הטכניון, מכון טכנולוגי לישראל

הפקולטה למדעי המחשב

מבוא למערכות לומדות

236756

סמסטר אביב תשע"ט

תרגיל מספר: 3

תאריך הגשה: 30/05/19

תא להחזרה: 68

מגישים:

עידו יחזקאל	8 6 3 7 9 3 4 0 2	idoye
שם מלא	מספר ת.ז.	דואר אלקטרוני ב- t2

אמיר אביבי	3 7 8 3 8 1 5 0 3	saavivi
שם מלא	מספר ת.ז.	דואר אלקטרוני ב- t2

Mandatory Part – Modeling:

ראשית כמתבקש טענו את הקובץ הדוגמאות שוב, והחלנו עליו את מניפולציות העיבוד המקדים להכנת המידע כפי שביצענו בתרגיל הקודם, מניפולציות אלה כוללות:

1. חלוקת סט הדוגמאות הכולל (על די Stratified Shuffle Split) לשלושה סטים:

Data Set	Percentage from Original Data Set
Train set	65%
Validating set	10%
Test set	25%

2. בחירת סט הפיצ'רים הנכון כפי שנבחר בתרגיל הנוכחי.
3. הוצאות ערכים שהינם outliers, לדוגמא ערכים שליליים.
4. השלמת ערכים חסרים לפי השיטות המקובלות: closest fit, feature correlation, mean and majority.
5. ביצוע נורמליזציה לערכים קטיגוריאליים וZ-scale לערכים נומינליים.
6. ייצוא המידע ל 3X2 קבצי CSV לפני ואחרי השנויים.

לאחר מכן, ניגשנו למשימת החיזוי כאשר בחלק החובה נדרשנו לחזות:

- מה היא המפלגה המנצחת?
- התפלגות חלוקת הקולות בין המצביעים
- אספקת שירותי הסעה למצביעים של כל מפלגה.

על מנת להתמודד עם משימות אלו נרצה למצוא את המסווג הטוב ביותר מבין קבוצת מסווגים ולכן ביצענו את התהליך הבא:

1. בחירת סוגי המסווגים איתם נרצה לבצע את התחזיות:

a. Random Forest Classifier- אלגוריתם ועדה, כאשר הסיווג נקבע על ידי הצבעת הרוב של עצי החלטה המשתתפים ב"יער" הנבנה, בחרנו בו מכיוון שהוא ידוע כאלגוריתם חזק, אשר מכליל את עקרון עצי החלטה ומכיוון שעצי החלטה היא משפחת אלגוריתמים שנלמדו בקורס ובקורס הקודם.

b. Stochastic Gradient Descent Classifier – בחרנו במודל זה מכיוון שאנו מכירים אותו ואת שיטת הפעולה שלו ומכיוון שרצינו לדעת האם קיים הפרדה לינארית לבעיה.

c. K- Nearest Neighbors – בחרנו במודל זה מכיוון שאנו מכירים אותו ואת שיטת הפעולה שלו, בנוסף מודל זה פשוט ביחס למודלים אחרים וקל יותר להבנה. בנוסף מתוך הנחה כי בני אדם בעלי אותם מאפיינים נוטים לבחור באופן דומה הרי שכדאי לבדוק את ביצועיו על המידע.

d. Decision Tree- עץ החלטה רגיל כפי שנלמד בהרצאה, בחרנו במודל זה למרות שהשתמשנו ביער של עצים מכיוון שרצינו לראות את ההבדל בביצועים בין Random Forest Classifier לבין מסווג זה וגם הצלחנו לצייר את עץ ההחלטה שנוצר.

2. בחירת הפרמטרים הטובים ביותר עבור כל סוג מסווג שנבחר. עבור כל אחד מסוגי מהמסווגים שהוזכר לעיל בנינו כמה דוגמאות ממנו עם סטים של היפר-פרמטרים שונים והערכנו את ביצועיו לפי מדד הדיוק(accuracy).

a.
```
random_forest_tuple = (

RandomForestClassifier(random_state=0, criterion='entropy',
min_samples_split=5, min_samples_leaf=3, n_estimators=50),

RandomForestClassifier(random_state=0, criterion='entropy',
min_samples_split=3, min_samples_leaf=1, n_estimators=500),

RandomForestClassifier(random_state=0, criterion='gini',
min_samples_split=3, min_samples_leaf=1, n_estimators=500)

 )
```

b.
```
sgd_tuple = (

SGDClassifier(random_state=0, max_iter=1000, tol=1e-3),

SGDClassifier(random_state=0, max_iter=1000, tol=1e-2),

SGDClassifier(random_state=0, max_iter=1500, tol=1e-4),
)
```

c.
```
knn_tuple = (

KNeighborsClassifier(n_neighbors=3, algorithm='auto'),

KNeighborsClassifier(n_neighbors=5, algorithm='auto'),
)
```

d.
```
tree_tuple = (

DecisionTreeClassifier(random_state=0, criterion='gini',
min_samples_split=5, min_samples_leaf=3),

DecisionTreeClassifier(random_state=0, criterion='entropy',
min_samples_split=3, min_samples_leaf=1)
)
```

הערכת כל מודל התבצעה באמצעות שיטת k- fold cross validation על סט האימון כאשר k = 5.

3. לאחר בחירת סט הפרמטרים הטוב ביותר לכל סוג מסווג, בחירת סוג המסווג הטוב ביותר התבצעה באמצעות בחינת הביצועים של המסווג על פני סט הוולידציה.
נדגיש כי מבחינתו הביצועים של המסווג הינם אחוז הדיוק (accuracy) על פני סט המבחן.
בחרנו דווקא בדיוק כמדד להערכת ביצועים מכיוון שהתחזיות המפלגה המנצחת וחלוקת הקולות בין המפלגות אנו זקוקים לדיוק גבוהה על מנת לספק תחזיות תואמות לסט הדוגמאות.
התוצאות שקבלנו:

Random Forest Classifier accuracy score on validation set is: 90.1%

SGD Classifier accuracy score on validation set is: 74.7%

KNN Classifier accuracy score on validation set is: 77.5%

Decision Tree Classifier accuracy score on validation set is: 85.8%

ולכן המסווג הנבחר הינו Random Forest Classifier, עם סט הפרמטרים הבאים:

