ALDA Fall 2021 – Homework 3

GITHUB Repository - https://github.ncsu.edu/sjanard/engr-ALDA-fall2021-H12

Homework Team 12 – HW12

Sudharsan Janardhanan – sjanard
Sriram Sudharsan – ssudhar
Pradyumna Vemuri – pvemuri

1. (16 points) [KNN + CV] [Ge Gao] Considering the dataset with two real-valued inputs x1 and x2 and one binary output y in the table below. Each data point will be referred using the first column "ID" in the following. You will use KNN with Manhattan distance to predict y.

Write code in Python to perform the following tasks; if needed, you are allowed to use scipy, sklearn, and numpy packages. Please submit one code file via the NCSU GitHub repository you have been given. Show your work. Show steps for reaching the answer.

ID	x1	x2	y
1	-5.86	-2.0	⋆
2	-10.97	-1.0	⋆
3	0.79	-2.0	♠
4	-0.59	1.0	♠
5	3.63	-2.0	♠
6	2.02	-5.0	♠
7	-6.41	-1.0	⋆
8	6.13	-7.0	♠
9	-2.35	6.0	⋆
10	2.66	-3.0	♠
11	-3.71	2.0	⋆
12	2.4	1.0	⋆

(a) (4 points) What is the leave-one-out cross-validation error of 1NN on this dataset?

```python
from sklearn.model_selection import LeaveOneOut
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from numpy import absolute
from numpy import mean
from sklearn.model_selection import GroupKFold

import pandas as pd
d = { 'ID': [1,2,3,4,5,6,7,8,9,10,11,12],
      'x1': [-5.86,-10.97,0.79,-0.59,3.63,2.02,-6.41,6.13,-2.35,2.66,-3.71,2.4],
      'x2': [-2.0,-1.0,-2.0,1.0,-2.0,-5.0,-1.0,-7.0,6.0,-3.0,2.0,1.0],
      'y': [0,0,1,1,1,1,0,1,0,1,0,0]}
df = pd.DataFrame(data=d)
X_data = df[['x1','x2']]
Y_data = df['y']

#-----------------------------------------------1(a)----------------------------------------------#

cv1 = LeaveOneOut()

knn1 = KNeighborsClassifier(n_neighbors=1,metric="manhattan")

MAE = cross_val_score(knn1,X_data,Y_data,scoring="neg_mean_absolute_error",cv=cv1, n_jobs=1)

print("The leave-one-out cross validation Mean absolute error is ",mean(absolute(MAE)),"\n")

#-----------------------------------------------1(a)----------------------------------------------#
```

- Here in the code, Star is encoded as 0 and Spade is encoded as 1.
- Data frame, df holds the data. The code then splits the data frame into X_data – training data points and Y_data – class labels.
- Scikit-learn's LeaveOneOut class is used provide indices of the split, dividing the data into training and testing samples.
- We then create a K-nearest neighbor classifier, and we use cross_val_score to fit the split data into KNN model and get the Mean Absolute Error.

The Mean Absolute error calculated for the leave-one-out cross-validation of the 1NN model is **0.25.**

Source Code "1.py" can be found in the link - https://github.ncsu.edu/sjanard/engr-ALDA-fall2021-H12 in the folder "HW3".

Output:

```
The leave-one-out cross-validation Mean absolute error is  0.25
```

(b) (2 points) What are the 3 nearest neighbors for data points 3 and 10 respectively.

```python
def get_my_key(obj):
    return obj['distance']

def sorting_dist(P):
  dist = [ {'index': i + 1, 'distance': pdist ( [P, [x,y]], metric = 'cityblock'), 'coords': (x,y)} for i,(x,y) in enumerate(zip(X_data['x1'],X_data['x2']))]
  dist.sort(key = get_my_key)
  return dist

nearest_pts_3 = sorting_dist([X_data['x1'][2],X_data['x2'][2]])
nearest_pts_10 = sorting_dist([X_data['x1'][9],X_data['x2'][9]])

print("For data point 3 the 3 closest neighbours respectively are data points:")
for i in range(1,4):
  print('Data Point:' ,nearest_pts_3[i]['index'],'with Distance',nearest_pts_3[i]['distance'])

print("For data point 10 the 3 closest neighbours respectively are data points:")
for i in range(1,4):
  print('Data Point:' ,nearest_pts_10[i]['index'],'Distance',nearest_pts_10[i]['distance'])
#---------------------------------------------1(b)---------------------------------------------#
```

- Here, for data points 3 and 10 we find the Manhattan distance to all the other points in the data set and store them.
- We then sort the distances in the ascending order, and based on these values print the 3 closes points to 3 and 10.

The 3 closest neighbors to data point 3 are, datapoints: 5, 10 and 6.
The 3 closest neighbors to data point 10 are, datapoints: 5, 6 and 3.

Source Code "1.py" can be found in the link - https://github.ncsu.edu/sjanard/engr-ALDA-fall2021-H12 in the folder "HW3".

Output:

```
For data point 3 the 3 closest neighbours respectively are data points:
Data Point: 5 with Distance [2.84]
Data Point: 10 with Distance [2.87]
Data Point: 6 with Distance [4.23]
For data point 10 the 3 closest neighbours respectively are data points:
Data Point: 5 Distance [1.97]
Data Point: 6 Distance [2.64]
Data Point: 3 Distance [2.87]
```

(c) (5 points) What is the 3-folded cross-validation error of 3NN on this dataset? For the ith fold, the testing dataset is composed of all the data points whose (ID mod 3 = i - 1).

```python
#---------------------------------------------1(c)---------------------------------------------#
groups = []
print("\nBased on the condition specified, all 12 data points are grouped and placed in respective folds - ",end='')

for i in range(1,len(X_data)+1,1):
  groups.append(i%3+1)

print(groups)

knn3 = KNeighborsClassifier(n_neighbors=3,metric="manhattan")

gf = GroupKFold(n_splits=3)

kf_error = cross_val_score(knn3,X_data,Y_data,cv=gf,scoring="neg_mean_absolute_error",groups=groups)

print("3-folded cross validation error for 3NN would be: ",mean(absolute(kf_error)))

#---------------------------------------------1(c)---------------------------------------------#
```

- According to the condition specified, data points whose ID mod 3 = i – 1, all the data points are being grouped into various folds.
- Example: for data point 1, 1 mod 3 would be 1, so data point one would be in fold 2.
- Then scikit-learn's GroupFold class is used to create a k-folding variant that produces non-overlapping groups of data points based on condition.
- We then use cross_val_score to evaluate the k-fold cross validation error.

The 3-folded cross-validation-error for 3NN is: 0.167.

Source Code "1.py" can be found in the link - https://github.ncsu.edu/sjanard/engr-ALDA-fall2021-H12 in the folder "HW3".

Output:

```
Based on the condition specified, all 12 data points are grouped and placed in
respective folds - [2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1]
3-folded cross validation error for 3NN would be:  0.16666666666666666
```

(d) (5 points) Based on the results of (a) and (c), can we determine which is a better classifier, 1NN or 3NN? Why? (Answers without a correct justification will get zero points.)

- As mentioned above, in (a) the mean absolute error for 1NN model is 0.25 and in (c) the cross- validation error for 3NN is 0.167.

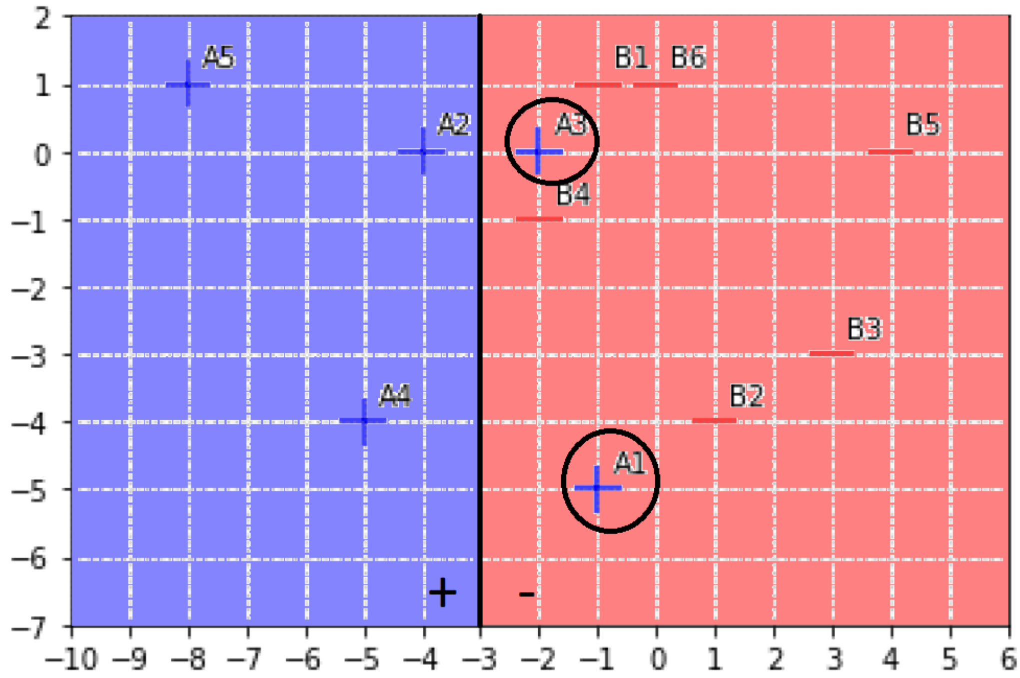
- In (a), Leave one out works by leaving out 1 data point for testing while training on the rest of the 11 data points, and based on the class label predicted the error is calculated. This splitting process is carried out for all possible combinations of original data, where the overall error is evaluated.

