45,110], alpha=0.1)

Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x11ef2130>
```



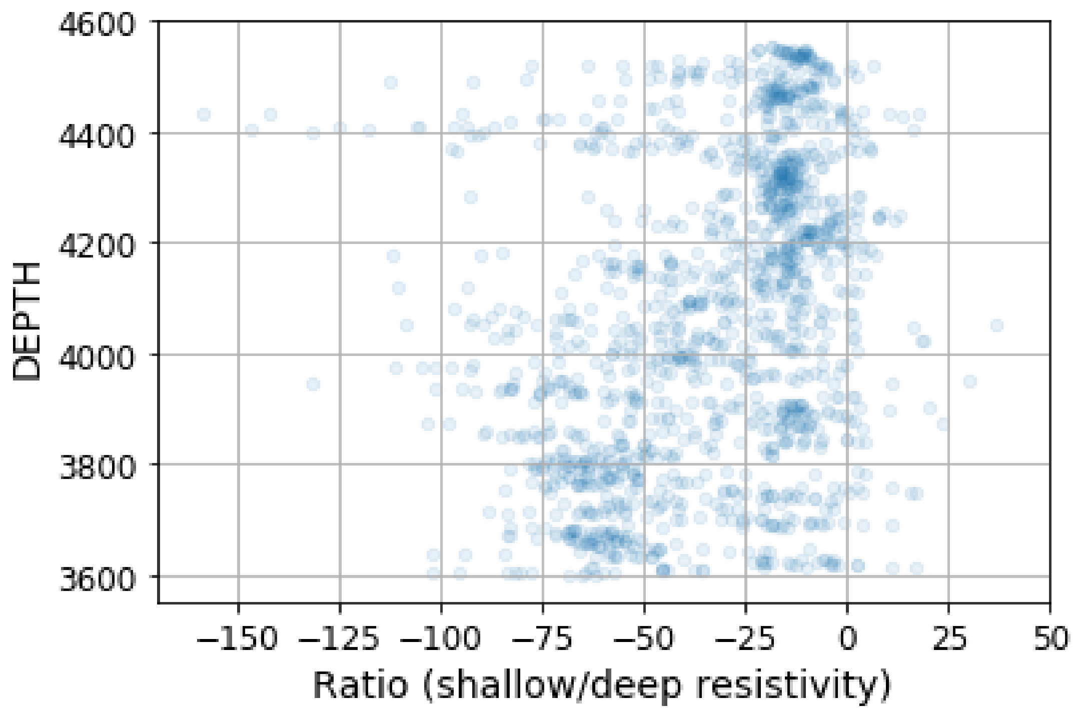
```
In [49]: LogDat_train_new.plot(kind = "scatter", x ="Density Correction", y ="DEPTH",grid = Tru
                             xlim=[-0.1,0.3], alpha=0.1)
```

```
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x11ee1f10>
```




```
In [50]: LogDat_train_new.plot(kind = "scatter", x ="Ratio (shallow/deep resistivity)", y ="DEI
                             xlim=[-170,50], alpha=0.1)
```

```
Out[50]: <matplotlib.axes._subplots.AxesSubplot at 0x12101e30>
```



6 ** Feature Scaling **

Since, Log data has a lot of noise in the data, if we choose the MinMax Scaler, which is sensitive to noise, we would have a bad training example. So, instead we are going to try and use StandardScaler instead which is much less sensitive to outliers.

```
In [51]: from sklearn.preprocessing import StandardScaler

         std_scaler = StandardScaler()
         LogDat_train_copy = LogDat_train_num.copy()
         LogDat_train_scaled = std_scaler.fit_transform(LogDat_train_copy)
         LogDat_train_scaled
```

```
Out[51]: array([[ -1.05005254e+00,   1.66138577e-03,   9.97496737e-02, ...,
                   2.48495548e-01,  -1.97490010e-02,  -1.97525730e-02],
                [  3.25716993e-01,  -1.33346182e+00,  -2.65951524e-01, ...,
                   1.55079941e+00,  -1.46864611e+00,  -1.46865324e+00],
                [  6.14963541e-02,   1.68133388e+00,  -1.14845866e+00, ...,
                   9.24625785e-02,   2.20562481e-01,   2.20560377e-01],
                ...,
                [  1.68508663e+00,   5.76360644e-01,  -6.94867182e-01, ...,
                  -4.15132656e-01,   1.29627074e-01,   1.29627855e-01],
                [  1.59033164e+00,   3.27573452e-01,  -1.32978435e+00, ...,
```

```
                 -2.60448770e-01,  -5.68468048e-01,  -5.68466011e-01],
               [ 6.57359450e-01,  -7.82045015e-01,  -2.07450423e-01, ...,
                 4.80340404e-01,  -5.75275738e-01,  -5.75285611e-01]])
```

Final training and test sets after standard scaling:

```
In [52]: X_train = LogDat_train_scaled

In [53]: X_train.shape

Out[53]: (1390, 17)

In [54]: X_train

Out[54]: array([[ -1.05005254e+00,   1.66138577e-03,   9.97496737e-02, ...,
                   2.48495548e-01,  -1.97490010e-02,  -1.97525730e-02],
                [ 3.25716993e-01,  -1.33346182e+00,  -2.65951524e-01, ...,
                   1.55079941e+00,  -1.46864611e+00,  -1.46865324e+00],
                [ 6.14963541e-02,   1.68133388e+00,  -1.14845866e+00, ...,
                   9.24625785e-02,   2.20562481e-01,   2.20560377e-01],
                ...,
                [ 1.68508663e+00,   5.76360644e-01,  -6.94867182e-01, ...,
                  -4.15132656e-01,   1.29627074e-01,   1.29627855e-01],
                [ 1.59033164e+00,   3.27573452e-01,  -1.32978435e+00, ...,
                  -2.60448770e-01,  -5.68468048e-01,  -5.68466011e-01],
                [ 6.57359450e-01,  -7.82045015e-01,  -2.07450423e-01, ...,
                   4.80340404e-01,  -5.75275738e-01,  -5.75285611e-01]])

In [55]: y_train = LogDat_train_labels.copy()
         y_train.head()

Out[55]:       Type of Formation_dolomite  Type of Formation_limestone  \
         364                            0                            1
         1119                           0                            1
         974                            0                            0
         481                            0                            0
         828                            0                            1

               Type of Formation_sandstone  Type of Formation_sandy limestone  \
         364                             0                                  0
         1119                            0                                  0
         974                             0                                  1
         481                             0                                  0
         828                             0                                  0

               Type of Formation_shale  Type of Formation_shaly limestone  \
         364                         0                                  0
         1119                        0                                  0
         974                         0                                  0
```

```
481                          1                                  0
828                          0                                  0

      Type of Formation_shaly sandstone
364                                   0
1119                                  0
974                                   0
481                                   0
828                                   0
```

In [56]:
```
y_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1390 entries, 364 to 1301
Data columns (total 7 columns):
Type of Formation_dolomite           1390 non-null uint8
Type of Formation_limestone          1390 non-null uint8
Type of Formation_sandstone          1390 non-null uint8
Type of Formation_sandy limestone    1390 non-null uint8
Type of Formation_shale              1390 non-null uint8
Type of Formation_shaly limestone    1390 non-null uint8
Type of Formation_shaly sandstone    1390 non-null uint8
dtypes: uint8(7)
memory usage: 20.4 KB
```

In [57]:
```
y_train.shape
```

Out[57]:
```
(1390, 7)
```

In [58]:
```
X_test_new = LogDat_test_num.copy()
X_test_new.head()
```

Out[58]:
```
       DEPTH  Neutron Porosity  Caliper  Density Porosity  Gamma ray  \
1501  4350.5           14.3377   7.9358            8.1210    53.0320
377   3788.5           21.5996   7.7935           20.2126    31.8369
1025  4112.5            9.2560   8.0716            6.8427    63.5457
819   4009.5            9.8961   8.1299            6.8641    45.8450
1364  4282.0           25.5932   9.6504           32.0067   179.7232

      Photoelectric  Bulk density  Density Correction  Resistivity (Deep)  \
1501         4.1425        2.5711             -0.0126              8.2467
377          3.9984        2.3644             -0.0032              1.3436
1025         3.9832        2.5930              0.1195              8.1046
819          4.8488        2.5926             -0.0653             18.2810
1364         2.5904        2.1627              0.1948              4.4241

      Resistivity (Medium)  Resistivity (Shallow)  \
1501                8.7727                13.0996
```

```
377                1.4921              5.5935
1025               7.6471             26.1792
819               18.2213             26.1286
1364               3.5417              4.1200

      Ratio (shallow/deep resistivity)  Spontaneous Potential  \
1501                          -18.0879                83.4069
377                           -55.7457                -7.8881
1025                          -45.8304                59.9619
819                           -13.9605                52.6570
1364                            2.7839                71.7486

      Micro-inverse resistivity (micro log)  \
1501                                14.2095
377                                  6.3832
1025                                19.7475
819                                 23.2709
1364                                 2.5252

      Micro-normal resistivity (micro log)  \
1501                               11.5789
377                                 9.2743
1025                               15.6692
819                                15.0720
1364                                1.8596

      Delta-t (interval transit time, or slowness)  Sonic porosity
1501                                       70.6689         16.3146
377                                        73.7622         18.5023
1025                                       68.4833         14.7690
819                                        61.6092          9.9075
1364                                       92.2882         31.6041
```

In [59]:
```
X_test_prepared = std_scaler.transform(X_test_new)
X_test_prepared
```

Out[59]:
```
array([[ 1.02180171, -0.07875738, -0.3125306 , ..., -0.22474743,
         0.1894101 ,  0.18940443],
       [-1.02636379,  0.94346041, -0.70706646, ..., -0.47751952,
         0.48600489,  0.48600959],
       [ 0.15442913, -0.79408038,  0.0639836 , ...,  0.22388301,
        -0.02015171, -0.02014575],
       ...,
       [ 1.63770913, -0.5776831 , -0.74920944, ...,  0.23111102,
        -1.01128436, -1.01127906],
       [ 1.2787335 ,  1.87064807, -0.8188008 , ..., -1.27187716,
         3.28191387,  3.28191009],
       [-1.0427637 , -0.06266799,  0.11777134, ..., -0.44686353,
         0.18954434,  0.18954001]])
```

```
In [60]: y_test_new = LogDat_test_labels.copy()
         y_test_new.head()
```

```
Out[60]:       Type of Formation_dolomite  Type of Formation_limestone  \
         1501                           0                            1
         377                            0                            1
         1025                           0                            0
         819                            0                            1
         1364                           0                            0

               Type of Formation_sandstone  Type of Formation_sandy limestone  \
         1501                            0                                  0
         377                             0                                  0
         1025                            0                                  0
         819                             0                                  0
         1364                            0                                  0

               Type of Formation_shale  Type of Formation_shaly limestone  \
         1501                        0                                  0
         377                         0                                  0
         1025                        1                                  0
         819                         0                                  0
         1364                        1                                  0

               Type of Formation_shaly sandstone
         1501                                  0
         377                                   0
         1025                                  0
         819                                   0
         1364                                  0
```

```
In [61]: y_train.sum()
```

```
Out[61]: Type of Formation_dolomite            96
         Type of Formation_limestone          698
         Type of Formation_sandstone           43
         Type of Formation_sandy limestone     28
         Type of Formation_shale              188
         Type of Formation_shaly limestone    333
         Type of Formation_shaly sandstone      4
         dtype: int64
```

```
In [62]: y_test_new.sum()
```

```
Out[62]: Type of Formation_dolomite            36
         Type of Formation_limestone          258
         Type of Formation_sandstone           16
         Type of Formation_sandy limestone     10
         Type of Formation_shale               70
```

```
Type of Formation_shaly limestone    123
Type of Formation_shaly sandstone      2
dtype: int64
```

```
In [63]: y_train_sl = y_train["Type of Formation_shaly limestone"]
         y_train_sl.head()
```

```
Out[63]: 364     0
         1119    0
         974     0
         481     0
         828     0
         Name: Type of Formation_shaly limestone, dtype: uint8
```

```
In [64]: y_train_sl.shape
```

```
Out[64]: (1390,)
```

```
In [65]: y_test_sl = y_test_new["Type of Formation_shaly limestone"]
         y_test_sl.head()
```

```
Out[65]: 1501    0
         377     0
         1025    0
         819     0
         1364    0
         Name: Type of Formation_shaly limestone, dtype: uint8
```

```
In [66]: y_test_sl.shape
```

```
Out[66]: (515,)
```

7 Feature relationships that might be helpful for dimensionality reduction

```
In [67]: corr_matrix = LogDat_train_new.corr()
         corr_matrix
```

```
Out[67]:
                                                   DEPTH  Neutron Porosity  \
         DEPTH                                  1.000000          0.026227
         Neutron Porosity                       0.026227          1.000000
         Caliper                               -0.160925          0.346474
         Density Porosity                       0.006949          0.792338
         Gamma ray                             -0.015251          0.492634
         Photoelectric                         -0.426672         -0.609484
         Bulk density                          -0.007264         -0.792846
         Density Correction                     0.288437          0.382943
         Resistivity (Deep)                     0.185614         -0.507277
```

```
Resistivity (Medium)                          0.092778        -0.454351
Resistivity (Shallow)                         0.110453        -0.485645
Ratio (shallow/deep resistivity)              0.309388         0.305985
Spontaneous Potential                         0.518315        -0.019830
Micro-inverse resistivity (micro log)         0.000968        -0.764768
Micro-normal resistivity (micro log)         -0.014199        -0.779691
Delta-t (interval transit time, or slowness)  0.086516         0.865844
Sonic porosity                                0.086517         0.865844

                                               Caliper  Density Porosity  \
DEPTH                                        -0.160925          0.006949
Neutron Porosity                              0.346474          0.792338
Caliper                                       1.000000          0.210884
Density Porosity                              0.210884          1.000000
Gamma ray                                     0.508586          0.350939
Photoelectric                                -0.154071         -0.719724
Bulk density                                 -0.210553         -0.999907
Density Correction                            0.407552          0.362429
Resistivity (Deep)                           -0.099120         -0.366280
Resistivity (Medium)                         -0.068459         -0.330421
Resistivity (Shallow)                        -0.120676         -0.362159
Ratio (shallow/deep resistivity)              0.351541          0.189555
Spontaneous Potential                         0.373152         -0.208932
Micro-inverse resistivity (micro log)        -0.151633         -0.661448
Micro-normal resistivity (micro log)         -0.258649         -0.632128
Delta-t (interval transit time, or slowness)  0.481840          0.726317
Sonic porosity                                0.481840          0.726317

                                              Gamma ray  Photoelectric  \
DEPTH                                         -0.015251      -0.426672
Neutron Porosity                               0.492634      -0.609484
Caliper                                        0.508586      -0.154071
Density Porosity                               0.350939      -0.719724
Gamma ray                                      1.000000      -0.289496
Photoelectric                                 -0.289496       1.000000
Bulk density                                  -0.350993       0.720067
Density Correction                             0.378688      -0.587235
Resistivity (Deep)                            -0.190353       0.331234
Resistivity (Medium)                          -0.181184       0.337320
Resistivity (Shallow)                         -0.180270       0.364613
Ratio (shallow/deep resistivity)               0.336097      -0.408454
Spontaneous Potential                          0.413603      -0.148010
Micro-inverse resistivity (micro log)         -0.236857       0.592701
Micro-normal resistivity (micro log)          -0.352367       0.617986
Delta-t (interval transit time, or slowness)   0.692064      -0.662070
Sonic porosity                                 0.692064      -0.662070

                                              Bulk density  \
```

```
DEPTH                                          -0.007264
Neutron Porosity                               -0.792846
Caliper                                        -0.210553
Density Porosity                               -0.999907
Gamma ray                                      -0.350993
Photoelectric                                   0.720067
Bulk density                                    1.000000
Density Correction                             -0.362460
Resistivity (Deep)                              0.367284
Resistivity (Medium)                            0.333102
Resistivity (Shallow)                           0.365009
Ratio (shallow/deep resistivity)               -0.190888
Spontaneous Potential                           0.208345
Micro-inverse resistivity (micro log)           0.662540
Micro-normal resistivity (micro log)            0.633809
Delta-t (interval transit time, or slowness)   -0.726637
Sonic porosity                                 -0.726636

                                             Density Correction  \
DEPTH                                                  0.288437
Neutron Porosity                                       0.382943
Caliper                                                0.407552
Density Porosity                                       0.362429
Gamma ray                                              0.378688
Photoelectric                                         -0.587235
Bulk density                                          -0.362460
Density Correction                                     1.000000
Resistivity (Deep)                                    -0.254869
Resistivity (Medium)                                  -0.245138
Resistivity (Shallow)                                 -0.203270
Ratio (shallow/deep resistivity)                       0.262263
Spontaneous Potential                                  0.314356
Micro-inverse resistivity (micro log)                 -0.373911
Micro-normal resistivity (micro log)                  -0.414038
Delta-t (interval transit time, or slowness)           0.537983
Sonic porosity                                         0.537983

                                             Resistivity (Deep)  \
DEPTH                                                  0.185614
Neutron Porosity                                      -0.507277
Caliper                                               -0.099120
Density Porosity                                      -0.366280
Gamma ray                                             -0.190353
Photoelectric                                          0.331234
Bulk density                                           0.367284
Density Correction                                    -0.254869
Resistivity (Deep)                                     1.000000
Resistivity (Medium)                                   0.864562
```

```
Resistivity (Shallow)                            0.579356
Ratio (shallow/deep resistivity)                 0.057239
Spontaneous Potential                            0.055853
Micro-inverse resistivity (micro log)            0.601999
Micro-normal resistivity (micro log)             0.607740
Delta-t (interval transit time, or slowness)    -0.486506
Sonic porosity                                  -0.486506

                                             Resistivity (Medium)  \
DEPTH                                                    0.092778
Neutron Porosity                                        -0.454351
Caliper                                                 -0.068459
Density Porosity                                        -0.330421
Gamma ray                                               -0.181184
Photoelectric                                            0.337320
Bulk density                                             0.333102
Density Correction                                      -0.245138
Resistivity (Deep)                                       0.864562
Resistivity (Medium)                                     1.000000
Resistivity (Shallow)                                    0.666370
Ratio (shallow/deep resistivity)                        -0.080670
Spontaneous Potential                                   -0.027903
Micro-inverse resistivity (micro log)                    0.569596
Micro-normal resistivity (micro log)                     0.584128
Delta-t (interval transit time, or slowness)            -0.442307
Sonic porosity                                          -0.442307

                                             Resistivity (Shallow)  \
DEPTH                                                     0.110453
Neutron Porosity                                         -0.485645
Caliper                                                  -0.120676
Density Porosity                                         -0.362159
Gamma ray                                                -0.180270
Photoelectric                                             0.364613
Bulk density                                              0.365009
Density Correction                                       -0.203270
Resistivity (Deep)                                        0.579356
Resistivity (Medium)                                      0.666370
Resistivity (Shallow)                                     1.000000
Ratio (shallow/deep resistivity)                         -0.471655
Spontaneous Potential                                     0.013215
Micro-inverse resistivity (micro log)                     0.616441
Micro-normal resistivity (micro log)                      0.695370
Delta-t (interval transit time, or slowness)             -0.449752
Sonic porosity                                           -0.449752

                                             Ratio (shallow/deep resistivity)  \
DEPTH                                                                0.309388
```

```
Neutron Porosity                                          0.305985
Caliper                                                   0.351541
Density Porosity                                          0.189555
Gamma ray                                                 0.336097
Photoelectric                                            -0.408454
Bulk density                                             -0.190888
Density Correction                                        0.262263
Resistivity (Deep)                                        0.057239
Resistivity (Medium)                                     -0.080670
Resistivity (Shallow)                                    -0.471655
Ratio (shallow/deep resistivity)                          1.000000
Spontaneous Potential                                     0.476085
Micro-inverse resistivity (micro log)                    -0.315014
Micro-normal resistivity (micro log)                     -0.477362
Delta-t (interval transit time, or slowness)              0.385793
Sonic porosity                                            0.385794

                                              Spontaneous Potential  \
DEPTH                                                      0.518315
Neutron Porosity                                          -0.019830
Caliper                                                    0.373152
Density Porosity                                          -0.208932
Gamma ray                                                  0.413603
Photoelectric                                             -0.148010
Bulk density                                               0.208345
Density Correction                                         0.314356
Resistivity (Deep)                                         0.055853
Resistivity (Medium)                                      -0.027903
Resistivity (Shallow)                                      0.013215
Ratio (shallow/deep resistivity)                           0.476085
Spontaneous Potential                                      1.000000
Micro-inverse resistivity (micro log)                      0.110212
Micro-normal resistivity (micro log)                      -0.048813
Delta-t (interval transit time, or slowness)               0.267278
Sonic porosity                                             0.267279

                                              Micro-inverse resistivity (micro log)  \
DEPTH                                                                      0.000968
Neutron Porosity                                                          -0.764768
Caliper                                                                   -0.151633
Density Porosity                                                          -0.661448
Gamma ray                                                                 -0.236857
Photoelectric                                                              0.592701
Bulk density                                                               0.662540
Density Correction                                                        -0.373911
Resistivity (Deep)                                                         0.601999
Resistivity (Medium)                                                       0.569596
Resistivity (Shallow)                                                      0.616441
```

```
Ratio (shallow/deep resistivity)                              -0.315014
Spontaneous Potential                                          0.110212
Micro-inverse resistivity (micro log)                          1.000000
Micro-normal resistivity (micro log)                           0.922005
Delta-t (interval transit time, or slowness)                  -0.664303
Sonic porosity                                                -0.664303

                                              Micro-normal resistivity (micro log)  \
DEPTH                                                                    -0.014199
Neutron Porosity                                                         -0.779691
Caliper                                                                  -0.258649
Density Porosity                                                         -0.632128
Gamma ray                                                                -0.352367
Photoelectric                                                             0.617986
Bulk density                                                              0.633809
Density Correction                                                       -0.414038
Resistivity (Deep)                                                        0.607740
Resistivity (Medium)                                                      0.584128
Resistivity (Shallow)                                                     0.695370
Ratio (shallow/deep resistivity)                                         -0.477362
Spontaneous Potential                                                    -0.048813
Micro-inverse resistivity (micro log)                                     0.922005
Micro-normal resistivity (micro log)                                      1.000000
Delta-t (interval transit time, or slowness)                             -0.743589
Sonic porosity                                                           -0.743589

                                              Delta-t (interval transit time, or slow
DEPTH                                                                             0.08
Neutron Porosity                                                                  0.86
Caliper                                                                           0.48
Density Porosity                                                                  0.72
Gamma ray                                                                         0.69
Photoelectric                                                                    -0.66
Bulk density                                                                     -0.72
Density Correction                                                                0.53
Resistivity (Deep)                                                               -0.48
Resistivity (Medium)                                                             -0.44
Resistivity (Shallow)                                                            -0.44
Ratio (shallow/deep resistivity)                                                  0.38
Spontaneous Potential                                                             0.26
Micro-inverse resistivity (micro log)                                            -0.66
Micro-normal resistivity (micro log)                                             -0.74
Delta-t (interval transit time, or slowness)                                      1.00
Sonic porosity                                                                    1.00

                                              Sonic porosity
DEPTH                                               0.086517
Neutron Porosity                                    0.865844
```

