Programming Assignment 2

Sankeerth.D
EE13B102
Electrical Engineering

October 7, 2016

Abstract

LibSVM and Neural Network has been used on the DS-2 Dataset for Probs 1 and 2. Weka library is used to classify the mushroom data as edible or poisonous in problem 3.

Note: For prob3, zoom into the pdf to clearly see the nodes' description of the tree.

Note: The histogram (the output of data_processing.py) is used as features for the classifier in problem 1 and problem 2. Also, on inspecting the features by training a neural network classifier using weka library, it is seen that the performance is similar to the ones below. Hence the reason for low accuracy. It is unlikely that it's due to a mistake in my code for the neural network (as on testing with ssample datasets online, it gives a very accurate test output). Hence it may be concluded histogram isn't a good set of features for scene classification.

1 Problem 1: LibSVM on DS-2

Usage of histogram data turns out to be a poor strategy in general, since we get high error rates oon using it with SVM or neural network classifer. The training data has been shuffled and 5-fold cross validation has been performed over it. The best accuracy parameters are chosen to be the model parameters.

NOTE: Shuffling the data and doing SVM is not a very good idea, as performance of SVM is best if all classes are in equal number in the training data. But since we have 1006 datapoints, and since implimentation is easier, this strategy is invoked before building cross validation set.

Few cases with not-so-significant amount of bias have been presented here. Note that these parameters have a small variance as running the program once again forms another set of shuffled data for cross validation.

1.1 Linear model:

On running the program to perform 5-fold cross validation and varying the parameter C, the following results are obtained (for one instance of test data):

These values may not exactly match those in the model file because the following were recorded at one particular split of the dataset used for cross validation.

C=1e-2: Best Training accuracy = 31.14%
Test Accuracy = 32.5%

C=1e-2	Precision	Recall	F-measure
coast	0.306	0.55	0.38
forest	0.27	0.15	0.12
inside city	0.347	0.4	0.3
mountain	0.4	0.2	0.35

,

C=1e-1: Best Training accuracy = 29.15%
Test Accuracy = 27.5%

C=1e-1	Precision	Recall	F-measure
coast	0.313	0.25	0.278
forest	0.22	0.2	0.21
inside city	0.304	0.35	0.325
mountain	0.26	0.3	0.279

,

C=1e0: Best Training accuracy = 29.15%
Test Accuracy = 27.5%

C=1e0	Precision	Recall	F-measure
coast	0.318	0.35	0.333
forest	0.136	0.15	0.143
inside city	0.313	0.25	0.278
mountain	0.35	0.35	0.35

,

C=1e1: Best Training accuracy = 30.65%
Test Accuracy = 30%

C=1e1	Precision	Recall	F-measure
coast	0.33	0.45	0.38
forest	0.15	0.1	0.12
inside city	0.3	0.3	0.3
mountain	0.35	0.35	0.35

,

C=1e2: Best Training accuracy = 29.5%
Test Accuracy = 31.25%

C=1e2	Precision	Recall	F-measure
coast	0.346	0.45	0.39
forest	0.2	0.1	0.13
inside city	0.33	0.3	0.32
mountain	0.307	0.4	0.347

,

C=1e3: Best Training accuracy = 31.63%
Test Accuracy = 26.25%

C=1e3	Precision	Recall	F-measure
coast	0.24	0.4	0.302
forest	0.267	0.2	0.229
inside city	0.286	0.3	0.293
mountain	0.27	0.15	0.193

1.2 Polynomial model:

The classifier is run for a polynomial kernel of degree 4. On trying out with different degrees, it's found that d=4 is the best suited for this case. For different C, the performance is as follows:

C=1e1: Best Training accuracy = 31.76%
Test Accuracy = 33.75%

C=1e1	Precision	Recall	F-measure
coast	0.30	0.95	0.457
forest	0.05	0.3	0.375
inside city	0.33	0.05	0.086
mountain	0.5	0.05	0.09

,

C=1e2: Best Training accuracy = 32.5%
Test Accuracy = 26.25%

C=1e2	Precision	Recall	F-measure
coast	0.27	0.9	0.42
forest	0.333	0.05	0.086
inside city	0.5	0.1	0.167
mountain	0.0	0.0	-

,

C=1e3: Best Training accuracy = 33.12%
Test Accuracy = 36.25%

C=1e3	Precision	Recall	F-measure
coast	0.32	0.9	0.47
forest	0.375	0.15	0.21
inside city	0.5	0.35	0.41
mountain	0.5	0.05	0.09

1.3 Gaussian RBF model:

A working value of gamma in this case turns out to be in the range of 1e-3 to 1e-1. The classifier is highly biased if we choose a gamma outside this range.

