CS 584-04: Machine Learning

Fall 2018 Assignment 3

Diego Martin Crespo A20432558

Question 1 (50 points)

You will use the CART algorithm to build profiles of credit card holders. The data is the CustomerSurveyData.csv. The analysis specifications are:

Target Variable

- **CreditCard**. The type of credit card held. This variable has five categories which are *American Express*, *Discover*, *MasterCard*, *Others*, and *Visa*.
- Drop all missing values in the target variable.

Nominal Predictors

- **CarOwnership**. The type of car ownership. This variable has three non-missing categories which are *Leased*, *None*, and *Own*.
- **JobCategory**. The category of the job held. This variable has six non-missing categories which are *Agriculture*, *Crafts*, *Labor*, *Professional*, *Sales*, and *Service*.
- Recode all the missing values into the *Missing* category.

You will use the Gini metric as the splitting criterion. You may want to write a Python program to assist you in answering the questions.

a) (5 points). What is the Gini metric for the root node?

The root node is the target value CreditCard. This following table shows the number of each category of Creditcard:

Discover	1344
Visa	1247
MasterCard	1200
American Express	986
Others	223

The Gini index is calculated as: $1 - \sum_{j=1}^{k} p_{ij}^2$, being p_{ji} the probabilities of each category. Each category probability is calculated as: number of category/ total number of credit cards. The total number of CreditCard is 5000. Therefore, we get a Gini of 0.7670692

b) (5 points). How many possible binary-splits that you can generate from the CarOwnership predictor?

The total possible binary-splits is calculated with: $2^{k-1} - 1$, being k the number of categories. For CarOwnership there are 3 categories. Therefore, there are 3 possible binary-splits.

c) (10 points). Calculate the Gini metric for each possibly binary split that you can generate from the CarOwnership predictor. List your answers in a table. The table should have three columns: the sequence index of the split, the contents of the two branches, the split Gini metric.

SeqIndexSplit	ContentBranches	Gini
(0), (1, 2)	(None), (Lease, Own)	0.766776
(1), (0, 2)	(Lease), (None, Own)	0.765709
(2), (0, 1)	(Own), (None, Lease)	0.766006

d) (5 points). What is the optimal split for the CarOwnership predictor?

The optimal split for CarOwnership will be (Lease), (None, Own) as it has the lowest Gini score of 0.765709.

e) (5 points). How many possible binary-splits that you can generate from the JobCategory predictor?

The total possible binary-splits is calculated with: $2^{k-1} - 1$, being k the number of categories. For JowCategory there are 6 categories. However, as it has some NaN, a new category named Missing is created for those. Therefore having 7 categories, there are 63 possible binary-splits.

f) (10 points). Calculate the Gini metric for each possibly binary split that you can generate from the JobCategory predictor. List your answers in a table. The table should have three columns: the sequence index of the split, the contents of the two branches, the split Gini metric.

Index	Index_Split	Split	Gini_Value
0	[(0,), [1, 2, 3, 4, 5, 6]]	[('Agriculture',), [1, 2, 3, 4, 5, 6]]	0.767051683
1	[(1,), [0, 2, 3, 4, 5, 6]]	[('Crafts',), [0, 2, 3, 4, 5, 6]]	0.766994208
2	[(2,), [0, 1, 3, 4, 5, 6]]	[('Labor',), [0, 1, 3, 4, 5, 6]]	0.767031516
3	[(3,), [0, 1, 2, 4, 5, 6]]	[('Professional',), [0, 1, 2, 4, 5, 6]]	0.766767818
4	[(4,), [0, 1, 2, 3, 5, 6]]	[('Sales',), [0, 1, 2, 3, 5, 6]]	0.766685431
5	[(5,), [0, 1, 2, 3, 4, 6]]	[('Service',), [0, 1, 2, 3, 4, 6]]	0.767015007
6	[(6,), [0, 1, 2, 3, 4, 5]]	[('Missing',), [0, 1, 2, 3, 4, 5]]	0.76689792
7	[(0, 1), [2, 3, 4, 5, 6]]	[('Agriculture', 'Crafts'), [2, 3, 4, 5, 6]]	0.767009188
8	[(0, 2), [1, 3, 4, 5, 6]]	[('Agriculture', 'Labor'), [1, 3, 4, 5, 6]]	0.76703902
9	[(0, 3), [1, 2, 4, 5, 6]]	[('Agriculture', 'Professional'), [1, 2, 4, 5, 6]]	0.766775204
10	[(0, 4), [1, 2, 3, 5, 6]]	[('Agriculture', 'Sales'), [1, 2, 3, 5, 6]]	0.766718785
11	[(0, 5), [1, 2, 3, 4, 6]]	[('Agriculture', 'Service'), [1, 2, 3, 4, 6]]	0.767027702
12	[(0, 6), [1, 2, 3, 4, 5]]	[('Agriculture', 'Missing'), [1, 2, 3, 4, 5]]	0.767022904
13	[(1, 2), [0, 3, 4, 5, 6]]	[('Crafts', 'Labor'), [0, 3, 4, 5, 6]]	0.766983521
14	[(1, 3), [0, 2, 4, 5, 6]]	[('Crafts', 'Professional'), [0, 2, 4, 5, 6]]	0.766701345
15	[(1, 4), [0, 2, 3, 5, 6]]	[('Crafts', 'Sales'), [0, 2, 3, 5, 6]]	0.766810602
16	[(1, 5), [0, 2, 3, 4, 6]]	[('Crafts', 'Service'), [0, 2, 3, 4, 6]]	0.767044034
17	[(1, 6), [0, 2, 3, 4, 5]]	[('Crafts', 'Missing'), [0, 2, 3, 4, 5]]	0.766993369
18	[(2, 3), [0, 1, 4, 5, 6]]	[('Labor', 'Professional'), [0, 1, 4, 5, 6]]	0.766819992
19	[(2, 4), [0, 1, 3, 5, 6]]	[('Labor', 'Sales'), [0, 1, 3, 5, 6]]	0.766748373
20	[(2, 5), [0, 1, 3, 4, 6]]	[('Labor', 'Service'), [0, 1, 3, 4, 6]]	0.766998687
21	[(2, 6), [0, 1, 3, 4, 5]]	[('Labor', 'Missing'), [0, 1, 3, 4, 5]]	0.767048763
22	[(3, 4), [0, 1, 2, 5, 6]]	[('Professional', 'Sales'), [0, 1, 2, 5, 6]]	0.767017929
23	[(3, 5), [0, 1, 2, 4, 6]]	[('Professional', 'Service'), [0, 1, 2, 4, 6]]	0.766836333
24	[(3, 6), [0, 1, 2, 4, 5]]	[('Professional', 'Missing'), [0, 1, 2, 4, 5]]	0.766744582
25	[(4, 5), [0, 1, 2, 3, 6]]	[('Sales', 'Service'), [0, 1, 2, 3, 6]]	0.766681418
26	[(4, 6), [0, 1, 2, 3, 5]]	[('Sales', 'Missing'), [0, 1, 2, 3, 5]]	0.766692943
27	[(5, 6), [0, 1, 2, 3, 4]]	[('Service', 'Missing'), [0, 1, 2, 3, 4]]	0.767027956
28	[(0, 1, 2), [3, 4, 5, 6]]	[('Agriculture', 'Crafts', 'Labor'), [3, 4, 5, 6]]	0.766992682
29	[(0, 1, 3), [2, 4, 5, 6]]	[('Agriculture', 'Crafts', 'Professional'), [2, 4, 5, 6]]	0.766700901

30	[(0, 1, 4), [2, 3, 5, 6]]	[('Agriculture', 'Crafts', 'Sales'), [2, 3, 5, 6]]	0.766827279
31	[(0, 1, 5), [2, 3, 4, 6]]	[('Agriculture', 'Crafts', 'Service'), [2, 3, 4, 6]]	0.767047498
32	[(0, 1, 6), [2, 3, 4, 5]]	[('Agriculture', 'Crafts', 'Missing'), [2, 3, 4, 5]]	0.767001469
33	[(0, 2, 3), [1, 4, 5, 6]]	[('Agriculture', 'Labor', 'Professional'), [1, 4, 5, 6]]	0.766814071
34	[(0, 2, 4), [1, 3, 5, 6]]	[('Agriculture', 'Labor', 'Sales'), [1, 3, 5, 6]]	0.766764183
35	[(0, 2, 5), [1, 3, 4, 6]]	[('Agriculture', 'Labor', 'Service'), [1, 3, 4, 6]]	0.767008889
36	[(0, 2, 6), [1, 3, 4, 5]]	[('Agriculture', 'Labor', 'Missing'), [1, 3, 4, 5]]	0.767047543
37	[(0, 3, 4), [1, 2, 5, 6]]	[('Agriculture', 'Professional', 'Sales'), [1, 2, 5, 6]]	0.767015398
38	[(0, 3, 5), [1, 2, 4, 6]]	[('Agriculture', 'Professional', 'Service'), [1, 2, 4, 6]]	0.76683154
39	[(0, 3, 6), [1, 2, 4, 5]]	[('Agriculture', 'Professional', 'Missing'), [1, 2, 4, 5]]	0.766749795
40	[(0, 4, 5), [1, 2, 3, 6]]	[('Agriculture', 'Sales', 'Service'), [1, 2, 3, 6]]	0.766699763
41	[(0, 4, 6), [1, 2, 3, 5]]	[('Agriculture', 'Sales', 'Missing'), [1, 2, 3, 5]]	0.766722013
42	[(0, 5, 6), [1, 2, 3, 4]]	[('Agriculture', 'Service', 'Missing'), [1, 2, 3, 4]]	0.767032064
43	[(1, 2, 3), [0, 4, 5, 6]]	[('Crafts', 'Labor', 'Professional'), [0, 4, 5, 6]]	0.766709046
44	[(1, 2, 4), [0, 3, 5, 6]]	[('Crafts', 'Labor', 'Sales'), [0, 3, 5, 6]]	0.766815537
45	[(1, 2, 5), [0, 3, 4, 6]]	[('Crafts', 'Labor', 'Service'), [0, 3, 4, 6]]	0.766997288
46	[(1, 2, 6), [0, 3, 4, 5]]	[('Crafts', 'Labor', 'Missing'), [0, 3, 4, 5]]	0.766997015
47	[(1, 3, 4), [0, 2, 5, 6]]	[('Crafts', 'Professional', 'Sales'), [0, 2, 5, 6]]	0.767023523
48	[(1, 3, 5), [0, 2, 4, 6]]	[('Crafts', 'Professional', 'Service'), [0, 2, 4, 6]]	0.766776099
49	[(1, 3, 6), [0, 2, 4, 5]]	[('Crafts', 'Professional', 'Missing'), [0, 2, 4, 5]]	0.76668191
50	[(1, 4, 5), [0, 2, 3, 6]]	[('Crafts', 'Sales', 'Service'), [0, 2, 3, 6]]	0.766800844
51	[(1, 4, 6), [0, 2, 3, 5]]	[('Crafts', 'Sales', 'Missing'), [0, 2, 3, 5]]	0.76681737
52	[(1, 5, 6), [0, 2, 3, 4]]	[('Crafts', 'Service', 'Missing'), [0, 2, 3, 4]]	0.767053523
53	[(2, 3, 4), [0, 1, 5, 6]]	[('Labor', 'Professional', 'Sales'), [0, 1, 5, 6]]	0.767051763
54	[(2, 3, 5), [0, 1, 4, 6]]	[('Labor', 'Professional', 'Service'), [0, 1, 4, 6]]	0.766830484
55	[(2, 3, 6), [0, 1, 4, 5]]	[('Labor', 'Professional', 'Missing'), [0, 1, 4, 5]]	0.766809953
56	[(2, 4, 5), [0, 1, 3, 6]]	[('Labor', 'Sales', 'Service'), [0, 1, 3, 6]]	0.766678592
57	[(2, 4, 6), [0, 1, 3, 5]]	[('Labor', 'Sales', 'Missing'), [0, 1, 3, 5]]	0.766763805
58	[(2, 5, 6), [0, 1, 3, 4]]	[('Labor', 'Service', 'Missing'), [0, 1, 3, 4]]	0.767018873
59	[(3, 4, 5), [0, 1, 2, 6]]	[('Professional', 'Sales', 'Service'), [0, 1, 2, 6]]	0.767000627
60	[(3, 4, 6), [0, 1, 2, 5]]	[('Professional', 'Sales', 'Missing'), [0, 1, 2, 5]]	0.76700398
61	[(3, 5, 6), [0, 1, 2, 4]]	[('Professional', 'Service', 'Missing'), [0, 1, 2, 4]]	0.766823396
62	[(4, 5, 6), [0, 1, 2, 3]]	[('Sales', 'Service', 'Missing'), [0, 1, 2, 3]]	0.766694338

g) (5 points). What is the optimal split for the JobCategory predictor?

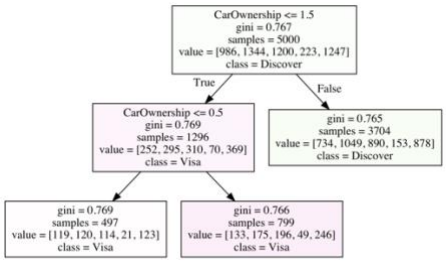
The optimal split for JobCategory will be: [('Labor', 'Sales', 'Service'), ['Agriculture', 'Crafts', 'Professional', 'Missing']] as it has the minimum Gini= 0.7666785924047139.

h) (5 points). Between the CarOwnership and the JobCategory predictors, which predictor will you choose for the second layer (i.e., depth 1) of your decision tree?

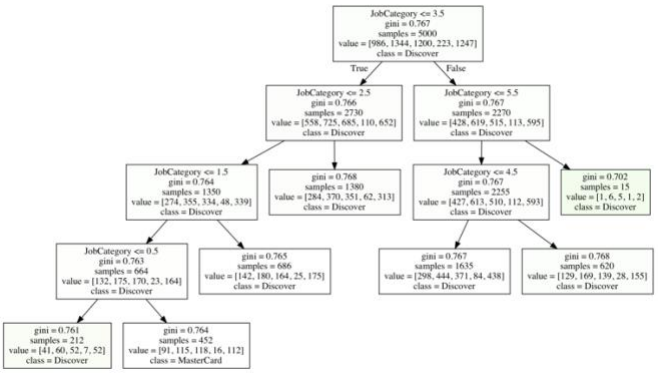
The predictor that should be used is the one with the lowest Gini value. Therefore, CarOwnership predictor should be used as it has its minimum Gini lower than the minimum from JobCategory.

It is not asked. However, I managed to plot both binary decision trees.

For CarOwnership this is the resulting tree:



On the other side in JobCategory this is the resulting tree:



Question 2 (50 points)

In 2014, Allstate provided the data on Kaggle.com for the Allstate Purchase Prediction Challenge which is open. The data contain transaction history for customers that ended up purchasing a policy. For each Customer ID, you are given their quote history and the coverage options they purchased.

The data is available on the Blackboard as Purchase_Likelihood.csv. It contains 665,249 observations on 97,009 unique Customer ID. We are going to use the MNLogit function to build a multinomial logistic model to predict purchase likelihood of coverage A using three predictors. The target variable is **A** which have these categories 0, 1, and 2. The nominal predictors are (categories are inside the parentheses):

1. **group_size**. How many people will be covered under the policy (1, 2, 3 or 4)?
2. **homeowner**. Whether the customer owns a home or not (0=no, 1=yes)
3. **married_couple**. Does the customer group contain a married couple (0=no, 1=yes)

Please build a multinomial logistic model using and answer the following questions.

a) (10 points) Suppose you start with a model with only the Intercept term (i.e., without any predictors). You are asked to mathematically calculate the maximum likelihood estimates of the predicted probabilities $\pi_{ij}, j = 1,2,3$ without calling the MNLogit function. Show all the necessary steps and the estimates for the $\pi_{ij}, j = 1,2,3$. (Hint: equate the first derivatives of the log-likelihood function to zeros for this Intercept-only model).

The first derivative of the log-likelihood function with respect to B_{j0} is:

$$\frac{\partial l}{\partial \beta_{j0}} = \sum_{i=1}^{m} x_{is}(n_{ij} - n_i \pi_{ij})$$

For β_{j0} the value of $x_{i0}$is equal to 1. We have only one subpopulation (m=1). If we make equal to zero the derivate, the values of π_{ij} are calculated as:

$$\pi_{ij} = \frac{n_{ij}}{n_i}$$

$$\pi_{i1} = \frac{143691}{665249} = 0.215995 \ (A = 0)$$

$$\pi_{i2} = \frac{143691}{665249} = 0.640462 \ (A = 1)$$

$$\pi_{i3} = \frac{143691}{665249} = 0.143541 \ (A = 2)$$

b) (10 points) Next, you are asked to mathematically calculate the maximum likelihood estimates of the Intercept terms $\beta_{j0}, j = 1, \dots, K$. The convention is to set the Intercept term to zero for the target category A = 0, i.e., $\beta_{10} = 0$. (Hint: use the mathematical formula of the logit of π_{ij} (i.e., $\log_e(\pi_{ij}/\pi_{i1})$ for this Intercept only model, then solve for the betas)?

We know that $\log_e\left(\frac{\pi_{ij}}{\pi_{iJ}}\right) = \boldsymbol{x}_i^T \boldsymbol{\beta}_j$ and for $\beta_{j0} = 0$ the values of x are 1. Therefore, the estimated of the Intercept terms $\beta_{j0}$are:

$$\beta_{j0} = \log_e\left(\frac{\pi_{ij}}{\pi_{iJ}}\right) = 0$$

$$\beta_{j0} = \log_e\left(\frac{\pi_{ij}}{\pi_{iJ}}\right) = 1.086931$$

$$\beta_{j0} = \log_e\left(\frac{\pi_{ij}}{\pi_{iJ}}\right) = -0.408633$$

c) (4 points) Now, you will use the MNLogit function to build the multinomial logistic model. What value of the target variable A is used by the MNLogit function as the reference category?
The statsmodels.api.MNLogit function conventionally takes the lexically first target category as the reference. In this case, A = 0 is the reference category.

d) (2 points) How many iterations are performed before convergence is achieved?
It takes 5 iterations to achieve the convergence with a tolerance of 1e-7 in the model.

```
Optimization terminated successfully.
         Current function value: 0.889797
         Iterations 5
```

e) (4 points) How many parameters (including the redundant ones) are in the model?
There are 7 parameters for each A characteristic. As the model takes characteristic A=0 as the reference there are 7 redundant parameters associated to this characteristic. On the other hand as it is shown in the following table, there are 7 parameters for characteristic A=1 (column 1) and another 7 for characteristic A=2 (column 2).

Index	0	1
const	0.408312734	-0.564585713
group_size_1	0.424290567	0.0285824484
group_size_2	0.437360521	0.0747731202
group_size_3	-0.0940892593	-0.105714307
group_size_4	-0.359273388	-0.562208177
homeowner	0.462188222	0.214712106
married_couple	0.116990053	0.0908341436

f) (5 points) When group_size = 2, homeowner = 1, and married_couple = 1, what are the predicted probabilities: Prob(A = 0), Prob(A = 1), and Prob(A = 2)?

- $\log_e$(Pr(A=1)/Pr(A=0)) = 0.408312734 + 0.437360521 + 0.462188222 + 0.116990053 = 1.42485153
  - Pr(A=1)/Pr(A=0) = 4.157
- $\log_e$(Pr(A=2)/Pr(A=0)) = -0.564585713 + 0.0747731202 + 0.214712106 + 0.0908341436 = -0.18426634
  - Pr(A=2)/Pr(A=0) = 0.8317
- Pr(A=0) + Pr(A=1) + Pr(A=2) = 1

Therefore:

- Pr(A=0) = 0.1669812
- Pr(A=1) = 0.6941406
- Pr(A=2) = 0.1388782

g) (10 points) What are the values of the predictors group_size, homeowner, and married_couple such that Prob(A = 0) will attain its maximum? What is the maximum Prob(A = 0) value?

The values for the predictors are:

- Group_size = 4
- Homeowner= 0
- Married_couple = 0

With these values the maximum probability Prob(A = 0) = 0.421171 is obtained.

h) (5 points) According to the logistic model, what is the odds ratio for group_size = 4 versus group_size = 1, and A = 1 versus A = 0? Mathematically, the odds ratio is (Prob(A=1)/Prob(A=0) | group_size = 4) / ((Prob(A=1)/Prob(A=0) | group_size = 1).

$L = \log_e \left(\frac{\Pr(A=1)/\Pr(A=0)|\ group\_size\ =4}{\Pr(A=1)/\Pr(A=0)|\ group\_size=1} \right) =$

$\log_e(\Pr(A = 1)/\Pr(A = 0)|\ group\_size\ = 4) - \log_e(\Pr(A = 1)/\Pr(A = 0)|\ group\_size\ = 1)$

$\log_e(\Pr(A = 1)/\Pr(A = 0)|\ group\_size\ = 4) = 0.408312734 - 0.359274 = 0.049039$

$\log_e(\Pr(A = 1)/\Pr(A = 0)|\ group\_size\ = 1) = 0.408312734 + 0.424290 = 0.832603$

L= 0.049039 - 0.832603= -0.783564

Odds ratio = e^{-L} = 0.4567751