- On the other hand, in (c) we split the data points into various folds based on the indices. The validation process runs 3 times, where one specific fold is a holdout set or testing set and the model is trained on the other folds. The overall error is then evaluated.

- Since in (a) 1NN and in (c) 3NN are trained on different data points, because of their different approaches, we will not be able to compare the effectiveness of these models based on their error values. Since they were not trained on the same dataset, we cannot conclude that one of these models is better than the other.

2. (30 points) [Adaboost] [John Wesley Hostetter] Consider the labeled data points in Figure 1, where '+' and `-' indicate class labels. We will use AdaBoost with Separating Hyperplane to train a classier for the `+' and `-' labels. Each boosting iteration will select a horizontal or vertical Separating Hyperplane: a vertical or horizontal line that would split the space into half-spaces with a goal of minimizing the weighted training error. Breaking ties by choosing `+'. All of the data points start with uniform weights. Please display your answers for (a), (b), (d) and (e) in a single figure.

a) **(3 points) In Figure 1, draw a decision boundary corresponding to the first decision stump that the algorithm would choose (the decision boundary should be either a vertical or horizontal straight line). Label the decision boundary as (1), also indicate the '+' / '-' sides of this boundary.**



Decision Boundary 1
We draw a line X = -3 and partition based on that

b) **(2 points) Circle the point(s) that have the highest weight after the first boosting iteration. Also, report the value of the highest weight and show your calculations.**

Here you can see points A1 and A3 have the highest weights of 2.515 (refer table 2).

Initial Weights = 1/n where n = 11.
-> w1 = 0.09

A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	B6
0.09	**0.09**	**0.09**	**0.09**	**0.09**	**0.09**	**0.09**	**0.09**	**0.09**	**0.09**	**0.09**

Table 1

$\in$ = 0.09 + 0.09 = 0.18
α = 0.5ln(1 - 0.18/0.18) = 0.7581

(Correct Prediction) w2 = w1 * e^(-α)/Z1 = 0.09 * 0.4685 = 0.042165/Z1
(Incorrect Prediction) w2 = w1 * e^(α)/Z1 = 0.09 * 2.134 = 0.19206/Z1

(Normalization factor) Z1 = (2 * 0.19206) + (9 * 0.042165) = 0.76352
(Correct Prediction) w2 = 0.042165/ 0.76352 = 0.0552
(Incorrect Prediction) w2 = 0.19206/0.76352 = **2.515 is the highest weight**

Updated Table for round 2

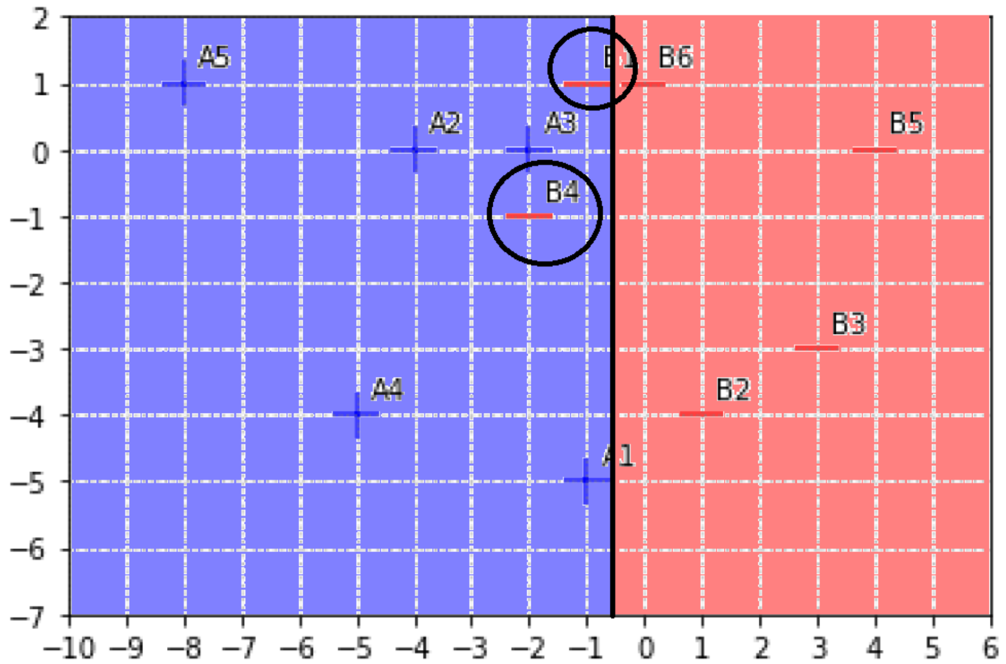
A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	B6
0.2515	**0.055**	**0.2515**	**0.055**	**0.055**	**0.055**	**0.055**	**0.055**	**0.055**	**0.055**	**0.055**

Table 2

(c) (5 points) After the labels have been re-weighted in the first boosting iteration, what is the weighted error of the decision boundary (1)?

The weighted error of decision boundary 1 is
$\in$ = 0.055 + 0.055 = 0.11

(d) (3 points) Draw the decision boundary corresponding to the second decision stump that the algorithm would choose. Label the decision boundary as (2), also indicate the '+' / '-' sides of this boundary.



Decision Boundary 2
We draw the partition at X = -0.5

α = 0.5ln(1 - 0.11/0.11) = 1.045
e^(-α) = 0.3516
e^(α) = 2.8433
(Weight for A1 and A3) = (0.055 * 2.8433) = 0.1563/Z1
(Weight for B1 and B4) = (0.2515 * 0.3516) = 0.0884/Z1
(Weight for rest of the points) = (0.055 * 0.3516) = 0.01933/Z1

(Normalization factor) Z1 = (7 * 0.01933) + (2 * 0.0884) + (2 * 0.1563) = 0.836
(Weight for A1 and A3) =0.1563/0.624 = 0.25
(Weight for B1 and B4) = 0.0884/0.624 = 0.141
(Weight for rest of the points) = 0.01938/0.624 = 0.0310

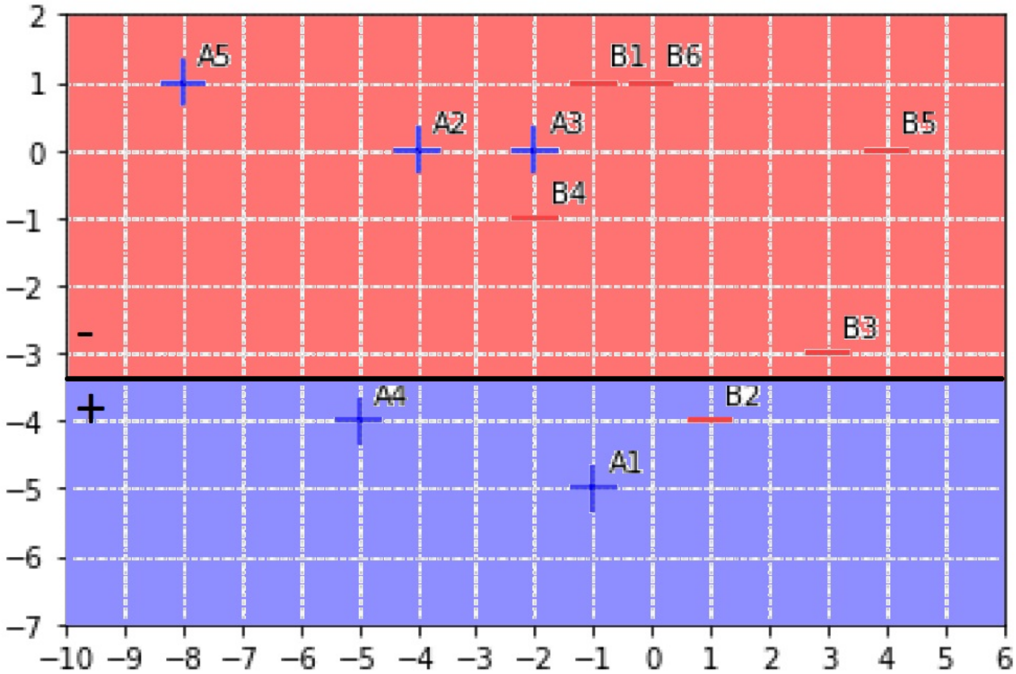
Updated Table for Round 3

A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	B6
0.141	**0.031**	**0.141**	**0.031**	**0.031**	**0.25**	**0.031**	**0.031**	**0.25**	**0.031**	**0.031**

Table 3

(e) (5 points) Next, compute the weighted error of the decision boundary (2) and draw a decision boundary corresponding to the third decision stump that the algorithm would choose. Label the decision boundary as (3), also indicate the '+' / '-' sides of this boundary.

