Ilan Coronel
Itay Israelov

Short Documentation

Mandatory Assignments:

Stage 1 – Prepare the data:

First, we load the original data and remove the features that were not selected (the features which are not in the "right features" set in the assignment file).
Then, we execute the data transformations from the previous assignment, for instance nominal_to_numerical_categories, complete_missing_values, remove_outliers and normalize.

Stage 2 – Train and test some models via cross-validation :

We have chosen the following models: SGDClassifier, KNeighborsClassifier, DecisionTreeClassifier, RandomForestClassifier.
And after a small test we saw that, the best accuracy is obtained with the DecisionTreeClassifier and RandomForestClassifier.
As part of the tests, we try to give different hyperparameters to get the best accuracy. For example: in the knn algorithm we change the number of neighbors from 3 to 4, 5 and more…
Also, we change the hyperparameters of the DecisionTreeClassifier and RandomForestClassifier algorithms, for example, criterion "entropy" and "Gini", min_samples_split for different values, and min_samples_leaf for different values.
Eventually, we get to the following conclusions:

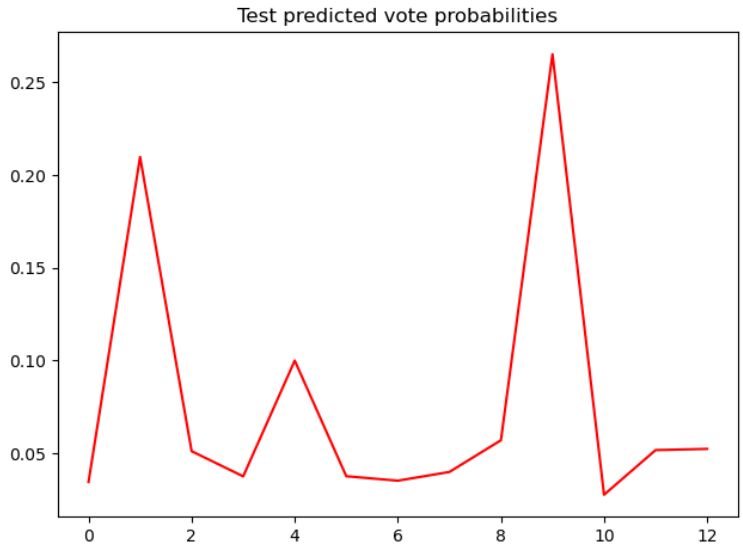
1. For DecisionTreeClassifier the best hyperparameters is: entropy, min_samples_split=5 min_samples_leaf=1
2. For RandomForestClassifier the best hyperparameters is: entropy, min_samples_split=3 min_samples_leaf=1

Finally, the algorithm that gives us the best accuracy is: RandomForestClassifier

