

PROJECT PHASE -2

A Predictive Assessment System for Diabetes Risk Factors

Mounika Pasupuleti
Engineering Science Data Science
University at Buffalo
Buffalo, New York
mpasupul@buffalo.edu

Sahithya Arveti Nagaraju
Engineering Science Data Science
University at Buffalo
Buffalo, New York
sarvetin@buffalo.edu

Sushmitha Manjunatha
Engineering Science Data Science
University at Buffalo
Buffalo, New York
smanjuna@buffalo.edu

Abstract—Diabetes has rapidly transformed from a relatively uncommon disease among a small number of people to a serious disease of global proportions affecting every age group. Diabetes Pathophysiology may be defined as a disease where the sugar levels in the blood are not well regulated leading to substantially deteriorated health and shortened life expectancy. In many cases treatment of diabetes comes with high costs which is an additional burden on the affected individual.

The goal of this project is to investigate the most common correlation with diabetes, factors which help in understanding possible causative agents and their potential risk. This analysis is expected to aid in prevention of diabetes cases and promote healthy behavior.

In this regard, we expect to evaluate how these basic attributes such as work performance, nutrition, and health history could be used to estimate the probability of an individual being at risk of diabetes by employing complex machine learning models and analyzing data from health activities. This contribution is crucial as it addresses the growing health challenges like diabetes by providing valuable insights that enhances overall community health.

Index Terms: Diabetes, sugar levels, lifestyle, machine learning models.

I. DATA SOURCES

The dataset is taken from the Kaggle website. Below is the link:

https://www.kaggle.com/datasets/prosperchuks/health-dataset?select=diabetes_data.csv

The dataset consists of 70,707 rows and 19 columns in which there are 10 numeric features and 9 categorical features. Numeric features in our dataset are:

- Age
- HighChol
- CholCheck
- BMI
- HeartDiseaseorAttack
- PhysActivity
- HvyAlcoholConsump
- MentHlth
- PhyHlth
- HighBP

Categorical features in our dataset are:

- Sex
- Smoker
- Fruits
- Veggies
- GenHlth
- DiffWalk
- Stroke
- SugarConsumption
- Diabetes

Observe the data in the dataset:

```
[2] import pandas as pd
import numpy as np
data = pd.read_csv("impure_dataset_final.csv");
data.head(5)
```

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Ve
0	4	Male	0	1	26	Non Smoker	0	1.0	Does not Eat	
1	12	Male	1	1	26	Smoker	0	0.0	Eat	
2	13	Male	0	1	26	Non Smoker	0	1.0	Eat	
3	11	Male	1	1	28	Smoker	0	1.0	Eat	
4	8	Female	0	1	29	Smoker	0	1.0	Eat	

```
[ ] data.shape
```

```
(70707, 19)
```

```
[ ] dataset_info = {
    "Number of rows in the dataset are": data.shape[0],
    "Number of columns in the dataset are ": data.shape[1],
    "Names of the columns": data.columns.tolist()
}
```

```
dataset_info
```

```
{'Number of rows in the dataset are': 70707,
 'Number of columns in the dataset are ': 19,
 'Names of the columns': ['Age',
 'Sex',
 'HighChol',
 'CholCheck',
 'BMI',
 'Smoker',
 'HeartDiseaseorAttack',
 'PhysActivity',
 'Fruits',
 'Vegetables',
 'GenHlth',
 'DiffWalk',
 'Stroke',
 'SugarConsumption',
 'Diabetes']
}
```

```

'HvyAlcoholConsump',
'GenHlth',
'MentHlth',
'PhysHlth',
'DiffWalk',
'Stroke',
'HighBP',
'SugarConsumption',
'Diabetes'])

[ ] data.describe()

```

	Age	HighChol	CholCheck	HeartDiseaseorAttack	PhysActivity	HvyAlcol
count	70707.000000	70707.000000	70707.000000	70707.000000	70511.000000	69
mean	8.581116	0.525747	0.975264	0.147878	0.703124	
std	2.864896	0.499340	0.155320	0.354981	0.456885	
min	-31.000000	0.000000	0.000000	0.000000	0.000000	
25%	7.000000	0.000000	1.000000	0.000000	0.000000	
50%	9.000000	1.000000	1.000000	0.000000	1.000000	
75%	11.000000	1.000000	1.000000	0.000000	1.000000	
max	13.000000	1.000000	1.000000	1.000000	1.000000	

II. DATA CLEANING/PROCESSING

Uncleaned data contains duplicate entries or non-unique records, missing values and inconsistencies in the data. This data needs to be thoroughly cleaned to remove such issues and make the dataset fit for analysis.

A. Removing Duplicate Rows:

```

#1 Removing duplicate rows
print("Number of duplicated rows in the dataset are:", data.duplicated().sum())
data = data.drop_duplicates()
print("Number of rows after removing duplicated data are:", len(data))

```

Number of duplicated rows in the dataset are: 5554
Number of rows after removing duplicated data are: 65153

Using `data.duplicated()` method, we found 5,554 records having duplicated data. These duplicates were deleted and reduced the dataset from 70,707 rows to 65,153 rows.

B. Removing Missing Values:

At first, there were 6,702 missing values in the dataset concerning several different columns such as “BMI”, “Smoker”, “HvyAlcoholConsump”, “GenHlth” etc.,

Next, there were extreme cases of datasets deprecated due to the number of missing values (300 missing). After filtering the columns with 300 missing values are removed, which made dataset 64,286 rows. The rest of the categorical/numerical missing values that remain were replaced by mode/ mean respectively.

```

#2 Remove missing values
data.isnull().sum()

```

Age	0
Sex	0
HighChol	0
CholCheck	0
BMI	987
Smoker	1013
HeartDiseaseorAttack	0
PhysActivity	196
Fruits	232
Veggies	215
HvyAlcoholConsump	932
GenHlth	989
MentHlth	947
PhysHlth	964
DiffWalk	227
Stroke	0
HighBP	0
SugarConsumption	0
Diabetes	0

dtype: int64

```

[79] total_missing_values = data.isnull().sum().sum()
print("Total number of missing values in the dataset are:", total_missing_values)

[80] missingvalues_less_than_300 = data.isnull().sum() < 300
columns_to_drop = data.columns[missingvalues_less_than_300]
data.dropna(subset=columns_to_drop, inplace=True)
print("Number of rows after removing columns with missing values less than 300 are:", len(data))

```

Total number of missing values in the dataset are: 6702
Number of rows after removing columns with missing values less than 300 are: 64286

C. Remove inconsistencies:

```

#3 Remove inconsistencies
for column in data.columns:
    unique_items = data[column].unique()
    print(f"Unique items in '{column}':", unique_items)

```

Unique items in 'Age': [4 12 13 11 8 1 6 3 -1 7 10 9 5 2 -9 -31 -12 -21 -3 -2 -7 -15 -19]
Unique items in 'Sex': ['Male' 'Female' 'M' 'F']
Unique items in 'HighChol': [0 1]
Unique items in 'CholCheck': [1 0]
Unique items in 'BMI': [26 28 29 31 32 33 34 35 36 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 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

```

Unique items in 'HighChol': [0 1]
Unique items in 'CholCheck': [1 0]
Unique items in 'BMI': [26. 28. 29. 18. 31. 32. 27. 24. 21. 58. 39. 20. 22. 38. 40. 25. 36. 47.
19. 37. 41. 23. 34. 35. 42. 17. 33. 44. nan 15. 52. 69. 56. 45. 39. 92.
98. 50. 46. 79. 48. 16. 63. 72. 54. 49. 68. 43. 84. 53. 73. 76. 55. 51.
75. 57. 60. 77. 82. 67. 71. 61. 14. 81. 59. 86. 13. 87. 65. 95. 89. 62.
64. 66. 85. 70. 83. 80. 78.]
Unique items in 'Smoker': ['Non Smoker' 'Smoker' nan]
Unique items in 'HeartDiseaseorAttack': [0 1]
Unique items in 'PhysActivity': [1. 0.]
Unique items in 'Fruits': ['Does not Eat' 'Eat']
Unique items in 'Veggies': ['Eat' 'Does not Eat']
Unique items in 'MyAlcoholConsump': [0. 1. nan]
Unique items in 'GenHlth': ['good' 'excellent' 'very good' 'fair' 'poor' nan]
Unique items in 'MentHlth': [5. 0. 7. 3. 4. 2. 30. 20. 1. 15. nan 25. 14. 28. 10. 6. 29. 26.
12. 22. 13. 8. 9. 18. 16. 21. 17. 27. 24. 23. 11. 19.]
Unique items in 'PhysHlth': [30. 0. 10. 3. 6. 4. 15. 1. 2. 14. 7. 25. nan 21. 20. 5. 8. 22.
23. 29. 12. 18. 28. 26. 24. 27. 11. 13. 16. 17. 9. 19.]
Unique items in 'DiffWalk': ['N' 'Y']
Unique items in 'Stroke': ['N' 'Y']
Unique items in 'HighBP': [1 0]
Unique items in 'SugarConsumption': ['High' 'Low']
Unique items in 'Diabetes': ['N' 'Y']

```

There are inconsistencies in data for instance 'Sex': 'Male', 'M', 'Female', 'F', 'DiffWalk', 'Stroke': 'Yes', 'Y', 'No', 'N' and in the 'BMI' column; being values measured in kg/m². This was also done for all inconsistencies to ensure they were uniform.

D. Filling Null Values with Mean & Mode:

```

#4 Filling null values with mode and mean
for col in data.columns:
    if data[col].isnull().any():
        if data[col].dtype == 'object':
            data[col].fillna(data[col].mode()[0], inplace=True)
        else:
            data[col].fillna(data[col].mean(), inplace=True)
print("After imputing the null values in the dataset are:", data.isnull().sum().sum())

```

After imputing the null values in the dataset are: 0

After filling null values in the dataset. There should not be any missing values after replacing null values of categorical columns with the mode and numerical columns with the mean.

E. Removing Negative Values:

```

#5 Removing negative values in age column and before and after removing negative values
print("Number of negative values in the 'Age' column:", (data['Age'] < 0).sum())
data = data[data['Age'] >= 0]
print("Number of rows after removing negative values:", len(data))

```

Number of negative values in the 'Age' column: 11
Number of rows after removing negative values: 64275

There are a total of 11 negative values in 'Age' column which are needed to be removed. Therefore, after removal total no of remaining rows in the dataset are 64,275.

F. Creating New Feature:

```

# 6. Creating new attribute AgeGroup to classify as kids and teens
age_bins = [0, 18, 100]
age_labels = ['Kid', 'Teen']
data['Age_Group'] = pd.cut(data['Age'], bins=age_bins, labels=age_labels)
print("Number of kids and teens in the dataset are:")
print(data['Age_Group'].value_counts())
data.head(5)

```

Number of kids and teens in the dataset are:
Age_Group
Kid 46796
Teen 17479
Name: count, dtype: int64

Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Ve
0	4	M	0	1 26.0	Non Smoker	0	1.0	Does not Eat	
1	12	M	1	1 26.0	Smoker	0	0.0	Eat	
2	13	M	0	1 26.0	Non Smoker	0	1.0	Eat	
3	11	M	1	1 28.0	Smoker	0	1.0	Eat	
4	8	F	0	1 29.0	Smoker	0	1.0	Eat	

Classifying individuals with respect to their age and creating a new column 'Age_Group' as follows:

- Kid: Age < 10
- Teen: Age ≥ 10

G. Data Conversion:

```

#7. Data type conversion for BMI column
data['BMI'] = data['BMI'].astype(int)
data['Age'] = data['Age'].astype(int)
data.head(5)

```

Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Ve
0	4	M	0	1 26	Non Smoker	0	1.0	Does not Eat	
1	12	M	1	1 26	Smoker	0	0.0	Eat	
2	13	M	0	1 26	Non Smoker	0	1.0	Eat	
3	11	M	1	1 28	Smoker	0	1.0	Eat	
4	8	F	0	1 29	Smoker	0	1.0	Eat	

Converting the datatype of 'BMI' and 'Age' columns as integer so that it helps to make analysis easy by using mathematical techniques where numerical precision is required.

H. BMI Group Classification:

```

bmi_bins = [18, 25, 30, 40, 70]
bmi_labels = ['UnderWeight', 'HealthyWeight', 'OverWeight', 'Obese']
data['BMI_Group'] = pd.cut(data['BMI'], bins=bmi_bins, labels=bmi_labels)
print("Number of BMI groups in the dataset are:")
print(data['BMI_Group'].value_counts())
data.head(5)

```

Number of BMI groups in the dataset are:
BMI_Group
HealthyWeight 22293
OverWeight 20612
UnderWeight 15842
Obese 4764
Name: count, dtype: int64

Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	F
0	4	M	0	1 26	Non Smoker	0	1.0	
1	12	M	1	1 26	Smoker	0	0.0	
2	13	M	0	1 26	Non Smoker	0	1.0	
3	11	M	1	1 28	Smoker	0	1.0	
4	8	F	0	1 29	Smoker	0	1.0	

Classifying individuals with respect to their BMI and creating a new feature named 'BMI_Group' by categorizing them with the labels 'UnderWeight', 'HealthyWeight', 'OverWeight' and 'Obese'.

I. Label Encoding:

```

#9. Label Encoding
from sklearn.preprocessing import LabelEncoder
lbi_encoded = LabelEncoder()

for column in data.columns:
    if data[column].dtype == 'object' or 'category':
        data[column] = lbi_encoded.fit_transform(data[column])
data.head(5)

```

Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies
0	3	1	0	1 13	0	0	1	0	1
1	11	1	1	1 13	1	0	0	1	0
2	12	1	0	1 13	0	0	1	1	0
3	10	1	1	1 15	1	0	1	1	1
4	7	0	0	1 16	1	0	1	1	1

5 rows × 21 columns

Label Encoding helps in transforming a categorical variable into a numerical format which helps in effective use of machine learning models.

J. Dropping Features:

```
#10. Dropping Features
print("Number of High and Low values in SugarConsumption:")
print(data['SugarConsumption'].value_counts())
data = data.drop(columns=['SugarConsumption'])
data.head(5)
```

Number of High and Low values in SugarConsumption:
SugarConsumption
0 64271
1 4
Name: count, dtype: int64

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	V
0	3	1	0	1	13	0	0	1	0	
1	11	1	1	1	13	1	0	0	1	
2	12	1	0	1	13	0	0	1	1	
3	10	1	1	1	15	1	0	1	1	
4	7	0	0	1	16	1	0	1	1	

By using feature extraction, we dropped the 'SugarConsumption' feature as it contains only 4 entries marked as 'High' making it low variance.