```
Caliper                                            0.481840
Density Porosity                                   0.726317
Gamma ray                                          0.692064
Photoelectric                                     -0.662070
Bulk density                                      -0.726636
Density Correction                                 0.537983
Resistivity (Deep)                                -0.486506
Resistivity (Medium)                              -0.442307
Resistivity (Shallow)                             -0.449752
Ratio (shallow/deep resistivity)                   0.385794
Spontaneous Potential                              0.267279
Micro-inverse resistivity (micro log)             -0.664303
Micro-normal resistivity (micro log)              -0.743589
Delta-t (interval transit time, or slowness)       1.000000
Sonic porosity                                     1.000000
```

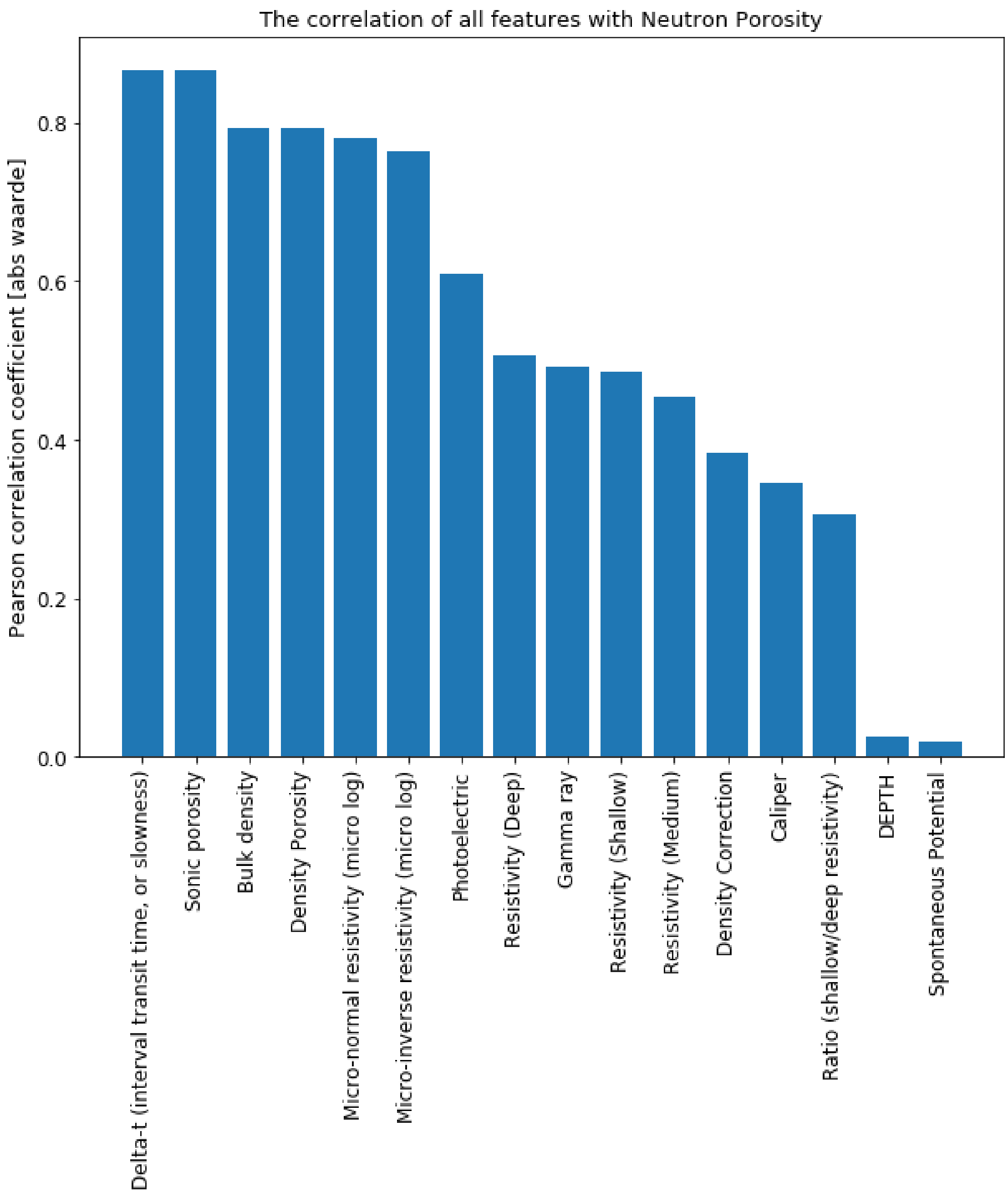
The above table individually shows relationship of each feature with each other. The more positive towards a feature a certain another feature is, the more likely it will increase with the increase in the corresponding quantity (e.g. Neutron log has a strong correlation with gamma ray, bulk density, caliper and photoelectrics logs etc. and are likely to go up with increase in any of these quantities).

0 means that a quantity has no relation with that feature.

The more negative a value would be in relation with that feature, the value has an inverse relation with that feature (e.g. -0.76 of relationship of Resistivity with SP log).

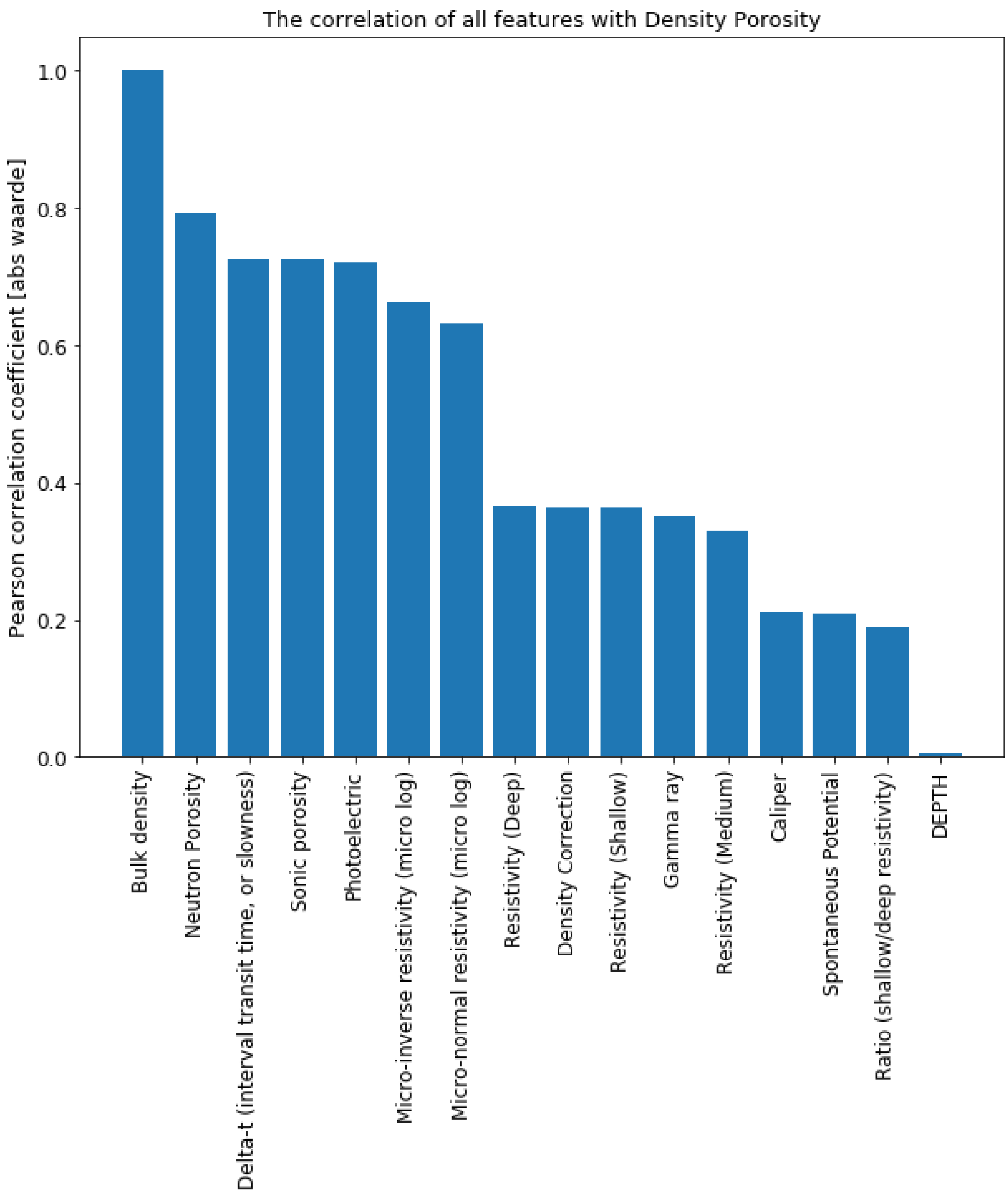
The correlation coefficient measures linear correlations though and does not account for non-linear correlations.

```
In [68]: def display_corr_with_col(df, col):
             correlation_matrix = LogDat_train_new.corr()
             correlation_type = correlation_matrix[col].copy()
             abs_correlation_type = correlation_type.apply(lambda x: abs(x))
             desc_corr_values = abs_correlation_type.sort_values(ascending=False)
             y_values = list(desc_corr_values.values)[1:]
             x_values = range(0,len(y_values))
             xlabels = list(desc_corr_values.keys())[1:]
             fig, ax = plt.subplots(figsize=(10,8))
             ax.bar(x_values, y_values)
             ax.set_title('The correlation of all features with {}'.format(col), fontsize=13)
             ax.set_ylabel('Pearson correlation coefficient [abs waarde]', fontsize=13)
             plt.xticks(x_values, xlabels, rotation='vertical')
             plt.show()

         display_corr_with_col(LogDat_train_new, 'Neutron Porosity')
```

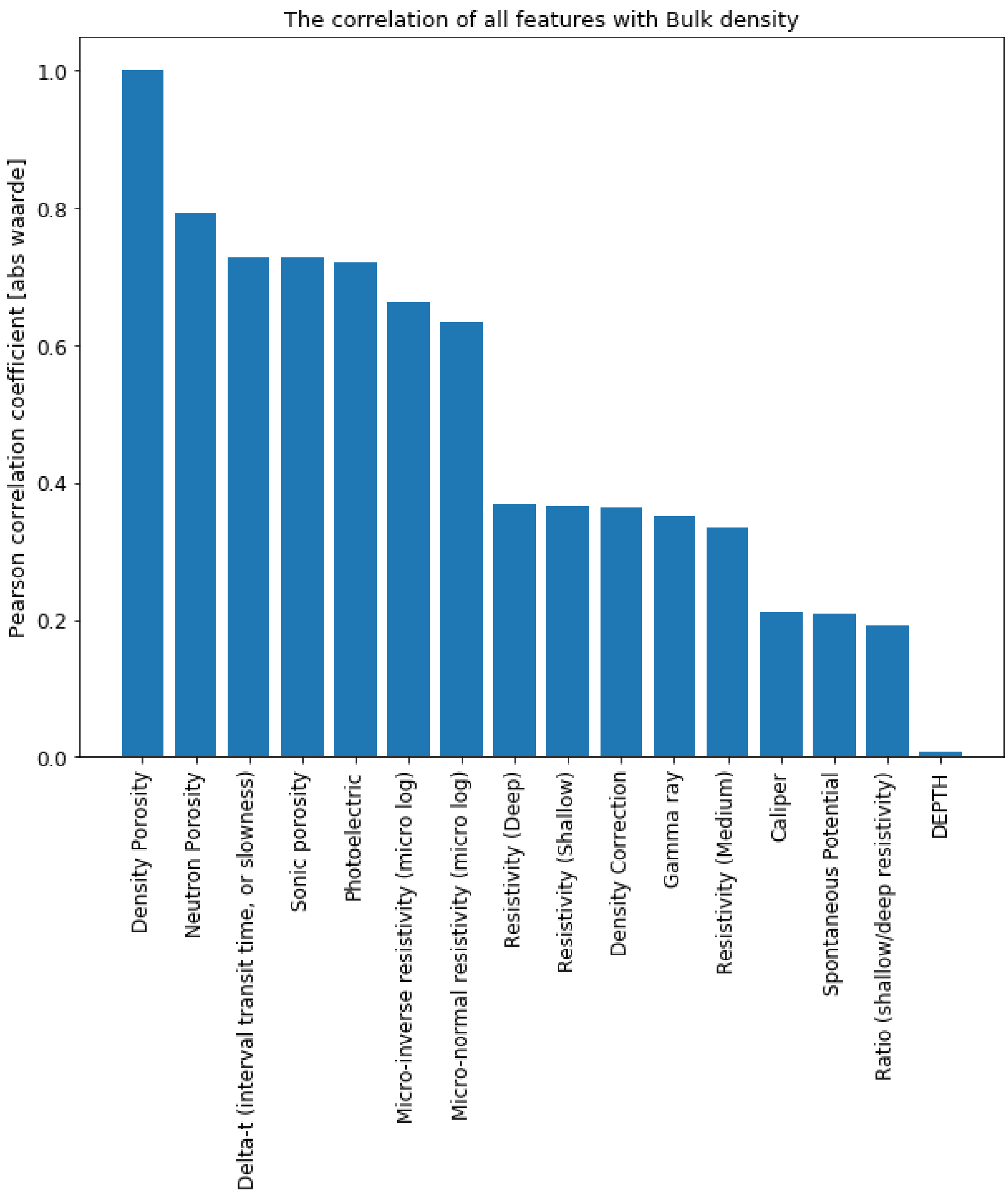


Strong relationship of Neutron Porosity with Sonic Porosity/ Delta-t which is a good question if it may not be necessary

```
In [69]: display_corr_with_col(LogDat_train_new, 'Density Porosity')
```

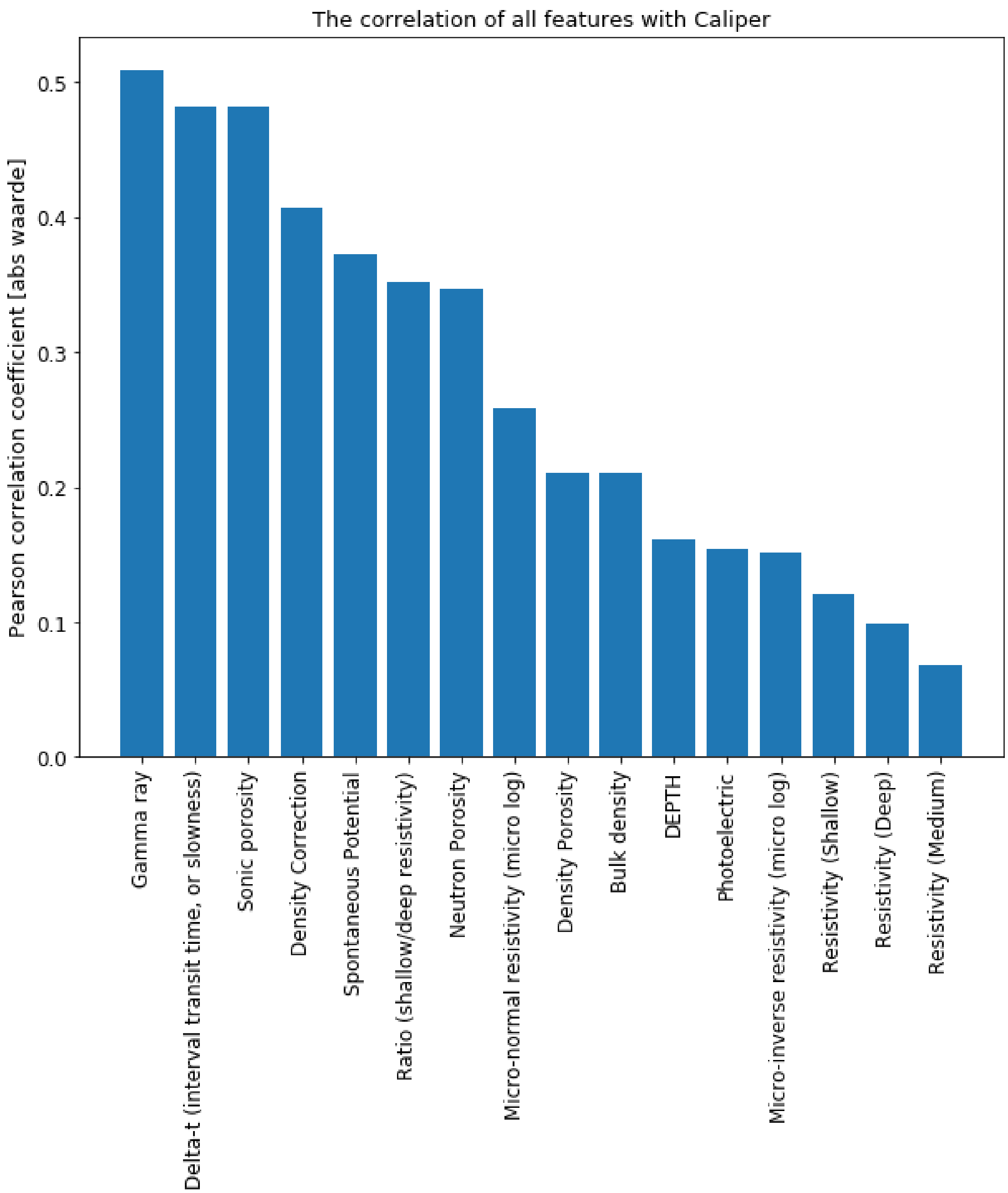



```
In [70]: display_corr_with_col(LogDat_train_new, 'Bulk density')
```



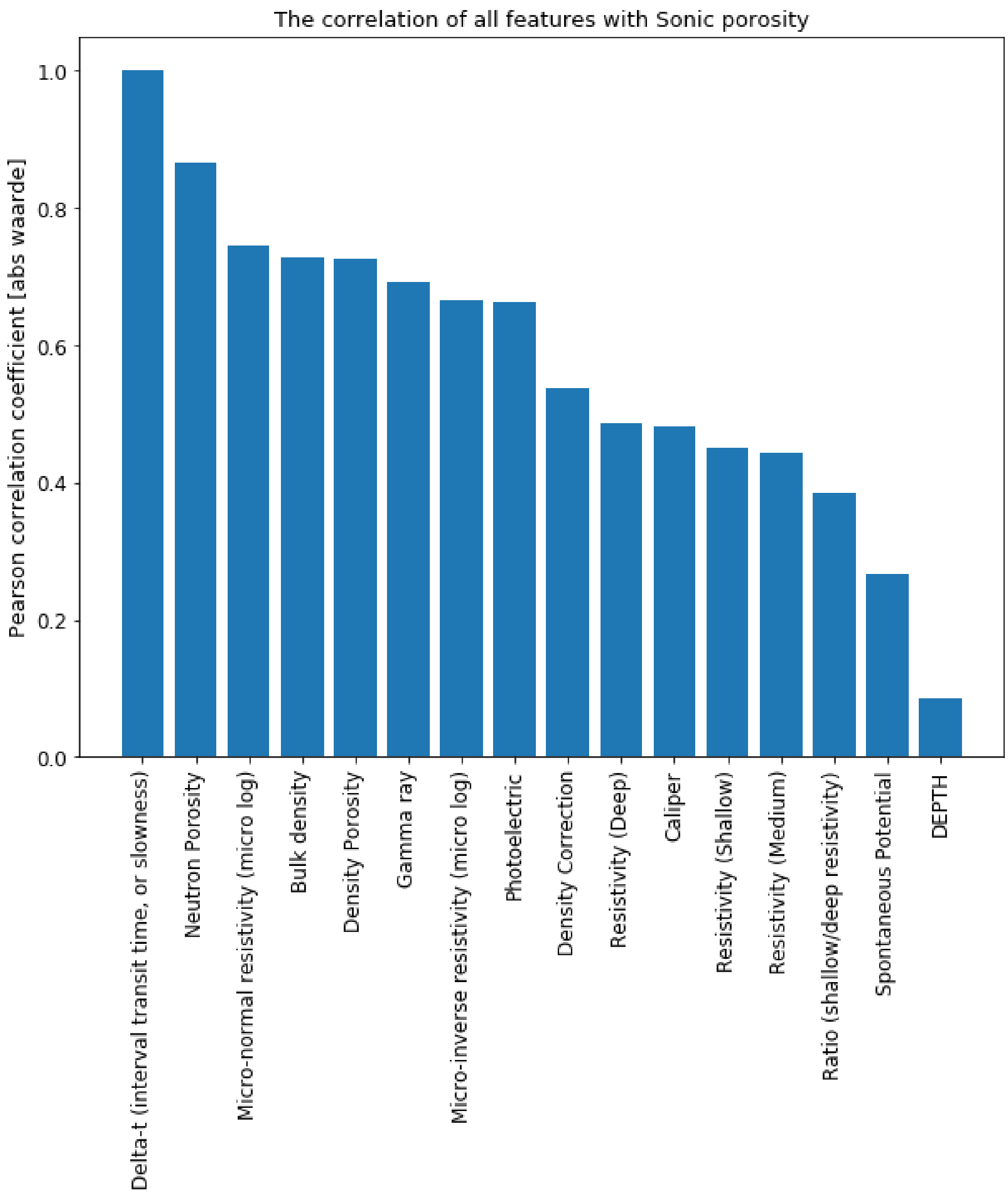
Since, density porosity is very highly related to bulk density we can eliminate either one of them because density porosity is calculated from bulk density after all.

```
In [71]: display_corr_with_col(LogDat_train_new, 'Caliper')
```



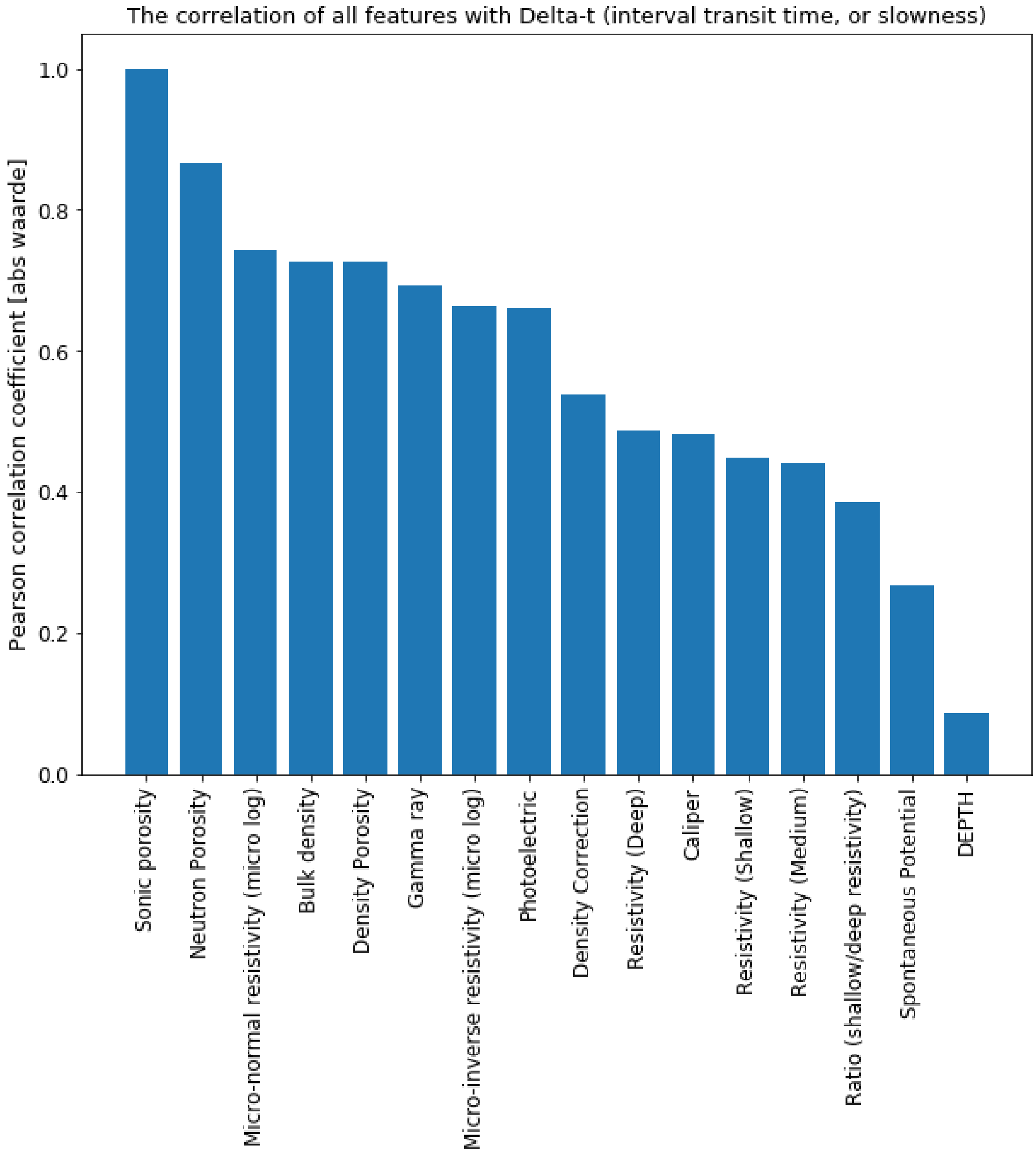
Caliper has weak relationships with everyone and maybe or may not be removed.

```
In [72]: display_corr_with_col(LogDat_train_new, 'Sonic porosity')
```

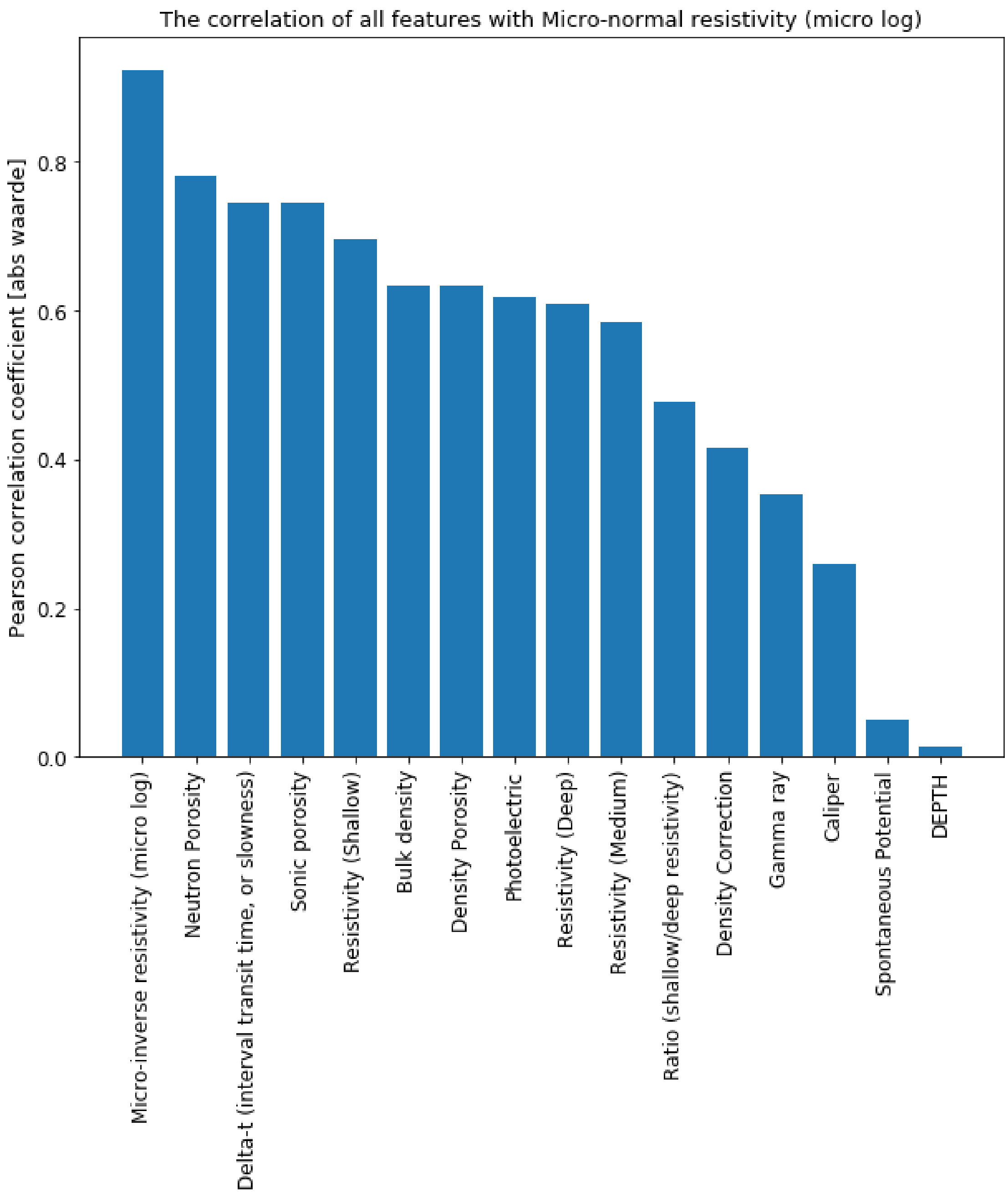


Sonic porosity has almost perfectly linear relationship with delta-t and can be removed as sonic porosity is after all calculated from delta-t.

```
In [73]: display_corr_with_col(LogDat_train_new, 'Delta-t (interval transit time, or slowness)
```

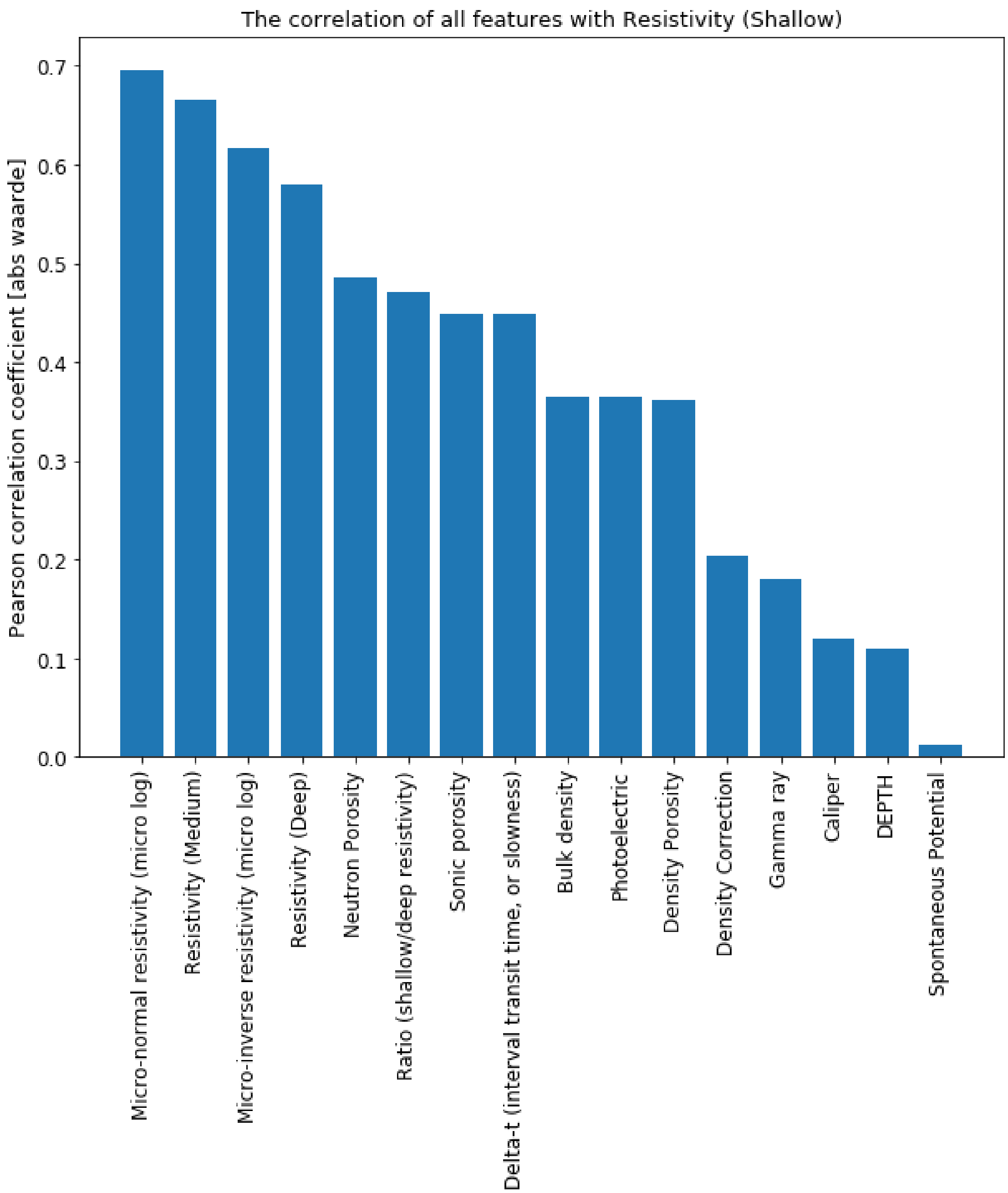



```
In [74]: display_corr_with_col(LogDat_train_new, 'Micro-normal resistivity (micro log)')
```

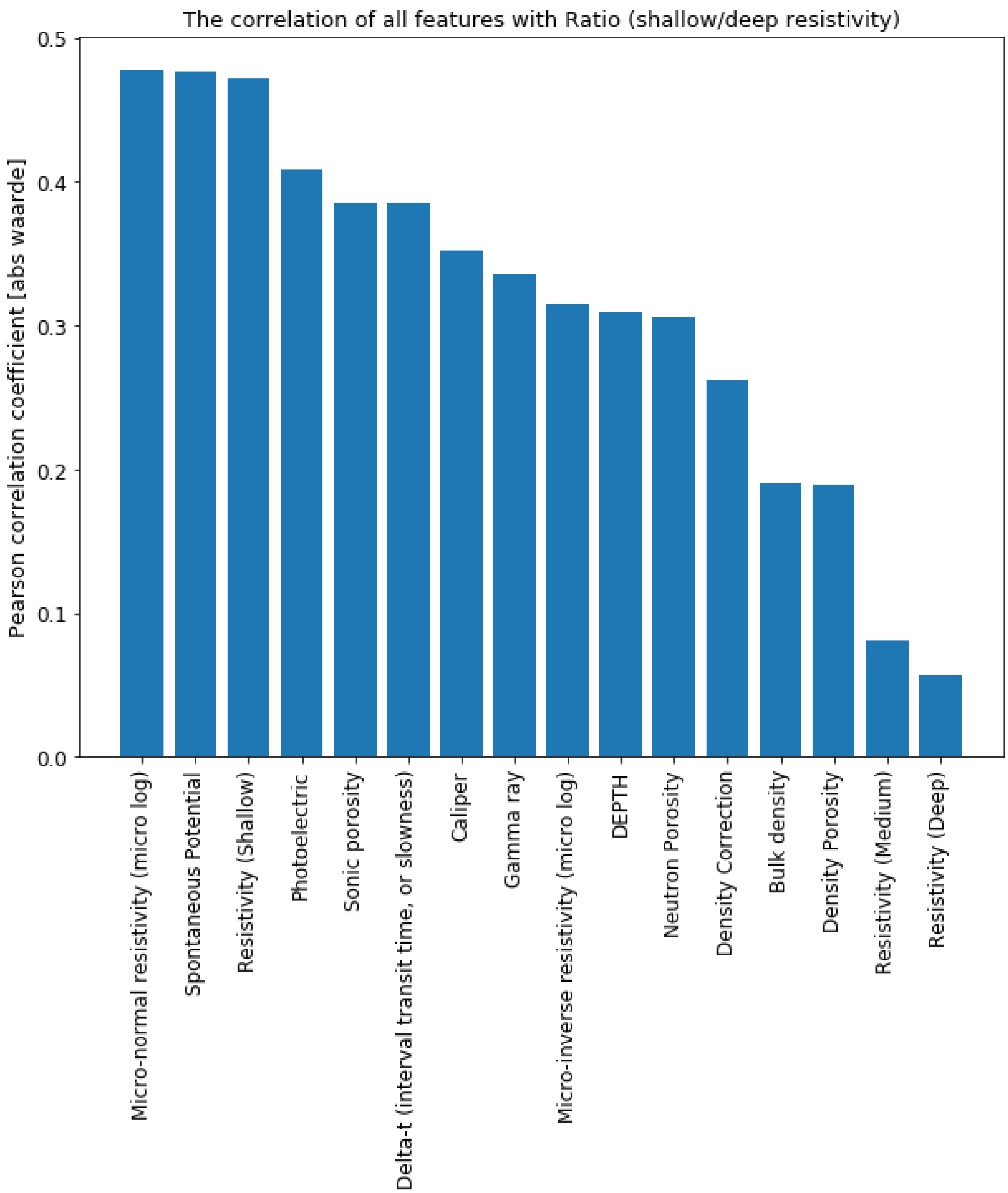


Micro-normal has perfectly linear relationship with micro-inverse so can also be removed.

```
In [75]: display_corr_with_col(LogDat_train_new, 'Resistivity (Shallow)')
```

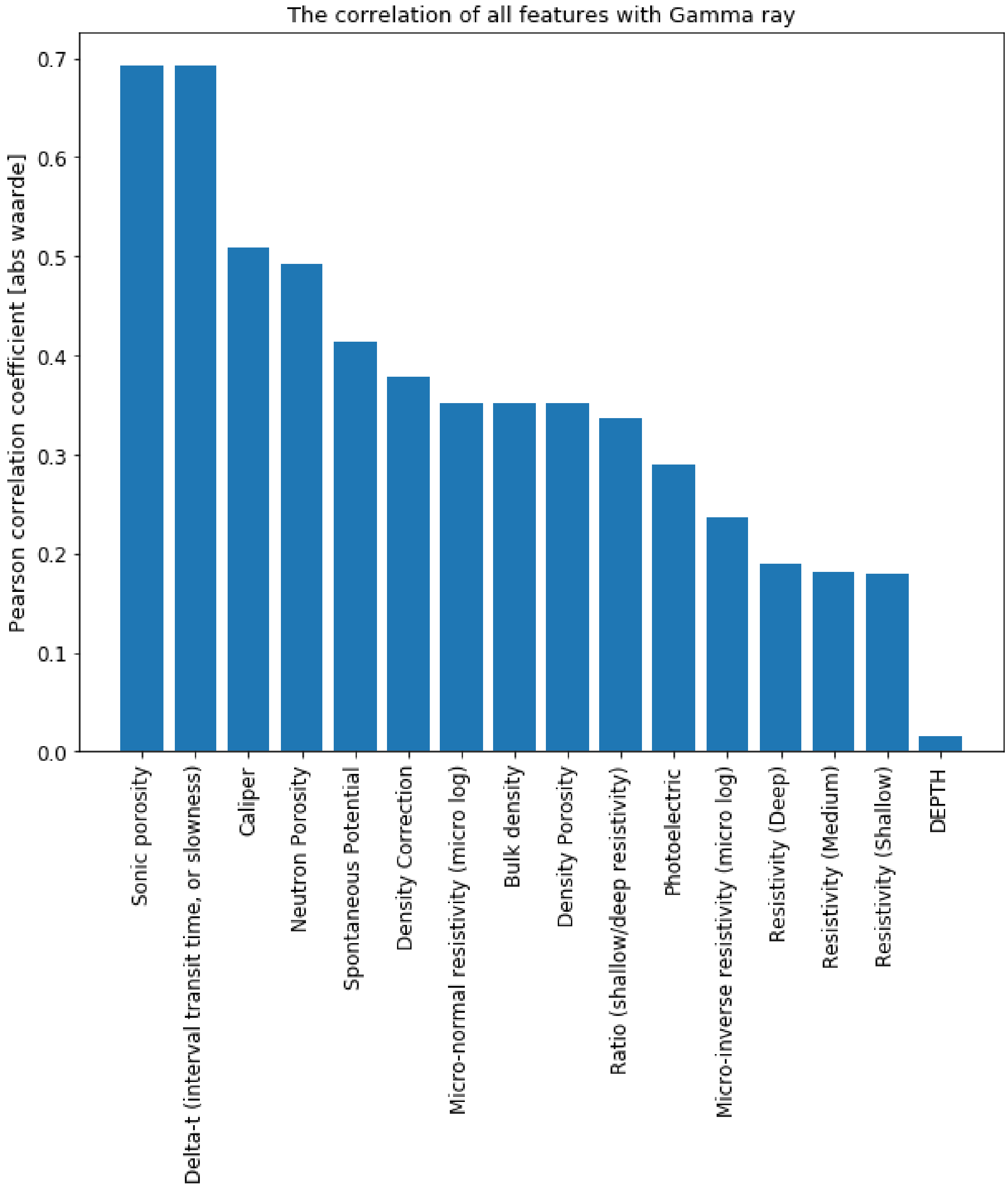



```
In [76]: display_corr_with_col(LogDat_train_new, 'Ratio (shallow/deep resistivity)')
```

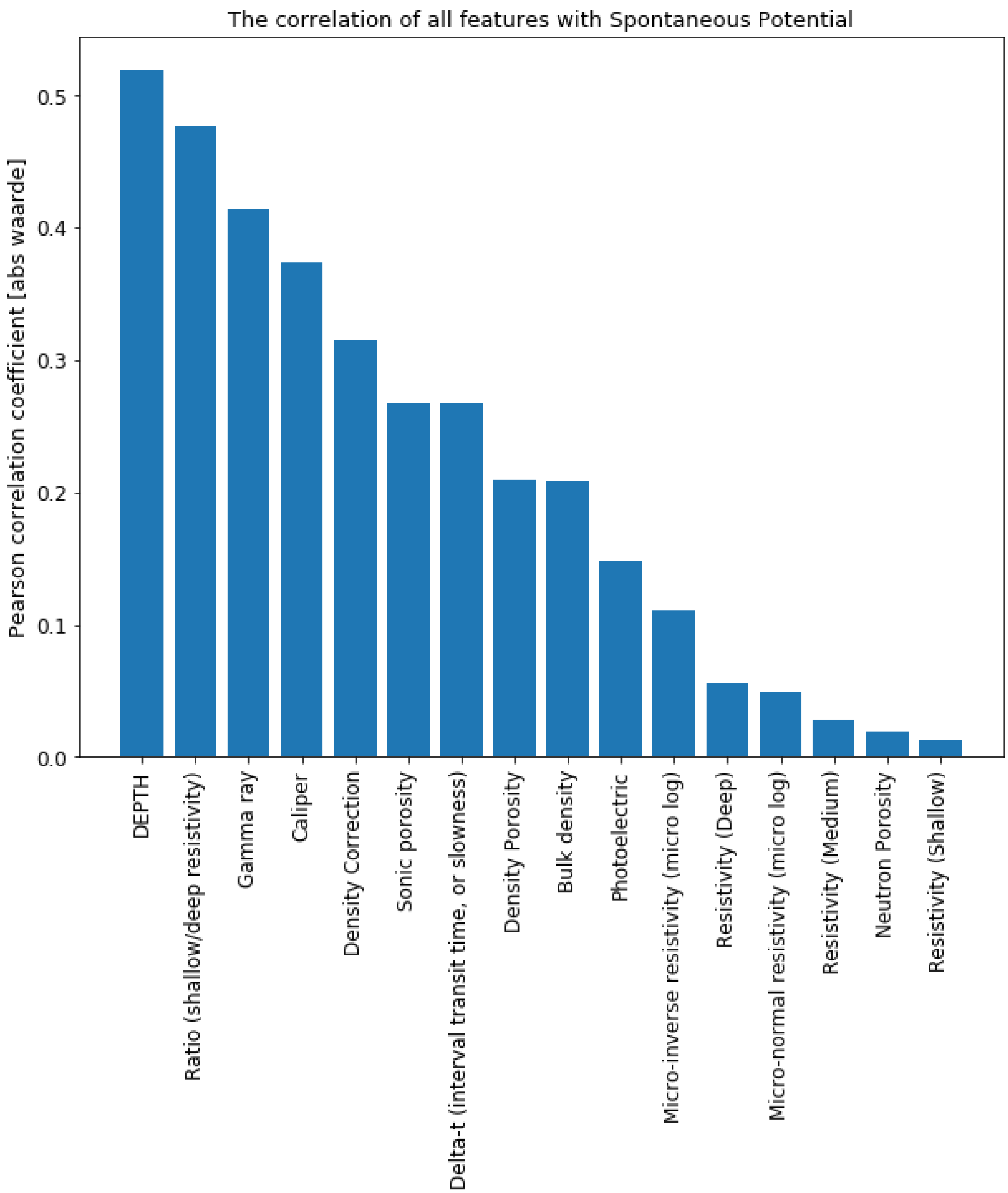


Weak relation with everyone like Caliper had and can be dropped.

```
In [77]: display_corr_with_col(LogDat_train_new, 'Gamma ray')
```

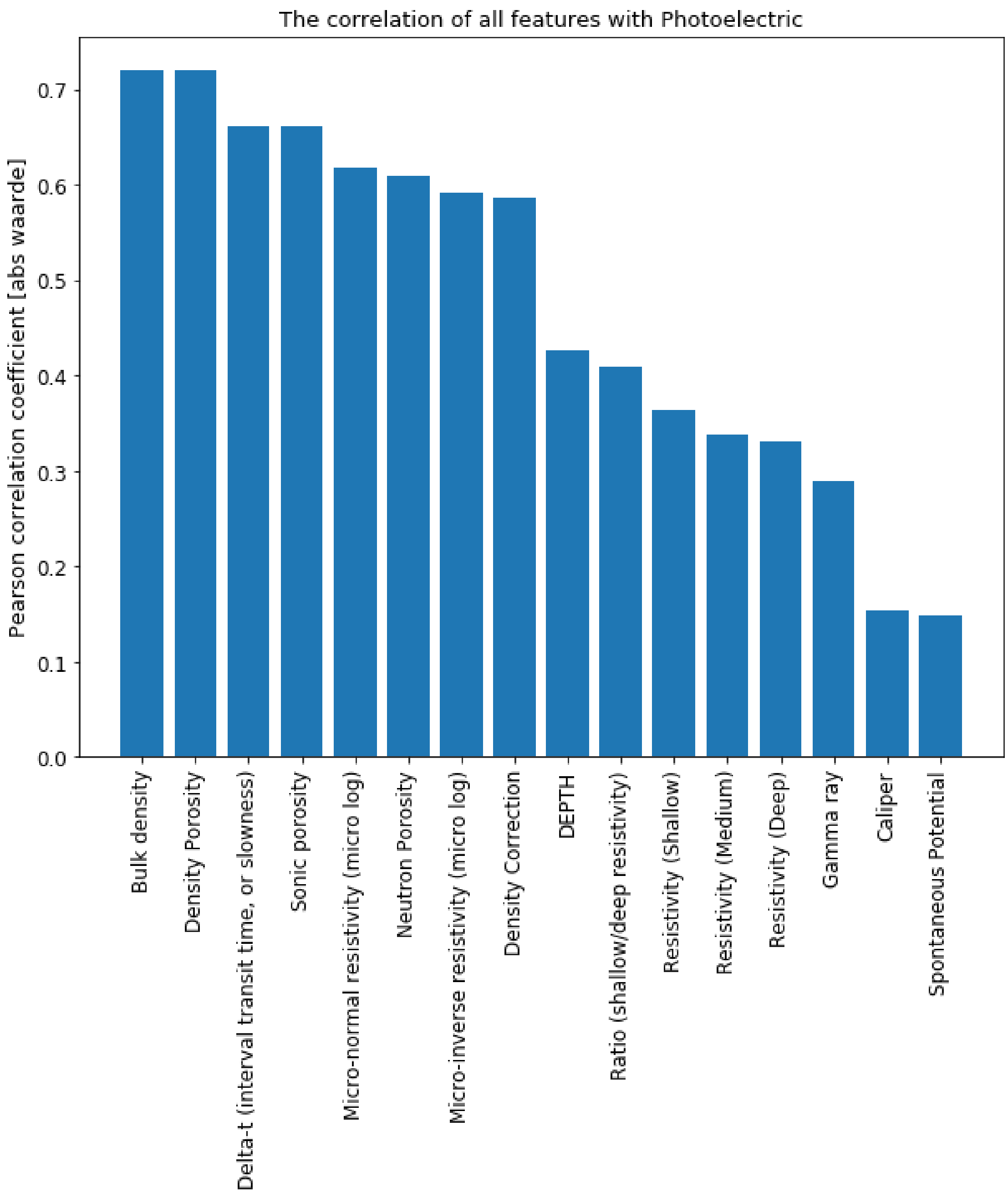



```
In [78]: display_corr_with_col(LogDat_train_new, 'Spontaneous Potential')
```

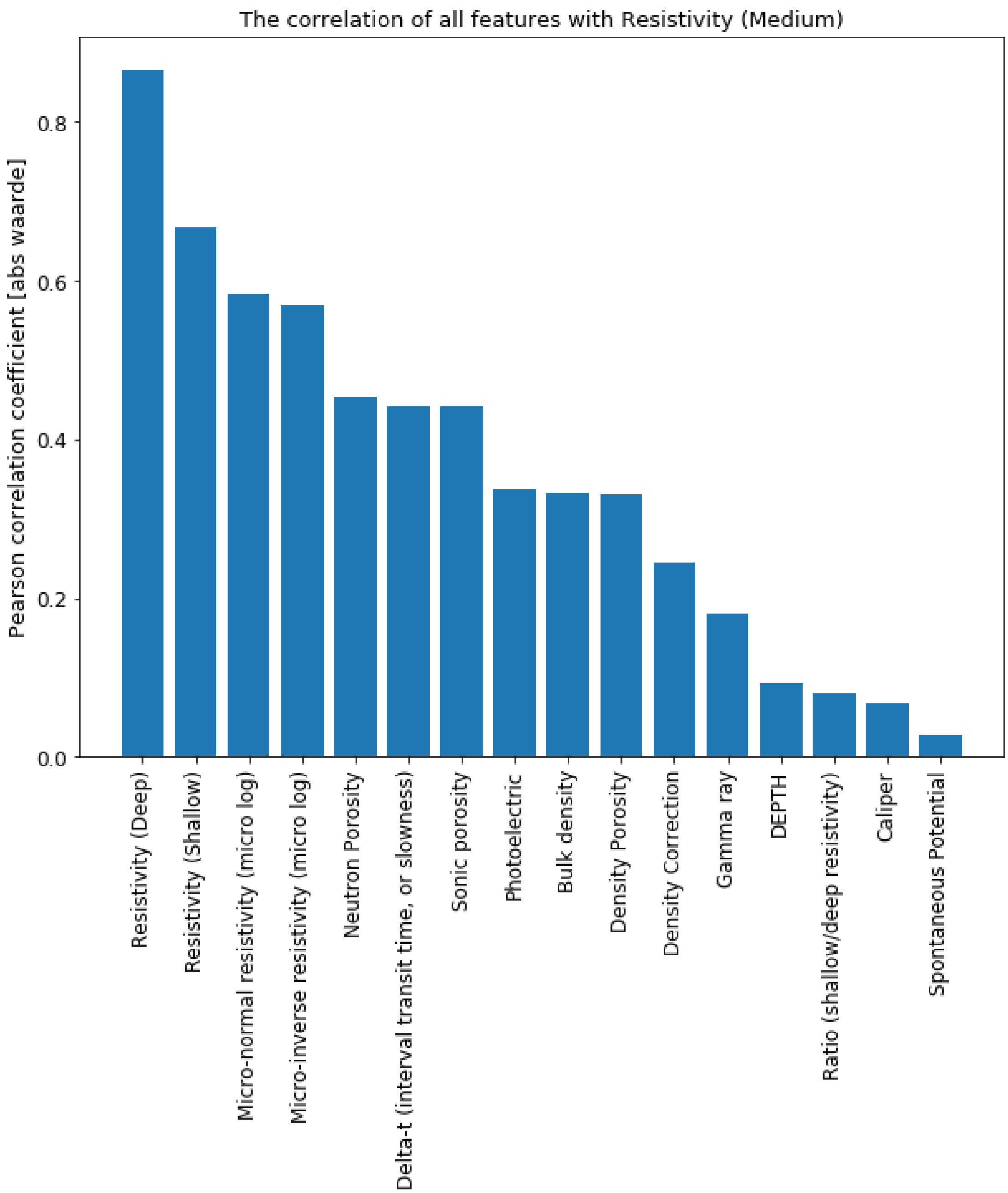


Weak relationship with everyone and can be dropped.

```
In [79]: display_corr_with_col(LogDat_train_new, 'Photoelectric')
```

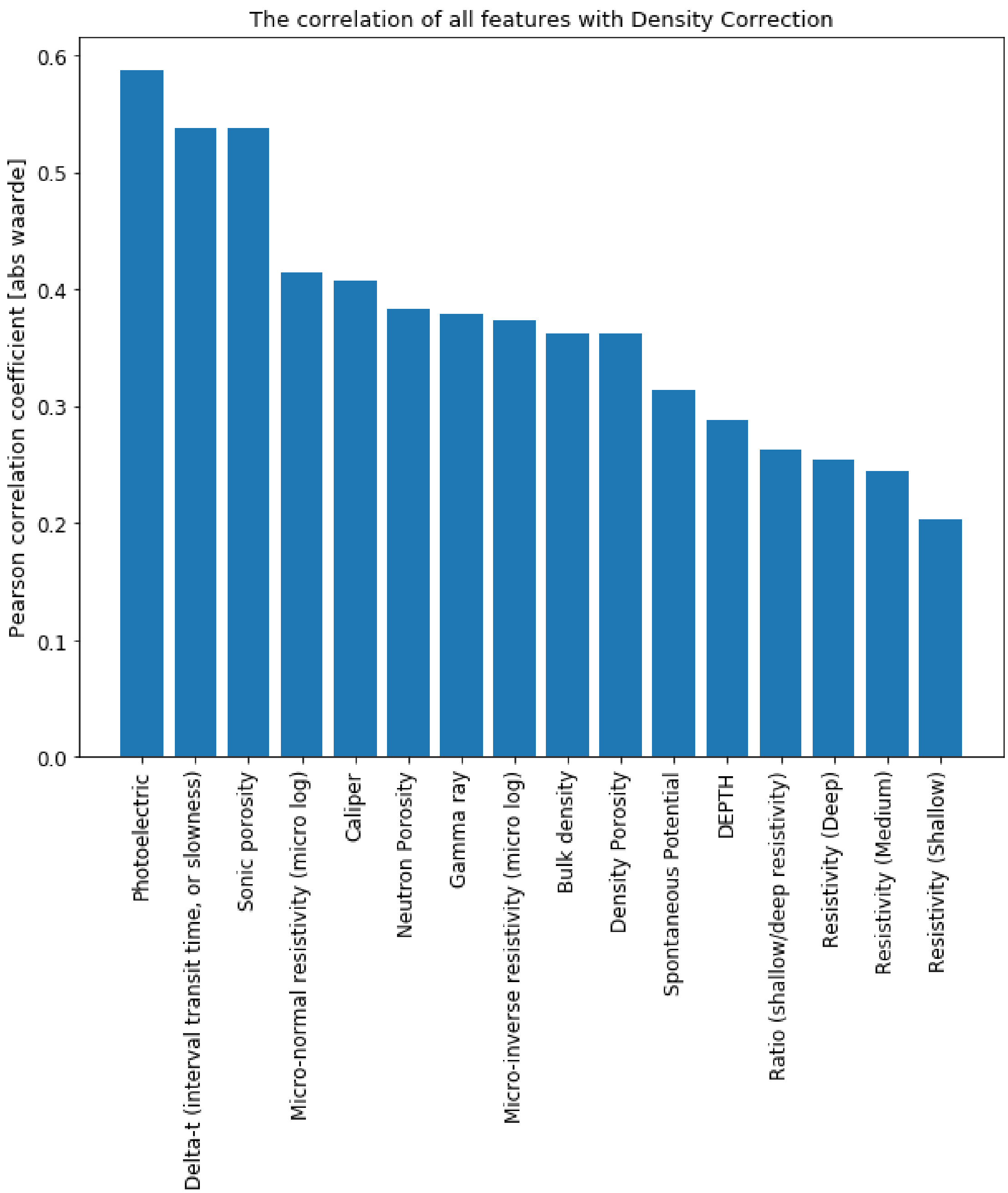



```
In [80]: display_corr_with_col(LogDat_train_new, 'Resistivity (Medium)')
```



Resistivity medium and deep have strong relationship and possibily could be dropped.

```
In [81]: display_corr_with_col(LogDat_train_new, 'Density Correction')
```



Somewhat weak relationships.

8 Training the data

** LogisticRegression on training set**

```
In [82]: from sklearn.linear_model import LogisticRegressionCV

         log_reg_n = LogisticRegressionCV(random_state=42)
```

In [83]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(log_reg_n, X_train, y_train_sl, cv=5, scoring ="f1_weighted")
```

Out[83]:
```
array([ 0.72389625,  0.75531066,  0.76169739,  0.73720477,  0.77486259])
```

In [84]:
```
y_predLog = log_reg_n.fit(X_train, y_train_sl).predict(X_train)
("Number of mislabeled points out of a total %d points : %d" % (X_train.shape[0],(y_t
```

Out[84]:
```
'Number of mislabeled points out of a total 1390 points : 299'
```

In [85]:
```
from sklearn.model_selection import cross_val_predict
y_log_crpred =cross_val_predict(log_reg_n, X_train, y_train_sl, cv=5, method ="predic
```

In [86]:
```
from sklearn.metrics import f1_score

f1_score(y_train_sl, y_log_crpred, average = "weighted")
```

Out[86]:
```
0.75125598355975243
```

** RandomForest on training set**

In [87]:
```
from sklearn.ensemble import RandomForestClassifier

forest_clf_n = RandomForestClassifier(random_state =42)
forest_clf_n.fit(X_train, y_train_sl)
```

Out[87]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=42, verbose=0, warm_start=False)
```

In [88]:
```
cross_val_score(forest_clf_n, X_train, y_train_sl, cv=5, scoring ="f1_weighted")
```

Out[88]:
```
array([ 0.95980709,  0.92280374,  0.92204216,  0.92467386,  0.9304593 ])
```

In [89]:
```
y_predfor_p_n = forest_clf_n.fit(X_train, y_train_sl).predict(X_train)
("Number of mislabeled points out of a total %d points : %d" % (X_train.shape[0],(y_t
```

Out[89]:
```
'Number of mislabeled points out of a total 1390 points : 4'
```

DecisionTree on training set

In [90]:
```
from sklearn.tree import DecisionTreeClassifier
tree_clf_n = DecisionTreeClassifier(random_state =42)
tree_clf_n.fit(X_train, y_train_sl)
```

Out[90]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=42,
            splitter='best')
```

```
In [91]: cross_val_score(tree_clf_n, X_train, y_train_sl, cv=5, scoring ="f1_weighted")

Out[91]: array([ 0.94321037,  0.91592645,  0.93032874,  0.91581915,  0.89419589])

In [92]: y_predtree = tree_clf_n.fit(X_train, y_train_sl).predict(X_train)
         ("Number of mislabeled points out of a total %d points : %d" % (X_train.shape[0],(y_t

Out[92]: 'Number of mislabeled points out of a total 1390 points : 0'
```

GaussianRBF SVM Classsifier on training set:

```
In [93]: from sklearn.svm import SVC

         rbf_kernel_svm_clf = SVC(C=1, probability = True)
         rbf_kernel_svm_clf.fit(X_train, y_train_sl)

Out[93]: SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
           decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
           max_iter=-1, probability=True, random_state=None, shrinking=True,
           tol=0.001, verbose=False)

In [94]: cross_val_score(rbf_kernel_svm_clf, X_train, y_train_sl, cv=5, scoring ="f1_weighted")

Out[94]: array([ 0.89339565,  0.86639757,  0.88002552,  0.85966672,  0.89526028])

In [95]: y_predsvm = rbf_kernel_svm_clf.fit(X_train, y_train_sl).predict(X_train)
         ("Number of mislabeled points out of a total %d points : %d" % (X_train.shape[0],(y_t

Out[95]: 'Number of mislabeled points out of a total 1390 points : 127'

In [96]: y_svm_cr = cross_val_predict(rbf_kernel_svm_clf, X_train, y_train_sl, cv =5, method =

In [97]: f1_score(y_train_sl, y_svm_cr, average = 'weighted')

Out[97]: 0.87894265229034829
```

KNeighborsClassifier on training set:

```
In [98]: from sklearn.neighbors import KNeighborsClassifier
         knn_clf_n = KNeighborsClassifier()
         knn_clf_n.fit(X_train, y_train_sl)

Out[98]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                    weights='uniform')

In [99]: cross_val_score(knn_clf_n, X_train, y_train_sl, cv=5, scoring ="f1_weighted")

Out[99]: array([ 0.91093171,  0.88642074,  0.89976849,  0.88987724,  0.85806555])

In [100]: y_predknn = knn_clf_n.fit(X_train, y_train_sl).predict(X_train)
          ("Number of mislabeled points out of a total %d points : %d" % (X_train.shape[0],(y_
```

Out[100]: 'Number of mislabeled points out of a total 1390 points : 85'

Gaussian Naive Bayes on training set:

In [101]:
```
from sklearn.naive_bayes import GaussianNB
gnb_clf_n = GaussianNB()
y_predNB = gnb_clf_n.fit(X_train, y_train_sl)
```

In [102]:
```
cross_val_score(gnb_clf_n, X_train, y_train_sl, cv=5, scoring ="f1_weighted")
```

Out[102]: array([0.73777109, 0.74831147, 0.69594624, 0.76289201, 0.70333631])

In [103]:
```
y_predNB_tr = gnb_clf_n.fit(X_train, y_train_sl).predict(X_train)
("Number of mislabeled points out of a total %d points : %d" % (X_train.shape[0],(y_t
```

Out[103]: 'Number of mislabeled points out of a total 1390 points : 395'

In [104]:
```
y_predgnb_cv = cross_val_predict(gnb_clf_n, X_train, y_train_sl, cv=5, method = "pred
```

In [105]:
```
f1_score(y_train_sl, y_predgnb_cv, average = 'weighted')
```

Out[105]: 0.730148543036074779

Confusion Matrix and Precision & Recall
Random Forest:

In [106]:
```
from sklearn.model_selection import cross_val_predict

y_train_slpred = cross_val_predict(forest_clf_n, X_train, y_train_sl, cv=5)
```

In [107]:
```
from sklearn.metrics import confusion_matrix

confusion_matrix(y_train_sl, y_train_slpred)
```

Out[107]:
```
array([[1034,   23],
       [  69,  264]], dtype=int64)
```

In [108]:
```
from sklearn.metrics import precision_score, recall_score

precision_score(y_train_sl, y_train_slpred, average = 'weighted')
```

Out[108]: 0.93323107581138565

In [109]:
```
recall_score(y_train_sl, y_train_slpred,average = 'weighted')
```

Out[109]: 0.93381294964028771

In [110]:
```
from sklearn.metrics import f1_score

f1_score(y_train_sl, y_train_slpred, average = 'weighted')
```

```
Out[110]: 0.93206239309627548
```

Decision Tree Classifier:

```
In [111]: y_train_slpred2 = cross_val_predict(tree_clf_n, X_train, y_train_sl, cv=5)

In [112]: confusion_matrix(y_train_sl, y_train_slpred2)

Out[112]: array([[995,  62],
                 [ 50, 283]], dtype=int64)

In [113]: precision_score(y_train_sl, y_train_slpred2,average = 'weighted')

Out[113]: 0.92056284824439183

In [114]: recall_score(y_train_sl, y_train_slpred2,average = 'weighted')

Out[114]: 0.91942446043165471

In [115]: f1_score(y_train_sl, y_train_slpred2, average = 'weighted')

Out[115]: 0.91990752152115784
```

Logisitc Regression:

```
In [116]: y_train_slpred3 = cross_val_predict(log_reg_n, X_train, y_train_sl, cv=5)

In [117]: confusion_matrix(y_train_sl, y_train_slpred3,)

Out[117]: array([[985,  72],
                 [235,  98]], dtype=int64)

In [118]: precision_score(y_train_sl, y_train_slpred3,average = 'weighted')

Out[118]: 0.75205917040716497

In [119]: recall_score(y_train_sl, y_train_slpred3,average = 'weighted')

Out[119]: 0.77913669064748203

In [120]: f1_score(y_train_sl, y_train_slpred3, average = 'weighted')

Out[120]: 0.75125598355975243
```

K-Nearest Neighbor Classifier:

```
In [121]: y_train_slpred4 = cross_val_predict(knn_clf_n, X_train, y_train_sl, cv=5)

In [122]: confusion_matrix(y_train_sl, y_train_slpred4)

Out[122]: array([[995,  62],
                 [ 90, 243]], dtype=int64)
```

```
In [123]: precision_score(y_train_sl, y_train_slpred4, average = 'weighted')
```

```
Out[123]: 0.88822358083918906
```

```
In [124]: recall_score(y_train_sl, y_train_slpred4,average = 'weighted')
```

```
Out[124]: 0.89064748201438848
```

```
In [125]: f1_score(y_train_sl, y_train_slpred4, average = 'weighted')
```

```
Out[125]: 0.88896261931999676
```

GaussianRBF SVM Classifier:

```
In [126]: y_train_slpred5 = cross_val_predict(rbf_kernel_svm_clf, X_train, y_train_sl, cv=5)
```

```
In [127]: confusion_matrix(y_train_sl, y_train_slpred5)
```

```
Out[127]: array([[1003,   54],
                 [ 109,  224]], dtype=int64)
```

```
In [128]: precision_score(y_train_sl, y_train_slpred5, average = 'weighted')
```

```
Out[128]: 0.87892642720356096
```

```
In [129]: recall_score(y_train_sl, y_train_slpred5, average = 'weighted')
```

```
Out[129]: 0.88273381294964026
```

```
In [130]: f1_score(y_train_sl, y_train_slpred5, average = 'weighted')
```

```
Out[130]: 0.87894265229034829
```

Gaussian Naive Bayes:

```
In [131]: y_train_slpred6 = cross_val_predict(gnb_clf_n, X_train, y_train_sl, cv=5)
```

```
In [132]: confusion_matrix(y_train_sl, y_train_slpred6)
```

```
Out[132]: array([[774, 283],
                 [115, 218]], dtype=int64)
```

```
In [133]: precision_score(y_train_sl, y_train_slpred6, average = 'weighted')
```

```
Out[133]: 0.76630641735845129
```

```
In [134]: recall_score(y_train_sl, y_train_slpred6,average = 'weighted')
```

```
Out[134]: 0.71366906474820146
```

```
In [135]: f1_score(y_train_sl, y_train_slpred6, average = 'weighted')
```

Out[135]: 0.73014854303607479

ROC Curve

```
In [136]: y_logscores = cross_val_predict(log_reg_n, X_train, y_train_sl, cv=5,
                                          method="decision_function")
```

```
In [137]: y_logscores
```

```
Out[137]: array([[ 0.        , -1.94919083],
                 [ 0.        , -5.98259537],
                 [ 0.        , -1.88523917],
                 ...,
                 [ 0.        , -4.16359282],
                 [ 0.        , -4.05007741],
                 [ 0.        , -1.55772301]])
```

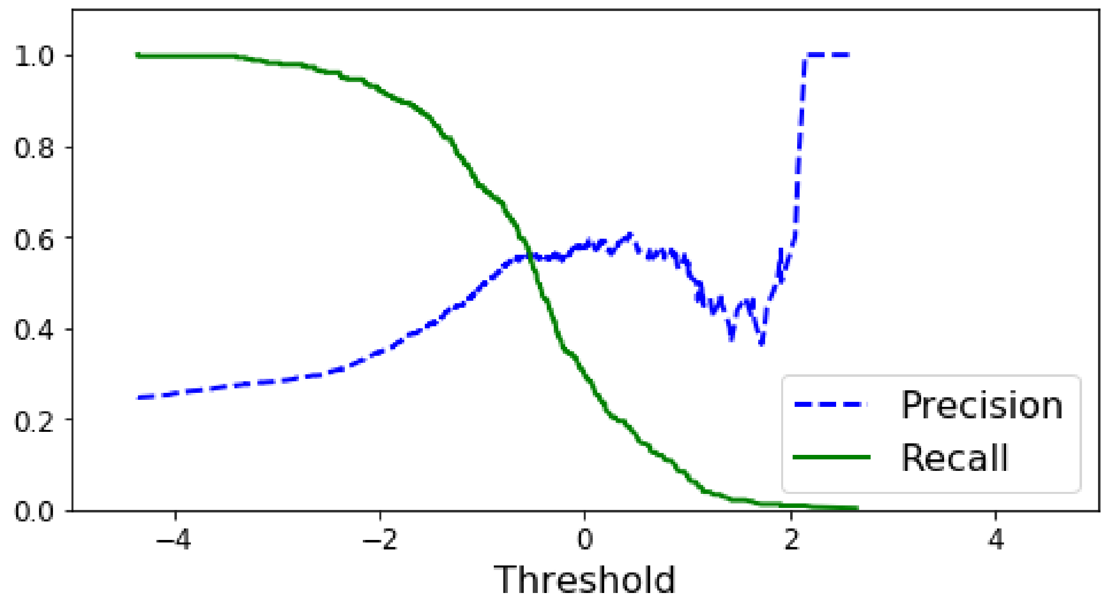
```
In [138]: y_logscores.shape
```

```
Out[138]: (1390, 2)
```

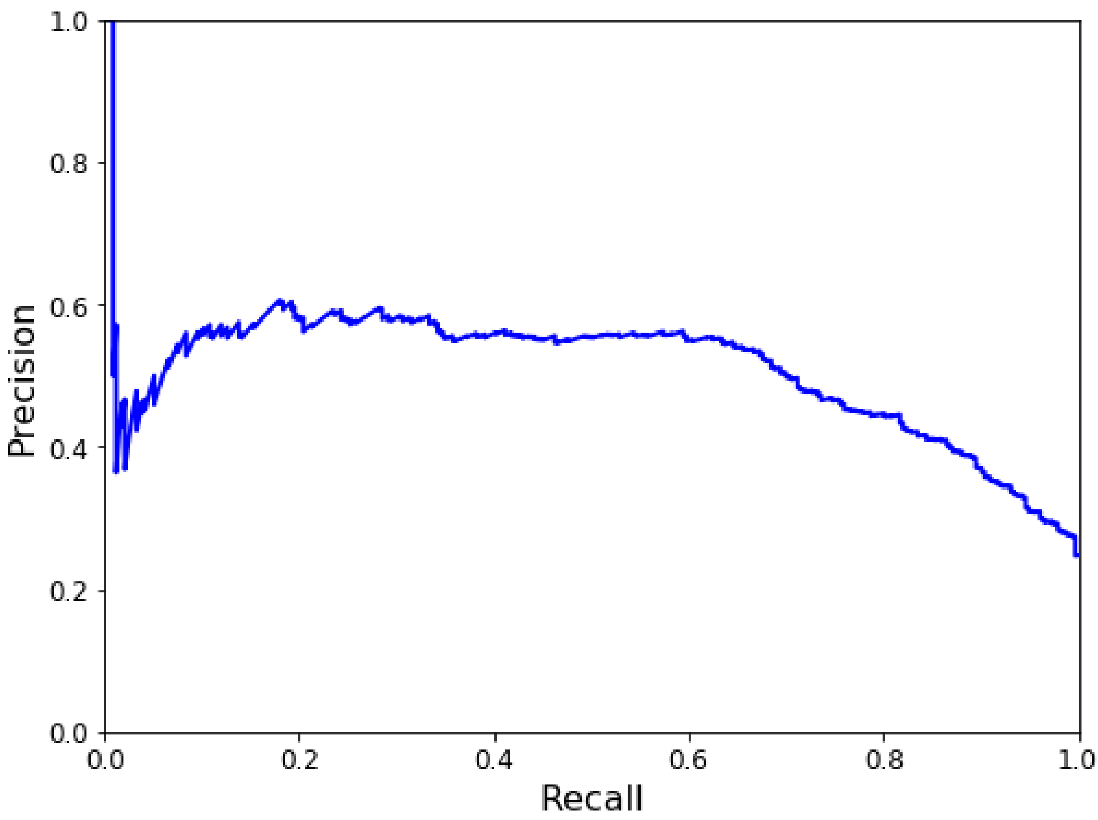
```
In [139]: #hack to work around issue #9589 introduced in Scikit-Learn 0.19.0
          if y_logscores.ndim == 2:
              y_logscores = y_logscores[:, 1]
```

```
In [140]: from sklearn.metrics import precision_recall_curve

          precisions, recalls, thresholds = precision_recall_curve(y_train_sl, y_logscores)
```

```
In [141]: def plot_precision_recall_vs_threshold(precisions, recalls, thresholds):
              plt.plot(thresholds, precisions[:-1], "b--", label="Precision", linewidth=2)
              plt.plot(thresholds, recalls[:-1], "g-", label="Recall", linewidth=2)
              plt.xlabel("Threshold", fontsize=16)
              plt.legend(loc="lower right", fontsize=16)
              plt.ylim([0, 1.1])

          plt.figure(figsize=(8, 4))
          plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
          plt.xlim([-5,5])
          plt.show()
```




```
In [142]: def plot_precision_vs_recall(precisions, recalls):
              plt.plot(recalls, precisions, "b-", linewidth=2)
              plt.xlabel("Recall", fontsize=16)
              plt.ylabel("Precision", fontsize=16)
              plt.axis([0, 1, 0, 1])

          plt.figure(figsize=(8, 6))
          plot_precision_vs_recall(precisions, recalls)
          plt.show()
```




```
In [143]: from sklearn.metrics import roc_auc_score
          roc_auc_score(y_train_sl, y_logscores, average = 'weighted')
```

```
Out[143]: 0.805966799344282722
```

Random Forest, Decision Tree does not have a decision function but a predict_proba method instead.

```
In [144]: from sklearn.ensemble import RandomForestClassifier

          roc_forest = RandomForestClassifier(random_state = 42)
          y_probas_forest = cross_val_predict(roc_forest, X_train, y_train_sl, cv =5, method =
```

```
In [145]: from sklearn.metrics import roc_curve

          y_scores_forest = y_probas_forest[:,1] #score = proba of positive class
          fpr_forest, tpr_forest, thresholds_forest= roc_curve(y_train_sl, y_scores_forest)
```

```
In [146]: from sklearn.tree import DecisionTreeClassifier

          roc_tree= DecisionTreeClassifier(random_state = 42)
          y_probas_tree = cross_val_predict(roc_tree, X_train, y_train_sl, cv =5, method = "pr
```

```
In [147]: roc_auc_score(y_train_sl, y_scores_forest, average ='weighted') #Best ROC-AUC Score
```

```
Out[147]: 0.97054670564604339
```

```
In [148]: y_scores_tree = y_probas_tree[:,1] #score = proba of positive class
          fpr_tree, tpr_tree, thresholds_tree = roc_curve(y_train_sl, y_scores_tree)
```

```
In [149]: roc_auc_score(y_train_sl, y_scores_tree, average = 'weighted')
```

```
Out[149]: 0.8955966373184916
```

```
In [150]: fpr_log, tpr_log, thresholds_log = roc_curve(y_train_sl, y_logscores)
```

```
In [151]: y_probas_gnb = cross_val_predict(gnb_clf_n, X_train, y_train_sl, cv =5, method = "pr
```

```
In [152]: y_scores_gnb = y_probas_gnb[:,1]
          fpr_gnb, tpr_gnb, thresholds_gnb = roc_curve(y_train_sl, y_scores_gnb)
```

```
In [153]: roc_auc_score(y_train_sl, y_scores_gnb, average = 'weighted')
```

```
Out[153]: 0.73798869825359892
```

```
In [154]: y_probas_knn = cross_val_predict(knn_clf_n, X_train, y_train_sl, cv =5, method = "pr
```

```
In [155]: y_scores_knn = y_probas_knn[:,1]
          fpr_knn, tpr_knn, thresholds_knn = roc_curve(y_train_sl, y_scores_knn)
```

```
In [156]: roc_auc_score(y_train_sl, y_scores_knn, average = 'weighted')
```

```
Out[156]: 0.91793449078217293
```

```
In [157]: y_svmscores = cross_val_predict(rbf_kernel_svm_clf, X_train, y_train_sl, cv =5, meth
```

```
In [158]: #hack to work around issue #9589 introduced in Scikit-Learn 0.19.0
          if y_svmscores.ndim == 2:
              y_svmscores = y_svmscores[:, 1]
```

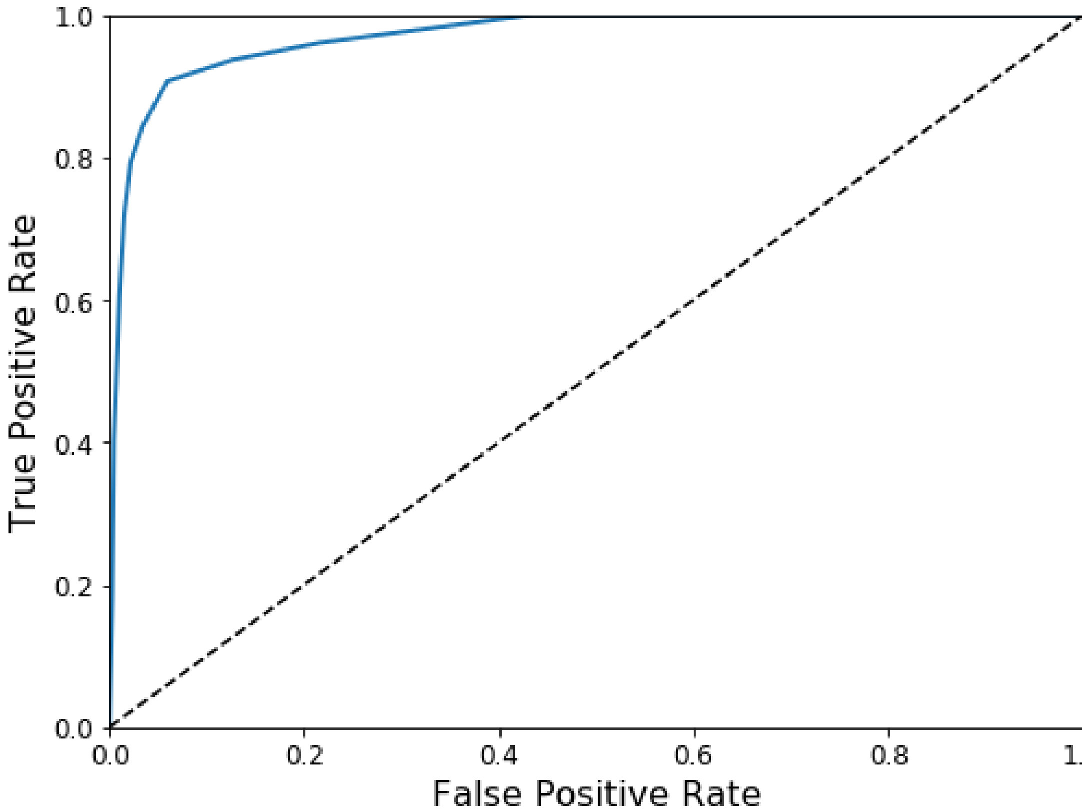
```
In [159]: fpr_svm, tpr_svm, thresholds_svm = roc_curve(y_train_sl, y_svmscores)
```

```
In [160]: roc_auc_score(y_train_sl, y_svmscores, average = 'weighted')
```