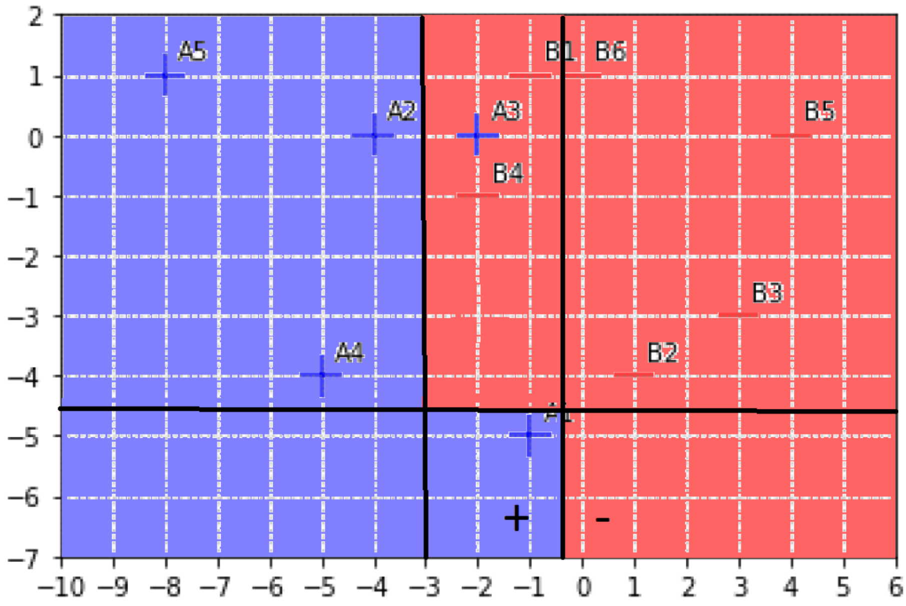
The weighted error of decision boundary 2 is
$\in$ = 0.0310 * 3 + 0.141 = 0.234
α = 0.5ln(1 - 0.234/0.234) = 0.593



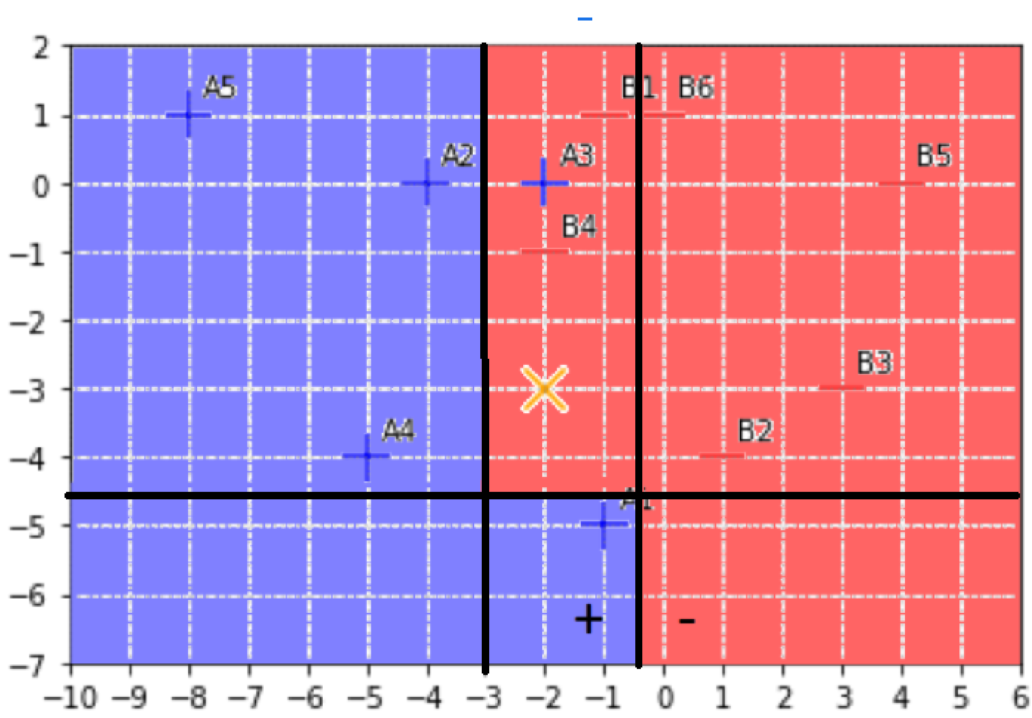
Decision Boundary 3
We draw the partition at Y = -3.5

Split #	Split at Point/Line	Alpha
1	X = - 3	0.7581
2	X = - 0.5	1.04
3	Y = -3.5	0.593

COMBINED DECISION TREE FOR A B D & E



(f) (7 points) Assuming that a "New Data point" is given (shown in the graph below), using your classier built from decision boundaries (1), (2) and (3) to predict the class label for the new data point. Provide your final classier along with the class label. Show your work.



Final Classifier

$H_{final} = \alpha_1 x(-1) + \alpha_2 x(1) + \alpha_3 x(-1)$ = -0.76 + 1.04 + -0.593 = **-0.313**

The new point(-2,-3) will belong to the Class '-' (negative) since H_{final} < 0.

3. (24 points) [Naive Bayes + Decision Tree] [Angela Zhang] Consider the training dataset below. Your goal is to build a classier to predict whether a customer will click on an ad. The output class is in the last column Clicked Ad" and the input attributes are: Image Colors", \Image Size", Product History" and Ad Placement". More specifically, you will compare Naive Bayes (NB) and Decision Tree (DT). For Naive Bayes (NB), you will use m-estimate from the lecture with m = 2 and p = 0:5 for probability estimations.

For Decision Tree (DT), you will follow the lecture's code to build your trees without pruning except that multiple-way splitting is allowed, and use Information Gain (IG) to select the best attribute. In the case of ties, break ties in favor of the leftmost feature.

(a) (18 points) Compare the performance of NB vs. DT using 2-fold cross validation (CV) and report their 2-fold CV accuracy. For the ith fold, the testing dataset is composed of all the data points whose (ID mod 2 = i   1). For each fold, show the induced Naive Bayes (in
order of left to right columns) and DT models.

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
1	Neutral	Large	Well-Known	Bottom	No
2	Neutral	Medium	Well-Known	Top	No
3	Bright	Large	New	Bottom	No
4	Neutral	Medium	New	Bottom	No
5	Neutral	Large	New	Bottom	No
6	Neutral	Large	New	Top	No
7	Bright	Large	New	Bottom	No
8	Bright	Medium	Well-Known	Bottom	Yes
9	Bright	Large	New	Top	Yes
10	Bright	Large	Well-Known	Bottom	Yes
11	Bright	Medium	Well-Known	Bottom	Yes
12	Neutral	Medium	Well-Known	Top	Yes
13	Bright	Large	New	Bottom	Yes
14	Bright	Large	Well-Known	Top	Yes

Abbreviation	Expansion
CA	Clicked Ad
I.C	Image Colors
P.H	Product History
A.P	Ad Placement
W.K	Well-Known

1-FOLD
TRAINING SET = Odd numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
1	Neutral	Large	Well-Known	Bottom	No
3	Bright	Large	New	Bottom	No
5	Neutral	Large	New	Bottom	No
7	Bright	Large	New	Bottom	No
9	Bright	Large	New	Top	Yes
11	Bright	Medium	Well-Known	Bottom	Yes
13	Bright	Large	New	Bottom	Yes

TESTING SET = Even numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
2	Neutral	Medium	Well-Known	Top	No
4	Neutral	Medium	New	Bottom	No
6	Neutral	Large	New	Top	No
8	Bright	Medium	Well-Known	Bottom	Yes
10	Bright	Large	Well-Known	Bottom	Yes
12	Neutral	Medium	Well-Known	Top	Yes
14	Bright	Large	Well-Known	Top	Yes

P(Clicked Ad = Yes) = 3/7 = 0.428
P(Clicked Ad = No) = 4/7 = 0.571

M-Estimate Probability:

$$P(a = a_j | c_i) = \frac{n_c + mp}{n + m}$$

Where m represents a parameter, p = prior probability
m=2, p=0.5

Image Color	Yes	M-Estimate Yes	No	M-Estimate No
Neutral	0/3	1/5 = 0.2	2/4	3/6 = 0.5
Bright	3/3	4/5 = 0.8	2/4	3/6 = 0.5

Image Size	Yes	M-Estimate Yes	No	M-Estimate No
Large	2/3	3/5 = 0.6	4/4	5/6 = 0.833
Medium	1/3	2/5 = 0.4	0/4	1/6 = 0.166

Product History	Yes	M-Estimate Yes	No	M-Estimate No
Well-Known	1/3	2/5 = 0.4	1/4	2/6 = 0.333
New	2/3	3/5 = 0.6	3/4	4/6 = 0.666

Ad Placement	Yes	M-Estimate Yes	No	M-Estimate No
Bottom	2/3	3/5 = 0.6	4/4	5/6 = 0.833
Top	1/3	2/5 = 0.4	0/4	1/6 = 0.166

Classification - Testing Data

<u>When ID = 2:</u>

Case 1: Clicked Ad = Yes

Prob: P(I.C = Neutral) * P(I.S = Medium) * P(P.H = Well-Known) * P(A.P =Top) * P(C.A = YES)

= 0.2 * 0.4 * 0.4 * 0.4 * 0.428

= 0.00547

Case 2: Clicked Ad = No

Prob: P(I.C = Neutral) * P(I.S = Medium) * P(P.H = Well-Known) * P(A.P =Top) * P(C.A = NO)

= 0.5 * 0.166 * 0.333 * 0.166 * 0.571

= 0.00261

P(Clicked Ad = Yes) > P(Clicked Ad = No)

Therefore, Prediction = Yes

<u>When ID = 4:</u>

Image Colors	Image Size	Product History	Ad Placement
Neutral	Medium	New	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Neutral) * P(I.S = Medium) * P(P.H = New) * P(A.P =Bottom) * P(C.A = YES)

= 0.2 * 0.4 * 0.6 * 0.6 * 0.428

= 0.01232

Case 2: Clicked Ad = No

Prob: P(I.C = Neutral) * P(I.S = Medium) * P(P.H = New) * P(A.P =Bottom) * P(C.A = NO)

= 0.5 * 0.166 * 0.666 * 0.833 * 0.571

= 0.02629

P(Clicked Ad = No) > P(Clicked Ad = Yes)

Therefore, Prediction = No

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value		
4	Neutral	Medium	New	Bottom	No	No		

When ID = 6:

Image Colors	Image Size	Product History	Ad Placement
Neutral	Large	New	Top

Case 1: Clicked Ad = Yes

Prob: P(I.C = Neutral) * P(I.S = Large) * P(P.H = New) * P(A.P =Top) * P(C.A = YES)
= 0.2 * 0.6 * 0.6 * 0.4 * 0.428
= 0.01232

Case 2: Clicked Ad = No

Prob: P(I.C = Neutral) * P(I.S = Large) * P(P.H = New) * P(A.P =Top) * P(C.A = NO)
= 0.5 * 0.833 * 0.666 * 0.166 * 0.571
= 0.02629

P(Clicked Ad = No) > P(Clicked Ad = Yes)
Therefore, Prediction = No

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
6	Neutral	Large	New	Top	No	No

When ID = 8:

Image Colors	Image Size	Product History	Ad Placement
Bright	Medium	Well-Known	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Medium) * P(P.H = W.K) * P(A.P =Bottom) * P(C.A = YES)
= 0.8 * 0.4 * 0.4 * 0.6 * 0.428
= 0.03287

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Medium) * P(P.H = W.K) * P(A.P =Bottom) * P(C.A = NO)
= 0.5 * 0.166 * 0.333 * 0.833 * 0.571
= 0.01314

P(Clicked Ad = Yes) > P(Clicked Ad = No)

Therefore, Prediction = Yes

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
8	Bright	Medium	Well-Known	Bottom	Yes	Yes

<u>When ID = 10:</u>

Image Colors	Image Size	Product History	Ad Placement
Bright	Large	Well-Known	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = W.K) * P(A.P =Bottom) * P(C.A = YES)
= 0.8 * 0.6 * 0.4 * 0.6 * 0.428
= 0.0493

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = W.K) * P(A.P =Bottom) * P(C.A = NO)
= 0.5 * 0.833 * 0.333 * 0.833 * 0.571
= 0.0659

P(Clicked Ad = No) > P(Clicked Ad = Yes)
Therefore, Prediction = No

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
10	Bright	Large	Well-Known	Bottom	Yes	No

<u>When ID = 12:</u>

Image Colors	Image Size	Product History	Ad Placement	
Neutral	Medium	Well-Known	Top	Yes

Case 1: Clicked Ad = Yes

Prob: P(I.C = Neutral) * P(I.S = Medium) * P(P.H = W.K) * P(A.P =Top) * P(C.A = YES)
= 0.2 * 0.4 * 0.4 * 0.4 * 0.428
= 0.00547

Case 2: Clicked Ad = No

Prob: P(I.C = Neutral) * P(I.S = Medium) * P(P.H = W.K) * P(A.P =Top) * P(C.A = NO)
= 0.5 * 0.166 * 0.333 * 0.166 * 0.571
= 0.00261

P(Clicked Ad = Yes) > P(Clicked Ad = No)
Therefore, Prediction = Yes

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
12	Neutral	Medium	Well-Known	Top	Yes	Yes

When ID = 14:

Image Colors	Image Size	Product History	Ad Placement
Bright	Large	Well-Known	Top

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = W.K) * P(A.P =Top) * P(C.A = YES)
= 0.8 * 0.6 * 0.4 * 0.4 * 0.428
= 0.0328

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = W.K) * P(A.P =Top)* P(C.A = NO)
= 0.5 * 0.833 * 0.333 * 0.166 * 0.571
= 0.01314

P(Clicked Ad = Yes) > P(Clicked Ad = No)
Therefore, Prediction = Yes

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
14	Bright	Large	Well-Known	Top	Yes	Yes

					Naïve Bayes Theorem		
			FOLD 1				
ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value	F
2	Neutral	Medium	Well-Known	Top	No	Yes	N
4	Neutral	Medium	New	Bottom	No	No	T
6	Neutral	Large	New	Top	No	No	
8	Bright	Medium	Well-Known	Bottom	Yes	Yes	
10	Bright	Large	Well-Known	Bottom	Yes	No	
12	Neutral	Medium	Well-Known	Top	Yes	Yes	
14	Bright	Large	Well-Known	Top	Yes	Yes	

True Positive(s): 3
True Negative(s): 2
False Positives(s): 1
False Negatives(s): 1

FOLD 2

Training Data consists of even numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
2	Neutral	Medium	Well-Known	Top	No
4	Neutral	Medium	New	Bottom	No
6	Neutral	Large	New	Top	No
8	Bright	Medium	Well-Known	Bottom	Yes
10	Bright	Large	Well-Known	Bottom	Yes
12	Neutral	Medium	Well-Known	Top	Yes
14	Bright	Large	Well-Known	Top	Yes

Testing Data consists of odd numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
1	Neutral	Large	Well-Known	Bottom	No
3	Bright	Large	New	Bottom	No
5	Neutral	Large	New	Bottom	No
7	Bright	Large	New	Bottom	No
9	Bright	Large	New	Top	Yes
11	Bright	Medium	Well-Known	Bottom	Yes
13	Bright	Large	New	Bottom	Yes

P(Clicked Ad = Yes) = 4/7 = 0.571
P(Clicked Ad = No) = 3/7 = 0.428

Image Color	Yes	M-Estimate Yes	No	M-Estimate No
Neutral	1/4	2/6 = 0.333	3/3	4/5 = 0.8
Bright	3/4	4/6 = 0.666	0/3	1/5 = 0.2

Image Size	Yes	M-Estimate Yes	No	M-Estimate No
Large	2/4	3/6 = 0.5	1/3	2/5 = 0.4
Medium	2/4	3/6 = 0.5	2/3	3/5 = 0.6

Product History	Yes	M-Estimate Yes	No	M-Estimate No
Well-Known	4/4	5/6 = 0.833	1/3	2/5 = 0.4
New	0/4	1/5 = 0.166	2/3	3/5 = 0.6

Ad Placement	Yes	M-Estimate Yes	No	M-Estimate No
Bottom	2/4	3/6 = 0.5	1/3	2/5 = 0.4
Top	2/4	3/6 = 0.5	2/3	3/5 = 0.6

Classification - Testing Data

When ID = 1:

ID	Image Colors	Image Size	Product History	Ad Placement
1	Neutral	Large	Well-Known	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Neutral) * P(I.S = Large) * P(P.H = Well-Known) * P(A.P =Bottom) * P(C.A = YES)

= 0.333 * 0.5 * 0.833 * 0.5 * 0.571

= 0.0395

Case 2: Clicked Ad = No

Prob: P(I.C = Neutral) * P(I.S = Large) * P(P.H = Well-Known) * P(A.P =Bottom) * P(C.A = NO)

= 0.8 * 0.4 * 0.4 * 0.4 * 0.428

= 0.0219

P(Clicked Ad = Yes) > P(Clicked Ad = No)

Therefore, Prediction = Yes

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
1	Neutral	Large	Well-Known	Bottom	No	Yes

When ID = 3:

ID	Image Colors	Image Size	Product History	Ad Placement
3	Bright	Large	New	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = YES)
= 0.666 * 0.5 * 0.166 * 0.5 * 0.571
= 0.01578

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = NO)
= 0.2 * 0.4 * 0.6 * 0.4 * 0.428
= 0.0082

P(Clicked Ad = Yes) > P(Clicked Ad = No)
Therefore, Prediction = Yes

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value	FOLD 2 Prediction Statistics
3	Bright	Large	New	Bottom	No	Yes	Total Number of Points: 7

When ID = 5:

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
5	Neutral	Large	New	Bottom	No

Case 1: Clicked Ad = Yes

Prob: P(I.C = Neutral) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = YES)
= 0.333 * 0.5 * 0.166 * 0.5 * 0.571
= 0.00789

Case 2: Clicked Ad = No

Prob: P(I.C = Neutral) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = NO)
= 0.8 * 0.4 * 0.6 * 0.4 * 0.428
= 0.0328

P(Clicked Ad = No) > P(Clicked Ad = Yes)
Therefore, Prediction = No

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
6	Neutral	Large	New	Top	No	No

When ID = 7:

ID	Image Colors	Image Size	Product History	Ad Placement
7	Bright	Large	New	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = YES)
= 0.666 * 0.5 * 0.166 * 0.5 * 0.571
= 0.0157

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = NO)
= 0.2 * 0.4 * 0.6 * 0.4 * 0.428
= 0.0082

