Team G20

Amal Sony	asony
Mohd Sharique Khan	mkhan8
Siddu Madhure Jayanna	smadhur

1)

1. **Constructing decision tree on the given data by using Information Gain (IG)**
   Assuming class T→TRUE , F→FALSE

➔ Entropy of Class: H(C)

Class	Frequency
True	7
False	9

$$H(C) = -\frac{9}{16}\log_2\frac{9}{16} - \frac{7}{16}\log_2\frac{7}{16} = 0.9886$$

➔ Checking for best split for continuous attribute V1:

When split happens at V1=7

$V1 \leq 7$

Class	Frequency
True	0
False	1

$$H(C|V1 \leq 7) = 0$$

$V1 > 7$

Class	Frequency
True	7
False	8

$$H(C|V1 > 7) = 0.99$$

$$H(C|V1) = 0 * \frac{1}{16} + \frac{15}{16} * .99 = 0.928$$

When split happens at V1=10

$V1 \leq 10$

Class	Frequency
True	0
False	2

$$H(C|V1 \leq 10) = 0$$

$V1 > 10$

Class	Frequency
True	7
False	7

$$H(C|V1 > 10) = 1$$

$$H(C|V1) = 0.875$$

When split happens at V1=11

$V1 \leq 11$

Class	Frequency
True	1
False	2

$$H(C|V1 \leq 11) = 0.91$$

$V1 > 11$

Class	Frequency
True	6
False	7

$$H(C|V1 > 11) = 0.99$$
$$H(C|V1) = 0.975$$

When split happens at V1=13

$V1 \leq 13$

Class	Frequency
True	1
False	3

$$H(C|V1 \leq 13) = 0.811$$

$V1 > 13$

Class	Frequency
True	6
False	6

$$H(C|V1 > 13) = 1$$
$$H(C|V) = 0.9527$$

When split happens at V1=15

$V1 \leq 15$

Class	Frequency
True	2
False	3

$$H(C|V1 \leq 15) = 0.970$$

$V1 > 15$

Class	Frequency
True	5
False	6

$$H(C|V1 > 15) = .994$$
$$H(C|V) = 0.9865$$

When split happens at V1=18

$V1 \leq 18$

Class	Frequency
True	2
False	4

$$H(C|V1 \leq 18) = 0.91$$

$V1 > 18$

Class	Frequency
True	5
False	5

$$H(C|V1 > 18) = 1$$
$$H(C|V) = 0.96625$$

When split happens at V1=20

$V1 \leq 20$

Class	Frequency
True	2
False	5

$$H(C|V1 \leq 20) = 0.86$$

$V1 > 20$

Class	Frequency
True	5
False	4

$$H(C|V1 > 20) = .991$$
$$H(C|V) = 0.933$$

When split happens at V1=22

$V1 \leq 22$

Class	Frequency
True	2
False	6

$$H(C|V1 \leq 22) = 0.811$$

$V1 > 22$

Class	Frequency
True	5
False	3

$$H(C|V1 > 22) = .954$$
$$H(C|V) = 0.8825$$

When split happens at V1=27

$V1 \leq 27$

Class	Frequency
True	3
False	6

$$H(C|V1 \leq 27) = 0.91$$

$V1 > 27$

Class	Frequency
True	4
False	3

$$H(C|V1 > 27) = .985$$
$$H(C|V) = 0.9428$$

When split happens at V1=30

$V1 \leq 30$

Class	Frequency
True	4
False	6

$$H(C|V1 \leq 30) = 0.970$$

$V1 > 30$

Class	Frequency

True	3
False	3

$$H(C|V1 > 30) = 1$$
$$H(C|V) = 0.98125$$

When split happens at V1=32

$V1 \leq 32$

Class	Frequency
True	5
False	6

$$H(C|V1 \leq 32) = 0.99$$

$V1 > 32$

Class	Frequency
True	2
False	3

$$H(C|V1 > 32) = 0.97$$
$$H(C|V) = 0.983$$

When split happens at V1=35

$V1 \leq 35$

Class	Frequency
True	6
False	6

$$H(C|V1 \leq 35) = 1$$

$V1 > 35$

Class	Frequency
True	1
False	3

$$H(C|V1 > 35) = 0.811$$
$$H(C|V) = 0.952$$

When split happens at V1=37

$V1 \leq 37$

Class	Frequency
True	6
False	7

$$H(C|V1 \leq 37) = 0.995$$

$V1 > 37$

Class	Frequency
True	1
False	2

$$H(C|V1 > 37) = 0.91$$
$$H(C|V) = 0.979$$

When split happens at V1=40

$V1 \leq 40$

Class	Frequency
True	7
False	7

$$H(C|V1 \leq 40) = 1$$

$V1 > 40$

Class	Frequency
True	0
False	2

$$H(C|V1 > 40) = 0$$
$$H(C|V) = 0.875$$

When split happens at V1=43

$V1 \leq 43$

Class	Frequency
True	7
False	8

$$H(C|V1 \leq 43) = 0.996$$

$V1 > 43$

Class	Frequency
True	0
False	1

$$H(C|V1 > 43) = 0$$
$$H(C|V) = 0.993$$

Information gain for the splits:

Split	IG
V1=7	0.9886-0.928 = 0.060
V1=10	0.1136
V1=11	0.0136
V1=13	0.035
V1=15	0.002
V1=18	0.02
V1=20	0.05
V1=22	0.106
V1=27	0.04
V1=30	0.007
V1=32	0.005
V1=35	0.03
V1=37	0.009
V1=40	0.113
V1=43	0.004

The maximum IG occurs at Split V1=10 and V1=40. Taking the first into consideration and modifying splitting the categorical attribute V1 as $V1 \leq 10$ and $V1 > 10$

Checking the best attribute to split the decision tree on below data:

V1	V2	V3	V4	V5	Class
<=10	BLUE	LONG	HOT	HIGH	F
<=10	WHITE	SHORT	COOL	HIGH	F

>10	BLUE	SHORT	HOT	HIGH	T
>10	WHITE	LONG	HOT	HIGH	F
>10	BLUE	SHORT	COOL	HIGH	T
>10	WHITE	SHORT	HOT	HIGH	F
>10	BLUE	LONG	COOL	HIGH	F
>10	WHITE	LONG	COOL	HIGH	F
>10	WHITE	LONG	COOL	LOW	T
>10	BLUE	SHORT	COOL	LOW	T
>10	WHITE	SHORT	COOL	LOW	T
>10	BLUE	SHORT	HOT	LOW	T
>10	WHITE	SHORT	HOT	LOW	F
>10	BLUE	LONG	COOL	LOW	T
>10	WHITE	LONG	HOT	LOW	F
>10	BLUE	LONG	HOT	LOW	F

Checking information gain for V5:

V5 = HIGH

Class	**Frequency**
True	2
False	6

$$H(C|V5 = HIGH) = 0.811$$

V5=LOW

Class	**Frequency**
True	5
False	3

$$H(C|V5 = LOW) = 0.954$$

$$H(C|V5) = 0.8825$$

Checking information gain for V4:

V4 = COOL

Class	**Frequency**
True	5
False	3

$$H(C|V4 = COOL) = 0.954$$

V4 = HOT

Class	**Frequency**
True	2
False	6

$$H(C|V4 = HOT) = 0.811$$

$$H(C|V4) = 0.8825$$

Checking information gain for V3:

V3=LONG

Class	Frequency
True	2
False	6

$$H(C|V3 = LONG) = 0.811$$

V3=SHORT

Class	Frequency
True	5
False	3

$$H(C|V3 = SHORT) = 0.954$$
$$H(C|V3) = 0.8825$$

Checking information gain for V2:

V2=BLUE

Class	Frequency
True	5
False	3

$$H(C|V2 = BLUE) = 0.954$$

V2=WHITE

Class	Frequency
True	2
False	6

$$H(C|V2 = WHITE) = 0.811$$
$$H(C|V2) = 0.8825$$

Checking information gain for V1:

$V1 \leq 7$

Class	Frequency
True	0
False	1

$$H(C|V1 \leq 7) = 0$$

$V1 > 7$

Class	Frequency
True	7
False	8

$$H(C|V1 > 7) = 0.99$$
$$H(C|V1) = 0 * \frac{1}{16} + \frac{15}{16} * .99 = 0.928$$

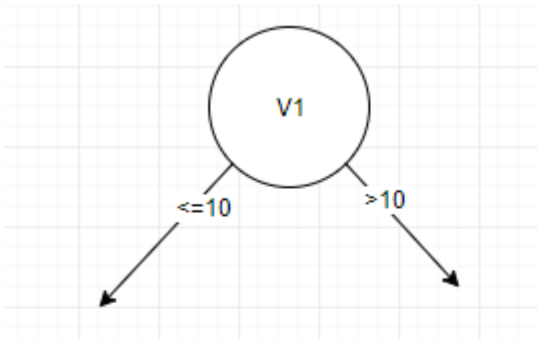
Information Gains Table:

Attribute	IG
V1	0.9886-0.875=0.116
V2	0.9886-0.8825=0.1061
V3	0.9886-0.8825=0.1061
V4	0.9886-0.8825=0.1061

V5	0.9886-0.8825=0.1061

The best attribute to split on is V1. Using the attribute V1 to split while building the decision tree:

Tree:



Checking for split if V1<=10:

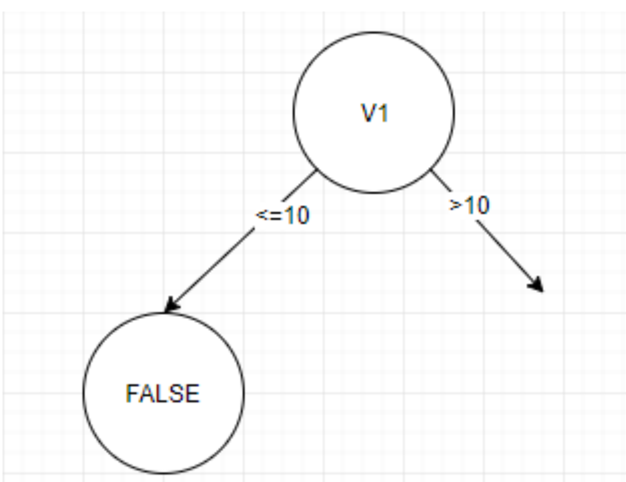
➔ Entropy of Class: H(C)

Class	Frequency
True	0
False	2

$$H(C) = 0$$

Therefor we don't need to split if V1<=10. It will be a leaf node.

Tree:



Checking for next split if V1>10:

➔ Entropy of Class: H(C)

Class	Frequency
True	7
False	7

$$H(C) = 1$$

Checking information gain for V5:

V5 = HIGH

Class	Frequency
True	2
False	4

$$H(C|V5 = HIGH) = 0.918$$

V5=LOW

Class	Frequency
True	5
False	3

$$H(C|V5 = LOW) = 0.954$$
$$H(C|V5) = 0.938$$

Checking information gain for V4:

V4 = COOL

Class	Frequency
True	5
False	2

$$H(C|V4 = COOL) = 0.863$$

V4 = HOT

Class	Frequency
True	2
False	5

$$H(C|V4 = HOT) = 0.863$$
$$H(C|V4) = 0.863$$

Checking information gain for V3:

V3=LONG

Class	Frequency
True	5
False	2

$$H(C|V3 = LONG) = 0.863$$

V3=SHORT

Class	Frequency
True	2
False	5

$$H(C|V3 = SHORT) = 0.863$$
$$H(C|V3) = 0.863$$

Checking information gain for V2:

V2=BLUE

Class	Frequency
True	5
False	2

$$H(C|V2 = BLUE) = 0.863$$

V2=WHITE

Class	Frequency
True	2
False	5

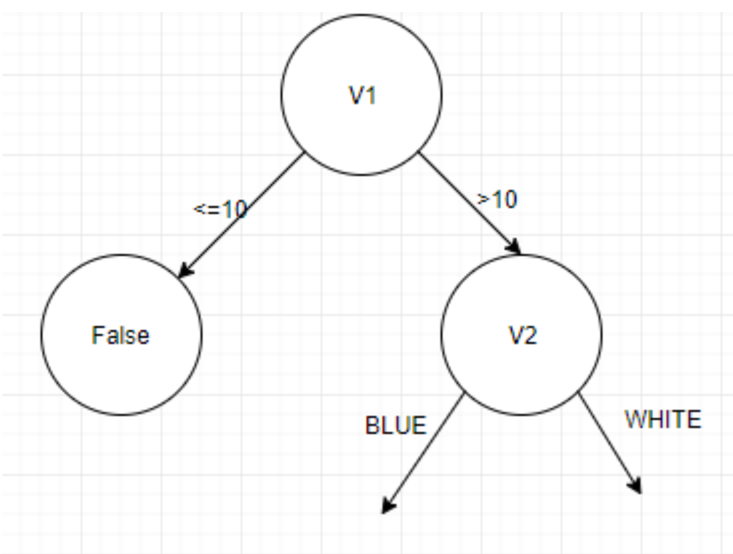
$$H(C|V2 = WHITE) = 0.863$$
$$H(C|V2) = 0.863$$

Information Gains Table:

Attribute	IG
V2	0.9886-0.863=0.131
V3	0.9886-0.863=0.131
V4	0.9886-0.863=0.131
V5	0.9886-0.938=0.06

The next best attribute to split on are V2, V3 and V4. Using the attribute V2 to split while building the decision tree:

Tree:



Checking for next split if V1>10 and V2 = BLUE:

➔ Entropy of Class: H(C)

Class	Frequency
True	5

False	2

$$H(C) = 0.863$$

Checking information gain for V5:

V5 = HIGH

Class	Frequency
True	2
False	1

$$H(C|V5 = HIGH) = 0.918$$

V5=LOW

Class	Frequency
True	3
False	1

$$H(C|V5 = LOW) = 0.811$$
$$H(C|V5) = 0.856$$

Checking information gain for V4:

V4 = COOL

Class	Frequency
True	3
False	1

$$H(C|V4 = COOL) = 0.811$$

V4 = HOT

Class	Frequency
True	2
False	1

$$H(C|V4 = HOT) = 0.918$$
$$H(C|V4) = 0.856$$

Checking information gain for V3:

V3=LONG

Class	Frequency
True	1
False	2

$$H(C|V3 = LONG) = 0.918$$

V3=SHORT

Class	Frequency
True	4
False	0

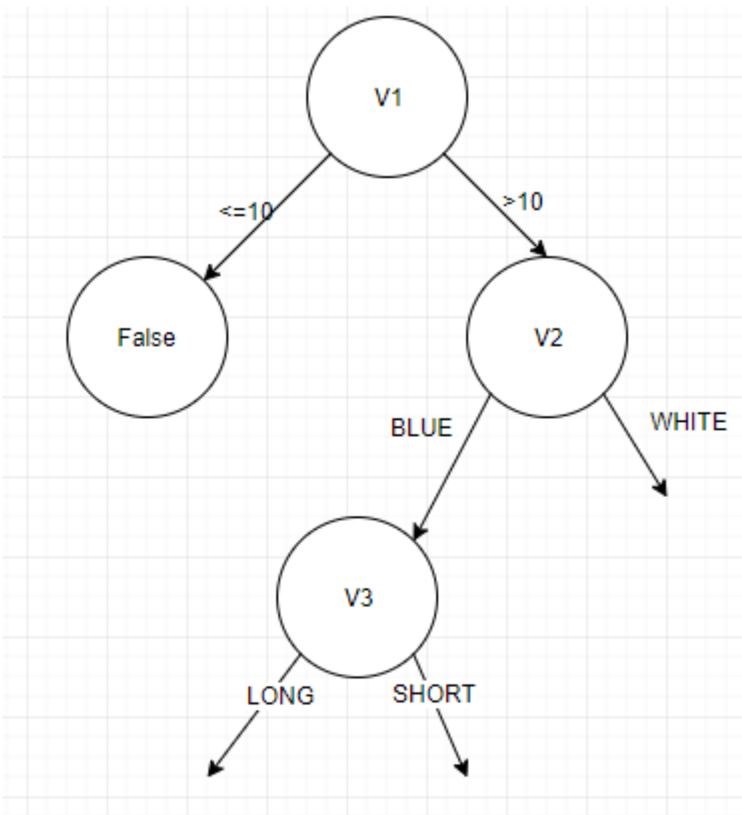
$$H(C|V3 = SHORT) = 0$$
$$H(C|V3) = 0.393$$

Information Gains Table:

Attribute	IG
V3	0.863-0.393=0.47
V4	0.863-0.856=0.006
V5	0.863-0.856=0.006

The next best attribute to split on are V3 based on IG. Using the attribute V3 to split while building the decision tree:

Tree:



Checking for split if V1>10 and V2=BLUE and V3 = SHORT:

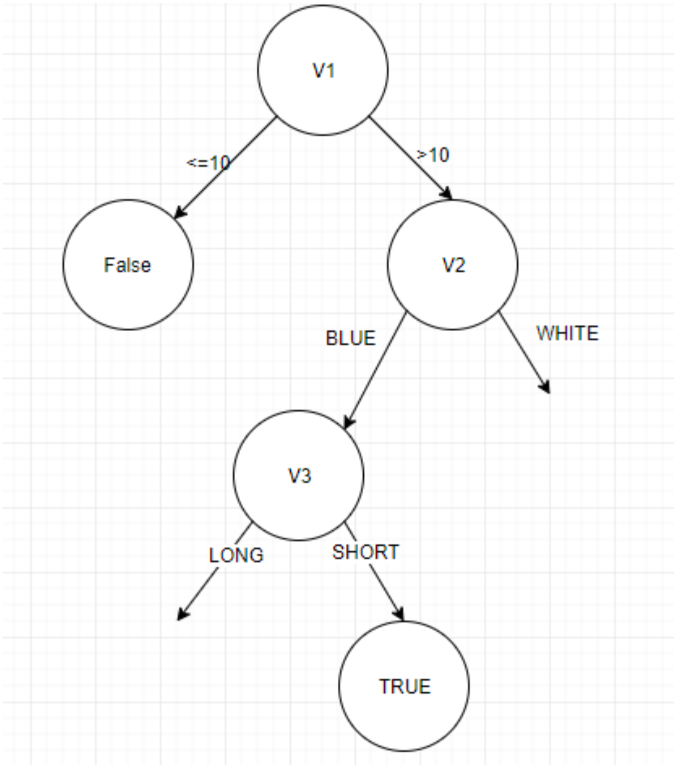
➔ Entropy of Class: H(C)

Class	Frequency
True	4
False	0

$$H(C) = 0$$

Therefor we don’t need to split if V1>10 and V2=BLUE and V3 = SHORT. It will be a leaf node.

Tree:



Checking for next split if V1>10 and V2 = BLUE and V3 = LONG:

➔ Entropy of Class: H(C)

Class	**Frequency**
True	1
False	2

$$H(C) = 0.918$$

Checking information gain for V5:

V5 = HIGH

Class	**Frequency**
True	0
False	1

$$H(C|V5 = HIGH) = 0$$

V5=LOW

Class	**Frequency**
True	1
False	1

$$H(C|V5 = LOW) = 1$$
$$H(C|V5) = 0.66$$

Checking information gain for V4:

V4 = COOL

Class	Frequency
True	1
False	1

$$H(C|V4 = COOL) = 1$$

V4 = HOT

Class	Frequency
True	0
False	1

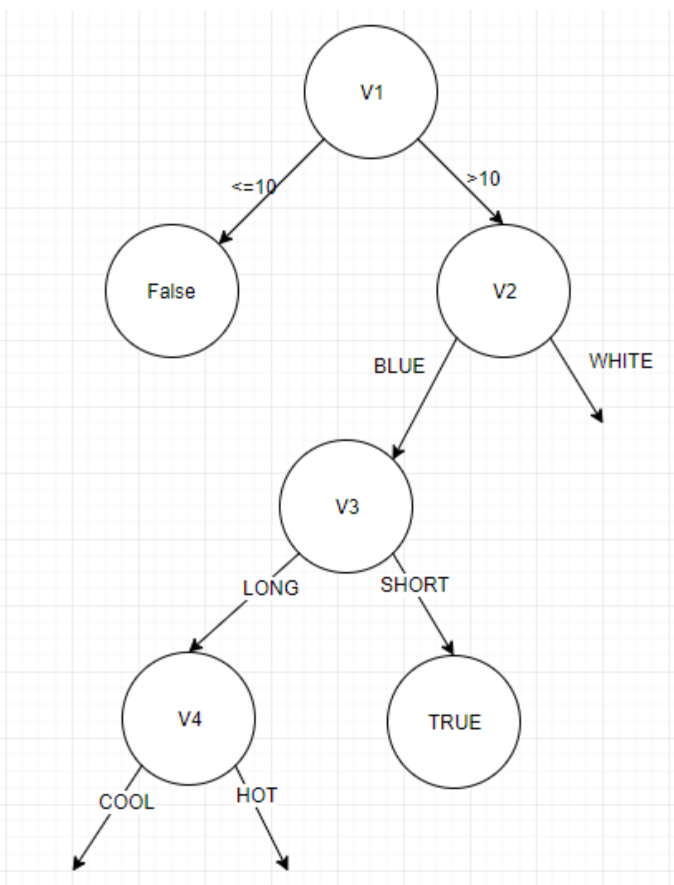
$$H(C|V4 = HOT) = 0$$
$$H(C|V4) = 0.66$$

Information Gains Table:

Attribute	IG
V4	0.918-0.66=0.258
V5	0.918-0.66=0.258

The next best attribute to split can be both V4 and V5 based on IG. Using the attribute V4 to split while building the decision tree:

Tree:



Checking for split if V1>10 and V2=BLUE and V3 = LONG and V4 = HOT:

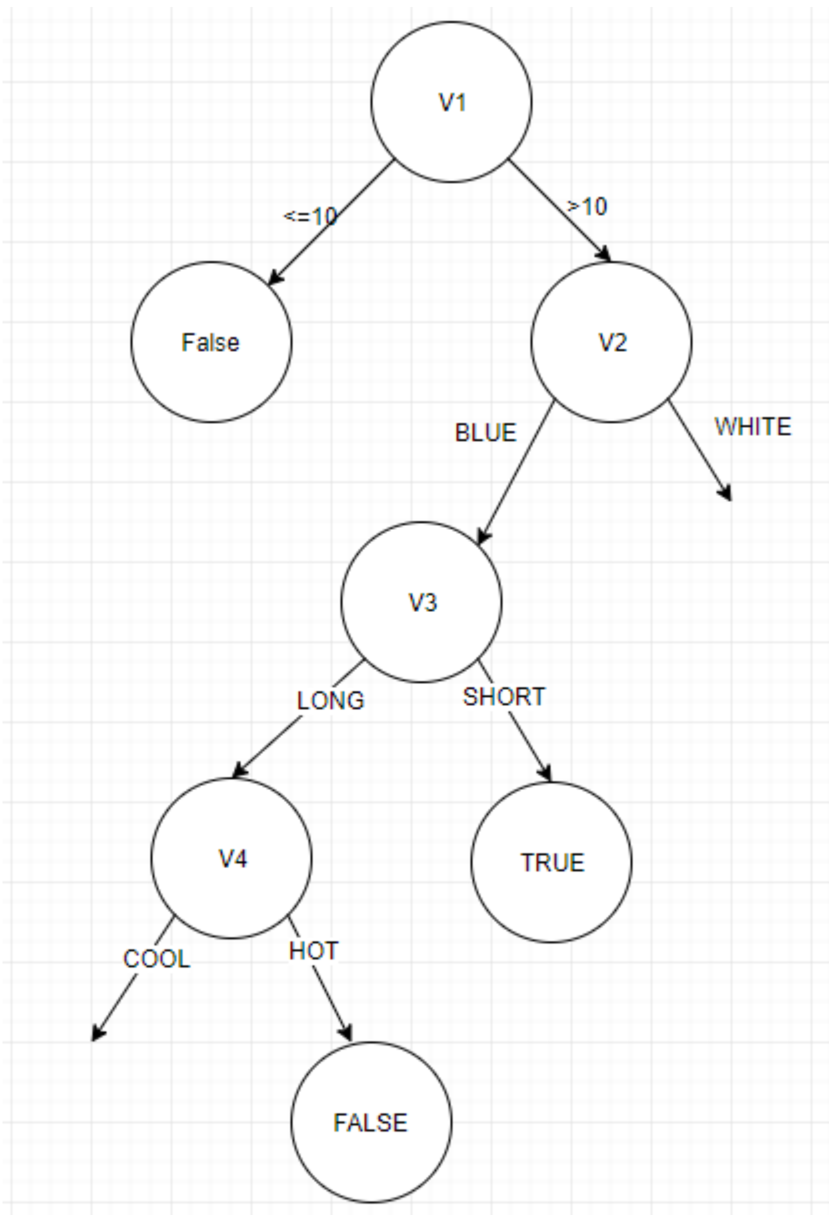
- ➔ Entropy of Class: H(C)

Class	**Frequency**
True	0
False	1

$$H(C) = 0$$

Therefor we don't need to split if V1>10 and V2=BLUE and V3 = SHORT and V4 = HOT. It will be a leaf node.

Tree:



Checking for next split if V1>10 and V2 = WHITE:

➔ Entropy of Class: H(C)

Class	Frequency
True	2
False	5

$$H(C) = 0.863$$

Checking information gain for V5:

V5 = HIGH

Class	Frequency
True	0
False	3

$$H(C|V5 = HIGH) = 0$$

V5=LOW

Class	Frequency
True	2
False	2

$$H(C|V5 = LOW) = 1$$
$$H(C|V5) = 0.571$$

Checking information gain for V4:

V4 = COOL

Class	Frequency
True	2
False	1

$$H(C|V4 = COOL) = 0.918$$

V4 = HOT

Class	Frequency
True	0
False	4

$$H(C|V4 = HOT) = 0$$
$$H(C|V4) = 0.393$$

Checking information gain for V3:

V3=LONG

Class	Frequency
True	1
False	3

$$H(C|V3 = LONG) = 0.811$$

V3=SHORT

Class	Frequency
True	1
False	2

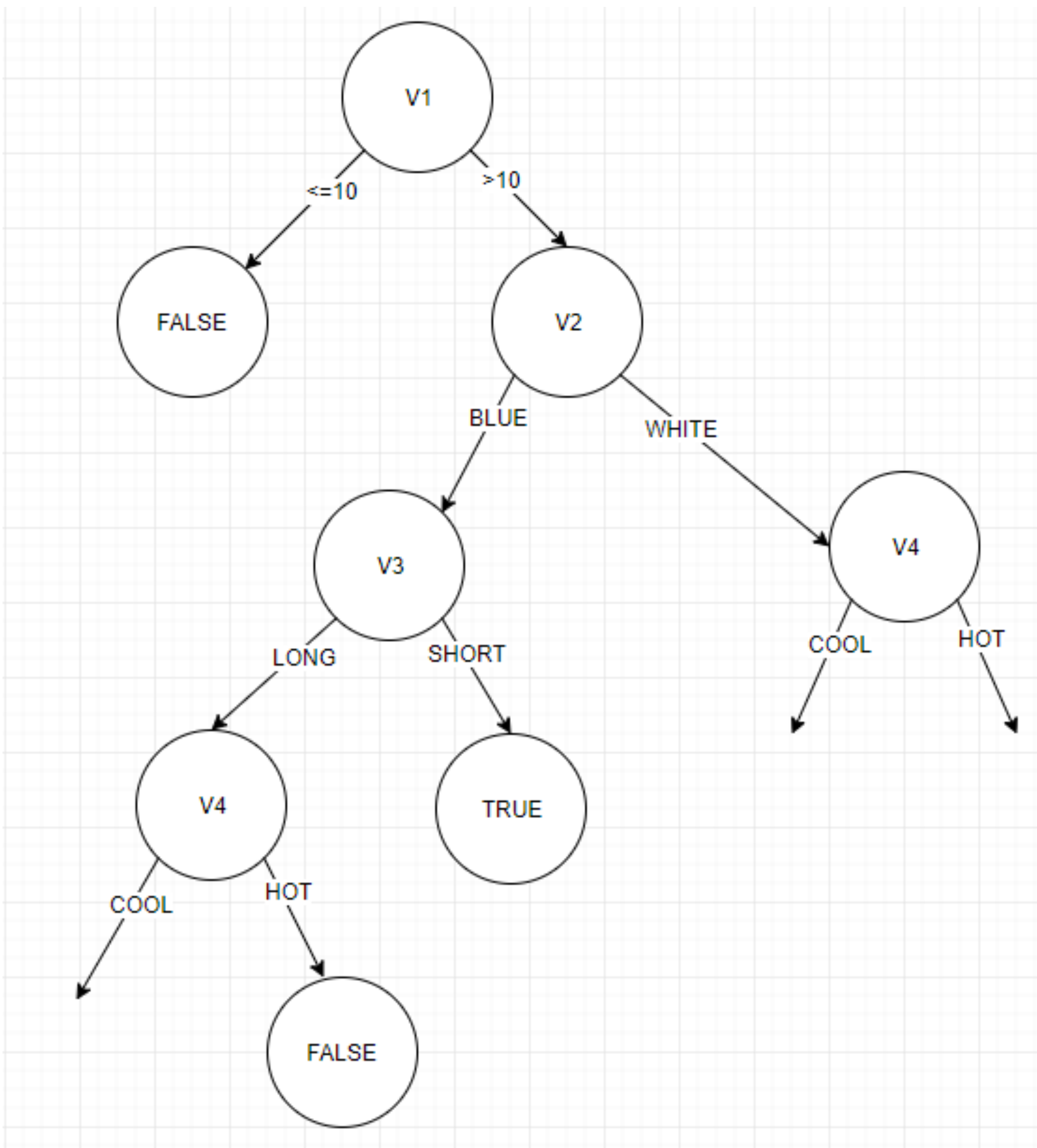
$$H(C|V3 = SHORT) = 918$$
$$H(C|V3) = 0.856$$

Information Gains Table:

Attribute	IG
V3	0.863-0.856=0.006
V4	0.863-0.393=0.46
V5	0.863-0.571=0.29

The next best attribute to split will be V4 based on IG. Using the attribute V4 to split while building the decision tree:

Tree:



Checking for split if V1>10 and V2=WHITE and V4 = HOT:

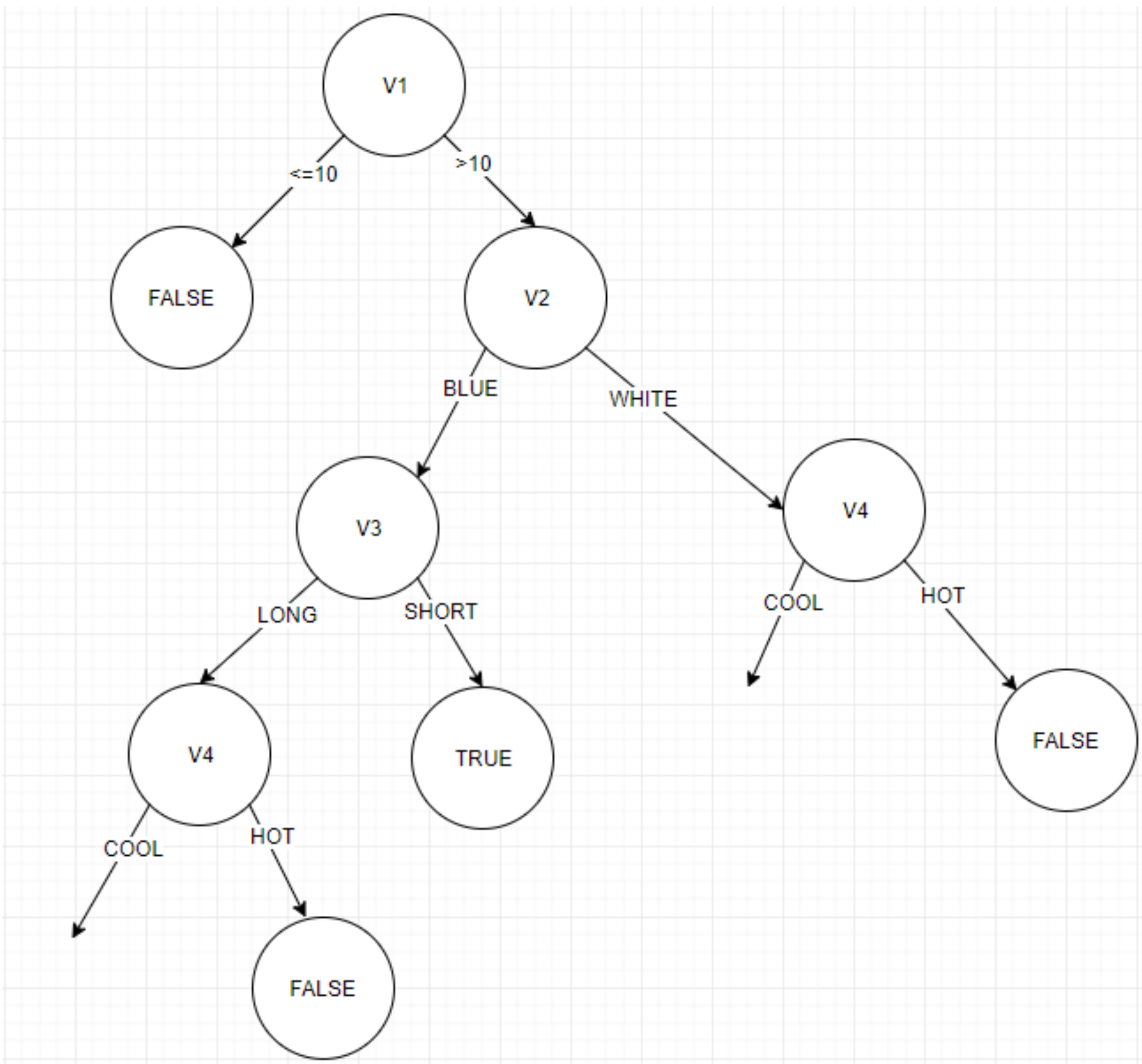
➔ Entropy of Class: H(C)

Class	**Frequency**
True	0
False	4

$$H(C) = 0$$

Therefor we don't need to split if V1>10 and V2=WHITE and V4 = HOT. It will be a leaf node.

Tree:



Checking for next split if V1>10 and V2 = WHITE and V4 = COOL:

➔ Entropy of Class: H(C)

Class	Frequency
True	2
False	1

$$H(C) = 0.918$$

Checking information gain for V5:

V5 = HIGH

Class	Frequency
True	0
False	1

$$H(C|V5 = HIGH) = 0$$

V5=LOW

Class	Frequency
True	2
False	0

$$H(C|V5 = LOW) = 0$$
$$H(C|V5) = 0$$

Checking information gain for V3:

V3=LONG

Class	Frequency
True	1
False	1

$$H(C|V3 = LONG) = 1$$

V3=SHORT

Class	Frequency
True	1
False	0

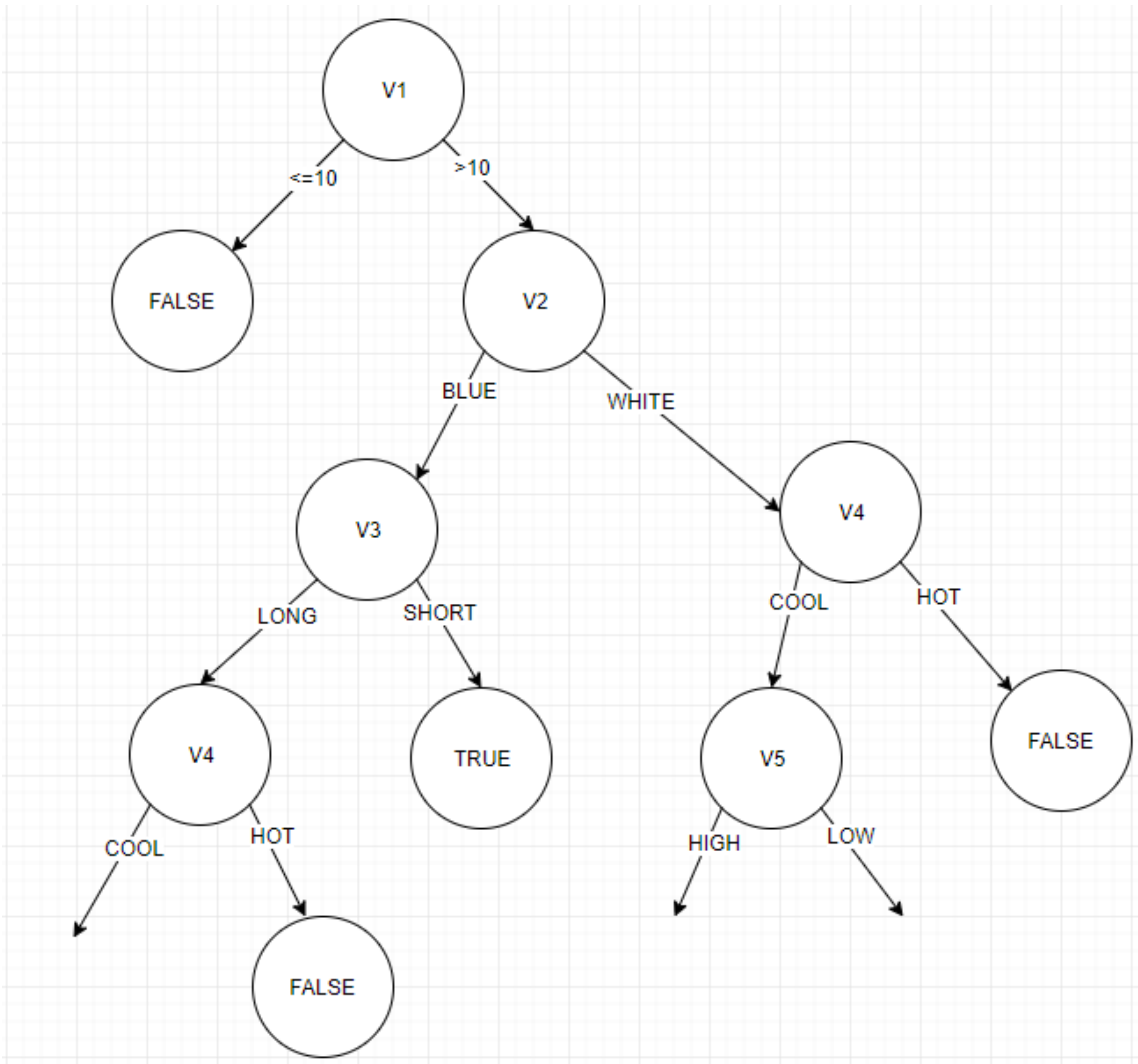
$$H(C|V3 = SHORT) = 0$$
$$H(C|V3) = 0.66$$

Information Gains Table:

Attribute	IG
V3	0.918-0.66=0.258
V5	0.918-0=0.918

The next best attribute to split will be V5 based on IG. Using the attribute V5 to split while building the decision tree:

Tree:



Checking for split if V1>10 and V2=WHITE and V4 = COOL and V5=HIGH:

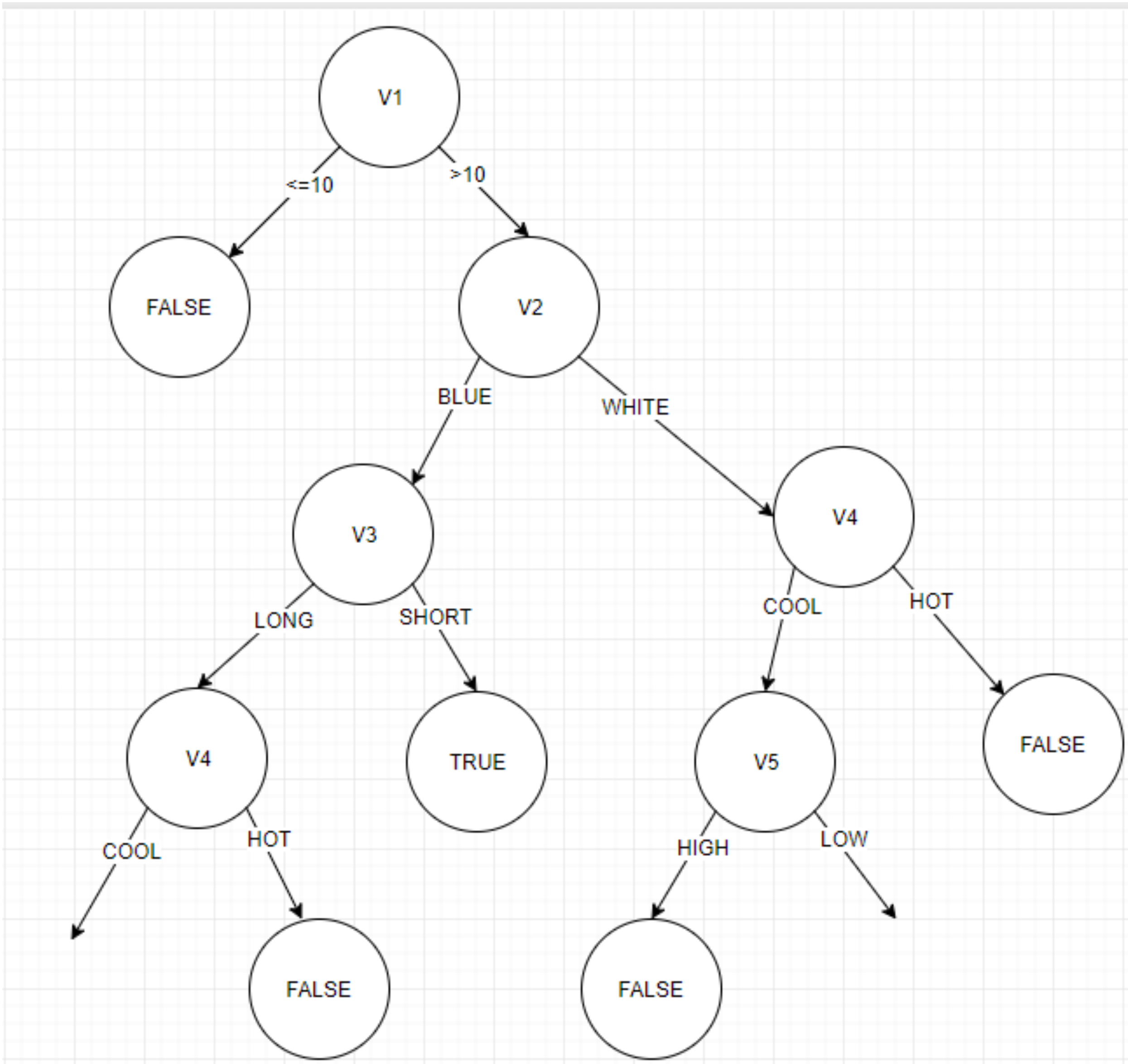
➔ Entropy of Class: H(C)

Class	**Frequency**
True	0
False	1

$$H(C) = 0$$

Therefor we don't need to split if V1>10 and V2=WHITE and V4 = COOL and V5=HIGH. It will be a leaf node.

Tree:



Checking for split if V1>10 and V2=WHITE and V4 = COOL and V5=LOW:

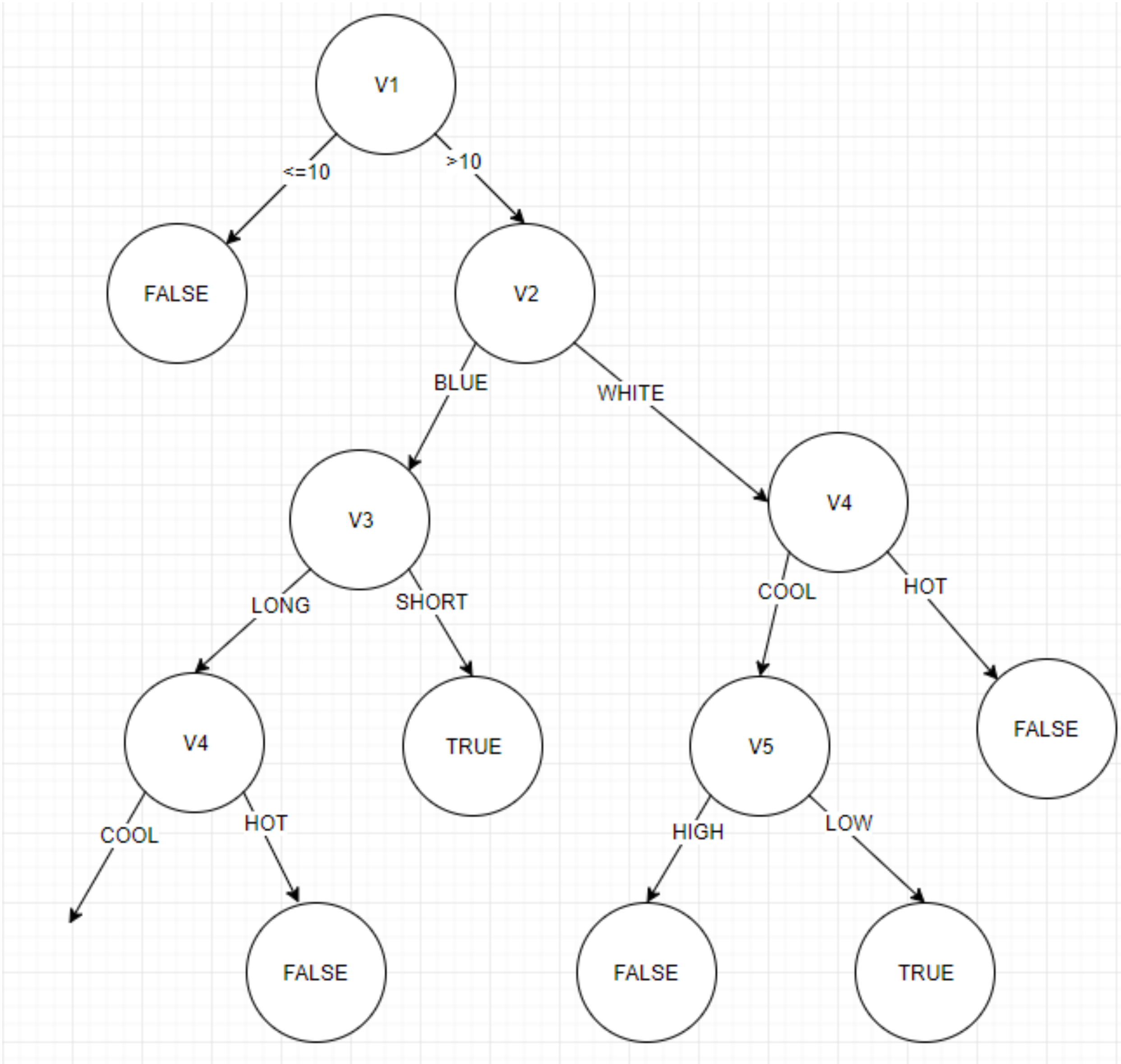
- ➔ Entropy of Class: H(C)

Class	Frequency
True	2
False	0

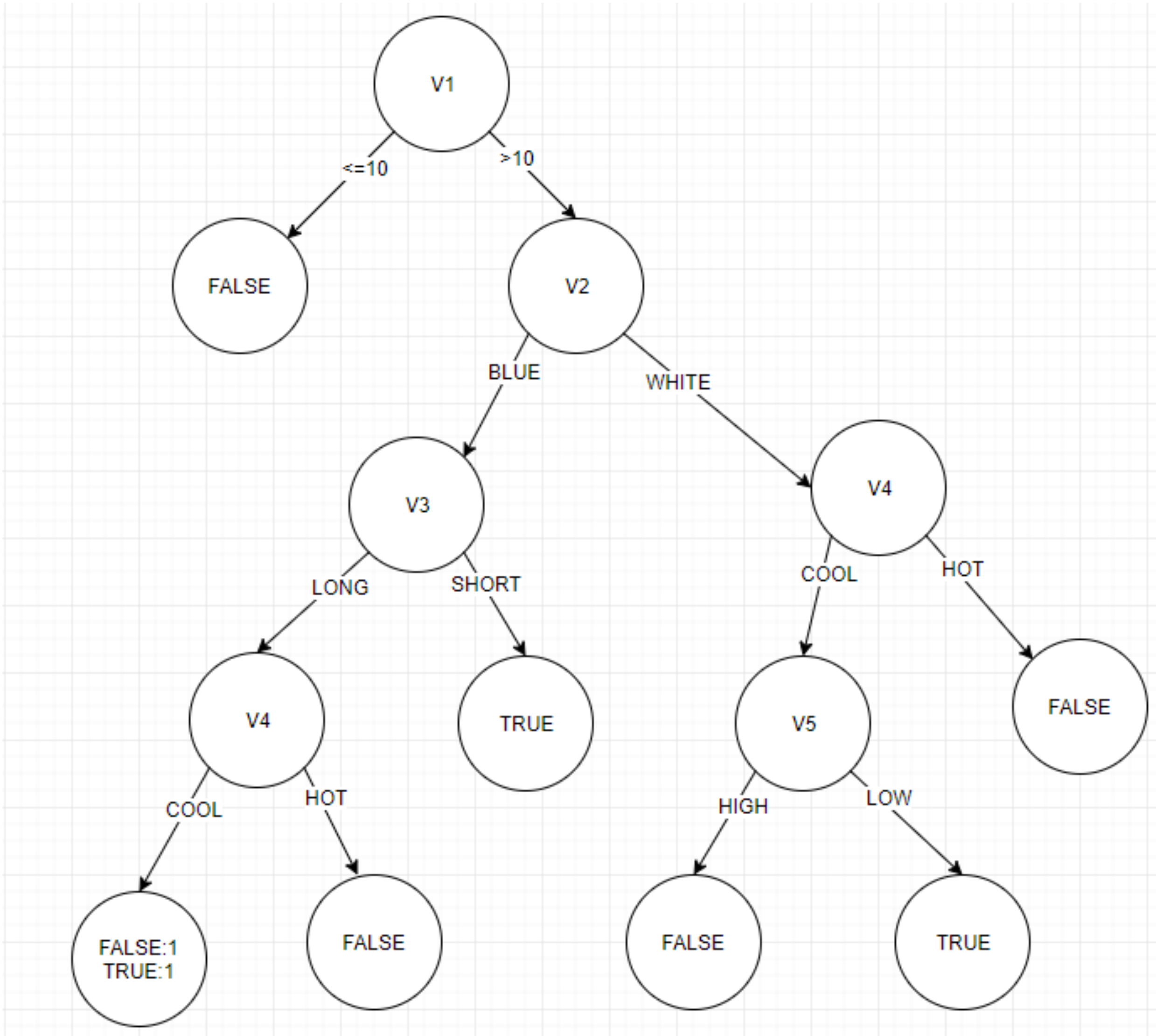
$$H(C) = 0$$

Therefor we don't need to split if V1>10 and V2=WHITE and V4 = COOL and V5=LOW. It will be a leaf node.

Tree:


V1
<=10
>10
FALSE
V2
BLUE
WHITE
V3
V4
LONG
SHORT
COOL
HOT
V4
TRUE
V5
FALSE
COOL
HOT
HIGH
LOW
FALSE
FALSE
TRUE

Final Decision Tree:



2. Constructing decision tree on the given data by using GINI INDEX

GINI INDEX of Class: H(C)

Class	Frequency
True	7
False	9

$$GINI(C) = 1 - \left(\frac{9}{16}\right)^2 - \left(\frac{7}{16}\right)^2 = 0.492$$

Checking for best split for continuous attribute V1:

When split happens at V1=7

$V1 \leq 7$

Class	Frequency

True	0
False	1

$$GINI(V1 \leq 7) = 0$$

$V1 > 7$

Class	Frequency
True	7
False	8

$$GINI(V1 > 7) = 0.4977$$
$$GINI(V1) = 0.465$$

When split happens at V1=10

$V1 \leq 10$

Class	Frequency
True	0
False	2

$$GINI(V1 \leq 10) = 0$$

$V1 > 10$

Class	Frequency
True	7
False	7

$$GINI(V1 > 10) = 0.5$$
$$GINI(V1) = 0.4375$$

When split happens at V1=11

$V1 \leq 11$

Class	Frequency
True	1
False	2

$$GINI(V1 \leq 11) = 0.44$$

$V1 > 11$

Class	Frequency
True	6
False	7

$$GINI(V1 > 11) = 0.497$$
$$GINI(V1) = 0.4863$$

When split happens at V1=13

$V1 \leq 13$

Class	Frequency
True	1
False	3

$$GINI(V1 \leq 13) = 0.375$$

$V1 > 13$

Class	Frequency
True	6
False	6

$$GINI(V1 > 13) = 0.5$$
$$GINI(V) = 0.468$$

When split happens at V1=15

$V1 \leq 15$

Class	Frequency
True	2
False	3

$$GINI(V1 \leq 15) = 0.48$$

$V1 > 15$

Class	Frequency
True	5
False	6

$$GINI(V1 > 15) = 0.495$$
$$GINI(V) = 0.49$$

When split happens at V1=18

$V1 \leq 18$

Class	Frequency
True	2
False	4

$$GINI(V1 \leq 18) = 0.44$$

$V1 > 18$

Class	Frequency
True	5
False	5

$$GINI(V1 > 18) = 0.5$$
$$GINI(V) = 0.4775$$

When split happens at V1=20

$V1 \leq 20$

Class	Frequency
True	2
False	5

$$GINI(V1 \leq 20) = 0.408$$

$V1 > 20$

Class	Frequency
True	5
False	4

$$GINI(V1 > 20) = .494$$
$$GINI(V) = 0.456$$

When split happens at V1=22

$V1 \leq 22$

Class	Frequency
True	2
False	6

$$GINI(V1 \leq 22) = 0.375$$

$V1 > 22$

Class	Frequency
True	5
False	3

$$GINI(V1 > 22) = .468$$
$$GINI(V1) = 0.4215$$

When split happens at V1=27

$V1 \leq 27$

Class	Frequency
True	3
False	6

$$GINI(V1 \leq 27) = 0.44$$

$V1 > 27$

Class	Frequency
True	4
False	3

$$GINI(V1 > 27) = .489$$
$$GINI(V1) = 0.4614$$

When split happens at V1=30

$V1 \leq 30$

Class	Frequency
True	4
False	6

$$GINI(V1 \leq 30) = 0.480$$

$V1 > 30$

Class	Frequency
True	3
False	3

$$GINI(V1 > 30) = 0.5$$
$$H(C|V) = 0.4875$$

When split happens at V1=32

$V1 \leq 32$

Class	Frequency
True	5
False	6

$$GINI(V1 \leq 32) = 0.496$$

$V1 > 32$

Class	Frequency
True	2
False	3

$$GINI(V1 > 32) = 0.48$$
$$GINI(V1) = 0.491$$

When split happens at V1=35

$V1 \leq 35$

Class	Frequency
True	6
False	6

$$GINI(V1 \leq 35) = 0.5$$

$V1 > 35$

Class	Frequency
True	1
False	3

$$GINI(V1 > 35) = 0.375$$
$$H(C|V) = 0.4687$$

When split happens at V1=37

$V1 \leq 37$

Class	Frequency
True	6
False	7

$$GINI(V1 \leq 37) = 0.497$$

$V1 > 37$

Class	Frequency
True	1
False	2

$$GINI(V1 > 37) = 0.44$$
$$GINI(V1) = 0.4375$$

When split happens at V1=40

$V1 \leq 40$

Class	Frequency
True	7
False	7

$$GINI(V1 \leq 40) = 0.5$$

$V1 > 40$

Class	Frequency
True	0
False	2

$$GINI(V1 > 40) = 0$$
$$H(C|V) = 0.4375$$

When split happens at V1=43

$V1 \leq 43$

Class	Frequency
True	7
False	8

$$GINI(V1 \leq 43) = 0.497$$

$V1 > 43$

Class	Frequency
True	0
False	1

$$GINI(V1 > 43) = 0$$
$$GINI(V1) = 0.465$$

The minimum GINI INDEX occurs at Split V1=22. Splitting the categorical attribute V1 as $V1 \leq 22$ and $V1 > 22$. Modified Data:

V1	V2	V3	V4	V5	Class

<=22	BLUE	LONG	HOT	HIGH	F
<=22	BLUE	LONG	COOL	HIGH	F
<=22	WHITE	LONG	HOT	HIGH	F
<=22	WHITE	LONG	COOL	HIGH	F
<=22	WHITE	SHORT	COOL	HIGH	F
<=22	WHITE	SHORT	HOT	HIGH	F
<=22	BLUE	SHORT	HOT	HIGH	T
<=22	BLUE	SHORT	COOL	HIGH	T
>22	BLUE	LONG	HOT	LOW	F
>22	WHITE	LONG	HOT	LOW	F
>22	WHITE	SHORT	HOT	LOW	F
>22	BLUE	LONG	COOL	LOW	T
>22	BLUE	SHORT	COOL	LOW	T
>22	BLUE	SHORT	HOT	LOW	T
>22	WHITE	LONG	COOL	LOW	T
>22	WHITE	SHORT	COOL	LOW	T

Checking GINI Index for V5:

V5 = HIGH

Class	**Frequency**
True	2
False	6

$$GINI(V5 = HIGH) = 0.375$$

V5=LOW

Class	**Frequency**
True	5
False	3

$$GINI(V5 = LOW) = 0.468$$
$$GINI(V5) = 0.4215$$

Checking GINI Index for V4:

V4 = COOL

Class	**Frequency**
True	5
False	3

$$GINI(V4 = COOL) = 0.468$$

V4 = HOT

Class	**Frequency**
True	2
False	6

$$GINI(V4 = HOT) = 0.375$$
$$GINI(V4) = 0.4215$$

Checking GINI Index for V3:

V3=LONG

Class	Frequency
True	2
False	6

$$GINI(V3 = LONG) = 0.375$$

V3=SHORT

Class	Frequency
True	5
False	3

$$GINI(V3 = SHORT) = 0.468$$
$$GINI(V3) = 0.4215$$

Checking GINI Index for V2:

V2=BLUE

Class	Frequency
True	5
False	3

$$GINI(V2 = BLUE) = 0.468$$

V2=WHITE

Class	Frequency
True	2
False	6

$$GINI(V2 = WHITE) = 0.375$$
$$GINI(V2) = 0.4215$$

Checking GINI Index for V1:

$V1 \leq 22$

Class	Frequency
True	2
False	6

$$GINI(V1 \leq 22) = .375$$

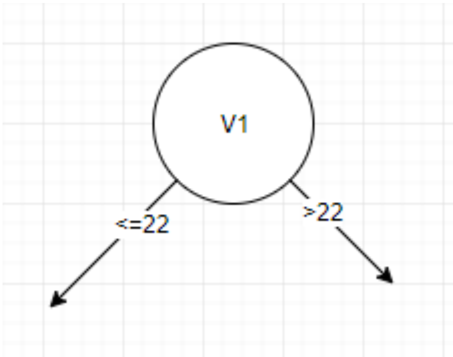
$V1 > 22$

Class	Frequency
True	5
False	3

$$GINI(V1 > 22) = 0.468$$
$$GINI(V1) = 0.4215$$

Since the GINI index for all the attributes Is same, we choose to V1 node to split upon.

Tree:



Checking for next split if V1<=22:

Checking GINI Index for V5:

Class	Frequency
True	2
False	6

$$GINI(V5) = 0.375$$

Checking GINI Index for V4:

V4 = COOL

Class	Frequency
True	1
False	3

$$GINI(V4 = COOL) = 0.375$$

V4 = HOT

Class	Frequency
True	1
False	3

$$GINI(V4 = HOT) = 0.375$$
$$GINI(V4) = 0.375$$

Checking GINI Index for V3:

V3=LONG

Class	Frequency
True	0
False	4

$$GINI(V3 = LONG) = 0$$

V3=SHORT

Class	Frequency
True	2
False	2

$$GINI(V3 = SHORT) = 0.5$$

$$GINI(V3) = 0.25$$

Checking GINI Index for V2:

V2=BLUE

Class	Frequency
True	2
False	2

$$GINI(V2 = BLUE) = 0.5$$

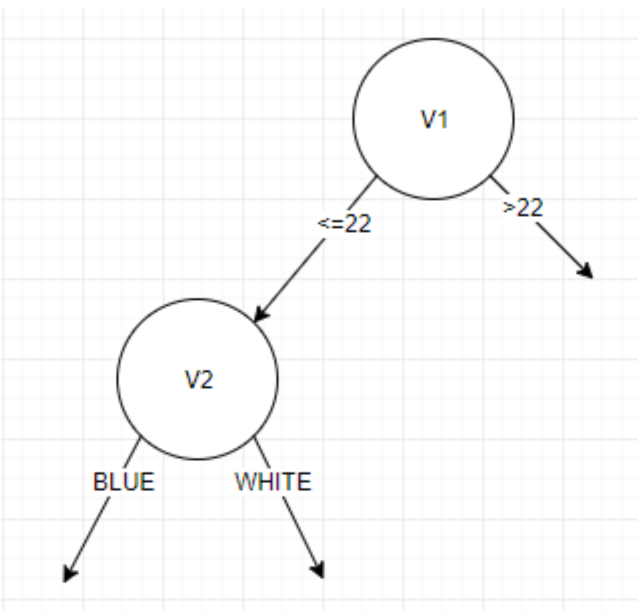
V2=WHITE

Class	Frequency
True	0
False	4

$$GINI(V2 = WHITE) = 0$$
$$GINI(V2) = 0.25$$

V2 and V3 both have the lowest GINI. Therefor selecting V2 to split upon next.

Tree:

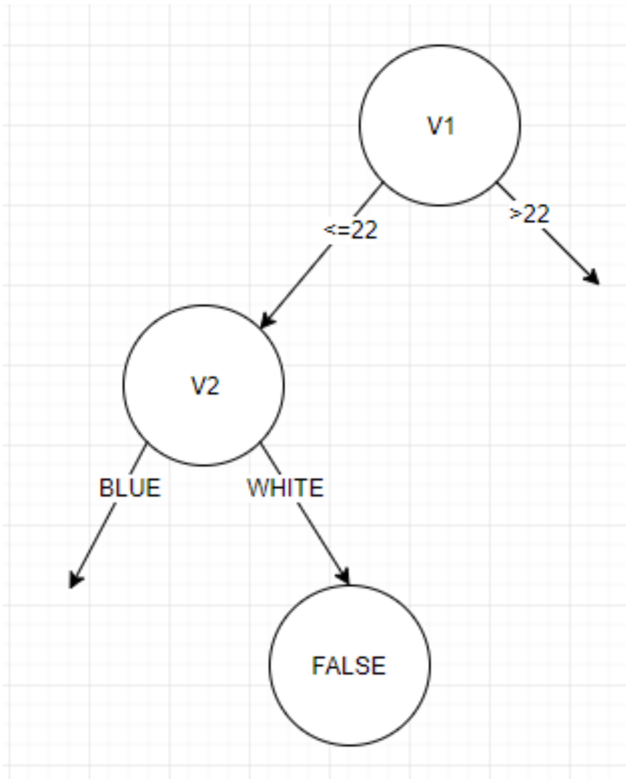


Checking for next split if V1<=22 and V2 = WHITE:

Class	Frequency
True	0
False	4

GINI = 0. There is no need to split. This will lead to leaf node in the tree.

Tree:



Checking for next split if V1<=22 and V2 = BLUE:

Class	Frequency
True	2
False	2

GINI(Class) = 0.5

Checking GINI Index for V5:

Class	Frequency
True	2
False	2

$$GINI(V5) = 0.5$$

Checking GINI Index for V4:

V4 = COOL

Class	Frequency
True	1
False	1

$$GINI(V4 = COOL) = 0.5$$

V4 = HOT

Class	Frequency
True	1
False	1

$$GINI(V4 = HOT) = 0.5$$
$$GINI(V4) = 0.5$$

Checking GINI Index for V3:

V3=LONG

Class	Frequency
True	0
False	2

$$GINI(V3 = LONG) = 0$$

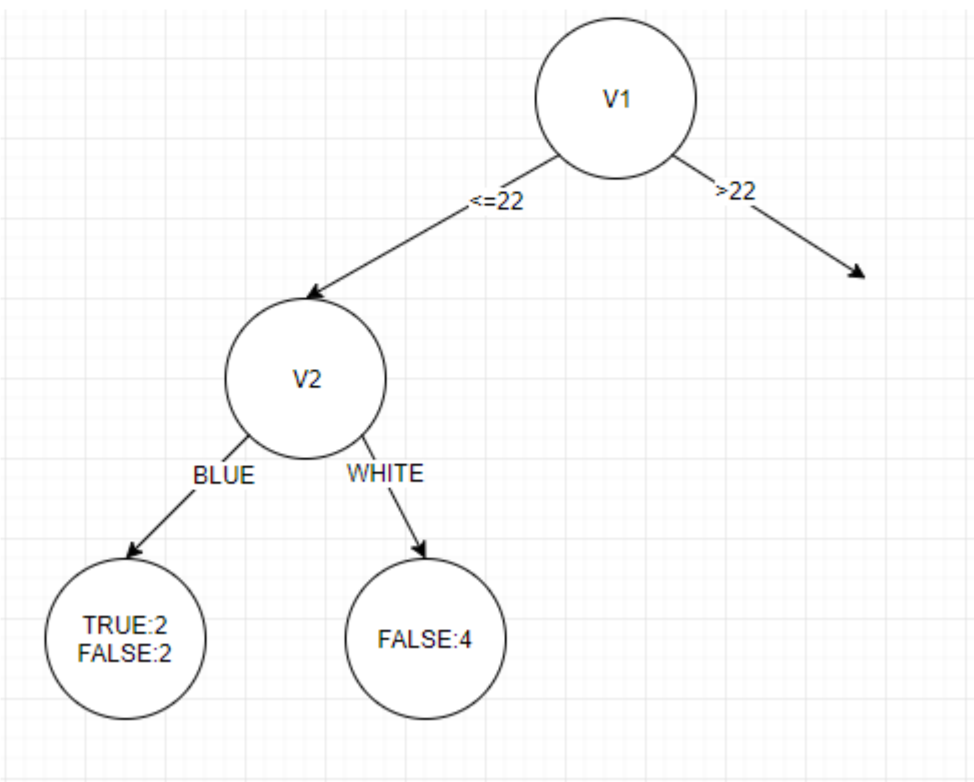
V3=SHORT

Class	Frequency
True	2
False	0

$$GINI(V3 = SHORT) = 0$$
$$GINI(V3) = 0$$

V3 has have the lowest GINI. Therefor selecting V3 to split upon next. Since we need to construct the tree only till depth 2.

Tree:



Checking for next split if V1>22:

Checking GINI Index for V5:

Class	Frequency
True	5
False	3

$$GINI(V5) = 0.468$$

Checking GINI Index for V4:

V4 = COOL

Class	Frequency
True	4
False	0

$$GINI(V4 = COOL) = 0$$

V4 = HOT

Class	Frequency
True	1
False	3

$$GINI(V4 = HOT) = 0.375$$
$$GINI(V4) = 0.1875$$

Checking GINI Index for V3:

V3=LONG

Class	Frequency
True	2
False	2

$$GINI(V3 = LONG) = 0.5$$

V3=SHORT

Class	Frequency
True	3
False	1

$$GINI(V3 = SHORT) = 0.375$$
$$GINI(V3) = 0.4375$$

Checking GINI Index for V2:

V2=BLUE

Class	Frequency
True	3
False	1

$$GINI(V2 = BLUE) = 0.375$$

V2=WHITE

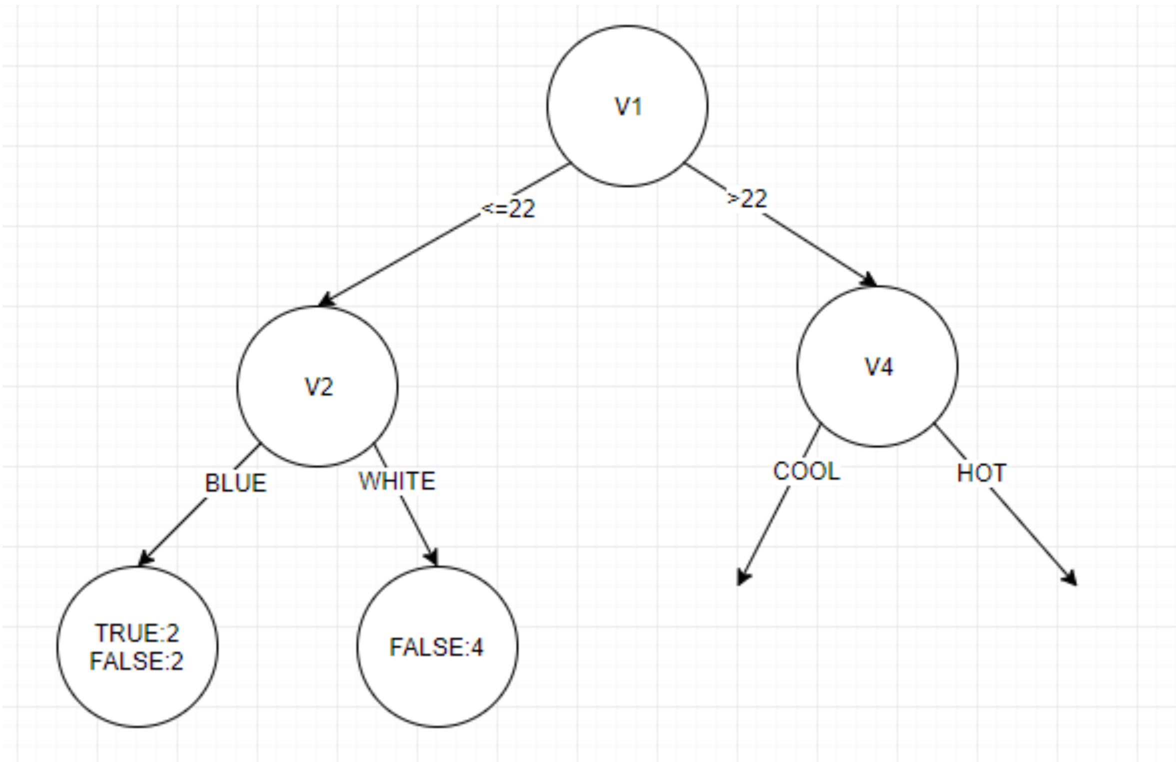
Class	Frequency
True	2

False	2

$$GINI(V2 = WHITE) = 0.5$$
$$GINI(V2) = 0.4375$$

V4 has the lowest GINI. Therefor selecting V4 to split upon next.

Tree:

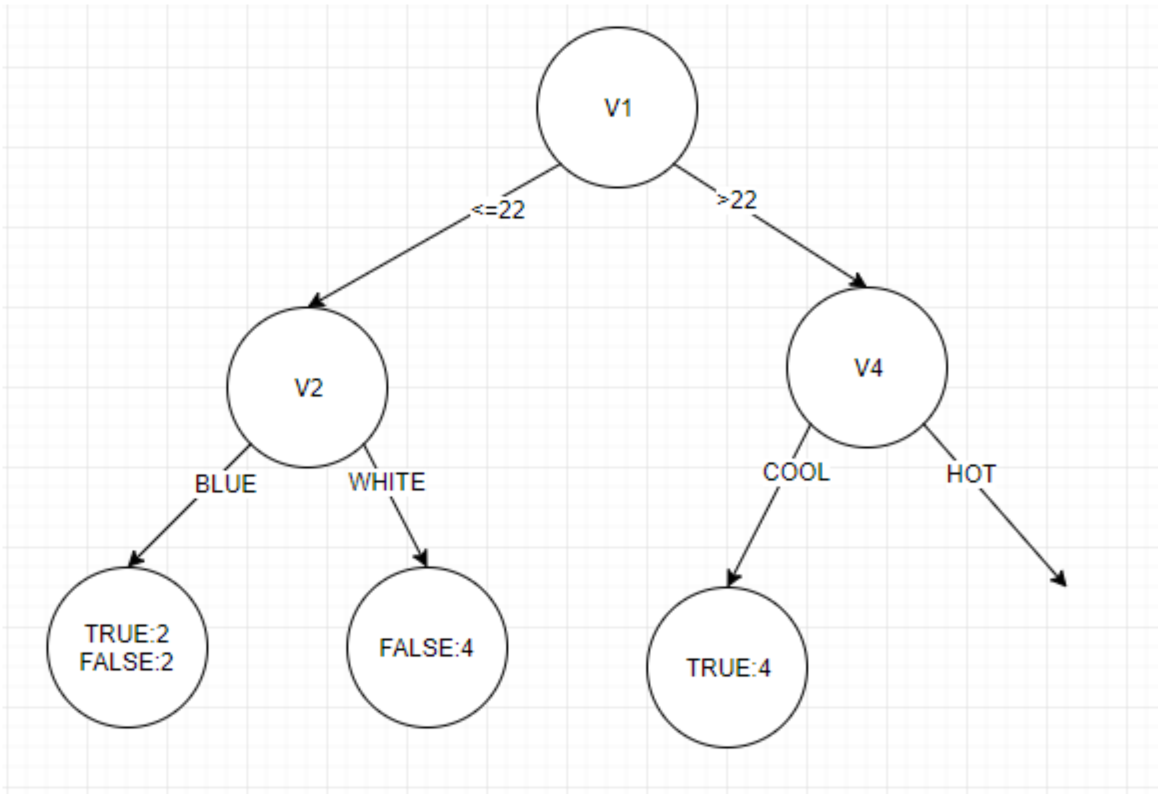


Checking for next split if V1>22 and V4=COLD:

Class	Frequency
True	4
False	0

GINI(C) = 0, We don need to split is further. It will be a leaf node in the tree.

Tree:



Checking for next split if V1>22 and V4=HOT:

Checking GINI Index for V5:

Class	Frequency
True	1
False	3

$$GINI(V5) = 0.375$$

Checking GINI Index for V3:

V3=LONG

Class	Frequency
True	0
False	2

$$GINI(V3 = LONG) = 0$$

V3=SHORT

Class	Frequency
True	1
False	1

$$GINI(V3 = SHORT) = 0.5$$
$$GINI(V3) = 0.25$$

Checking GINI Index for V2:

V2=BLUE

Class	Frequency
True	1
False	1

$$GINI(V2 = BLUE) = 0.5$$

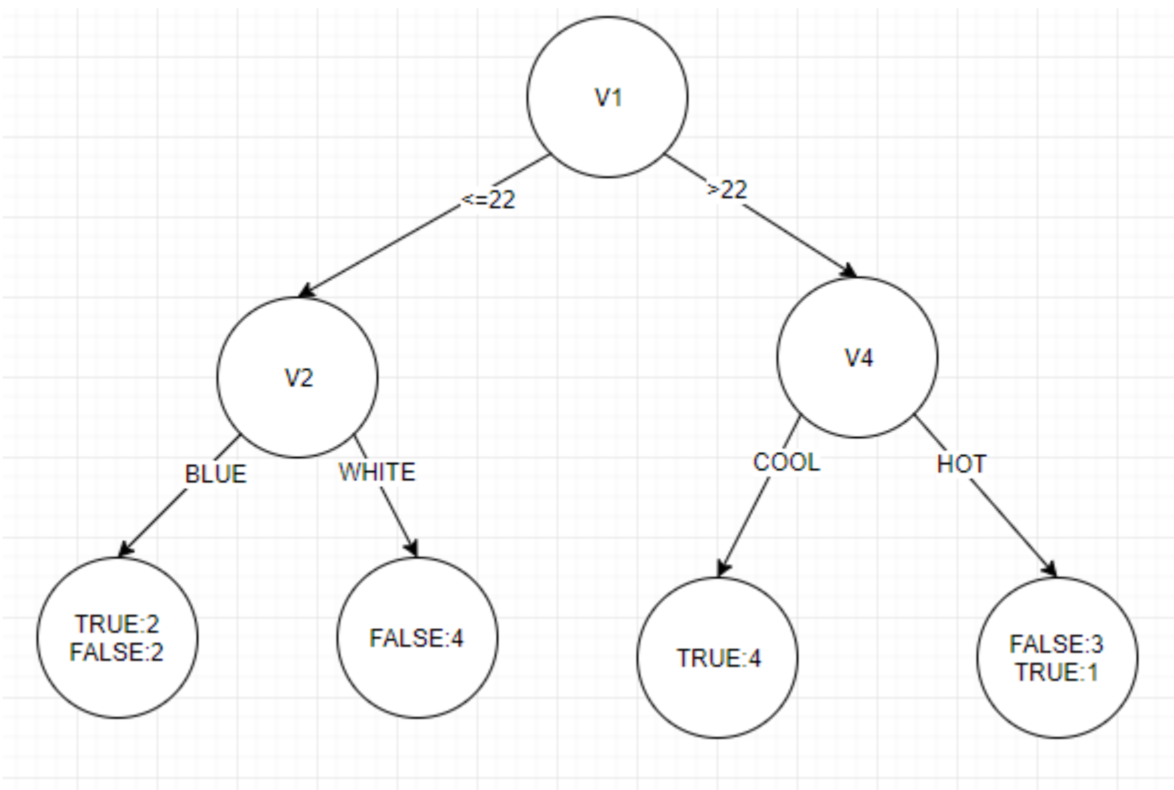
V2=WHITE

Class	Frequency
True	0
False	2

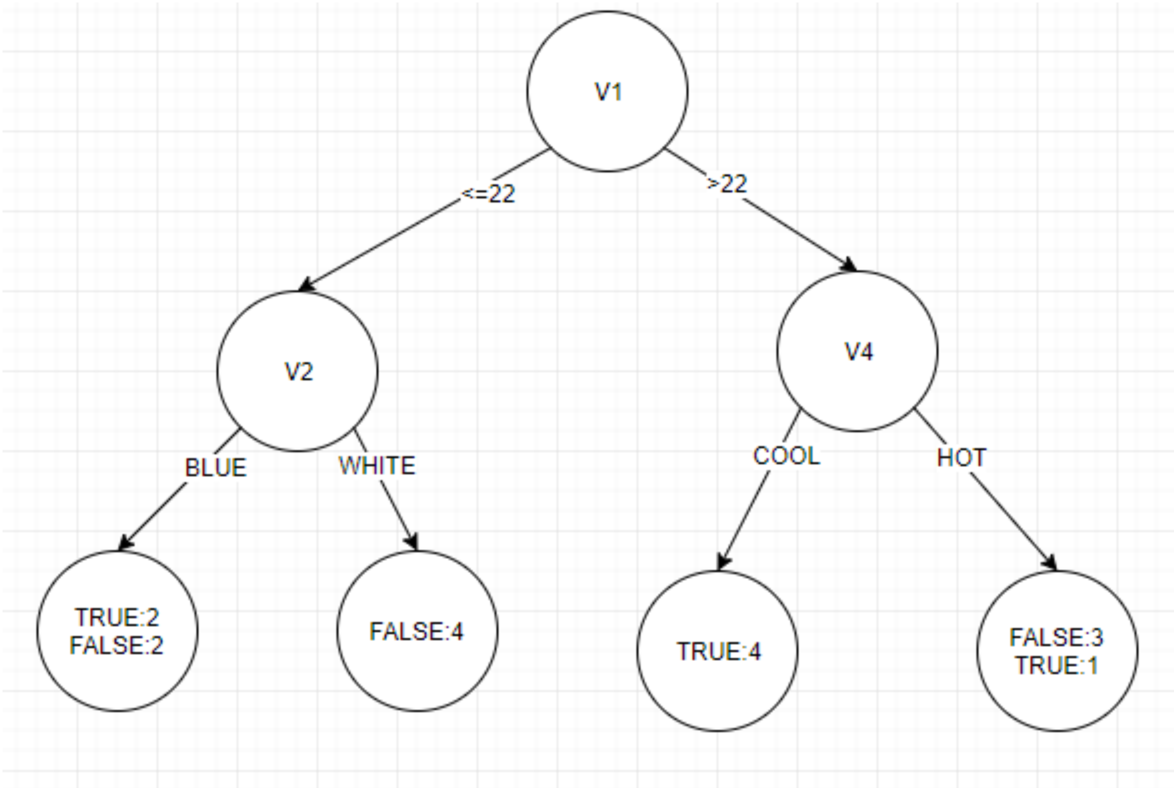
$$GINI(V2 = WHITE) = 0$$
$$GINI(V2) = 0.25$$

V2 and V3 have the lowest GINI. Therefore, V2 will be selected to split on next. As the decision tree should have max depth of 2. We will not split the tree anymore.

Tree:



Final Decision Tree Using GINI Index:



3. The two Trees mainly differs with respect to the value of V1(The continuous attribute value). The first Tree splits the continuous attribute into range V1<=10 and V1>10 and the second tree splits the same as V1<=22 and V1>22. Both trees may behave similar in some cases but are very different from each other due to the different split of the continuous attribute.

   - The trees may result in different class label when V1<=10. The First tree doesn't consider any other attributes on the range V1<=10 and directly classifies the record with a label 'F' while the second tree will consider other attributes as well and hence may result in a different class label.
     For example: Consider the test record R(10,BLUE,SHORT,HOT,HIGH)
     (10,BLUE,SHORT,HOT,HIGH)→ Classified as False by first tree but the second tree can classify the same as true (if the tree gives priority to 'T' label in case of a tie).
   - Similarly, the trees will behave differently when V1>22 and V4 = COOL. The first try will look for other attributes and try to assign a label based on the attributes whereas the second tree will directly assign the label 'T' for such records.
     For Example: R(50,WHITE,LONG,COOL,HIGH)
     (50,WHITE,LONG,COOL,HIGH)→Classifies as 'False' by the first decision tree while as 'True' by the second decision tree.

4. Consider the training data set:
   Assumption: the tree gives priority to 'T' label in case of a tie.

V1	V2	V3	V4	V5	Class	Classification Decision tree (IG)	Classification Decision tree (GINI)
7	BLUE	LONG	HOT	HIGH	F	F	T
10	WHITE	SHORT	COOL	HIGH	F	F	F
11	BLUE	SHORT	HOT	HIGH	T	T	T
13	WHITE	LONG	HOT	HIGH	F	F	F
15	BLUE	SHORT	COOL	HIGH	T	T	T
18	WHITE	SHORT	HOT	HIGH	F	F	F
20	BLUE	LONG	COOL	HIGH	F	T	T
22	WHITE	LONG	COOL	HIGH	F	F	F
27	WHITE	LONG	COOL	LOW	T	T	T
30	BLUE	SHORT	COOL	LOW	T	T	T
32	WHITE	SHORT	COOL	LOW	T	T	T
35	BLUE	SHORT	HOT	LOW	T	T	F
37	WHITE	SHORT	HOT	LOW	F	F	F
40	BLUE	LONG	COOL	LOW	T	T	T
43	WHITE	LONG	HOT	LOW	F	F	F
50	BLUE	LONG	HOT	LOW	F	F	F

Optimistic training error for DT1(IG) = 1/16
Optimistic training error for DT1(GINI) = 3/16
So, we can say that the decision tree constructed using Information Gain performs better on the training data than the decision tree constructed using GINI index.

The outcome of the test data can’t be determined. The performance the decision trees will completely depend upon the type of the test data used.

2)

a. Number of misclassifications = 2+2+4=8
   Number of instances classified = 34

$$Optimistic\ error = \frac{8}{34} = 0.235$$

$$Pessimistic\ error = \frac{8 + (7 * 0.5)}{34} = \frac{11.5}{34} = 0.338$$

b.

Width	Temperature	Size	Color	Label	Predicted Label
Long	Low	Small	White	No	Yes
Short	Low	Big	Red	No	Yes
Short	Low	Big	Red	No	Yes
Short	Low	Big	Blue	No	No
Short	Low	Small	Blue	No	No
Short	Low	Big	White	No	Yes
Long	Low	Big	Blue	Yes	Yes
Long	Low	Big	Red	Yes	Yes
Long	Low	Big	Blue	Yes	Yes
Long	Low	Small	Red	Yes	Yes
Long	Low	Small	Red	Yes	Yes
Long	Low	Small	White	Yes	Yes
Short	Low	Big	Green	Yes	Yes
Short	Low	Big	Red	Yes	Yes
Short	High	Big	Blue	Yes	Yes
Short	Low	Small	Blue	Yes	No
Short	High	Small	Red	Yes	No
Short	Low	Small	Red	Yes	Yes
Short	High	Big	Green	Yes	Yes
Short	Low	Big	White	Yes	Yes

<table>
<tr><td></td><td colspan="3">Predicted</td></tr>
<tr><td rowspan="3">Actual</td><td></td><td>Yes</td><td>No</td></tr>
<tr><td>Yes</td><td>12</td><td>2</td></tr>
<tr><td>No</td><td>4</td><td>2</td></tr>
</table>

Accuracy:	14/20	0.7
Precision:	12/16	0.75
Recall:	12/14	0.857
F1 score:	24/30	0.8
Error rate:	6/20	0.3

3)

a. <u>Before splitting on node 'Color'</u>
   Number of misclassifications = 10
   Total number of instances =25

   Optimistic error = 10/25 = 0.4

   <u>After splitting on node 'Color'</u>
   Number of misclassifications = 8
   Total number of instances = 25

   Optimistic error = 8/25 = 0.32

   Since the optimistic error reduced after splitting, the node should not be pruned.

b. <u>Before splitting on node 'Color'</u>
   Number of misclassifications = 10
   Number of leaf nodes = 1
   Total number of instances =25

   Pessimistic error = (10+0.8)/25 = 0.432

   <u>After splitting on node 'Color'</u>
   Number of misclassifications = 8
   Number of leaf nodes = 4
   Total number of instances = 25

   Pessimistic error = (8+4*0.8)/25 = 0.448

   Since the pessimistic error increased after splitting, the node should be pruned.

c.

Classification when ‘Color’ node is pruned

Width	Temperature	Size	Label	Predicted Label
Long	Low	Small	No	Yes
Short	Low	Big	No	Yes
Short	Low	Big	No	Yes
Short	Low	Big	No	Yes
Short	Low	Small	No	Yes
Short	Low	Big	No	Yes
Long	Low	Big	Yes	Yes
Long	Low	Big	Yes	Yes
Long	Low	Big	Yes	Yes
Long	Low	Small	Yes	Yes
Long	Low	Small	Yes	Yes
Long	Low	Small	Yes	Yes
Short	Low	Big	Yes	Yes
Short	Low	Big	Yes	Yes
Short	High	Big	Yes	Yes
Short	Low	Small	Yes	Yes
Short	High	Small	Yes	No
Short	Low	Small	Yes	Yes
Short	High	Big	Yes	Yes
Short	Low	Big	Yes	Yes

	Predicted		
Actual		Yes	No
	Yes	13	1
	No	6	0

Accuracy:	13/20	0.65
Precision:	13/19	0.68
Recall:	13/14	0.93
F1 score:	24/30	0.79
Error rate:	7/20	0.35

	Before Splitting	After Splitting
Training Error	0.294	0.235
Test Error	0.35	0.3

Since both the training error and test error reduced after splitting, the original tree was not over fitting.

4)

a. Euclidean distance formula:

euclidean_distance <- function(x1, y1, x2, y2){

sqrt((x1 – y1)^2) + (x2 – y2)^2))
}

Distance Matrix:

ID	1	2	3	4	5	6	7	8	9
1	0	32.7452	24.6272	9.8488	33.5596	13.6014	10.1980	5.8309	27.5408
2	32.7452	0	8.1394	41.5000	2.2360	45.5000	42.5470	35.5140	5.5901
3	24.6272	8.1394	0	33.5335	9.0138	37.5299	34.5036	27.6134	3.1622
4	9.8488	41.5000	33.5335	0	42.5470	4.0000	2.2360	6.0827	36.5855
5	33.5596	2.2360	9.0138	42.5470	0	46.5429	43.5000	36.6230	6.0207
6	13.6014	45.5000	37.5299	4.0000	46.5429	0	3.6055	10.0498	40.5770
7	10.1980	42.5470	34.5036	2.2360	43.5000	3.6055	0	7.6157	37.5033
8	5.8309	35.5140	27.6134	6.0827	36.6230	10.0498	7.6157	0	30.7001
9	27.5408	5.5901	3.1622	36.5855	6.0207	40.5770	37.5033	30.7001	0

(b)

i. A holdout test dataset consisting of last 4 instances

Test set:

6	48.0	11.0	+
7	45.0	13.0	-
8	38.0	10.0	+
9	7.5	13.5	-

Closest point to 6 is 4 and 4’s class is (+)

Closest point to 7 is 4 and 4’s class is (+)

Closest point to 8 is 1 and 1’s class is (-)

Closest point to 9 is 3 and 3’s class is (+)

<u>Confusion Matrix:</u>

	Actual		
Prediction		TRUE	FALSE
	TRUE	1	2
	FALSE	1	0

Testing accuracy:1/4 = 0.25

ii. 3-fold cross-validation, using the following folds with IDs: [3,6,9], [1,4,7], [2,5,8] respectively

Round 1 – when [3,6,9] is used as test set:

Closest point to 3 is 2 and 2’s class is (-)

Closest point to 6 is 7 and 7’s class is (-)

Closest point to 9 is 2 and 2’s class is (-)

Confusion Matrix:

	Actual		
Prediction		TRUE	FALSE
	TRUE	0	0
	FALSE	2	1

Testing accuracy:1/3 = 0.33

Round 2 – when [1,4,7] is used as test set:

Closest point to 1 is 8 and 8's class is (+)

Closest point to 4 is 6 and 6's class is (+)

Closest point to 7 is 6 and 6's class is (+)

Confusion Matrix:

	Actual		
Prediction		TRUE	FALSE
	TRUE	1	2
	FALSE	0	0

Testing accuracy:1/3 = 0.33

Round 3 – when [2,5,8] is used as test set:

Closest point to 2 is 9 and 9's class is (-)

Closest point to 5 is 9 and 9's class is (-)

Closest point to 8 is 1 and 1's class is (-)

Confusion Matrix:

	Actual		
Prediction		TRUE	FALSE
	TRUE	0	0
	FALSE	1	2

Testing accuracy:2/3 = 0.66

Overall testing accuracy = (0.66 + 0.33 + 0.33) / 3 = 0.44

iii. Leave one out cross validation (LOOCV)

Closest point to 2 is 9 and 9's class is (-)

Test Set Item	Closest Item	Predicted Value	Actual Value	Testing Accuracy
1	8	+	-	0
2	5	-	-	1
3	7	-	+	0
4	7	-	+	0
5	2	-	-	1
6	7	-	+	0
7	4	+	-	0
8	1	-	+	0
9	3	+	-	0

Overall testing accuracy = 2 / 9 = 0.22

c)

Initially, the ratio of positives to negatives will be equal(given). But, when we take one item out as a test item (as part of LOOCV), the majority class will be always opposite to the class of the test item. Therefore, the prediction will always be wrong and thus the accuracy of the model drops to zero.

5)

e)

Overall accuracy comparison

Analysis of KNN Models:

Model	Accuracy
KNN (euclidean)	0.56
KNN (cosine)	0.88
KNN (confidence)	0.9

From the overall accuracy calculations, we can see the KNN Model with confidence performed the best.

Analysis of Decision Tree Models:

Model	Accuracy
Decision Tree	0.58
Decision Tree with cross validation and complexity parameter tuning	0.46

From the overall accuracy calculations, we can see that the basic decision tree model performed the best.

<u>Comparison in terms of confusion matrix</u>

<u>KNN Classifier using Euclidean distance</u>

		Reference			
		1	2	3	4
Prediction	1	8	5	7	6
	2	0	9	1	0
	3	0	0	6	0
	4	1	1	1	5

class 1 instances misclassified = 1
class 2 instances misclassified = 6
class 3 instances misclassified = 9
class 4 instances misclassified = 6

Class 3 has had the greatest number of misclassifications.

<u>KNN classifier using cosine distance</u>

		Reference			
Prediction		1	2	3	4
	1	9	0	1	2
	2	0	15	1	1
	3	0	0	13	1
	4	0	0	0	7

class 1 instances misclassified = 0
class 2 instances misclassified = 0
class 3 instances misclassified = 2
class 4 instances misclassified = 4

Class 4 has had the greatest number of misclassifications, but this number is smaller when compared to the KNN Model using Euclidean distance.

<u>KNN classifier with confidence calculation</u>

		Reference			
Prediction		1	2	3	4
	1	9	0	0	1
	2	0	14	1	1
	3	0	0	14	1
	4	0	1	0	8

class 1 instances misclassified = 0
class 2 instances misclassified = 1
class 3 instances misclassified = 1
class 4 instances misclassified = 3

The statistic for this model is even better.

Class 4 still has the greatest number of misclassifications, but the number has gone down again when compared to the previous model.

Also, the total number of misclassifications has also gone down.

Decision Tree Model

		Reference			
Prediction		1	2	3	4
	1	8	6	7	5
	2	0	8	0	0
	3	0	1	8	1
	4	1	0	0	5

class 1 instances misclassified = 1
class 2 instances misclassified = 7
class 3 instances misclassified = 7
class 4 instances misclassified = 6

Class 2 and 3 has the greatest number of misclassifications.

Decision Tree with cross validation and hyperparameter tuning

		Reference			
Prediction		1	2	3	4
	1	8	7	13	6
	2	0	8	0	0
	3	0	0	2	0
	4	1	0	0	5

class 1 instances misclassified = 1
class 2 instances misclassified = 7
class 3 instances misclassified = 13
class 4 instances misclassified = 6

Class 3 has the greatest number of misclassifications.

Out of all the classes, the number of misclassifications for class 1 is the least. So, we can conclude that class 1 performs better than all the other classes.