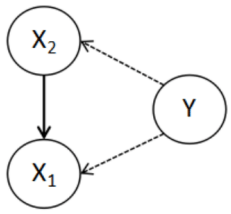
Week 6 Lab Solutions

Question 1 - Bayesian network classifier

Given a training dataset including two predictors (i.e. X1 and X2) and one class attribute (i.e. Y), please create the conditional probability tables (CPTs) according to the predictor dependencies shown in the Bayesian network, then predict the class label of the testing instances(observations).



Training dataset

	X_1	X_2	Y
Instance_1	1	0	1
Instance_2	0	0	0
Instance_3	1	0	0
Instance_4	0	1	0
Instance_5	0	0	1
Instance_6	1	1	0
Instance_7	0	1	1
Instance_8	1	1	1
Instance_9	0	0	1
Instance_10	1	1	0

Testing dataset

	X_1	X_2	Y
Instance_11	1	1	?
Instance_12	0	0	?

Solution:

-Step 1:

Estimate the parameters of Bayesian network classifier, and create the conditional probability tables for X1, X2 and Y.

The conditional probability tables for X_1, X_2 and Y

		X_1=1	X_1=0
Y=1	*X_2=1*	*0.500*	*0.500*
Y=1	*X_2=0*	*0.333*	*0.667*
Y=0	*X_2=1*	*0.667*	*0.333*
Y=0	*X_2=0*	*0.500*	*0.500*

	X_2=1	X_2=0
Y=1	*0.400*	*0.600*
Y=0	*0.600*	*0.400*

Y=1	Y=0
0.500	*0.500*

-Step 2:

For predicting the class label of testing instance_11, according to the Bayesian rule and the created CPTs, we can obtain

$$
\begin{aligned}
&\mathrm{P}(\mathrm{Y}=1 \mid \mathrm{X1}=1, \mathrm{X2}=1) \\
&\propto \mathrm{P}(\mathrm{Y}=1, \mathrm{X1}=1, \mathrm{X2}=1) \\
&= \mathrm{P}(\mathrm{X1}=1 \mid \mathrm{X2}=1, \mathrm{Y}=1) * \mathrm{P}(\mathrm{X2}=1 \mid \mathrm{Y}=1) * \mathrm{P}(\mathrm{Y}=1)
\end{aligned}
$$

$= 0.500 * 0.400 * 0.500$
$= 0.100$

$P(Y = 0 \mid X1 = 1, X2 = 1)$
$\propto P(Y = 0, X1 = 1, X2 = 1)$
$= P(X1 = 1 \mid X2 = 1, Y = 0) * P(X2 = 1 \mid Y = 0) * P(Y = 0)$
$= 0.667 * 0.600 * 0.500$
$= 0.200$

The predicted class label is 0, because $P(Y = 0 \mid X1 = 1, X2 = 1)$ is greater than $P(Y = 1 \mid X1 = 1, X2 = 1)$.

-For predicting the class label of testing instance_12, according to the Bayesian rule and the created CPTs, we can obtain

$P(Y = 1 \mid X1 = 0, X2 = 0)$
$\propto P(Y = 1, X1 = 0, X2 = 0)$
$= P(X1 = 0 \mid X2 = 0, Y = 1) * P(X2 = 0 \mid Y = 1) * P(Y = 1)$
$= 0.667 * 0.600 * 0.500$
$= 0.200$

$P(Y = 0 \mid X1 = 0, X2 = 0)$
$\propto P(Y = 0, X1 = 0, X2 = 0)$
$= P(X1 = 0 \mid X2 = 0, Y = 0) * P(X2 = 0 \mid Y = 0) * P(Y = 0)$
$= 0.500 * 0.400 * 0.500$
$= 0.100$

The predicted class label is 1, because $P(Y = 1 \mid X1 = 0, X2 = 0)$ is greater than $P(Y = 0 \mid X1 = 0, X2 = 0)$.

Question 2 - Classification tree

1) Survived is a numeric value. We need to first transform it to a categorical value. Use titanic3$survived = as.factor(titanic3$survived) to do so.

```
library(readr)
library(dplyr)
library(tree)
titanic3 <- "https://goo.gl/At238b" %>%
read_csv %>% # read in the data
select(survived, embarked, sex, sibsp, parch, fare) %>%
mutate(embarked = factor(embarked), sex = factor(sex))
titanic3$survived <- as.factor(titanic3$survived)
```

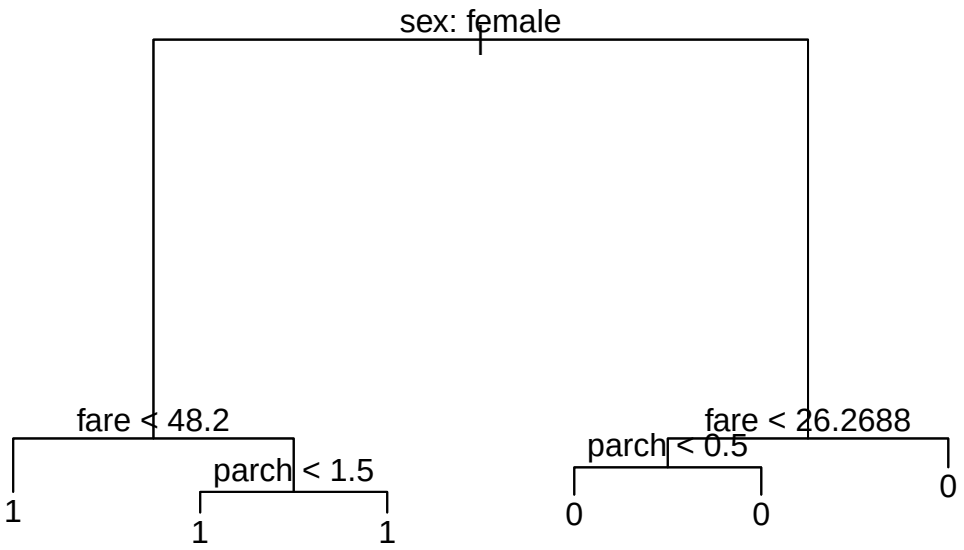
2) Fit a classification tree using all the instances. Find out which predictors actually contribute to building this tree. Plot the tree.

```
tree.titanic3 <- tree(survived ~ embarked+sex+sibsp+parch+fare, titanic3)
summary(tree.titanic3)
```

```
##
## Classification tree:
## tree(formula = survived ~ embarked + sex + sibsp + parch + fare,
##     data = titanic3)
## Variables actually used in tree construction:
## [1] "sex"   "fare"  "parch"
## Number of terminal nodes:  6
## Residual mean deviance:  0.9582 = 1246 / 1300
```

```
## Misclassification error rate: 0.2205 = 288 / 1306
```

```
plot(tree.titanic3)
text(tree.titanic3,pretty=0)
```



Predictors actually used in tree construction: sex, fare and parch.

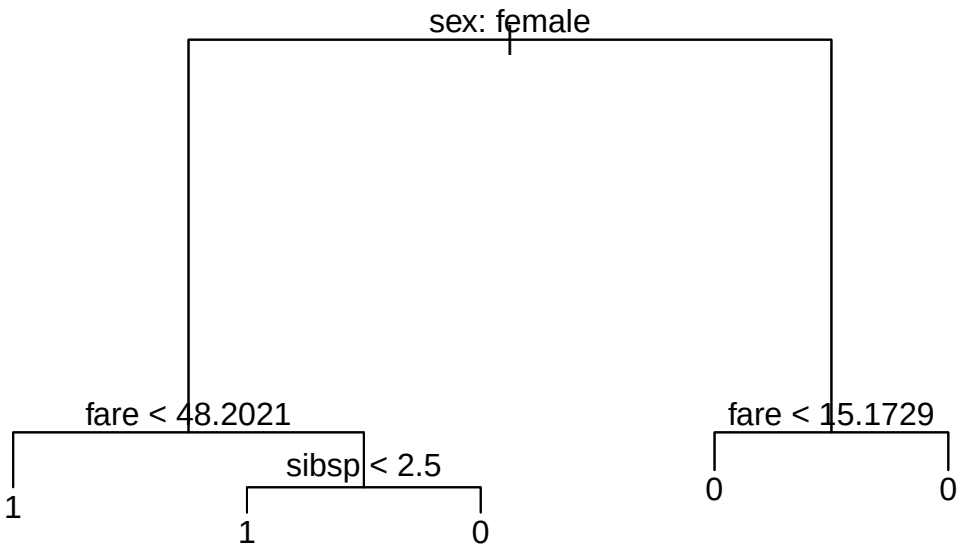
3) Now we are going to estimate the testing error:

a. Split the instances into a training dataset and a testing dataset (with set.seed(2)).

```
set.seed(2)
train <- sample(1:nrow(titanic3), nrow(titanic3)/2)
titanic3.test <- titanic3[-train,]
survived.test <- titanic3$survived[-train]
```

b. Build the tree using the training dataset, and plot the tree.

```
tree.titanic3.train <- tree(survived ~ embarked+sex+sibsp+parch+fare, titanic3, subset=train)
plot(tree.titanic3.train)
text(tree.titanic3.train,pretty=0)
```



c. Evaluate its performance on the testing dataset.

```
tree.titanic3.pred <- predict(tree.titanic3.train, titanic3.test, type="class")
mean(tree.titanic3.pred!= survived.test)
```

```
## [1] 0.2122137
```

Alternatively, use

```
table(tree.titanic3.pred, survived.test)
```

```
##                    survived.test
## tree.titanic3.pred   0   1
##                  0 349  79
##                  1  60 167
```

4) Next, let's find out whether pruning the tree might lead to improved results.

a. Use cv.tree() to determine the optimal level of tree complexity (with set.seed(21)).

```
set.seed(21)
cv.titanic3 <- cv.tree(tree.titanic3.train, FUN=prune.misclass)
print(cv.titanic3)
```

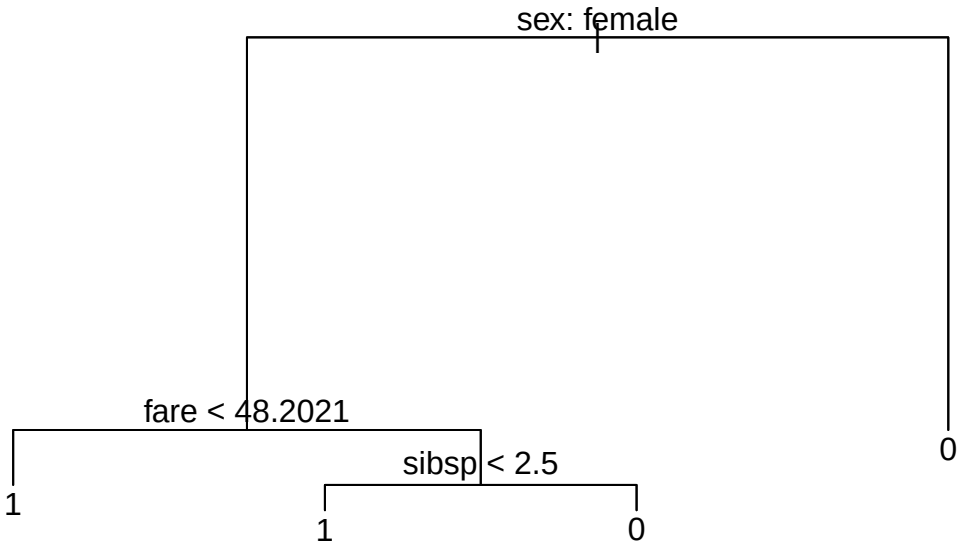
```
## $size
## [1] 5 4 2 1
##
## $dev
## [1] 152 152 154 253
##
## $k
## [1]  -Inf   0.0   0.5 104.0
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

b. According to the result, do you think pruning is necessary? Why or why not?

The results show that the best tree has 5 or 4 leaves. There is no need to prune. But we can try to prune the tree to 4 leaves.

c. If you think it is necessary, or would like to give it a try, use prune.misclass() to prune the tree and evaluate the performance of the pruned tree.

```
prune.titanic3 <- prune.misclass(tree.titanic3.train, best=4)
plot(prune.titanic3)
text(prune.titanic3, pretty=0)
```




```
tree.prune.titanic3.pred <- predict(prune.titanic3, titanic3.test, type="class")
mean(tree.prune.titanic3.pred!= survived.test)
```

```
## [1] 0.2122137
```

This error rate is the same as the tree with 5 leaves (in my case, the tree is tree.titanic3.train). However, considering the interpretability, the tree with 4 leaves is better. You might have different results as mine if you set different seeds. Any reasonable answers are acceptable.