Homework 2

Due : April 4^{th} (11:59 pm)

Submitted by: Jyotsna Rajaraman

Q1 AIMA3 → Ex 18.13, Ex 18.14

18.13) P.T :

Num (rules in decision list) $\leq$ leaves (decision tree)

We know that in decision lists, each rule has a binary outcome, YES and NO. If yes → there will be a prediction, if no, we will proceed to the next rule.

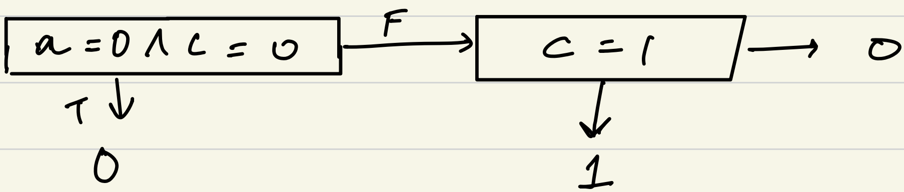
Let us make the assumption that, in worst case, each of the 'n' leaves in the decision tree corresponds to an unique outcome. This means that we would need the same number of rules to create n unique outcomes.

If not, we may actually be able to combine the nodes of a decision tree to create a rule that may reduce the overall number of rules together.

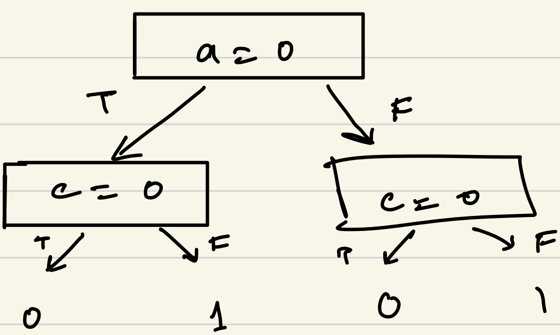
$\therefore$ Num(rules) in list < Num(leaves) in tree

a	b	c	outcome.
0	0	0	0
0	0	1	1
1	0	0	0
1	0	1	1

Decision List



Decision Tree



18.4) Expressiveness of decision lists

(a)

Each rule in a decision list is a conjunction of Boolean variable and thus we can essentially create a expression for any Boolean function as:

$$A \wedge B \xrightarrow{\text{False}} \bar{A} \wedge B \xrightarrow{\text{False}} A \wedge \bar{B} \xrightarrow{\text{False}} \bar{A} \wedge \bar{B} \quad 1$$

$A \wedge B \downarrow$ True o/p; $\bar{A} \wedge B \downarrow$ True o/p; $A \wedge \bar{B} \downarrow$ True o/p; $\bar{A} \wedge \bar{B} \downarrow$ True o/p

For 2 variables

In this way we can actually write an expression for each out come and represent it with decision lists.

(b)

∵ The decision tree has depth $= k$. At most, each path has k nodes. ∴ Taking the longest path, we can convert in to a decision list. Hence, we can use at most k literals to build a decision list for the function

Q2) Generalization Error:

* T → training set ; V → validation set
  (both i.i.d)

*

Max Depth	Err Rate
4	0.45
5	0.35
6	0.40

* Gen Error : E(misclassification err over all alg.)

STATEMENT #1:
An unbiased estimate of the generalized error rate for a decision tree w/ max depth 5 = 0.35
Since T and V are i.i.d samples & are random it is true.
We can explain this in detail by considering that V (validation) set of data is independent of the training data. which means that

the generalized error is an unbiased estimate

Statement 2:
An unbiased estimate of a tuned decision tree is 0.85

False, because the tuned decision tree has a validation set that is dependent. This means that the tree is built based on the training & validation. Therefore cannot be generalized over the universe U.
∴ It is NOT an unbiased estimate.

Statement : 3

The unbiased estimate of the random
decisim tree is 0·35

0·35 is the unbiased estimate specifically
for the random case chosen since
T and V are not dependent.
However, in the case of a random
decision tree - it may be the
case that the max depth could change
and ∴ The generilized error would
also vary.

Q3) CROSS VALIDATION:

Task $\rightarrow$ Binary Classification

$U \rightarrow$ Universe $\rightarrow$ p = class 1, $1-p$ = class 0

$0.5 < p \leq 1$

$\therefore$ Class 1 is the majority class.

$A \rightarrow$ classification algorithm.

$\hookrightarrow$ always predicts the majority in training set or if equal $\rightarrow$ picks random.

(R. sample)

$D \rightarrow$ dataset of m examples. (i.i.d) $\in U$

$C \rightarrow$ accuracy of A on D w k fold CV

$\hookrightarrow$ (R.V)

a) $E[C] = ?$

$X_j \rightarrow$ class of example j

$1 \leq j \leq m \Rightarrow D$ has m samples.

training & test for i^{th} fold $= T_i, S_i \rightarrow 1 \leq i \leq k$.

$\Rightarrow k$ folds.