Parameter C below 1e0 is highly biased.
gamma = 5e-3
Best Training accuracy = 33.00%
Test Accuracy = 31.25%

C=1e0	Precision	Recall	F-measure
coast	0.325	0.65	0.433
forest	0.307	0.2	0.242
inside city	0.5	0.15	0.231
mountain	0.238	0.25	0.244

’

gamma = 5e-3
Best Training accuracy = 33.12%
Test Accuracy = 33.75%

C=1e1	Precision	Recall	F-measure
coast	0.389	0.7	0.433
forest	-	0.	0.242
inside city	0.29	0.35	0.231
mountain	0.3	0.3	0.3

’

gamma = 5e-3
Best Training accuracy = 31.88%
Test Accuracy = 31.25%

C=1e2	Precision	Recall	F-measure
coast	0.39	0.45	0.418
forest	0.23	0.15	0.182
inside city	0.313	0.25	0.278
mountain	0.286	0.4	0.333

’

1.4 Sigmoid model:

Choosing the C parameter below 1e0 gives a high bias to the classifier.
’

Best Training accuracy = 30.15%
Test Accuracy = 30%

C=1e0	Precision	Recall	F-measure
coast	0.349	0.6	0.436
forest	0.2	0.1	0.133
inside city	0.3	0.3	0.3
mountain	0.267	0.2	0.228

’

Best Training accuracy = 28.28%
Test Accuracy = 26.25%

C=1e1	Precision	Recall	F-measure
coast	0.307	0.4	0.348
forest	0.125	0.1	0.143
inside city	0.21	0.2	0.205
mountain	0.259	0.35	0.298

’

Best Training accuracy = 29.4%
Test Accuracy = 32.5%

C=1e2	Precision	Recall	F-measure
coast	0.409	0.45	0.429
forest	0.21	0.2	0.205
inside city	0.35	0.35	0.35
mountain	0.368	0.35	0.359

’

Best Training accuracy = 28.9%
Test Accuracy = 26.25%

C=1e1	Precision	Recall	F-measure
coast	0.323	0.5	0.392
forest	0	0	-
inside city	0.25	0.1	0.143
mountain	0.257	0.45	0.327

2 Problem 2: 3-Layered ANN

The gradient of the loss function is obtained using backpropagation. As mentioned previously, the histogram dataset is a bad set of parameters to classify the data. We get a poor test-set accuracy, and on increasing the accuracy, we only end up overfitting the training data.

NOTE: Vanilla gradient descent has been implimented and it is seen that the learning rate alpha needs to be constantly changed and obtaine by trial and error on changing the parameters. It takes around a 1000 iterations even after that. Instead, the CG gradient descent algorithm completes it in around 200 iterations, and is much faster. This is implimented in Scipy.optimize package. Hence to get results in a speedy manner, this has been used, however, gradient descent is impleented and shown for the purpose of this assignment.

To check validity of backpropagation, a numerical gradient test has also been shown in the code.

The output layer is the softmax layer., and the layers in the middle have a exponent-in-denominator activation function. Backpropagation has been implimented with the notations and variables in a way that it can also be used for deeper networks.

.

Defined functions:

def train_net(no.of_layers, [l1_size, l2_size, ...]): Trains neural network to get the parameters using gradient optimization algorithm.

class neural_net: The data structure ’neural_net’ class is implimented which stores the coefficients between layers as a list of matrices.

def forward_prop(net, x): This function calculates the output layer given the input layer

def backprop_grad(net, x): calculates gradient using backpropagation.

def predict(net, x): given network, and input list x, we get the output.

def get_confusion_matrix(net, Xtest, ytest): calculated confusion matrix, precision, recall, accuracy.

max. iterations is set to 250. Early stopping isn’t done as 2250 usually optimizes close enough.

Regularization parameter gamma, above the value of 10 gives a sparse coefficient matrix.

’

’Few of the are cases shown here:

’

For hidden layers=20, gamma = 1e-2:
Training set accuracy:98.47%
Training set confusion matrix:

259	3	2	2
0	231	2	2
5	3	218	3
1	2	1	272

Test set accuracy:28.74%

9	4	4	5
3	4	4	8
3	5	7	4
5	7	5	3

gamma=1e-2	Precision	Recall	F-measure
coast	0.409	0.45	0.43
forest	0.21	0.2	0.205
inside city	0.37	0.35	0.359
mountain	0.15	0.15	0.15

.

For hidden layers=20, gamma = 1e-1:
Training set accuracy:98.31% Training set confusion matrix:

265	1	3	4
0	237	1	4
0	3	217	1
0	0	2	270

Test set accuracy:26.25%

7	3	6	8
4	5	2	6
2	5	6	3
7	7	6	3

gamma=1e-1	Precision	Recall	F-measure
coast	0.29	0.35	0.32
forest	0.29	0.25	0.27
inside city	0.375	0.30	0.33
mountain	0.13	0.15	0.14

.

For hidden layers=20, gamma = 1e0:
Training set accuracy: 79.52%
Training set confusion matrix:

231	26	18	20
16	176	9	20
11	17	170	16
7	20	26	223

Test set accuracy:33.75%

9	5	5	6
2	7	2	5
2	2	6	4
7	6	7	5

gamma=1e0	Precision	Recall	F-measure
coast	0.36	0.45	0.4
forest	0.438	0.35	0.389
inside city	0.428	0.3	0.353
mountain	0.2	0.25	0.22

.

For hidden layers=20, gamma = 1e1:
Training set accuracy:33.1% Training set confusion matrix:

177	118	59	123
0	0	0	0
0	0	0	0
88	121	164	156

Test set accuracy:34.9%

14	9	9	6
0	0	0	0
0	0	0	0
6	11	11	14

gamma=1e1	Precision	Recall	F-measure
coast	0.37	0.7	0.48
forest	-	-	-
inside city	-	-	-
mountain	0.33	0.7	0.45

.

.

For the case of 7 nodes in the hidden layer, the results are as follows:
For hidden layers=7, gamma = 1e-2:
Training set accuracy:85.18%
Training set confusion matrix:

234	10	11	11
9	198	8	7
15	9	178	14
7	22	26	247

Test set accuracy:28.75%

9	3	4	4
3	6	6	5
4	6	3	6
4	5	7	5

gamma=1e-2	Precision	Recall	F-measure
coast	0.45	0.45	0.45
forest	0.3	0.3	0.3
inside city	0.16	0.15	0.154
mountain	0.23	0.25	0.244

.

For hidden layers=7, gamma = 1e-1:
Training set accuracy:84.89%.
Training set confusion matrix:

233	12	12	5
11	187	16	3
12	13	171	8
9	27	24	263

Test set accuracy:32.5%

7	3	2	6
4	7	7	6
2	5	5	1
7	5	6	7

gamma=1e-1	Precision	Recall	F-measure
coast	0.388	0.35	0.368
forest	0.29	0.35	0.318
inside city	0.384	0.25	0.30
mountain	0.28	035	0.31

.

For hidden layers=7, gamma = 1e0:

Training set accuracy: 67.29%

Training set confusion matrix:

214	39	39	31
18	141	27	25
17	29	126	27
16	30	31	196

Test set accuracy:35%

12	5	7	7
2	6	2	4
2	3	5	4
4	6	6	5

gamma=1e0	Precision	Recall	F-measure
coast	0.387	0.6	0.47
forest	0.428	0.3	0.35
inside city	0.357	0.25	0.294
mountain	0.238	0.25	0.244

.

Apart from the cases shown above, the algorithm has been run from hidden layers between2 to twenty. Overfitting starts at around 10 hiden layers. Only a few instances have been presented in this report.

If better quality data is given (not only histogram), the performance can be greatly improved.

3 Problem 3: J48 Decision tree

The weka program provides the algorithm for J48 tree. The .arff format has been provided after converting it from .csv. The performance of the tree yields a straight up 100% over training and test datasets. Only on assigning a very high value of MinNumObj (setting it to 30) yields a very small reduction in precision and recall.

Working with 1124 training instances implies we're splitting the data into 86.16% training data and the rest as test data.

From the trees below, the parent decider is always odor. In decreasing order of importance, the relavent features for classification are (in general, looking at all the models):

1.Odor

2.Spore print color

3.gill-size

4.gill-spacing

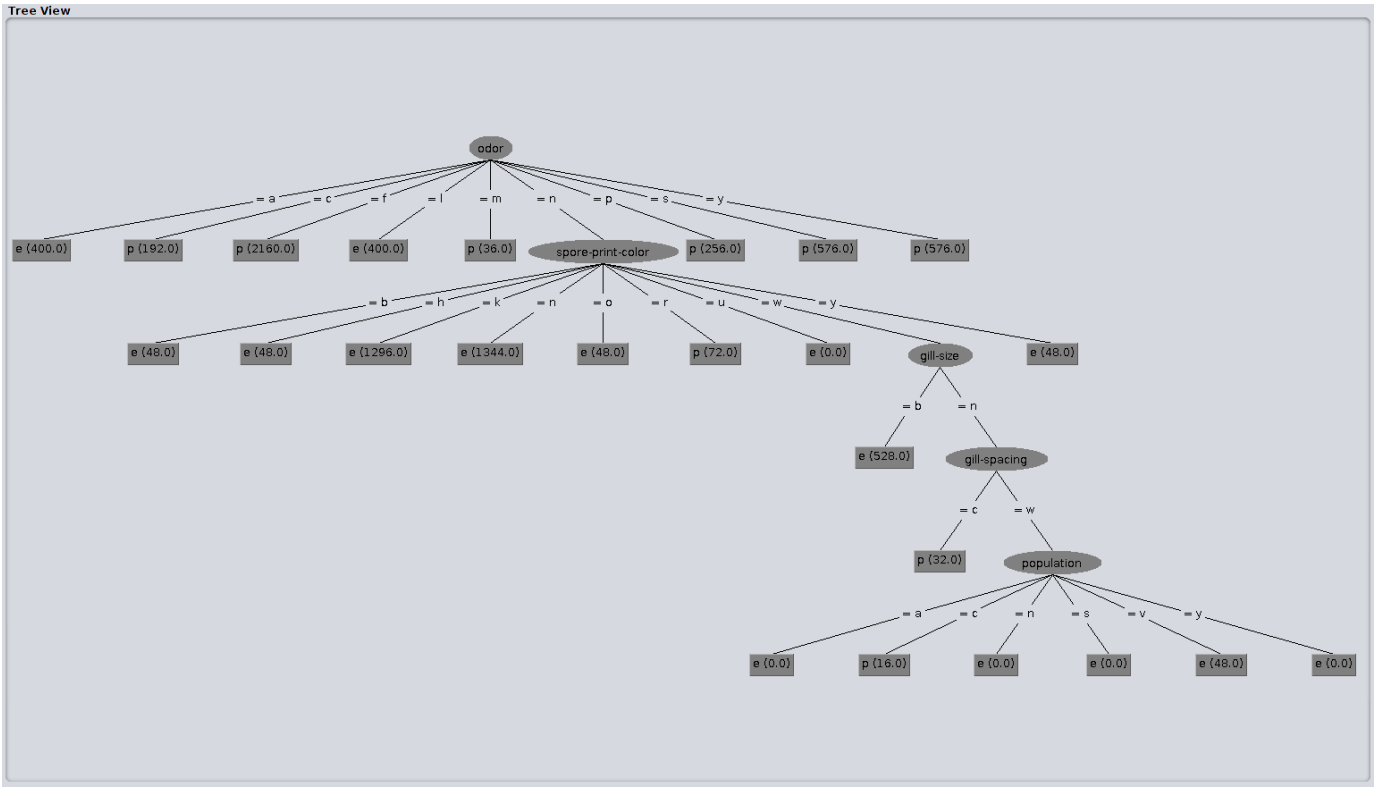
5. Stalk surface below ring

'

Initially, choose MinNumObj=2. We obtain the same tree whether weturn on reduceErrorPruning or not (however the probabilities change). The performance parameters are as follows:

<Either class> class	Value
Accuracy	1.0
Precision	1.0
Recall	1.0
F-measure	1.0

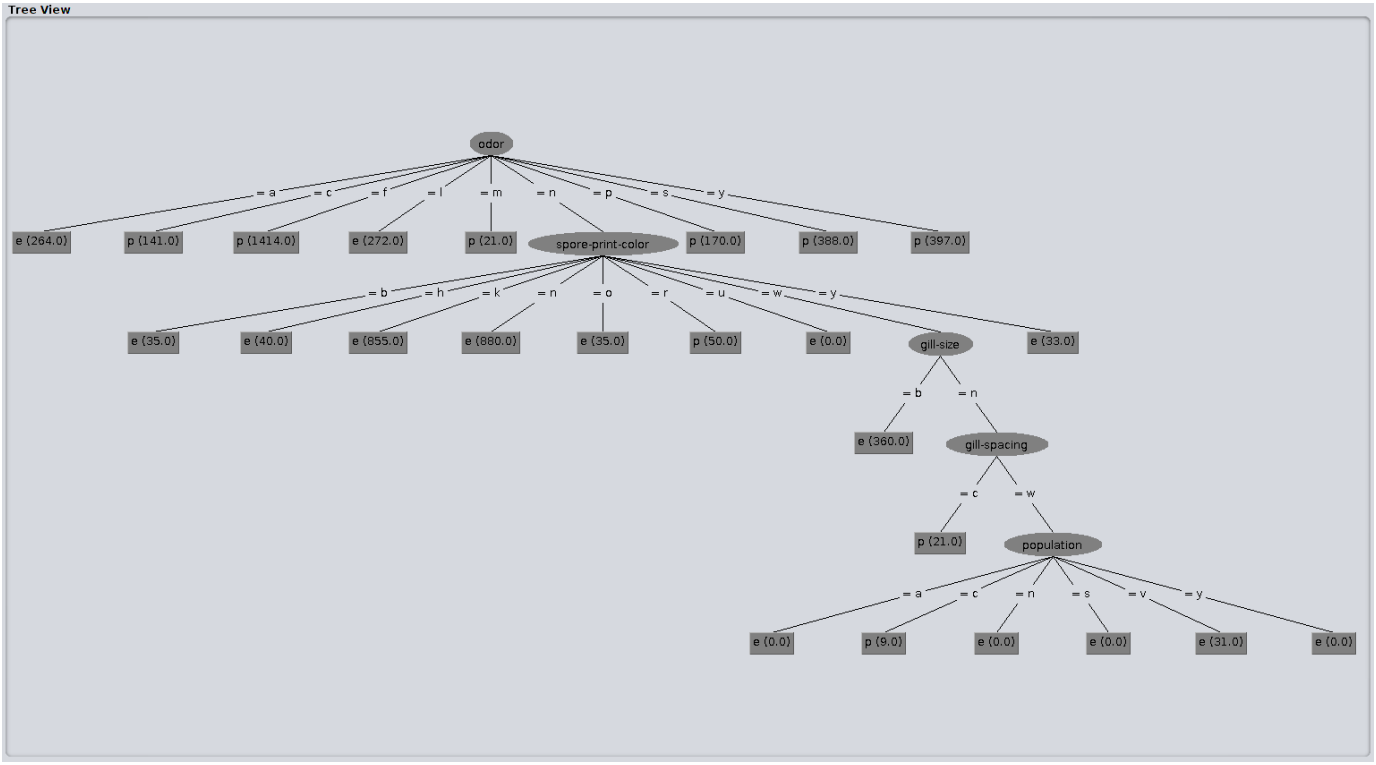
The correspondong tree is (reducePruningError=False):



,

,

With reduced error pruning Set to true, the tree is:



,

,

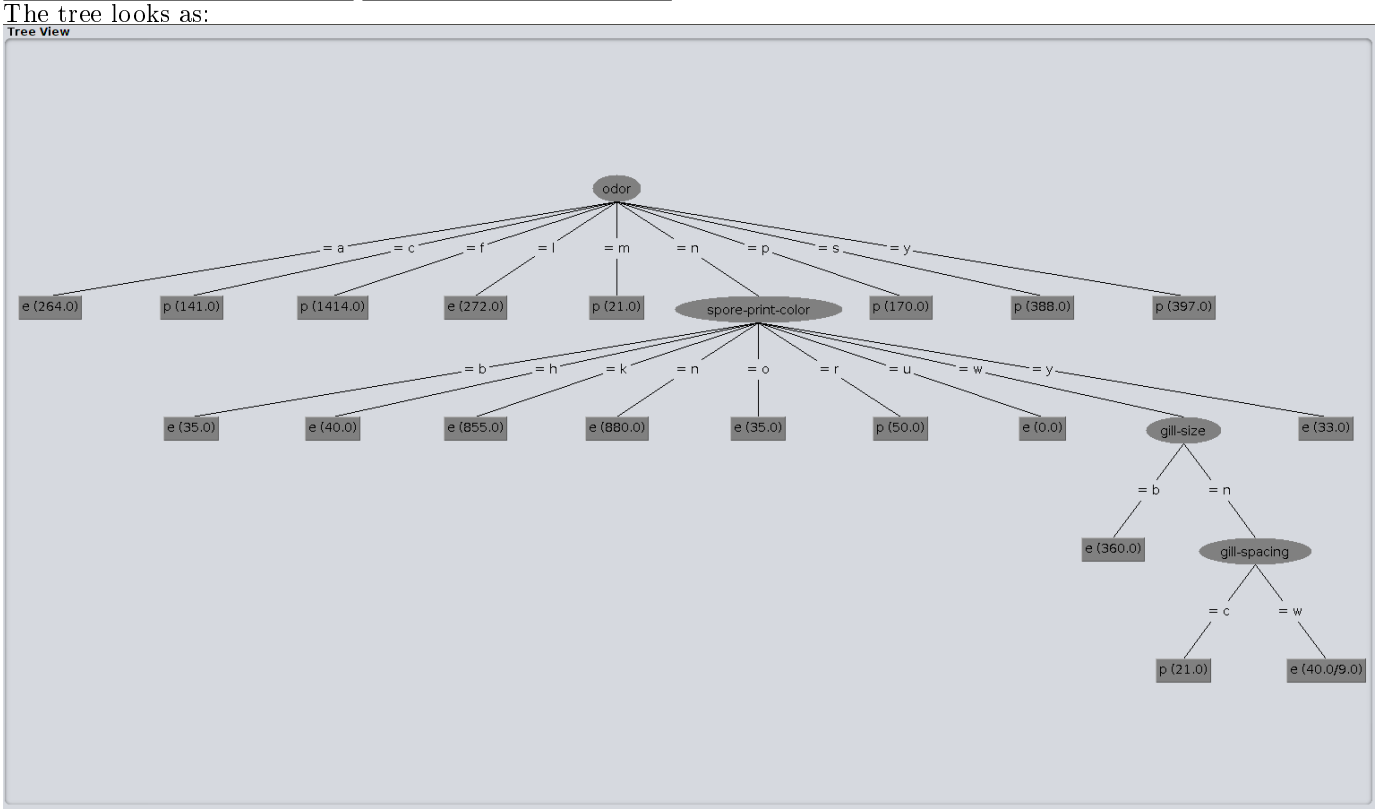
,

On selecting the MinNumObj = 20, and reducePruningError is True. The performance parameters of the tree are:

Poisonous class	Value
Accuracy	99.5552%
Precision	0.992
Recall	1.0
F-measure	0.996

Edible class	Value
Accuracy	99.5552%
Precision	1.0
Recall	0.991
F-measure	0.995

The tree looks as:

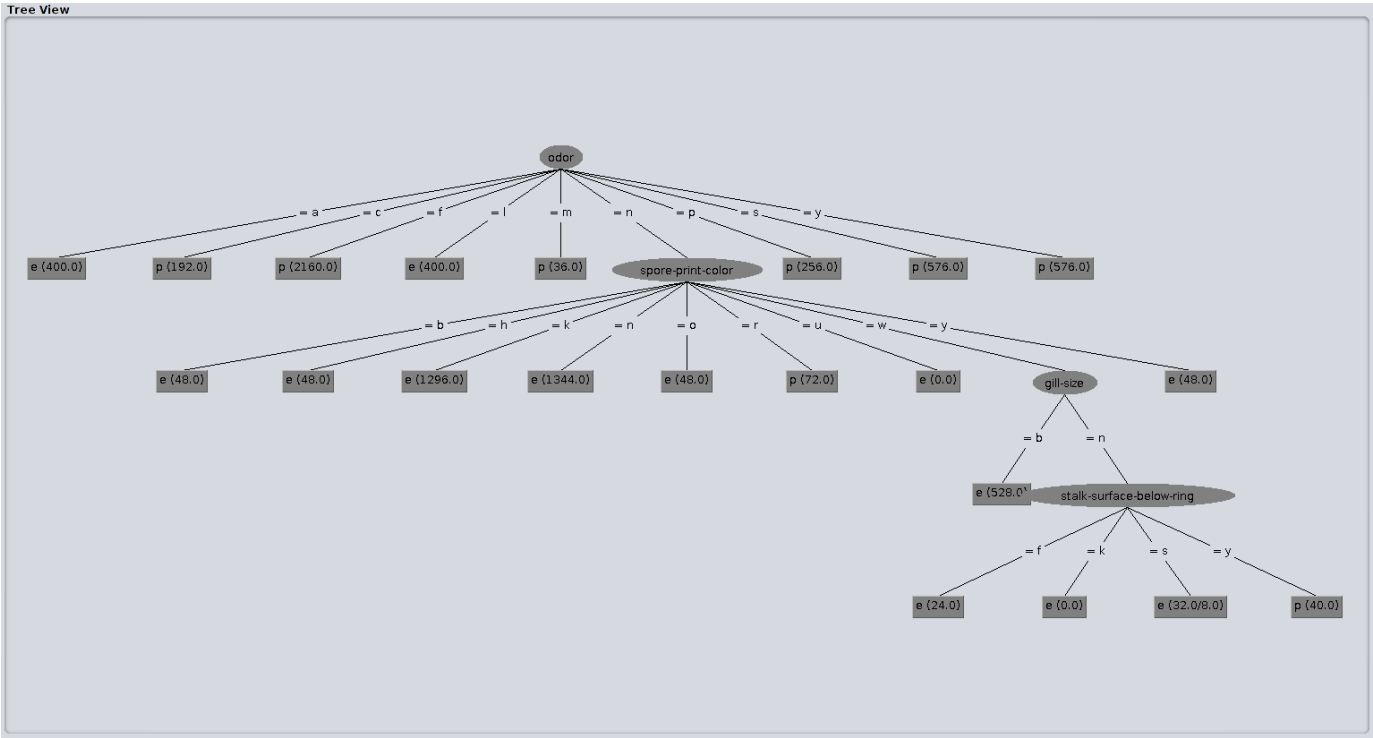


,

On setting reduceErrorPruning to false, THe performance is as follows:

<Either class>	Value
Accuracy	1.0
Precision	1.0
Recall	1.0
F-measure	1.0

Tree for this case:


Tree View
odor
= a
= c
= f
= l
= m
= n
= p
= s
= y
e (400.0)
p (192.0)
p (2160.0)
e (400.0)
p (36.0)
spore-print-color
p (256.0)
p (576.0)
p (576.0)
= b
= h
= k
= n
= o
= r
= u
= w
= y
e (48.0)
e (48.0)
e (1296.0)
e (1344.0)
e (48.0)
p (72.0)
e (0.0)
gill-size
e (48.0)
= b
= n
e (528.0
stalk-surface-below-ring
= f
= k
= s
= y
e (24.0)
e (0.0)
e (32.0/8.0)
p (40.0)