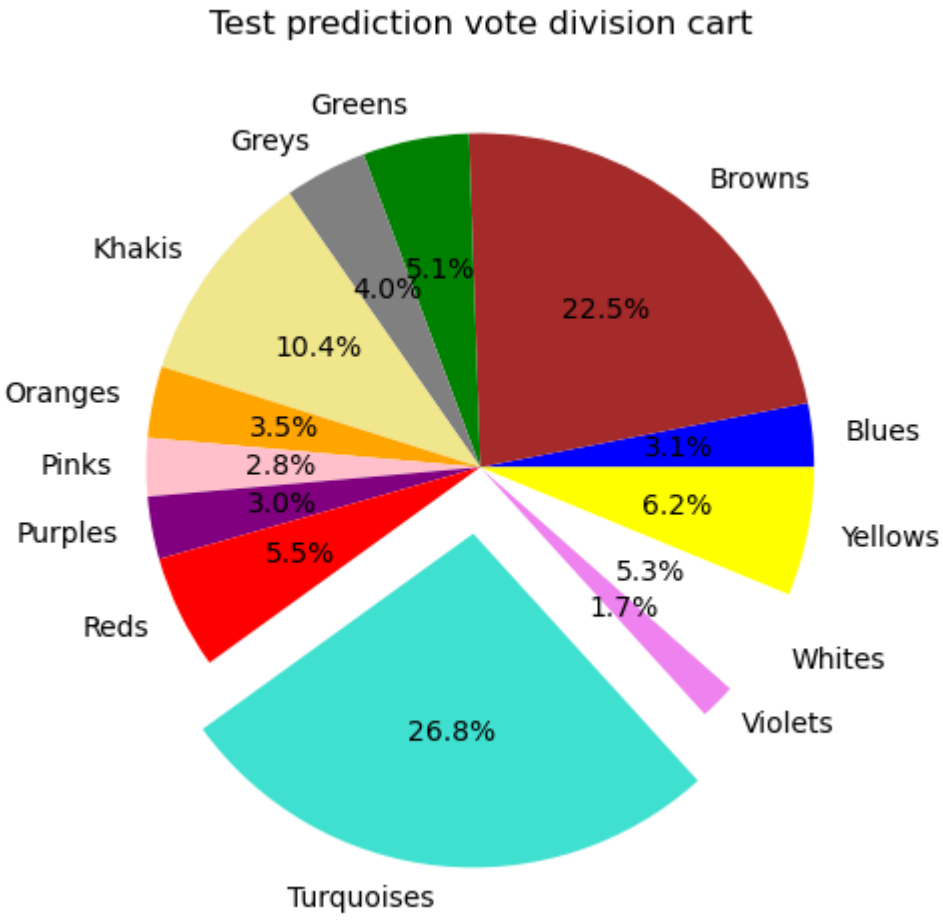
Stage 3 – The results:

The party that wins the elections is: ***'Turquoises'***.
We use the predict_proba function to choose the winner of the elections. We compute the "mean" of the columns in the result of predict_proba function and take the label, who gives us the max probability. Finally, we plot this graph and get these results:

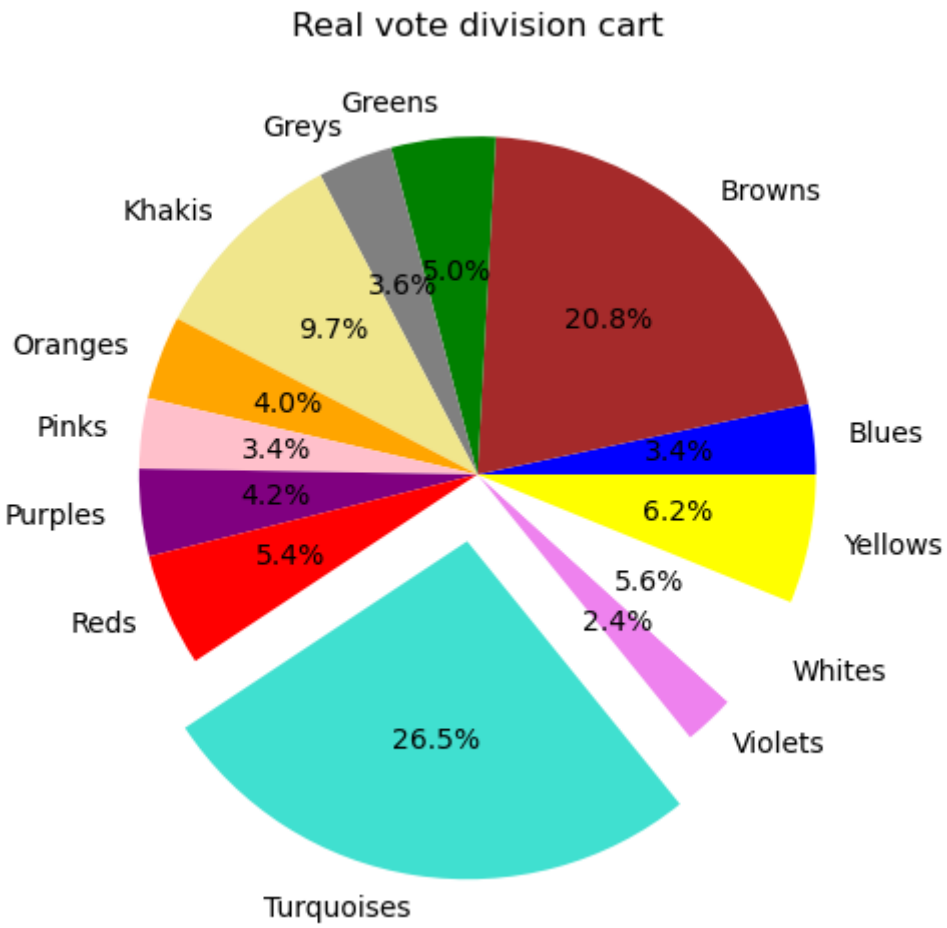
Ilan Coronel
Itay Israelov



The division of voters between the various parties is:



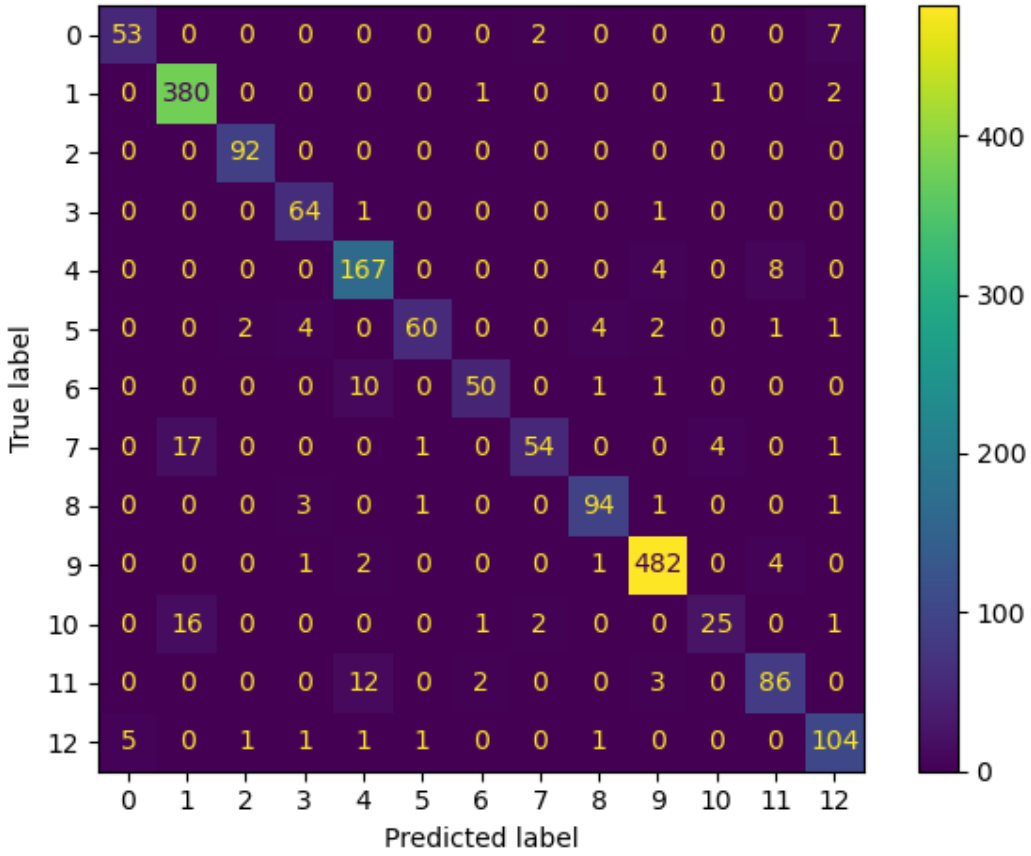
Ilan Coronel
Itay Israelov



The CSV file that contain the voting predictions (predicted labels) on the test set is: 'test_voting_predictions.csv'

Ilan Coronel
Itay Israelov

The (test) confusion matrix is:



Test error for the predicted votes is: 0.0779*100 = 7.79%

Ilan Coronel
Itay Israelov

Stage 4 - Transportation services:

We use the predict_proba function to compute the probability that each citizen votes for some party, and if the probability is greater than the transportation threshold (which we estimated to be 0.8), we add this citizen to the result list of a specific label.

