11_DSI_Carsten_Brauer

Mittwoch, 3. Juni 2020 14:29

Task A)

Age	Income	Student	Credit-Rat	Buys_com
<=30	High	No	Excellent	No
<=30	Low	No	Excellent	No
<=30	Medium	No	Excellent	No
<=30	High	No	Fair	No
<=30	Low	No	Fair	No
<=30	Medium	No	Fair	No
>40	Medium	No	Excellent	No
>40	Low	Yes	Excellent	No
<=30	Medium	Yes	Excellent	Yes
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Excellent	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
31...40	High	No	Excellent	Yes
31...40	Medium	No	Excellent	Yes
31...40	Low	Yes	Excellent	Yes
31...40	High	No	Fair	Yes
31...40	High	Yes	Fair	Yes
31...40	Low	Yes	Fair	Yes

Einflussgrößen x_i: Age, Income, Student, Credit-Rating
Gesucht: y (Buys_computer y/n)

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

$$E(Buys - computer) = E(8;12) = E(0{,}4;0{,}6) = -(0{,}4 * \log_2(0{,}4)) - (0{,}6 * \log_2(0{,}6)) = 0{,}97$$

Wir teilen den Datensatz nach Attributen und rechnen E(Buys_computer, x_i) aus:

		Buys Computer		Entropy	Proportional
		yes	no		
Age	<=30	2	6	0,81127812	0,32451125
	31...40	6	0	0	0
	>40	4	2	0,91829583	0,27548875
Gain = E(Buys_computer) - E(Buys_computer, Age)			0,37	Summe:	0,6

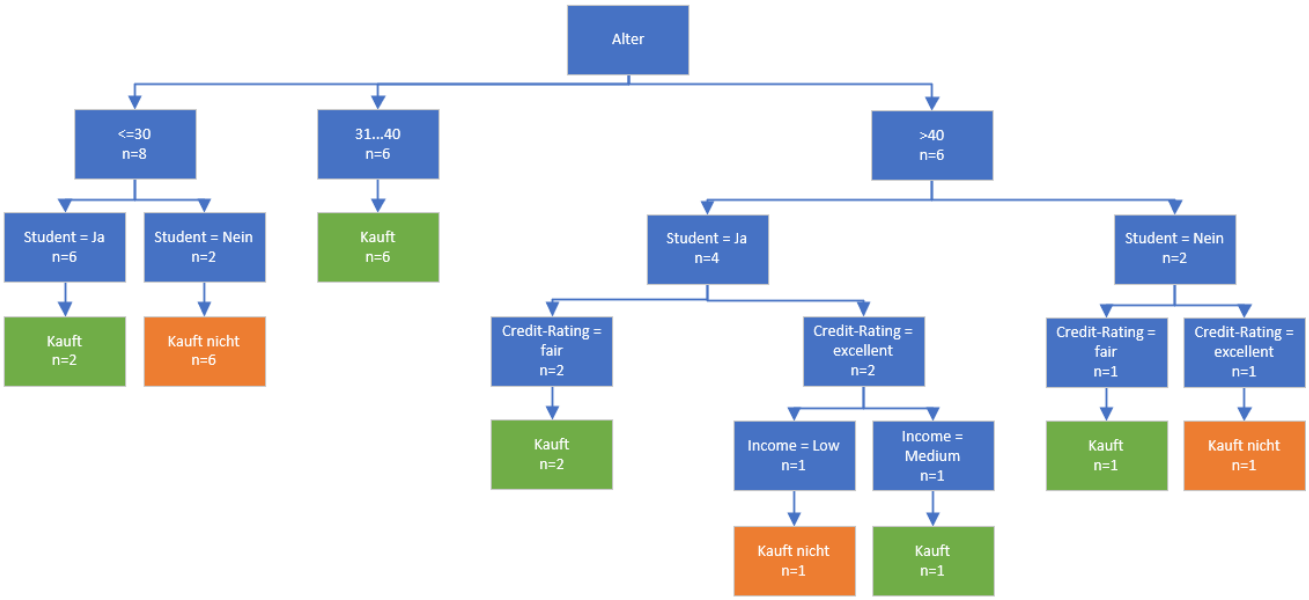
		Buys Computer		Entropy	Proportional
		yes	no		
Income	Low	4	3	0,98522814	0,344829848
	Medium	5	3	0,954434	0,381773601
	High	3	2	0,97095059	0,242737649

Income	Medium	5	3	0,954434	0,381773601
	High	3	2	0,97095059	0,242737649
Gain = E(Buys_computer) - E(Buys_computer, Income)			0,0006589	Summe:	0,969341097

		Buys Computer		Entropy	Proportional
		yes	no		
Student	Yes	8	1	0,50325833	0,226466251
	No	4	7	0,9456603	0,520113168
Gain = E(Buys_computer) - E(Buys_computer, Student)			0,22342058	Summe:	0,746579418

		Buys Computer		Entropy	Proportional
		yes	no		
Credit-Rating	Fair	7	3	0,8812909	0,44064545
	Excellent	5	5	1	0,5
Gain = E(Buys_computer) - E(Buys_computer, Credit-Rating)			0,02935455	Summe:	0,94064545

Unser erstes Kriterium ist also das Alter, dann Student, dann Credit-Rating, dann Income:



Task B)

```
import numpy as np
import pandas as pd
from sklearn.tree import export_graphviz
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from IPython.display import Image

df = pd.read_csv('computer_purchase_data.csv')
```

```
lenc = LabelEncoder()
lenc.fit(['<=30', '31...40', '>40', 'High', 'Medium', 'Low', 'Fair', 'Excellent', 'Yes', 'No'])
raw_values = df.values.reshape(-1, 1)
encoded_values = lenc.transform(raw_values).reshape(-1, 5)

X = encoded_values[:, 0:4]
y = encoded_values[:, 4:5]

tree_classifier = DecisionTreeClassifier(max_depth=10)
tree_classifier.fit(X, y)

export_graphviz(
    tree_classifier,
    out_file="computer_purchase_decision_tree.dot",
    feature_names=df.columns.values[0:4].tolist(),
    class_names=["Buying", "Not Buying"],
    #rounded=True,
    filled=True
)
```

Ergibt: