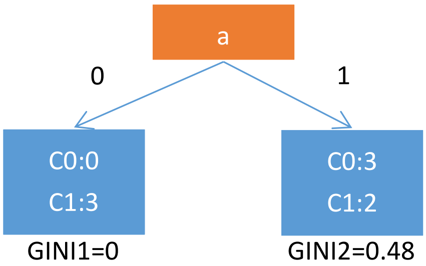
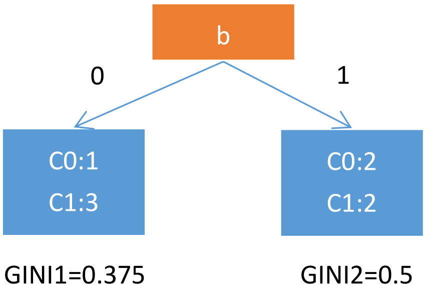
Q1

a) Parent: C0=3 C1=5 Gini = $1-(3/8)^2-(5/8)^2$ =0.46875

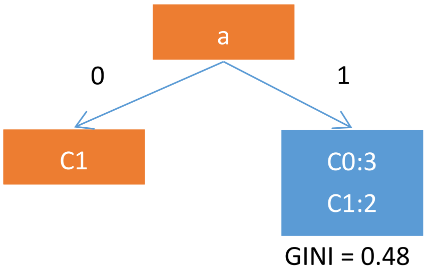
Split according to 'a', Gini = $3/8*0 + 5/8(1-(3/5)^2-(2/5)^2)$=0.3



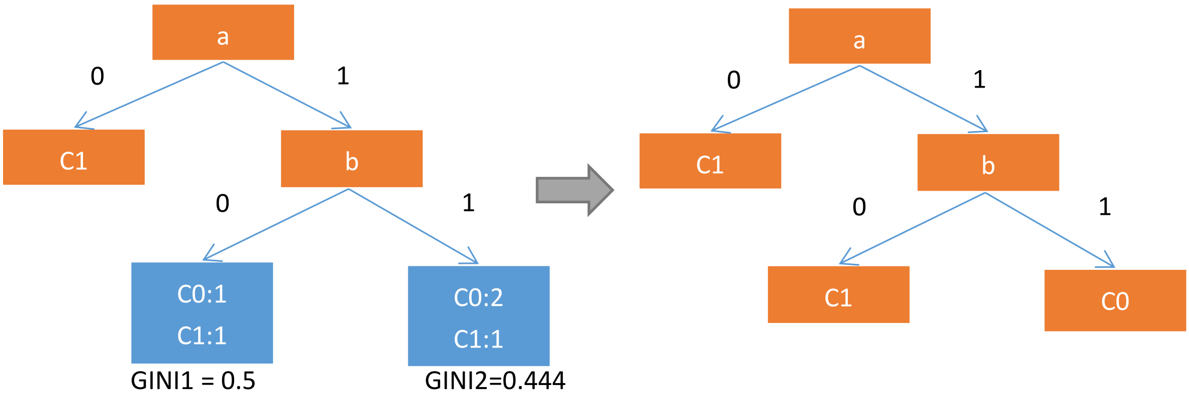
Split according to 'b', Gini = $4/8*(1-(1/4)^2-(3/4)^2) + 4/8(1-(2/4)^2-(2/4)^2)$=0.4375



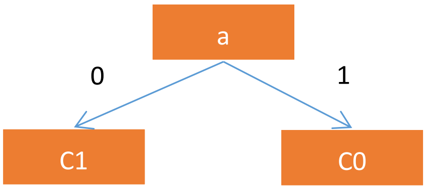
So we choose split according to 'a',and since left node's gini is 0， no more need to split.



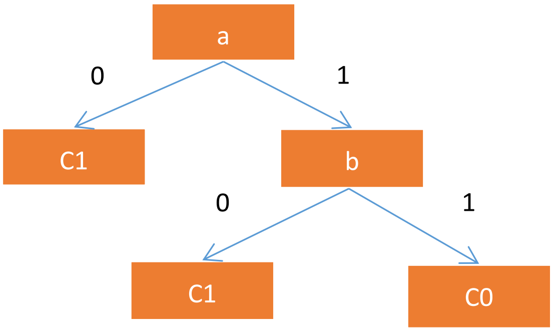
Next split according to 'b' in the right node. And no further split is possible.



B) Before splitting according to 'b'



After splitting according to 'b'



Training error = 2

Number of leaves = 2

Generalization error = 2+0.5*2 = 3

Training error = 2

Number of leaves = 3

Generalization error = 2+0.5*3 = 3.5

So we prune the sub-tree of ‘b’

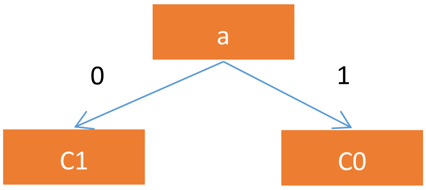
Before splitting according to ’a’

Training error = 3

Number of leaves = 1

Generalization error = 3+0.5*1= 3.5

So we don’t prune the sub-tree of ‘a’. the result is:



C) the class value of (a=’1’,b=’1’) is Class 0.

Q2

P(cj|a1,a2...an)=P(a1,a2...an|cj)P(cj)/P(a1,a2...an), so we only need to compare P(a1,a2...an|cj)P(cj).

P(X|Status = junior) = P(Department=’marketing’|Status = junior)* P(Age=’31-35’|Status = junior)* P(Salary=’46k-50k’|Status = junior)=2/7*3/7*3/7 = 0.052478

P(X|Status = senior) = P(Department=’marketing’|Status = senior)* P(Age=’31-35’|Status = senior)* P(Salary=’46k-50k’|Status = senior)=1/4*1/4*1/4 =0.015625

P(X|Status = junior)P(Status = junior) = 0.052478*7/11 = 0.033395

P(X|Status = senior)P(Status =senior) =0.015625*4/11 = 0.00568182

0.033395>0.00568182

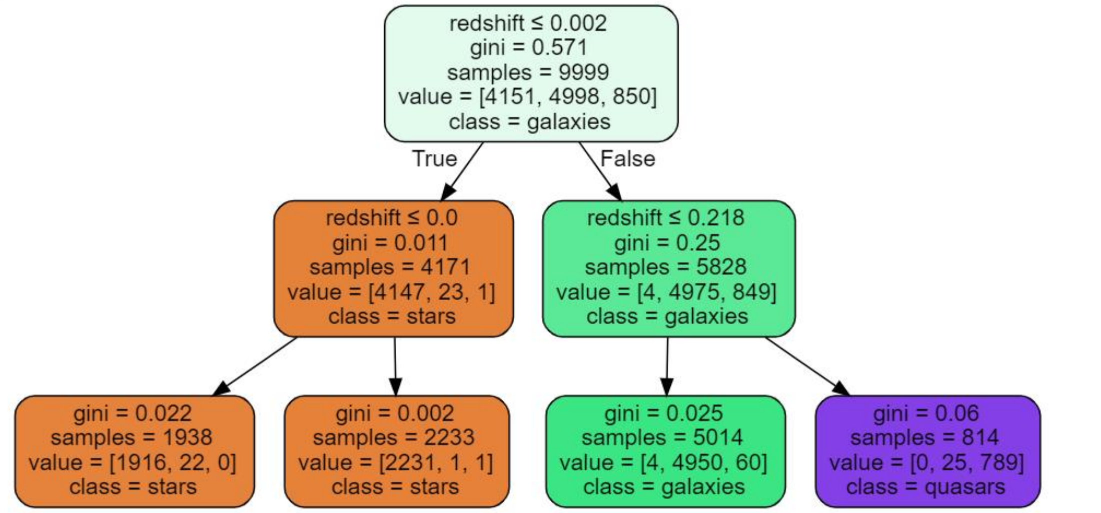
(‘marketing’, 31-35,46k-50k)’s status is junior.

Q3

1. The result is in ’SkySurvey.pdf’
2. generalization error:140.4999999999996
3. I get the depth of the tree is 11 without restriction, so the depth is in range 1 to 11. max_depth is used to control the size of the tree to prevent overfitting, if max_depth is too small, the model did not learn the structure of the data and performs poorly in both training set and test set, if max_depth is too large, the model becomes too specialized, good in training set but poor in test set. From the result, we get generalization error is smallest when max_depth=2.

```
generalization error when max_depth=1: 878.5000000000002
generalization error when max_depth=2: 116.49999999999959
generalization error when max_depth=3: 120.49999999999959
generalization error when max_depth=4: 125.49999999999959
generalization error when max_depth=5: 129.4999999999996
generalization error when max_depth=6: 130.4999999999996
generalization error when max_depth=7: 132.4999999999996
generalization error when max_depth=8: 135.4999999999996
generalization error when max_depth=9: 138.49999999999996
generalization error when max_depth=10: 139.49999999999996
generalization error when max_depth=11: 140.49999999999996
```

Max_depth =2 decision tree is：



4. I would choose the best one in point 3, because according to Occcam’s Razor, giventwo models with similar generation errors, one should prefer the simple model over the more complex model. And the first one is overfitting, the best one has the smallest generation error.

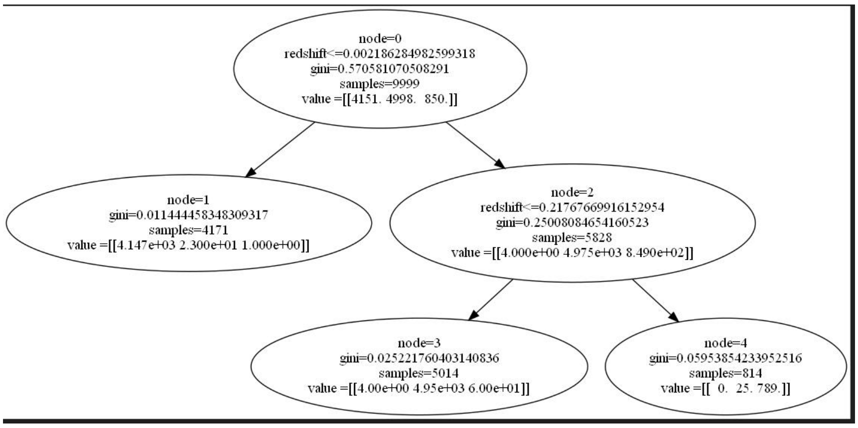
5. Redshift is the most important feature. If redshift<=0.002, it will be stars, if 0.002<redshift<=0.218, it will be galaxies, if redshift>0.218, it will be quasars.

6. For the left nodes of stars, before splitting the training error is 23+1 = 24, generation error is (24+0.5*1)=24.5. After splitting, the training error is 22+1+1=24,generation error is (24+0.5*2)=25. So it could be pruned so as to improve the generalization error.

For the right nodes, before splitting the training error is 4+849 = 853, generation error is (853+0.5*1)=853.5. After splitting, the training error is 4+60+25=89,generation error is (89+0.5*2)=90. This shouldn’t be pruned.

Therefore, the left two leaf nodes should be pruned.

7. Yes, it indeed improve the generalization error. The result is:


node=0
redshift<=0.002186284982599318
gini=0.570581070508291
samples=9999
value =[[4151. 4998. 850.]]
node=1
gini=0.011444458348309317
samples=4171
value =[[4.147e+03 2.300e+01 1.000e+00]]
node=2
redshift<=0.21767669916152954
gini=0.25008084654160523
samples=5828
value =[[4.000e+00 4.975e+03 8.490e+02]]
node=3
gini=0.025221760403140836
samples=5014
value =[[4.00e+00 4.95e+03 6.00e+01]]
node=4
gini=0.05953854233952516
samples=814
value =[[0. 25. 789.]]

Q4

1. After turn each document into a vector in the Euclidean space, we get X_train_counts.shape

with (800, 14551).

a) Average accuracy of MultinomialNB is 0.781250, Average accuracy of GaussianNB is 0.720000

b) The best classifier is multinomial naive Bayes classifier, its accuracy is 0.781250, a random classifier's accuracy is 0.112500. Random classifier is not as good as multinomial naive Bayes classifier.

c) Average accuracy of MultinomialNB with stopwords is 0.697500. The accuracy is worsen. Because stop words are words like 'and', 'the', 'him', which are presumed to be uninformative in representing the content of a text, and which may be removed to avoid them being construed as signal for prediction. The removal of stop words would allow really meaningful words to come into play, and their presence would mislead the classification model.

d) In the first task, considering only the documents related to "use of guns", "hockey","Mac hardware", its average accuracy is 0.940000. In the second task , considering "Mac hardware","IBM hardware" and "electronics", its average accuracy is 0.686667.

The first tasks has great differences in each topic and can be accurately classified by word vector. The second tasks are all related to hardware electronics, which are very similar with many overlapping contents and cannot be well classified.