IDA Assignment #2

Aadil Khan

QUESTION 1: Load the dataset in MATLAB as an array named d1 and compute the covariance matrix of the data by using the command: “cm = cov(d1);”. Write all the information that you can infer by interpreting the values in the covariance matrix cm.

ANSWER:

Column names:

1. variance of Wavelet Transformed image (continuous)   
2. skewness of Wavelet Transformed image (continuous)   
3. curtosis of Wavelet Transformed image (continuous)   
4. entropy of image (continuous)

5. class (integer)

cm =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 8.0813 | 4.4051 | -4.6663 | 1.6533 | -1.0243 |
| 4.4051 | 34.4457 | -19.9051 | -6.49 | -1.2974 |
| -4.6663 | -19.9051 | 18.5764 | 2.8872 | 0.334 |
| 1.6533 | -6.49 | 2.8872 | 4.4143 | -0.0245 |
| -1.0243 | -1.2974 | 0.334 | -0.0245 | 0.2471 |

Observations:

Covariance matrix basically helps us to determine the linear relationship which exists between the variables (i.e. columns). In simple terms, it’s a measure of extent to which corresponding elements from two sets of ordered data move in the same direction.

For diagonal values (marked in red), the highest is for 2nd attribute, which means that 2nd attribute (skewness of Wavelet Transformed image) is more variable than the others. The least variable is 5th column which is the class. This is intuitive as there are only two values for it.

For any value (I,J) and I is not equal to J (I🡪 row, J🡪 column), if it is positive, then it means the values of that attribute are moving in same direction. If the value is negative, then it means that are moving in opposite direction.

Attribute #3 and #2 are moving in opposite direction more frequently.

QUESTION 2: Do a scatter plot of attribute-1 and attribute-2. Use different colors to mark the points from the two different classes. Write your interpretation of the separability of the two classes using attribute1 and attribute2, and also any other insights that you can obtain from this scatter plot

ANSWER:

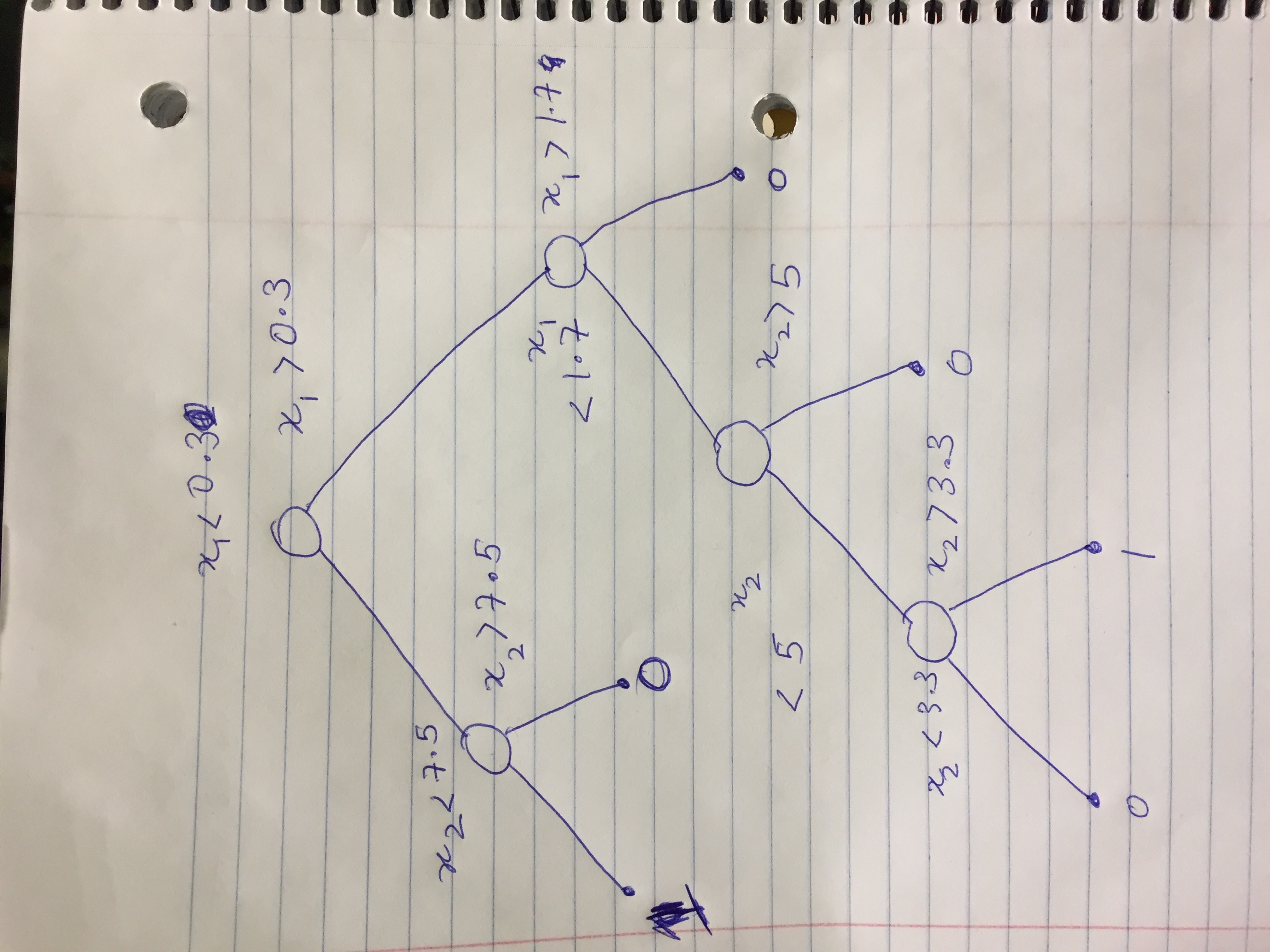


The data is linearly inseparable. We can see that lot of points are sort of mixed together. The fake notes (class 0) have higher values for both x1 and x2.

QUESTION 3: On the scatter plot obtained in Q#2 above, draw the axis-parallel boundaries, using your intuition, to suggest the boundaries that an efficient decision tree may draw. Show the resulting partitions in the form of a decision tree.

ANSWER:





QUESTION 4: From the dataset randomly select 800 records for testing, 200 for validation, and the rest for testing. These selections MUST be random. Use a random number generator function to generate random numbers between 1 and 1372, and then choose records for training, validation, and testing. Show the code used for selecting the three different sets of records. Write the record numbers of rows that are chosen for your test set.

ANSWER:

Code for selecting records:

randData = d1(randperm(1372),:);

training= randData(1:800,:);

validation = randData(801:1000,:);

testing = randData(1001:1372,:);

Size:

training = 800 5

validation = 200 5

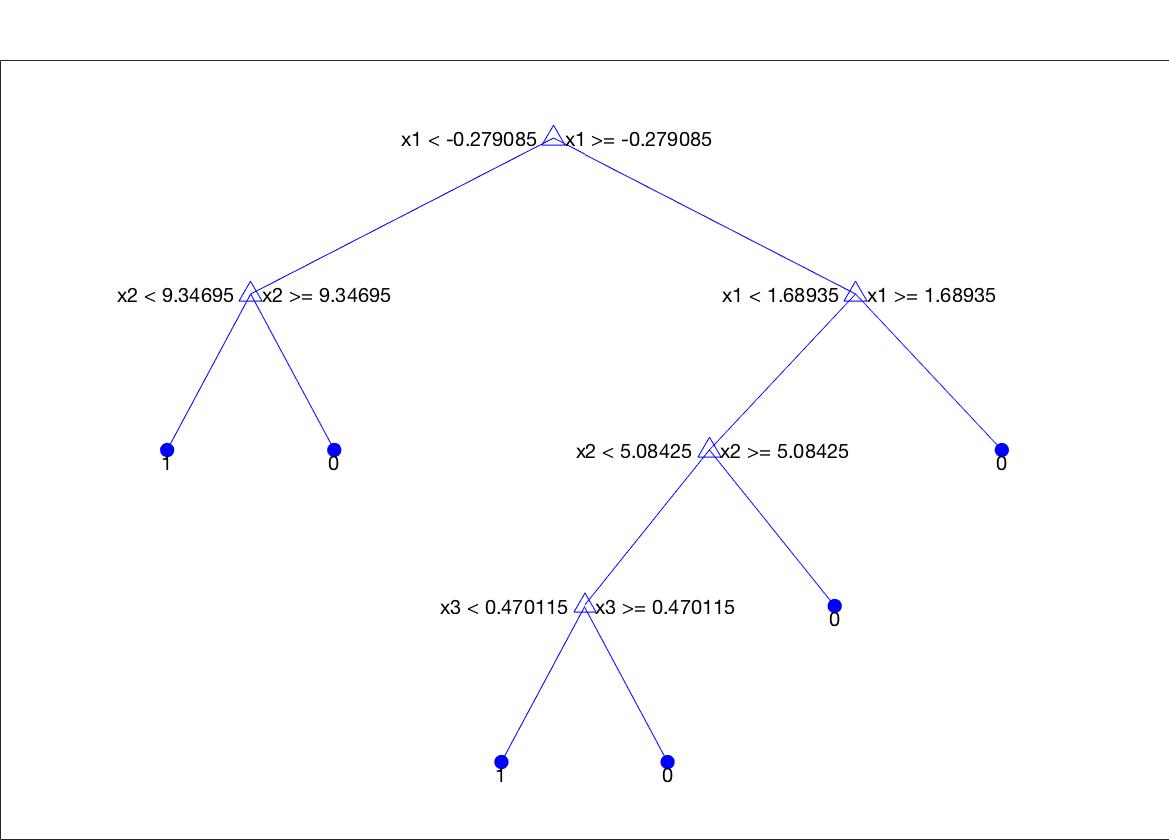
testing = 372 5

Testing Records:

1, 5, 7, 15, 16, 20, 21, 26, 29, 30, 36, 37, 38, 42, 44, 50, 51, 52, 59, 66, 67, 68, 76, 78, 80, 87, 92, 93, 98, 100, 104, 105, 107, 110, 111, 113, 115, 118, 120, 126, 130, 132, 134, 139, 141, 143, 144, 148, 162, 163, 164, 166, 171, 172, 179, 180, 189, 194, 198, 201, 202, 211, 213, 214, 217, 218, 219, 220, 222, 223, 224, 228, 232, 235, 236, 237, 242, 243, 245, 246, 248, 249, 262, 263, 267, 271, 272, 273, 277, 281, 290, 291, 292, 294, 303, 305, 307, 311, 313, 318, 319, 331, 336, 338, 341, 342, 348, 349, 351, 352, 357, 364, 371, 373, 374, 380, 382, 384, 385, 391, 392, 401, 404, 406, 409, 410, 414, 418, 423, 424, 433, 434, 437, 445, 447, 448, 449, 450, 453, 456, 458, 459, 464, 469, 477, 479, 480, 481, 482, 483, 486, 488, 493, 494, 496, 499, 504, 509, 518, 519, 528, 529, 535, 537, 546, 549, 553, 557, 558, 561, 562, 565, 575, 579, 583, 584, 591, 593, 595, 598, 601, 605, 612, 623, 626, 630, 632, 636, 637, 638, 639, 642, 647, 648, 651, 652, 653, 658, 666, 668, 671, 673, 675, 677, 683, 689, 690, 693, 695, 696, 697, 701, 703, 717, 719, 722, 727, 731, 737, 738, 742, 745, 750, 757, 765, 769, 771, 772, 777, 779, 780, 781, 788, 792, 795, 801, 811, 816, 819, 822, 825, 830, 839, 842, 844, 845, 850, 856, 869, 871, 872, 873, 876, 877, 882, 885, 886, 888, 891, 896, 901, 902, 903, 904, 905, 907, 915, 917, 925, 926, 930, 933, 934, 944, 945, 946, 948, 951, 952, 963, 973, 977, 981, 984, 986, 988, 989, 991, 994, 996, 998, 1007, 1010, 1011, 1014, 1015, 1019, 1020, 1026, 1028, 1029, 1043, 1045, 1057, 1058, 1060, 1066, 1069, 1071, 1073, 1075, 1079, 1086, 1101, 1102, 1113, 1114, 1121, 1124, 1127, 1128, 1131, 1138, 1139, 1156, 1159, 1160, 1164, 1175, 1178, 1182, 1193, 1194, 1198, 1201, 1214, 1216, 1217, 1219, 1221, 1223, 1228, 1232, 1234, 1242, 1249, 1253, 1266, 1269, 1275, 1280, 1283, 1289, 1291, 1292, 1293, 1294, 1307, 1311, 1312, 1322, 1323, 1334, 1335, 1342, 1343, 1346, 1354, 1361, 1366, 1367, 1369, Trial>>

QUESTION 5: Use the training set to learn decision trees (use fitctree command of Matlab) by varying the parameter that controls the minimum number of records in a leaf node. For this parameter use the values of 1, 2, 5, 10, 15, 20, 25, 30, . . , 50. For each generated tree find its performance (accuracy) on the test and validation sets. Plot this accuracy value against the parameter values for both the data sets. Choose that value of parameter which in your view does not over fit the data. **Justify your choice of the parameter value and the corresponding decision tree**. Show the selected tree graphically using Matlab’s view command. The role of the validation data set is to help you pick that decision tree which gives the best performance without overfitting the training data. **plot the accuracy values of the training (800 records) and validation (200 records) datasets to select the best possible decision tree.**

ANSWER:







I have chosen the minimum leaf size as 20. As we can see from both the graph (validation and testing), the behavior of accuracy versus min leaf size is same. That means our decision tree will not overfit the data. However, if we have minLeaf size too small, the tree will be deep and chances of error increases, if it is too shallow, it’s performance is impacted. Hence I have chosen a balance between the two.

QUESTION 6: Test the selected decision tree using the test portion of the dataset. Report accuracy, precision, and recall for the test dataset. Report the precision and recall values for both classes separately.

ANSWER:

Confusion for test dataset:

confMat =

184 15

9 164

Accuracy: 184+164/184+164+15+9

Precision: 184/193

Recall: 184/184+15

Confusion for individual classes

Class 1:

Acc =

0.9607

confMat =

7 171

0 0

Precision: 7/7+0

Recall: 7/7+171

Class 0:

confMat =

196 16

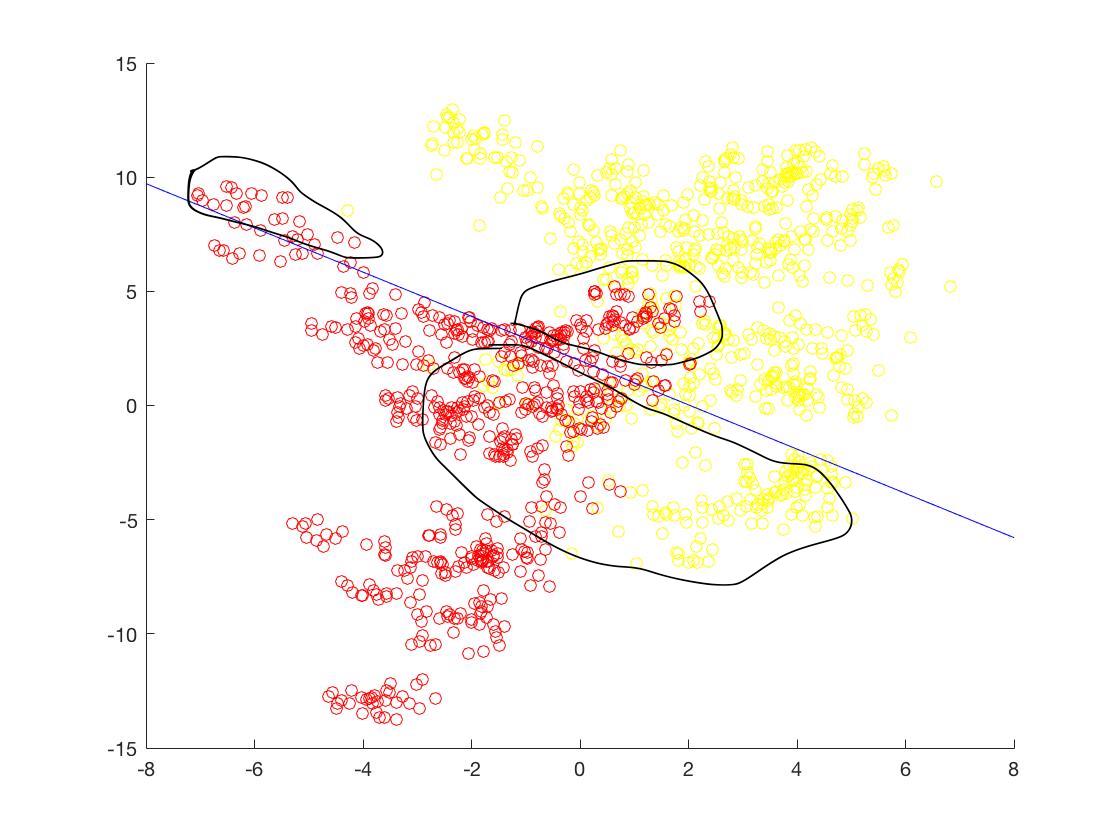
1. 0

Precision: 196/196+0

Recall: 196/196+16

QUESTION 7: Consider only the attribute-1 and attribute-2 of the dataset for predicting the class label. Train a perceptron using Matlab’s Train function that finds the best discriminating surface for this 2-D data space. Report the weight vector output by this function and report the number of data points that are misclassified. Plot all the data points as done in the scatter plot in Q#2 above. Now plot the line representing the learned perceptron on this scatter plot. Mark the points that are misclassified.

ANSWER:



Points which are misclassified are marked.(\*yellow ones on red side are misclassified.

weights = -8.3672 2.3159

bias = 29

To test whether a point is misclassified, I used weight vectors and bias to calculate the class using the perceptron formula w.x+b.

Then, cross checked who were classified.

correctly classified= 210

misclassified = 162

acc = 0.5645