P(Clicked Ad = Yes) > P(Clicked Ad = No)
Therefore, Prediction = Yes

When ID = 9:

x	Image Colors	Image Size	Product History	Ad Placement
9	Bright	Large	New	Top

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Top) * P(C.A = YES)
= 0.666 * 0.5 * 0.166 * 0.5 * 0.571
= 0.0157

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Top) * P(C.A = NO)
= 0.2 * 0.4 * 0.6 * 0.6 * 0.428
= 0.0118

P(Clicked Ad = Yes) > P(Clicked Ad = No)
Therefore, Prediction = Yes

When ID = 11:

x	Image Colors	Image Size	Product History	Ad Placement
11	Bright	Medium	Well-Known	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Medium) * P(P.H = W.K) * P(A.P =Bottom) * P(C.A = YES)
= 0.666 * 0.5 * 0.833 * 0.5 * 0.571
= 0.0791

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Medium) * P(P.H = W.K) * P(A.P =Bottom) * P(C.A = NO)
= 0.2 * 0.6 * 0.4* 0.4 * 0.428
= 0.0082

P(Clicked Ad = Yes) > P(Clicked Ad = No)
Therefore, Prediction =Yes

x	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
11	Bright	Medium	Well-Known	Bottom	Yes	Yes

When ID = 13:

x	Image Colors	Image Size	Product History	Ad Placement
13	Bright	Large	New	Bottom

Case 1: Clicked Ad = Yes

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = YES)
= 0.666 * 0.5 * 0.166 * 0.5 * 0.571
= 0.0157

Case 2: Clicked Ad = No

Prob: P(I.C = Bright) * P(I.S = Large) * P(P.H = New) * P(A.P =Bottom) * P(C.A = NO)
= 0.2 * 0.4 * 0.6 * 0.4 * 0.428
= 0.0082

P(Clicked Ad = Yes) > P(Clicked Ad = No)
Therefore, Prediction = Yes

					Naïve Bayes Theorem		
			FOLD 1				
ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value	F
2	Neutral	Medium	Well-Known	Top	No	Yes	N
4	Neutral	Medium	New	Bottom	No	No	T
6	Neutral	Large	New	Top	No	No	
8	Bright	Medium	Well-Known	Bottom	Yes	Yes	
10	Bright	Large	Well-Known	Bottom	Yes	No	
12	Neutral	Medium	Well-Known	Top	Yes	Yes	
14	Bright	Large	Well-Known	Top	Yes	Yes	
			FOLD 2				
x	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value	F
1	Neutral	Large	Well-Known	Bottom	No	Yes	N
3	Bright	Large	New	Bottom	No	Yes	T
5	Neutral	Large	New	Bottom	No	No	
7	Bright	Large	New	Bottom	No	Yes	
9	Bright	Large	New	Top	Yes	Yes	
11	Bright	Medium	Well-Known	Bottom	Yes	Yes	
13	Bright	Large	New	Bottom	Yes	Yes	

Total number of correct predictions = 9
Total number of predictions = 14

Accuracy = (9/14)*100 = 0.6428*100 = 64.28%

Training Set = Odd Numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
1	Neutral	Large	Well-Known	Bottom	No
3	Bright	Large	New	Bottom	No
5	Neutral	Large	New	Bottom	No
7	Bright	Large	New	Bottom	No
9	Bright	Large	New	Top	Yes
11	Bright	Medium	Well-Known	Bottom	Yes
13	Bright	Large	New	Bottom	Yes

Testing Set = Even Numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
2	Neutral	Medium	Well-Known	Top	No
4	Neutral	Medium	New	Bottom	No
6	Neutral	Large	New	Top	No
8	Bright	Medium	Well-Known	Bottom	Yes
10	Bright	Large	Well-Known	Bottom	Yes
12	Neutral	Medium	Well-Known	Top	Yes
14	Bright	Large	Well-Known	Top	Yes

Level 0

H(CA) = $-\ 3/7\ log\ _2(3/7)\ -\ 4/7 log\ _2(4/7)\ =$

= 0.985

IMAGE COLORS

H(CA|IC=Neutral): $0\ -\ 2/2\ log\ _2(2/2)$

= 0

H(CA|IC=Bright): $-\ 3/5\,log\ _2(3/5)\ -\ 2/5\,log\ _2(2/5)$
= 0.97095

H(CA | IC): $2/7\ *\ (0)\ +\ 5/7\ *\ (0.97095)$
= 0.69353

Information Gain (IC): 0.985-0.69353
= 0.2914

IMAGE SIZE

H(CA|IS=Large): $-\ 2/6\,log\ _2(2/6)\ -\ 4/6\,log\ _2(4/6)$
= 0.9182

H(CA|IS=Medium): $-\ 1/1\,log\ _2(1/1)\ -\ 0$
= 0

H(CA | IS): $1/7\ *\ (0)\ +\ 6/7\ *\ (0.9182)$
= 0.78702

Information Gain (IS): 0.985-0.78702
= 0.1979

PRODUCT HISTORY

H(CA|PH=W.K)=$-\ 1/2\,log\ _2(1/2)\ -\ 1/2\,log\ _2(1/2)$
= 1

H(CA|PH=New)=$-\ 2/5\,log\ _2(2/5)\ -\ 3/5\,log\ _2(3/5)$
= 0.97095

H(CA | PH) = $2/7\ *\ (1)\ +\ 5/7\ *\ (0.97095)$
= 0.9792
IG(PH) = 0.985-0.9792
= 0.005

Ad Placement

H(CA | AP = Bottom) = $-\ 2/6\, log\ _2(2/6)\ -\ 4/6\, log\ _2(4/6)$

= 0.91829

H(CA | AP = Top) = $-\ 1/1\, log\ _2(1/1)\ -\ 0$

= 0

H(CA | AP) = $1/7\ *\ (0)\ +\ 6/7\ *\ (0.9182)$

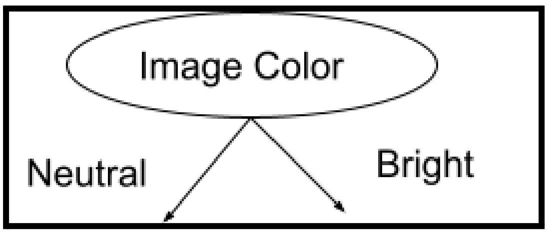
= 0.78702

IG (AP) = 0.985 - 0.78702

= 0.19798

Attribute	Information Gain
Image Color	0.2914
Image Size	0.1979
Product History	0.005
Ad Placement	0.19798

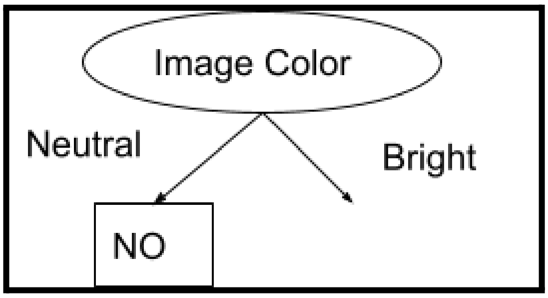
Image Color has the highest information Gain and is selected as the subsequent root.



When Image Color = Neutral

Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
Neutral	Large	Well-Known	Bottom	No
Neutral	Large	New	Bottom	No

From the table above, we see when Image Color = Neutral, Clicked Ad = “No”.



When Image Color = Bright

Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
Bright	Large	New	Bottom	No
Bright	Large	New	Bottom	No
Bright	Large	New	Top	Yes
Bright	Medium	Well-Known	Bottom	Yes
Bright	Large	New	Bottom	Yes

H(CA) = $- 3/5\, log\ _2(3/5)\ -\ 2/5\, log\ _2(2/5)$

= 0.97095

Image Size

H(CA|IS=Large): $- 2/4\, log\ _2(2/4)\ -\ 2/4\, log\ _2(2/4)$

= 1

H(CA|IS=Medium): $- 1/1\, log_{2}(1/1) - 0$

= 0

H(CA | IS): $1/5 * (0) + 4/5 * (1)$

= 0.80

Information Gain (IS): 0.97095-0.80

= 0.17095

Product History

H(CA | P.H = W.K) = $- 1/1\, log_{2}(1/1) - 0$

= 0

H(CA | P.H = New) = $- 2/4\, log_{2}(2/4) - 2/4\, log_{2}(2/4)$

= 1

H(CA | P.H) = $1/5 * (0) + 4/5 * (1)$

= 0.80

Information Gain (PH): 0.97095-0.80

= 0.17095

Ad Placement

H(CA | A.P = Bottom) = $- 2/4\, log_{2}(2/4) - 2/4\, log_{2}(2/4)$

= 1

H(CA | P.H = Top) = $- 1/1\, log_{2}(1/1) - 0$

= 0

H(CA | A.P) = $1/5 * (0) + 4/5 * (1)$

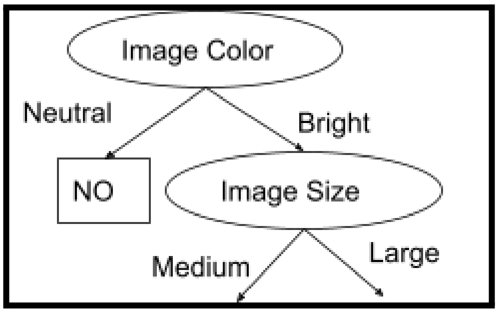
= 0.80

Information Gain (PH): 0.97095-0.80

= 0.17095

Attribute	Information Gain
Image Size	0.17095
Product History	0.17095
Ad Placement	0.17095

As our attributes have equal Information Gain, we break the tie with the left-most attribute. In our case, this is "Image Size"

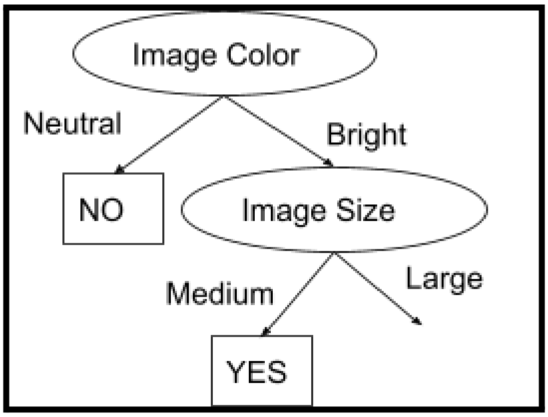


Level 2

When Image colors = Bright and Image Size = Medium

Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
Bright	Medium	Well-Known	Bottom	Yes

From the table above, we see when Image Color = Bright, Image Size = Medium and Clicked Ad = "Yes".