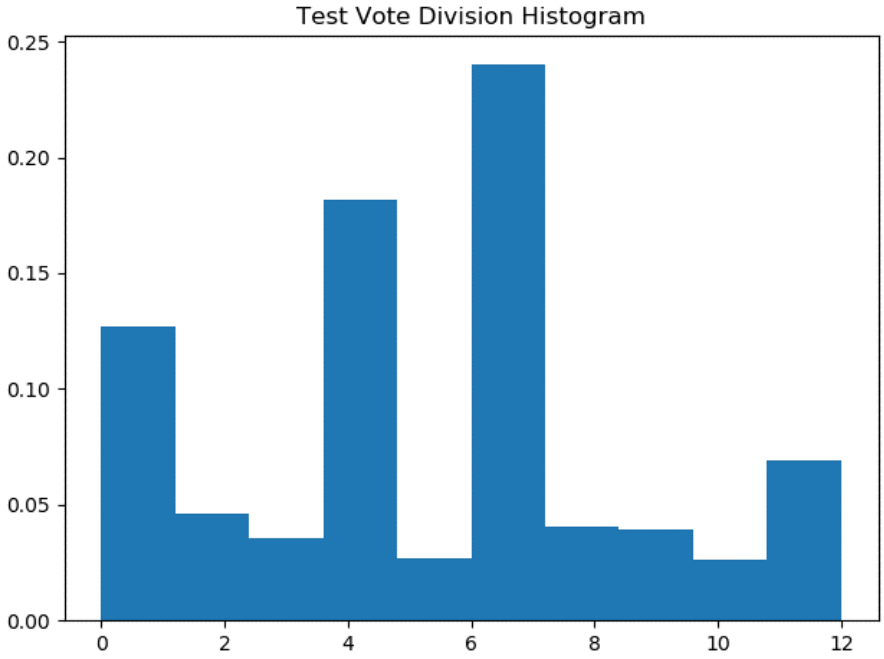
```
RandomForestClassifier(random_state=0, criterion='gini;, n_samples_split=3,
min_samples_leaf=1, n_estimators=500)
```

4. ביצוע התחזיות התבצע על ידי המסווג הנבחר כאשר הוא אימנו אותו מחדש על סט האימון וסט הוולידציה ביחד. המוטיבציה לבצע זאת היא מכיוון שכבר השתמשנו בסט הוולידציה על מנת להעריך באיזה מסווג להשתמש ואנו לא מתכוונים להשתמש בו שוב וכמובן שסט הוולידציה מכיל דוגמאות שנרצה שהמסווג שלנו יתאמן, באופן כללי ככל שסט האימון גדול יותר כך המסווג טוב יותר.
<u>להלן ביצועי המסווג על סט המבחן דיוק של 92.72% ושגיאה של 7.28%.</u>
כפי שניתן לשים לב קיבלנו דיוק יותר גבוה על סט המבחן מאשר בבחינת ביצועי המסווג ע"י k- fold cross validation וגם מאשר בחינת ביצועי המסווג על סט הוולידציה דבר המעיד כי המסווג אינו סובל מ overfitting על סט האימון שלו.

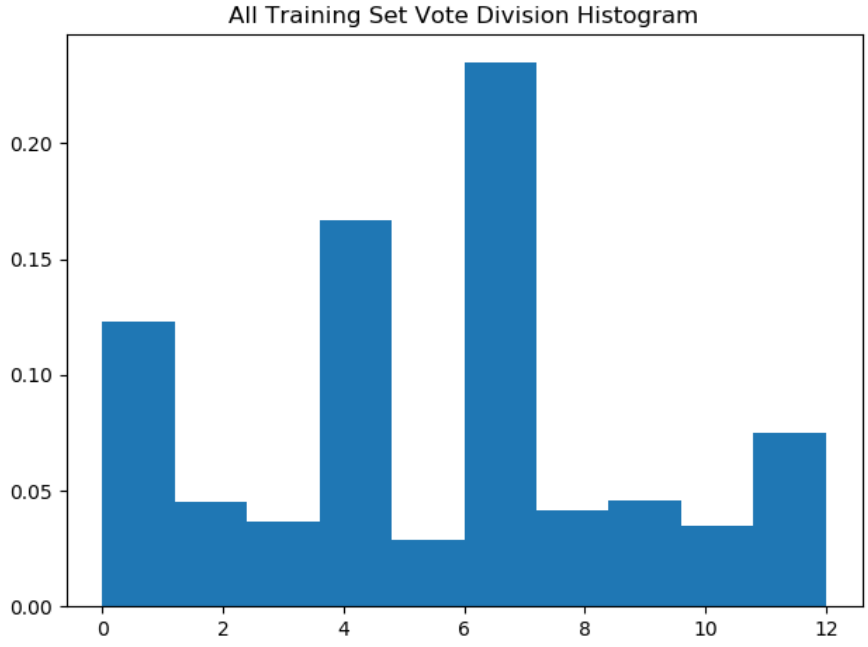
5. כעת נסביר כיצד פעלנו עבור כל אחת מהתחזיות שנתבקשנו לבצע:
   a. עבור תחזית המפלגה המנצחת ביצענו חיזוי על סט המבחן ובחרנו במפלגה שקיבלה הכי הרבה קולות, ולפי המסווג שלנו המפלגה המנצחת הינה: הסגולים 😊.
   b. עבור חלוקת הקולות בין המפלגות ביצענו חיזוי על סט המבחן ובנינו היסטוגרמה לפי המפלגות:

Color	Vote Percentage
Blues	5.44%
Browns	9.8%
Greens	5.52%
Greys	4.27%
Khakis	**21.84%**
Oranges	3.24%
Pinks	2.92%
Purples	**25.92%**
Reds	4.84%
Turquoises	4.72%
Violets	3.16%
Whites	4.52%
Yellows	3.8%

חלוקת קולות המצביעים כפי שחזה המסווג:



חלוקת קולות המצביעים לפי סט האימון:



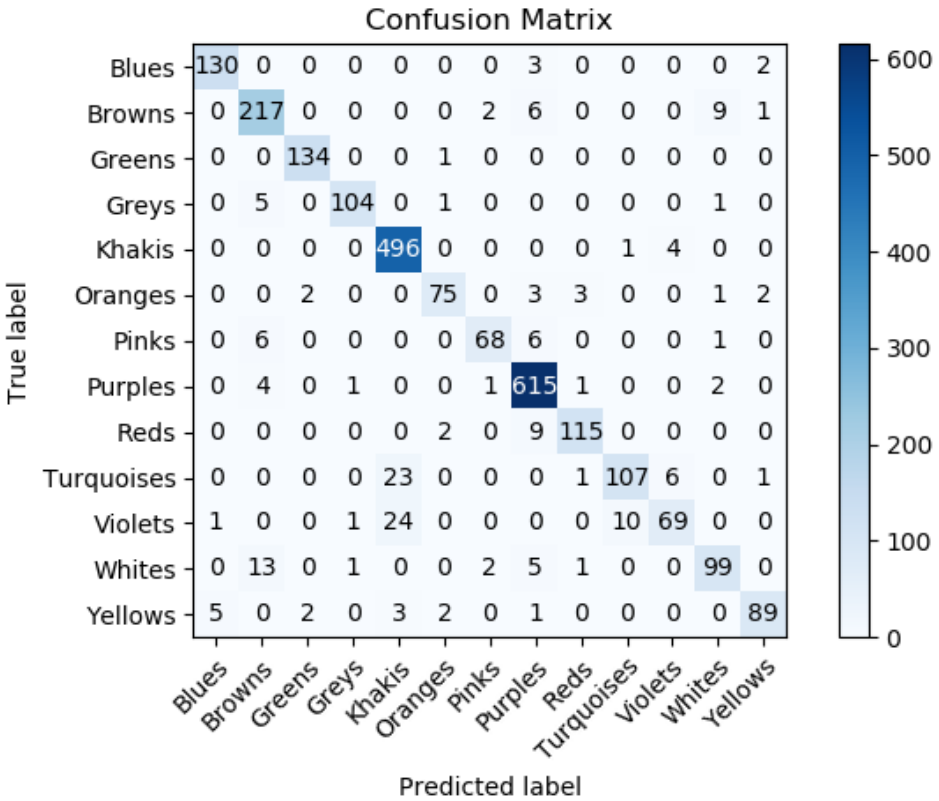
ניתן להבחין כי תוצאות המסווג אינן תואמות לתוצאות סט האימון במפלגות 8 ו-9 כלומר בצבעים טורקיז ואדום.

c. עבור שירות ההסעות למצביעים ביצענו חיזוי על סט המבחן כאשר במקום לחזות סיווג יחיד לכל מצביע חזינו מהי ההסתברות שיצביע לכל אחת מהמפלגות. לאחר מכן קבענו סף עבור שירות ההסעות אשר קובע כי בהינתן מצביע וסיכוי ההצבעה שלו למפלגה מסוימת האם המפלגה צריכה לספק עבורו שירות הסעות. את הסף קבענו ל 60%, להלן שירות ההסעות של כל מפלגה:

- Blues: [1, 4, 37, 66, 70, 100, 120, 132, 133, 167, 171, 186, 212, 244, 246, 258, 271, 275, 317, 336, 338, 391, 405, 420, 423, 425, 434, 453, 477, 561, 564, 644, 704, 782, 784, 786, 840, 903, 905, 938, 944, 945, 977, 993, 996, 1051, 1055, 1070, 1092, 1117, 1143, 1144, 1181, 1207, 1214, 1225, 1245, 1263, 1289, 1298, 1345, 1372, 1393, 1397, 1432, 1434, 1435, 1478, 1523, 1545, 1566, 1595, 1599, 1621, 1633, 1663, 1670, 1691, 1723, 1734, 1737, 1742, 1764, 1770, 1845, 1852, 1876, 1937, 1951, 1953, 1989, 2043, 2064, 2081, 2093, 2099, 2151, 2152, 2160, 2161, 2213, 2224, 2235, 2240, 2261, 2313, 2314, 2328, 2359, 2362, 2385, 2427, 2466, 2468, 2487]
- Reds: [11, 17, 43, 95, 118, 136, 188, 192, 247, 263, 266, 270, 319, 354, 361, 388, 397, 401, 449, 539, 551, 585, 667, 700, 701, 721, 723, 813, 824, 885, 897, 952, 966, 1036, 1056, 1058, 1081, 1098, 1141, 1161, 1203, 1230, 1264, 1267, 1269, 1292, 1307, 1316, 1318, 1319, 1354, 1416, 1455, 1480, 1521, 1534, 1543, 1557, 1564, 1570, 1600, 1608, 1651, 1715, 1719, 1729, 1755, 1776, 1786, 1794, 1805, 1879, 1918, 1921, 1922, 1932, 1940, 1946, 1956, 1966, 1972, 2061, 2072, 2073, 2082, 2105, 2133, 2226, 2242, 2333, 2372, 2377, 2397, 2401, 2408, 2414, 2423, 2425, 2433, 2458, 2462, 2493]
- Khakis: [2, 8, 14, 16, 26, 29, 33, 34, 35, 38, 48, 50, 55, 68, 72, 76, 79, 82, 85, 90, 91, 96, 102, 103, 105, 112, 124, 126, 137, 139, 147, 148, 154, 157, 166, 168, 178, 180, 185, 189, 194, 198, 199, 211, 213, 224, 230, 231, 238, 242, 245, 251, 267, 269, 278, 279, 283, 286, 290, 303, 304, 305, 327, 331, 332, 342, 355, 369, 376, 385, 387, 396, 398, 406, 408, 410, 419, 438, 439, 448, 451, 454, 459, 460, 467, 473, 474, 475, 489, 492, 500, 507, 522, 523, 532, 534, 554, 555, 562, 565, 571, 579, 584, 590, 591, 596, 599, 609, 610, 613, 619, 620, 624, 630, 632, 635, 657, 658, 663, 664, 668, 685, 690, 692, 693, 694, 696, 703, 717, 722, 726, 732, 743, 744, 748, 760, 766, 785, 794, 797, 800, 804, 805, 810, 822, 833, 847, 850, 851, 857, 860, 876, 883, 893, 894, 900, 904, 916, 920, 922, 923, 924, 927, 930, 940, 947, 951, 960, 970, 972, 979, 982, 988, 992, 995, 1002, 1004, 1012, 1019, 1026, 1032, 1035, 1040, 1044, 1054, 1059, 1066, 1067, 1068, 1074, 1080, 1087, 1089, 1091, 1093, 1096, 1102, 1103, 1104, 1114, 1118, 1121, 1129, 1130, 1132, 1137, 1149, 1155, 1159, 1165, 1174, 1178, 1185, 1189, 1193, 1206, 1209, 1213, 1228, 1234, 1239, 1240, 1250, 1251, 1258, 1265, 1266, 1274, 1276, 1278, 1284, 1287, 1291, 1297, 1304, 1308, 1312, 1315, 1317, 1320, 1327, 1334, 1342, 1350, 1361, 1368, 1370, 1388, 1395, 1401, 1403, 1407, 1412, 1417, 1420, 1422, 1436, 1441, 1444, 1457, 1464, 1470, 1474, 1482, 1500, 1501, 1502, 1508, 1509, 1510, 1520, 1531, 1536, 1542, 1552, 1556, 1567, 1580, 1582, 1584, 1586, 1588, 1589, 1590, 1591, 1593, 1597, 1598, 1601, 1602, 1609, 1610, 1619, 1625, 1636, 1641, 1654, 1662, 1679, 1687, 1697, 1704, 1714, 1725, 1743, 1744, 1746, 1767, 1769, 1772, 1775, 1799, 1801, 1815, 1817, 1823, 1834, 1843, 1846, 1854, 1855, 1857, 1860, 1869, 1884, 1889, 1913, 1914, 1916, 1923, 1924, 1925, 1960, 1969, 1985, 1988, 1991, 1997, 2010, 2019, 2023, 2030, 2035, 2036, 2047, 2049, 2053, 2054, 2058, 2066, 2069, 2086, 2096, 2097, 2109, 2110, 2122, 2135, 2145, 2153, 2163, 2165, 2166, 2171, 2172, 2173, 2176, 2179, 2181, 2183, 2184, 2186, 2187, 2193, 2203, 2204, 2207, 2208, 2211, 2218, 2219, 2248, 2256, 2258, 2259, 2260, 2262, 2263, 2268, 2274, 2294, 2307, 2308, 2311, 2315, 2322, 2323, 2334, 2336, 2340, 2366, 2379, 2382, 2383, 2384, 2386, 2387, 2391, 2406, 2415, 2418, 2422, 2441, 2442, 2445, 2453, 2464, 2472, 2492, 2496]
- Purples: [0, 3, 6, 13, 15, 20, 22, 25, 32, 39, 52, 54, 57, 58, 62, 64, 67, 69, 83, 86, 93, 99, 101, 104, 107, 110, 117, 119, 121, 123, 125, 130, 131, 142, 151, 155, 156, 159, 163, 164, 169, 173, 174, 176, 187, 193, 202, 203, 207, 208, 215, 218, 226, 228, 240, 241, 254, 256, 261, 264, 289, 295, 296, 301, 302, 309, 311, 312, 313, 321, 326, 328, 335, 347, 356, 357, 368, 370, 371, 373, 392, 394, 395, 399, 402, 404, 409, 413, 417, 418, 422, 424, 428, 432, 435, 440, 446, 447, 450, 455, 461, 470, 471, 479, 491, 501, 506, 509, 510, 514, 516, 527, 529, 531, 538, 541, 547, 553, 557, 569, 572, 573, 577, 578, 582, 586, 589, 593, 594, 595, 606, 607, 615, 616, 621, 625, 628, 629, 631, 634, 637, 641, 642, 643, 645, 648, 649, 650, 652, 654, 655, 665, 666, 669, 671, 673, 683, 684, 686, 689, 699, 706, 707, 711, 712, 713, 720, 724, 727, 728, 731, 735, 736, 738, 740, 741, 742, 746, 749, 751, 753, 757, 765, 767, 769, 774, 776, 777, 780, 781, 783, 789, 795, 796, 799, 802, 811, 812, 820, 825, 831, 834, 849, 861, 863, 864, 865, 867, 873, 874, 878, 889, 892, 901, 908, 909, 910, 913, 915, 921, 925, 929, 932, 934, 936, 948, 950, 958, 965, 976, 984, 989, 999, 1006, 1016, 1020, 1021, 1024, 1025, 1030, 1031, 1033, 1038, 1041, 1042, 1047, 1050, 1052, 1069, 1072, 1073, 1075, 1077, 1084, 1086, 1095, 1099, 1106, 1109, 1111, 1116, 1122, 1125, 1128, 1131, 1134, 1138, 1145, 1146, 1148, 1154, 1160, 1168, 1171, 1172, 1183, 1190, 1195, 1196, 1199, 1201, 1212, 1220, 1232, 1233, 1242, 1247, 1256, 1257, 1279, 1280, 1285, 1286, 1290, 1293, 1294, 1300, 1311, 1324, 1325, 1330, 1332, 1340, 1341, 1359, 1373, 1375, 1376, 1378, 1380, 1383, 1385, 1391, 1394, 1402, 1408, 1409, 1419, 1421, 1423, 1425, 1429, 1443, 1445, 1448, 1449, 1452, 1453, 1459, 1462, 1467, 1468, 1477, 1487, 1489, 1491, 1495, 1496, 1499, 1503, 1504, 1511, 1524, 1525, 1526, 1529, 1532, 1537, 1538, 1547, 1548, 1550, 1561, 1562, 1569, 1571, 1572, 1585, 1594, 1603, 1605, 1606, 1617, 1618, 1620, 1630, 1634, 1637, 1638, 1639, 1640, 1647, 1649, 1650, 1656, 1658, 1659, 1666, 1669, 1672, 1676, 1689, 1690, 1693, 1698, 1708, 1710, 1711, 1712, 1730, 1732, 1733, 1735, 1757, 1759, 1760, 1763, 1773, 1777, 1778, 1780, 1784, 1785, 1788, 1793, 1795, 1796, 1797, 1798, 1803, 1808, 1809, 1810, 1812, 1814, 1819, 1821, 1825, 1837, 1841, 1844, 1848, 1862, 1864, 1868, 1870, 1872, 1878, 1880, 1882, 1887, 1890, 1891, 1894, 1899, 1901, 1902, 1907, 1909, 1911, 1912, 1917, 1919, 1927, 1929, 1931, 1935, 1939, 1941, 1944, 1952, 1954, 1959, 1962, 1970, 1974, 1978, 1980, 1983, 1986, 1995, 1996, 1998, 1999, 2001, 2003, 2006, 2011, 2012, 2020, 2027, 2038, 2039, 2040, 2041, 2042, 2044, 2056, 2057, 2062, 2067, 2071, 2074, 2080, 2084, 2092, 2100, 2104, 2106, 2108, 2113, 2116, 2123, 2124, 2129, 2137, 2139, 2148, 2150, 2155, 2158, 2159, 2162, 2164, 2168, 2174, 2177, 2178, 2182, 2185, 2188, 2190, 2198, 2201, 2202, 2205, 2214, 2221, 2222, 2223, 2228, 2231, 2241, 2249, 2250, 2252, 2253, 2255, 2269, 2270, 2271, 2272, 2276, 2278, 2282, 2291, 2293, 2297, 2298, 2300, 2302, 2312, 2318, 2320, 2324, 2330, 2335, 2344, 2347, 2349, 2352, 2364, 2368, 2369, 2371, 2373, 2376, 2380, 2389, 2395, 2405, 2431, 2432, 2434, 2440, 2443, 2449, 2452, 2455, 2461, 2471, 2473, 2477, 2478, 2480, 2488, 2489, 2490, 2499]
- Whites: [182, 221, 222, 239, 277, 288, 292, 308, 329, 352, 367, 416, 431, 444, 463, 485, 494, 537, 662, 687, 719, 762, 775, 778, 846, 877, 891, 942, 1078, 1126, 1157, 1164, 1167, 1194, 1204, 1362, 1364, 1367, 1377, 1381, 1454, 1513, 1578, 1615, 1653, 1688, 1716, 1721, 1754, 1835, 1863, 1867, 1871, 2014, 2026, 2079, 2112, 2125, 2138, 2200, 2237, 2245, 2266, 2301, 2337, 2357, 2454]
- Browns: [5, 12, 24, 36, 42, 56, 74, 81, 109, 141, 144, 162, 172, 175, 201, 204, 227, 229, 233, 234, 255, 262, 268, 280, 316, 318, 320, 322, 337, 339, 350, 351, 362, 372, 429, 443, 458, 462, 465, 484, 486, 505, 508, 515, 567, 575, 581, 640, 672, 676, 678, 679, 681, 688, 718, 737, 745, 755, 788, 819, 828, 842, 844, 848, 858, 882, 888, 907, 912, 914, 918, 937, 941, 959, 961, 974, 986, 987, 1008, 1010, 1034, 1037, 1046, 1049, 1057, 1060, 1061, 1062, 1079, 1082, 1101, 1107, 1120, 1135, 1142, 1158, 1179, 1184, 1197, 1211, 1215, 1217, 1221, 1223, 1227, 1244, 1255, 1272, 1281, 1295, 1301, 1321, 1323, 1328, 1338, 1343, 1353, 1355, 1357, 1360, 1366, 1371, 1387, 1390, 1400, 1427, 1451, 1456, 1472, 1490, 1494, 1505, 1515, 1517, 1549, 1554, 1568, 1573, 1577, 1611, 1626, 1629, 1632, 1683, 1684, 1685, 1694, 1696, 1699, 1720, 1745, 1811, 1827, 1828, 1840, 1856, 1858, 1904, 1905, 1906, 1908, 1928, 1945, 1947, 1957, 1963, 1965, 1967, 1977, 1979, 1981, 1992, 2009, 2017, 2024, 2028, 2033, 2034, 2046, 2085, 2101, 2114, 2142, 2154, 2180, 2191, 2220, 2244, 2246, 2277, 2292, 2329, 2338, 2350, 2351, 2353, 2361, 2402, 2403, 2411, 2413, 2456, 2476, 2479]
- Greens: [7, 10, 23, 49, 51, 61, 140, 150, 158, 181, 197, 217, 249, 250, 252, 259, 260, 284, 365, 383, 407, 412, 430, 490, 521, 540, 558, 627, 636, 674, 698, 756, 779, 792, 798, 815, 827, 871, 872, 898, 931, 933, 935, 946, 949, 955, 969, 983, 1013, 1048, 1063, 1094, 1113, 1124, 1139, 1150, 1151, 1169, 1170, 1176, 1192, 1229, 1231, 1277, 1302, 1310, 1322, 1336, 1337, 1356, 1358, 1384, 1392, 1399, 1424, 1469, 1544, 1555, 1576, 1614, 1616, 1644, 1645, 1660, 1674, 1675, 1701, 1709, 1717, 1752, 1756, 1779, 1791, 1829, 1830, 1831, 1836, 1842, 1920, 1961, 1976, 1993, 2016, 2018, 2032, 2051, 2091, 2117, 2169, 2189, 2199, 2206, 2212, 2225, 2232, 2280, 2327, 2341, 2404, 2419, 2424, 2438, 2444, 2448, 2463, 2465, 2475, 2494, 2497]
- Greys: [84, 88, 135, 170, 291, 306, 324, 378, 403, 487, 495, 512, 513, 549, 601, 602, 614, 638, 691, 708, 709, 771, 843, 869, 899, 906, 957, 968, 1014, 1015, 1076, 1105, 1123, 1379, 1442, 1493, 1512, 1516, 1546, 1575, 1579, 1622, 1671, 1722, 1726, 1740, 1750, 1751, 1761, 1813, 1826, 1833, 1849, 1851, 1942, 1943, 1948, 1955, 1964, 2004, 2005, 2090, 2095, 2102, 2121, 2126, 2134, 2209, 2215, 2236, 2251, 2275, 2289, 2316, 2321, 2356, 2367, 2392, 2409, 2410, 2426]

- Yellows: [248, 293, 333, 345, 415, 519, 543, 546, 659, 677, 705, 807, 818, 884, 890, 896, 1000, 1001, 1011, 1028, 1133, 1180, 1191, 1222, 1260, 1329, 1347, 1415, 1458, 1479, 1507, 1596, 1655, 1661, 1703, 1790, 1807, 1838, 1850, 1930, 2077, 2078, 2088, 2156, 2243, 2345, 2421, 2498]
- Turquoises: [45, 78, 127, 232, 323, 504, 544, 568, 611, 623, 653, 680, 733, 801, 854, 868, 875, 886, 943, 953, 978, 985, 998, 1005, 1027, 1029, 1039, 1043, 1097, 1127, 1156, 1182, 1273, 1314, 1386, 1438, 1439, 1475, 1484, 1514, 1612, 1665, 1695, 1700, 1702, 1724, 1749, 1820, 1847, 1859, 1950, 1982, 2052, 2089, 2094, 2119, 2265, 2287, 2288, 2296, 2304, 2326, 2342, 2358, 2439, 2470]
- Pinks: [28, 30, 41, 128, 129, 179, 210, 225, 235, 276, 281, 325, 358, 377, 393, 426, 436, 488, 496, 497, 503, 517, 524, 548, 556, 587, 597, 697, 816, 826, 839, 1152, 1208, 1216, 1254, 1426, 1430, 1461, 1518, 1540, 1642, 1686, 1782, 1802, 1818, 1832, 1839, 1865, 1866, 1892, 1968, 2007, 2008, 2065, 2132, 2170, 2306, 2428]
- Oranges: [21, 89, 145, 253, 274, 340, 375, 445, 478, 570, 612, 656, 660, 695, 761, 887, 917, 954, 1188, 1333, 1346, 1348, 1404, 1405, 1418, 1440, 1450, 1541, 1565, 1574, 1667, 1692, 1800, 1893, 1895, 2068, 2075, 2115, 2127, 2264, 2267, 2281, 2305, 2332, 2343, 2348, 2388]
- Violets: [19, 94, 111, 153, 161, 196, 380, 400, 603, 675, 793, 806, 967, 1009, 1187, 1283, 1349, 1528, 1533, 1604, 1627, 1680, 1705, 1881, 1885, 1938, 2025, 2045, 2063, 2070, 2194, 2354]

6. להלן Confusion Matrix :



ניתן להבחין במגמות קריטיות של המסווג, לדוגמא סיווג כמות גדולה קולות לצבע חאקי במקום לצבע הטורקיז (מתן קולות של הצבע הטורקיז לחאקי הוא שגרם להבדל בהיסטוגרמה שראינו קודם) וסיגל. מתן כמות קולות לצבע הלבן במקום לחום ולהפך, כלומר **המסווג נוטה להתבלבל בין סיווג לצבע חום ולצבע לבן.**

First Bonus :

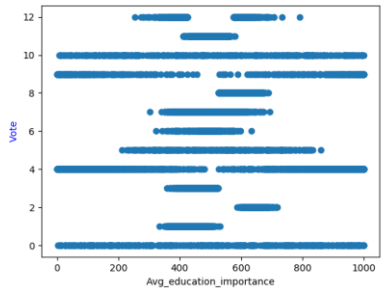
1. אוטומציה של בחירת המודל הטוב ביותר:
ראשית נדגיש כי מההתחלה ביצענו בחירה אוטומטית של המודל הנבחר על ידי מקסום מדד הדיוק על פני סט הוולידציה. בחרנו במדד הדיוק זאת מכיוון שאנו מבינים כי על מנת לספק את התחזיות של מפלגה מנצחת והתפלגות הקולות הרי שדיוק הינו המדד הכי חשוב ולכן בחנו במסווג אשר ממקסם מדד זה.
לאחר בחירת הפרמטרים עבור כל סוג מסווג, ביצענו באופן אוטומטי מעבר כל סוג המסווגים ונבחר סוג המסווג בעל אחוז הדיוק המקסימלי על פני סט הוולידציה, כפי שהוזכר קודם המסווג הנבחר הינו: Random Forest Classifier.
2. בסעיפי החובה בחרנו במדד הדיוק על מנת לבחור במסווג הטוב ביותר אמנם עבור כל אחת ממשימות החיזוי נבחר כעת מדד שנראה לנו הכי טוב לביצוע המשימה ולפיו נבחר את המסווג:
   a. לפי סט האימון ניתן להבחין כי המפלגה המנצחת היא הסגולים, לכן המדד בו בחרנו לביצוע המשימה הוא מדד בינארי, האם המסווג הצליח לנחש מה היא המפלגה המנצחת בחנו את אותם סוגי מודלים כמו קודם וראינו כי כל המודלים הצליחו לחזות מי המפלגה המנצחת ולכן למשימה זאת נעדיף לבחור את המסווג הפשוט ביותר כמו KNN אשר הינו קל להבנה, אינו דורש למידה ארוכה.
   b. עבור משימת התפלגות הקולות באוכלוסייה בחרנו במדד הדיוק מתוך הבנה כי ככל שהמסווג מדיוק כך התפלגות הקולות תהיה מדויקת יותר ודומה יותר לסט האימון ולכן בחרנו במסווג Random Forest Classifier, כמו קודם. נרחיב ונומר שמכיוון שבאלגוריתמים מסוג עצי החלטה, ניתן לקבל את הסתברות הסיווגים באופן ישיר (מתוך התפלגות הדוגמאות בעלים, כמו שנלמד בתרגול ובהרצאה), ולכן אלגוריתמים אלו עתידים להיות יציבים יותר, כאשר נשאלים על מדדים הסתברותיים.
   c. עבור משימת ההסעות ישנן כמה שיקולים אשר לא התחשבנו בהן בחיזוי הקודם, לדוגמא: שיקול של איכות השירות: נרצה לשלוח שירותי הסעה לכל מי שמתכוון להצביע למפלגה ולכן נרצה למקסם את מדד True Positive ולמזער את False Negative עבור מפלגה מסוימת כלומר נרצה למקסם את הrecall.
שיקולים כספיים, לא נרצה לשלוח שירותי הסעה עבור מצביע שבסוף לא יצביע למפלגה, כלומר נרצה שהמסווג שלנו ימזער את מדד False Positive ולכן נרצה למקסם את מדד הprecision.
כעת נוכל לספק לכל מפלגה מסווג טוב ביותר עבורה בהתאם לשיקולים שלה, לדוגמא עבור מפלגה שמתחשבת בשני השיקולים נרצה למקסם את מדד F1.
מימשנו זאת בסקריפט המצורף.

לאחר הרצת הסקריפט ניתן לבחון את תוצאותיו ולהבחין לדוגמא כי עבור מפלגת הכחולים דווקא KNN משיג עבורה precision גבוה להבדיל מתהליך הקודם שבחר Random Forest לכל המשימה.

3. עבור משימת החיזוי הרביעית, זיהוי מאפייני מצביעים אשר שינויים יכול להוביל למנצח אחר בבחירות ביצענו את התהליך הבא:
   a. הבנה של המאפיינים אשר יכולים לגרום לשינוי במנצח בבחירות, נשים לב כי הצבע המנצח הינו סגול ומיד אחריו חאקי, לכן נרצה לחפש מניפולציות אשר יגרמו למסווג שלנו להעביר קולות מהצבע הסגול לצבעים אחרים ובפרט לחאקי.