We get the following results:

{'Blues': [90, 138, 246, 333, 346, 394, 405, 450, 495, 526, 529, 530, 569, 573,
642, 705, 762, 804, 818, 859, 928, 944, 978, 982, 1032, 1048, 1062, 1146, 1289,
1362, 1525, 1538, 1548, 1599, 1614, 1755, 1772, 1790, 1823, 1838],
'Browns': [6, 8, 13, 15, 17, 32, 41, 46, 60, 66, 76, 88, 89, 95, 96, 101,
108, 116, 130, 135, 151, 152, 153, 154, 156, 159, 166, 172, 173, 175, 177,
178, 179, 186, 190, 191, 196, 197, 213, 219, 228, 258, 274, 280, 286, 296,
300, 304, 306, 307, 308, 312, 321, 324, 334, 336, 351, 357, 358, 362, 369,
370, 376, 386, 388, 404, 423, 429, 432, 437, 440, 443, 444, 446, 455, 456,
459, 462, 464, 474, 477, 482, 487, 498, 502, 506, 520, 522, 534, 539, 541,
544, 546, 549, 552, 555, 559, 563, 585, 591, 592, 595, 605, 615, 618, 621,
624, 631, 660, 667, 668, 669, 674, 684, 691, 692, 694, 697, 710, 713, 726,
728, 730, 734, 735, 736, 744, 745, 746, 748, 754, 756, 763, 769, 777, 779,
785, 786, 789, 793, 803, 810, 815, 817, 824, 828, 832, 846, 852, 854, 861,
871, 876, 881, 894, 895, 901, 906, 918, 931, 937, 938, 939, 959, 970, 972,
995, 997, 1001, 1013, 1015, 1019, 1023, 1031, 1040, 1042, 1049, 1065, 1071,
1072, 1074, 1077, 1081, 1085, 1086, 1090, 1094, 1099, 1108, 1113, 1124, 1125,
1130, 1132, 1144, 1148, 1149, 1157, 1158, 1169, 1172, 1176, 1181, 1188, 1194,
1195, 1197, 1206, 1230, 1231, 1236, 1240, 1249, 1259, 1268, 1277, 1288, 1298,
1305, 1307, 1322, 1323, 1327, 1337, 1346, 1348, 1349, 1355, 1357, 1363, 1373,
1374, 1377, 1394, 1395, 1407, 1413, 1422, 1425, 1427, 1431, 1432, 1434, 1437,
1455, 1456, 1470, 1494, 1501, 1510, 1516, 1521, 1541, 1546, 1552, 1554, 1555,
1564, 1569, 1570, 1581, 1588, 1590, 1592, 1597, 1603, 1630, 1639, 1642, 1659,
1665, 1668, 1672, 1680, 1684, 1691, 1698, 1703, 1705, 1706, 1711, 1716, 1718,
1725, 1726, 1729, 1734, 1744, 1748, 1749, 1763, 1765, 1769, 1777, 1783, 1784,
1788, 1802, 1808, 1809, 1817, 1819, 1842, 1843],
'Greens': [16, 38, 78, 123, 133, 137, 140, 161, 206, 210, 216, 220, 225,
230, 231, 232, 250, 330, 350, 384, 403, 447, 478, 507, 510, 577, 633, 687,
714, 753, 761, 784, 800, 807, 808, 813, 842, 856, 864, 875, 889, 890, 897,
950, 955, 964, 986, 1038, 1053, 1092, 1096, 1097, 1165, 1167, 1177, 1244,
1256, 1257, 1264, 1269, 1335, 1356, 1418, 1428, 1453, 1454, 1474, 1478, 1519,
1559, 1578, 1580, 1582, 1609, 1625, 1636, 1649, 1655, 1670, 1720, 1727, 1742,
1770, 1771, 1821, 1839],
'Greys': [37, 168, 169, 249, 260, 261, 328, 340, 365, 372, 381, 421, 476, 485,
523, 564, 604, 654, 703, 820, 896, 917, 957, 977, 1191, 1266, 1267, 1285, 1291,
1296, 1354, 1419, 1448, 1463, 1520, 1523, 1607, 1692, 1713, 1759, 1795, 1847],
'Khakis': [20, 21, 39, 42, 43, 52, 69, 77, 112, 128, 129, 149, 194, 203, 204, 205,
207, 208, 209, 212, 224, 243, 256, 257, 272, 299, 311, 332, 361, 385, 433, 436,
445, 486, 494, 517, 519, 527, 543, 567, 575, 587, 589, 593, 594, 597, 620, 630, 635,
645, 646, 648, 662, 671, 681, 699, 704, 732, 740, 747, 760, 781, 787, 802, 849, 866,
873, 883, 884, 892, 915, 929, 940, 951, 953, 960, 1005, 1030, 1043, 1064, 1087, 1089,
1119, 1145, 1173, 1186, 1190, 1201, 1221, 1225, 1229, 1252, 1253, 1276, 1292, 1386,
1389, 1401, 1402, 1424, 1469, 1473, 1495, 1498, 1508, 1511, 1526, 1529, 1556, 1557,
1560, 1565, 1567, 1587, 1589, 1593, 1594, 1601, 1620, 1671, 1696, 1697, 1712, 1714,

