Mariam Ashraf Mohamed

Sec 2 | BN 24

Problem 1.b)

Recommend another type of trees than ID3 that would build a less deep tree than ID3, assume that you are building a complete ID3 tree. Justify your choice.

- CART decision tree may build a less deep tree than id3 as id3 branches depends on the number of values found in the feature value like in the outlook feature it had 3 branches based on sunny, overcast and rainy while in cart dtree it splits the data if it higher or not than a specific value (numerical data) or if it joins a specific category or not so t always has 2 branches (left, right) so it has a smaller number of choices to make.
- **Rotation forest** – in which every decision tree is trained by first applying principal component analysis (PCA) on a random subset of the input features so it may have smaller number of nodes.

Problem 2)

GINI_IMPURITY (FEATURE):

Given a Pandas Series, it calculates the Gini Impurity according to this rule:

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2$$

- FEATURE: variable with which calculate Gini Impurity.
- Return: Gini impurity of the given feature

ENTROPY (FEATURE):

Given a Pandas Series, it calculates the entropy according to this rule:

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

- FEATURE: variable with which calculate entropy.
- Return: Entropy of the given feature

INFORMATION_GAIN (FEATURE, MASK, FUNCTION):

It returns the Information Gain of a variable given a loss function (gini_impurity or entropy) according to this rule:

$$InformationGain_{Classification} = E(d) - \sum \frac{|s|}{|d|} E(s)$$

- feature: target variable.
- mask: split choice.
- function: function to be used to calculate Information Gain in case of classification.

CATEGORICAL_OPTIONS(A):

- Creates all possible combinations from a Pandas Series.
- a: Pandas Series from where to get all possible combinations.

MAX_INFORMATION_GAIN_SPLIT (X, Y, FUNCTION):

Given a predictor & target variable

- x: predictor variable as Pandas Series.
- y: target variable as Pandas Series.
- function: function to be used to calculate the best split.
- Return: the best split, the error and the type of variable based on a selected cost function.

GET_BEST_SPLIT (Y, DATA, METHOD):

Given a data, select the best split

- y: name of the target variable
- data: data-frame where to find the best split.
- Method: a loss function with which it calculates information gain.
- Return: the variable, the value, the variable type and the information gain.

MAKE_SPLIT(VARIABLE, VALUE, DATA, IS_NUMERIC):

Given a data and a split conditions, do the split.

- variable: variable with which make the split.
- value: value of the variable to make the split.
- data: data to be splitted.
- is_numeric: boolean considering if the variable to be splitted is numeric or not.
- Return: The splitted data set.

MAKE_PREDICTION (DATA, TARGET_FACTOR):

Given the target variable, make a prediction.

- data: pandas series for target variable
- target_factor: boolean considering if the variable is a factor or not
- Return: Prediction

TRAIN_TREE (DATA, Y, TARGET_FACTOR, FUNCTION, MAX_DEPTH = NONE, MIN_SAMPLES_SPLIT = NONE, MIN_INFORMATION_GAIN = 1E-20, COUNTER=0, MAX_CATEGORIES = 20):

Trains a Decision Tree

- data: Data to be used to train the Decision Tree
- y: target variable column name
- target_factor: boolean to consider if target variable is factor or numeric.
- max_depth: maximum depth to stop splitting.
- min_samples_split: minimum number of observations to make a split.
- min_information_gain: minimum ig gain to consider a split to be valid.
- max_categories: maximum number of different values accepted for categorical values. High number of values will slow down learning process.
- Return: The learnt decision tree

CLASSIFING_NODES(OBSERVATION, DTREE):

Predict using our trained decision tree, Break the decision into several chunks, Check the type of decision that it is (numerical or categorical) and considering the type of variable that it is, check the decision boundary. If the decision is fulfilled, return the result, if it is not, then continue with the decision.

- observation: Data to be used to test the Decision Tree
- Dtree: trained decision tree.
- Return: the predicted label of the sample

PREDICT (TEST_FEATURES, DECISIONES_TREE):

Given the test data, it iterates over each sample and make a prediction based on Make prediction function.

- test_features: Data to be used to test the Decision Tree
- decisiones_tree: trained decision tree.
- Return: the predicted labels of the whole test data.

EVALUATE_PERFORMANCE (TEST_FEATURES_LABELS, PREDICTION_LABELS):

Evaluates the accuracy of the learnt decision tree by computing the confusion matrix.

- test_features_labels: the true test-data labels.
- prediction_labels: the predicted labels of the test data based on the learnt decision tree.
- Return: Accuracy of Decision tree

EVALUATED PERFORMANCE EITH OUR MODEL COMPARED TO SKLEARN DECISION TREE CLASSIFIER

BY CHOOSING MAX_DEPTH = 5 & MIN_SAMPLE_SPLIT = 20:

DTREE MODEL:

```
Accuray Based on Entropy 0.577
Accuray Based on Gini Impurity 0.577
Total run time:  53.861211442947386
```

SKLEARN:

```
Accuray Based on Entropy 0.7265714285714285
Accuray Based on Gini Impurity 0.7255714285714285
```

Problem 1-a

task 3 - Prob 1)

$$\text{Entropy} = -\sum_{i=1}^{N} P(x_i) \log_2 P(x_i)$$

$S = [8+, 6-]$

$\text{Entropy}(S) = -\frac{8}{14}\log_2\frac{8}{14} - \frac{6}{14}\log_2\frac{6}{14} = 0.985$

~~Early registeration~~ ER ⇒ Early registeration

$S(ER-1) = [4+, 2-]$

$\text{Entropy}[S(ER-1)] = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.918$

$S(ER-0) = [4+, 4-]$

$\text{Entropy}[S(ER-0)] = -\frac{4}{8}\log_2\frac{4}{8} - \frac{4}{8}\log_2\frac{4}{8} = 1$

$$\text{Gain}(S, ER) = \text{Entropy}(S) - \sum_{v \in (1,0)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S, ER) = 0.985 - \frac{6}{14}\text{Entropy}[S(ER-1)] - \frac{8}{14}\text{Entropy}[S(ER-0)]$$

$$= 0.985 - \frac{6}{14} * 0.918 - \frac{8}{14} * 1 = \boxed{0.02}$$

FH → finished homework

$S(FH-1) = [5+, 2-]$

$E[S(FH-1)] = -\frac{5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7} = 0.863$

$S(FH-0) = [3+, 4-]$

$E[S(FH-0)] = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.985$

Gain (S, fH) = 0.985 - $\frac{7}{14}$ * 0.863 - $\frac{7}{14}$ * 0.985
= **0.061**

S(senior-1) = [5+, 3-]
E[S(senior-1) = $-\frac{5}{8}\log_2\frac{5}{8} - \frac{3}{8}\log_2\frac{3}{8}$
= 0.954

S(senior-0) = [3+, 3-]
E[S(senior-0)] = $-\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}$ = 1

Gain(S, senior) = 0.985 - $\frac{8}{14}$ * 0.954 - $\frac{6}{14}$ * 1
= **0.011**

Likes Coffee ⇒ LC
S(LC-1) = [3+, 1-]
E[S(LC-1)] = $-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$ = 0.811

S(LC-0) = [5+, 5-]
E[S(LC-1)] = $-\frac{5}{10}\log_2\frac{5}{10} - \frac{5}{10}\log_2\frac{5}{10}$ = 1

Gain (S, LC) = 0.985 - $\frac{4}{14}$ * 0.811 - $\frac{10}{14}$ * 1
= **0.039**

Liked the last homework ⇒ LLH
S(LLH-1) = [5+, 4-]
E[S(LLH-1)] = $-\frac{5}{9}\log_2\frac{5}{9} - \frac{4}{9}\log_2\frac{4}{9}$ = 0.991

S(LLH-0) = [3+, 2-]
E[S(LLH-0)] = $-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$ = 0.971

Page / Date /

Gain(S, LLH) = 0.985 − $\frac{9}{14}$ * 0.991 − $\frac{5}{14}$ * 0.971
= 0.00114

∴ Root Node is finished homework.

Depth 1: Yes branch (1)

S(FH-1) = [5+, 2−]
E[S(FH-1)] = $\frac{-5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7}$ = 0.863

S(ER-1) = [3+, 0−]

E[S(ER-1)] = $-\frac{3}{3}\log_2\frac{3}{3} - 0$ = 0

S(ER-0) = [2+, 2−]

E[S(ER-0)] = $\frac{-2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$ = 1

Gain(S(FH-1), ER) = 0.863 − $\frac{3}{7}$ * 0 − $\frac{4}{7}$ * 1
= 0.292

S(senior-1) = [3+, 2−]
E[S(senior-1)] = $\frac{-3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$ = 0.971

S(senior-0) = [2+, 0−]
E[S(senior-0)] = $\frac{-2}{2}\log_2\frac{2}{2} - 0$ = 0

Gain(S(FH-1), senior) = 0.863 − $\frac{5}{7}$ * 0.971 − $\frac{2}{7}$ * 0 = 0.169

$S(LC-1) = [1+, 1-]$

$E[S(LC-1)] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

$S(LC-0) = [4+, 1-]$

$E[S(LC-0)] = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5} = 0.722$

$Gain(S(FH-1), LC) = 0.863 - \frac{2}{7}*1 - \frac{5}{7}*0.722$

$= \boxed{0.062}$

$S(LLH-1) = [3+, 2-]$

$E[S(LLH-1)] = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$

$S(LLH-0) = [2+, 0-]$

$E[S(LLH-0)] = -\frac{2}{2}\log_2\frac{2}{2} - 0 = 0$

$Gain(S(FH-1), LLH) = 0.863 - \frac{5}{7}*0.971 - \frac{2}{7}*0$

$= \boxed{0.169}$

∴ first decision Node in Yes branch is ER

No branch (0) :

$S(FH-0) = [3+, 4-]$

$E[S(FH-0)] = \frac{-3}{7}*\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = \boxed{0.985}$

$S(ER-1) = [1+, 2-]$

$E[S(ER-1)] = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$

Page / Date /

$S(ER-0) = [2+, 2-]$

$E[S(ER-0)] = \frac{-2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$

$Gain(S(FH-0), ER) = 0.985 - \frac{3}{7} * 0.918 - \frac{4}{7} * 1$

$= \boxed{0.02}$

$S(senior-1) = [24, 1-]$

$E[S(senior-1)] = \frac{-2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$

$S(senior-0) = [1+, 3-]$

$E[S(senior-0)] = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.811$

$Gain(S(FH-0), senior) = 0.985 - \frac{3}{7} * 0.918 - \frac{4}{7} * 0.811$

$= \boxed{0.128}$

$S(LC-1) = [2+, 0-]$

$E[S(LC-1)] = \frac{-2}{2}\log_2\frac{2}{2} - 0 = 0$

$S(LC-0) = [1+, 4-]$

$E[S(LC-0)] = \frac{-1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.722$

$Gain(S(FH-0), LC) = 0.985 - \frac{2}{7} * 0 - \frac{5}{7} * 0.722$

$= \boxed{0.469}$

$S(LLH-1) = [2+, 2-]$

$E[S(LLH-1)] = \frac{-2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$

$S(LLH-0) = [1+, 2-]$

$E[S(LLH-0)] = \frac{-1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$

Gain (S(FH-0), LLH) = 0.985 - 4/7 * 1 - 3/7 * 0.918
= 0.02

the first decision node in No branch is LC

Depth 2 : ER-Yes branched from FH-Yes

S(ER-1) = [3+, 0-]
E[S(ER-1)] = $-\frac{3}{3}\log_2\frac{3}{3}$ - 0 = 0

So this branch gives us Class A = 1

ER - No branched from FH-Yes

S(ER-0) = [2+, 2-]
E[S(ER-0)] = $-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$ = 1

S(Senior-1) = [1+, 2-]
E[S(Senior-1)] = $-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}$ = 0.918

S(Senior-0) = [1+, 0-]
E[S(Senior-0)] = $-\frac{1}{1}\log_2 1$ - 0 = 0

Gain (S(ER-0), Senior) = 1 - 3/4 * 0.918 - 1/4 * 0
= 0.312

S(LC-1) = [1+, 1-]
E[S(LC-1)] = $-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$ = 1

Page / Date /

$S(LC-0) = [1+, 1-]$

$E[S(LC-0)] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

$Gain(S(ER-0), LC) = 1 - \frac{2}{4} * 1 - \frac{2}{4} * 1 = \boxed{0}$

$S(LLH-1) = [1+, 2-]$

$E[S(LLH-1)] = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$

$S(LLH-0) = [1+, 0-]$

$E[S(LLH-0)] = -1\log_2 1 - 0 = 0$

$Gain(S(ER-0), LLH) = 1 - \frac{3}{4} * 0.918 - \frac{1}{4} * 0$

$= \boxed{0.312}$

So Gain of both Senior & LLH is equal so we take anyone of them → we choose the 1st feature

⇒ Senior

---

depth 2: LC-Yes branched from fH-No

$S(LC-1) = [2+, 0-]$

$E[S(LC-1)] = -\frac{2}{2}\log_2\frac{2}{2} - 0 = 0$

So this branch gives Class A = 1

LC - No branched from fH - No

$S(LC-0) = [1+, 4-]$

$E[S(LC-0)] = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = \boxed{0.722}$

$S(ER\_1) = [0+, 2-]$

$E[S(ER\_1)] = 0 - \frac{2}{2}\log_2\frac{2}{2} = 0$

$S(ER\_0) = [1+, 2-]$

$E[S(ER\_0)] = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$

$Gain(S(LC\_0), ER) = 0.722 - \frac{2}{5}\times 0 - \frac{3}{5}\times 0.918$

$= \boxed{0.171}$

$S(Senior\_1) = [1+, 1-]$

$E[S(senior\_1)] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

$S(senior\_0) = [0+, 3-]$

$E[S(senior\_0)] = 0 - \frac{3}{3}\log_2\frac{3}{3} = 0$

$Gain(S(LC\_0), senior) = 0.722 - \frac{2}{5}\times 1 - \frac{3}{5}\times 0$

$= \boxed{0.322}$

$S(LLH\_1) = [1+, 2-]$

$E[S(LLH\_1)] = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$

$S(LLH\_0) = [0+, 2-]$

$E[S(LLH\_0)] = 0 - \frac{2}{2}\log_2\frac{2}{2} = 0$

$Gain(S(LC\_0), LLH) = 0.722 - \frac{3}{5}\times 0.918 - \frac{2}{5}\times 0$

$= \boxed{0.171}$

So the second decision node in IFH-NO branch is senior

Depth 3: FH-Yes → ER-No → Senior-Yes

S(Senior-1) = [1+, 2-]
E[S(Senior-1)] = $-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}$ = 0.918

S(LC-1) = [0+, 1-]
E[S(LC-1)] = 0 - $1\log_2 1$ = 0

S(LC-0) = [1+, 1-]
E[S(LC-0)] = $-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$ = 1

G(S(Senior-1), LC) = 0.918 - $\frac{1}{3}$*0 - $\frac{2}{3}$*1
= 0.251

S(LLH-1) = [0+, 2-]
E[S(LLH-1)] = 0 - $\frac{2}{2}\log_2\frac{2}{2}$ = 0

S(LLH-0) = [1+, 0-]
E[S(LLH-0)] = $-1\log_2 1$ - 0 = 0

G(S(Senior-1), LLH) = 0.918 - $\frac{2}{3}$*0 - $\frac{1}{3}$*0
= 0.918

So the third node in this branch is LLH

FH-Yes → ER-No → Senior-No

S(Senior-0) = [1+, 0-]
E[S(Senior-0)] = $-1\log_2 1$ - 0 = 0

So this node gives Class A - 1

---

depth 3 fH-No → LC-No → Senior-Yes

S(senior-1) : [1+, 1-]
E[S(senior-1)] = $-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

S(ER-1) = [0+, 0-]
E[S(ER-1)] = 0

S(ER-0) = [1+, 1-]
E[S(ER-0)] = $-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

Gain(S(senior-1), ER) = $1 - 0 - \frac{2}{2} * 1 = 0$

S(LLH-1) = [1+, 0-]
E[S(LLH-1)] = $-1\log_2 1 - 0 = 0$

S(LLH-0) = [0+, 1-]
E[S(LLH-0)] = $0 - 1\log_2 1 = 0$

Gain(S(senior-1), LLH) = $1 - 0 - 0 = 1$

So the third node in this branch is LLH

---

fH-No → LC-No → Senior-No

S(senior-0) = [0+, 3-]
E[S(senior-0)] = $-\frac{3}{3}\log_2\frac{3}{3} - 0 = 0$
So this branch gives Class B = 0