Случайный лес (Random forest)

Ставровская Елена

2016

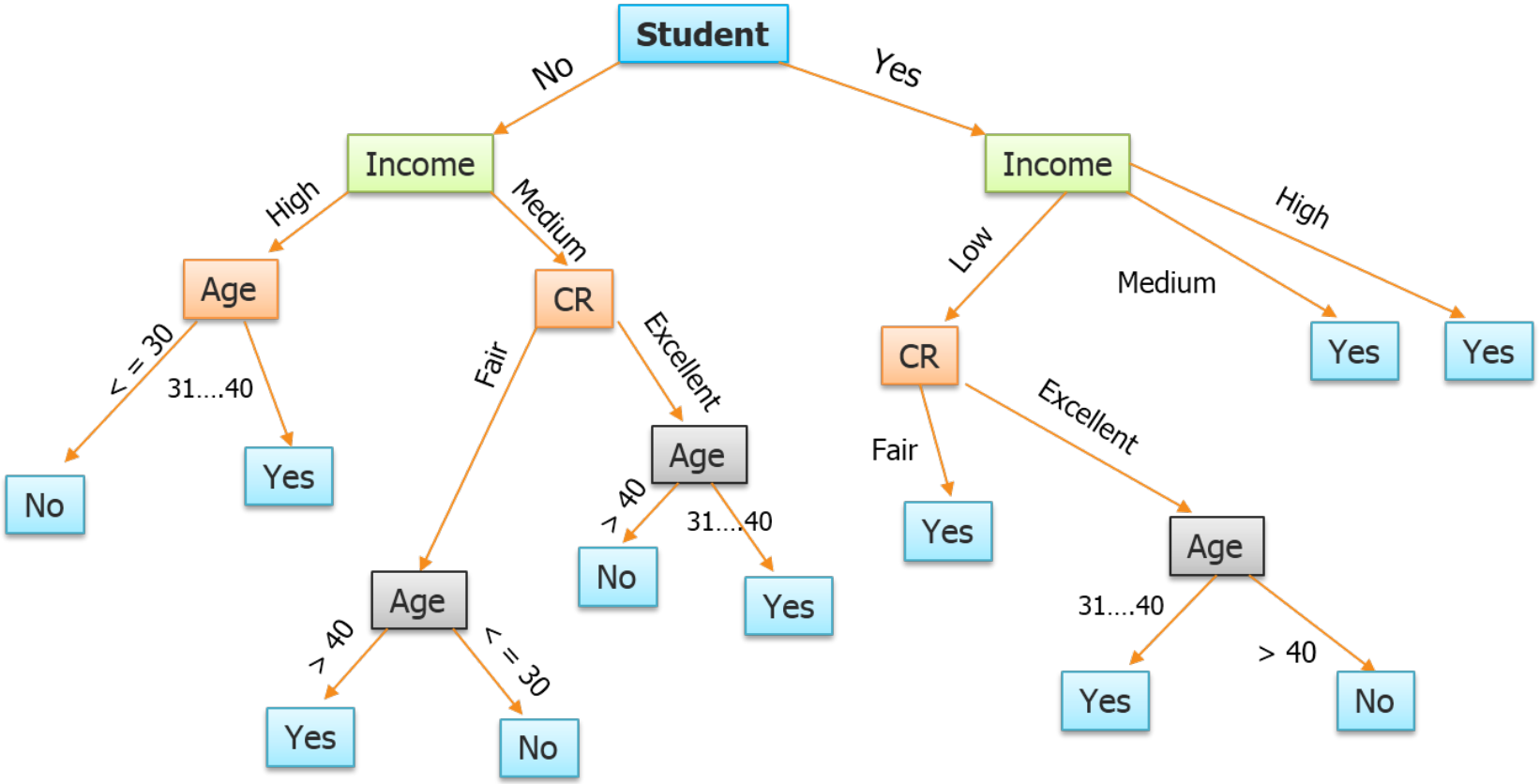
Классификатор

Покупает ли студент компьютер?