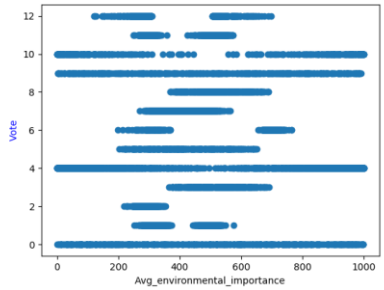
כדי להבין מי הן התכונות הסתכלנו שוב על הקשר בין התכונות לבין הסיווג שביצענו בתרגיל הקודם והגענו למסקנות הבאות:

מאפיין: *Avg_education_importance*



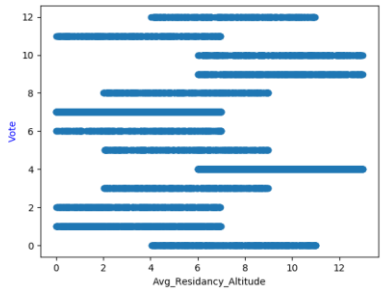
לצבע הסגול (7) מצביעים בעיקר ערכים באמצע הסקלה ולצבע החאקי (4)ערכים קטנים וגדולים בעיקר, לכן מניפולציה אפשרית הינה הורדה של ערכים במאפיין זה.

מאפיין: *Avg_environmental_importance*



לצבע הסגול (7) מצביעים בעיקר ערכים נמוכים באמצע הסקלה ולצבע החאקי(4) ערכים קטנים וגדולים בעיקר, לכן מניפולציה אפשרית הינה דווקא העלאה של ערכים במאפיין זה.

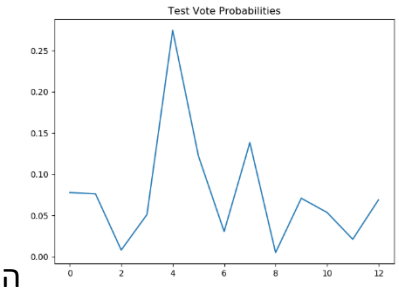
מאפיין: *Avg_Residancy_Altitude*



לצבע הסגול (7) מצביעים רק עד לערך 7 ולצבע החאקי(4) רק ערכים מערך 6 לכן מניפולציה אפשרית הינה דווקא העלאה של ערכים במאפיין זה.

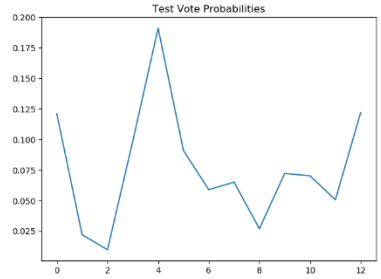
b. את המניפולציות ביצענו על גבי סט המבחן כאשר את סט האימון הותרנו ללא שינוי.
c. תוצאות המניפולציות:

מניפולציה הורדת ערכי *Avg_education_importance:*



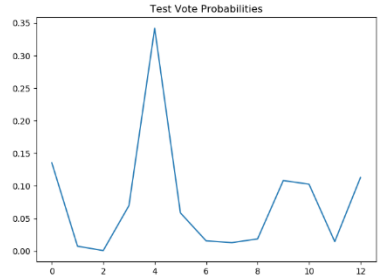
החאקי מנצחים את הסגולים.

מניפולציה העלאת ערכי *Avg_environmental_importance:*



החאקי מנצחים את הסגולים.

מניפולציית העלאת ערכי *Avg_Residancy_Altitude:*



החאקי מנצחים בבחירות וניתן לראות כי הסגולים הפסידו הרבה קולות.

Second Bonus LMS Vs. Perceptron:

1. השוואה בין האלגוריתמים בהינתן היפר פרמטרים הבאים:

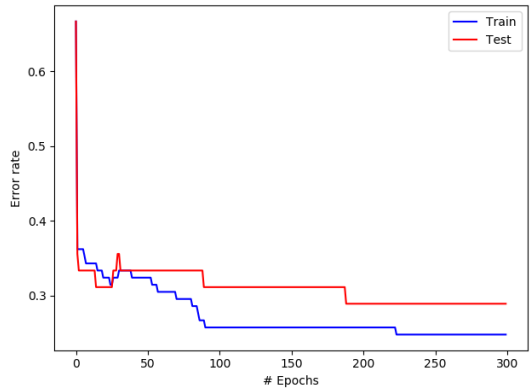
```
Perceptron(alpha=0.0001, max_iter=300, verbose=True, tol=1e-3)
AdalineSGD(eta=0.001, epochs=300)
```

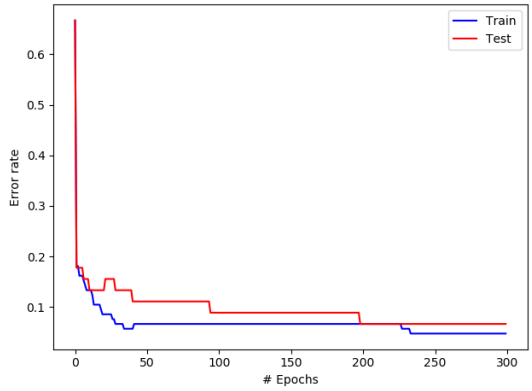
בנוסף המסווגים התאמנו על 70% מסט המידע ונבדק על 30%.

a. סט הדוגמאות הינו Iris:

1 Vs. All	Perceptron		Adaline, LMS Widrow-Hoff	
Class	Accuracy	Convergent [epochs]	Accuracy	Convergent [epochs]
0	Train:100% Test:97.77%	1	Train:100% Test:100%	1
1	Train:76.19% Test:77.77%	13	Train: 74.3% Test: 71.11%	188
2	Train: 97.14% Test:97.77%	8	Train: 93.3% Test: 93.3%	198

להלן גרפים הממחישים את שיעור השגיאה ביחס לכמות ה-epochs שרץ הLMS





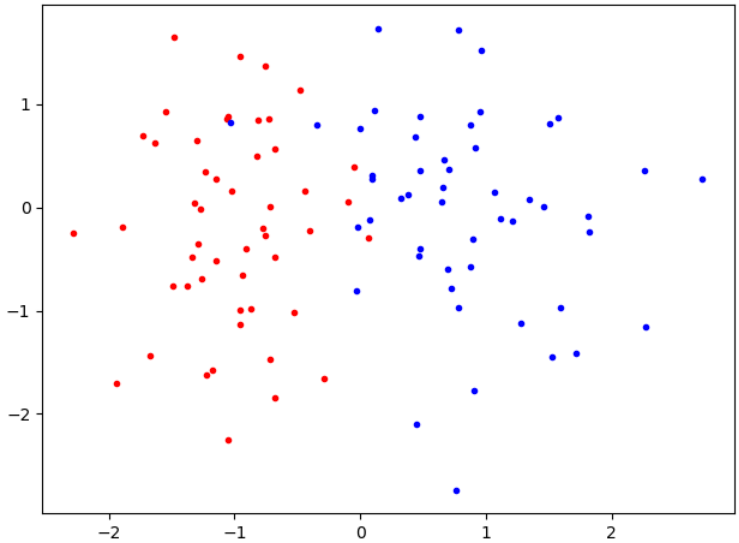
b. סט הדוגמאות הינו digits:

1 Vs. All	Perceptron		Adaline, LMS, Widrow-Hoff	
Class	Accuracy	Convergent [epochs]	Accuracy	Convergent [epochs]
0	Train: 100.00% Test: 99.07%	17	Train: 99.76% Test: 99.44%	3
1	Train: 98.89% Test: 95.93%	20	Train: 98.01% Test: 97.41%	89
2	Train: 100.00% Test: 99.44%	18	Train: 98.65% Test: 99.81%	7
3	Train: 98.97% Test: 96.30%	15	Train: 98.01% Test: 97.96%	1
4	Train: 99.92% Test: 98.89%	20	Train: 99.44% Test: 99.26%	8
5	Train: 99.52% Test: 99.26%	17	Train: 99.12% Test: 98.70%	2
6	Train: 99.76% Test: 99.63%	7	Train: 99.44% Test: 98.89%	77
7	Train: 99.52% Test: 98.89%	19	Train: 99.05% Test: 99.63%	2
8	Train: 96.50% Test: 95.19%	10	Train: 96.02% Test: 96.11%	83
9	Train: 98.09% Test: 96.30%	19	Train: 97.30% Test: 96.85%	11

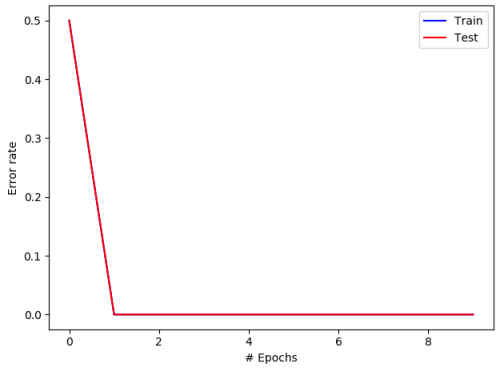
הבחנו כי אלגוריתם ה-Adaline מתכנס מהר יותר על סט המידע digits ובנוסף עבור רוב הסיווגים משיג דיוק טוב יותר מאשר ה-Perceptron.

2. על מנת ליצור סט מידע אשר ה- Perceptron מתכנס יותר מהר ניצור סט מידע שקל מאוד להפריד אותו לינארית ואילו סט המידע אשר לא ניתן להפרידו לינארית אזי ה- Perceptron כלל לא יתכנס ועומת ה – Adaline שמנסה למזער את פונקציית ה- cost שהינה MSE.

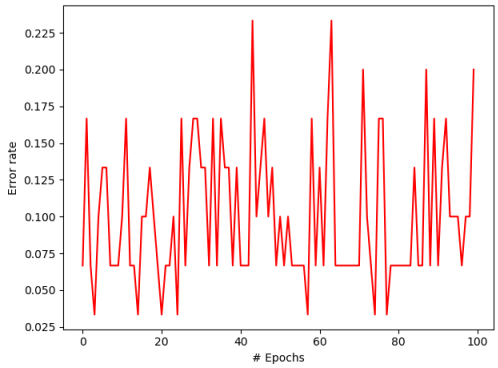
<u>סט המידע הראשון מכיל 2 תכונות ולא ניתן להפרדה לינארית:</u>



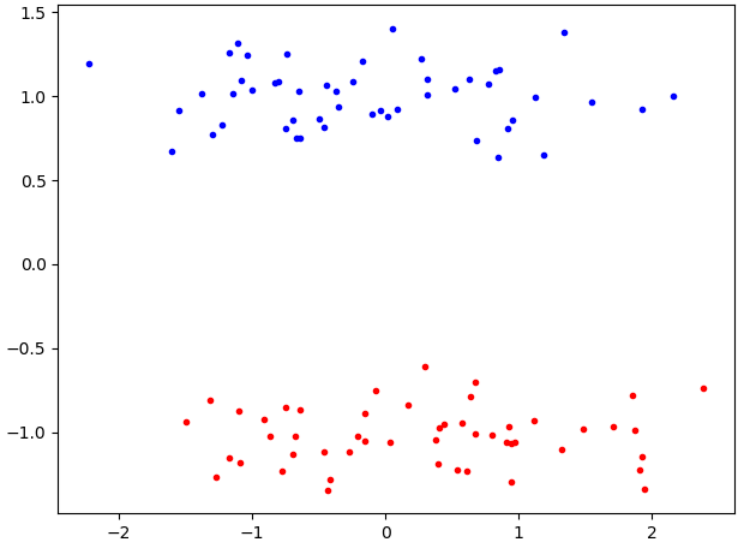
התכנסות הAdaline ב-epoch הראשון:



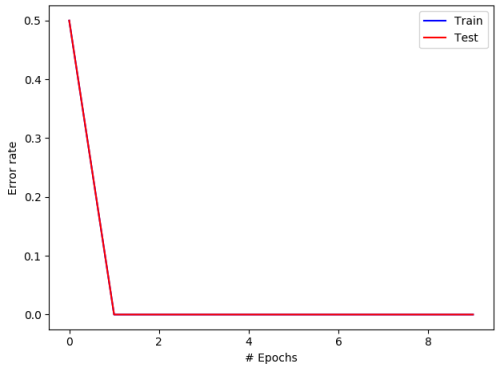
ניתן לראות כי הPerceptron אינו בעל מגמה של ירידה על סט האימון גם כאשר מגדילים את כמות המעבר על המידע:



1. <u>סט המידע השני מכיל 2 תכונות וניתן להפרדה לינארית:</u>



התכנסות הAdaline ב-epoch הראשון:



ניתן לראות כי הPerceptron מתכנס כבר אחרי המעבר הראשון עם דיוק של 100% ומעברים נוספים אינם משנים דבר.