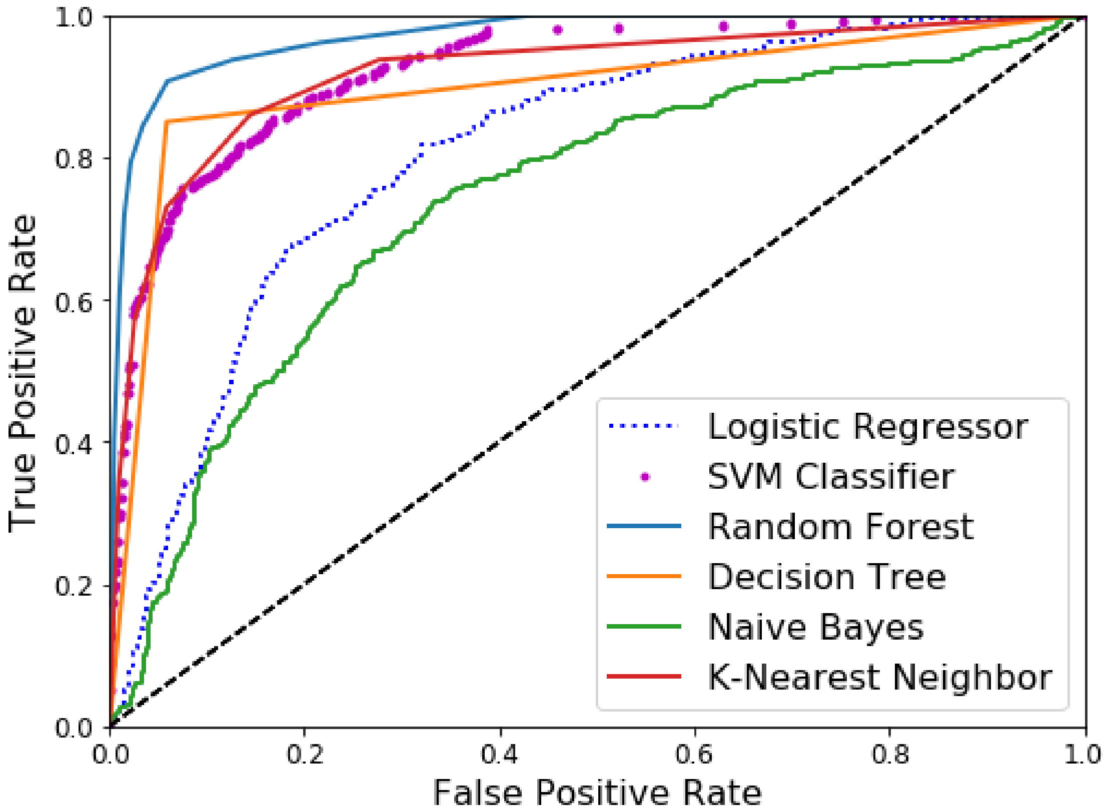
```
Out[160]: 0.92393339413206976
```

ROC Curve for Random Forest alone:

```
In [161]: def plot_roc_curve(fpr, tpr, label=None):
              plt.plot(fpr, tpr, linewidth=2, label=label)
              plt.plot([0, 1], [0, 1], 'k--')
              plt.axis([0, 1, 0, 1])
              plt.xlabel('False Positive Rate', fontsize=16)
              plt.ylabel('True Positive Rate', fontsize=16)

          plt.figure(figsize=(8, 6))
          plot_roc_curve(fpr_forest, tpr_forest)
          plt.show()
```



In [162]:
```
plt.figure(figsize=(8, 6))
plt.plot(fpr_log, tpr_log, "b:", linewidth=2, label="Logistic Regressor")
plt.plot(fpr_svm, tpr_svm, "m.", linewidth=1, label="SVM Classifier")
plot_roc_curve(fpr_forest, tpr_forest, "Random Forest")
plot_roc_curve(fpr_tree, tpr_tree, "Decision Tree")
plot_roc_curve(fpr_gnb, tpr_gnb, "Naive Bayes")
plot_roc_curve(fpr_knn, tpr_knn, "K-Nearest Neighbor")
plt.legend(loc="lower right", fontsize=16)
plt.show()
```



Grid search on three best classifiers:
KNN on shaly limestone formation:

```
In [163]: from sklearn.model_selection import GridSearchCV

          weights_sl = ['uniform','distance']
          numNeighbors_sl = np.array([3,5,7,9])

In [164]: param_grid_knn_sl  = dict(weights=weights_sl,n_neighbors=numNeighbors_sl)

In [165]: grid_knn_sl = GridSearchCV(knn_clf_n,param_grid=param_grid_knn_sl,cv=5, n_jobs =-1)

In [166]: grid_knn_sl.fit(X_train,y_train_sl)

Out[166]: GridSearchCV(cv=5, error_score='raise',
                 estimator=KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkows
                     metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                     weights='uniform'),
                 fit_params=None, iid=True, n_jobs=-1,
                 param_grid={'weights': ['uniform', 'distance'], 'n_neighbors': array([3, 5, 7
                 pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
                 scoring=None, verbose=0)
```

```
In [167]: grid_knn_sl.best_params_
```

```
Out[167]: {'n_neighbors': 5, 'weights': 'distance'}
```

```
In [168]: grid_knn_sl.best_estimator_
```

```
Out[168]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                     weights='distance')
```

```
In [169]: knn_clf_sl_GS = KNeighborsClassifier(n_neighbors = 5, weights = 'distance')
```

```
In [170]: knn_clf_sl_GS.fit(X_train, y_train_sl)
```

```
Out[170]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                     weights='distance')
```

```
In [171]: y_pred_knn_sl  = cross_val_predict(knn_clf_sl_GS, X_train, y_train_sl, cv=5)
          f1_score(y_train_sl, y_pred_knn_sl , average="weighted")
```

```
Out[171]: 0.900765228938344 54
```

```
In [172]: y_test_knn_pred = knn_clf_sl_GS.predict(X_test_prepared)
```

```
In [173]: f1_score(y_test_sl, y_test_knn_pred , average="weighted")
```

```
Out[173]: 0.91092555406127373
```

```
In [174]: roc_auc_score(y_test_sl, y_test_knn_pred, average = 'weighted')
```

```
Out[174]: 0.87995686079309776
```

Comparing without grid search:

```
In [175]: y_test_knn_noGS = knn_clf_n.predict(X_test_prepared)
```

```
In [176]: f1_score(y_test_sl, y_test_knn_noGS, average = 'weighted')
```

```
Out[176]: 0.90498621431611748
```

```
In [177]: roc_auc_score(y_test_sl, y_test_knn_noGS, average = 'weighted')
```

```
Out[177]: 0.87055126928820303
```

ROC Curve performance prep for KNN on test set:

```
In [178]: y_probas_knn_sl = knn_clf_sl_GS.predict_proba(X_test_prepared)
```

```
In [179]: y_scores_knn_sl = y_probas_knn_sl[:, 1]
```

```
In [180]: from sklearn.metrics import roc_curve

          fpr_knn_sl, tpr_knn_sl, thresholds_knn_sl = roc_curve(y_test_sl, y_scores_knn_sl)
```

Gaussian RBF SVM Grid search:

```
In [181]: Cs = [5, 100, 200]
          kernels = ['linear', 'rbf']
          decision = ['ovo', 'ovr']

In [182]: param_grid_svm_sl  = dict(C=Cs,kernel = kernels, decision_function_shape =decision)

In [183]: grid_svm_sl = GridSearchCV(rbf_kernel_svm_clf,param_grid=param_grid_svm_sl,cv=5, n_jo

In [184]: grid_svm_sl.fit(X_train, y_train_sl)

Out[184]: GridSearchCV(cv=5, error_score='raise',
                 estimator=SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
            decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
            max_iter=-1, probability=True, random_state=None, shrinking=True,
            tol=0.001, verbose=False),
                 fit_params=None, iid=True, n_jobs=-1,
                 param_grid={'C': [5, 100, 200], 'kernel': ['linear', 'rbf'], 'decision_functio
                 pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
                 scoring=None, verbose=0)

In [185]: grid_svm_sl.best_params_

Out[185]: {'C': 100, 'decision_function_shape': 'ovo', 'kernel': 'rbf'}

In [186]: grid_svm_sl.best_estimator_

Out[186]: SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,
            decision_function_shape='ovo', degree=3, gamma='auto', kernel='rbf',
            max_iter=-1, probability=True, random_state=None, shrinking=True,
            tol=0.001, verbose=False)

In [187]: svm_clf_sl_GS = SVC(C= 100, decision_function_shape = 'ovo', random_state =41, probab

In [188]: svm_clf_sl_GS.fit(X_train, y_train_sl)

Out[188]: SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,
            decision_function_shape='ovo', degree=3, gamma='auto', kernel='rbf',
            max_iter=-1, probability=True, random_state=41, shrinking=True,
            tol=0.001, verbose=False)

In [189]: y_pred_svm_sl  = cross_val_predict(svm_clf_sl_GS, X_train, y_train_sl, cv=5, n_jobs=
          f1_score(y_train_sl, y_pred_svm_sl , average="weighted")

Out[189]: 0.92902906979407762
```

```
In [190]: y_test_svm_pred = svm_clf_sl_GS.predict(X_test_prepared)

In [191]: f1_score(y_test_sl, y_test_svm_pred , average="weighted")

Out[191]: 0.93652014192326571

In [192]: roc_auc_score(y_test_sl, y_test_svm_pred, average = 'weighted')

Out[192]: 0.92164426746308281
```

Comparing with previous version before Grid Search:

```
In [193]: rbf_kernel_svm_clf.fit(X_train, y_train_sl)

Out[193]: SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
            decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
            max_iter=-1, probability=True, random_state=None, shrinking=True,
            tol=0.001, verbose=False)

In [194]: y_test_svm_noGS = rbf_kernel_svm_clf.predict(X_test_prepared)

In [195]: f1_score(y_test_sl, y_test_svm_noGS , average="weighted")

Out[195]: 0.88354820395494993

In [196]: roc_auc_score(y_test_sl, y_test_svm_noGS, average = 'weighted')

Out[196]: 0.8271113323378132
```

ROC Curve preparation for SVM Classifier on Shaly Limestone:

```
In [197]: y_probas_svm_sl = rbf_kernel_svm_clf.predict_proba(X_test_prepared)

In [198]: y_scores_svm_sl = y_probas_svm_sl[:, 1] # score = proba of positive class
          fpr_svm_sl, tpr_svm_sl, thresholds_svm_sl = roc_curve(y_test_sl,y_scores_svm_sl)
```

RandomForest on shaly limestone grid search

```
In [199]: numEstim_for_sl = [200, 400, 500]
          criteria_for_sl = ['gini', 'entropy']

In [200]: param_grid_for_sl  = dict(n_estimators = numEstim_for_sl, criterion = criteria_for_s

In [201]: grid_for_sl = GridSearchCV(forest_clf_n,param_grid=param_grid_for_sl,cv=5, n_jobs =-

In [202]: grid_for_sl.fit(X_train, y_train_sl)
```

```
Out[202]: GridSearchCV(cv=5, error_score='raise',
                 estimator=RandomForestClassifier(bootstrap=True, class_weight=None, criterion=
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
                      oob_score=False, random_state=42, verbose=0, warm_start=False),
                 fit_params=None, iid=True, n_jobs=-1,
                 param_grid={'n_estimators': [200, 400, 500], 'criterion': ['gini', 'entropy']]
                 pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
                 scoring=None, verbose=0)
```

```
In [203]: grid_for_sl.best_params_
```

```
Out[203]: {'criterion': 'gini', 'n_estimators': 500}
```

```
In [204]: grid_for_sl.best_estimator_
```

```
Out[204]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1,
                      oob_score=False, random_state=42, verbose=0, warm_start=False)
```

```
In [205]: for_clf_sl_GS = RandomForestClassifier(n_estimators = 500, random_state =42)
```

```
In [206]: for_clf_sl_GS.fit(X_train, y_train_sl)
```

```
Out[206]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1,
                      oob_score=False, random_state=42, verbose=0, warm_start=False)
```

```
In [207]: y_pred_for_sl  = cross_val_predict(for_clf_sl_GS, X_train, y_train_sl, cv=5, n_jobs=
          f1_score(y_train_sl, y_pred_for_sl , average="weighted")
```

```
Out[207]: 0.94498612769060919
```

```
In [208]: y_test_for_pred = for_clf_sl_GS.predict(X_test_prepared)
```

```
In [209]: f1_score(y_test_sl, y_test_for_pred , average="weighted")
```

```
Out[209]: 0.95460353713498058
```

```
In [210]: roc_auc_score(y_test_sl, y_test_for_pred, average = 'weighted')
```

```
Out[210]: 0.92324124771859972
```

Comparing without GridSearch:

```
In [211]: y_test_for_noGS= forest_clf_n.predict(X_test_prepared)

In [212]: f1_score(y_test_sl, y_test_for_noGS, average="weighted")

Out[212]: 0.95636735796639238

In [213]: roc_auc_score(y_test_sl, y_test_for_noGS, average = 'weighted')

Out[213]: 0.92172722747635638
```

ROC Curve prepration of Random Forest on Shaly Limestone:

```
In [214]: y_probas_forest_sl = for_clf_sl_GS.predict_proba(X_test_prepared)

In [215]: y_scores_forest_sl = y_probas_forest_sl[:, 1] # score = proba of positive class
          fpr_forest_sl, tpr_forest_sl, thresholds_forest_sl = roc_curve(y_test_sl,y_scores_fo
```

Voting on best classifiers after GridSearch:

```
In [216]: from sklearn.ensemble import VotingClassifier

          voting_clf_sl = VotingClassifier(estimators = [('svmGS', svm_clf_sl_GS),
                                                          ('knnGSsl', knn_clf_sl_GS),
                                                          ('rnfGS', for_clf_sl_GS)], voting = '

In [217]: voting_clf_sl.fit(X_train, y_train_sl)

Out[217]: VotingClassifier(estimators=[('svmGS', SVC(C=100, cache_size=200, class_weight=None,
            decision_function_shape='ovo', degree=3, gamma='auto', kernel='rbf',
            max_iter=-1, probability=True, random_state=41, shrinking=True,
            tol=0.001, verbose=False)), ('knnGSsl', KNeighborsClassifier(algorithm='auto', lea
                       oob_score=False, random_state=42, verbose=0, warm_start=False))],
                  flatten_transform=None, n_jobs=1, voting='soft', weights=None)

In [218]: y_pred_voting_sl  = cross_val_predict(voting_clf_sl, X_train, y_train_sl, cv=5, n_jo
          f1_score(y_train_sl, y_pred_voting_sl , average="weighted")

Out[218]: 0.93531625139763241

In [219]: y_test_voting_pred = voting_clf_sl.predict(X_test_prepared)

In [220]: f1_score(y_test_sl, y_test_voting_pred , average="weighted")

Out[220]: 0.95138816147323446

In [221]: roc_auc_score(y_test_sl, y_test_voting_pred)

Out[221]: 0.93184834909573577
```

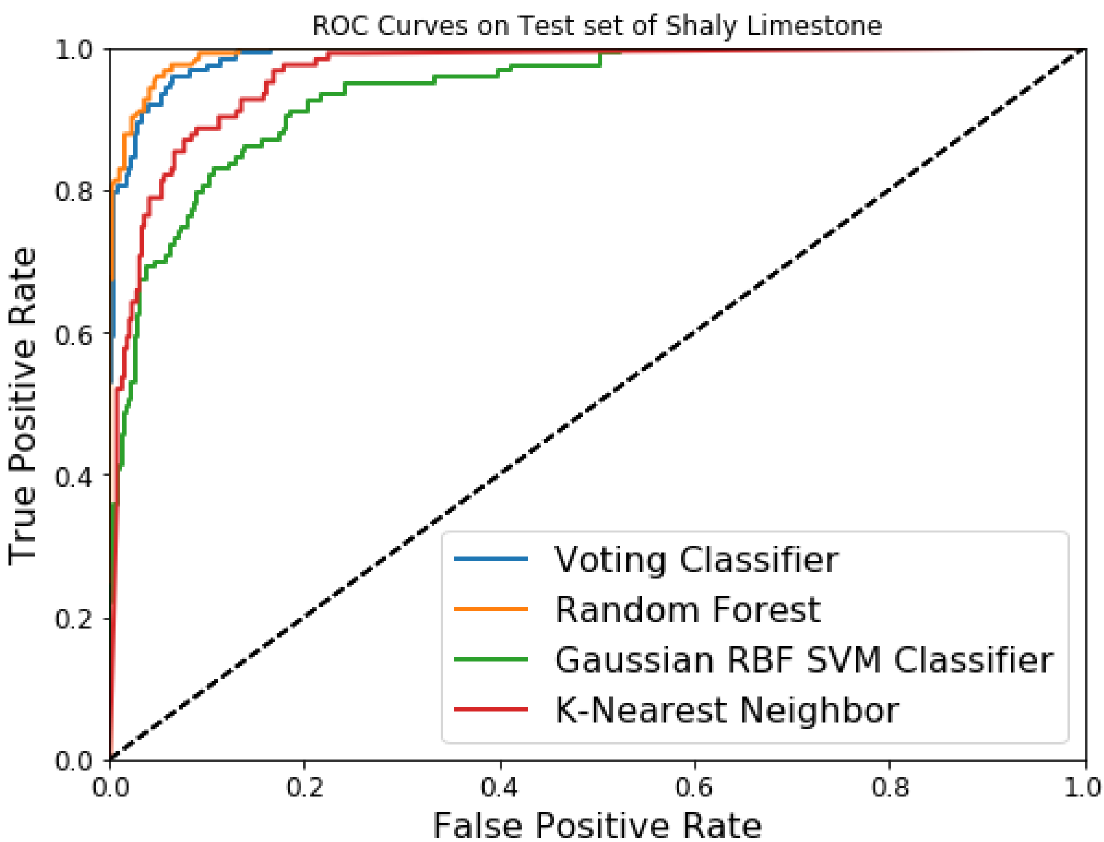
ROC Curve of Shaly Limestone of three best classifiers on test set:

```
In [222]: y_probas_voting_sl = voting_clf_sl.predict_proba(X_test_prepared)
```

```
In [223]: y_scores_voting_sl = y_probas_voting_sl[:, 1] # score = proba of positive class
          fpr_voting_sl, tpr_voting_sl, thresholds_voting_sl = roc_curve(y_test_sl,y_scores_vo
```

ROC Curves on Test Set:

```
In [224]: plt.figure(figsize=(8, 6))
          plot_roc_curve(fpr_voting_sl, tpr_voting_sl, "Voting Classifier")
          plot_roc_curve(fpr_forest_sl, tpr_forest_sl, "Random Forest")
          plot_roc_curve(fpr_svm_sl, tpr_svm_sl, "Gaussian RBF SVM Classifier")
          plot_roc_curve(fpr_knn_sl, tpr_knn_sl, "K-Nearest Neighbor")
          plt.legend(loc="lower right", fontsize=16)
          plt.title("ROC Curves on Test set of Shaly Limestone")
          plt.show()
```



Grid Search Feature importances with Shaly Limestone:

```
In [225]: feature_importances_for = grid_for_sl.best_estimator_.feature_importances_
```

```
In [226]: attributes = ["Depth", "Neutron Porosity", "Caliper", "Density Porosity", "Gamma Ray"
                        "Bulk Density", "Density Correction", "Resistivity (Deep)", "Resistivity
                        "Ratio(Shallow/Deep resistivity)", "SP", "Micro-inverse (resistivity) mi
                         "Micro-normal (resistivity) micro-log", "Delta-t (transit time)", "Soni
          sorted(zip(feature_importances_for*100, attributes), reverse=True)
```

```
Out[226]: [(32.698676805752939, 'Depth'),
           (11.437112671467331, 'Gamma Ray'),
           (7.0256677233376346, 'SP'),
           (5.2919610063076963, 'Density Correction'),
           (4.69972333501756, 'Resistivity (Deep)'),
           (4.3250282898616081, 'Caliper'),
           (3.876972448849632, 'Micro-normal (resistivity) micro-log'),
           (3.8373287915693957, 'Resistivity (Medium)'),
           (3.6854852718774076, 'Photoelctric'),
           (3.6552972972995796, 'Sonic Porosity'),
           (3.5872666770164936, 'Micro-inverse (resistivity) micro-log'),
           (3.5667515140079744, 'Delta-t (transit time)'),
           (2.7913023534006003, 'Resistivity (Shallow)'),
           (2.714697087901786, 'Neutron Porosity'),
           (2.4667204236786993, 'Ratio(Shallow/Deep resistivity)'),
           (2.1928475922610642, 'Density Porosity'),
           (2.1471607103925967, 'Bulk Density')]
```

Performance on the rest of the classes:
Making new test and training sets for rest of classes first:

```
In [227]: y_train.head()
```

```
Out[227]:       Type of Formation_dolomite  Type of Formation_limestone  \
          364                            0                            1
          1119                           0                            1
          974                            0                            0
          481                            0                            0
          828                            0                            1

                Type of Formation_sandstone  Type of Formation_sandy limestone  \
          364                             0                                  0
          1119                            0                                  0
          974                             0                                  1
          481                             0                                  0
          828                             0                                  0

                Type of Formation_shale  Type of Formation_shaly limestone  \
          364                         0                                  0
          1119                        0                                  0
          974                         0                                  0
          481                         1                                  0
          828                         0                                  0
```

```
      Type of Formation_shaly sandstone
364                                   0
1119                                  0
974                                   0
481                                   0
828                                   0
```

```
In [228]: y_train_lim = y_train['Type of Formation_limestone']
          y_train_lim.head()
```

```
Out[228]: 364     1
          1119    1
          974     0
          481     0
          828     1
          Name: Type of Formation_limestone, dtype: uint8
```

```
In [229]: y_train_sand = y_train['Type of Formation_sandstone']
          y_train_sand.head()
```

```
Out[229]: 364     0
          1119    0
          974     0
          481     0
          828     0
          Name: Type of Formation_sandstone, dtype: uint8
```

```
In [230]: y_train_dol = y_train['Type of Formation_dolomite']
          y_train_dol.head()
```

```
Out[230]: 364     0
          1119    0
          974     0
          481     0
          828     0
          Name: Type of Formation_dolomite, dtype: uint8
```

```
In [231]: y_train_lim = y_train['Type of Formation_limestone']
          y_train_lim.head()
```

```
Out[231]: 364     1
          1119    1
          974     0
          481     0
          828     1
          Name: Type of Formation_limestone, dtype: uint8
```

```
In [232]: y_train_sandlim = y_train['Type of Formation_sandy limestone']
          y_train_sandlim.head()
```

```
Out[232]: 364     0
          1119    0
          974     1
          481     0
          828     0
          Name: Type of Formation_sandy limestone, dtype: uint8
```

```
In [233]: y_train_shale = y_train['Type of Formation_shale']
          y_train_shale.head()
```

```
Out[233]: 364     0
          1119    0
          974     0
          481     1
          828     0
          Name: Type of Formation_shale, dtype: uint8
```

```
In [234]: y_train_ss = y_train['Type of Formation_shaly sandstone']
          y_train_ss.head()
```

```
Out[234]: 364     0
          1119    0
          974     0
          481     0
          828     0
          Name: Type of Formation_shaly sandstone, dtype: uint8
```

```
In [235]: y_test_new.head()
```

```
Out[235]:       Type of Formation_dolomite  Type of Formation_limestone  \
          1501                           0                            1
          377                            0                            1
          1025                           0                            0
          819                            0                            1
          1364                           0                            0

                Type of Formation_sandstone  Type of Formation_sandy limestone  \
          1501                            0                                  0
          377                             0                                  0
          1025                            0                                  0
          819                             0                                  0
          1364                            0                                  0

                Type of Formation_shale  Type of Formation_shaly limestone  \
          1501                        0                                  0
          377                         0                                  0
          1025                        1                                  0
          819                         0                                  0
          1364                        1                                  0
```

```
      Type of Formation_shaly sandstone
1501                                  0
377                                   0
1025                                  0
819                                   0
1364                                  0
```

```
In [236]: y_test_sand = y_test_new['Type of Formation_sandstone']
          y_test_sand.head()
```

```
Out[236]: 1501    0
          377     0
          1025    0
          819     0
          1364    0
          Name: Type of Formation_sandstone, dtype: uint8
```

```
In [237]: y_test_dol = y_test_new['Type of Formation_dolomite']
          y_test_dol.head()
```

```
Out[237]: 1501    0
          377     0
          1025    0
          819     0
          1364    0
          Name: Type of Formation_dolomite, dtype: uint8
```

```
In [238]: y_test_lim = y_test_new['Type of Formation_limestone']
          y_test_lim.head()
```

```
Out[238]: 1501    1
          377     1
          1025    0
          819     1
          1364    0
          Name: Type of Formation_limestone, dtype: uint8
```

```
In [239]: y_test_sandlim = y_test_new['Type of Formation_sandy limestone']
          y_test_sandlim.head()
```

```
Out[239]: 1501    0
          377     0
          1025    0
          819     0
          1364    0
          Name: Type of Formation_sandy limestone, dtype: uint8
```

```
In [240]: y_test_shale = y_test_new['Type of Formation_shale']
          y_test_shale.head()
```

Out[240]:
```
1501    0
377     0
1025    1
819     0
1364    1
Name: Type of Formation_shale, dtype: uint8
```

In [241]:
```
y_test_ss = y_test_new['Type of Formation_shaly sandstone']
y_test_ss.head()
```

Out[241]:
```
1501    0
377     0
1025    0
819     0
1364    0
Name: Type of Formation_shaly sandstone, dtype: uint8
```

Feature importances on rest of the classes:

In [242]:
```
y_trains_classes= (y_train_sl, y_train_lim, y_train_shale, y_train_sandlim,
                   y_train_ss, y_train_dol, y_train_sand)
y_classes_names = ("shaly limestone", "limestone", "shale", "sandy lime",
                   "shaly sandstone", "dolomite", "sandstone")
y_test_classes = (y_test_sl, y_test_lim, y_test_shale, y_test_sandlim, y_test_ss, y_
```

In [243]:
```
numEstim_for = [200, 400, 500]
criteria_for = ['gini', 'entropy']
param_grid_for  = dict(n_estimators = numEstim_for, criterion = criteria_for)
for_clf_fi = RandomForestClassifier(random_state =42)
attributes_all = ["Depth" ,"Neutron Porosity","Caliper ", "Density Porosity","Gamma 
                     "Photoelectric", "Bulk Density", "Density Correction", "Resist
                      "Resistivity (Medium)", "Resistivity (Shallow)","Ratio(Shallo
                     "SP",  "Micro-inverse (resistivity) micro-log", "Micro-normal 
                   "Delta-t (transit time)", "Sonic Porosity"]
```

Feature importances of limestone:

In [244]:
```
grid_for_lim = GridSearchCV(for_clf_fi,param_grid=param_grid_for,cv=5, n_jobs =-1)
grid_for_lim.fit(X_train, y_train_lim)
feature_importances_lim = grid_for_lim.best_estimator_.feature_importances_
sorted(zip(feature_importances_lim*100, attributes_all), reverse=True)
```