rec	Age	Income	Student	Credit_rating	Buys_computer
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31…40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<-=30	Medium	Yes	Excellent	Yes
r12	31…40	Medium	No	Excellent	Yes
r13	31…40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

Дерево принятия решений

- Узел – один из признаков
- Ребро – значение признака
- Лист – значение целевой функции(класс)



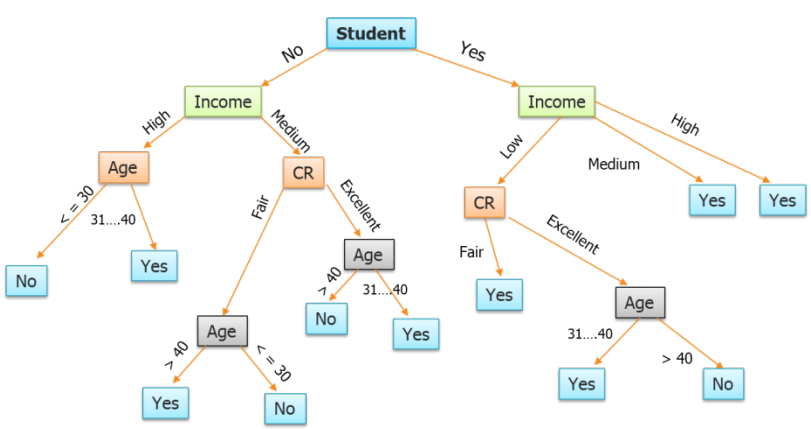
Дерево принятия решений

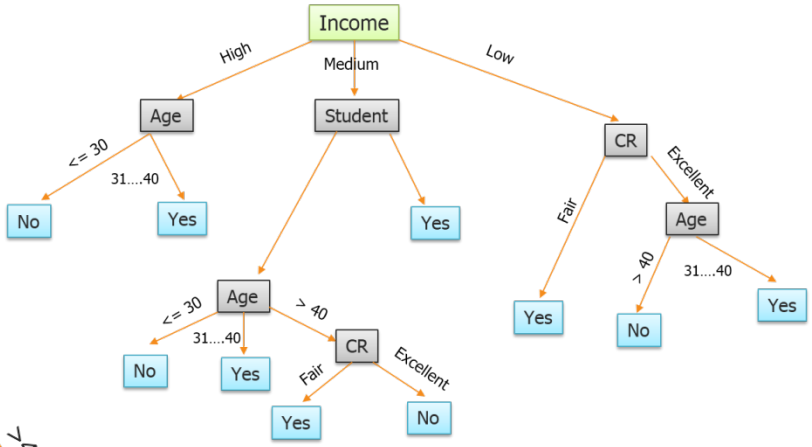
Построение:

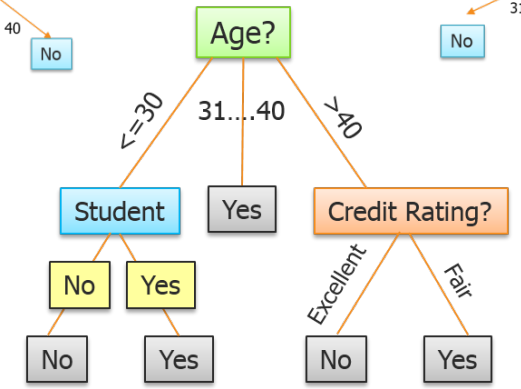
- Выбираем очередной признак *P*, помещаем его в корень.
- Для всех его значений *v*:
  - Оставляем из тестовых примеров только те, у которых значение атрибута *P* равно *v*
  - Рекурсивно строим дерево в этом потомке

Дерево принятия решений

В каком порядке выбирать признаки?







Идея: на каждом шаге выбирать признак, по значениям которого можно лучше разбить данные на классы (например, минимум энтропии)

Information Gain: Attribute Selection Measure

edureka!

