Quarter 1 Project Report: Chlorophyll-a Concentrations in Water Bodies Predictive Model

Group Members:

Jacob Dipasupil, Petr Kisselev, Nikhil Alladi

Table of Contents

1 Data Information

Our dataset contains data compiled by the U.S. Environmental Protection Agency on certain characteristics of lakes[1]. The dataset has 67 different attributes, some of which include lake name, date of sampling, total phosphorus concentration, area of lake surface, monthly and yearly average precipitation across the watershed, annual average nitrogen from human waste, lake depth, log of chlorophyll-*a* concentration, and more. The meaning of every attribute is contained in the data dictionary found in the provided link to the dataset. Since there are 67 different attributes, and we are classifying a class attribute of our choosing, our dataset has a dimensionality of 66. There are 2,226 instances with 45 missing values for lake name, 41 missing values for both nitrogen concentration and phosphorus concentration, 75 missing values for depth, and 132 missing values for the log of chlorophyll-*a* concentration, our class. Since we are trying to classify chlorophyll-*a* concentration, we will have to remove those 132 instances where the values are missing. The distribution of data is slightly right skewed with a mean of 1.053 and a standard deviation of 0.563 ranging from 0.029 to 2.941. Since the log of chlorophyll-*a* is a numerical variable, we will discretize the data into three bins: low, medium, and high. The class distribution is quite heavily skewed to the right, with 1,084 instances in low (-inf-0.999795], 864 in medium (0.999795-1.970205], and 146 in high (1.970205-inf).

2 Model and Rationale

Our model will use data on lakes to predict if the concentration of chlorophyll-*a* is high, medium, or low in order to give us information about the state of the lake ecosystem. Chlorophyll-*a* concentrations can be used as a measure of the amount of algae growing in a water body and give us information on the trophic condition of a waterbody. High levels of chlorophyll-*a* concentrations and the subsequent algae growth can lead to harmful algal bloom, characterized by excessive algae growth producing toxins in water bodies, and hypoxia, which is when oxygen concentrations are too low for most organisms to survive in. Both of which are detrimental to the organisms living in and drinking from water bodies and can have harmful effects to the surrounding ecosystem. Being able to predict chlorophyll-*a* concentrations before permanent damage is done can help save some of these ecosystems.

3 Preprocessing

The first step of our preprocessing was done in Google Sheets. Many values in our dataset caused errors when trying to open in Weka. In order to allow Weka to open the dataset, all apostrophes in the *LAKENAME* attribute values were replaced with spaces. Additionally, there were 287 cells that contained one of the following values: "#NUM!", "#DIV/0!", "#VALUE!". These obvious error values lead Weka to decide that certain attributes are string when they should be numeric attributes. To fix this we simply converted all data cells with those values into empty cells.

Pushing this data into WEKA, some further steps for preprocessing present themselves. To begin with, 130 instances in our dataset are missing values for our assigned class attribute *logchl_A*. As supervised learning requires labeled class attributes, we removed these instances from our dataset (note that due to the values here being positive decimals, we set the split point to be above the maximum of 2.941 in this attribute so that the filter did not inadvertently remove valid instances as well). Additionally, we renamed the labels for the discretized class to low, medium, and high, instead of the ranges listed earlier. We also removed attributes that can be clearly reasoned to have no relation to the class attribute of any kind, including *LAKENAME, Survey Number, and SITE_ID.*

Looking at this data in a spreadsheet view, we noticed that some attributes had a notably high amount of the value 0 in them. Due to their numerical basis, we took this 0 to be a default value, and analyzed the amount of zeros per attribute. In order to perform this analysis, we created a Python script using the Pandas library to load in the .csv version of our file that we got from the previous step and measure the percent of each attribute that consisted of zero values.
Doing so, we came to this result, seen below.

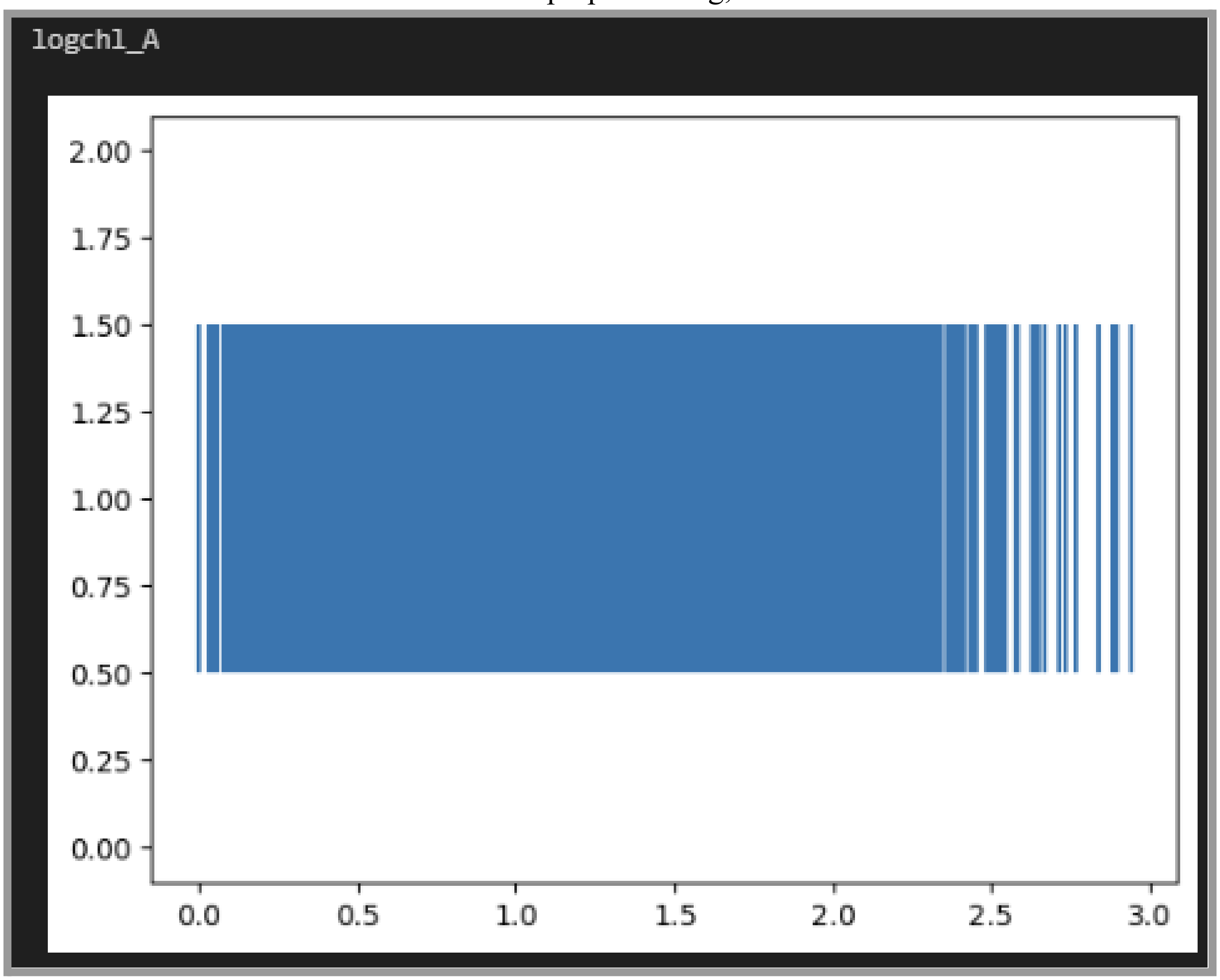
```
0: {'Tmean', 'WSAREA_km2', 'Tot_Sdep_2000', 'Year', 'LST_YrMean', 'Tmean_YrMean', 'OmWs', 'nani', 'SandWs'
1: {"'Total Input'"} -> 0.048%
2: {'logchl_A'} -> 0.095%
32: {'Precip'} -> 1.527%
294: {'wetlands'} -> 14.027%
330: {'SNOW_YrMean'} -> 15.744%
456: {'Human_N_Demand_2007', 'N_Human_Waste_2007', 'N_Fert_Urban_2007'} -> 21.756%
485: {'PctWdWet2011Ws'} -> 23.139%
511: {'AgKffactWs'} -> 24.38%
518: {'P_human_nonfood_demand_kg_2007', 'P_nf_fertilizer_2007', 'P_human_food_demand_kg_2007', 'P_human_wa
615: {'N_Livestock_Food_Demand_2007', 'N_Livestock.Waste_2007', 'PctHbWet2011Ws', 'N_Livestock_N_Content_2
625: {'N_Fert_Farm_2007', 'N_Crop_N_Rem_2007', 'N_CBNF_2007'} -> 29.819%
656: {'P_Accumulated_ag_inputs_2007'} -> 31.298%
659: {'P_livestock_production_2007', 'P_livestock_Waste_2007', 'P_livestock_demand_2007'} -> 31.441%
672: {'P_Crop_removal_2007'} -> 32.061%
713: {'NAPI'} -> 34.017%
718: {'P_f_fertilizer_2007'} -> 34.256%
760: {'Legacy'} -> 36.26%
1080: {'DamDensWs'} -> 51.527%
1925: {'SNOW'} -> 91.842%
```

We also analyzed the number of 0s per instance. This was also done using Pandas and Python on a Jupyter notebook file, resulting in this sort:

```
0: {257, 1925, 646, 648, 138, 12, 140, 142, 148, 918, 895, 155, 539, 1950, 418, 293, 933, 1958, 296, 939, 172, 1580, 821, 949, 824, 955, 61, 830, 1091, 68, 69, 581, 328, 716, 972, 464, 210, 211, 595, 853, 4
1: {1025, 5, 518, 7, 519, 10, 523, 524, 525, 16, 1041, 18, 530, 534, 1558, 24, 2071, 26, 28, 540, 2079, 2081, 34, 547, 1059, 2082, 1062, 39, 551, 1065, 1575, 555, 2044, 45, 557, 2093, 561, 53, 54, 1079, 56,
2: {6, 2055, 2056, 11, 2059, 2063, 2065, 19, 21, 2070, 30, 36, 2089, 57, 63, 83, 85, 88, 91, 92, 94, 96, 98, 115, 116, 120, 121, 125, 130, 135, 136, 139, 146, 147, 153, 158, 169, 171, 173, 174, 184, 200, 21
3: {2048, 1537, 2, 1538, 2051, 1542, 2054, 521, 2058, 13, 14, 15, 1042, 2067, 20, 1556, 1560, 29, 1055, 1568, 545, 546, 35, 38, 1574, 1576, 41, 1066, 43, 1578, 2086, 560, 1074, 1081, 1593, 1596, 62, 1601, 1
4: {512, 1536, 8, 9, 522, 2066, 1563, 2041, 2077, 1056, 1571, 1573, 44, 1587, 568, 569, 1600, 65, 67, 582, 1607, 1613, 1617, 1620, 87, 602, 1626, 1628, 1630, 99, 100, 1636, 108, 110, 1646, 1654, 123, 1663,
5: {1286, 1655, 1798, 1289, 2057, 653, 271, 1911, 1657, 276, 788, 1303, 1913, 2074, 2076, 669, 1310, 1320, 1916, 42, 943, 1583, 49, 1585, 51, 1331, 1918, 1975, 1595, 702, 958, 1728, 1473, 1734, 203, 1740, 7
6: {1284, 389, 1288, 536, 25, 1309, 1317, 166, 1833, 1590, 311, 1727, 320, 66, 964, 1990, 1991, 202, 1742, 1362, 1749, 1366, 1879, 1753, 1763, 743, 1515, 500, 2037} -> 9.524%
7: {1921, 387, 1417, 1548, 2061, 1294, 1553, 1306, 2075, 1438, 1315, 1189, 1335, 1213, 1730, 1870, 1238, 1625, 1627, 1375, 1376, 1635, 1510, 1651, 2038} -> 11.111%
8: {1287, 1671, 1552, 163, 1832, 2092, 1965, 1454, 817, 1586, 1726, 191, 73, 1354, 1739, 1228, 206, 592, 1872, 469, 725, 1623, 1880, 97, 1772, 1395, 1780, 1909, 1527} -> 12.698%
9: {258, 1797, 390, 137, 1935, 1173, 1814, 1559, 1941, 2073, 1659, 1787, 290, 37, 1194, 2091, 1452, 1970, 1078, 1207, 1846, 1979, 1482, 1866, 1994, 208, 1108, 1748, 216, 346, 993, 2024, 748, 495, 2036, 1017
10: {1792, 3, 1283, 1545, 1929, 1555, 1684, 2069, 537, 1564, 1182, 286, 544, 1312, 1698, 298, 684, 47, 1713, 1341, 1088, 1089, 1090, 1859, 584, 721, 852, 1113, 217, 1500, 1888, 1764, 1511, 1384, 1520, 1009,
11: {1285, 1802, 1291, 526, 157, 1825, 309, 1211, 1599, 1222, 1608, 1493, 598, 1494, 2007, 218, 988, 1886, 1503, 1506, 996, 2020, 1642, 1517, 1262, 1521, 1138} -> 17.46%
12: {1597, 1156, 679, 264, 1160, 1223, 1259, 301, 1758, 1752, 1844, 1181, 1176, 1339, 573, 414, 31} -> 19.048%
13: {777, 2062, 786, 1688, 1693, 1057, 1186, 1330, 192, 1348, 1353, 970, 974, 597, 1622, 1754, 362, 107, 1771, 1518, 1908, 886, 2042, 1534} -> 20.635%
14: {2050, 1412, 1546, 2080, 165, 421, 40, 435, 1717, 313, 1340, 1982, 1609, 1487, 731, 990, 1127, 874, 1770, 627, 892, 1279} -> 22.222%
15: {2049, 770, 1795, 1796, 262, 1801, 1679, 144, 528, 529, 914, 1170, 533, 406, 663, 1812, 283, 1051, 292, 423, 1963, 1708, 1581, 1966, 1711, 1584, 305, 1075, 52, 820, 825, 58, 59, 60, 1472, 195, 580, 1989
16: {2060, 22, 541, 46, 558, 2094, 564, 1076, 587, 1103, 95, 1125, 104, 1132, 1133, 1135, 1150, 132, 1157, 665, 1193, 1706, 686, 1714, 1205, 695, 193, 214, 745, 1769, 1778, 768, 1803, 782, 1810, 1816, 818,
17: {1158, 647, 1544, 1936, 17, 401, 1942, 1052, 1696, 1953, 1442, 1316, 549, 808, 1068, 1836, 1582, 433, 178, 50, 1459, 1589, 696, 1464, 1984, 1733, 1480, 1225, 75, 204, 78, 1358, 1874, 606, 2015, 608, 176
18: {1155, 1678, 1168, 417, 554, 1067, 1212, 705, 1346, 76, 1356, 1614, 1114, 480, 1377, 2016, 870, 871, 360, 1131, 1136, 1011, 890} -> 28.571%
19: {907, 1933, 1680, 1043, 1046, 1304, 1319, 439, 1336, 314, 72, 1355, 205, 1357, 1871, 338, 2010, 1120, 1122, 2018, 749, 1774, 1142, 248, 1532, 1662} -> 30.159%
20: {1146, 1147, 1149, 1775} -> 31.746%
21: {1217, 2052, 2085, 1959, 617, 650, 1547, 1898, 1652, 1141, 2009, 1437} -> 33.333%
22: {1676, 1554, 2068, 1815, 791, 1821, 2078, 927, 1954, 550, 810, 182, 1104, 2002, 1504, 738, 1897, 1265, 754, 638} -> 34.921%
23: {642, 516, 645, 263, 1031, 1415, 1034, 1674, 1549, 1275, 1044, 1172, 2072, 1049, 1277, 27, 1690, 1184, 680, 1707, 1709, 175, 559, 2095, 562, 308, 566, 567, 570, 837, 456, 329, 74, 1995, 460, 82, 722, 20
24: {4, 1030, 1033, 1035, 1036, 1037, 1038, 1039, 1050, 1053, 1565, 1566, 552, 1070, 1071, 48, 1072, 563, 1087, 70, 610, 614, 1638, 1129, 1643, 1644, 1656, 131, 1165, 1171, 1178, 1692, 162, 1703, 1704, 1705
25: {1, 513, 515, 1409, 1539, 1665, 1543, 392, 520, 1416, 2053, 656, 913, 1809, 2064, 532, 1813, 1658, 280, 1689, 1948, 542, 415, 671, 33, 1183, 1185, 548, 676, 1569, 1695, 1952, 1957, 2088, 1451, 1579, 685
26: {1930, 1940, 1557, 1685, 1177, 1562, 1567, 32, 2084, 1577, 1322, 2090, 323, 741, 1257, 1261, 751, 1776, 1781, 1151} -> 41.27%
27: {0, 897, 1152, 643, 1159, 1932, 1934, 1040, 1938, 1045, 662, 23, 920, 1047, 154, 1048, 1175, 1180, 1561, 1686, 1694, 1946, 419, 1572, 2087, 1837, 1839, 1712, 1715, 436, 1716, 697, 1849, 321, 1857, 579,
28: {2017, 1162, 1163, 1869, 1144, 1145, 1759} -> 44.444%
```

We chose to drop the *SNOW* attribute from this analysis as it had more than 70% of its values missing, and kept the instances intact.

Due to the extreme variance in magnitude of the data, we decided to normalize all attributes in the dataset. Some attributes had notable outliers, so we used z-score normalization for these. These attributes were as follows: *wsarea_km2, lake_area_ha, fire, fire_yrmean, lst, lst_yrmean, precip_yrmean, tmean, tmean_yrmean, atmo_pdep_2002, atmo_pdep_2007, n_cbnf_2007, n_crop_n_rem_2007, n_fert_farm_2007, n_livestock.waste_2007, n_livestock-_n_content_2007, p_crop_removal_2007, p_livestock_demand_2007, p_livestock_waste_2007, p_livestock_production_2007, p_nf_fertilizer_2007, p_human_food_demand_kg_2007, p_human_nonfood_demand_kg_2007, p_human_waste_kg_2007, p_accumulated_ag_inputs-_2007, napi, total input'* [note that the ' here is not an accidental typo and is included in the name of the attribute], *legacy, damdensws, pcthbwet2011ws,* and *p2o5ws.* For all other attributes, we used min-max normalization. A quick plot of *logchl_A* shows that the data is generally uniformly distributed, likely due to the log scale applied in this dataset (see below image), so we can use random sampling to split the dataset. We used 10-fold validation for this dataset, without it being stratified for the reason listed above. After preprocessing, we had this distribution:



4 Attribute Selection Algorithms and Model Classifiers

After data cleaning and preprocessing, our dataset still had a dimension of 59. This is simply too large for classification algorithms to be used effectively as we would quickly run into issues typical of the curse of dimensionality. What this means is that the algorithms would have a hard time "finding" the trends within the data, as well as the model being far more complex and taking up more space. As the model would be more complex and take up more space, both in memory and in storage, the execution time would greatly increase as well. Thus, it is imperative that the dimensionality of the dataset is reduced. To do this we employed four attribute selection algorithms as well as choosings a set of attributes by hand through a subjective analysis. These attribute selection algorithms will be detailed below.

4.1 Attribute Selection Algorithms Used

4.1a Information Gain

For this approach to attribute selection, we used Weka for the computation. The concept of "gain" in a dataset means the amount of information that can be determined about one variable from another variable, randomly [4]. The methodology behind Information Gain attribute selection is as follows:

A given attribute is represented by A, D is the dataset and p_i is the probability of a given tuple found in D to belong to the class C_i, m is the number of classes in the dataset

Information to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

Same, after splitting D in v partitions by attribute A where D_j is a given partition:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \cdot Info(D_j)$$

The gain of an attribute may be defined as:

$$Gain(A) = Info(D) - Info_A(D)$$

Now, for every attribute in the dataset, its gain is calculated and they are ranked highest to lowest where the higher the gain, the better. The result of this can be seen below.

```
Ranked attributes:
 0.43808     8 ptl
 0.40161     7 ntl
 0.13352     4 lon_dd
 0.12736    59 depth
 0.1254     13 lst_yrmean
 0.10355     9 snow_yrmean
 0.10173    18 tmean
 0.10135    19 tmean_yrmean
 0.10105    57 agkffactws
 0.09842    28 n_fert_farm_2007
 0.0983     12 lst
 0.09745    53 clayws
 0.09506    47 total input
 0.09349    27 n_crop_n_rem_2007
 0.09       36 p_crop_removal_2007
 0.08875    38 p_livestock_demand_2007
 0.08832    14 npp
 0.08732    40 p_livestock_production_2007
 0.08668    39 p_livestock_waste_2007
 0.08393    26 n_cbnf_2007
 0.08101    32 n_livestock_food_demand_2007
 0.08066    34 n_livestock_n_content_2007
 0.07993    45 p_accumulated_ag_inputs_2007
 0.07856     1 nani
 0.07651    56 bfiws
 0.07322    48 legacy
 0.073      37 p_f_fertilizer_2007
 0.0712     33 n_livestock.waste_2007
 0.0613     54 sandws
 0.05872    23 tot_ndep_2007
 0.05764    21 atmo_pdep_2007
 0.0552     52 runoffws
 0.05017     3 lat_dd
 0.04872    22 tot_ndep_2000
 0.04641    15 npp_yrmean
 0.04218    35 n_rock_2007
 0.03531    29 n_fert_urban_2007
 0.03214    20 atmo_pdep_2002
 0.03157    16 precip
 0.03119    46 napi
 0.02726    24 tot_sdep_2000
 0.02677    31 n_human_waste_2007
 0.02677    30 human_n_demand_2007
 0.0265     41 p_nf_fertilizer_2007
 0.0242     17 precip_yrmean
 0.0232     25 tot_sdep_2007
 0.02064    43 p_human_nonfood_demand_kg_2007
 0.01975    44 p_human_waste_kg_2007
 0.01972    42 p_human_food_demand_kg_2007
 0.01969    51 pcthbwet2011ws
 0.01581    55 omws
 0.01463    49 damdensws
 0.01341    11 fire_yrmean
 0.01293     5 wsarea_km2
 0.00981    10 fire
 0          50 pctwdwet2011ws
 0           6 lake_area_ha
 0           2 wetlands
 0          58 p2o5ws
```

We chose a cutoff of 0.1, leaving us with nine attributes: *ptl*, *ntl*, *lon_dd*, *depth*, *lst_yrmean*, *snow_yrmean*, *mean*, *tmean_yrmean*, and *agkffactws*. The reason for this choice of cutoff is that is leaves us with a dimension of nine – neither too high as to induce the curse of dimensionality, nor too low as to leave no information for the classifier algorithms to use.

4.2b *Principal Component Analysis*

We once again used Weka here to perform all of the computations associated with PCA. In essence, what PCA does is that it transforms the initial dataset into a new one with new attributes where the values and attributes are selected in such a way as to maximize variance, with the highest variance attributes being ranked the highest [6]. The result of running PCA on our dataset can be found below.

```
Ranked attributes:
 0.717      1 -0.213total input-0.213nani-0.211p_accumulated_ag_inputs_2007-0.211n_livestock_food_demand_2007-0.209n_livestock.waste_2007...
 0.5708     2 -0.241n_human_waste_2007-0.239human_n_demand_2007-0.239p_human_waste_kg_2007-0.237p_human_food_demand_kg_2007-0.227n_fert_urban_2007...
 0.475      3 0.307lst+0.271lst_yrmean-0.247npp-0.216omws-0.191lat_dd...
 0.4106     4 -0.252lat_dd+0.224atmo_pdep_2002+0.215atmo_pdep_2007-0.211p_human_food_demand_kg_2007-0.207p_human_nonfood_demand_kg_2007...
 0.3622     5 0.41 wetlands+0.349pctwdwet2011ws+0.314pcthbwet2011ws-0.29depth+0.225ntl...
 0.3244     6 0.284ntl-0.254depth+0.249ptl-0.223napi+0.218n_rock_2007...
 0.2986     7 0.536wsarea_km2+0.51 lake_area_ha+0.237depth-0.21runoffws-0.208npp_yrmean...
 0.2749     8 0.368lake_area_ha+0.346wsarea_km2-0.292sandws-0.255atmo_pdep_2007+0.255npp...
 0.2523     9 -0.469fire_yrmean-0.42fire-0.339p_f_fertilizer_2007-0.246p2o5ws+0.208wsarea_km2...
 0.2326    10 0.328napi+0.321precip-0.265n_cbnf_2007+0.25 ptl-0.246bfiws...
 0.2141    11 0.524n_rock_2007+0.499damdensws+0.257fire-0.24omws+0.196fire_yrmean...
 0.1973    12 0.626p2o5ws-0.36fire+0.278damdensws-0.278fire_yrmean+0.207wetlands...
 0.181     13 -0.583p2o5ws+0.489damdensws+0.247p_f_fertilizer_2007-0.173fire+0.172depth...
 0.1668    14 0.406damdensws-0.397clayws+0.341p2o5ws-0.301pctwdwet2011ws+0.262sandws...
 0.1535    15 -0.458fire+0.358fire_yrmean+0.342pcthbwet2011ws-0.323damdensws+0.319n_rock_2007...
 0.141     16 0.527fire-0.466fire_yrmean-0.249bfiws+0.242pcthbwet2011ws+0.221n_rock_2007...
 0.1293    17 -0.414omws-0.382fire_yrmean+0.343runoffws+0.284p_f_fertilizer_2007-0.264precip...
 0.1186    18 -0.519n_rock_2007+0.334pcthbwet2011ws-0.294omws-0.275tmean+0.251atmo_pdep_2002...
 0.109     19 0.389precip-0.388lake_area_ha+0.327wsarea_km2-0.317pcthbwet2011ws+0.268lon_dd...
 0.0998    20 -0.384pcthbwet2011ws+0.322npp+0.317p_f_fertilizer_2007-0.312p_nf_fertilizer_2007-0.302lat_dd...
 0.0916    21 0.488wsarea_km2-0.447lake_area_ha-0.412precip-0.308bfiws-0.193p_nf_fertilizer_2007...
 0.0838    22 -0.4omws-0.364wsarea_km2+0.348lake_area_ha+0.279pctwdwet2011ws-0.255pcthbwet2011ws...
 0.0763    23 0.482precip_yrmean-0.423npp-0.252lon_dd-0.204tot_sdep_2000-0.188lat_dd...
 0.0694    24 -0.429p_nf_fertilizer_2007+0.3  bfiws-0.251p_crop_removal_2007+0.222human_n_demand_2007+0.216n_human_waste_2007...
 0.063     25 0.457atmo_pdep_2002-0.383bfiws-0.337tot_ndep_2007-0.326tot_ndep_2000+0.258atmo_pdep_2007...
 0.0571    26 0.642depth+0.382ntl-0.363p_f_fertilizer_2007+0.278ptl+0.145p_nf_fertilizer_2007...
 0.0515    27 0.491legacy+0.33 p_accumulated_ag_inputs_2007-0.311agkffactws-0.251tmean+0.236p_nf_fertilizer_2007...
 0.0464    28 0.383n_fert_urban_2007-0.323legacy-0.29p_human_nonfood_demand_kg_2007+0.288p_nf_fertilizer_2007-0.229precip_yrmean...

Selected attributes: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28 : 28
```

Here we chose a threshold of 0.2, as this would leave with 11 attributes, not seriously deviating from the previous amount so as to maintain consistency and also continuing to walk the balance between the curse of dimensionality and oversimplification.

4.1c Learner Based w/ J48

Once again, we used Weka for this attribute selection algorithm. J48 is an open-source Java implementation of the popular C4.5 decision tree algorithm [7]. This algorithm utilizes gain as defined earlier to continually split the dataset and thus generate an effective decision tree. The algorithm then chooses the most important features for prediction and the results of this can be seen below.

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 651
        Merit of best subset found:    0.784

Attribute Subset Evaluator (supervised, Class (nominal): 60 logchl_A):
        Wrapper Subset Evaluator
        Learning scheme: weka.classifiers.trees.J48
        Scheme options: -C 0.25 -M 2
        Subset evaluation: classification accuracy
        Number of folds for accuracy estimation: 5

Selected attributes: 4,7,8,20,38,40,50 : 7
                     lon_dd
                     ntl
                     ptl
                     atmo_pdep_2002
                     p_livestock_demand_2007
                     p_livestock_production_2007
                     pctwdwet2011ws
```

4.1d OneR

For our final attribute selection algorithm, we chose OneR attribute evaluation. This algorithm produces a single rule for any given pairing of attribute and value and ranks these rules by accuracy to find the best one [5]. The pseudo code is below.

For each attribute in the dataset
        For each value in the current attribute
                Find most frequent class for given value of the given attribute
                Create rule that assigns most frequent class to this attribute-value pairing
        Compute error of feature by summing all rule error values
Rank attributes by error with lowest error rate being best

The result of evaluating the features of our dataset using this algorithm are below.

```
Attribute Evaluator (supervised, Class (nominal): 60 logchl_A):
        OneR feature evaluator.

        Using 10 fold cross validation for evaluating attributes.
        Minimum bucket size for OneR: 6

Ranked attributes:
71.56489     8 ptl
68.2729      7 ntl
59.58969    28 n_fert_farm_2007
59.25573     4 lon_dd
59.16031    45 p_accumulated_ag_inputs_2007
58.77863    57 agkffactws
58.6355     20 atmo_pdep_2002
58.58779    36 p_crop_removal_2007
58.54008    56 bfiws
58.49237    33 n_livestock.waste_2007
58.06298    18 tmean
58.01527    39 p_livestock_waste_2007
57.96756    21 atmo_pdep_2007
57.96756    32 n_livestock_food_demand_2007
57.87214    40 p_livestock_production_2007
57.6813     47 total input
57.58588    27 n_crop_n_rem_2007
57.06107    34 n_livestock_n_content_2007
56.91794    38 p_livestock_demand_2007
56.91794    19 tmean_yrmean
56.82252     9 snow_yrmean
56.82252    12 lst
56.7271     13 lst_yrmean
56.29771    53 clayws
56.25        1 nani
56.10687    22 tot_ndep_2000
56.10687    48 legacy
56.05916    54 sandws
55.96374    59 depth
55.67748    26 n_cbnf_2007
54.77099     3 lat_dd
54.3416     51 pcthbwet2011ws
54.00763    37 p_f_fertilizer_2007
53.43511    46 napi
53.43511    35 n_rock_2007
53.05344    23 tot_ndep_2007
52.95802    14 npp
52.48092    58 p2o5ws
52.43321    11 fire_yrmean
52.33779    10 fire
52.19466    49 damdensws
52.00382    52 runoffws
51.66985     6 lake_area_ha
51.52672    41 p_nf_fertilizer_2007
51.28817     2 wetlands
51.09733    15 npp_yrmean
51.04962    29 n_fert_urban_2007
50.66794    24 tot_sdep_2000
50.52481    43 p_human_nonfood_demand_kg_2007
50.38168    16 precip
50.04771    44 p_human_waste_kg_2007
49.95229    55 omws
49.85687    50 pctwdwet2011ws
49.80916    42 p_human_food_demand_kg_2007
49.37977     5 wsarea_km2
49.37977    31 n_human_waste_2007
49.28435    30 human_n_demand_2007
49.0458     25 tot_sdep_2007
46.18321    17 precip_yrmean
```

We chose a threshold of 58.5, as this would leave us with 9 attributes – a reasonable amount and one that is similar to the amounts of attributes kept using other attribute selection algorithms, thus preventing too much inconsistency. We are left with *ptl*, *ntl*, *n_fert_farm_2007*, *lon_dd*, *p_accumulated_ag_inputs_2007*, *agkffactws*, *atmo_pdep_2002*, *p_crop_removal_2007*, *bfiws*.

4.1e Subjective Analysis / Hand-picking

Finally, for the subjective approach we picked 10 attributes which we thought could contribute to chlorophyll-a levels. *lon_dd*, the longitude of the lake would be important since longitude can indirectly imply climate patterns like distance from coastlines and the presence of mountain ranges. We also chose *lat_dd*, the latitude of the lake, as it is another geographical measure and could give some indication as to the climate or overall temperature around a lake, as high latitudes are generally colder and drier. *ntl,* total nitrogen concentration, and *ptl,* total phosphorus concentration, would be a great indicator for algae growth as they both directly affect the rate of algae growth. Too much nitrogen or phosphorus will cause algae to grow faster than ecosystems can handle, which is what we are trying to prevent with chlorophyll-a concentrations. *atmo_pdep_2002,* the annual average phosphorus deposition in 2002, will also give our model more information on the amount of phosphorus in the lake. There is also *atmo_pdep_2007,* but we decided that only one indicator of phosphorus deposition would be necessary. *n_human_waste_2007* and *n_livestock_waste_2007* give the annual average of nitrogen from human and livestock waste, and *p_human_waste_2007* and *p_livestock_waste_2007* give the annual average phosphorus from human and livestock demand. These four attributes were selected since we suspect that humans and animals farms have a large influence on the surrounding environments (i.e. lakes) and the phosphorus and nitrogen from our waste and demands are contributing to algae growth. Our last attribute selected was *runoffws*, the mean runoff within the given watershed, which we suspected contributed to algae growth as runoff would carry nutrients, like phosphorus and nitrogen, to the lakes to feed the algae.

4.2 Classifier Models

4.2a Naive Bayes

The Naive Bayes classifier works as follows:

Given a training set of labeled tuples, *D*, some tuple *X* with *n* attributes ($x_1, x_2, x_3, ..., x_n$), where x_i is the value for the attribute A_i and *m* classes are represented by $C_1, C_2, C_3, ..., C_m$

The probability of *X* belonging to class C_k can be predicted recursively as:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}$$

When assuming no dependence between attributes the formula can be simplified to:

$$P(X|C_k) = \prod_{i=1}^{n} P(x_i|C_k)$$

This can be used to predict the class of an instance by performing this probability calculation on every class (C_1 -> C_k) and taking the class with the highest probability as the prediction [9].

4.2b Logistic Regression

This algorithm works through the estimation of the parameters of a logistic model. This is somewhat similar to a linear regression except that the parameters are μ and *s* instead of *m* and *b*. In the logistic model μ controls the location of the midpoint while *s* controls the scale of the curve. These two values are optimized to minimize the error. After this, predictions are made by using the output of the model as the prediction and the input into the function as the input for the unlabeled instances [10].

4.3c Learner Based w/ J48

The J48 classification algorithm works in much the same way as the same attribute selection algorithm, except instead of using the decision tree to determine the most important attributes, the decision tree is used to actually predict the class for new instances[7].

4.4d RandomTree

The random tree algorithm is in a way similar to the J48/C4.5 algorithm in that it constructs a tree, but in this case at each node there are *k* branches. What gives the algorithm is “random” name is that at each node, it considers *k* attributes entirely randomly [8]. This then builds out into a decision tree that can be used for prediction of class on unlabeled data.

5 Results and Analysis

5.1 Results

Results ordered by attribute selection algorithm:

5.1a Information Gain Attribute Selection

InfoGainAttributeEval with Naive Bayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1524               72.7099 %
Incorrectly Classified Instances       572               27.2901 %
Kappa statistic                          0.5249
Mean absolute error                      0.1985
Root mean squared error                  0.3754
Relative absolute error                 53.1074 %
Root relative squared error             86.8351 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.726    0.149    0.835      0.726    0.777      0.581    0.871     0.876     low
                 0.741    0.262    0.669      0.741    0.703      0.474    0.814     0.722     medium
                 0.654    0.051    0.503      0.654    0.568      0.535    0.935     0.564     high
Weighted Avg.    0.727    0.189    0.742      0.727    0.731      0.533    0.852     0.789

=== Confusion Matrix ===

   a   b   c   <-- classified as
 777 268  25 |   a = low
 152 647  74 |   b = medium
   1  52 100 |   c = high
```

InfoGainAttributeEval with Logistic

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1637               78.1011 %
Incorrectly Classified Instances       459               21.8989 %
Kappa statistic                          0.6019
Mean absolute error                      0.2213
Root mean squared error                  0.3286
Relative absolute error                 59.1931 %
Root relative squared error             76.0223 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.845    0.188    0.824      0.845    0.834      0.657    0.893     0.899     low
                 0.759    0.196    0.734      0.759    0.747      0.561    0.851     0.769     medium
                 0.458    0.013    0.729      0.458    0.562      0.553    0.948     0.635     high
Weighted Avg.    0.781    0.179    0.780      0.781    0.778      0.609    0.879     0.826

=== Confusion Matrix ===

   a   b   c   <-- classified as
 904 161   5 |   a = low
 189 663  21 |   b = medium
   4  79  70 |   c = high
```

InfoGainAttributeEval with J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1604               76.5267 %
Incorrectly Classified Instances       492               23.4733 %
Kappa statistic                          0.5826
Mean absolute error                      0.1923
Root mean squared error                  0.3595
Relative absolute error                 51.4484 %
Root relative squared error             83.1558 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.824    0.195    0.815      0.824    0.820      0.630    0.841     0.785     low
                 0.715    0.182    0.737      0.715    0.726      0.535    0.784     0.685     medium
                 0.641    0.036    0.587      0.641    0.612      0.581    0.862     0.479     high
Weighted Avg.    0.765    0.178    0.766      0.765    0.765      0.587    0.819     0.721

=== Confusion Matrix ===

   a   b   c   <-- classified as
 882 177  11 |   a = low
 191 624  58 |   b = medium
   9  46  98 |   c = high
```

InfoGainAttributeEval with RandomTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1445               68.9408 %
Incorrectly Classified Instances       651               31.0592 %
Kappa statistic                          0.4493
Mean absolute error                      0.2071
Root mean squared error                  0.455
Relative absolute error                 55.388  %
Root relative squared error            105.2659 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.749    0.238    0.767      0.749    0.757      0.511    0.755     0.702     low
                 0.648    0.260    0.640      0.648    0.644      0.388    0.694     0.562     medium
                 0.510    0.046    0.467      0.510    0.488      0.446    0.732     0.274     high
Weighted Avg.    0.689    0.233    0.692      0.689    0.691      0.455    0.728     0.612

=== Confusion Matrix ===

   a   b   c   <-- classified as
 801 257  12 |   a = low
 230 566  77 |   b = medium
  14  61  78 |   c = high
```

5.1b Learner Based Classification Attribute Selection

LearnerBased (J48) with Naive Bayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1519               72.4714 %
Incorrectly Classified Instances       577               27.5286 %
Kappa statistic                          0.5166
Mean absolute error                      0.2073
Root mean squared error                  0.3645
Relative absolute error                 55.4388 %
Root relative squared error             84.3141 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.797    0.220    0.791      0.797    0.794      0.577    0.864     0.880     low
                 0.652    0.195    0.705      0.652    0.677      0.463    0.820     0.712     medium
                 0.634    0.058    0.462      0.634    0.534      0.499    0.930     0.554     high
Weighted Avg.    0.725    0.198    0.731      0.725    0.726      0.524    0.851     0.786

=== Confusion Matrix ===

   a   b   c   <-- classified as
 853 186  31 |   a = low
 222 569  82 |   b = medium
   4  52  97 |   c = high
```

LearnerBased (J48) with Logistic

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1609               76.7653 %
Incorrectly Classified Instances       487               23.2347 %
Kappa statistic                          0.5771
Mean absolute error                      0.2303
Root mean squared error                  0.3359
Relative absolute error                 61.6165 %
Root relative squared error             77.7099 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.836    0.200    0.814      0.836    0.825      0.637    0.884     0.894     low
                 0.743    0.208    0.719      0.743    0.731      0.533    0.837     0.733     medium
                 0.425    0.014    0.699      0.425    0.528      0.518    0.945     0.615     high
Weighted Avg.    0.768    0.190    0.766      0.768    0.764      0.585    0.869     0.807

=== Confusion Matrix ===

   a   b   c   <-- classified as
 895 170   5 |   a = low
 201 649  23 |   b = medium
   4  84  65 |   c = high
```

LearnerBased (J48) with J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1635               78.0057 %
Incorrectly Classified Instances       461               21.9943 %
Kappa statistic                          0.6056
Mean absolute error                      0.2005
Root mean squared error                  0.3412
Relative absolute error                 53.6255 %
Root relative squared error             78.9204 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.831    0.187    0.822      0.831    0.827      0.644    0.860     0.818     low
                 0.745    0.186    0.741      0.745    0.743      0.559    0.813     0.710     medium
                 0.627    0.022    0.696      0.627    0.660      0.635    0.870     0.549     high
Weighted Avg.    0.780    0.174    0.779      0.780    0.780      0.608    0.841     0.753

=== Confusion Matrix ===

   a   b   c   <-- classified as
 889 175   6 |   a = low
 187 650  36 |   b = medium
   5  52  96 |   c = high
```

LearnerBased (J48) with RandomTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1481               70.6584 %
Incorrectly Classified Instances       615               29.3416 %
Kappa statistic                          0.4786
Mean absolute error                      0.1956
Root mean squared error                  0.4423
Relative absolute error                 52.3251 %
Root relative squared error            102.3139 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.779    0.233    0.777      0.779    0.778      0.546    0.773     0.718     low
                 0.653    0.235    0.665      0.653    0.659      0.419    0.709     0.579     medium
                 0.510    0.046    0.467      0.510    0.488      0.446    0.732     0.274     high
Weighted Avg.    0.707    0.220    0.708      0.707    0.707      0.486    0.743     0.628

=== Confusion Matrix ===

   a   b   c   <-- classified as
 833 226  11 |   a = low
 225 570  78 |   b = medium
  14  61  78 |   c = high
```

5.1c Principal Component Analysis Attribute Selection

PCA with Naive Bayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1455               69.4179 %
Incorrectly Classified Instances       641               30.5821 %
Kappa statistic                          0.442
Mean absolute error                      0.2808
Root mean squared error                  0.3792
Relative absolute error                 75.1019 %
Root relative squared error             87.7266 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.726    0.212    0.782      0.726    0.753      0.515    0.822     0.819     low
                 0.740    0.327    0.618      0.740    0.673      0.407    0.750     0.612     medium
                 0.209    0.012    0.571      0.209    0.306      0.317    0.844     0.350     high
Weighted Avg.    0.694    0.245    0.698      0.694    0.687      0.456    0.794     0.699

=== Confusion Matrix ===

   a   b   c   <-- classified as
 777 285   8 |   a = low
 211 646  16 |   b = medium
   6 115  32 |   c = high
```

PCA with Logistic

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1548               73.855  %
Incorrectly Classified Instances       548               26.145  %
Kappa statistic                          0.5212
Mean absolute error                      0.2505
Root mean squared error                  0.3539
Relative absolute error                 67.0183 %
Root relative squared error             81.8626 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.817    0.219    0.795      0.817    0.806      0.598    0.861     0.874     low
                 0.718    0.244    0.678      0.718    0.697      0.471    0.802     0.697     medium
                 0.307    0.013    0.653      0.307    0.418      0.420    0.916     0.505     high
Weighted Avg.    0.739    0.214    0.736      0.739    0.732      0.532    0.841     0.773

=== Confusion Matrix ===

   a   b   c   <-- classified as
 874 193   3 |   a = low
 224 627  22 |   b = medium
   1 105  47 |   c = high
```

PCA with J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1406               67.0802 %
Incorrectly Classified Instances       690               32.9198 %
Kappa statistic                          0.4052
Mean absolute error                      0.2567
Root mean squared error                  0.4218
Relative absolute error                 68.667  %
Root relative squared error             97.5778 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.755    0.279    0.739      0.755    0.747      0.477    0.764     0.711     low
                 0.637    0.270    0.628      0.637    0.632      0.366    0.691     0.577     medium
                 0.275    0.038    0.362      0.275    0.312      0.269    0.701     0.205     high
Weighted Avg.    0.671    0.257    0.665      0.671    0.667      0.416    0.729     0.618

=== Confusion Matrix ===

   a   b   c   <-- classified as
 808 240  22 |   a = low
 265 556  52 |   b = medium
  21  90  42 |   c = high
```

PCA with RandomTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1294               61.7366 %
Incorrectly Classified Instances       802               38.2634 %
Kappa statistic                          0.3174
Mean absolute error                      0.2551
Root mean squared error                  0.5051
Relative absolute error                 68.2353 %
Root relative squared error            116.838  %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.710    0.317    0.700      0.710    0.705      0.394    0.697     0.645     low
                 0.561    0.298    0.574      0.561    0.567      0.265    0.632     0.505     medium
                 0.288    0.058    0.280      0.288    0.284      0.227    0.615     0.133     high
Weighted Avg.    0.617    0.290    0.617      0.617    0.617      0.328    0.664     0.549

=== Confusion Matrix ===

   a   b   c   <-- classified as
 760 280  30 |   a = low
 300 490  83 |   b = medium
  25  84  44 |   c = high
```

5.1d OneR Classifier Attribute Selection

OneR with Naive Bayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1440               68.7023 %
Incorrectly Classified Instances       656               31.2977 %
Kappa statistic                          0.448
Mean absolute error                      0.2183
Root mean squared error                  0.4049
Relative absolute error                 58.3892 %
Root relative squared error             93.6596 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.285    0.740      0.778    0.758      0.494    0.837     0.859     low
                 0.585    0.208    0.668      0.585    0.624      0.387    0.786     0.646     medium
                 0.634    0.057    0.469      0.634    0.539      0.503    0.925     0.536     high
Weighted Avg.    0.687    0.236    0.690      0.687    0.686      0.450    0.822     0.747

=== Confusion Matrix ===

   a   b   c   <-- classified as
 832 210  28 |   a = low
 280 511  82 |   b = medium
  12  44  97 |   c = high
```

OneR with Logistic

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1616               77.0992 %
Incorrectly Classified Instances       480               22.9008 %
Kappa statistic                          0.5829
Mean absolute error                      0.23
Root mean squared error                  0.3358
Relative absolute error                 61.5363 %
Root relative squared error             77.6753 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.842    0.202    0.813      0.842    0.827      0.641    0.885     0.894     low
                 0.742    0.202    0.724      0.742    0.733      0.538    0.838     0.741     medium
                 0.438    0.013    0.720      0.438    0.545      0.536    0.946     0.609     high
Weighted Avg.    0.771    0.188    0.769      0.771    0.767      0.591    0.870     0.810

=== Confusion Matrix ===

   a   b   c   <-- classified as
 901 165   4 |   a = low
 203 648  22 |   b = medium
   4  82  67 |   c = high
```

OneR with J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1610               76.813  %
Incorrectly Classified Instances       486               23.187  %
Kappa statistic                          0.5839
Mean absolute error                      0.2025
Root mean squared error                  0.3549
Relative absolute error                 54.1781 %
Root relative squared error             82.0986 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.824    0.199    0.812      0.824    0.818      0.626    0.844     0.788     low
                 0.731    0.192    0.731      0.731    0.731      0.539    0.789     0.687     medium
                 0.588    0.024    0.657      0.588    0.621      0.594    0.851     0.469     high
Weighted Avg.    0.768    0.183    0.767      0.768    0.767      0.587    0.822     0.723

=== Confusion Matrix ===

   a   b   c   <-- classified as
 882 181   7 |   a = low
 195 638  40 |   b = medium
   9  54  90 |   c = high
```

OneR with RandomTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1496               71.374  %
Incorrectly Classified Instances       600               28.626  %
Kappa statistic                          0.4921
Mean absolute error                      0.1908
Root mean squared error                  0.4369
Relative absolute error                 51.0489 %
Root relative squared error            101.0585 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.773    0.225    0.782      0.773    0.777      0.548    0.774     0.720     low
                 0.670    0.233    0.672      0.670    0.671      0.437    0.719     0.588     medium
                 0.549    0.043    0.500      0.549    0.523      0.485    0.753     0.307     high
Weighted Avg.    0.714    0.215    0.716      0.714    0.715      0.497    0.749     0.635

=== Confusion Matrix ===

   a   b   c   <-- classified as
 827 228  15 |   a = low
 219 585  69 |   b = medium
  12  57  84 |   c = high
```

5.1e Subjective/Hand-picked Attribute Selection

Hand-picked with Naive Bayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1518               72.4237 %
Incorrectly Classified Instances       578               27.5763 %
Kappa statistic                          0.5203
Mean absolute error                      0.2053
Root mean squared error                  0.3727
Relative absolute error                 54.9299 %
Root relative squared error             86.2239 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.771    0.182    0.815      0.771    0.793      0.589    0.864     0.878     low
                 0.684    0.215    0.694      0.684    0.689      0.470    0.812     0.698     medium
                 0.627    0.066    0.429      0.627    0.509      0.473    0.914     0.505     high
Weighted Avg.    0.724    0.187    0.737      0.724    0.729      0.531    0.846     0.776

=== Confusion Matrix ===

   a   b   c   <-- classified as
 825 208  37 |   a = low
 185 597  91 |   b = medium
   2  55  96 |   c = high
```

Hand-picked with Logistic

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1628               77.6718 %
Incorrectly Classified Instances       468               22.3282 %
Kappa statistic                          0.5943
Mean absolute error                      0.2178
Root mean squared error                  0.327
Relative absolute error                 58.2606 %
Root relative squared error             75.6474 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.848    0.192    0.822      0.848    0.834      0.656    0.896     0.902     low
                 0.747    0.196    0.731      0.747    0.739      0.549    0.852     0.778     medium
                 0.451    0.016    0.690      0.451    0.545      0.531    0.951     0.633     high
Weighted Avg.    0.777    0.181    0.774      0.777    0.773      0.603    0.882     0.831

=== Confusion Matrix ===

   a   b   c   <-- classified as
 907 158   5 |   a = low
 195 652  26 |   b = medium
   2  82  69 |   c = high
```

Hand-picked with J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1588               75.7634 %
Incorrectly Classified Instances       508               24.2366 %
Kappa statistic                          0.5645
Mean absolute error                      0.2041
Root mean squared error                  0.3612
Relative absolute error                 54.5893 %
Root relative squared error             83.5475 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.835    0.230    0.791      0.835    0.812      0.606    0.836     0.782     low
                 0.693    0.178    0.735      0.693    0.713      0.520    0.784     0.676     medium
                 0.588    0.028    0.625      0.588    0.606      0.576    0.858     0.481     high
Weighted Avg.    0.758    0.194    0.756      0.758    0.756      0.568    0.816     0.716

=== Confusion Matrix ===

   a   b   c   <-- classified as
 893 166  11 |   a = low
 225 605  43 |   b = medium
  11  52  90 |   c = high
```

Hand-picked with RandomTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1491               71.1355 %
Incorrectly Classified Instances       605               28.8645 %
Kappa statistic                          0.485
Mean absolute error                      0.1924
Root mean squared error                  0.4387
Relative absolute error                 51.4743 %
Root relative squared error            101.4787 %
Total Number of Instances             2096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.782    0.251    0.764      0.782    0.773      0.531    0.765     0.709     low
                 0.647    0.226    0.671      0.647    0.659      0.423    0.710     0.581     medium
                 0.582    0.036    0.560      0.582    0.571      0.536    0.773     0.356     high
Weighted Avg.    0.711    0.225    0.711      0.711    0.711      0.487    0.743     0.630

=== Confusion Matrix ===

   a   b   c   <-- classified as
 837 228   5 |   a = low
 243 565  65 |   b = medium
  15  49  89 |   c = high
```

5.2 Analysis

Model	Accuracy (%)	TPR High	FPR High	ROC High	TPR Weighted Avg.	FPR Weighted Avg.	ROC Weighted Avg.
InfoGain-Bayes	72.71%	0.654	0.051	0.935	0.727	0.189	0.852
InfoGain-Logistic	78.10%	0.458	0.013	0.948	0.781	0.179	0.879
InfoGain-J48	76.53%	0.641	0.036	0.862	0.765	0.178	0.819
InfoGain-Tree	68.94%	0.51	0.046	0.732	0.689	0.233	0.728
PCA-Bayes	69.42%	0.209	0.012	0.844	0.694	0.245	0.794
PCA-Logistic	73.86%	0.307	0.013	0.916	0.739	0.214	0.841
PCA-J48	67.08%	0.275	0.038	0.701	0.671	0.257	0.729
PCA-Tree	61.74%	0.288	0.058	0.615	0.617	0.29	0.664
Learner-Bayes	72.47%	0.634	0.058	0.93	0.725	0.198	0.851
Learner-Logistic	76.77%	0.425	0.014	0.945	0.768	0.19	0.869
Learner-J48	78.01%	0.627	0.022	0.87	0.78	0.174	0.841
Learner-Tree	70.66%	0.51	0.046	0.732	0.707	0.22	0.743
OneR-Bayes	68.70%	0.634	0.057	0.925	0.687	0.236	0.822
OneR-Logistic	77.10%	0.438	0.013	0.946	0.771	0.188	0.87
OneR-J48	76.81%	0.588	0.024	0.851	0.768	0.183	0.822
OneR-Tree	71.37%	0.549	0.043	0.753	0.714	0.215	0.749
HandPicked-Bayes	72.42%	0.627	0.066	0.914	0.724	0.187	0.846
HandPicked-Logistic	77.67%	0.451	0.016	0.951	0.777	0.181	0.882
Handpicked-J48	75.76%	0.588	0.028	0.858	0.758	0.194	0.816
HandPicked-Tree	71.14%	0.582	0.036	0.773	0.711	0.225	0.743

Note: Highlighted top ten scores in each column of measure

After running 4 models on our 5 datasets, we found results with accuracies ranging from 61-79%. We noted six different measures of our results aside from accuracy: true positive rate (TPR), false positive rate (FPR), and ROC area of high concentrations, and the weighted averages of true positive rate, false positive rate, and ROC area. In the table above, our results with these measures are shown. We highlighted the top ten highest accuracies, highest TPR, lowest FPR, and highest ROC for both high concentrations and the weighted averages for each model. We want to find the most well-rounded model with good results for each measure, so we noted the models with the most amount of good measure scores (top ten scores). Some well-rounded models include InfoGain-Logistic, Learner-Logistic, OneR-J48, and

HandPicked-Logistic which have top ten measures in 6/7 of the columns. All of these four models except OneR-J48 do not achieve top ten in TPR of high concentration, which is not ideal for our model as we are trying to predict if a lake will have a high concentration of chlorophyll-a or not. Because of this, another strong model would be InfoGain-Bayes, which has the highest TPR for high concentration (0.654) and top ten scores for accuracy, ROC area of high concentration, FPR for the weighted average, and ROC area of the weighted average. Although InfoGain-Bayes has one of the worst FPR for high concentration (0.051), this is not a major problem as it is better to check a healthy lake for signs of harmful algal bloom.

We found that the Learner-Based attribute selection algorithm with the J48 classification algorithm produced the best results. Learner-J48 was the only model with top ten measures in every column. Learner-J48 had the second highest accuracy, fifth highest TPR for high concentrations, seventh lowest FPR for high concentrations, tenth highest ROC area for high concentrations, second highest TPR for the weighted average, tenth lowest FPR for weighted average, and seventh highest ROC area for the weighted average. Although InfoGain-Bayes has a higher TPR for high concentrations, we have confidence that the Learner-J48 model will give us better predictions overall because of its higher accuracies for all seven measures.

For future reference, we can observe which attributes were most important in each attribute selection algorithm to gain an understanding of which factors play the most important role in ecosystem health.

InfoGain	OneR	LearnerBased (J48)
0.43808 8 ptl 0.40161 7 ntl 0.13352 4 lon_dd 0.12736 59 depth 0.1254 13 lst_yrmean 0.10355 9 snow_yrmean 0.10173 18 tmean 0.10135 19 tmean_yrmean 0.10105 57 agkffactws 0.09842 28 n_fert_farm_2007 0.0983 12 lst 0.09745 53 clayws 0.09506 47 total input 0.09349 27 n_crop_n_rem_2007 0.09 36 p_crop_removal_2007 0.08875 38 p_livestock_demand_2007 0.08832 14 npp 0.08732 40 p_livestock_production_2007 0.08668 39 p_livestock_waste_2007 0.08393 26 n_cbnf_2007 0.08101 32 n_livestock_food_demand_2007 0.08066 34 n_livestock_n_content_2007 0.07993 45 p_accumulated_ag_inputs_2007 0.07856 1 nani 0.07651 56 bfiws 0.07322 48 legacy 0.073 37 p_f_fertilizer_2007	71.56489 8 ptl 68.2729 7 ntl 59.58969 28 n_fert_farm_2007 59.25573 4 lon_dd 59.16031 45 p_accumulated_ag_inputs_2007 58.77863 57 agkffactws 58.6355 20 atmo_pdep_2002 58.58779 36 p_crop_removal_2007 58.54008 56 bfiws 58.49237 33 n_livestock.waste_2007 58.06298 18 tmean 58.01527 39 p_livestock_waste_2007 57.96756 21 atmo_pdep_2007 57.96756 32 n_livestock_food_demand_2007 57.87214 40 p_livestock_production_2007 57.6813 47 total input 57.58588 27 n_crop_n_rem_2007 57.06107 34 n_livestock_n_content_2007 56.91794 38 p_livestock_demand_2007 56.91794 19 tmean_yrmean 56.82252 9 snow_yrmean 56.82252 12 lst 56.7271 13 lst_yrmean 56.29771 53 clayws 56.25 1 nani 56.10687 22 tot_ndep_2000 56.10687 48 legacy	lon_dd ntl ptl atmo_pdep_2002 p_livestock_demand_2007 p_livestock_production_2007 pctwdwet2011ws

0.0712 33 n_livestock.waste_2007	56.05916 54 sandws	
0.0613 54 sandws	55.96374 59 depth	
0.05872 23 tot_ndep_2007	55.67748 26 n_cbnf_2007	
0.05764 21 atmo_pdep_2007	54.77099 3 lat_dd	
0.0552 52 runoffws	54.3416 51 pcthbwet2011ws	
0.05017 3 lat_dd	54.00763 37 p_f_fertilizer_2007	
0.04872 22 tot_ndep_2000	53.43511 46 napi	
0.04641 15 npp_yrmean	53.43511 35 n_rock_2007	
0.04218 35 n_rock_2007	53.05344 23 tot_ndep_2007	
0.03531 29 n_fert_urban_2007	52.95802 14 npp	
0.03214 20 atmo_pdep_2002	52.48092 58 p2o5ws	
0.03157 16 precip	52.43321 11 fire_yrmean	
0.03119 46 napi	52.33779 10 fire	
0.02726 24 tot_sdep_2000	52.19466 49 damdensws	
0.02677 31 n_human_waste_2007	52.00382 52 runoffws	
0.02677 30 human_n_demand_2007	51.66985 6 lake_area_ha	
0.0265 41 p_nf_fertilizer_2007	51.52672 41 p_nf_fertilizer_2007	
0.0242 17 precip_yrmean	51.28817 2 wetlands	
0.0232 25 tot_sdep_2007	51.09733 15 npp_yrmean	
0.02064 43 p_human_nonfood_demand_kg_2007	51.04962 29 n_fert_urban_2007	
0.01975 44 p_human_waste_kg_2007	50.66794 24 tot_sdep_2000	
0.01972 42 p_human_food_demand_kg_2007	50.52481 43 p_human_nonfood_demand_kg_2007	
0.01969 51 pcthbwet2011ws	50.38168 16 precip	
0.01581 55 omws	50.04771 44 p_human_waste_kg_2007	
0.01463 49 damdensws	49.95229 55 omws	
0.01341 11 fire_yrmean	49.85687 50 pctwdwet2011ws	
0.01293 5 wsarea_km2	49.80916 42 p_human_food_demand_kg_2007	
0.00981 10 fire	49.37977 5 wsarea_km2	
0 50 pctwdwet2011ws	49.37977 31 n_human_waste_2007	
0 6 lake_area_ha	49.28435 30 human_n_demand_2007	
0 2 wetlands	49.0458 25 tot_sdep_2007	
0 58 p2o5ws	46.18321 17 precip_yrmean	

ptl, ntl, and *lon_dd* were all among the highest in each attribute selection algorithm. *ptl,* total phosphorus concentration, and *ntl*, total nitrogen concentration, intuitively would correlate with chlorophyll-a concentration as excessive algae growth can be caused by high nitrogen or phosphorus concentrations. Other attributes such as *snow_yrmean, n_fert_farm_2007, atmo_pdep_2002,* and *p_livestock_demand_2007,* all relate to sources of phosphorus or nitrogen as well. *lon_dd,* longitude, is slightly less intuitive, but we can infer that the surrounding climate and environment show similar trends for certain longitudes. Some climates are better for ecosystem health than others, with differences in temperatures and sources of runoff carrying nutrients, like phosphorus and nitrogen, for algal bloom.

6 Conclusion and Reproduction

The J48 model with Learner-based attribute selection gave us the most well-rounded model out of the 20 different models of this project. Using k-fold validation, we found that our model had a 78.01% average accuracy for ten folds. Although this is not a high accuracy, we are confident that our model could be useful in the environmental health sphere. The ability to predict high chlorophyll-*a* concentrations can help to warn environmentalists if there is a problem in the ecosystem in the form of excessive algae growth, hypoxia, or harmful algal bloom. These problems can be detrimental to the organisms living within and surrounding the lake ecosystem which is why it is important to take preventative measures towards ecosystems which exhibit certain significant attributes found by our models. For future work, we suggest compiling more recent data, as our dataset mainly contains data from the years 2002 and 2007. By using more recent data, our model will be able to capture the patterns found in the current ecosystem which is heavily influenced by climate change. We would also suggest using data with a more even class distribution, as our model is heavily right skewed with the majority of instances in the low concentration levels. Since we wish to predict high or medium concentrations to alert us of any concerns in the ecosystem, we would want more data on those classes to gain a better understanding of the patterns and trends associated with those concentration levels. We would also encourage exploring the combination of different attributes as many of the attributes in our dataset are concerned with phosphorus or nitrogen. Combining these could save both space and time while keeping a good amount of the information from the data.

Steps to Reproduce our J48 model with Learner-Based Selection:

1. Open *original_data.csv* in Google Sheets.
2. Under the Edit tab, select find and replace.
    a. In the Find box, type ‘ and type a space into the Replace with box.
    b. Select ‘This sheet’ or ‘All sheets’ in the Search drop down menu.
    c. Click Replace all, and repeat by replacing the following values with blank (leave the Replace box empty)
        i. “#NUM!”, “#DIV/0!”, “#VALUE!”, “#VALUE”, and “#NUM”
    d. Click Done
    e. On line 2,227 and the last column (BO), type -1 into the empty cell–this is necessary for weka to open the CSV file without error
3. Save as *replaced_data.csv*
4. Open Weka Explorer, and open *replaced_data.csv*.
    a. Click Edit.. to open the Viewer
    b. Scroll down to the final instance, right click, and delete the instance (should have the name DIPPER LAKE)
    c. Click OK
    d. Under Filter, click Choose, weka > filters > unsupervised > instance > RemoveWithValues
    e. Click on the horizontal bar with **RemoveWithValues**, set attributeIndex to last, matchMissingValues to True, and click OK
    f. Click Apply (the number of instances should have gone down to 2095)

5. In the Attributes box, select attributes *LAKENAME, Survey Number, SITE_ID, Year, Month, Day,* and *SNOW* (These attributes have little correlation with our class or do not have enough values)
    a. Click Remove
6. Discretize the data
    a. Under Filter, click Choose, weka > filters > unsupervised > attribute > Discretize
    b. Set attributeIndices to 8, bins to 3, and ignoreClass to True
    c. Click OK, and click Apply
7. Save the file as *discretized_data.arff*
8. Steps for normalization:
    a. For convenience, move the data to a .csv file.
    b. Open the .csv in *pandas*, then use the formulas for z-score normalization for those attributes that need to be z-score normalized (list in the *Preprocessing* section) or min-max normalization for all others.
        i. These formulas are as follows:
        ii. min-max: data[attr] = (data[attr] - data[attr].min()) / (data[attr].max() - data[attr].min())
        iii. Z-score: data[attr] = (data[attr] - data[attr].mean())/data[attr].std(ddof=0)
        iv. Attr here represents the attributes, looped using a simple "for attr in data".
    c. Save the file as *normalized_data.csv* using data.to_csv("normalized_data.csv")
9. In Weka, open the *normalized_data.csv* file
10. In the Select Attributes tab, under Attribute Evaluator click Choose > attributeSelection > WrapperSubsetEval
    a. Click on the horizontal bar with **WrapperSubsetEval** and click Choose next to classifier and select J48 under the trees folder
    b. Click OK
    c. Select Use full training set in Attribute Selection Mode
    d. Choose BestFirst for the Search Method
    e. Select (Nom) logchl_A as the class
    f. Click Start
11. Drop unwanted attributes
    a. Back in the preprocess tab, select attributes LON_DD, NTL, PTL, Atmo_Pdep_2002, P_livestock_demand_2007, P_livestock_production_2007, PctWdWet2011Ws, and logchl_A
    b. Click Invert, then click the Remove button at the bottom; you should be left with eight attributes including the class, logchl_A
12. Save the file as *selected_data.arff*
13. In the classify tab, choose J48 under the trees folder
    a. For test options, choose Cross-validation with 10 folds, or click Supplied test set and upload your own test set
    b. Set logchl_A as the class
    c. Press start
    d. Optional: Save the model to your local device by right clicking the result in the results list and selecting save model

7 Team Members and Tasks Performed

Jacob Dipasupil
- Dataset selection
- Attribute Selection
- Hand-picked attribute selection
- Classification
- Report writing
    - Section 1, 5, 6

Petr Kisselev
- Data Cleaning
- Attribute Selection
- Report writing
    - Section 2, 4, 7, 8
- Presentation Creation

Nikhil Alladi
- Preprocessing
- Data Cleaning
- Attribute Selection
- Report writing
    - Section 3, 4
- Presentation Creation

8 Appendix and Sources

Certain steps of data cleaning were performed through the use of Python and the Pandas library on Jupyter Notebooks.

Weka was used for all classification and attribute selection.

8.1 Citations

[1] Data source website: Estimates of lake nitrogen, phosphorus, and chlorophyll-a concentrations to characterize harmful algal bloom risk across the United States
[2] https://weka.sourceforge.io/doc.dev/weka/attributeSelection/package-summary.html
[3] https://weka.sourceforge.io/doc.dev/weka/classifiers/package-summary.html
[4] https://weka.sourceforge.io/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html
[5] https://weka.sourceforge.io/doc.dev/weka/attributeSelection/OneRAttributeEval.html
[6] https://weka.sourceforge.io/doc.dev/weka/attributeSelection/PrincipalComponents.html
[7] https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/J48.html
[8] https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/RandomTree.html
[9] https://weka.sourceforge.io/doc.dev/weka/classifiers/bayes/NaiveBayes.html
[10] https://weka.sourceforge.io/doc.dev/weka/classifiers/functions/Logistic.html