$C_i \rightarrow$ accuracy of each fold.

For fold i:

$E[\lambda \mid Y = y] = \sum_{x} x f_{X|Y}(x|y)$

$E[C_i \mid T_{i(maj)} = 1] \rightarrow \quad C = \text{accuracy} = \dfrac{\text{correct}}{\text{total}}.$

If $T_{imaj} = 1 \rightarrow$

$$C_i \mid T_{imaj} = 1 \quad = \begin{cases} \dfrac{p}{m} & \text{w prob } 1 \\ \dfrac{1-p}{m} & \text{w prob } 0 \end{cases}$$

$$C_i \mid T_{imaj} = 0 \quad : \begin{cases} \dfrac{1-p}{m} & \text{w prob } 1 \\ \dfrac{p}{m} & \text{w prob } 0 \end{cases}$$

Total no. of correctly classified instances in each fold = no. of instances of majority class in that fold.

$\therefore$ Tot no. of correctly classified instances $\left.\right\} = \left[\left(p \times \frac{m}{k}\right) + (1-p)\ \frac{m}{k}\right] k$

$\frac{m}{k} \rightarrow$ no. of instances in each fold.

$$E(c) = \frac{\text{Correct}}{\text{Total}}$$

$$= \frac{k\left[\left(p \times \frac{m}{k}\right) + (1-p) \times \frac{m}{k}\right)\right]}{m}$$

(b), (c) $\rightarrow$ Python.

(d) if no. of samples, $m \to \infty$

$$C = \frac{\text{correct}}{\text{total}}$$

$\because$ class 1 is $= p$ = majority

Prediction : class 1

$\therefore$ correct $= p$
total $= m$

Weak Law: Sample av converges towards expected value.

$$\therefore \lim_{m \to \infty} Pr(C) = p$$

(e) $m \neq 0$, $p \neq 0$ and k.

such that $C = 0$

Suppose there are:

$m = 10$

$p = 6 \Rightarrow$ Class 1

$1 - p = 4 \Rightarrow$ Class 0

Say training set = all class 1

test set = all class 0

$\therefore$ accuracy will be $\underline{\underline{0}}$

(Q4)

i/p → integer ; o/p ⇒ Boolean

$$y(x) = (a \le x_1 \le b) \text{ and } (c \le x_2 \le d)$$

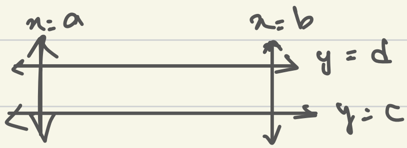
ⓐ S.T the no. of hyp is

$$= \left(\frac{n(n+1)}{2}\right)^2$$

No. of hypothesis = no. of distinct rect restricted to 1 to n int position.

We know → $a, b, c, d \in \{1 \dots n\}$.

if $x_1 \in [a, b]$ & $x_2 \in [c, d]$



} Bounds for x_1 & x_2

Number of rectangles:

$$\underbrace{\text{choices}[A] + \text{choices}[B]}_{\text{can't be same}} + \underbrace{\text{choices}[C] + \text{choices}[D]}_{\text{can't be same}}$$

$\therefore$

We pick two lines for a, b & 2 for c, d.
from $\{0, 1, 2 \ldots, n\}$.

$$\therefore {}^{n+1}C_2 \times {}^{n+1}C_2 = \frac{n(n+1)}{2} \times \frac{n(n+1)}{2}$$

$$= \left[\frac{n(n+1)}{2}\right]^2$$

Proved.

(b) How many training samples for 0 training error to have $E_{gen} \leq 10\%$ with prob 95% → prob → $1 - \underbrace{0.05}_{\delta}$.

$$N \geq \frac{1}{\varepsilon} \times \left[\ln|H| + \ln 1/\delta\right] \geq \frac{1}{\left(\frac{10}{100}\right)} \times \ln|H| + \ln \frac{1}{0.05}$$

$$N \geq \frac{100}{10}\left[\ln|H| + \ln 20\right]$$

$$N \geq 10\left[\left(\frac{n(n+1)}{2}\right)^2 + 2.99\right]$$

Q5. Python code has been submitted

Q6. Discussion:

* I used 10,000 rows of data

* For Imputer
  > for classification / discrete → 'most freq'
    ∵ we don't want to end up with predictions that do not exist.
  > for continuous values → 'median'
    ∵ to avoid creating a bias within the mean/expectation of data

* It may be useful to check for which rows have missing values — especially for smaller data sets to eliminate rows with little to no info.
  For instance we may have data samples that do not have any cols except for 1 - 2 → ∴ introducing error

However for larger datasets it may be a trivial result requiring additional computation & analysis.