- → This measure is used in algorithms ID3 and C4.5
- → Heuristic: Select the attribute with the highest information gain i.e., attribute that results in most homogeneous branches
- → Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- → Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- → Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- → Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection Example

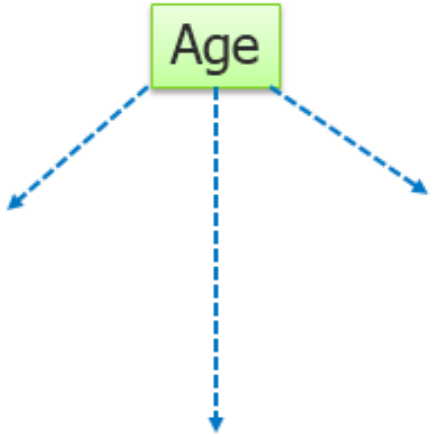
edureka!

$Gain(Age) = 0.246$

$Gain(income) = 0.029$

$Gain(student) = 0.151$

$Gain(credit\_rating) = 0.048$



Age	Income	Student	Credit Rating	Buys Computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes

Age	Income	Student	Credit Rating	Buys Computer
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
>40	Medium	Yes	Fair	Yes
>40	High	No	Excellent	No

Age	Income	Student	Credit Rating	Buys Computer
31...40	High	No	Fair	Yes
31...40	Low	Yes	Excellent	Yes
31...40	Medium	No	Excellent	Yes
31...40	High	Yes	Fair	Yes

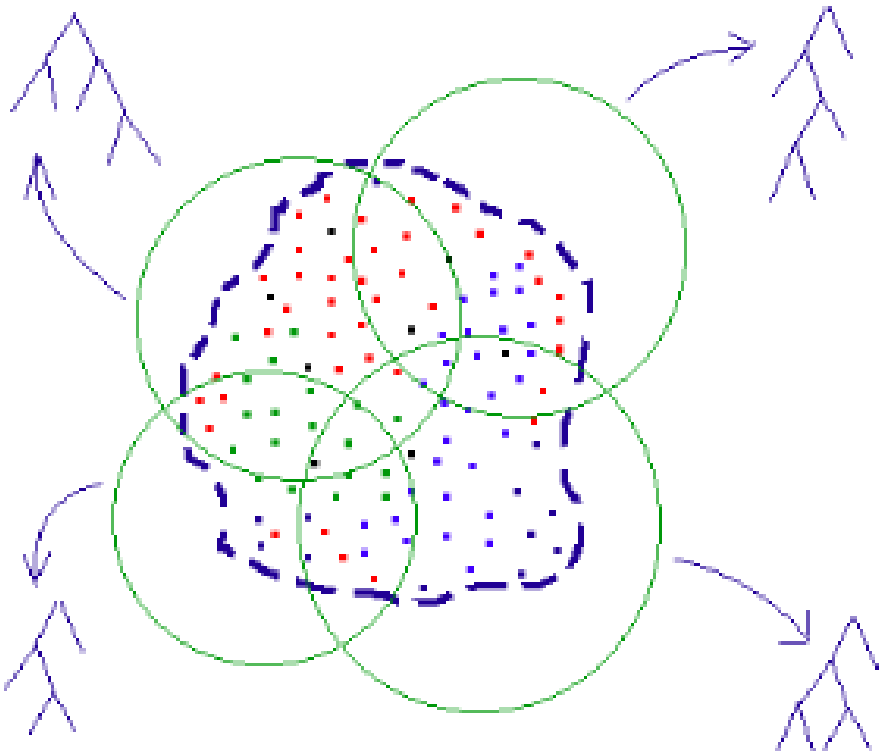
Дерево принятия решений

Проблема:

Жадный алгоритм, результат зависит от порядка выбора признаков

Случайный лес (Random Forest)

- Создаем лес случайных решающих деревьев:
  - Выбираем из обучающей выборки N объектов **с повторением**
  - Выбираем некоторое случайное подмножество признаков
  - Строим дерево решений



Случайный лес (Random Forest)

- Классификация объектов проводится путём голосования: каждое дерево леса относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев.

Random Forest в R

```
library(randomForest)
model = randomForest(Type ~ ., train)
#важность переменных для классификатора
importance(model)
#вытащим одно из деревьев леса
getTree(model, k=10, labelVar=T)
```