Ilan Coronel
Itay Israelov

1750, 1757, 1791, 1793, 1801, 1822, 1824, 1825, 1830],
'Oranges': [19, 61, 269, 452, 500, 504, 609, 664, 702, 848, 949, 1020, 1114, 1139, 1171,
1192, 1222, 1246, 1250, 1321, 1333, 1776], 'Pinks': [27, 262, 310, 345, 382, 410, 418,
441, 467, 554, 582, 711, 801, 863, 933, 935, 979, 991, 996, 1007, 1045, 1127, 1138,
1183, 1273, 1275, 1446, 1490, 1527, 1533, 1566, 1633, 1657, 1693, 1699, 1721, 1816,
1828, 1837],
'Purples': [147, 349, 1406],
'Reds': [1, 26, 54, 63, 70, 73, 119, 192, 237, 267, 273, 318, 327, 329, 338, 339, 355,
356, 373, 431, 513, 566, 579, 649, 650, 659, 700, 716, 722, 724, 749, 773, 790,
835, 907, 910, 919, 930, 936, 973, 1024, 1035, 1109, 1131, 1162, 1175, 1204,
1216, 1234, 1238, 1241, 1260, 1274, 1283, 1295, 1297, 1302, 1306, 1316, 1331,
1340, 1351, 1372, 1391, 1411, 1421, 1482, 1485, 1492, 1503, 1515, 1550, 1574,
1577, 1591, 1631, 1643, 1661, 1662, 1701, 1723, 1737, 1745, 1758, 1786, 1792],
'Turquoises': [0, 4, 9, 10, 11, 12, 25, 30, 33, 36, 44, 48, 49, 50, 51, 65, 72,
74, 75, 82, 85, 86, 91, 92, 94, 100, 104, 111, 113, 118, 122, 124, 125, 132, 139,
141, 144, 146, 150, 158, 160, 164, 170, 176, 184, 185, 189, 193, 199, 223, 226,
229, 233, 235, 239, 240, 242, 245, 252, 259, 263, 265, 268, 270, 275, 276, 283,
284, 285, 288, 291, 294, 303, 313, 319, 322, 325, 326, 331, 337, 347, 352, 353,
354, 378, 383, 389, 392, 393, 396, 397, 400, 401, 406, 407, 408, 409, 413, 414,
415, 424, 428, 438, 448, 458, 468, 470, 479, 480, 489, 490, 492, 493, 511, 512,
514, 521, 528, 540, 547, 553, 556, 560, 561, 568, 578, 580, 581, 583, 584, 586,
590, 598, 600, 606, 611, 612, 613, 614, 616, 617, 625, 626, 628, 632, 634, 637,
641, 643, 647, 653, 657, 658, 661, 675, 678, 679, 680, 685, 690, 695, 701, 708,
709, 712, 715, 717, 720, 725, 727, 729, 731, 733, 738, 750, 755, 764, 765, 768,
772, 776, 778, 783, 794, 796, 805, 806, 812, 821, 822, 826, 827, 830, 833, 834,
839, 840, 841, 843, 853, 855, 862, 879, 880, 886, 893, 898, 899, 911, 912, 921,
922, 923, 926, 932, 946, 958, 968, 971, 980, 981, 984, 985, 987, 988, 990, 994,
999, 1000, 1004, 1006, 1010, 1014, 1017, 1021, 1025, 1028, 1029, 1034, 1036,
1037, 1039, 1041, 1046, 1047, 1050, 1051, 1055, 1058, 1060, 1067, 1068, 1078,
1079, 1083, 1084, 1091, 1093, 1095, 1098, 1100, 1103, 1111, 1112, 1118, 1122,
1135, 1143, 1150, 1152, 1154, 1156, 1161, 1174, 1178, 1179, 1180, 1184, 1187,
1198, 1200, 1207, 1209, 1210, 1212, 1213, 1214, 1215, 1226, 1232, 1233, 1235,
1237, 1243, 1255, 1258, 1265, 1271, 1282, 1284, 1286, 1293, 1294, 1304, 1308,
1311, 1313, 1315, 1318, 1324, 1325, 1328, 1329, 1334, 1338, 1342, 1343, 1344,
1345, 1347, 1352, 1359, 1360, 1361, 1366, 1367, 1369, 1371, 1375, 1376, 1381,
1383, 1385, 1387, 1390, 1393, 1396, 1399, 1400, 1404, 1405, 1409, 1420, 1430,
1443, 1451, 1457, 1460, 1466, 1467, 1471, 1475, 1476, 1483, 1484, 1487, 1491,
1499, 1502, 1504, 1506, 1507, 1514, 1517, 1518, 1522, 1528, 1530, 1532, 1534,
1535, 1537, 1539, 1542, 1543, 1547, 1549, 1558, 1563, 1568, 1572, 1573, 1576,
1579, 1583, 1584, 1595, 1602, 1605, 1608, 1611, 1616, 1621, 1622, 1623, 1624,
1627, 1629, 1634, 1635, 1641, 1656, 1660, 1663, 1664, 1674, 1676, 1678, 1679,
1686, 1687, 1695, 1700, 1708, 1709, 1715, 1717, 1724, 1731, 1740, 1743, 1747,
1754, 1756, 1762, 1768, 1773, 1774, 1778, 1779, 1780, 1782, 1787, 1789, 1799,
1800, 1805, 1806, 1807, 1812, 1813, 1814, 1820, 1829, 1833, 1835, 1836, 1846],
'Violets': [292],
'Whites': [62, 163, 200, 298, 391, 508, 537, 576, 599, 673, 677, 775,
782, 850, 869, 956, 998, 1018, 1052, 1088, 1202, 1205, 1228, 1261, 1310,
1320, 1436, 1472, 1551, 1553, 1652, 1702, 1707, 1751, 1794, 1804, 1832, 1845],
'Yellows': [106, 364, 475, 652, 676, 739, 741, 829, 867, 868, 1070, 1141, 1185,
1287, 1412, 1447, 1449, 1486, 1585, 1606, 1644, 1645, 1732, 1811]}

Ilan Coronel
Itay Israelov

Non-Mandatory Assignments:

A.

We wrote a script that takes as an input the different models we trained and their accuracies, and that returns the model with the best accuracy.
The output we got is the following model :

```
RandomForestClassifier(n_jobs=-1, random_state=0, criterion='entropy')
```

C.

After doing many manipulations, we found out that modifying the Yearly_IncomeK and the Number_of_differnt_parties_voted_for has no impact on the elections' winner.
However, every other modifications has an impact:

Feature	Operation	Winner
Political_interest_Total_Score	+24	Browns
Avg_Satisfaction_with_previous_vote	+2	Browns
Avg_monthly_income_all_years	+0.5	Browns
Most_Important_Issue	+4	Browns
Overall_happiness_score	+3	Browns
Avg_size_per_room	+3	Purples
Weighted_education_rank	+3	Browns

(The operations were made on the prepared test data set)

Widrow-Hoff Assignement:

Incomplete. A tentative of implementation is present in Widrow-Hoff.py