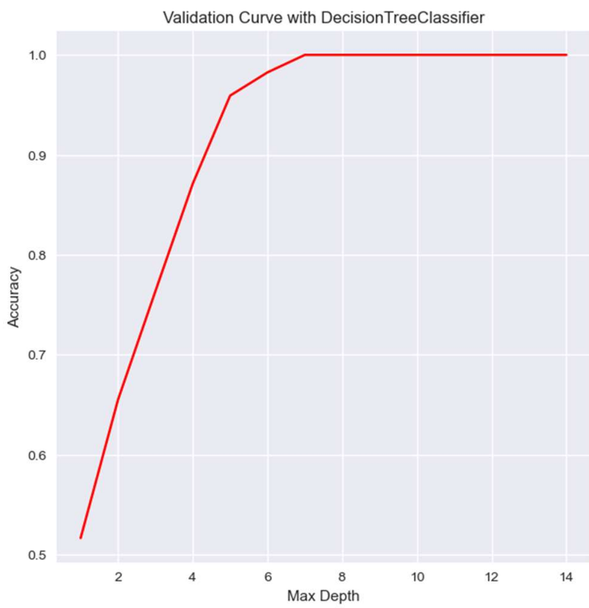
(This has been reflected in the data test)

Learning Curves for DecisionTree Vs LogisticReg.

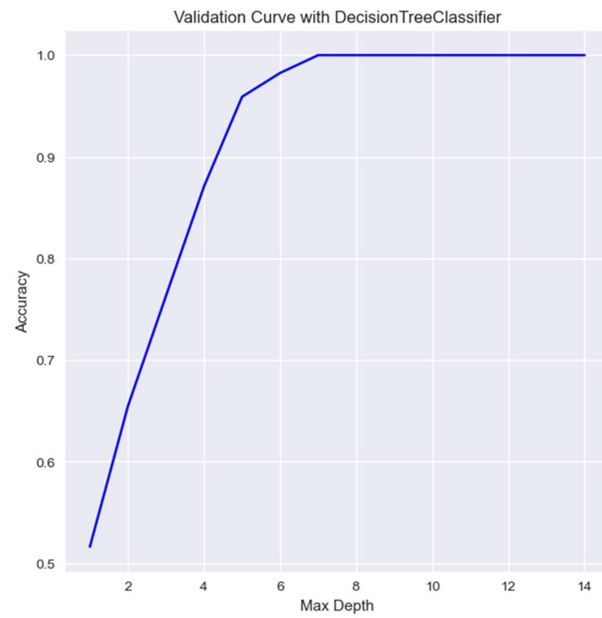
For decision tree, the error goes slightly ↑ as training set increases but for logistic regression ↑ in training set size decreases the error.

For Decision Tree Classifier Model:

Validation Curves:



Validation Curve - Training Data



Validation Curve - Testing Data

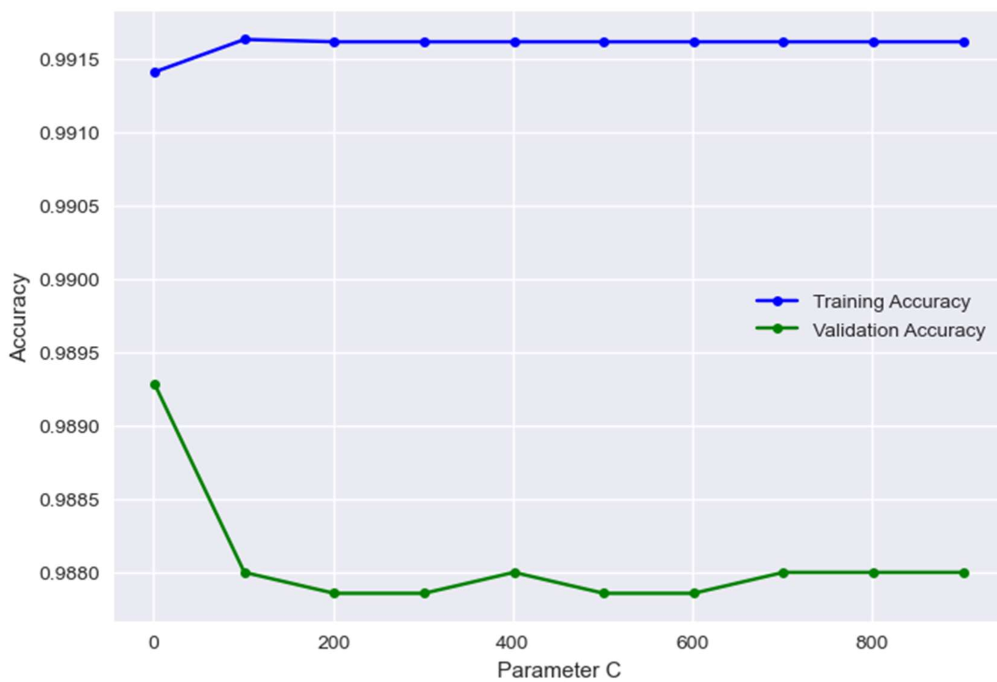
Learning Curves:

Learning Curve : Depth = 5

Learning Curve : Depth = 7

For Logistic Regression Model:

Validation Curve:



Learning Curve: