Aufgabe DT.01:

Nr.	Alter	Einkommen	Bildung	Kandidat
1	≥ 35	hoch	Abitur	O
2	< 35	niedrig	Master	O
3	≥ 35	hoch	Bachelor	M
4	≥ 35	niedrig	Abitur	M
5	≥ 35	hoch	Master	O
6	< 35	hoch	Bachelor	O
7	< 35	niedrig	Abitur	M

Cal3:

S1 = 4, S2 = 0.7

Alter (>=35, <35), Einkommen (niedrig, hoch), Bildung (Arbitur, Bachelor, Master)

Start: *

1. /O1/
2. /O2/
3. /O2M1/
4. /O2M2/

P(O) = 2/4 = 0.5, P(M) = 2/4 = 0.5, P(O) < 0.7, P(M) < 0.7

Alter(/M1/, *)

5. Alter(/O1M1/, *)
6. Alter(/O1M1/, /O1/)
7. Alter(/O1M1/, /O1M1/)
1. Alter(/O2M1/, /O1M1/)

P0(O) = 2/3 = 0,66 < 0.7, P0(M) = 1/3 = 0,33 < 0.7 ; P1(O) = 0.5 < 0.7, P1(M) = 0.5 < 0.7

Alter(Einkommen(*, /O1/), Einkommen(*, /O1/))

2. Alter(Einkommen(/O1/, /O1/), Einkommen(/O1/, /O1/))
3. Alter(Einkommen(/O1/, /O1M1/), Einkommen(/O1/, /O1M1/))
4. Alter(Einkommen(/O1M1/, /O1M1/), Einkommen(/O1M1/, /O1M1/))
5. Alter(Einkommen(/O1M1/, /O2M1/), Einkommen(/O1M1/, /O2M1/))

P0(O) = 0.5 < 0.7, P0(M) = 0.5 < 0.7 ; P1(O) = 2/3 = 0.66 < 0.7, P1(M) = 1/3 = 0.33 < 0.7

Alter(Einkommen(Bildung(*, *, /O1/)), Einkommen(Bildung(*, *, /O1/)))

6. Alter(Einkommen(Bildung(*, /O1/, /O1/)), Einkommen(Bildung(*, /O1/, /O1/)))

7. Alter(Einkommen(Bildung(/M1/, /O1/, /O1/)), Einkommen(Bildung(/M1/, /O1/, /O1/)))
1. Alter(Einkommen(Bildung(/O1M1/, /O1/, /O1/)), Einkommen(Bildung(/O1M1/, /O1/, /O1/)))
2. Alter(Einkommen(Bildung(/O1M1/, /O1/, /O2/)), Einkommen(Bildung(/O1M1/, /O1/, /O2/)))

P0(O) = 0.5 < 0.7, P0(M) = 0.5 < 0.7 ; P1(O) = 1 >= 0.7, P1(M) = 0 < 0.7 ; P2(O) = 2 >= 0.7, P2(M) = 0 < 0.7

Keine Differenzierung mehr möglich für Bildung(Arbitur) und gleiche Anzahl an ‚O' und ‚M', also zufälliges Auswählen: Bildung(Arbitur) = O

End: Alter(Einkommen(Bildung(O, O, O), Bildung(O, O, O)), Einkommen(Bildung(O, O, O), Bildung(O, O, O)))

ID3:

- v1 = O, v2 = M
- |v1| = 4, |v2| = 3
- p1 = 4/7 = 0.57, p2 = 3/7 = 0.43

$H(V) = -(0.57*\log_2(0.57) + 0.43*\log_2(0.43)) = 0.986$

- Alter = >=35 liefert $S_0 = \{1, 3, 4, 5\}$
- Alter = <35 liefert $S_1 = \{2, 6, 7\}$
- P(>=35) = 4/7 = 0.57, P(<35) = 3/7 = 0.43

$R(S, Alter) = 0.57 * H(\{1, 3, 4, 5\}) + 0.43 * H(\{2, 6, 7\})$

$= 0.57 * (-1/2 * \log_2(1/2) – 1/2 * \log_2(1/2)) +$

$0.43 * (-2/3 * \log_2(2/3) – 1/3 * \log_2(1/3))$

$= 0.57 + 0.395 = 0.965Bit$

- Einkommen = niedrig liefert $S_0 = \{2, 4, 7\}$
- Einkommen = hoch liefert $S_1 = \{1, 3, 5, 6\}$
- P(niedrig) = 3/7, P(hoch) = 4/7

$R(S, Einkommen) = 3/7 * H(\{2, 4, 7\}) + 4/7 * H(\{1, 3 , 5, 6\})$

$= 3/7 * (-1/3 * \log_2(1/3) - 2/3 * \log_2(2/3))$

$4/7 * (-3/4 * \log_2(3/4) – 1/4 * \log_2(1/4))$

$= 0.394 + 0.464 = 0.858$

- Bildung = Arbitur liefert $S_0 = \{1, 4, 7\}$
- Bildung = Bachelor liefert $S_1 = \{3, 6\}$
- Bildung = Master liefert $S_2 = \{2, 5\}$
- P(Arbitur) = 3/7, P(Bachelor) = 2/7, P(Master) = 2/7

R(S, Bildung) $= 3/7 * H(\{1, 4, 7\}) + 2/7 * H(\{3, 6\}) + 2/7 * H(\{2, 5\})$

$= 3/7 * (-1/3 * \log_2(1/3) – 2/3 * \log_2(2/3)) +$

$2/7 * (-1/2 * \log_2(1/2) – 1/2 * \log_2(1/2)) +$

$2/7 * (-2/2* \log_2(2/2) – 0/2 * \log_2(0/2))$

$= 0.394 + 0.286 + 0 = 0.68$

Gain(S, Alter) = H(V) – R(S, Alter) = 0.986 – 0.956 = 0.03Bit

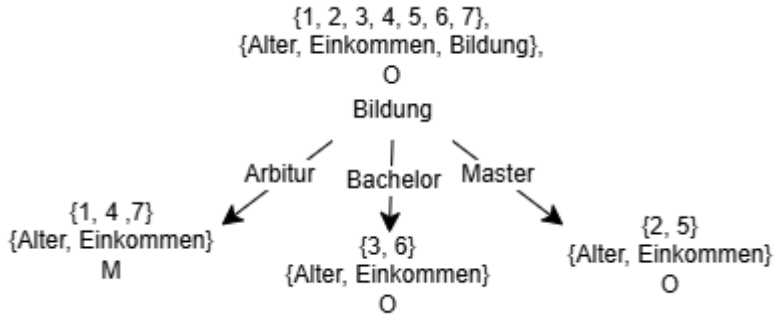
Gain(S, Einkommen) = 0.986 – 0.858 = 0.128Bit

Gain(S, Bildung) = 0.986 – 0.68 = 0.306Bit

{1, 2, 3, 4, 5, 6, 7}, {Alter, Einkommen, Bildung}, O

Bildung hat den höchsten Gain-Wert:

Teil-Baum:



{1, 4, 7}, {Alter, Einkommen}, M

H(V(Arbitur)) = $-(1/3 * \log_2(1/3) + 2/3 * \log_2(2/3)) = 0.918$

- Arbitur(Alter) = >=35 liefert $S_0 = \{1, 4\}$
- Arbitur(Alter) = <35 liefert $S_1 = \{7\}$
- P(>=35) = 2/3, P(<35) = 1/3

R(S, Arbitur(Alter) $= 2/3 * H(\{1, 4\} + 1/3 * H(\{7\}$

= 2/3 * (1) + 1/3 * 0

= 2/3 = 0.66

- Arbitur(Einkommen) = niedrig liefert S_0 = {4, 7}
- Arbitur(Einkommen) = hoch liefert S_1 = {1}
- P(niedrig) = 2/3, P(hoch) = 1/3

R(S, Arbitur(Einkommen)) = 2/3 * H({4, 7} + 1/3 * H({1})

= 2/3 * (0) + 1/3 * (0)

= 0

Gain(S, Arbitur(Alter) = 0.918 – 0.66 = 0.258

Gain(S, Arbitur(Einkommen) = 0.918 - 0 = 0.918

Arbitur(Einkommen) hat den höchsten Gain-Wert.

{3, 6}, {Alter, Einkommen} O

H(V(Bachelor)) = -(1/2 * $\log_2$(1/2) + ½ * $\log_2$(1/2)) = 1

- Bachelor(Alter) = >=35 liefert S_0 = {3}
- Bachelor(Alter) = <35 liefert S_1 = {6}
- P(>=35) = 1/2, P(<35) = 1/2

R(S, Bachelor(Alter)) = ½ * H({3} + ½ * H({6} = 0

- Bachelor(Einkommen) = niedrig liefert S_0 = {}
- Bachelor(Einkommen) = hoch liefert S_1 = {3, 6}
- P(niedrig) = 0, P(hoch) = 1

R(S, Bachelor(Einkommen) = 0 * H({}) + 1 * H({3, 6})

= 0 + 1 * (1) = 1

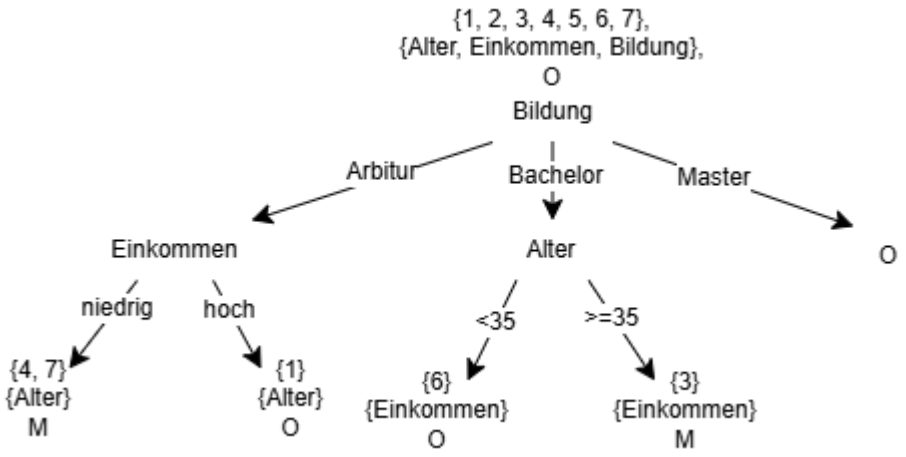
Gain(S, Bachelor(Alter)) = 1 – 0 = 1

Gain(S, Bachelor(Einkommen)) = 1 – 1 = 0

Bachelor(Alter) hat einen höheren Gain-Wert.

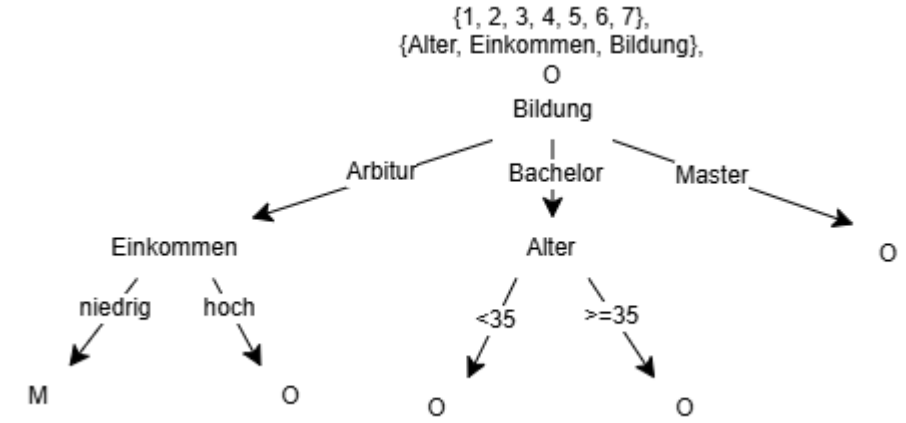
Master Pfad hat zweimal die selben Kandidaten also wird der Knoten zu ‚O‘.

Teil-Baum:



- Pfad Arbitur(niedrig) hat zwei gleiche Kandidaten somit wird der knoten zu ‚M‘
- Pfad Bachelor(<35) hat nur noch einen Vektor also wird der Knoten zu ‚O‘
- Pfad Bachelor(>=35) hat nur noch einen Vektor also wird der Knoten zu ‚M‘

Ganzer-Baum:



DTL.02:

$$x_3(x_2(x_1(C, A), x_1(B, A)), x_1(x_2(C, B), A))$$

1. Schritt: Transformationsregel:
   $x_2(x_1(C, A), x_1(B, A)) <=> x_1(x_2(C, B), x_2(A, A))$
   Baum: $x_3(x_1(x_2(C, B), x_2(A, A)), x_1(x_2(C, B), A))$

2. Schritt: bedingt irrelevante Attribute:
   $x_1(x_2(C, B), x_2(A, A))$ -> $x_1(x_2(C, B), A)$
   Baum: $x_3(x_1(x_2(C, B), A), x_1(x_2(C, B), A))$
3. Schritt: bedingt irrelevante Attribute:
   $x_3(x_1(x_2(C, B), A), x_1(x_2(C, B), A))$ -> $x_3(x_2(C, B), A))$
   fertiger Baum: $x_3(x_2(C, B), A))$

Aufgabe DT.03:

1.

Zoo:

```
J48 pruned tree
------------------

feathers <= 0
|   milk <= 0
|   |   backbone <= 0
|   |   |   airborne <= 0
|   |   |   |   predator <= 0
|   |   |   |   |   legs <= 2: shellfish (2.0)
|   |   |   |   |   legs > 2: insect (2.0)
|   |   |   |   predator > 0: shellfish (8.0)
|   |   |   airborne > 0: insect (6.0)
|   |   backbone > 0
|   |   |   fins <= 0
|   |   |   |   tail <= 0: amphibian (3.0)
|   |   |   |   tail > 0: reptile (6.0/1.0)
|   |   |   fins > 0: fish (13.0)
|   milk > 0: mammal (41.0)
feathers > 0: bird (20.0)
```

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 41  0  0  0  0  0  0 |  a = mammal
  0 13  0  0  0  0  0 |  b = fish
  0  0 20  0  0  0  0 |  c = bird
  0  0  0 10  0  0  0 |  d = shellfish
  0  0  0  0  8  0  0 |  e = insect
  0  0  0  0  0  3  1 |  f = amphibian
  0  0  0  0  0  0  5 |  g = reptile
```

Erklärung: 20 Tiere wurden richtig als Vögel klassifiziert, 41 als Säugetiere, 13 als Fische, 6 als Reptile, 3 als Amphibien, 8 als Insekten, 10 als Shellfish.

Ein Tier wurde falsch als Reptil klassifiziert.

Die Fehlerrate liegt bei 0.9901%

Durch die Confusion Matrix sieht man das alle Tiere außer einem richtig klassifiziert wurde.

Restaurant:

```
=== Confusion Matrix ===

 a b c d e f g   <-- classified as
 4 0 0 0 0 0 0 | a =    Yes
 0 0 1 0 0 0 0 | b =   No
 0 0 2 0 0 0 0 | c =   Yes
 0 0 0 2 0 0 0 | d =     No
 0 0 0 0 1 0 0 | e =    No
 0 0 1 0 0 0 0 | f =   No
 0 0 0 0 1 0 0 | g =    No
```

```
J48 pruned tree
------------------

patrons =  Some:    Yes (3.0)
patrons =  Full:   Yes (4.0/2.0)
patrons =   Some:    Yes (1.0)
patrons =   Full:     No  (2.0)
patrons =   None:    No (2.0/1.0)
```

Erklärung: Der Baum zeigt, wann in der Regel ein Kunde warten muss. Dabei ist das wichtigste Attribut ‚patrons' was die Anzahl der Gäste ist. Wenn also nur ein paar Gäste im Restaurant dann warten Gäste in drei Trainingsbeispielen oder nur in einem. Wenn es voll ist, warten Gäste bei 6 Beispielen (wobei zwei davon falsch klassifiziert wurden) oder es wartet kein Gast in zwei Beispielen. Und wenn es leer ist, wartet keine in drei Beispielen, wo auch wieder ein Trainingsbeispiel falsch klassifiziert wurde.

Die Fehlerrate liegt bei 25%.

2.

Unterschied Nominal, Ordinal, String: Ordinal sind reals und integer, nominal sind Objekte oder Klassen und Strings sind Text-Werte.

Zoo:

```
J48 pruned tree
------------------

feathers <= 0
|   milk <= 0
|   |   backbone <= 0
|   |   |   airborne <= 0
|   |   |   |   predator <= 0
|   |   |   |   |   legs <= 2: shellfish (2.0)
|   |   |   |   |   legs > 2: insect (2.0)
|   |   |   |   predator > 0: shellfish (8.0)
|   |   |   airborne > 0: insect (6.0)
|   |   backbone > 0
|   |   |   fins <= 0
|   |   |   |   tail <= 0: amphibian (3.0)
|   |   |   |   tail > 0: reptile (6.0/1.0)
|   |   |   fins > 0: fish (13.0)
|   milk > 0: mammal (41.0)
feathers > 0: bird (20.0)
```

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 41  0  0  0  0  0  0 |  a = mammal
  0 13  0  0  0  0  0 |  b = fish
  0  0 20  0  0  0  0 |  c = bird
  0  0  0 10  0  0  0 |  d = shellfish
  0  0  0  0  8  0  0 |  e = insect
  0  0  0  0  0  3  1 |  f = amphibian
  0  0  0  0  0  0  5 |  g = reptile
```

Fehlerrate = 0.9901%

Fazit: Gleicher Baum und selbe Fehlerrate wie bei dem ersten Teil.

Restaurant:

```
J48 pruned tree
------------------

patrons =  Some:    Yes (3.0)
patrons =  Full:   Yes (4.0/2.0)
patrons =   Some:    Yes (1.0)
patrons =   Full:     No  (2.0)
patrons =   None:    No (2.0/1.0)
```

Fehlerrate = 25%

```
=== Confusion Matrix ===

 a b c d e f g   <-- classified as
 4 0 0 0 0 0 0 | a =    Yes
 0 0 1 0 0 0 0 | b =   No
 0 0 2 0 0 0 0 | c =   Yes
 0 0 0 2 0 0 0 | d =     No
 0 0 0 0 1 0 0 | e =    No
 0 0 1 0 0 0 0 | f =   No
 0 0 0 0 1 0 0 | g =    No
```

Fazit: Gleicher Baum und selbe Fehlerrate wie bei dem ersten Teil.

```
Id3


est =   0-10
|  patrons =  Some:    Yes
|  patrons =  Full: null
|  patrons =   Some:    Yes
|  patrons =   Full: null
|  patrons =   None:    No
est =     30-60:   No
est =     10-30:   Yes
est =   >60:     No
est =  0-10:    Yes
est =     0-10
|  hun =   Yes:    Yes
|  hun =   No:    No
|  hun =  Yes: null
|  hun =  No: null
est =  10-30:   No
est =   30-60:   Yes
```

Fehlerrate = 0%

```
=== Confusion Matrix ===

 a b c d e f g   <-- classified as
 4 0 0 0 0 0 0 | a =    Yes
 0 1 0 0 0 0 0 | b =   No
 0 0 2 0 0 0 0 | c =   Yes
 0 0 0 2 0 0 0 | d =     No
 0 0 0 0 1 0 0 | e =    No
 0 0 0 0 0 1 0 | f =   No
 0 0 0 0 0 0 1 | g =    No
```