

DATS 6103 – Intro to Data Mining

Individual Final Report
Group – 1
Dec 7, 2021
Aasish Kumar Immadisetty

Table of Contents

1. Introduction:	2
2. Personal Contribution:	3
Code:	3
Pre-processing:	3
EDA Analysis:	3
Model Building:	3
Support Vector Machine:	3
Presentation:	3
Group Final Report:	3
3. Personal Contribution in detail:	4
Model Building (PYQT5 code):	6
4. Results:	10
Histogram:	10
Model Building:	10
Support Vector Machine (SVM):	10
5. Summary and Conclusion:	12
Summary:	12
SVM Model:	12
Conclusion:	12
6. Percent of Code:	12
7. References:	13

1. Introduction:

Public health has an important role in promoting the population's welfare, maintaining its security, and protecting it from infectious disease and environmental risks, and assisting in assuring the population's safety and quality treatment. The Internet is transforming business, education, government, healthcare, and even how we communicate with our loved ones on a daily, and it has emerged as one of the most important drivers of social change. People have started using the incredible internet and started online gaming, spending a lot of time on it because of the internet's evolution. People's mental and psychological health is being harmed by online gaming, which is leading to a variety of mental diseases. This dataset contains information gathered as part of a global survey of gamers. The survey asked questions that psychologists typically ask persons with anxiety, social phobia, and low to no life satisfaction. The questionnaire is made up of a series of questions that would be asked in a psychological investigation.

In analyzing the influence of online gaming on General Anxiety Disorder, we will use different features from the models, such as GAD, Game, Playstyle, Platform on which they play, Age, Gender, Hours, Work, and Residence. The project is illustrated by developing a GUI-based application that displays the end-to-end modeling utilizing three machine learning algorithms: Random Forest Classifier, Decision Tree, and Support Vector Machine, respectively.

Cleaning the dataset, exploratory data analysis, preprocessing, modeling, model comparison, GUI creation, preparing a power point presentation, writing the group report, and generating a demo of the GUI were all part of the shared effort.

2. Personal Contribution:

Code:

Pre-processing:

- Heat map was plotted to see whether there was a correlation between different features and to narrow down the dataset to only the features that were required.
- Because most of the variables with missing values are categorical data, I filled in the blank values using the most often occurring phrase in that column.
- The z-score is used to standardize data to improve the quality of the data in the dataset and to remove outliers.
- To convert multi-class labels to binary labels, the target variable was label binarized, which makes learning one regressor or binary classifier per class much easier.
- Divide the data set into two sections: training and testing.

EDA Analysis:

- Built scatter plot for variables Hours and Age to understand the relation between the variables.

Model Building:

- For classification and regression, created a Support Vector Machine model.

Support Vector Machine:

- SVMs (support vector machines) are a class of supervised learning algorithms for classification and regression. It is, however, mostly employed to solve categorization difficulties. The decision function of an SVM is based on a collection of training data. It also enables a 'one-versus-one' technique for multi-class classification, in which the receiver operator characteristic for each class is calculated separately.
- The confusion matrix, classification report, accuracy, ROC curve, and Area under the Curve were calculated for the Support Vector Machine.

Presentation:

- Assisted with the preparation of presentations to explain the EDA Analysis and SVM Model.

Group Final Report:

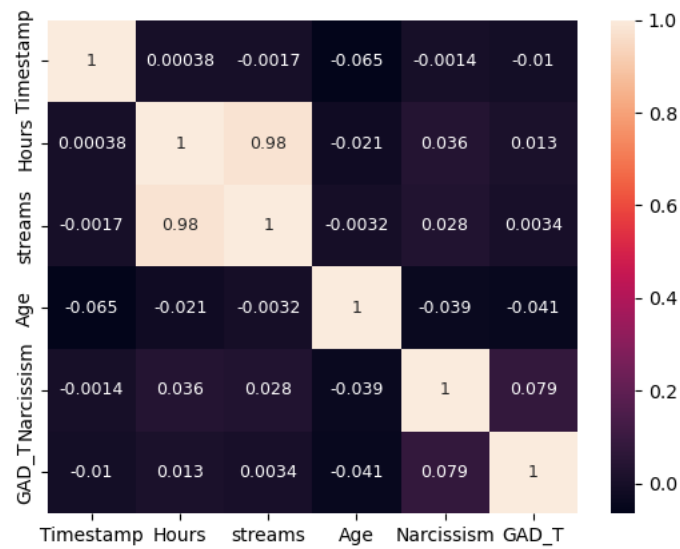
- I worked on a few aspects of data pre-processing, EDA analysis, and the outputs of the Support Vector Machine (SVM) model.

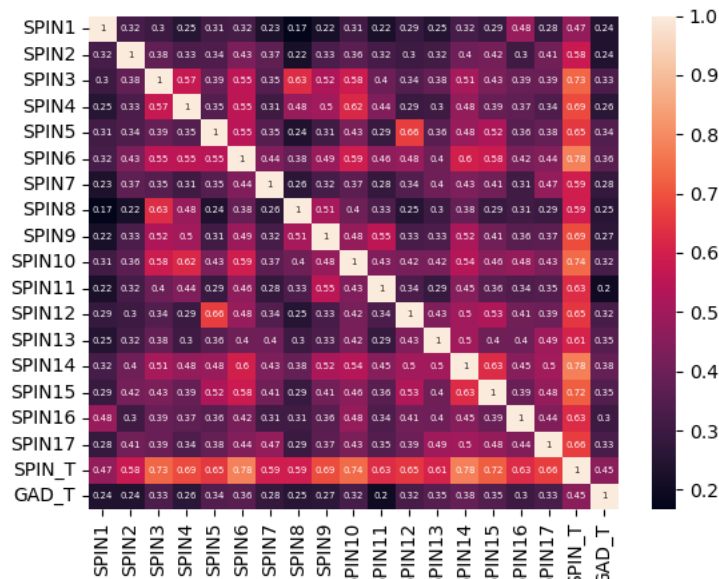
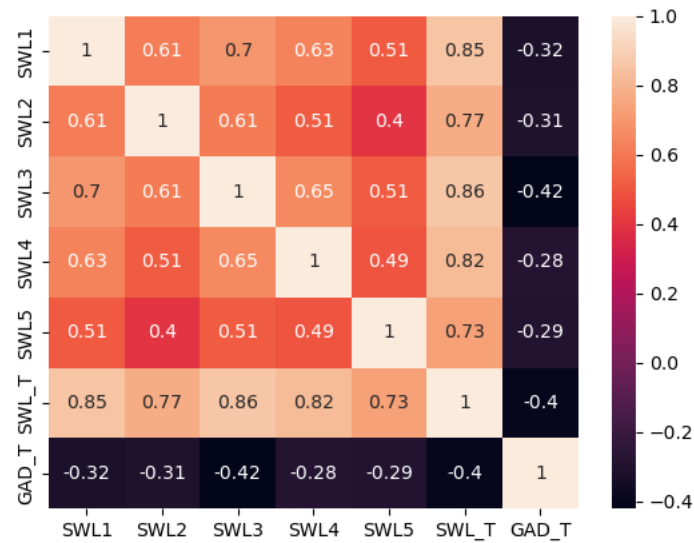
3. Personal Contribution in detail:

- Because most of the variables with missing values are categorical data, I filled in the blank values using the most often occurring phrase in that column

```
df = df.apply(lambda x: x.fillna(x.value_counts().index[0]))
```

- Heat map was plotted to see whether there was a correlation between different features and to narrow down the dataset to only the features that were required.





- The z-score is used to standardize data to improve the quality of the data in the dataset and to remove outliers.

```
df = df[(-3 < zscore(df['Hours'])) & (zscore(df['Hours']) < 3)]
df = df[(-3 < zscore(df['Age'])) & (zscore(df['Age']) < 3)]
df = df[(-3 < zscore(df['GAD_T'])) & (zscore(df['GAD_T']) < 3)]
df = df[(-3 < zscore(df['SWL_T'])) & (zscore(df['SWL_T']) < 3)]
```

- Divide the data set into two sections: training and testing.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)

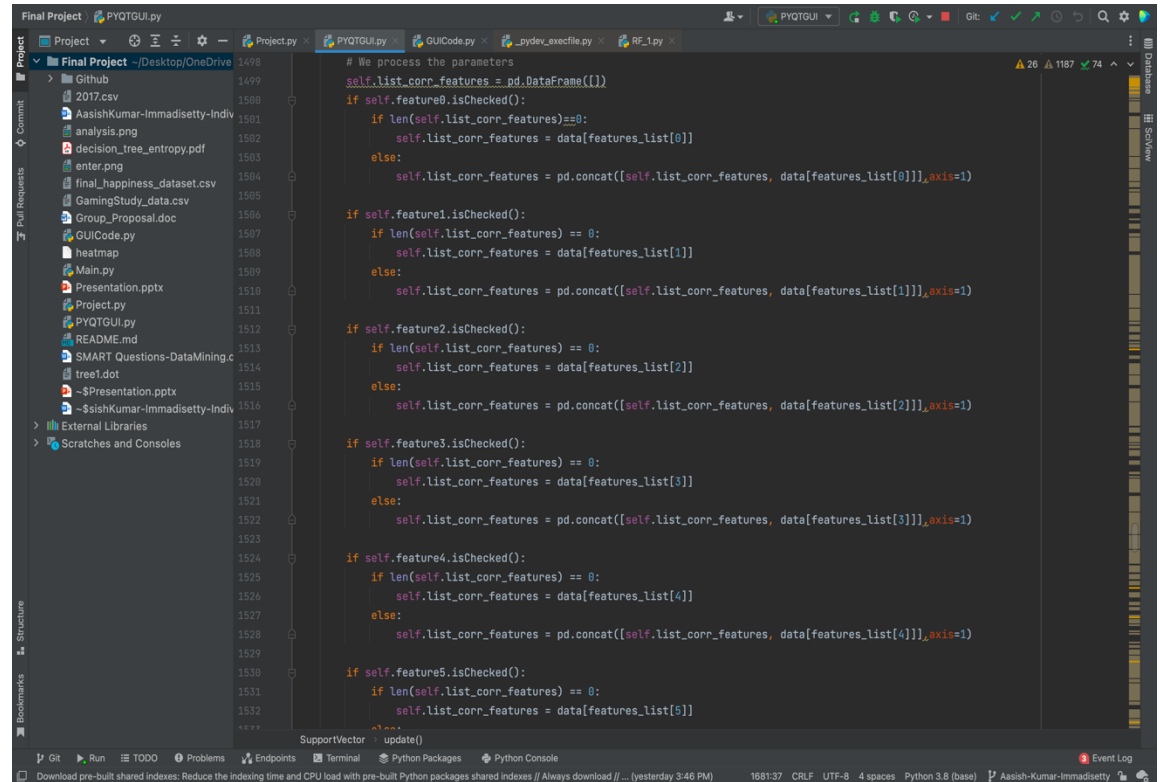
x_train1, x_test1, y_train1, y_test1 = train_test_split(x, y1, test_size=0.3, random_state=1)
```

- To convert multi-class labels to binary labels, the target variable was label binarized, which makes learning one regressor or binary classifier per class much easier.

```
y1 = label_binarize(y, classes=[0, 1, 2])
```

Model Building (PYQT5 code):

- For classification and regression, created a Support Vector Machine model. The confusion matrix, classification report, accuracy, ROC curve, and Area under the Curve were calculated for the Support Vector Machine.



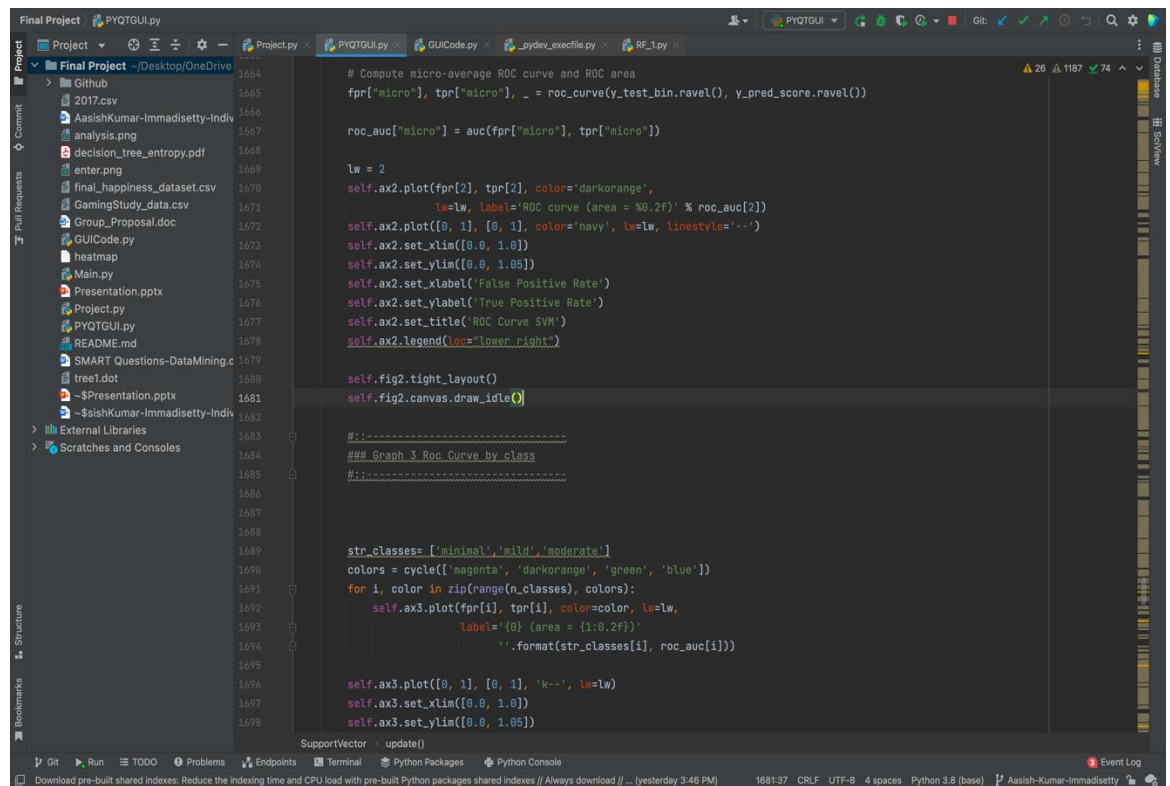
```
Final Project - PYQTGUI.py
Project
  Final Project - Desktop/OneDrive
    2017.csv
    AasishKumar-Immadisetty-Indiv
    analysis.png
    decision_tree_entropy.pdf
    enter.png
    final_happiness_dataset.csv
    GamingStudy_data.csv
    Group_Proposal.doc
    GUICode.py
    heatmap
    Main.py
    Presentation.pptx
    Project.py
    PYQTGUI.py
    README.md
    SMART Questions-DataMining.c
    tree1.dot
    ~$Presentation.pptx
    ~$sishKumar-Immadisetty-Indiv
  External Libraries
  Scratches and Consoles
  PYQTGUI.py
    1533
    1534
    1535
    1536
    1537
    1538
    1539
    1540
    1541
    1542
    1543
    1544
    1545
    1546
    1547
    1548
    1549
    1550
    1551
    1552
    1553
    1554
    1555
    1556
    1557
    1558
    1559
    1560
    1561
    1562
    1563
    1564
    1565
    1566
    1567
    1568
    1569
    1570
    1571
    1572
    1573
    1574
    1575
    1576
    1577
    1578
    1579
    1580
    1581
    1582
    1583
    1584
    1585
    1586
    1587
    1588
    1589
    1590
    1591
    1592
    1593
    1594
    1595
    1596
    1597
    1598
    1599
    1600
    1601
    1602
    1603
    1604
    1605
    1606
    1607
    1608
    1609
    1610
    1611
    1612
    1613
    1614
    1615
    1616
    1617
    1618
    1619
    1620
    1621
    1622
    1623
    1624
    1625
    1626
    1627
    1628
    1629
    1630
    1631
    1632
    1633
    1634
    1635
    1636
    1637
    1638
    1639
    1640
    1641
    1642
    1643
    1644
    1645
    1646
    1647
    1648
    1649
    1650
    1651
    1652
    1653
    1654
    1655
    1656
    1657
    1658
    1659
    1660
    1661
    1662
    1663
    1664
    1665
    1666
    1667
    1668
    1669
    1670
    1671
    1672
    1673
    1674
    1675
    1676
    1677
    1678
    1679
    1680
    1681
    1682
    1683
    1684
    1685
    1686
    1687
    1688
    1689
    1690
    1691
    1692
    1693
    1694
    1695
    1696
    1697
    1698
    1699
    1700
    1701
    1702
    1703
    1704
    1705
    1706
    1707
    1708
    1709
    1710
    1711
    1712
    1713
    1714
    1715
    1716
    1717
    1718
    1719
    1720
    1721
    1722
    1723
    1724
    1725
    1726
    1727
    1728
    1729
    1730
    1731
    1732
    1733
    1734
    1735
    1736
    1737
    1738
    1739
    1740
    1741
    1742
    1743
    1744
    1745
    1746
    1747
    1748
    1749
    1750
    1751
    1752
    1753
    1754
    1755
    1756
    1757
    1758
    1759
    1760
    1761
    1762
    1763
    1764
    1765
    1766
    1767
    1768
    1769
    1770
    1771
    1772
    1773
    1774
    1775
    1776
    1777
    1778
    1779
    1780
    1781
    1782
    1783
    1784
    1785
    1786
    1787
    1788
    1789
    1790
    1791
    1792
    1793
    1794
    1795
    1796
    1797
    1798
    1799
    1800
    1801
    1802
    1803
    1804
    1805
    1806
    1807
    1808
    1809
    1810
    1811
    1812
    1813
    1814
    1815
    1816
    1817
    1818
    1819
    1820
    1821
    1822
    1823
    1824
    1825
    1826
    1827
    1828
    1829
    1830
    1831
    1832
    1833
    1834
    1835
    1836
    1837
    1838
    1839
    1840
    1841
    1842
    1843
    1844
    1845
    1846
    1847
    1848
    1849
    1850
    1851
    1852
    1853
    1854
    1855
    1856
    1857
    1858
    1859
    1860
    1861
    1862
    1863
    1864
    1865
    1866
    1867
    1868
    1869
    1870
    1871
    1872
    1873
    1874
    1875
    1876
    1877
    1878
    1879
    1880
    1881
    1882
    1883
    1884
    1885
    1886
    1887
    1888
    1889
    1890
    1891
    1892
    1893
    1894
    1895
    1896
    1897
    1898
    1899
    1900
    1901
    1902
    1903
    1904
    1905
    1906
    1907
    1908
    1909
    1910
    1911
    1912
    1913
    1914
    1915
    1916
    1917
    1918
    1919
    1920
    1921
    1922
    1923
    1924
    1925
    1926
    1927
    1928
    1929
    1930
    1931
    1932
    1933
    1934
    1935
    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
    1989
    1990
    1991
    1992
    1993
    1994
    1995
    1996
    1997
    1998
    1999
    2000
    2001
    2002
    2003
    2004
    2005
    2006
    2007
    2008
    2009
    2010
    2011
    2012
    2013
    2014
    2015
    2016
    2017
    2018
    2019
    2020
    2021
    2022
    2023
    2024
    2025
    2026
    2027
    2028
    2029
    2030
    2031
    2032
    2033
    2034
    2035
    2036
    2037
    2038
    2039
    2040
    2041
    2042
    2043
    2044
    2045
    2046
    2047
    2048
    2049
    2050
    2051
    2052
    2053
    2054
    2055
    2056
    2057
    2058
    2059
    2060
    2061
    2062
    2063
    2064
    2065
    2066
    2067
    2068
    2069
    2070
    2071
    2072
    2073
    2074
    2075
    2076
    2077
    2078
    2079
    2080
    2081
    2082
    2083
    2084
    2085
    2086
    2087
    2088
    2089
    2090
    2091
    2092
    2093
    2094
    2095
    2096
    2097
    2098
    2099
    2100
    2101
    2102
    2103
    2104
    2105
    2106
    2107
    2108
    2109
    2110
    2111
    2112
    2113
    2114
    2115
    2116
    2117
    2118
    2119
    2120
    2121
    2122
    2123
    2124
    2125
    2126
    2127
    2128
    2129
    2130
    2131
    2132
    2133
    2134
    2135
    2136
    2137
    2138
    2139
    2140
    2141
    2142
    2143
    2144
    2145
    2146
    2147
    2148
    2149
    2150
    2151
    2152
    2153
    2154
    2155
    2156
    2157
    2158
    2159
    2160
    2161
    2162
    2163
    2164
    2165
    2166
    2167
    2168
    2169
    2170
    2171
    2172
    2173
    2174
    2175
    2176
    2177
    2178
    2179
    2180
    2181
    2182
    2183
    2184
    2185
    2186
    2187
    2188
    2189
    2190
    2191
    2192
    2193
    2194
    2195
    2196
    2197
    2198
    2199
    2200
    2201
    2202
    2203
    2204
    2205
    2206
    2207
    2208
    2209
    2210
    2211
    2212
    2213
    2214
    2215
    2216
    2217
    2218
    2219
    2220
    2221
    2222
    2223
    2224
    2225
    2226
    2227
    2228
    2229
    2230
    2231
    2232
    2233
    2234
    2235
    2236
    2237
    2238
    2239
    2240
    2241
    2242
    2243
    2244
    2245
    2246
    2247
    2248
    2249
    2250
    2251
    2252
    2253
    2254
    2255
    2256
    2257
    2258
    2259
    2260
    2261
    2262
    2263
    2264
    2265
    2266
    2267
    2268
    2269
    2270
    2271
    2272
    2273
    2274
    2275
    2276
    2277
    2278
    2279
    2280
    2281
    2282
    2283
    2284
    2285
    2286
    2287
    2288
    2289
    2290
    2291
    2292
    2293
    2294
    2295
    2296
    2297
    2298
    2299
    2300
    2301
    2302
    2303
    2304
    2305
    2306
    2307
    2308
    2309
    2310
    2311
    2312
    2313
    2314
    2315
    2316
    2317
    2318
    2319
    2320
    2321
    2322
    2323
    2324
    2325
    2326
    2327
    2328
    2329
    2330
    2331
    2332
    2333
    2334
    2335
    2336
    2337
    2338
    2339
    2340
    2341
    2342
    2343
    2344
    2345
    2346
    2347
    2348
    2349
    2350
    2351
    2352
    2353
    2354
    2355
    2356
    2357
    2358
    2359
    2360
    2361
    2362
    2363
    2364
    2365
    2366
    2367
    2368
    2369
    2370
    2371
    2372
    2373
    2374
    2375
    2376
    2377
    2378
    2379
    2380
    2381
    2382
    2383
    2384
    2385
    2386
    2387
    2388
    2389
    2390
    2391
    2392
    2393
    2394
    2395
    2396
    2397
    2398
    2399
    2400
    2401
    2402
    2403
    2404
    2405
    2406
    2407
    2408
    2409
    2410
    2411
    2412
    2413
    2414
    2415
    2416
    2417
    2418
    2419
    2420
    2421
    2422
    2423
    2424
    2425
    2426
    2427
    2428
    2429
    2430
    2431
    2432
    2433
    2434
    2435
    2436
    2437
    2438
    2439
    2440
    2441
    2442
    2443
    2444
    2445
    2446
    2447
    2448
    2449
    2450
    2451
    2452
    2453
    2454
    2455
    2456
    2457
    2458
    2459
    2460
    2461
    2462
    2463
    2464
    2465
    2466
    2467
    2468
    2469
    2470
    2471
    2472
    2473
    2474
    2475
    2476
    2477
    2478
    2479
    2480
    2481
    2482
    2483
    2484
    2485
    2486
    2487
    2488
    2489
    2490
    2491
    2492
    2493
    2494
    2495
    2496
    2497
    2498
    2499
    2500
    2501
    2502
    2503
    2504
    2505
    2506
    2507
    2508
    2509
    2510
    2511
    2512
    2513
    2514
    2515
    2516
    2517
    2518
    2519
    2520
    2521
    2522
    2523
    2524
    2525
    2526
    2527
    2528
    2529
    2530
    2531
    2532
    2533
    2534
    2535
    2536
    2537
    2538
    2539
    2540
    2541
    2542
    2543
    2544
    2545
    2546
    2547
    2548
    2549
    2550
    2551
    2552
    2553
    2554
    2555
    2556
    2557
    2558
    2559
    2560
    2561
    2562
    2563
    2564
    2565
    2566
    2567
    2568
    2569
    2570
    2571
    2572
    2573
    2574
    2575
    2576
    2577
    2578
    2579
    2580
    2581
    2582
    2583
    2584
    2585
    2586
    2587
    2588
    2589
    2590
    2591
    2592
    2593
    2594
    2595
    2596
    2597
    2598
    2599
    2600
    2601
    2602
    2603
    2604
    2605
    2606
    2607
    2608
    2609
    2610
    2611
    2612
    2613
    2614
    2615
    2616
    2617
    2618
    2619
    2620
    2621
    2622
    2623
    2624
    2625
    2626
    2627
    2628
    2629
    2630
    2631
    2632
    2633
    2634
    2635
    2636
    2637
    2638
    2639
    2640
    2641
    2642
    2643
    2644
    2645
    2646
    2647
    2648
    2649
    2650
    2651
    2652
    2653
    2654
    2655
    2656
    2657
    2658
    2659
    2660
    2661
    2662
    2663
    2664
    2665
    2666
    2667
    2668
    2669
    2670
    2671
    2672
    2673
    2674
    2675
    2676
    2677
    2678
    2679
    2680
    2681
    2682
    2683
    2684
    2685
    2686
    2687
    2688
    2689
    2690
    2691
    2692
    2693
    2694
    2695
    2696
    2697
    2698
    2699
    2700
    2701
    2702
    2703
    2704
    2705
    2706
    2707
    2708
    2709
    2710
    2711
    2712
    2713
    2714
    2715
    2716
    2717
    2718
    2719
    2720
    2721
    2722
    2723
    2724
    2725
    2726
    2727
    2728
    2729
    2730
    2731
    2732
    2733
    2734
    2735
    2736
    2737
    2738
    2739
    2740
    2741
    2742
    2743
    2744
    2745
    2746
    2747
    2748
    2749
    2750
    2751
    2752
    2753
    2754
    2755
    2756
    2757
    2758
    2759
    2760
    2761
    2762
    2763
    2764
    2765
    2766
    2767
    2768
    2769
    2770
    2771
    2772
    2773
    2774
    2775
    2776
    2777
    2778
    2779
    2780
    2781
    2782
    2783
    2784
    2785
    2786
    2787
    2788
    2789
    2790
    2791
    2792
    2793
    2794
    2795
    2796
    2797
    2798
    2799
    2800
    2801
    2802
    2803
    2804
    2805
    2806
    2807
    2808
    2809
    2810
    2811
    2812
    2813
    2814
    2815
    2816
    2817
    2818
    2819
    2820
    2821
    2822
    2823
    2824
    2825
    2826
    2827
    2828
    2829
    2830
    2831
    2832
    2833
    2834
    2835
    2836
    2837
    2838
    2839
    2840
    2841
    2842
    2843
    2844
    2845
    2846
    2847
    2848
    2849
    2850
    2851
    2852
    2853
    2854
    2855
    2856
    2857
    2858
    2859
    2860
    2861
    2862
    2863
    2864
    2865
    2866
    2867
    2868
    2869
    2870
    2871
    2872
    2873
    2874
    2875
    2876
    2877
    2878
    2879
    2880
    2881
    2882
    2883
    2884
    2885
    2886
    2887
    2888
    2889
    2890
    2891
    2892
    2893
    2894
    2895
    2896
    2897
    2898
    2899
    2900
    2901
    2902
    2903
    2904
    2905
    2906
    2907
    2908
    2909
    2910
    2911
    2912
    2913
    2914
    2915
    2916
    2917
    2918
    2919
    2920
    2921
    2922
    2923
    2924
    2925
    2926
    2927
    2928
    2929
    2930
    2931
    2932
    2933
    2934
    2935
    2936
    2937
    2938
    2939
    2940
    2941
    2942
    2943
    2944
    2945
    2946
    2947
    2948
    2949
    2950
    2951
    2952
    2953
    2954
    2955
    2956
    2957
    2958
    2959
    2960
    2961
    2962
    2963
    2964
    2965
    2966
    2967
    2968
    2969
    2970
    2971
    2972
    2973
    2974
    2975
    2976
    2977
    2978
    2979
    2980
    2981
    2982
    2983
    2984
    2985
    2986
    2987
    2988
    2989
    2990
    2991
    2992
    2993
    2994
    2995
    2996
    2997
    2998
    2999
    3000
    3001
    3002
    3003
    3004
    3005
    3006
    3007
    3008
    3009
    3010
    3011
    3012
    3013
    3014
    3015
    3016
    3017
    3018
    3019
    3020
    3021
    3022
    3023
    3024
    3025
    3026
    3027
    3028
    3029
    3030
    3031
    3032
    3033
    3034
    3035
    3036
    3037
    3038
    3039
    3040
    3041
    3042
    3043
    3044
    3045
    3046
    3047
    3048
    3049
    3050
    3051
    3052
    3053
    3054
    3055
    3056
    3057
    3058
    3059
    3060
    3061
    3062
    3063
    3064
    3065
    3066
    3067
    3068
    3069
    3070
    3071
    3072
    3073
    3074
    3075
    3076
    3077
    3078
    3079
    3080
    3081
    3082
    3083
    3084
    3085
    3086
    3087
    3088
    3089
    3090
    3091
    3092
    3093
    3094
    3095
    3096
    3097
    3098
    3099
    3100
    3101
    3102
    3103
    3104
    3105
    3106
    3107
    3108
    3109
    3110
    3111
    3112
    3113
    3114
    3115
    3116
    3117
    3118
    3119
    3120
    3121
    3122
    3123
    3124
    3125
    3126
    3127
    3128
    3129
    3130
    3131
    3132
    3133
    3134
    3135
    3136
    3137
    3138
    3139
    3140
    3141
    3142
    3143
    3144
    3145
    3146
    3147
    3148
    3149
    3150
    3151
    3152
    3153
    3154
    3155
    3156
    3157
    3158
    3159
    3160
    3161
    3162
    3163
    3164
    3165
    3166
    3167
    3168
    3169
    3170
    3171
    3172
    3173
    3174
    3175
    3176
    3177
    3178
    3179
    3180
    3181
    3182
    3183
    3184
    3185
    3186
    3187
    3188
    3189
    3190
    3191
    3192
    3193
    3194
    3195
    3196
    3197
    3198
    3199
    3200
    3201
    3202
    3203
    3204
    3205
    3206
    3207
    3208
    3209
    3210
    3211
    3212
    3213
    3214
    3215
    3216
    3217
    3218
    3219
    3220
    3221
    3222
    3223
    3224
    3225
    3226
    3227
    3228
    3229
    3230
    3231
    3232
    3233
    3234
    3235
    3236
    3237
    3238
    3239
    3240
    3241
    3242
    3243
    3244
    3245
    3246
    3247
    3248
    3249
    3250
    3251
    3252
    3253
    3254
    3255
    3256
    3257
    3258
    3259
    3260
    3261
    3262
    3263
    3264
    3265
    3266
    3267
    3268
    3269
    3270
    3271
    3272
    3273
    327
```

- The three images above show how data is put into X and Y variables for training, testing, and calculating roc, as well as visualizing roc, using Label Encoder and Label Binarizer.

```

1627 # Graph1 -- Confusion Matrix
1630 #-----
1631 class_names1 = [0, 1, 2]
1632
1633 self.ax1.imshow(conf_matrix, cmap=plt.cm.get_cmap('Blues', 14))
1634 self.ax1.set_ylabel(class_names1)
1635 self.ax1.set_xlabel(class_names1)
1636
1637 self.ax1.set_xlabel('Predicted label')
1638 self.ax1.set_ylabel('True label')
1639
1640
1641 for i in range(len(class_names1)):
1642     for j in range(len(class_names1)):
1643         y_pred_score = self.clf_entropy.decision_function(X_test)
1644         self.ax1.text(j, i, str(conf_matrix[i][j]))
1645
1646 self.fig.tight_layout()
1647 self.fig.canvas.draw_idle()
1648
1649 #-----
1650 # Graph 2 -- ROC Curve
1651 #-----
1652
1653 y_test_bin = label_binarize(y_test, classes=[0, 1, 2])
1654 n_classes = y_test_bin.shape[1]
1655
1656 fpr = dict()
1657 tpr = dict()
1658 roc_auc = dict()
1659 for i in range(n_classes):
1660     fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_pred_score[:, i])
1661     roc_auc[i] = auc(fpr[i], tpr[i])
1662
1663 # Compute micro-average ROC curve and ROC area
1664 SupportVector.update()
  
```

- The code in the image above can be used to generate a confusion matrix from the features chosen and plot the roc curve.



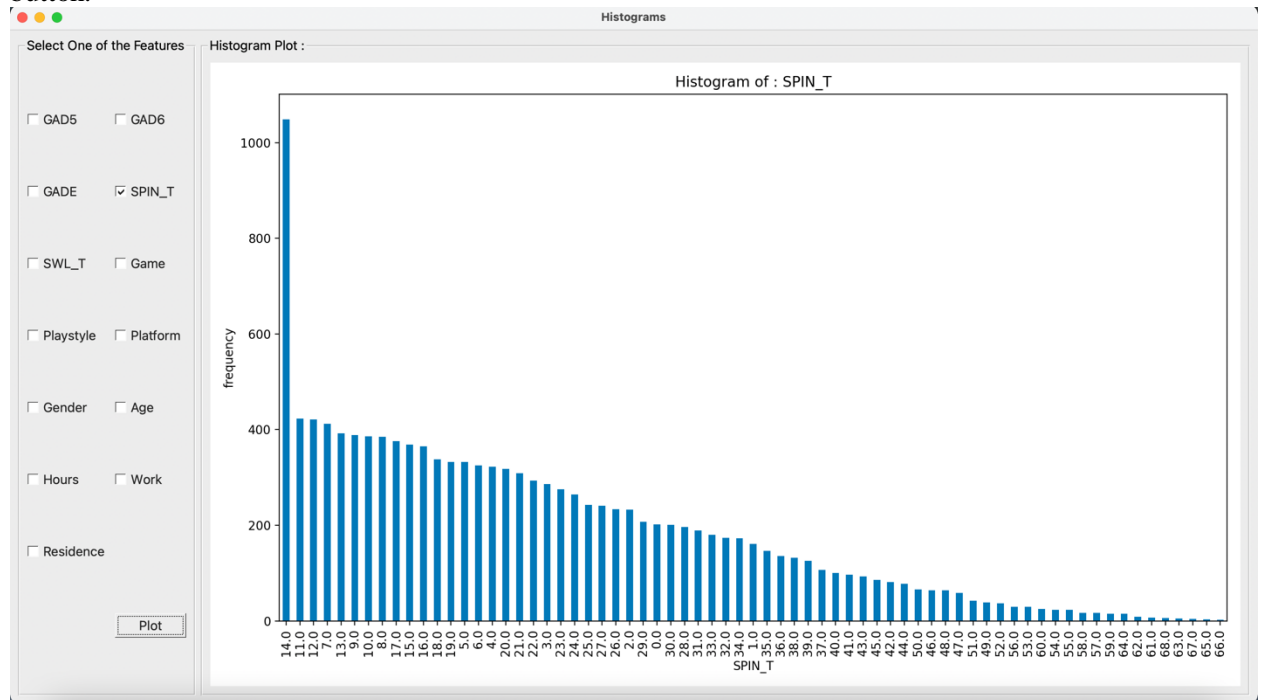
```
1664 # Compute micro-average ROC curve and ROC area
1665 fpr["micro"], tpr["micro"], _ = roc_curve(y_test_bin.ravel(), y_pred_score.ravel())
1666
1667 roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])
1668
1669 lw = 2
1670 self.ax2.plot(fpr[2], tpr[2], color='darkorange',
1671              lw=lw, label='ROC curve (area = %0.2f)' % roc_auc[2])
1672 self.ax2.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
1673 self.ax2.set_xlim([0.0, 1.0])
1674 self.ax2.set_ylim([0.0, 1.05])
1675 self.ax2.set_xlabel('False Positive Rate')
1676 self.ax2.set_ylabel('True Positive Rate')
1677 self.ax2.set_title('ROC Curve SVM')
1678 self.ax2.legend(loc="lower right")
1679
1680 self.fig2.tight_layout()
1681 self.fig2.canvas.draw_idle()
1682
1683 #!-----
1684 ### Graph 3 Roc_Curve_by_class
1685 #!-----
1686
1687
1688
1689 str_classes= ['minimal', 'mild', 'moderate']
1690 colors = cycle(['magenta', 'darkorange', 'green', 'blue'])
1691 for i, color in zip(range(n_classes), colors):
1692     self.ax3.plot(fpr[i], tpr[i], color=color, lw=lw,
1693                 label='{0} (area = {1:0.2f})'
1694                 ''.format(str_classes[i], roc_auc[i]))
1695
1696 self.ax3.plot([0, 1], [0, 1], 'k--', lw=lw)
1697 self.ax3.set_xlim([0.0, 1.0])
1698 self.ax3.set_ylim([0.0, 1.05])
1699
1700 SupportVector : update()
```

Because our target variable is multi-class, the above code will assist in plotting the roc curve for each class.

4. Results:

Histogram:

The graphic depicts the pictorial representation of the histogram as well as the numerous features available on the left grid, and the histogram is presented on the canvas after pressing the plot button.

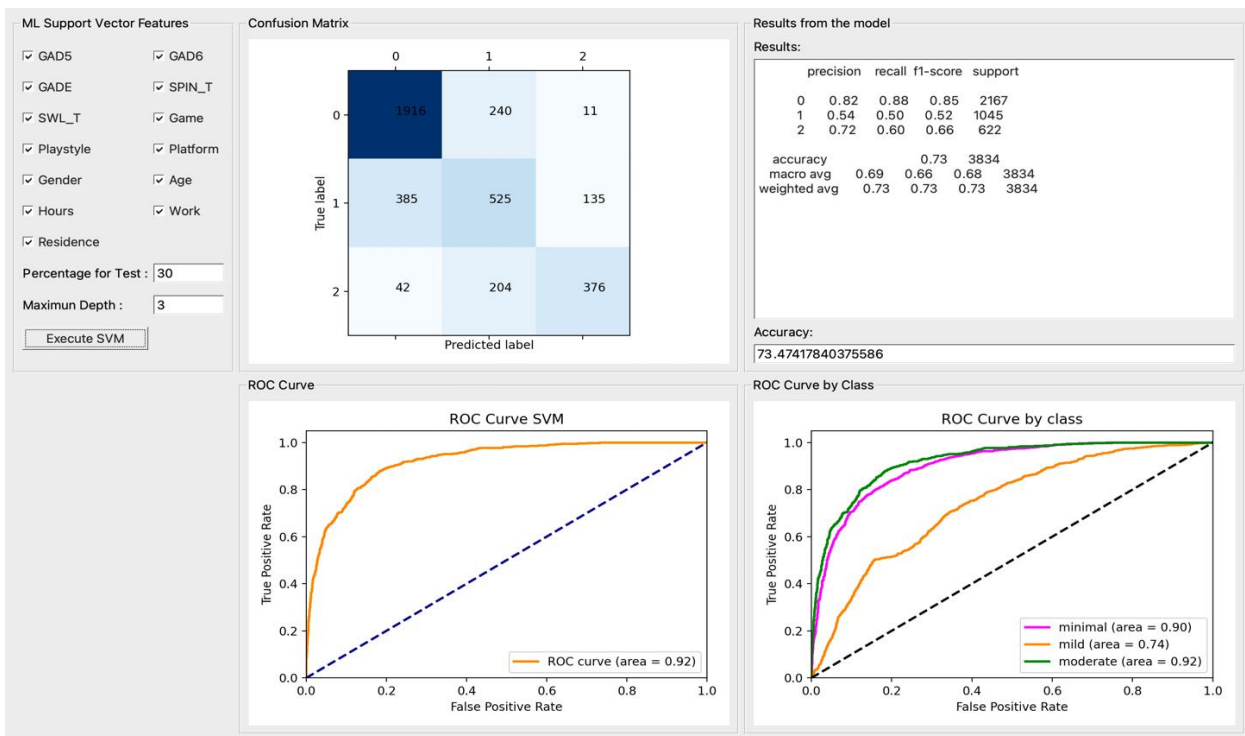


Model Building:

To display the roc curve and calculate the score, we first select all X-variables that are not target variables and load the target variable Y, which is the target variable and is label binarized as well as label encoded to convert labels to numeric values.

Support Vector Machine (SVM):

The image displays the SVM dashboard, and the user can manually change the percentage for test. The features can be selected as per user's choice and click on "Execute SVM" to run the model which displays confusion matrix, results from the model, ROC curve, ROC curve by each class.



5. Summary and Conclusion:

Summary:

SVM Model:

- Accuracy of SVM Model, Accuracy = 73.47
- From the classification report of SVM model:
 - F1 score for 0's: 0.85, 1's: 0.52, 2's: 0.66
 - Precision for 0's: 0.82, 1's: 0.54, 2's: 0.72
 - Recall for 0's: 0.88, 1's: 0.50, 2's: 0.60
- Area under curve for SVM Model is 0.92. An ideal model should have AUC value above 0.8 which is why we can say model is decent.
- The features with high importance are:

'GAD5', 'GAD6', 'GADE', 'SPIN_T', 'SWL_T', 'Game', 'Playstyle', 'Platform',
'Gender', 'Age', 'Hours', 'Work', 'Residence'

Conclusion:

- Since SVM has the highest accuracy and precision of the three possible classifiers, the Support Vector Machine is the best model. It also has the highest AUC of the three classifiers. In addition, we developed the model with the highest importance of features, making it the most efficient of the three classifiers in terms of accuracy.
- Overall, we can say that the Support Vector Machine model can accurately predict how online gaming will influence Generalized Anxiety Disorder with 73.47 percent accuracy.
- Using the dataset collected, we can investigate how online gaming affects other illnesses in the future.

6. Percent of Code:

The internet is utilized for only reference and none of the code is copied. Searching for Syntaxes to apply a function was done, but it was never downloaded from the internet. All of the code was written by hand, using skills learned through Prof. Amir Jafari's Lectures and GitHub exercises.

7. References:

- <https://numpy.org/doc/>
- <https://matplotlib.org/stable/users/index.html>
- <https://matplotlibguide.readthedocs.io/en/latest/>
- https://sklearn.org/user_guide.html
- <https://www.javatpoint.com/>