When Image colors = Bright and Image Size = Large

Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
Bright	Large	New	Bottom	No
Bright	Large	New	Bottom	No
Bright	Large	New	Top	Yes
Bright	Large	New	Bottom	Yes

H(I.C = Bright and I.S = Large) = $-\ 2/4\,log\ _2(2/4)\ -\ 2/4\,log\ _2(2/4)$

= 1

<u>Product History</u>

H(CA | P.H = New) = $-\ 2/4\,log\ _2(2/4)\ -\ 2/4\,log\ _2(2/4)$

= 1

H(CA | P.H = W.K) = 0

= 0

H(CA | P.H) = $4/4\ *\ (1)\ +\ 0$

= 1

Information Gain (Product History): 1 - 1
= 0

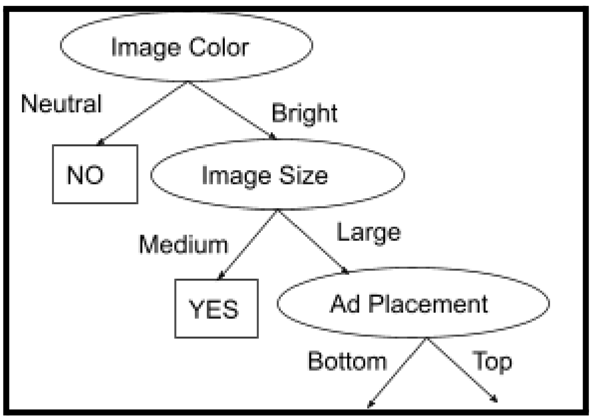
Ad Placement

H(CA | A.P = Bottom) = $-\ 1/3\, log\ _2(1/3)\ -\ 2/3\, log\ _2(2/3)$
= 0.91829
H(CA | A.P = Top) = $-\ 1/1\, log\ _2(1/1)$
= 0
H(CA | A.P) = $3/4\ *\ (0.91829)$**+**$1/4\ *\ (0)$
= 0.6887

Information Gain (Product History): 1 - 0.6887
= 0.3113

Attribute	**Information Gain**
Product History	**0**
Ad Placement	**0.3113**

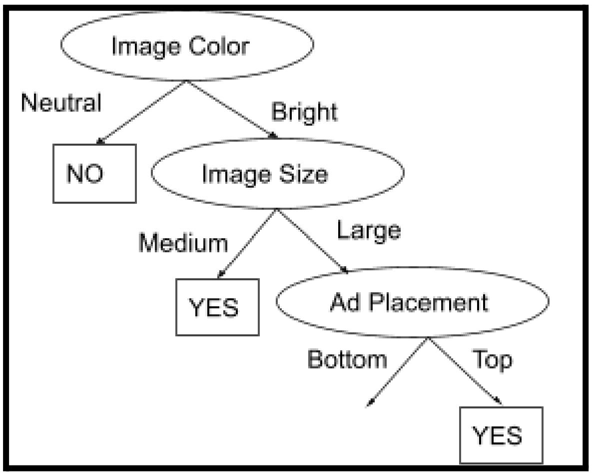
LEVEL 3



When Image colors = Bright and Image Size = Large and Ad Placement = Top

Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
Bright	Large	New	Top	Yes

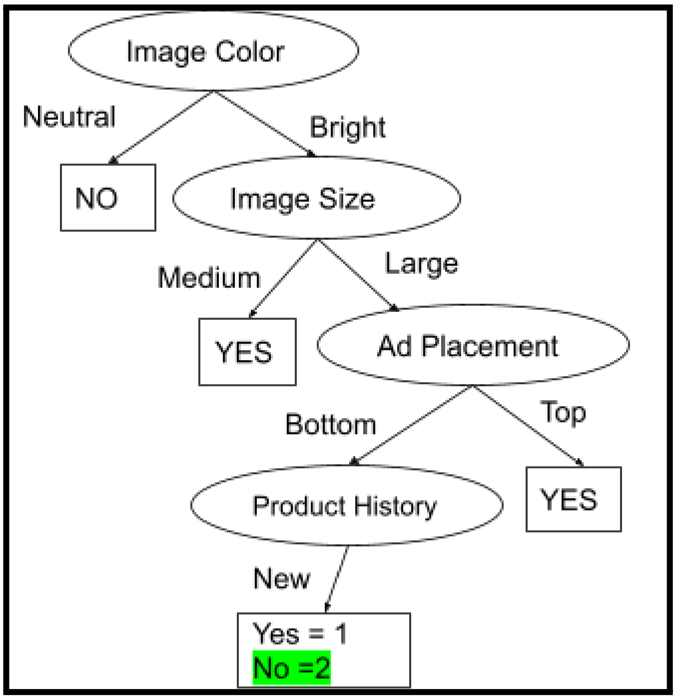
When Image colors = Bright and Image Size = Large and Ad Placement = Top, Clicked Ad is set to Yes.



When Image colors = Bright and Image Size = Large and Ad Placement = Bottom

Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
Bright	Large	New	Bottom	No
Bright	Large	New	Bottom	No
Bright	Large	New	Bottom	Yes

We cannot classify this further as we have 2 No and 1 Yes for the same set of attributes. We take majority No as our result.



ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
2	Neutral	Medium	Well-Known	Top	No	No
4	Neutral	Medium	New	Bottom	No	No
6	Neutral	Large	New	Top	No	No
8	Bright	Medium	Well-Known	Bottom	Yes	Yes
10	Bright	Large	Well-Known	Bottom	Yes	No
12	Neutral	Medium	Well-Known	Top	Yes	No
14	Bright	Large	Well-Known	Top	Yes	Yes

FOLD 2

Training Data: Even numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
2	Neutral	Medium	Well-Known	Top	No
4	Neutral	Medium	New	Bottom	No
6	Neutral	Large	New	Top	No
8	Bright	Medium	Well-Known	Bottom	Yes
10	Bright	Large	Well-Known	Bottom	Yes
12	Neutral	Medium	Well-Known	Top	Yes
14	Bright	Large	Well-Known	Top	Yes

Testing Data: Odd numbered IDs

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
1	Neutral	Large	Well-Known	Bottom	No
3	Bright	Large	New	Bottom	No
5	Neutral	Large	New	Bottom	No
7	Bright	Large	New	Bottom	No
9	Bright	Large	New	Top	Yes
11	Bright	Medium	Well-Known	Bottom	Yes
13	Bright	Large	New	Bottom	Yes

LEVEL 0

H(Clicked Ad [CA]) = $-\ 4/7\, log\ _2(4/7)\ -\ 3/7\, log\ _2(3/7)\ =$

= 0.985

Image Colors

H(CA | IC = Neutral) = $-\ 1/4\, log\ _2(1/4)\ -\ 3/4\, log\ _2(3/4)\ =$

= 0.8112

H(CA | IC = Bright) = $-\ 3/3\, log\ _2(3/3)\ -\ 0\ =$

= 0

H(CA | IC) = $4/7 \ * \ (0.8112) \ + \ 3/7 \ * \ (0) \ =$
= 0.4635

Information Gain (Image Colors) = 0.985 - 0.4635
= 0.5215

<u>Image Size</u>

<u>H(CA | IS = Large) =</u> $- \ 2/3\, log\ _2(2/3) \ - \ 1/3\, log\ _2(1/3)$
0.9182

<u>H(CA | IS = Medium) =</u> $- \ 2/4\, log\ _2(2/4) \ - \ 2/4\, log\ _2(2/4)$
= 1
H(CA | IS) = $4/7 \ * \ (1) \ + \ 3/7 \ * \ (0.9182) \ =$
= 0.9649

Information Gain (Image Colors) = 0.985 - 0.9649
= 0.0201

<u>Product History</u>

<u>H(CA | P.H=W.K) =</u> $- \ 4/5\, log\ _2(4/5) \ - \ 1/5\, log\ _2(1/5)$
0.7219

<u>H(CA | P.H=New) =</u> $0 \ - \ 2/2\, log\ _2(2/2)$
= 0
H(CA | PH) = $2/7 \ * \ (0) \ + \ 5/7 \ * \ (0.7219) \ =$
= 0.51564