K. Removing Outliers:

```
#11. Removing outliers
def remove_outliers(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    print(f"Lower bound for '{column}': {lower_bound}")
    print(f"Upper bound for '{column}': {upper_bound}")
    return data[(data[column] >= lower_bound) & (data[column] <= upper_bound)]

columns_to_clean = ['BMI', 'Age']
for col in columns_to_clean:
    data = remove_outliers(data, col)

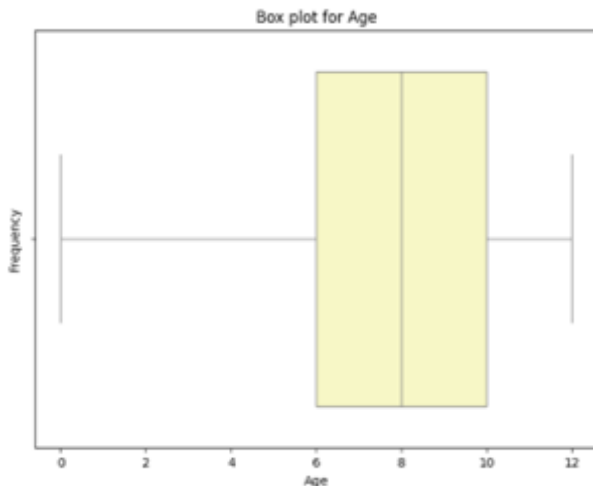
print("Number of rows after removing outliers:", len(data))
```

Lower bound for 'BMI': 0.0
Upper bound for 'BMI': 32.0
Lower bound for 'Age': 0.0
Upper bound for 'Age': 16.0
Number of rows after removing outliers: 62168

By using IQR method, we can find the outliers using upper and lower bounds. After removing these outliers, we finally have 62,168 rows in our dataset.

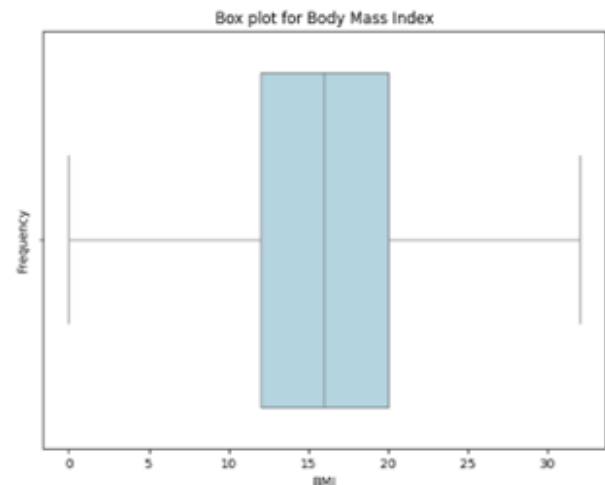
III. EXPLORATORY DATA ANALYSIS

A. Box Plots:



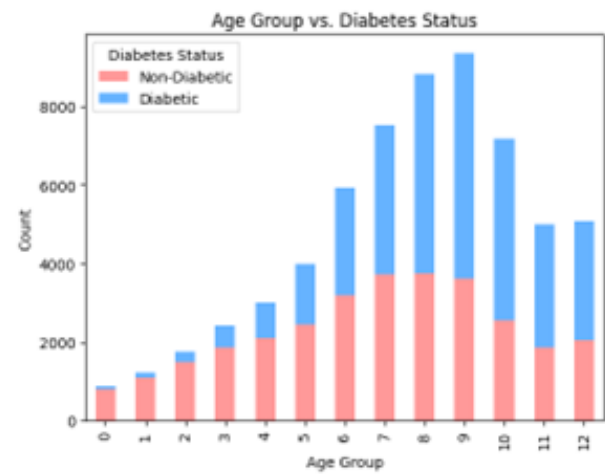
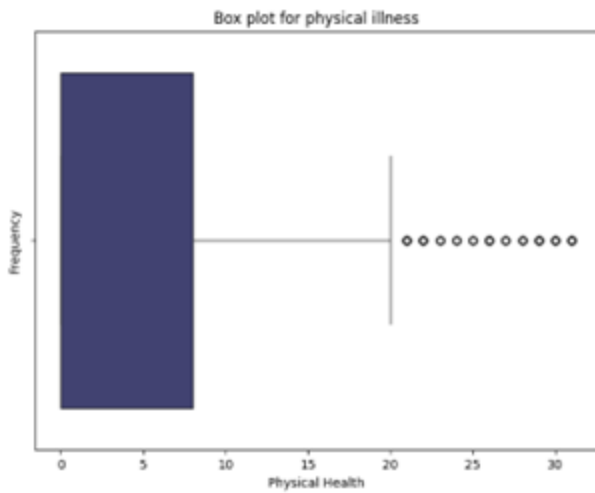
Observations:

- The median of the age distribution is 8, implying that the average population is relatively homogenous.
- There is no possible reason for concerns toward the existence of outliers, as the whiskers reach irritably both the lower and upper advents without any glaring irregularities.
- The central IQR was found to range between 6 and 10 with the latter edge extending the middle range of 50 percent of the data.
- The length of the whiskers on either side is about the same, thus implying that data is normally distributed with no considerable skewness.



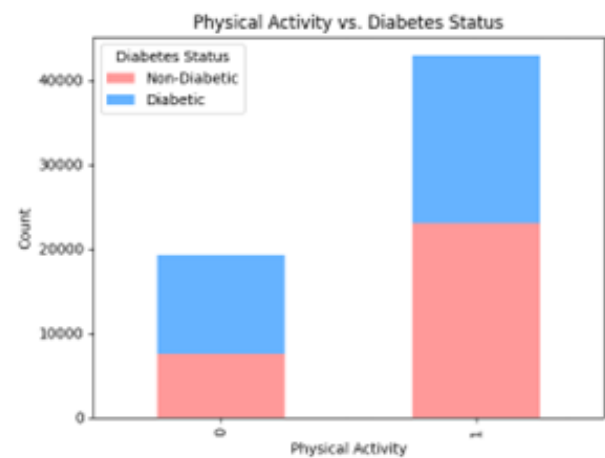
Observations:

- The median BMI value seems to be in the middle of the lower range on the BM index with a value of about 17.
- The concept of outliers is not in this data as evidenced by the absence of any points that are beyond the whiskers.
- The IQR is roughly between 15 and 20; thus middle 50 values.
- 'Whiskers' extends in equal lengths on the diagrams both to the right and left sides, which suggests that there are no significant distortions in the distribution of the BMI values.



Observations:

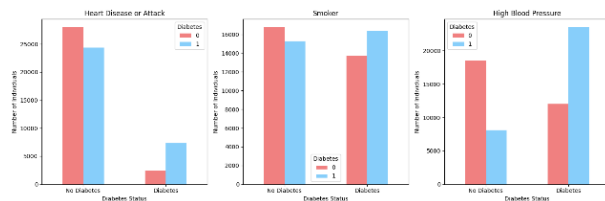
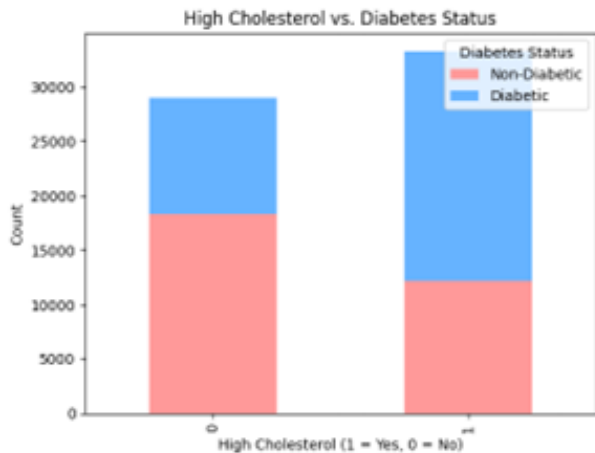
- In the second box plot, the distribution of physical health appears to be negative as many people report low value mostly in keeping physical wellbeing more so within (0 to 5 days).
- For example, the median is about 2-3 days, suggesting that people do not tend to report a lot of physical health problems.
- Some maximum values which are more than 15 days there are many outliers outside the whiskers diagrams this illustrates that some people were ill or injured for more days than others.
- In contrast, the range shown by the spread of the whiskers was even wider indicating that respondents reported experiencing different levels of physical health.



Observations:

- From the first graph above, it can be inferred that people with a high daily cholesterol intake have a higher chance of diabetes than those without a high daily cholesterol intake.
- The second graph very clearly shows that as a person grows older, the chances of having diabetes increase with the highest numbers lying in the middle age categories.
- The third graph conclusively shows that those who participate in various forms of physical activities are at a lesser risk of acquiring diabetes as opposed to those who are inactive.

B. Bar Plots:



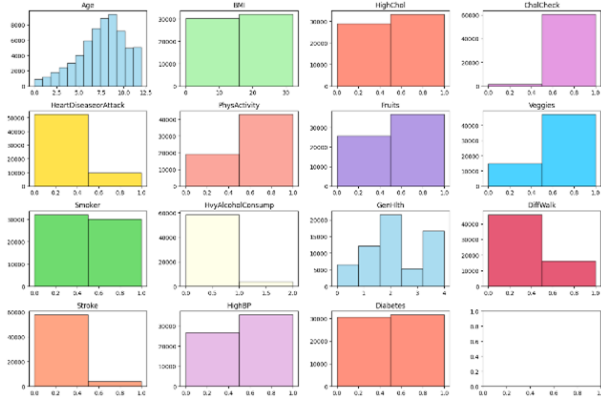
Observations:

- From the first graph, it seems that people suffering from diabetes have the higher chances of having either heart

disease or one in the form of heart attack, however, it is the total number who do not suffer from diabetes who have a higher number of cases.

- In the other graph, it is shown that the propensity to smoke is almost the same among people with diabetes and without diabetes, therefore, it cannot be concluded that smokers with diabetes are more than nonsmokers.
- The third graph says that high blood pressure, or hypertension, seems to strike people with diabetes more than those without, thus linking diabetes and hypertension heavily.

C. Histograms:

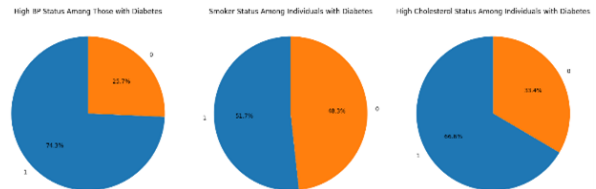


Observations from the Multiple Histogram Plots:

- Considering the sample demographics, there are more middle-aged respondents with the distribution pattern observed to be concentrated in the mid-age groups, a range which is probably between 6 and 9 units. This suggests age distribution is skewed.
- BMI parameters are equally distributed, showing good representation of respondents in both categories.
- Many of the respondents have reported having high cholesterol which indicates a greater number of persons suffering from the condition that the sample represent.
- Since most of the respondents have undergone a cholesterol check, it follows that cholesterol screening is a common practice among the respondents.
- Most of the respondents have never had a case of heart disease or heart attack. This reveals a good cardiovascular health status amongst the sample.
- A larger number of my respondents go out to exercise, this shows increased levels of physical activity among the respondents.
- While a slightly larger portion of the respondents consumes fruits on regular basis, a considerable portion does not, therefore these have varied dietary practices.
- Regular vegetable consumption is to some extent higher as more respondents report to regular meal with vegetables.
- Smokers are slightly fewer than non-smokers indicating that the smoking status is almost evenly spread within the population sample.

- It is uncommon to have heavy drinkers as most respondents indicate only moderate or lower drinking levels.
- Health ratings are corroborated by majority of the people with moderate health ratings accordingly and is perceived on average level.
- While a small subgroup does walk, difficulty in mobilization is not prevalent.

D. Pie Charts:



- High Blood Pressure (BP) Status Among Those with Diabetes:
 - 74.3% high blood pressure.
 - 25.7% not have high blood pressure.

Observation:

A large proportion of people with diabetes (nearly three-quarters) also suffer from high blood pressure, which suggests a strong link between diabetes and high BP.

- Smoker Status Among Individuals with Diabetes:
 - 51.7% smokers.
 - 48.3% non-smokers.

Observation:

Slightly more than half of the individuals with diabetes are smokers, which is concerning as smoking can exacerbate diabetes-related complications. There is an almost equal split between smokers and non-smokers in this population.

- High Cholesterol Status Among Individuals with Diabetes:
 - 66.6% high cholesterol.
 - 33.4% not have high cholesterol.

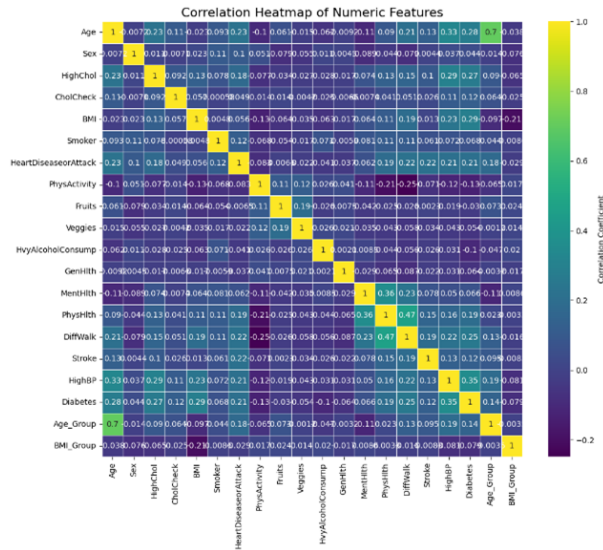
Observation:

A significant majority (two-thirds) of individuals with diabetes also have high cholesterol. This indicates a high comorbidity of cholesterol problems in diabetic individuals, increasing their risk for cardiovascular complications.

General Observations:

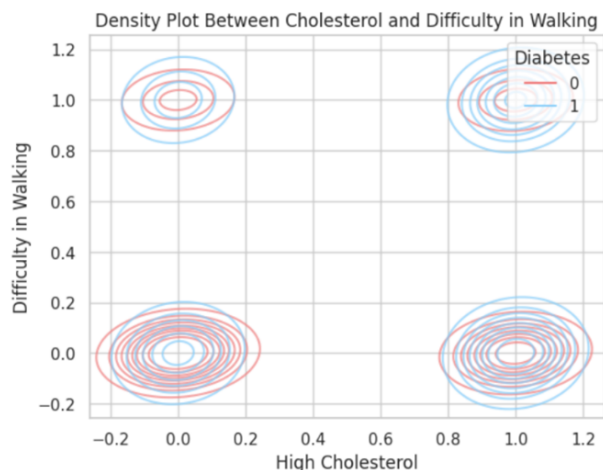
- Comorbidities: The high percentages of people with diabetes suffering from high blood pressure (74.3%) and high cholesterol (66.6%) indicate that these comorbidities are common in diabetic populations.
- Smoking Risk: More than half (51.7%) of diabetic individuals are smokers, which could further increase the risk of cardiovascular diseases and complications associated with diabetes.

E. Heatmap:



- Age correlates with high blood pressure, high cholesterol, and diabetes, indicating age-related increases in these conditions.
- BMI is linked to high cholesterol and high blood pressure, showing that higher body weight contributes to both.
- General health and physical health are closely related, with poor general health strongly tied to difficulty walking and poor physical health.
- High BP, BMI, high cholesterol, and age are the strongest predictors of diabetes in the dataset.
- Physical activity is positively associated with better general health and physical health, but negatively related to difficulty walking.
- Smoking negatively correlates with fruit and vegetable consumption, indicating poorer diet choices among smokers.

F. Density plot between Cholesterol and Difficulty in Walking:



Observations:

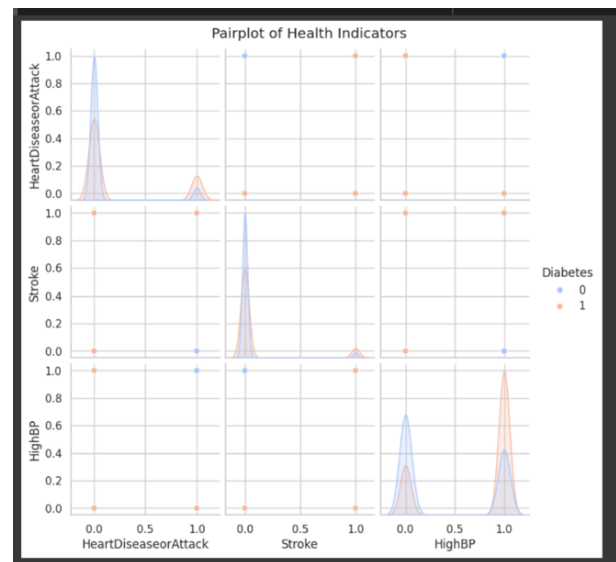
The plot clearly delineates two different density contour clusters and presents a clear relation of high cholesterol with the difficulty in walking. Those having had greater cholesterol levels have been seen to have difficulties in walking.

Impact of Diabetes:

- The plot suggests that diabetes patients are more likely to be hypercholesterolemic and have difficulty in walking and the excursion is extended further.
- Diabetes is probably not the sole cause of difficulty in walking since there is a clear overlap of the two groups.
- For cholesterol levels, it is observed that most of the scores are not clustered but rather a majority of them fall at the peak around 0.8.
- For difficulty in walking as well, only a few are appor- tioning around the concentration of 0.2.

This indicates that there is an indelible link between high cholesterol levels, diabetes and the inability to walk.

G. Pair plot of Health indicators:

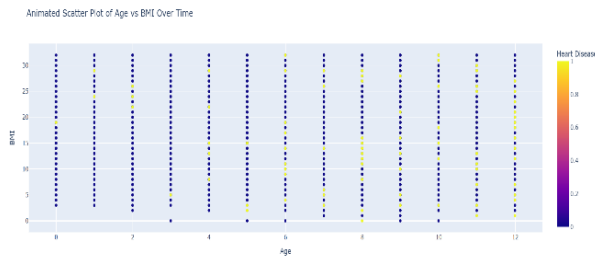


Observations:

- Positive Correlations: There is a positive correlation in all three health conditions, and each one will increase the risk for the others.
- Strongest Association: Heart disease/attack and high blood pressure have the most closely related association.
- Skewed Distributions: The distributions of heart disease/attack and stroke are left-skewed, while the high blood pressure values are dispersed.

High blood pressure acts as a common risk factor between heart disease/attack and stroke, while the prevalence of both heart disease/attack and stroke is comparably low to that of high blood pressure.

H. Scatter Plot of Age vs BMI Over Time:



Observations:

- **Age Range:** The dataset encompasses a wide age bracket, ranging from early childhood to teenage. This infers a heterogeneous child population sample.
- **Distribution of BMI:** The graph peaks in the middle range between 15-25. That is indicative of at least a sizeable percentage of the population falling within a healthy range.
- There is a general trend of rising BMI with age, more so in later years. It shows that weight gain could be related to the factors of lifestyle, such as reduced physical activity and change in diet, through the years.
- Higher values of BMI are associated with increased risks of heart disease, again wellknown, which gives emphasis to the need for keeping weight at as healthy a level as possible.
- Animation of the plot could show possible changes in the relationship of age to BMI with heart disease over time.

I. Correlation Bar Plot:



- **Strong Positive Correlation:** The important risk factors that are associated with diabetes include age, high cholesterol, BMI, difficulty walking, and high blood pressure. This has brought the need to attend to these factors through lifestyle changes, medical interventions, and preventive measures.
- **Positive Correlation-Moderate:** At this level, are considered the following as moderate risks: sex, cholesterol checks, smoking, heart disease/attack, mental health, physical health, and stroke. While these do not relate as

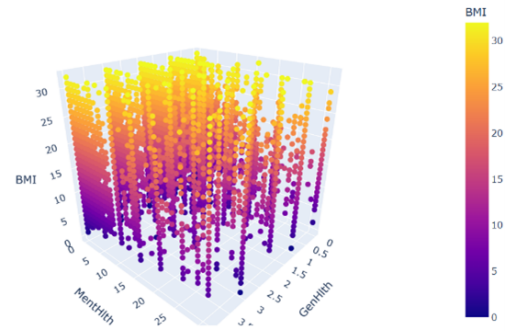
highly with diabetes as the above-named key risk factors, they do add to the overall risk.

- **Negative Correlation:** Physical activity, fruits, vegetables, heavy alcohol consumption, general health, and the group of BMI are associated with lower risk of diabetes. To prevent diabetes, healthy lifestyle habits can be promoted.

J. 3D Scatter Plot:

K. Correlation Bar Plot:

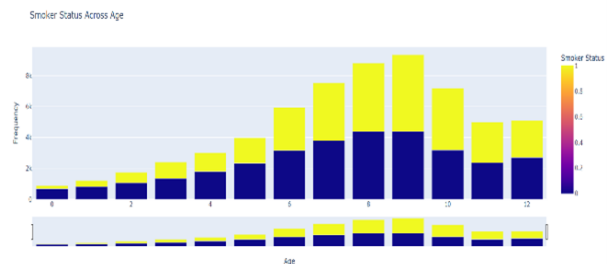
3D Scatter Plot



This plot shows a complex interrelationship among these three variables.

- There is a slight negative correlation between BMI and mental health. It could be interpreted that for those people having higher values of BMI, their scores of mental health may be lower, though the relation is not exactly strong.
- The correlation for BMI in relation to general health is less clear. There seems to be a slight negative correlation, but it is less than the relationship of BMI with mental health.

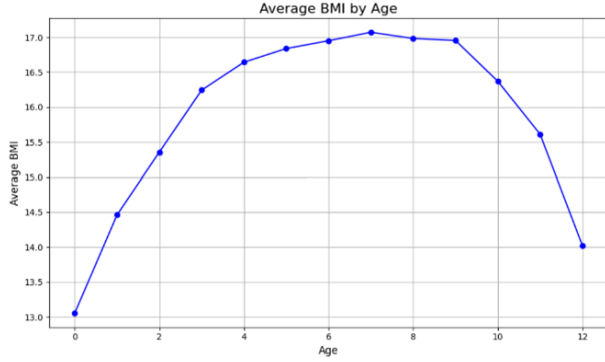
L. Stacked bar graph for Smoker Status Across Age Graph:



The height of each bar will show the number of people in that age group who are smokers and who are not smokers.

- **Increasing Smoking Prevalence with Age:** The graph above clearly indicates a very strong positive correlation between age and smoking prevalence. As age increases, so does the probability of being a smoker.
- **Peak Smoking Rates:** It reaches the peak in the mid-to-late adult years, around 8-10 years. This may indicate that smoking initiation or continuation is more vulnerable in these age groups.

M. Line Graph of BMI vs Age:



Observations:

- The graph shows a clear U-shaped curve, indicating that average BMI increases from early childhood to middle age, reaches a peak, and then declines in later adulthood.
- The highest average BMI is observed in midadulthood (around age 7-8). This suggests that individuals in this age group tend to have the highest body mass index.
- In early childhood (ages 0-2), average BMI is relatively low. During adolescence (ages 3-6), BMI increases steadily, reflecting the typical growth spurt.
- In late adulthood (ages 10-12), average BMI starts to decline. This could be attributed to various factors, such as decreased physical activity, changes in metabolism, or health conditions.

IV. ALGORITHMS

In this project, we applied six significant algorithms to predict diabetes risk. The selected algorithms include both commonly discussed methods and those not covered in class.

Algorithms Applied:

- K-Nearest Neighbors (KNN)
- Naïve Bayes
- Logistic Regression
- Support Vector Machines (SVM)
- Random Forest
- Extreme Gradient Boosting (XGBoost)
- Decision Tree

A. K-Nearest Neighbors (KNN)

There are several reasons to select the KNN algorithm for our diabetes prediction model. One key advantage is that it captures complex, non-linear relationships in the dataset. Diabetes prediction relies on comparing multiple parameters, including physical activity, smoking status, BMI, age, and more.

Additionally, KNN is robust against outliers, a common issue in healthcare datasets due to variations in individual health. This resilience ensures that the model's predictions remain reliable despite the presence of atypical data points. Furthermore, the interpretability of KNN is a significant benefit in healthcare contexts, as it provides transparency

in decision-making by determining outcomes based on the majority class of the nearest neighbors.

- **Tuning:** In our investigation, Initially we got accuracy 0.68 and then we did two key hyperparameters that influence the results obtained from K-Nearest Neighbors were targeted: the number of neighbors-`n_neighbors`-and a metric of distance. This is resolved through systematic testing for `n_neighbors`, namely, 3, 5, 7, and 9, whereas distance metric evaluation considered Euclidean, Manhattan, and Minkowski. A thorough cross-validation using `GridSearchCV` can find the best configuration. This was particularly when applying the 5 neighbors with the Minkowski distance metric. This therefore points to the importance of adjustment in hyperparameters in order to improve performance in KNN algorithms for our classification problem in diabetes.
- **Effectiveness:** From the cross-validation training and testing and classification report, we can observe the accuracy is 70.5%.

KNN Accuracy & Report:

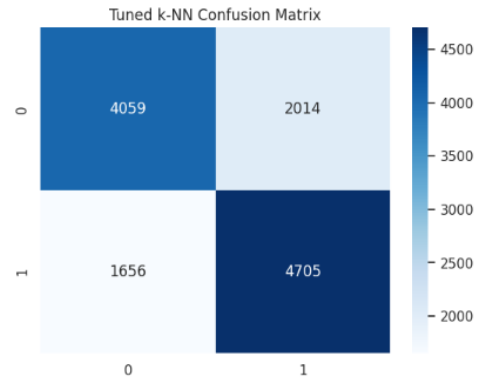
```
/usr/local/lib/python3.10/dist-packages/numpy/ma/core.py:2820: RuntimeWarning:
```

```
invalid value encountered in cast
```

```
Best KNN Parameters: {'metric': 'manhattan', 'n_neighbors': 9}  
Tuned KNN Accuracy: 0.7048415634550427
```

Training Report				
	precision	recall	f1-score	support
0	0.77	0.72	0.74	24428
1	0.75	0.80	0.77	25314
accuracy			0.76	49734
macro avg	0.76	0.76	0.76	49734
weighted avg	0.76	0.76	0.76	49734

Testing Report				
	precision	recall	f1-score	support
0	0.71	0.67	0.69	6073
1	0.70	0.74	0.72	6361
accuracy			0.70	12434
macro avg	0.71	0.70	0.70	12434
weighted avg	0.71	0.70	0.70	12434



B. Naive Bayes

Using the Naive Bayes algorithm for diabetes prediction presents several benefits that align well with the characteristics of the dataset. One of its primary advantages is its ability to handle categorical data effectively, making it particularly appropriate for features like gender, smoking status, history of heart disease, and stroke, which are crucial risk factors in diabetes-related datasets. Given that diabetes can lead to

severe health complications if not identified early, accurately predicting risk is essential. Naive Bayes assumes that the features are independent, simplifying the computation and modeling process, especially when dealing with the categorical aspects of certain variables.

The probabilistic nature of the model allows for clear insights into how predictions are made, which is vital for users to understand the model's logic. This transparency is especially important in the context of diabetes prediction, where trusting the model's conclusions can significantly impact healthcare decisions, potentially leading to timely interventions. Additionally, Naive Bayes boasts a quick training time, making it an excellent option for projects that are resource-limited or require prompt results, such as those involving diabetes datasets.

- **Tuning:** Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- **Effectiveness:** From the cross-validation training and testing and classification report, we can observe the accuracy is 71.9%.

Naive Bayes Accuracy & Report:

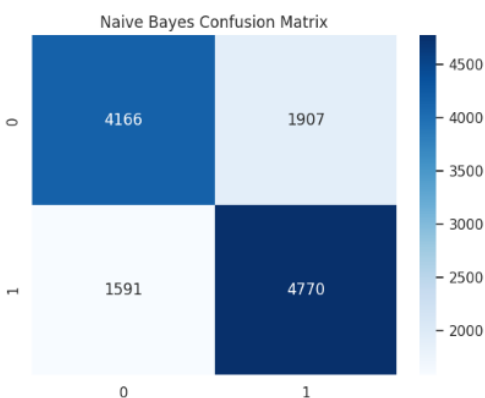
Naive Bayes Accuracy: 0.7186746018980216

Training Report

	precision	recall	f1-score	support
0	0.72	0.68	0.70	24420
1	0.71	0.75	0.73	25314
accuracy			0.71	49734
macro avg	0.71	0.71	0.71	49734
weighted avg	0.71	0.71	0.71	49734

Testing Report

	precision	recall	f1-score	support
0	0.72	0.69	0.70	6073
1	0.71	0.75	0.73	6361
accuracy			0.72	12434
macro avg	0.72	0.72	0.72	12434
weighted avg	0.72	0.72	0.72	12434



C. Logistic Regression

Logistic regression is a widely used supervised machine learning technique designed to estimate the likelihood of binary outcomes, such as whether a patient is diagnosed with diabetes.

The main strength of logistic regression is its straightforwardness. The model provides clear insights into how different variables contribute to predicted outcomes, making it easier for users to understand the relationship between predictors and diabetes probability. Coefficients from the logistic regression equation represent the odds associated with each predictor.

Furthermore, logistic regression is known for its complexity. In general, it performs better when applied to new data, as it is less likely to overfit the training data. This characteristic is particularly important in real-world applications, where the training data may not fully reflect the state of the data used to predict. By balancing accuracy, logistic regression is a reliable tool in the healthcare context for diabetes prediction.

- **Tuning:** Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- **Effectiveness:** From the cross-validation training and testing and classification report, we can observe accuracy is 72.5%.

Logistic Regression Accuracy & Report:

Logistic Regression Accuracy: 0.72534984719318

Training Report

	precision	recall	f1-score	support
0	0.73	0.69	0.71	24420
1	0.72	0.75	0.73	25314
accuracy			0.72	49734
macro avg	0.72	0.72	0.72	49734
weighted avg	0.72	0.72	0.72	49734

Testing Report

	precision	recall	f1-score	support
0	0.73	0.70	0.71	6073
1	0.72	0.75	0.74	6361
accuracy			0.73	12434
macro avg	0.73	0.72	0.72	12434
weighted avg	0.73	0.73	0.73	12434



D. Support Vector Machines (SVM)

One of the robust supervised learning techniques that form quite ideal choices for the classification of diabetes status, a factor very critical in managing a chronic condition affecting millions across the globe, is Support Vector Machines. Among

the key merits of SVM is handling high-dimensional data with great effectiveness, which comes in pretty useful considering the scope of risk factors associated with diabetes—anything from obesity to age to family and lifestyle choices. By using kernel functions, SVM is able to model complex nonlinear relationships between features that can reveal patterns in the dataset for indicating a person’s susceptibility to diabetes.

Besides this, SVM is resistant to overfitting when the number of features outnumbers the samples, which also is very common in medical datasets. This property ensures that the model generalizes well for unseen data, crucial for proper prediction in real-world applications. The clear margin of separation obtained while using SVM enhances interpretability by allowing health professionals to more easily understand how the model differentiates between diabetic versus nondiabetic patients. Overall, the support vector machine will make correct predictions that would enable the doctor or physician to make prudent decisions in terms of assessment and management of the risk of diabetes to improve patient outcomes.

- Tuning: Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model’s performance in classifying diabetes in our data was already satisfactory.
- effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 73%.

SVM Accuracy & Report:

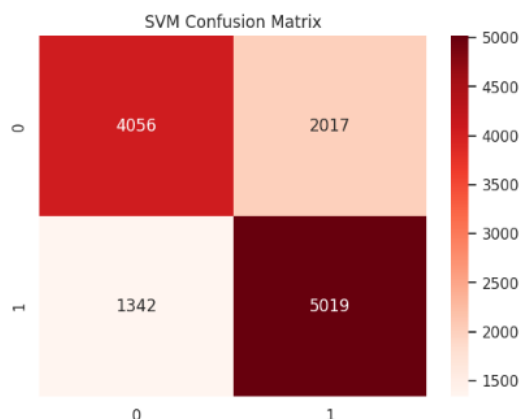
SVM Accuracy: 0.7298536271513592

Training Report

	precision	recall	f1-score	support
0	0.75	0.66	0.70	24420
1	0.71	0.79	0.74	25314
accuracy			0.72	49734
macro avg	0.73	0.72	0.72	49734
weighted avg	0.73	0.72	0.72	49734

Testing Report

	precision	recall	f1-score	support
0	0.75	0.67	0.71	6073
1	0.71	0.79	0.75	6361
accuracy			0.73	12434
macro avg	0.73	0.73	0.73	12434
weighted avg	0.73	0.73	0.73	12434



E. Random Forest

Random Forest is particularly effective at modeling the complex relationships inherent in datasets, making it essential for accurately assessing the diverse risk factors associated with diabetes. By creating multiple decision trees and aggregating their predictions through ensemble learning, Random Forest enhances the model’s resilience, effectively managing the variability and noise that often characterize healthcare data.

Another significant advantage of Random Forest is its ability to evaluate feature importance, which is crucial in the context of diabetes prediction. This capability provides valuable insights into the significance of various factors, such as BMI, blood pressure, and physical activity, allowing for a clearer understanding of how each parameter contributes to diabetes risk. In healthcare settings, where transparency is vital, this analysis of feature importance fosters trust and comprehension among healthcare providers and stakeholders.

- Tuning: Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model’s performance in classifying diabetes in our data was already satisfactory.
- Effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 70.0%.

Random Forest Accuracy & Report:

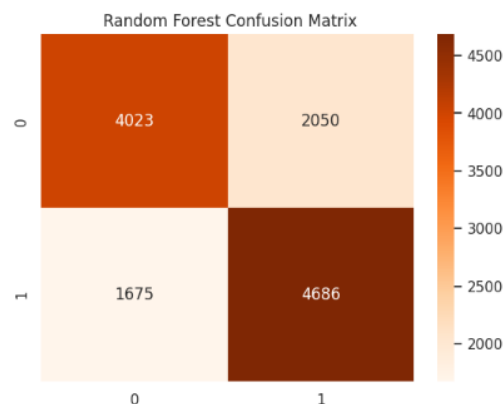
Random Forest Accuracy: 0.7004182081389738

Training Report

	precision	recall	f1-score	support
0	0.97	0.98	0.98	24420
1	0.98	0.97	0.98	25314
accuracy			0.98	49734
macro avg	0.98	0.98	0.98	49734
weighted avg	0.98	0.98	0.98	49734

Testing Report

	precision	recall	f1-score	support
0	0.71	0.66	0.68	6073
1	0.70	0.74	0.72	6361
accuracy			0.70	12434
macro avg	0.70	0.70	0.70	12434
weighted avg	0.70	0.70	0.70	12434



F. XGBoost

XGBoost yields highly accurate predictions of diabetes, a chronic disease characterized by high levels of blood glucose, together with serious subsequent health complications. This algorithm works miracles on complex data and can also reflect nonlinear relationships between various risk factors such as age, BMI, and physical activity, critical in diabetic risk assessment. Utilizing gradient boosting, XGBoost ensembles a large number of weak learners to generate a strong predictive model while integrating techniques for regularization that avoid overfitting.

Also, XGBoost handles missing values and feature importance scores, thus enabling healthcare practitioners to identify important risk factors. Such a scalable and fast algorithm is suitable for large datasets and helps to make timely predictions in the clinical environment. Customizable hyperparameters allow XGBoost to be fine-tuned for the peculiarities in diabetes prediction, therefore turning even more effective for healthcare applications.

- **Tuning:** Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- **Effectiveness:** From the cross-validation training and testing and classification report, we can observe the accuracy is 73.7%.

XGBoost Accuracy & Report:

```
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:158: UserWarning:
```

```
[20:56:59] WARNING: /workspace/src/learner.cc:740:  
Parameters: { "use_label_encoder" } are not used.
```

XGBoost Accuracy: 0.7370114202991797

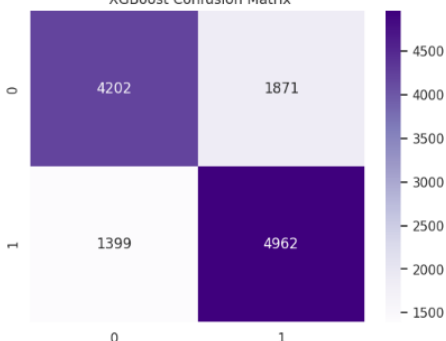
Training Report

	precision	recall	f1-score	support
0	0.80	0.73	0.76	24420
1	0.76	0.82	0.79	25314
accuracy			0.78	49734
macro avg	0.78	0.78	0.78	49734
weighted avg	0.78	0.78	0.78	49734

Testing Report

	precision	recall	f1-score	support
0	0.75	0.69	0.72	6073
1	0.73	0.78	0.75	6361
accuracy			0.74	12434
macro avg	0.74	0.74	0.74	12434
weighted avg	0.74	0.74	0.74	12434

XGBoost Confusion Matrix



G. Decision Tree

Decision Trees are effective for predicting diabetes, capturing complex relationships between risk factors like age, BMI, and physical activity. Their intuitive structure allows for clear, interpretable outcomes, making them valuable in clinical settings. The algorithm can handle missing values and provides insights into feature importance, helping identify key diabetes risk factors. While they may overfit, techniques like pruning and ensemble methods can enhance their accuracy, making Decision Trees adaptable for various healthcare applications.

- **Tuning:** In our investigation, we initially achieved an accuracy of 0.62 with the Decision Tree model. We focused on three key hyperparameters that significantly impact the model's performance: `max_depth`, `min_samples_split`, and `min_samples_leaf`. We systematically evaluated various configurations for these parameters, testing `max_depth` values of None, 5, 10, and 15, as well as `min_samples_split` options of 2, 5, and 10, and `min_samples_leaf` settings of 1, 2, and 4. Using GridSearchCV for comprehensive cross-validation, we identified the optimal configuration that improved the model's accuracy to 72.6. This process highlights the importance of adjusting hyperparameters to enhance model performance in our diabetes classification problem.
- **Effectiveness:** From the cross-validation training and testing and classification report, we can observe the accuracy is 72.5%.

Decision Tree Accuracy & Report:

```
Best Decision Tree Parameters: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 2}  
Tuned Decision Tree Accuracy: 0.7252694225510696
```

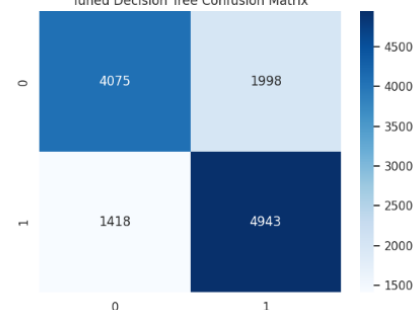
Training Report

	precision	recall	f1-score	support
0	0.77	0.69	0.73	24420
1	0.73	0.80	0.76	25314
accuracy			0.75	49734
macro avg	0.75	0.75	0.75	49734
weighted avg	0.75	0.75	0.75	49734

Testing Report

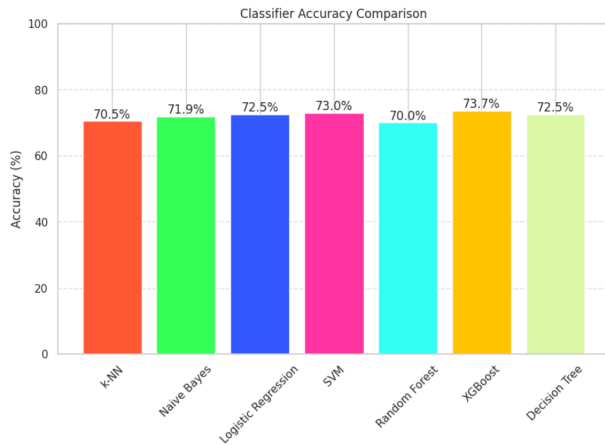
	precision	recall	f1-score	support
0	0.74	0.67	0.70	6073
1	0.71	0.78	0.74	6361
accuracy			0.73	12434
macro avg	0.73	0.72	0.72	12434
weighted avg	0.73	0.73	0.72	12434

Tuned Decision Tree Confusion Matrix



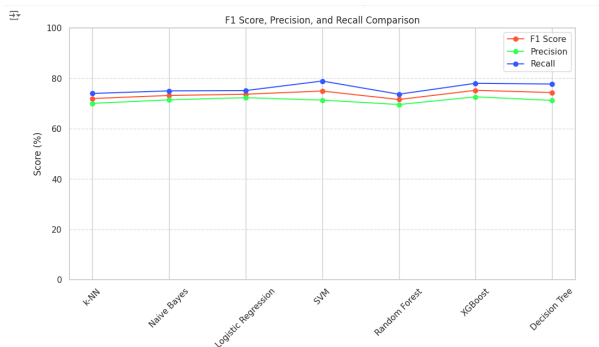
V. EVALUATION

The bar graph compares the accuracy of different classification algorithms (K-NN, Naive Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and Decision Tree) on a diabetes prediction problem.



- XGBoost and Decision Tree achieved the highest accuracy of 73.7% and 72.5%, respectively, indicating that these algorithms might be the most suitable for this particular diabetes prediction task.
- K-NN and Naive Bayes didn't perform well, with accuracies of 70.5% and 71.9%, respectively.
- Logistic Regression, SVM, and Random Forest had accuracies in the mid-range, with values between 72.5% and 73.0%.

The below line graph compares the F1 score, precision, and recall of different classification algorithms (K-NN, Naive Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and Decision Tree) on a diabetes prediction problem.



- XGBoost and Decision tree generally exhibits the best performance across all three metrics, achieving the highest F1 score, precision, and recall. This suggests that XGBoost is a well-rounded algorithm for this particular diabetes prediction task.
- Logistic Regression and SVM achieve decent F1 scores but have lower precision and recall compared to XGBoost and Decision Tree.

Overall, the graph suggests that XGBoost and Decision Tree are promising algorithms for diabetes prediction.

REFERENCES

- [1] <https://scikit-learn.org/stable/modules/tree.html>
- [2] <https://www.datacamp.com/tutorial/xgboost-in-python>
- [3] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

Peer Evaluation Form for Final Group Work

CSE 487/587B

- Please write the names of your group members.

Group member 1 : Sahithya Arveti Nagaraju

Group member 2 : Sushmitha Manjunatha

Group member 3 : Mounika Pasupuleti

- Rate each groupmate on a scale of 5 on the following points, with 5 being HIGHEST and 1 being LOWEST.

Evaluation Criteria	Group member 1	Group member 2	Group member 3
How effectively did your group mate work with you?	5	5	5
Contribution in writing the report	5	5	5
Demonstrates a cooperative and supportive attitude.	5	5	5
Contributes significantly to the success of the project .	5	5	5
TOTAL	20	20	20

Also please state the overall contribution of your teammate in percentage below, with total of all the three members accounting for 100% (33.33+33.33+33.33 ~ 100%) :

Group member 1 :

33.333

Group member 2 :

33.33

Group member 3 :

33.33