Out[244]:
```
[(20.490873610563774, 'Depth'),
 (12.09548368712561, 'Photoelectric'),
 (7.5876354108716608, 'Gamma Ray'),
 (7.199221662111591, 'Sonic Porosity'),
 (5.9924574197004841, 'SP'),
 (5.6892619027191973, 'Density Correction'),
 (5.4727581603199775, 'Delta-t (transit time)'),
```

```
 (4.037953279469467, 'Neutron Porosity'),
 (3.9772708914231534, 'Resistivity (Deep)'),
 (3.9157159352337554, 'Bulk Density'),
 (3.8785590650677042, 'Resistivity (Medium)'),
 (3.8378661049938105, 'Density Porosity'),
 (3.8082657018084038, 'Micro-inverse (resistivity) micro-log'),
 (3.4521001554844304, 'Micro-normal (resistivity) micro-log'),
 (3.2643772249487251, 'Caliper '),
 (2.6545476810977231, 'Resistivity (Shallow)'),
 (2.645652107060513, 'Ratio(Shallow/Deep resistivity)')]
```

Feature importances of shale:

```
In [245]: grid_for_shale = GridSearchCV(for_clf_fi,param_grid=param_grid_for,cv=5, n_jobs =-1)
          grid_for_shale.fit(X_train, y_train_shale)
          feature_importances_shale = grid_for_shale.best_estimator_.feature_importances_
          sorted(zip(feature_importances_shale*100, attributes_all), reverse=True)
```

```
Out[245]: [(20.950655660649275, 'Gamma Ray'),
           (8.7649983045863511, 'Depth'),
           (7.8745311068874013, 'SP'),
           (7.544206710447952, 'Sonic Porosity'),
           (6.632776920956335, 'Delta-t (transit time)'),
           (6.1290876680814366, 'Density Correction'),
           (5.1297756155294962, 'Caliper '),
           (4.7669950717085996, 'Micro-normal (resistivity) micro-log'),
           (4.6914561836071549, 'Resistivity (Deep)'),
           (4.6769052063688381, 'Resistivity (Medium)'),
           (3.9460133944234212, 'Micro-inverse (resistivity) micro-log'),
           (3.417788758909404, 'Ratio(Shallow/Deep resistivity)'),
           (3.2810824051178686, 'Photoelectric'),
           (3.2502028490509072, 'Neutron Porosity'),
           (3.1743094666705343, 'Resistivity (Shallow)'),
           (2.8865801898158385, 'Density Porosity'),
           (2.8826344871891747, 'Bulk Density')]
```

Feature importances of Sandy-Limestone:

```
In [246]: grid_for_sandlim = GridSearchCV(for_clf_fi,param_grid=param_grid_for,cv=5, n_jobs =-
          grid_for_sandlim.fit(X_train, y_train_sandlim)
          feature_importances_sandlim = grid_for_sandlim.best_estimator_.feature_importances_
          sorted(zip(feature_importances_sandlim*100, attributes_all), reverse=True)
```

```
Out[246]: [(14.707880882714836, 'Resistivity (Shallow)'),
           (13.364730255845391, 'Bulk Density'),
           (11.461630168056711, 'Resistivity (Deep)'),
           (11.340180227209865, 'Density Porosity'),
           (9.2415438757067445, 'Resistivity (Medium)'),
           (8.8391414847332239, 'Neutron Porosity'),
```

```
 (5.7003725884107324, 'Sonic Porosity'),
 (5.5837958853639584, 'Delta-t (transit time)'),
 (4.1046578990570186, 'Micro-normal (resistivity) micro-log'),
 (3.2183081956192647, 'SP'),
 (2.6933720135368087, 'Depth'),
 (2.4151182886140723, 'Micro-inverse (resistivity) micro-log'),
 (2.3275541139553777, 'Ratio(Shallow/Deep resistivity)'),
 (1.9521452493236229, 'Photoelectric'),
 (1.7586431580477573, 'Gamma Ray'),
 (0.94090848517096937, 'Density Correction'),
 (0.35001722863365575, 'Caliper ')]
```

Feature importances of Shaly Sandstone:

```
In [247]: grid_for_ss = GridSearchCV(for_clf_fi,param_grid=param_grid_for,cv=4, n_jobs =-1)
          grid_for_ss.fit(X_train, y_train_ss)
          feature_importances_ss = grid_for_ss.best_estimator_.feature_importances_
          sorted(zip(feature_importances_ss*100, attributes_all), reverse=True)
```

```
Out[247]: [(21.596809649545197, 'Depth'),
 (15.914938857340951, 'Photoelectric'),
 (7.1646710110631329, 'Bulk Density'),
 (7.136841528247909, 'SP'),
 (6.1458162965539538, 'Density Correction'),
 (5.9321190717924184, 'Density Porosity'),
 (5.7715629320794433, 'Ratio(Shallow/Deep resistivity)'),
 (5.0460116670441417, 'Gamma Ray'),
 (4.0500786800342006, 'Caliper '),
 (3.2224969769496266, 'Micro-normal (resistivity) micro-log'),
 (3.1413058863587016, 'Resistivity (Deep)'),
 (3.0305749558090955, 'Neutron Porosity'),
 (2.8731256647119627, 'Resistivity (Medium)'),
 (2.5257474251301901, 'Sonic Porosity'),
 (1.9758618619006039, 'Delta-t (transit time)'),
 (1.5080152949014241, 'Micro-inverse (resistivity) micro-log'),
 (1.4640222405370318, 'Resistivity (Shallow)')]
```

Feature importances of Dolomite:

```
In [248]: grid_for_dol = GridSearchCV(for_clf_fi,param_grid=param_grid_for,cv=5, n_jobs =-1)
          grid_for_dol.fit(X_train, y_train_dol)
          feature_importances_dol = grid_for_dol.best_estimator_.feature_importances_
          sorted(zip(feature_importances_dol*100, attributes_all), reverse=True)
```

```
Out[248]: [(53.509366629544509, 'Depth'),
 (11.949567088220114, 'Caliper '),
 (6.1893849633695854, 'SP'),
 (5.9758459514743212, 'Gamma Ray'),
 (3.9081768771959293, 'Photoelectric'),
```

```
 (3.5748245958727347, 'Neutron Porosity'),
 (2.0693306345007256, 'Resistivity (Deep)'),
 (2.0206560970330965, 'Ratio(Shallow/Deep resistivity)'),
 (1.9364464930680714, 'Density Porosity'),
 (1.8154415745453487, 'Bulk Density'),
 (1.6393981485793421, 'Density Correction'),
 (1.5617315647734751, 'Resistivity (Medium)'),
 (1.1101369349903425, 'Delta-t (transit time)'),
 (0.7678410807304582, 'Resistivity (Shallow)'),
 (0.75319282785050801, 'Sonic Porosity'),
 (0.68842656439892325, 'Micro-normal (resistivity) micro-log'),
 (0.53023197385250531, 'Micro-inverse (resistivity) micro-log')]
```

Feature importances of Sandstone:

```
In [249]: grid_for_sand = GridSearchCV(for_clf_fi,param_grid=param_grid_for,cv=5, n_jobs =-1)
          grid_for_sand.fit(X_train, y_train_sand)
          feature_importances_sand = grid_for_sand.best_estimator_.feature_importances_
          sorted(zip(feature_importances_sand*100, attributes_all), reverse=True)
```

```
Out[249]: [(36.309046019421423, 'Photoelectric'),
 (16.988444654943709, 'Depth'),
 (10.625009294149367, 'Gamma Ray'),
 (7.0372921222701041, 'Bulk Density'),
 (6.2082456104717005, 'Neutron Porosity'),
 (6.1994382438241598, 'Density Porosity'),
 (2.3100252234031458, 'Caliper '),
 (2.1072911414102631, 'Delta-t (transit time)'),
 (1.9954540514952033, 'Sonic Porosity'),
 (1.994056257803549, 'Resistivity (Deep)'),
 (1.8722947450454466, 'SP'),
 (1.5215307078817599, 'Density Correction'),
 (1.1586896609418966, 'Resistivity (Medium)'),
 (1.0650840258643131, 'Resistivity (Shallow)'),
 (0.98184477645290591, 'Ratio(Shallow/Deep resistivity)'),
 (0.92377160122838375, 'Micro-normal (resistivity) micro-log'),
 (0.70248186339264063, 'Micro-inverse (resistivity) micro-log')]
```

Automation for training all classes with all classifiers applied uptil now:

```
In [250]: from sklearn.svm import SVC

          svm_clf_GS = SVC(C= 100, decision_function_shape = 'ovo', random_state =41, probabili

          for y_train_all, y_test_all, y_strings_all in zip(y_trains_classes,
                                                            y_test_classes, y_classes_names)
              svm_clf_GS.fit(X_train, y_train_all)
              y_cv_svm = cross_val_score(svm_clf_GS, X_train, y_train_all, cv = 4)
              print("\n","cross-validation score on training set for", y_strings_all, "with SVM
```

```
y_cvp_svm = cross_val_predict(svm_clf_GS, X_train, y_train_all, cv=4)
f1_cvp_train_svm = f1_score(y_train_all, y_cvp_svm, average = 'weighted', labels
print("f1 scores of cross validation prediction training set for", y_strings_all
roc_auc_score_svm_train = roc_auc_score(y_train_all, y_cvp_svm, average = 'weigh
print("roc auc score on cross validation prediction training set for",
      y_strings_all,"with SVMClassifier =", roc_auc_score_svm_train)


y_cvp_svm_test = svm_clf_GS.predict(X_test_prepared)
f1_svm_test_all = f1_score(y_test_all, y_cvp_svm_test, average = 'weighted', labe
print("f1 score of actual prediction on test set of", y_strings_all,
      "with SVMClassifier =", f1_svm_test_all)
roc_auc_score_svm_test_all = roc_auc_score(y_test_all, y_cvp_svm_test, average =
print("roc auc score of actual prediction on test set of",y_strings_all, "with S
      roc_auc_score_svm_test_all)
```

```
 cross-validation score on training set for shaly limestone with SVMClassifier = [ 0.93696275
f1 scores of cross validation prediction training set for shaly limestone with SVMClassifier =
roc auc score on cross validation prediction training set for shaly limestone with SVMClassifi
f1 score of actual prediction on test set of shaly limestone with SVMClassifier = 0.93652014191
roc auc score of actual prediction on test set of shaly limestone with SVMClassifier = 0.92164

 cross-validation score on training set for limestone with SVMClassifier = [ 0.91091954  0.899
f1 scores of cross validation prediction training set for limestone with SVMClassifier = 0.915
roc auc score on cross validation prediction training set for limestone with SVMClassifier = 0
f1 score of actual prediction on test set of limestone with SVMClassifier = 0.939770401009
roc auc score of actual prediction on test set of limestone with SVMClassifier = 0.93985461345

 cross-validation score on training set for shale with SVMClassifier = [ 0.93965517  0.9080459
f1 scores of cross validation prediction training set for shale with SVMClassifier = 0.9265162
roc auc score on cross validation prediction training set for shale with SVMClassifier = 0.826
f1 score of actual prediction on test set of shale with SVMClassifier = 0.952510764485
roc auc score of actual prediction on test set of shale with SVMClassifier = 0.882744783307

 cross-validation score on training set for sandy lime with SVMClassifier = [ 1.          0.99
f1 scores of cross validation prediction training set for sandy lime with SVMClassifier = 0.99
roc auc score on cross validation prediction training set for sandy lime with SVMClassifier =
f1 score of actual prediction on test set of sandy lime with SVMClassifier = 0.996116504854
roc auc score of actual prediction on test set of sandy lime with SVMClassifier = 0.9490099009

 cross-validation score on training set for shaly sandstone with SVMClassifier = [ 0.99712644
f1 scores of cross validation prediction training set for shaly sandstone with SVMClassifier =
roc auc score on cross validation prediction training set for shaly sandstone with SVMClassifi
f1 score of actual prediction on test set of shaly sandstone with SVMClassifier = 1.0
roc auc score of actual prediction on test set of shaly sandstone with SVMClassifier = 1.0
```

```
 cross-validation score on training set for dolomite with SVMClassifier = [ 1.         1.
f1 scores of cross validation prediction training set for dolomite with SVMClassifier = 0.9992
roc auc score on cross validation prediction training set for dolomite with SVMClassifier = 0.
f1 score of actual prediction on test set of dolomite with SVMClassifier = 1.0
roc auc score of actual prediction on test set of dolomite with SVMClassifier = 1.0

 cross-validation score on training set for sandstone with SVMClassifier = [ 1.  1.  1.  1.]
f1 scores of cross validation prediction training set for sandstone with SVMClassifier = 1.0
roc auc score on cross validation prediction training set for sandstone with SVMClassifier = 1
f1 score of actual prediction on test set of sandstone with SVMClassifier = 0.996226826208
roc auc score of actual prediction on test set of sandstone with SVMClassifier = 0.99799599198
```

In [251]:
```python
from sklearn.neighbors import KNeighborsClassifier

knn_clf_GS = KNeighborsClassifier(n_neighbors = 5, weights = 'distance')

for y_train_all, y_test_all, y_strings_all in zip(y_trains_classes,
                                                  y_test_classes, y_classes_names):
    knn_clf_GS.fit(X_train, y_train_all)
    y_cv_knn = cross_val_score(knn_clf_GS, X_train, y_train_all, cv = 4)
    print("\n","cross-validation score on training set for", y_strings_all, "with KN

    y_cvp_knn = cross_val_predict(knn_clf_GS, X_train, y_train_all, cv=4)
    f1_cvp_train_knn = f1_score(y_train_all, y_cvp_knn, average = 'weighted', labels
    print("f1 scores of cross validation prediction training set for", y_strings_all
          "with KNeighbor Classifier =", f1_cvp_train_knn)
    roc_auc_score_knn_train = roc_auc_score(y_train_all, y_cvp_knn, average = 'weigh
    print("roc auc score on cross validation prediction training set for",
          y_strings_all,"with KNeighbors Classifier =", roc_auc_score_knn_train)


    y_cvp_knn_test = knn_clf_GS.predict(X_test_prepared)
    f1_knn_test_all = f1_score(y_test_all, y_cvp_knn_test, average = 'weighted', labe
    print("f1 score of actual prediction on test set of", y_strings_all,
          "with KNeighbors Classifier =", f1_knn_test_all)
    roc_auc_score_knn_test_all = roc_auc_score(y_test_all, y_cvp_knn_test, average =
    print("roc auc score of actual prediction on test set of",y_strings_all, "with K
          roc_auc_score_knn_test_all)
```

```
 cross-validation score on training set for shaly limestone with KNeighbor Classifier = [ 0.91
f1 scores of cross validation prediction training set for shaly limestone with KNeighbor Class
roc auc score on cross validation prediction training set for shaly limestone with KNeighbors
f1 score of actual prediction on test set of shaly limestone with KNeighbors Classifier = 0.91
roc auc score of actual prediction on test set of shaly limestone with KNeighbors Classifier =

 cross-validation score on training set for limestone with KNeighbor Classifier = [ 0.90517241
```

```
f1 scores of cross validation prediction training set for limestone with KNeighbor Classifier =
roc auc score on cross validation prediction training set for limestone with KNeighbors Classi
f1 score of actual prediction on test set of limestone with KNeighbors Classifier = 0.90873786
roc auc score of actual prediction on test set of limestone with KNeighbors Classifier = 0.908

 cross-validation score on training set for shale with KNeighbor Classifier = [ 0.92816092  0.
f1 scores of cross validation prediction training set for shale with KNeighbor Classifier = 0.
roc auc score on cross validation prediction training set for shale with KNeighbors Classifier
f1 score of actual prediction on test set of shale with KNeighbors Classifier = 0.913566048202
roc auc score of actual prediction on test set of shale with KNeighbors Classifier = 0.7613162

 cross-validation score on training set for sandy lime with KNeighbor Classifier = [ 1.
f1 scores of cross validation prediction training set for sandy lime with KNeighbor Classifier
roc auc score on cross validation prediction training set for sandy lime with KNeighbors Class
f1 score of actual prediction on test set of sandy lime with KNeighbors Classifier = 0.9980081
roc auc score of actual prediction on test set of sandy lime with KNeighbors Classifier = 0.95

 cross-validation score on training set for shaly sandstone with KNeighbor Classifier = [ 0.99
f1 scores of cross validation prediction training set for shaly sandstone with KNeighbor Class
roc auc score on cross validation prediction training set for shaly sandstone with KNeighbors
f1 score of actual prediction on test set of shaly sandstone with KNeighbors Classifier = 0.99
roc auc score of actual prediction on test set of shaly sandstone with KNeighbors Classifier =

 cross-validation score on training set for dolomite with KNeighbor Classifier = [ 1.
f1 scores of cross validation prediction training set for dolomite with KNeighbor Classifier =
roc auc score on cross validation prediction training set for dolomite with KNeighbors Classif
f1 score of actual prediction on test set of dolomite with KNeighbors Classifier = 1.0
roc auc score of actual prediction on test set of dolomite with KNeighbors Classifier = 1.0

 cross-validation score on training set for sandstone with KNeighbor Classifier = [ 0.99712644
f1 scores of cross validation prediction training set for sandstone with KNeighbor Classifier =
roc auc score on cross validation prediction training set for sandstone with KNeighbors Classi
f1 score of actual prediction on test set of sandstone with KNeighbors Classifier = 0.99808669
roc auc score of actual prediction on test set of sandstone with KNeighbors Classifier = 0.998
```

In [252]:
```python
from sklearn.ensemble import RandomForestClassifier

for_clf_GS = RandomForestClassifier(n_estimators = 500, random_state =42)

for y_train_all, y_test_all, y_strings_all in zip(y_trains_classes,
                                                    y_test_classes, y_classes_names
    for_clf_GS.fit(X_train, y_train_all)
    y_cv_randfor = cross_val_score(for_clf_GS, X_train, y_train_all, cv = 4)
    print("\n","cross-validation score on training set for", y_strings_all, "with Ra

    y_cvp_randfor = cross_val_predict(for_clf_GS, X_train, y_train_all, cv=4)
    f1_cvp_train = f1_score(y_train_all, y_cvp_randfor, average = 'weighted', labels
```

```python
            print("f1 scores of cross validation prediction training set for", y_strings_all
            roc_auc_score_randfor_train = roc_auc_score(y_train_all, y_cvp_randfor, average =
            print("roc auc score on cross validation prediction training set for",
                  y_strings_all,"with RandomForestClassifier =", roc_auc_score_randfor_train)


            y_cvp_randfor_test = for_clf_GS.predict(X_test_prepared)
            f1_for_test_all = f1_score(y_test_all, y_cvp_randfor_test, average = 'weighted',
            print("f1 score of actual prediction on test set of", y_strings_all,
                  "with RandomForestClassifier =", f1_for_test_all)
            roc_auc_score_randfor_test_all = roc_auc_score(y_test_all, y_cvp_randfor_test, av
            print("roc auc score of actual prediction on test set of",y_strings_all, "with Ra
                  roc_auc_score_randfor_test_all)
```

```
 cross-validation score on training set for shaly limestone with RandomForestClassifier = [ 0.9
f1 scores of cross validation prediction training set for shaly limestone = 0.944084567529
roc auc score on cross validation prediction training set for shaly limestone with RandomFores
f1 score of actual prediction on test set of shaly limestone with RandomForestClassifier = 0.95
roc auc score of actual prediction on test set of shaly limestone with RandomForestClassifier =

 cross-validation score on training set for limestone with RandomForestClassifier = [ 0.933908
f1 scores of cross validation prediction training set for limestone = 0.928053383173
roc auc score on cross validation prediction training set for limestone with RandomForestClass
f1 score of actual prediction on test set of limestone with RandomForestClassifier = 0.9611650
roc auc score of actual prediction on test set of limestone with RandomForestClassifier = 0.96

 cross-validation score on training set for shale with RandomForestClassifier = [ 0.93965517  0
f1 scores of cross validation prediction training set for shale = 0.910755728115
roc auc score on cross validation prediction training set for shale with RandomForestClassifie
f1 score of actual prediction on test set of shale with RandomForestClassifier = 0.9310069641
roc auc score of actual prediction on test set of shale with RandomForestClassifier = 0.800401

 cross-validation score on training set for sandy lime with RandomForestClassifier = [ 0.991379
f1 scores of cross validation prediction training set for sandy lime = 0.994610770556
roc auc score on cross validation prediction training set for sandy lime with RandomForestClas
f1 score of actual prediction on test set of sandy lime with RandomForestClassifier = 0.998008
roc auc score of actual prediction on test set of sandy lime with RandomForestClassifier = 0.95

 cross-validation score on training set for shaly sandstone with RandomForestClassifier = [ 0.9
f1 scores of cross validation prediction training set for shaly sandstone = 0.99855907781
roc auc score on cross validation prediction training set for shaly sandstone with RandomFores
f1 score of actual prediction on test set of shaly sandstone with RandomForestClassifier = 0.99
roc auc score of actual prediction on test set of shaly sandstone with RandomForestClassifier =

 cross-validation score on training set for dolomite with RandomForestClassifier = [ 1.  1.  1
f1 scores of cross validation prediction training set for dolomite = 1.0
roc auc score on cross validation prediction training set for dolomite with RandomForestClassi
```

```
f1 score of actual prediction on test set of dolomite with RandomForestClassifier = 0.99807053
roc auc score of actual prediction on test set of dolomite with RandomForestClassifier = 0.998

 cross-validation score on training set for sandstone with RandomForestClassifier = [ 0.997126
f1 scores of cross validation prediction training set for sandstone = 0.998544555623
roc auc score on cross validation prediction training set for sandstone with RandomForestClass
f1 score of actual prediction on test set of sandstone with RandomForestClassifier = 0.99808669
roc auc score of actual prediction on test set of sandstone with RandomForestClassifier = 0.998
```

```
In [253]: from sklearn.ensemble import VotingClassifier

          voting_clf_all = VotingClassifier(estimators = [('knnGS', knn_clf_GS),
                                                           ('svmGS', svm_clf_GS),
                                                          ('rnfGS', for_clf_GS)], voting = 'sof

          for y_train_all, y_test_all, y_strings_all in zip(y_trains_classes,
                                                                 y_test_classes, y_classes_names

              voting_clf_all.fit(X_train, y_train_all)
              y_cv_voting = cross_val_score(voting_clf_all, X_train, y_train_all, cv = 4)
              print("\n","cross-validation score on training set for", y_strings_all,
                    "with Voting b/w three classifier =", y_cv_voting )

              y_pred_voting  = cross_val_predict(voting_clf_all, X_train, y_train_all, cv=4, n
              f1_cvp_train_voting = f1_score(y_train_all, y_pred_voting , average="weighted", 
              print("f1 scores of cross validation prediction training set for", y_strings_all
                    "with Voting b/w three classifier =", f1_cvp_train_voting)
              roc_auc_score_voting_train = roc_auc_score(y_train_all, y_pred_voting, average =
              print("roc auc score on cross validation prediction training set for"
                    , y_strings_all,"with Voting b/w three classifier =", roc_auc_score_voting

              y_test_voting_pred_all = voting_clf_all.predict(X_test_prepared)
              f1_test_voting_all = f1_score(y_test_all, y_test_voting_pred_all,
                                            average= 'weighted', labels = np.unique(y_test_vot
              print("f1 score of actual prediction on test set of", y_strings_all,
                    "with Voting b/w three classifier =", f1_test_voting_all)
              roc_auc_score_voting_test_all = roc_auc_score(y_test_all, y_test_voting_pred_all
              print("roc auc score of actual prediction on test set of",y_strings_all,
                    "with Voting b/w three classifier =", roc_auc_score_voting_test_all)
```

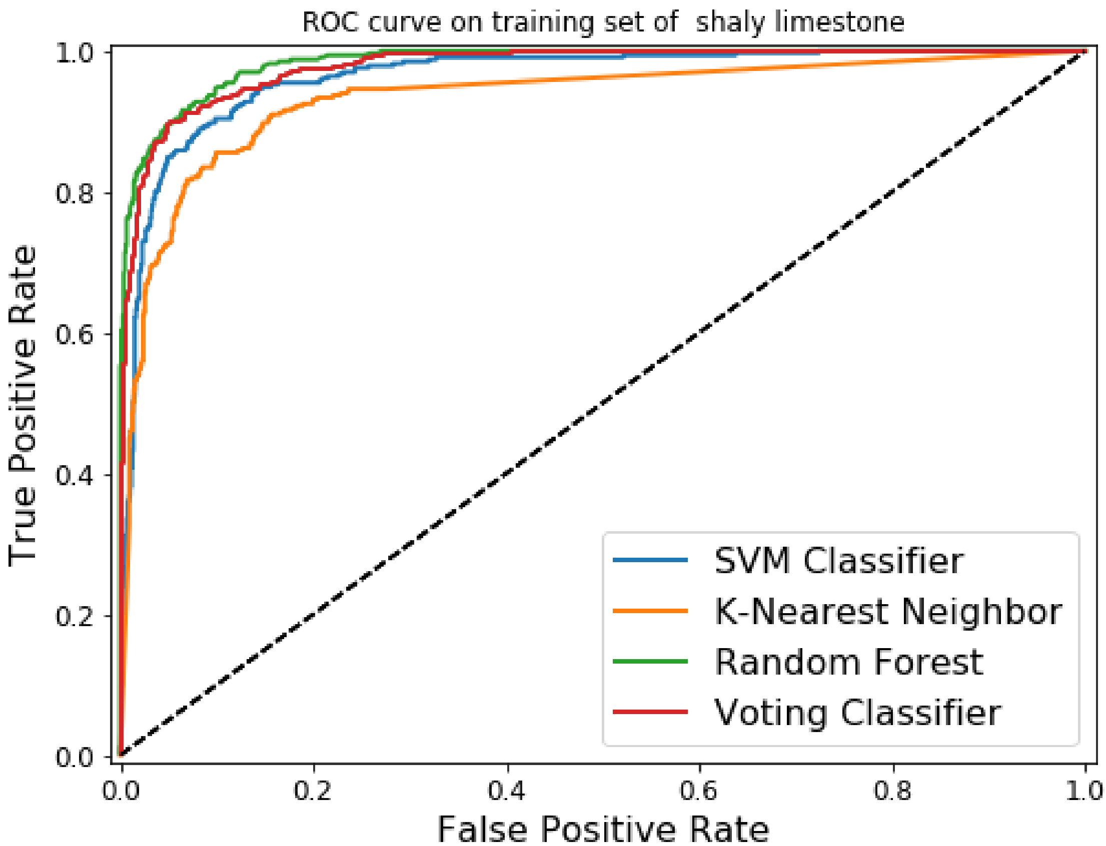
```
 cross-validation score on training set for shaly limestone with Voting b/w three classifier =
f1 scores of cross validation prediction training set for shaly limestone with Voting b/w thre
roc auc score on cross validation prediction training set for shaly limestone with Voting b/w 
f1 score of actual prediction on test set of shaly limestone with Voting b/w three classifier 
roc auc score of actual prediction on test set of shaly limestone with Voting b/w three classi
```