Information Gain (Image Colors) = 0.985 - 0.51564
= 0.46936

<u>Ad Placement</u>

<u>H(CA | A.P = Bottom) =</u> $- \ 1/3\, log\ _2(1/3) \ - \ 2/3\, log\ _2(2/3)$
0.91829

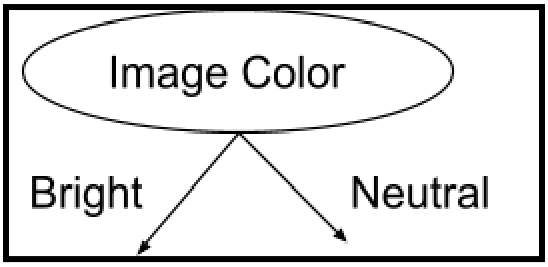
<u>H(CA | P.H=Top) =</u> $- \ 2/4\, log\ _2(2/4) \ - \ 2/4\, log\ _2(2/4)$
= 1

H(CA | AP) = $4/7 \ * \ (1) \ + \ 3/7 \ * \ (0.91829) \ =$
= 0.96498

Information Gain (Image Colors) = 0.985 - 0.96498
= 0.02002

Attribute	**Information Gain**
Image Color	**0.5215**
Image Size	**0.0201**
Product History	**0.46936**
Ad Placement	**0.02002**

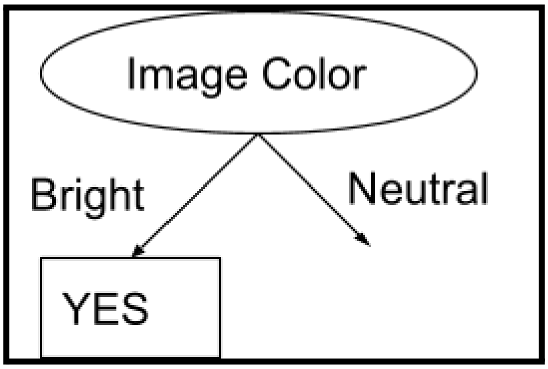
Image Color has the highest Information Gain. It is selected as our root.



When Image Color = Bright

8	Bright	Medium	Well-Known	Bottom	Yes
10	Bright	Large	Well-Known	Bottom	Yes
14	Bright	Large	Well-Known	Top	Yes

When Image color = Bright, Clicked Ad = Yes



When Image Color = Neutral

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
2	Neutral	Medium	Well-Known	Top	No
4	Neutral	Medium	New	Bottom	No
6	Neutral	Large	New	Top	No
12	Neutral	Medium	Well-Known	Top	Yes

H(I.C) = $-\ 1/4\, log\ _2(1/4)\ -\ 3/4\, log\ _2(3/4)$

= 0.81127

Image Size

H(CA | I.S = Large) = $-\ 1/1\, log\ _2(1/1)\ -\ 0$

= 0

H(CA | I.S = Medium) = $-\ 1/3\, log\ _2(1/3)\ -\ 2/3\, log\ _2(2/3)$

= 0.91829

H(CA | I.S) = $1/4\ *\ (0)\ +\ 3/4\ *\ (0.91829)\ =$

= 0.6887

Information Gain (Image Colors) = 0.81127 - 0.6887

= 0.1225

Product History

H(CA | P.H = W.K) = $-\ 1/2\, log\ _2(1/2)\ -\ 1/2\, log\ _2(1/2)$

= 1

H(CA | P.H = New) = $2/2\,log_{2}(2/2) - 0$

= 0

H(CA | P.H) = $2/4 * (1) + 2/4 * (0) =$

= 0.50

Information Gain(Product History) = 0.81127 - 0.50

= **0.31127**

Ad Placement

H(CA | A.P = Top) = $-1/3\,log_{2}(1/3) - 2/3\,log_{2}(2/3)$

= 0.91829

H(CA | A.P = Bottom) = $1/1\,log_{2}(1/1) - 0$

= 0

H(CA | A.P) = $1/4 * (0) + 3/4 * (0.91829) =$

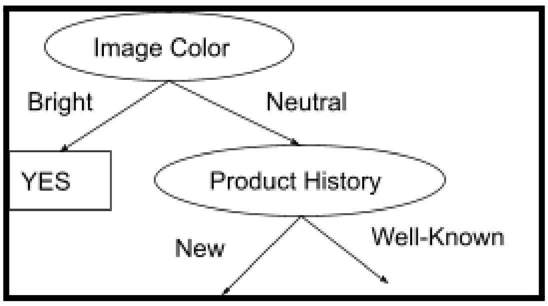
= 0.6887

Information Gain(A.P) = 0.81127 - 0.6887

= **0.12257**

Attribute	**Information Gain**
Image Size	**0.1225**
Product History	**0.31127**
Ad Placement	**0.12257**

Product History has the highest Information Gain and will be selected as our next node.

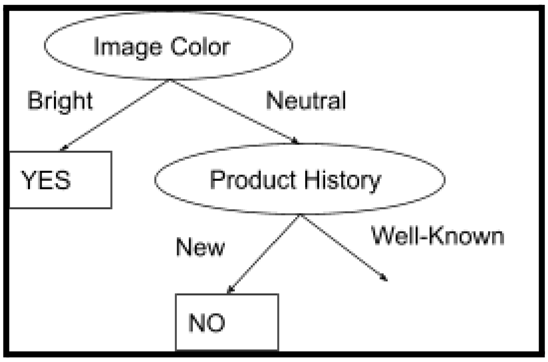


LEVEL 2

<u>When Image Color = Neutral and Product History = New</u>

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
4	Neutral	Medium	New	Bottom	No
6	Neutral	Large	New	Top	No

When Image Color = Neutral and Product History = New, Clicked Ad = No.



<u>When Image Color = Neutral and Product History = Well Known</u>

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
2	Neutral	Medium	Well-Known	Top	No
12	Neutral	Medium	Well-Known	Top	Yes

<u>When Image Color = Neutral and Product History = Well Known</u>

<u>H(CA) =</u> $- 1/2\, log\ _2 (1/2)\ -\ 1/2\, log\ _2 (1/2)$

= 1

<u>Image Size</u>

H(CA | IS = Medium) = $- 1/2\, log\ _2 (1/2)\ -\ 1/2\, log\ _2 (1/2)$

= 1

Information Gain(I.S) = 1 - (2/2) * 1 = 0

<u>Ad Placement</u>

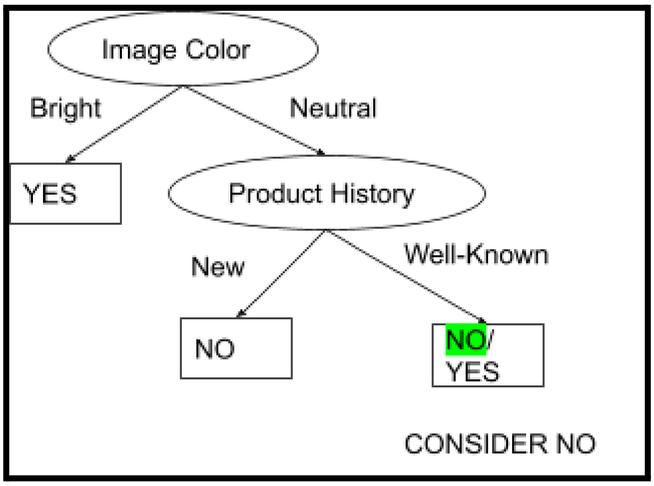
H(CA | IS = Top) =

= 1

Information Gain(I.S) = 1 - (2/2) * 1

= 0

We notice no additional information gain post splitting. Hence we stop building the tree at this point. As instructed, we break the split of Product History → well known by breaking in favour of NO



Fold 2 Predictions

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value	FOLD 2 Prediction Statistics
1	Neutral	Large	Well-Known	Bottom	No	No	Number of Correct Predictions: 5
3	Bright	Large	New	Bottom	No	Yes	Total Number of Points: 7
5	Neutral	Large	New	Bottom	No	No	
7	Bright	Large	New	Bottom	No	Yes	
9	Bright	Large	New	Top	Yes	Yes	
11	Bright	Medium	Well-Known	Bottom	Yes	Yes	
13	Bright	Large	New	Bottom	Yes	Yes	

True Positive(s) = 3
True Negative(s) = 2
False Positive(s) = 2
False Negative(s) = 0

Consolidated Fold1/Fold2

			FOLD 1			
ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
2	Neutral	Medium	Well-Known	Top	No	No
4	Neutral	Medium	New	Bottom	No	No
6	Neutral	Large	New	Top	No	No
8	Bright	Medium	Well-Known	Bottom	Yes	Yes
10	Bright	Large	Well-Known	Bottom	Yes	No
12	Neutral	Medium	Well-Known	Top	Yes	No
14	Bright	Large	Well-Known	Top	Yes	Yes
			FOLD 2			
ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad	Predicted Value
1	Neutral	Large	Well-Known	Bottom	No	No
3	Bright	Large	New	Bottom	No	Yes
5	Neutral	Large	New	Bottom	No	No
7	Bright	Large	New	Bottom	No	Yes
9	Bright	Large	New	Top	Yes	Yes
11	Bright	Medium	Well-Known	Bottom	Yes	Yes
13	Bright	Large	New	Bottom	Yes	Yes

Total correctly classified points : 10
Total number of points : 14

Accuracy : 71.428 %

(b) (6 points) Based on the 2-fold CV accuracy from (a), which classifier, NB or DT, would you choose? Report your final model for the selected classifier.