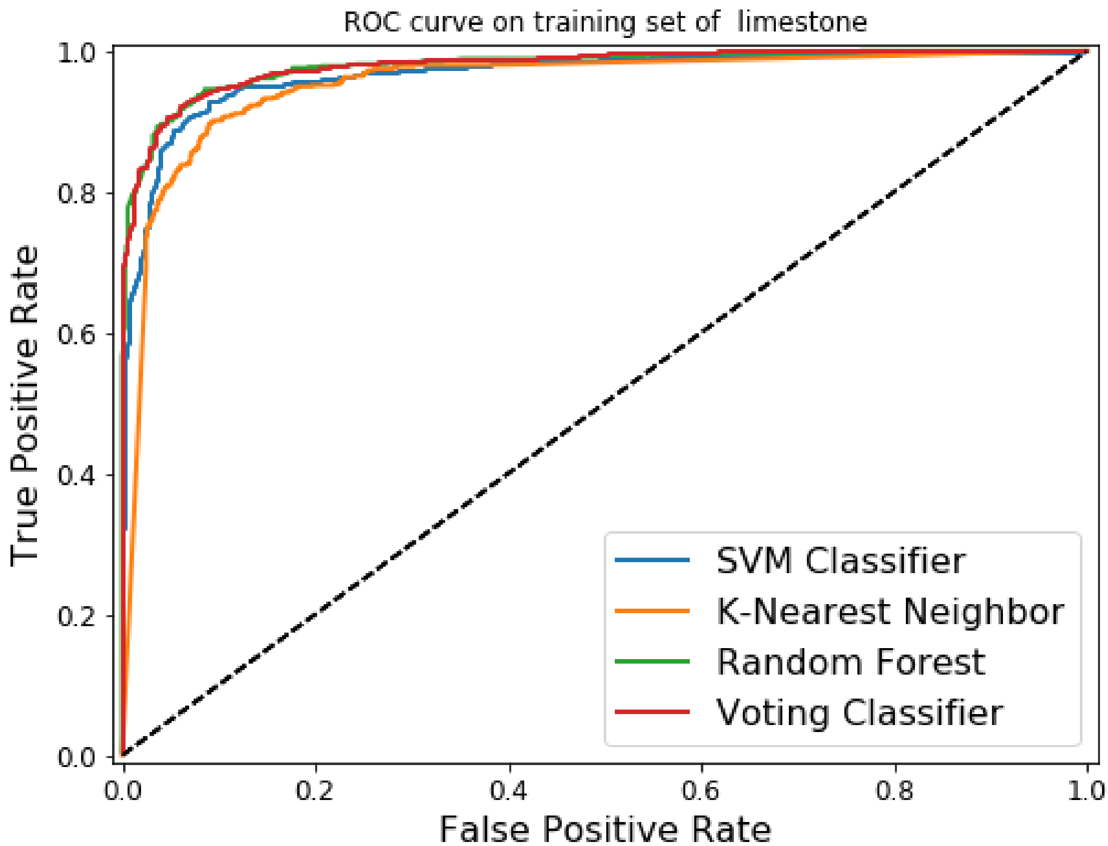
```
 cross-validation score on training set for limestone with Voting b/w three classifier = [ 0.93
f1 scores of cross validation prediction training set for limestone with Voting b/w three class
roc auc score on cross validation prediction training set for limestone with Voting b/w three c
f1 score of actual prediction on test set of limestone with Voting b/w three classifier = 0.953
roc auc score of actual prediction on test set of limestone with Voting b/w three classifier =

 cross-validation score on training set for shale with Voting b/w three classifier = [ 0.933908
f1 scores of cross validation prediction training set for shale with Voting b/w three classifie
roc auc score on cross validation prediction training set for shale with Voting b/w three class
f1 score of actual prediction on test set of shale with Voting b/w three classifier = 0.9416977
roc auc score of actual prediction on test set of shale with Voting b/w three classifier = 0.82

 cross-validation score on training set for sandy lime with Voting b/w three classifier = [ 1.
f1 scores of cross validation prediction training set for sandy lime with Voting b/w three clas
roc auc score on cross validation prediction training set for sandy lime with Voting b/w three
f1 score of actual prediction on test set of sandy lime with Voting b/w three classifier = 0.99
roc auc score of actual prediction on test set of sandy lime with Voting b/w three classifier =

 cross-validation score on training set for shaly sandstone with Voting b/w three classifier =
f1 scores of cross validation prediction training set for shaly sandstone with Voting b/w three
roc auc score on cross validation prediction training set for shaly sandstone with Voting b/w t
f1 score of actual prediction on test set of shaly sandstone with Voting b/w three classifier =
roc auc score of actual prediction on test set of shaly sandstone with Voting b/w three classif

 cross-validation score on training set for dolomite with Voting b/w three classifier = [ 1.   1
f1 scores of cross validation prediction training set for dolomite with Voting b/w three classi
roc auc score on cross validation prediction training set for dolomite with Voting b/w three cl
f1 score of actual prediction on test set of dolomite with Voting b/w three classifier = 1.0
roc auc score of actual prediction on test set of dolomite with Voting b/w three classifier = 1

 cross-validation score on training set for sandstone with Voting b/w three classifier = [ 1.
f1 scores of cross validation prediction training set for sandstone with Voting b/w three class
roc auc score on cross validation prediction training set for sandstone with Voting b/w three c
f1 score of actual prediction on test set of sandstone with Voting b/w three classifier = 0.998
roc auc score of actual prediction on test set of sandstone with Voting b/w three classifier =
```

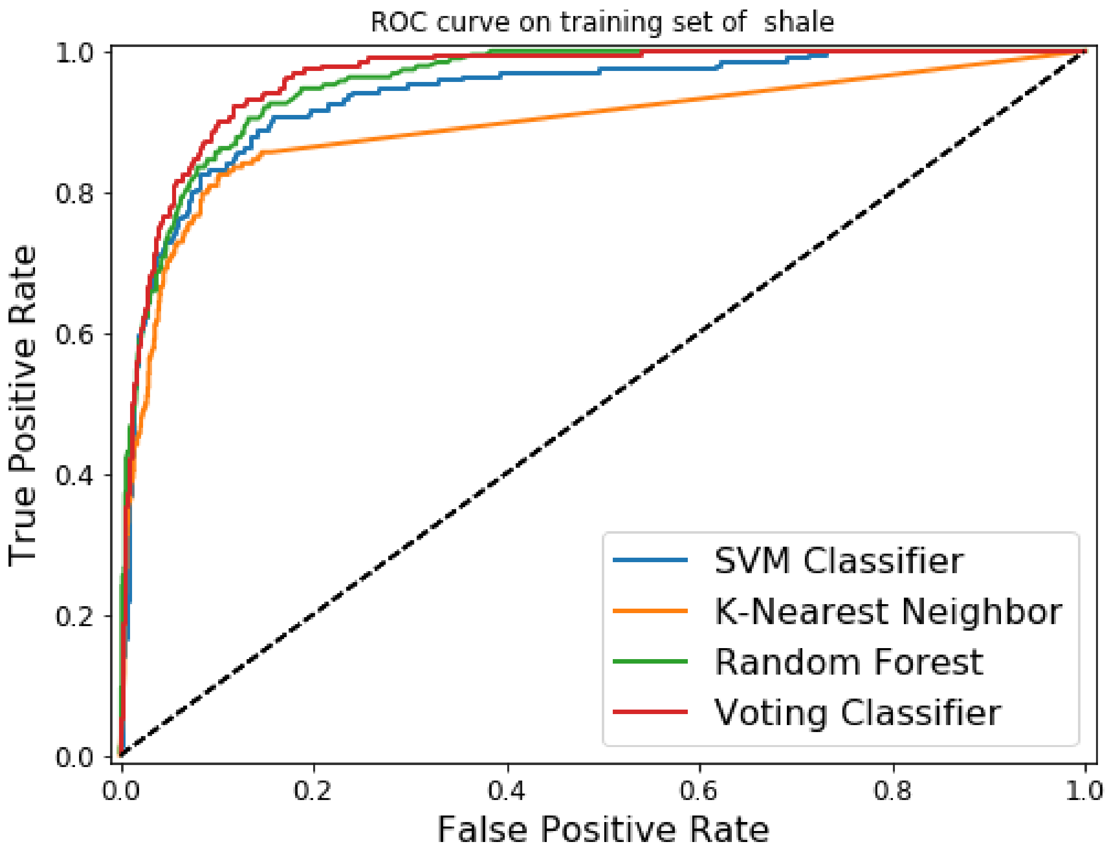
ROC-performance on training sets of all classes:

```
In [254]: for y_train_all, y_test_all, y_strings_all in zip(y_trains_classes,
                                                        y_test_classes, y_classes_names)
              #ROC-curves of SVM
              y_svm_proba_train = cross_val_predict(svm_clf_GS, X_train, y_train_all, cv=4,
                                   method="predict_proba")
              y_svm_scores_train = y_svm_proba_train[:, 1]

              fpr_svm_train, tpr_svm_train, thresholds_svm_train = roc_curve(
                                                                    y_train_all,
```

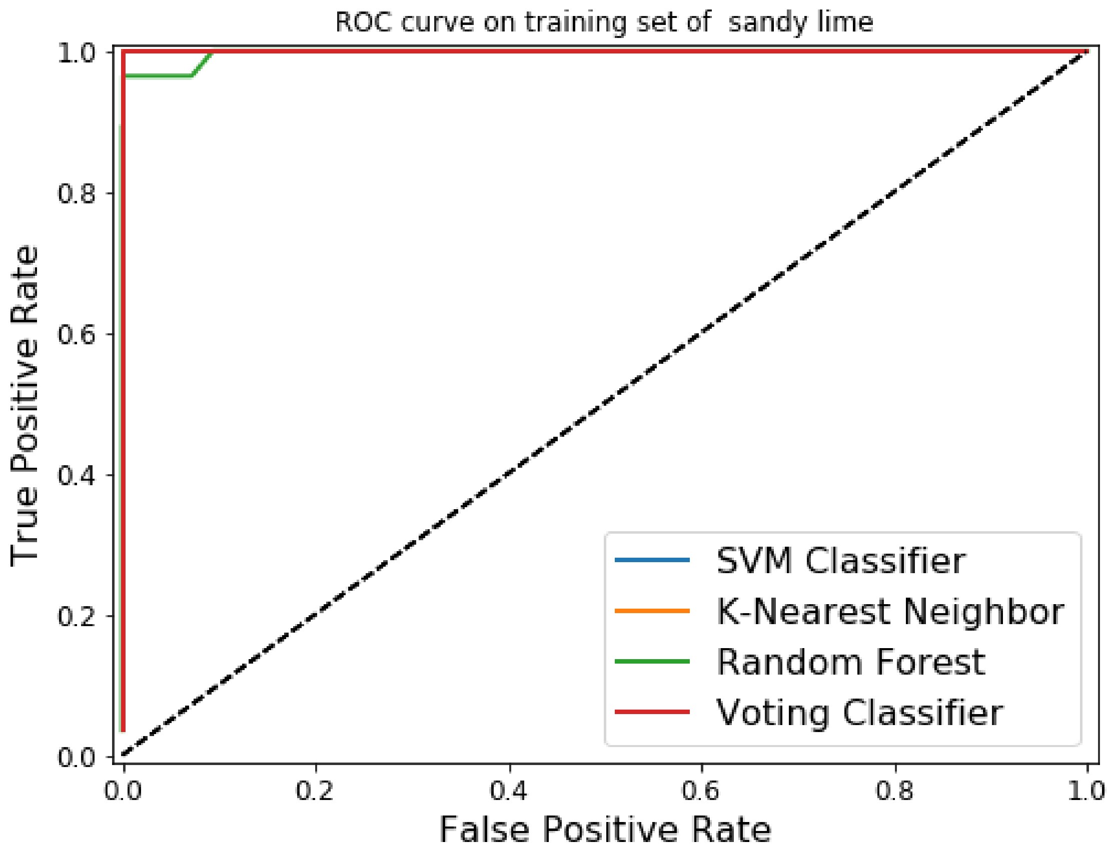
```
                                                               y_svm_scores_tra


#ROC-curves for KNN
y_knn_proba_train = cross_val_predict(knn_clf_GS, X_train, y_train_all, cv=4,
                         method="predict_proba")
y_knn_scores_train = y_knn_proba_train[:, 1]

fpr_knn_train, tpr_knn_train, thresholds_knn_train = roc_curve(
                                                               y_train_all,
                                                               y_knn_scores_tra


#ROC-curves for RandomForestClf
y_randfor_proba_train = cross_val_predict(for_clf_GS, X_train, y_train_all, cv=4
                         method="predict_proba")
y_randfor_scores_train = y_randfor_proba_train[:, 1]

fpr_randfor_train, tpr_randfor_train, thresholds_randfor_trainr = roc_curve(
                                                                           y_tra
                                                                           y_ran


#ROC-curves for voting

y_voting_proba_train_all = cross_val_predict(voting_clf_all, X_train, y_train_al
                         method="predict_proba")
y_voting_scores_train_all = y_voting_proba_train_all[:, 1]
fpr_voting_train_all, tpr_voting_train_all, thresholds_voting_train_all = roc_cu
                                                                            y_vo

#Plotting ROC-curves for each class
plt.figure(figsize=(8, 6))
plot_roc_curve(fpr_svm_train, tpr_svm_train, "SVM Classifier")
plot_roc_curve(fpr_knn_train, tpr_knn_train, "K-Nearest Neighbor")
plot_roc_curve(fpr_randfor_train, tpr_randfor_train, "Random Forest")
plot_roc_curve(fpr_voting_train_all, tpr_voting_train_all, "Voting Classifier")
plt.legend(loc="lower right", fontsize=16)
plt.title('ROC curve on training set of  %s'%(y_strings_all))
plt.axis([-0.01, 1.01, -0.01, 1.01])
plt.show()
```

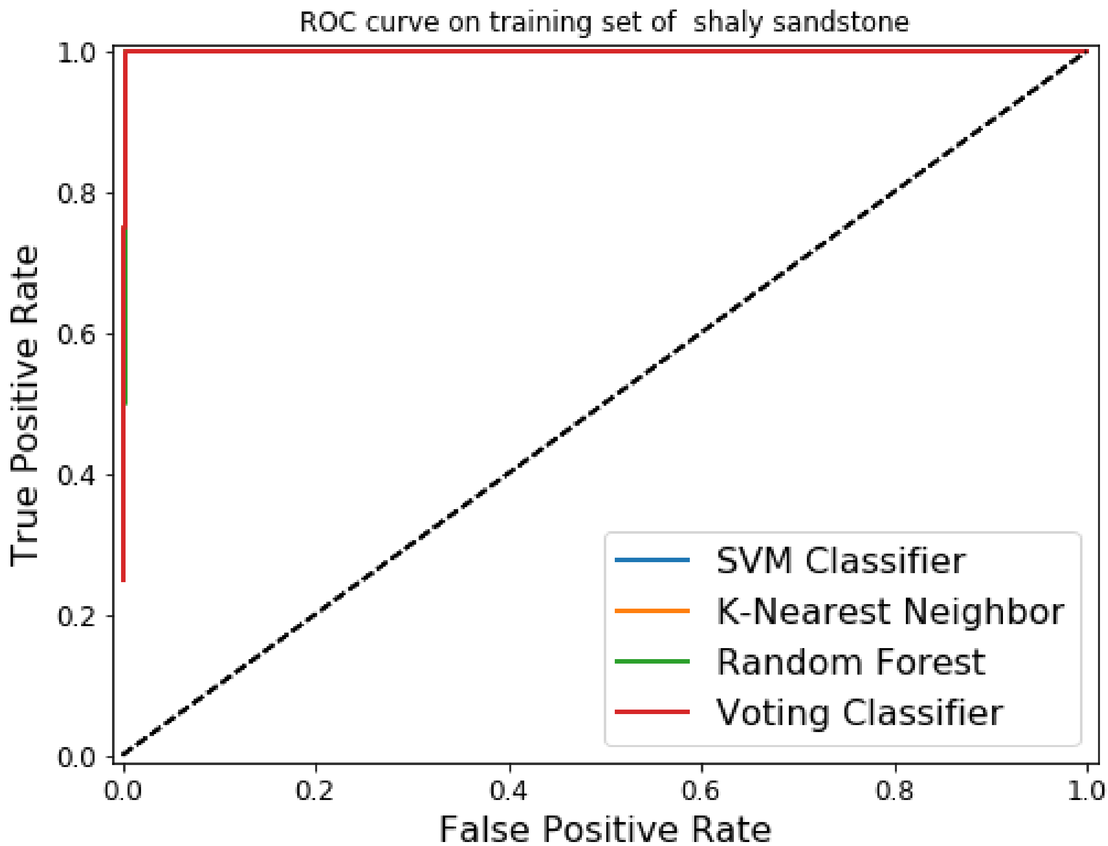

ROC curve on training set of shaly limestone
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier

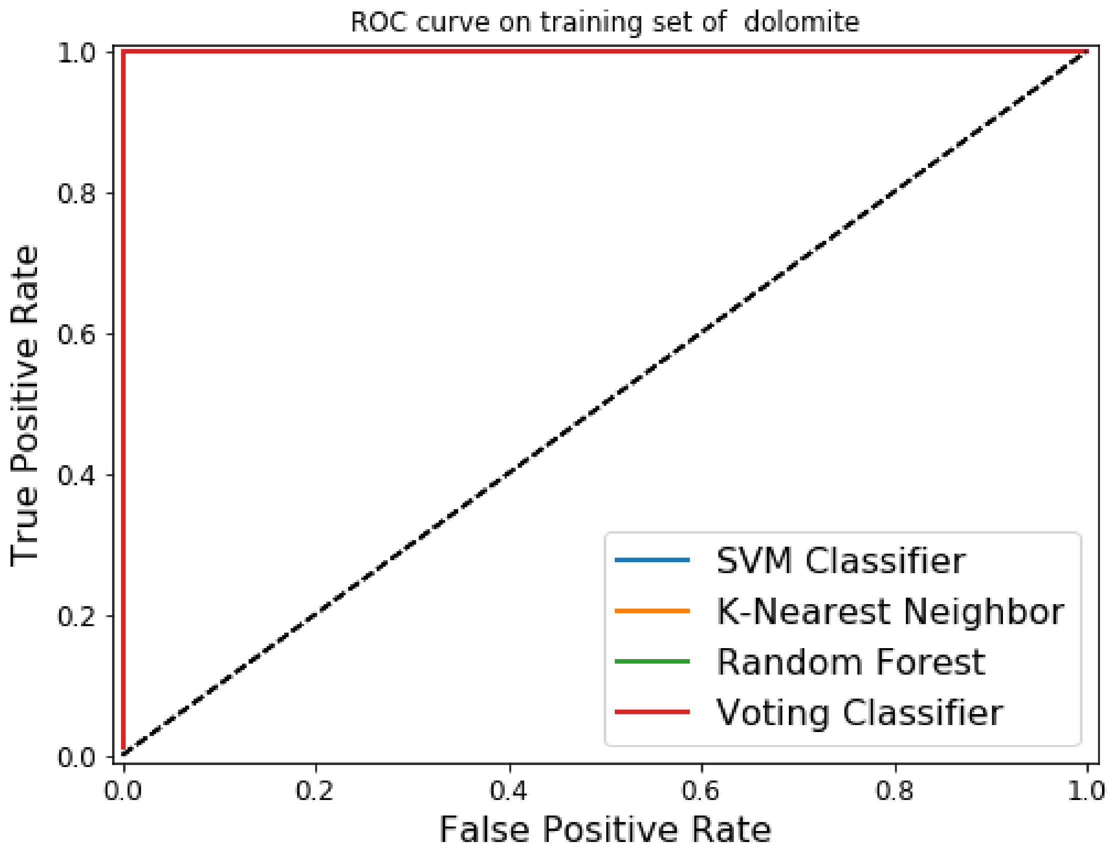

ROC curve on training set of limestone
True Positive Rate
False Positive Rate
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier
0.0
0.2
0.4
0.6
0.8
1.0

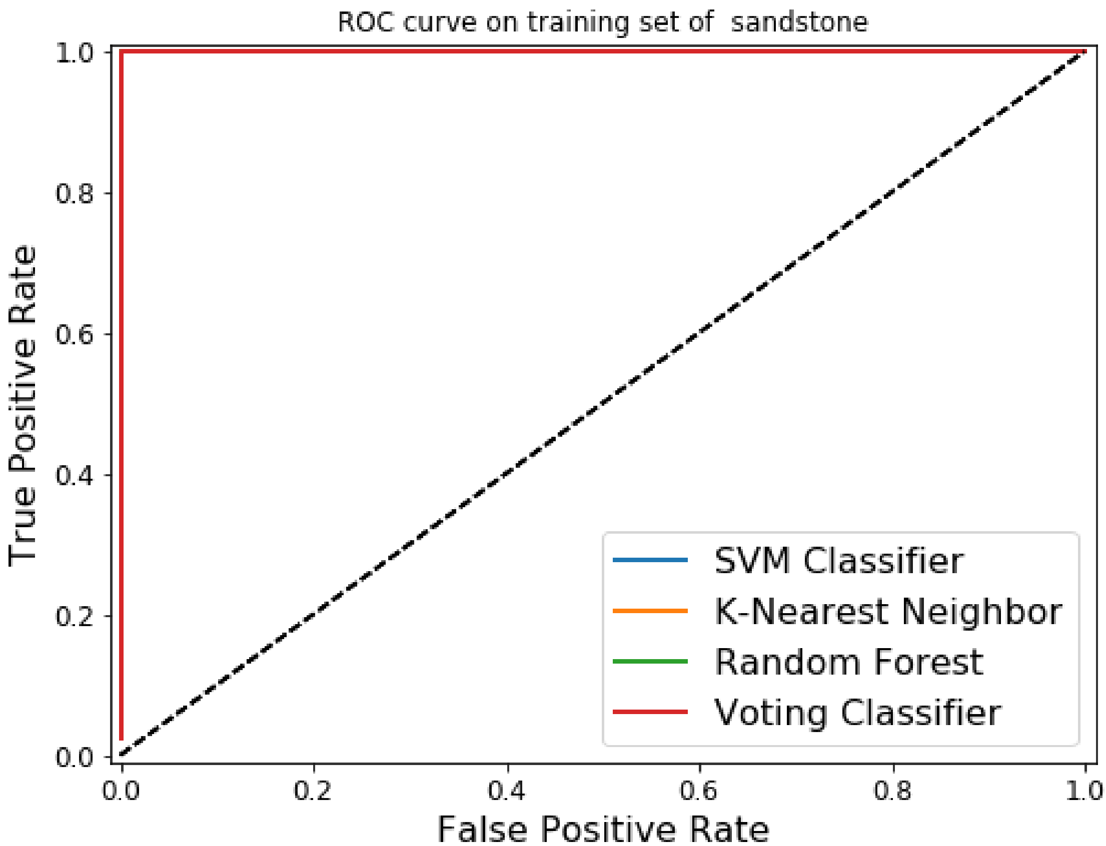


ROC curve on training set of shale
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier


ROC curve on training set of sandy lime
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier


ROC curve on training set of shaly sandstone
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier

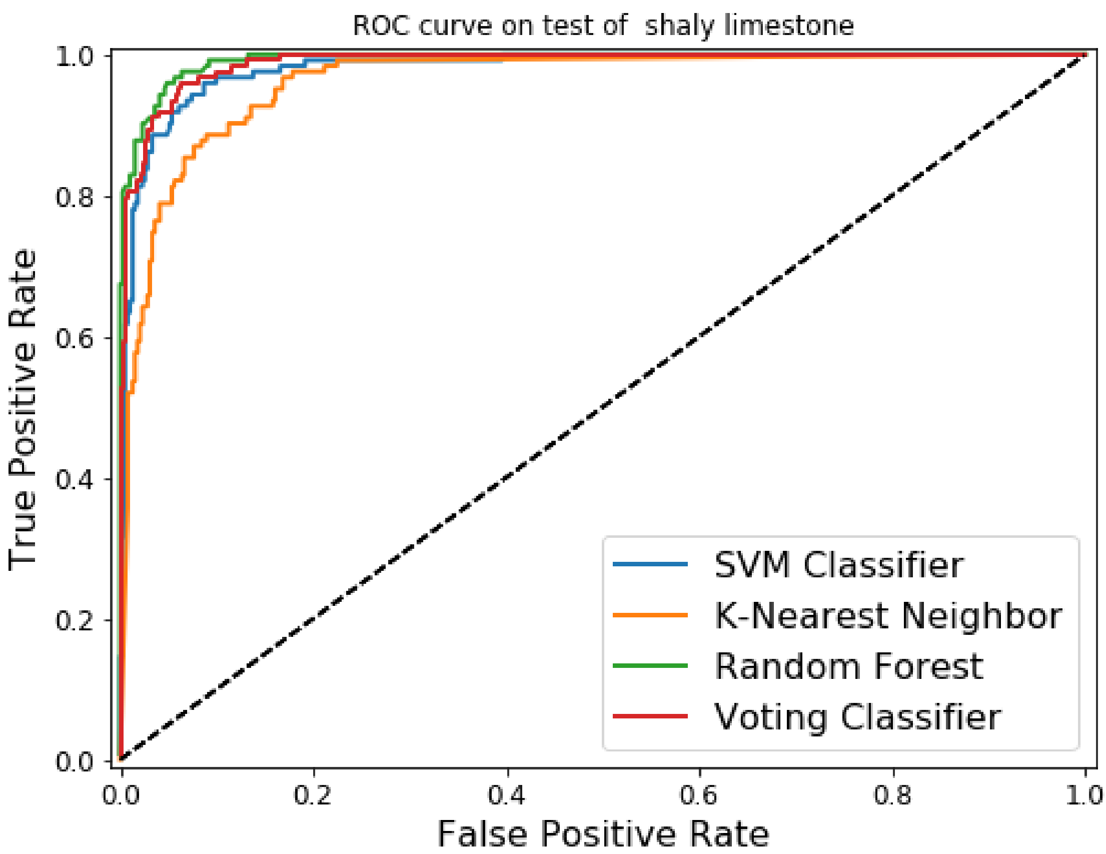

ROC curve on training set of dolomite
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier

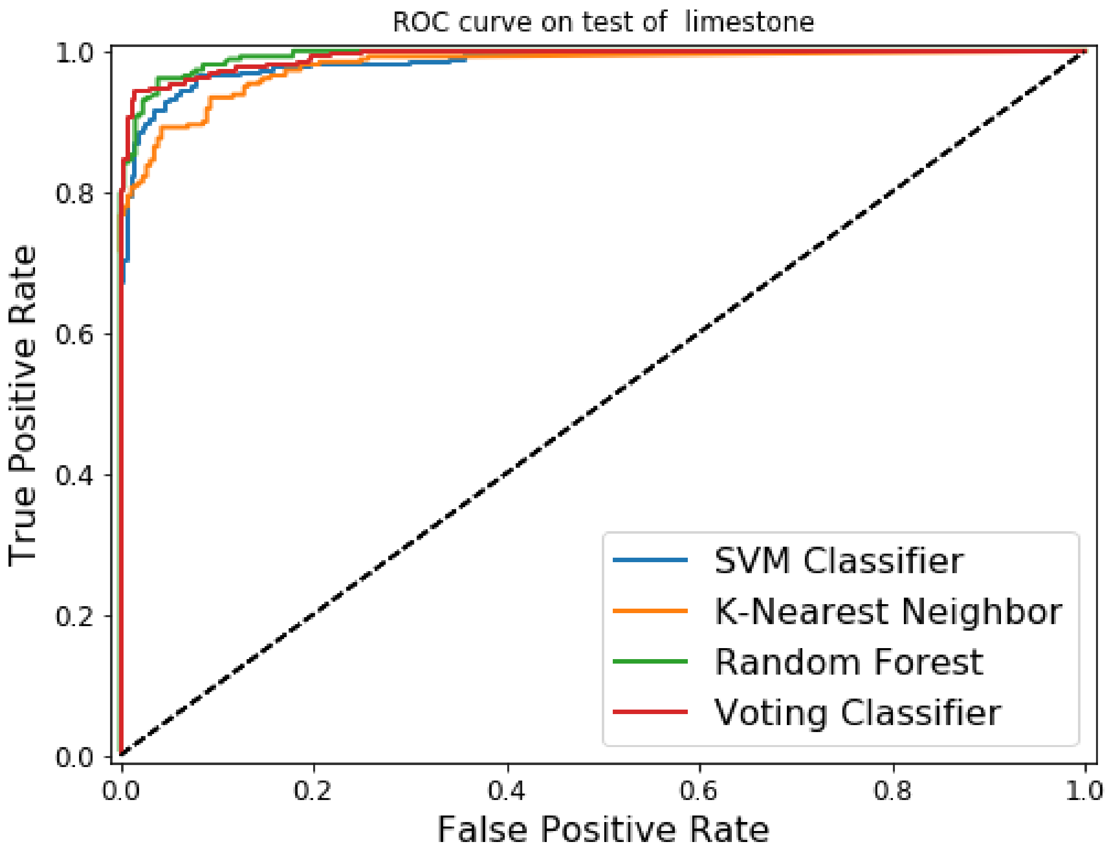


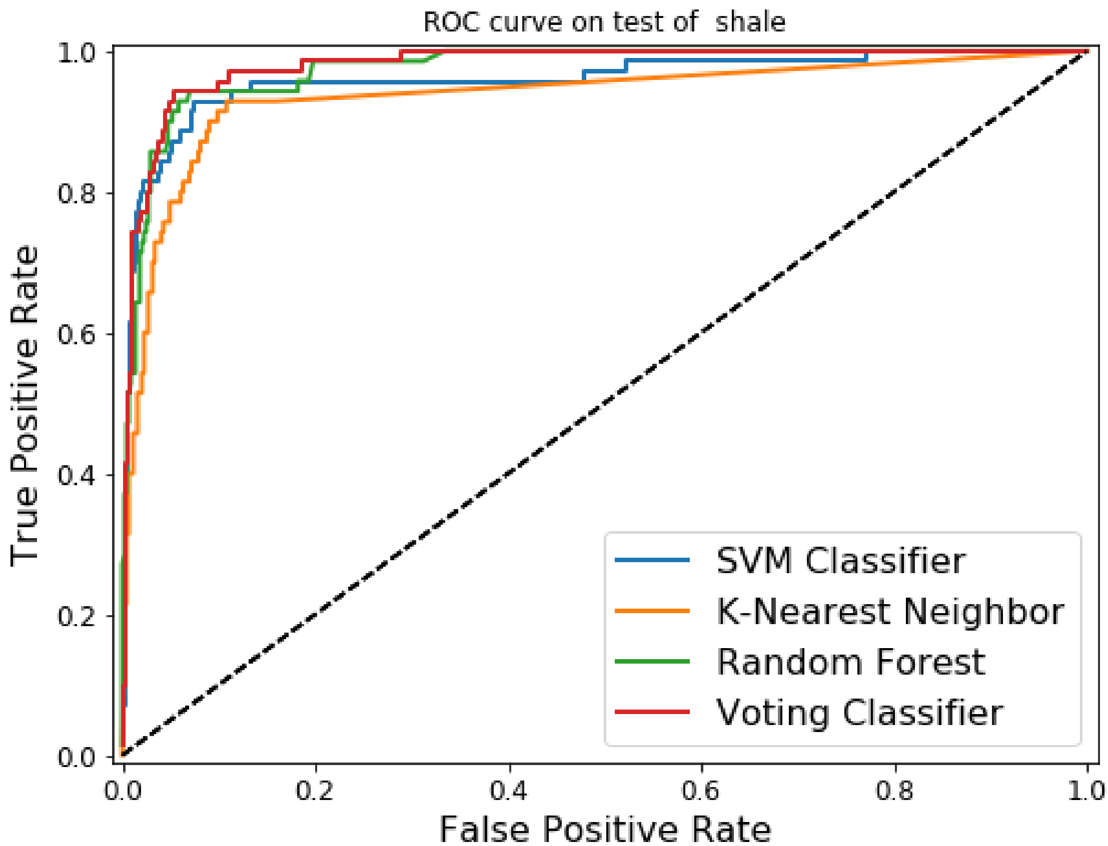
ROC-performance on all classes of test sets:

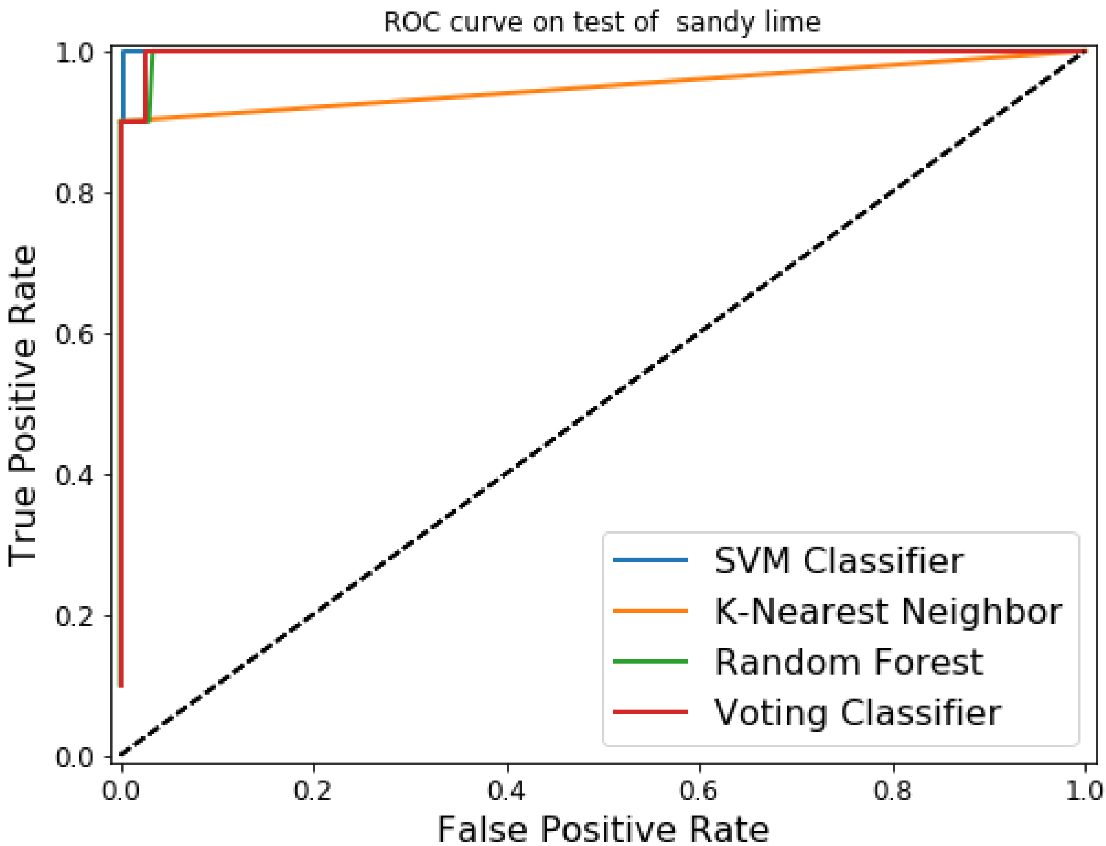
```
In [255]: for y_train_all, y_test_all, y_strings_all in zip(y_trains_classes,
                                                            y_test_classes, y_classes_names):

              svm_clf_GS.fit(X_train, y_train_all)
              y_svm_proba_test = svm_clf_GS.predict_proba(X_test_prepared)
              y_svm_scores_test = y_svm_proba_test[:, 1]

              fpr_svm_test, tpr_svm_test, thresholds_svm_test = roc_curve(y_test_all,
                                                                           y_svm_scores_test)

              knn_clf_GS.fit(X_train, y_train_all)
              y_knn_proba_test = knn_clf_GS.predict_proba(X_test_prepared)
              y_knn_scores_test = y_knn_proba_test[:, 1]

              fpr_knn_test, tpr_knn_test, thresholds_knn_test = roc_curve(y_test_all,
                                                                           y_knn_scores_test)

              for_clf_GS.fit(X_train, y_train_all)
              y_randfor_proba_test = for_clf_GS.predict_proba(X_test_prepared)
```

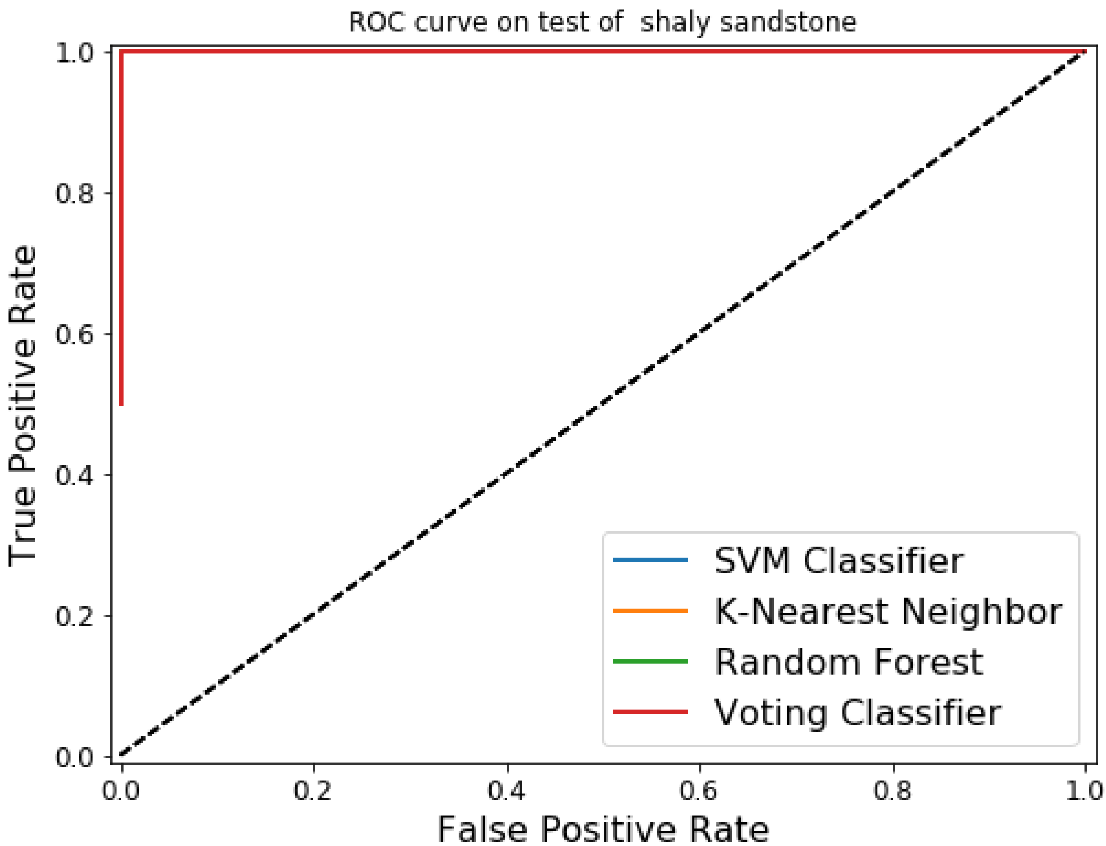
```
y_randfor_scores_test = y_randfor_proba_test[:, 1]

fpr_randfor_test, tpr_randfor_test, thresholds_randfor_test = roc_curve(y_test_al
                                                                        y_randfo


voting_clf_all.fit(X_train, y_train_all)
y_probas_voting_all = voting_clf_all.predict_proba(X_test_prepared)
y_scores_voting_all = y_probas_voting_all[:, 1] # score = proba of positive clas
fpr_voting_test_all, tpr_voting_test_all, thresholds_voting_test_all = roc_curve
                                                                         y_score


plt.figure(figsize=(8, 6))
plot_roc_curve(fpr_svm_test, tpr_svm_test, "SVM Classifier")
plot_roc_curve(fpr_knn_test, tpr_knn_test, "K-Nearest Neighbor")
plot_roc_curve(fpr_randfor_test, tpr_randfor_test, "Random Forest")
plot_roc_curve(fpr_voting_test_all, tpr_voting_test_all, "Voting Classifier")
plt.legend(loc="lower right", fontsize=16)
plt.title('ROC curve on test of  %s'%(y_strings_all))
plt.axis([-0.01, 1.01, -0.01, 1.01])
plt.show()
```

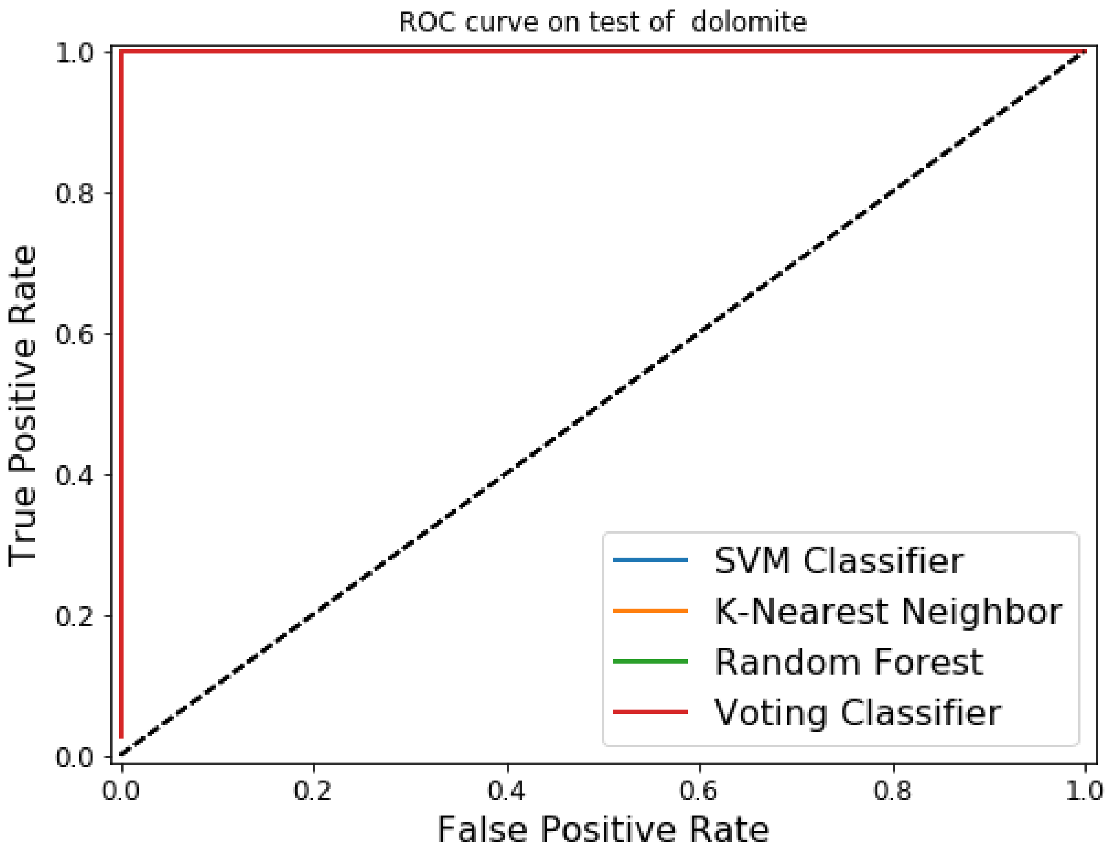


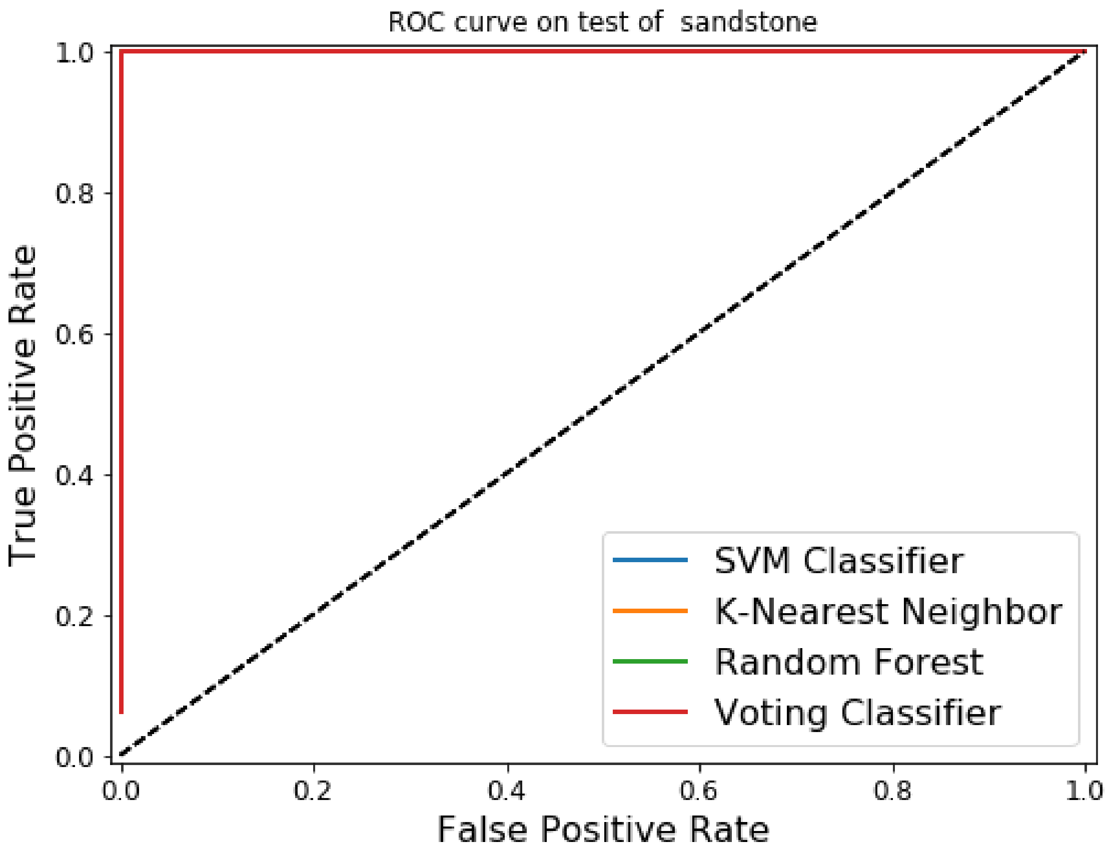

ROC curve on test of limestone
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier


ROC curve on test of shale
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier


ROC curve on test of sandy lime
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier


ROC curve on test of shaly sandstone
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier


ROC curve on test of dolomite
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
SVM Classifier
K-Nearest Neighbor
Random Forest
Voting Classifier



Manual Voting on Shaly Limestone:

```
In [256]: # parameters for random forest
          rfclf_params_sl = {
              'n_estimators': 500,
              'bootstrap': True,
              'class_weight':None,
              'criterion':'gini',
              'max_depth':None,
              'max_features':'auto',
              'random_state' : 41

              # ... fill in the rest you want here
          }

          # Fill in svm params here
          svm_params_sl = {
              'C': 100,
              'probability':True,
              'random_state' : 42
          }
```

```
# KNeighbors params go here
kneighbors_params_sl= {
    'n_neighbors': 5,
    'weights':'distance'
}

params_sl = [rfclf_params_sl, svm_params_sl, kneighbors_params_sl]
classifiers_sl = [RandomForestClassifier, SVC, KNeighborsClassifier]
```

In [257]:
```
def ensemble_test(classifiers, params, X_train, y_train, X_test):
    best_preds_test = np.zeros((len(X_test), 2))
    classes_test = np.unique(y_train)

    for i in range(len(classifiers)):
        # Construct the classifier by unpacking params
        # store classifier instance
        clf_test = classifiers[i](**params[i])
        # Fit the classifier as usual and call predict_proba
        clf_test.fit(X_train, y_train)
        y_preds_test = clf_test.predict_proba(X_test)
        # Take maximum probability for each class on each classifier
        # This is done for every instance in X_test
        best_preds_test = np.maximum(best_preds_test, y_preds_test)

    # map the maximum probability for each instance back to its corresponding class
    preds_test = np.array([classes_test[np.argmax(pred)] for pred in best_preds_test]
    return preds_test
```

In [258]:
```
from sklearn.metrics import accuracy_score, f1_score
y_preds_test_sl = ensemble_test(classifiers_sl, params_sl, X_train, y_train_sl, X_tes
print('Accuracy score = ',accuracy_score(y_test_sl, y_preds_test_sl),'\n',
     'f1_score = ', f1_score(y_test_sl, y_preds_test_sl, average = 'weighted'),'\n',
     'roc_auc_score = ', roc_auc_score(y_test_sl, y_preds_test_sl, average = 'weighte
```

```
Accuracy score =  0.949514563107
 f1_score =  0.949653574035
 roc_auc_score =  0.933362369338
```

Generalizing manual voting for all classes:

In [259]:
```
# parameters for random forest
rfclf_params = {
    'n_estimators': 500,
    'bootstrap': True,
    'class_weight':None,
    'criterion':'gini',
    'max_depth':None,
```

```
            'max_features':'auto',
            'warm_start': True,
            'random_state': 41
            # ... fill in the rest you want here
        }

        # Fill in svm params here
        svm_params = {
            'C': 100,
            'probability':True,
            'random_state':42
        }

        # KNeighbors params go here
        kneighbors_params= {
            'n_neighbors': 5,
            'weights':'distance'
        }
```

In [260]:
```
y_test_classes = (y_test_sl, y_test_lim, y_test_shale, y_test_sandlim, y_test_ss, y_t
classifiers = [RandomForestClassifier, SVC, KNeighborsClassifier]
params = [rfclf_params, svm_params, kneighbors_params]
y_trains_classes= (y_train_sl, y_train_lim, y_train_shale, y_train_sandlim,
                   y_train_ss, y_train_dol, y_train_sand)
y_classes_names = ("shaly limestone", "limestone", "shale", "sandy lime",
                   "shaly sandstone", "dolomite", "sandstone")
```

In [261]:
```
#Just get predictions
for y_trains, y_test, y_strings in zip(y_trains_classes, y_test_classes, y_classes_na
    y_preds_test = ensemble_test(classifiers, params, X_train, y_trains, X_test_prepa
    print("\n","Accuracy score for", y_strings, "=", accuracy_score(y_test, y_preds_t
    print("f1_score for", y_strings, "=", f1_score(y_test, y_preds_test,
                                                        average = 'weighted', labels=
    print("roc auc score for", y_strings, "=", roc_auc_score(y_test, y_preds_test,
                                                                  average = 'weighte
```

```
 Accuracy score for shaly limestone = 0.949514563107
f1_score for shaly limestone = 0.949653574035
roc auc score for shaly limestone = 0.933362369338

 Accuracy score for limestone = 0.957281553398
f1_score for limestone = 0.957272532095
roc auc score for limestone = 0.957311555515

 Accuracy score for shale = 0.95145631068
f1_score for shale = 0.948556595316
roc auc score for shale = 0.845505617978
```

```
 Accuracy score for sandy lime = 0.998058252427
f1_score for sandy lime = 0.998008114117
roc auc score for sandy lime = 0.95

 Accuracy score for shaly sandstone = 0.996116504854
f1_score for shaly sandstone = 0.998054474708
roc auc score for shaly sandstone = 0.5

 Accuracy score for dolomite = 1.0
f1_score for dolomite = 1.0
roc auc score for dolomite = 1.0

 Accuracy score for sandstone = 0.996116504854
f1_score for sandstone = 0.996226826208
roc auc score for sandstone = 0.997995991984
```