On comparing the accuracies of the Naive bayes model and the Decision Tree model, we have:
Accuracy is defined as the ratio of correct predictions to the total number of predictions.

At its best case, we can classify all elements correctly, which will yield an accuracy of 1.

At its worst case, we can classify all elements incorrectly, which will yield an accuracy of 0.

Accuracy (Naive Bayes - 2-Fold) : **(9/14)*100 = 64.28%**
Accuracy (Decision Tree -2-Fold) : (10/14)*100 =71.428%

From the results, we notice our decision tree makes MORE accurate predictions and is decided to be the better classifier among the two.

Final Decision Tree:

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
1	Neutral	Large	Well-Known	Bottom	No
2	Neutral	Medium	Well-Known	Top	No
3	Bright	Large	New	Bottom	No
4	Neutral	Medium	New	Bottom	No
5	Neutral	Large	New	Bottom	No
6	Neutral	Large	New	Top	No
7	Bright	Large	New	Bottom	No
8	Bright	Medium	Well-Known	Bottom	Yes
9	Bright	Large	New	Top	Yes
10	Bright	Large	Well-Known	Bottom	Yes
11	Bright	Medium	Well-Known	Bottom	Yes
12	Neutral	Medium	Well-Known	Top	Yes
13	Bright	Large	New	Bottom	Yes
14	Bright	Large	Well-Known	Top	Yes

iNTERMEDIARY STEPS:

H(CA) = $-\ 7/14\, log\ _2(7/14)\ -\ 7/14\, log\ _2(7/14)$

= 1

Image Colors

H(CA | I.C = Neutral) = $-\ 1/6\, log\ _2(1/6)\ -\ 5/6\, log\ _2(5/6)$

= 0.650

H(CA | I.S = Bright) = $-\ 6/8\, log\ _2(6/8) - 2/8\, log\ _2(2/8)$

= 0.811

H(CA | I.S) = $6/14\ *\ (0.65)\ +\ 8/14\ *\ (0.811)\ =$

= 0.742

Information Gain (Image Colors) = 1-0.719

= 0.258

Image Size

H(CA | I.C = Large) = $-\ 4/9\ log\ _2(4/9)\ -\ 5/9\ log\ _2(5/9)$

________________**= 0.991**

H(CA | I.S = Bright) = $-\ 3/5\ log\ _2(3/5)\ -\ 2/5\ log\ _2(2/5)$

________________**= 0.9709**

H(CA | I.S) = $9/14\ *\ (0.991)\ +\ 5/14\ *\ (0.9709)\ =$

= 0.9838

Information Gain (Image Size) = 1-0.9838

= 0.0162

Product History

H(CA | P.H=W.K) = $-\ 5/7\ log\ _2(5/7)\ -\ 2/7\ log\ _2(2/7)$

= 0.8631

H(CA | P.H = New) = $-\ 2/7\ log\ _2(2/7)\ -\ 5/7\ log\ _2(5/7)$

________________= **0.8631**

H(CA | P.H) = $7/14\ *\ (0.8631)\ +\ 7/14\ *\ (0.8631)$

= 0.8631

Information Gain (PH) = 1- 0.8631 = 0.1369

Ad Placement

H(CA | I.C = Bottom) = $-\ 4/9\ log\ _2(4/9)\ -\ 5/9\ log\ _2(5/9)$

________________**= 0.991**

H(CA | I.S = Bright) = $-\ 3/5\ log\ _2(3/5)\ -\ 2/5\ log\ _2(2/5)$

________________**= 0.9709**

H(CA | I.S) = $9/14\ *\ (0.991)\ +\ 5/14\ *\ (0.9709)\ =$

= 0.9838

Information Gain (Image Size) = 1-0.9838

= 0.0162

Image Color has the highest Information Gain.

When Image Color = Bright

H(Clicked Ad) = $-\ 6/8\,log\ _2(6/8)\ -\ 2/8\,log\ _2(2/8)$
= 0.811

H(CA | IS = Large) = $-\ 4/6\,log\ _2(4/6)\ -\ 2/6\,log\ _2(2/6)$
= 0.918

H(CA | IS = Medium) = $-\ 2/2\,log\ _2(2/2)\ -\ 0$
= 0

H(CA | IS) = $6/8\ *\ (0.918)\ +\ 2/8\ *\ (0)$
= 0.688

Information Gain (IS) = 0.811 - 0.688
= 0.123

Product History

H(CA | PH = W.K) = $-\ 4/4\,log\ _2(4/4)\ -\ 0$
= 0
H(CA | PH = New) = $-\ 2/4\,log\ _2(2/4)\ -\ 2/4\,log\ _2(2/4)$
= 1
H(CA | PH) = $4/8\ *\ (0)\ +\ 4/8\ *\ (1)$
= 0.5
Information Gain (IS) = 0.811 - 0.5
= 0.311

Ad Placement

H(CA | AP = Bottom) = $-\ 4/6\,log\ _2(4/6)\ -\ 2/6\,log\ _2(2/6)$
= 0.918
H(CA | AP = Top) = $-\ 2/2\,log\ _2(2/2)\ -\ 0$
= 0
H(CA | AP) = $6/8\ *\ (0.918)\ +\ 2/8\ *\ (0)$
= 0.688
Information Gain (AP) = 0.811 - 0.688
= 0.123

Highest IG = Product History

H(Clicked Ad) = $-\ 1/6\,log\ _2(1/6)\ -\ 5/6\,log\ _2(5/6)$

= 0.650

Image Size

H(CA | IS = Large) = $0 - 3/3\,log_2(3/3)$

= 0

H(CA | IS = Medium) = $-1/3\,log_2(1/3) - 2/3\,log_2(2/3)$

= 0.918

H(CA | IS) = $3/6 * (0) + 3/6 * (0.918)$

= 0.459

Information Gain (IS) = 0.650 - 0.459

= 0.191

Product History

H(CA | PH = W.K) = $-1/3\,log_2(1/3) - 2/3\,log_2(2/3)$

= 0.918

H(CA | PH = New) = $0 - 3/3\,log_2(3/3)$

= 0

H(CA | PH) = $3/6 * (0.918) + 3/6 * (0)$

= 0.459

Information Gain (IS) = 0.650 - 0.459

= 0.191

Ad Placement

H(CA | AP = Bottom) = $0 - 3/3\,log_2(3/3)$

= 0

H(CA | AP = Top) = $-1/3\,log_2(1/3) - 2/3\,log_2(2/3)$

= 0.918

H(CA | AP) = $3/6 * (0) + 3/6 * (0.918)$

= 0.459

Information Gain (AP) = 0.650 - 0.459

= 0.191

Equal IG. Left most split (Image Size)

IC = Neutral and IS = Medium

Product History

H(CA) = $- 1/3\, log\ _2(1/3)\ -\ 2/3\, log\ _2(2/3)$
= 0.918
H(CA | PH = W.K) = $- 1/2\, log\ _2(1/2)\ -\ 1/2\, log\ _2(1/2)$
= 1
H(CA | PH = New) = $0\ -\ 1/1\, log\ _2(1/1)$
= 0
H(CA | PH) = $2/3\ *\ (1)\ +\ 1/3\ *\ (0)$
= 0.66
Information Gain (PH) = 0.918 - 0.66
= 0.258

Ad Placement

H(CA | AP = Bottom) = $0\ -\ 1/1\, log\ _2(1/1)$
= 0
H(CA | PH = New) = $- 1/2\, log\, 1/2\ -\ 1/2\, log\ _2(1/2)$
= 1
H(CA | AP) = $1/3\ *\ (0)\ +\ 2/3\ *\ (1)$
= 0.66
Information Gain (PH) = 0.918 - 0.66
= 0.258

Equal Information Gain. Breaking using leftmost split

Product History

When IC = Neutral, IS = Medium and P.H = W.K, we have 1 yes and 1 no.
When IC = Neutral, IS = Medium, P.H = New, we have 1 No
When IC = Bright, P.H = New
H(CA) = $- 2/4\, log\, 2/4\ -\ 2/4\, log\ _2(2/4)$ **= 1**

Image Size

H(CA | IS = Large) = $- 2/4\, log\, 2/4\ -\ 2/4\, log\ _2(2/4)$ **= 1**

H(CA | IS = Medium) = 0
H(CA | IS) = $4/4 \ * \ (1) \ + \ 0 \ * \ (1)$ **= 1**
IG(IS) = 1-1 = 0

Ad Placement

H(CA | AP = Bottom) = $-\ 1/3\ log\ 1/3 \ - \ 2/3\ log\ _2 (2/3)$ **= 0.918**
H(CA | IS = Top) = $-\ 1/1\ log\ 1/1 \ - \ 0$ **= 0**
H(CA | AP) = $3/4 \ * \ (0.918) \ + \ 1/4 \ * \ (0)$ **= 0.688**
IG(AP) = 1-0.688 = 0.312

Considering IC: Bright, PH: New, and AP: Top, we get Yes.
Considering IC: Bright, PH: New, and AP: Bottom, we get 1 Yes and 2 No.