Classfication

Jimmy Harvin, Yixin Sun,

10-8-2022

load the data

```
adult <- read.csv("C:/Users/Yixin Sun/Documents/Assignment3/adult.csv", header = T)
str(adult)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : chr  " State-gov" " Self-emp-not-inc" " Private" " Private" ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449
...
##  $ education     : chr  " Bachelors" " Bachelors" " HS-grad" " 11th" ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status: chr  " Never-married" " Married-civ-spouse" " Divorced" " Married-civ-spou
se" ...
##  $ occupation    : chr  " Adm-clerical" " Exec-managerial" " Handlers-cleaners" " Handlers-cl
eaners" ...
##  $ relationship  : chr  " Not-in-family" " Husband" " Not-in-family" " Husband" ...
##  $ race          : chr  " White" " White" " White" " Black" ...
##  $ sex           : chr  " Male" " Male" " Male" " Male" ...
##  $ capital.gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hoursperweek  : int  40 13 40 40 40 40 16 45 50 40 ...
##  $ nativecountry : chr  " United-States" " United-States" " United-States" " United-States"
...
##  $ salary        : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

logistic regression parts

data cleaning and divide into train and test

```
adult <- adult[,c(1,5,10,13,15)]
adult1 <- adult
adult2 <- adult
adult3 <- adult
str(adult1)
```

```
## 'data.frame':    32561 obs. of  5 variables:
##  $ age          : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ education.num: int  13 13 9 7 13 14 5 9 14 13 ...
##  $ sex          : chr  " Male" " Male" " Male" " Male" ...
##  $ hoursperweek : int  40 13 40 40 40 40 16 45 50 40 ...
##  $ salary       : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

```
set.seed(8)
i <- sample(1:nrow(adult1), 0.8*nrow(adult1), replace = F)
train <- adult1[i,]
test <- adult1[-i,]
str(train)
```

```
## 'data.frame':    26048 obs. of  5 variables:
##  $ age          : int  27 21 34 67 20 41 37 25 50 32 ...
##  $ education.num: int  9 10 9 9 10 13 9 13 13 13 ...
##  $ sex          : chr  " Male" " Male" " Female" " Female" ...
##  $ hoursperweek : int  45 10 35 20 14 70 46 40 40 8 ...
##  $ salary       : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

data exploration and informative graphs

```
names(train)
```

```
## [1] "age"           "education.num" "sex"           "hoursperweek"
## [5] "salary"
```

```
dim(train)
```

```
## [1] 26048     5
```

```
summary(train)
```

```
##       age         education.num       sex             hoursperweek  
##  Min.   :17.00   Min.   : 1.00   Length:26048       Min.   : 1.00  
##  1st Qu.:28.00   1st Qu.: 9.00   Class :character   1st Qu.:40.00  
##  Median :37.00   Median :10.00   Mode  :character   Median :40.00  
##  Mean   :38.56   Mean   :10.09                      Mean   :40.45  
##  3rd Qu.:48.00   3rd Qu.:12.00                      3rd Qu.:45.00  
##  Max.   :90.00   Max.   :16.00                      Max.   :99.00  
##     salary         
##  Length:26048      
##  Class :character  
##  Mode  :character  
##                    
##                    
## 
```

```
str(train)
```

```
## 'data.frame':    26048 obs. of  5 variables:
##  $ age          : int  27 21 34 67 20 41 37 25 50 32 ...
##  $ education.num: int  9 10 9 9 10 13 9 13 13 13 ...
##  $ sex          : chr  " Male" " Male" " Female" " Female" ...
##  $ hoursperweek : int  45 10 35 20 14 70 46 40 40 8 ...
##  $ salary       : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

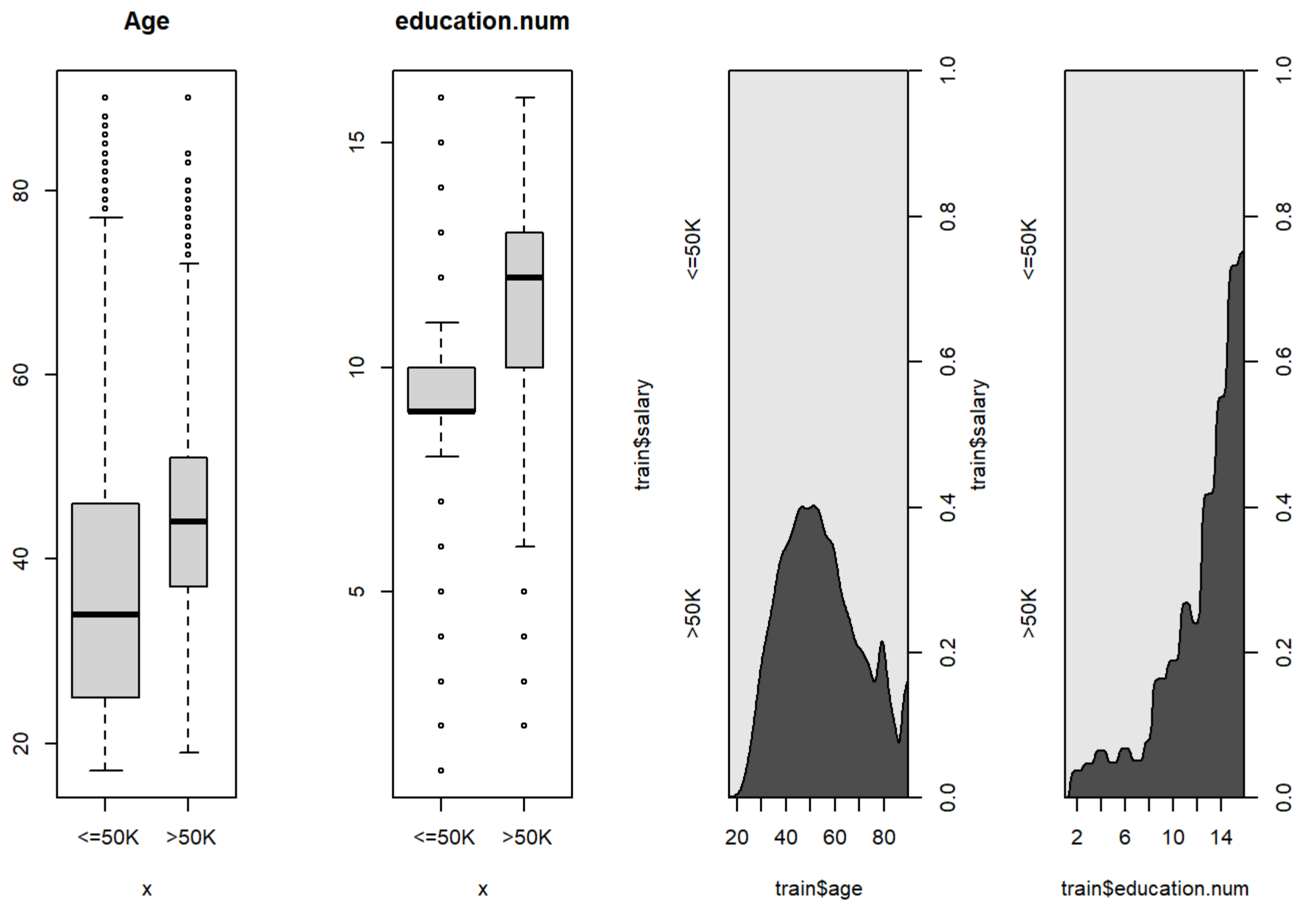
```
head(train, n = 15)
```

	age <int>	education.num <int>	sex <chr>	hoursperweek <int>	salary <chr>
30560	27	9	Male	45	<=50K
13620	21	10	Male	10	<=50K
19639	34	9	Female	35	<=50K
9954	67	9	Female	20	<=50K
21071	20	10	Male	14	<=50K
14348	41	13	Female	70	<=50K
19063	37	9	Male	46	>50K
28330	25	13	Male	40	>50K
17639	50	13	Male	40	>50K
9470	32	13	Female	8	>50K

1-10 of 15 rows Previous **1** 2 Next

```
train$sex <- as.factor(train$sex)
train$salary <- as.factor(train$salary)

par(mfrow=c(1,4))
plot(train$salary, train$age, main="Age", ylab="", varwidth=TRUE)
plot(train$salary, train$education.num, main="education.num", ylab="", varwidth=TRUE)
cdplot(train$salary~train$age)
cdplot(train$salary~train$education.num)
```

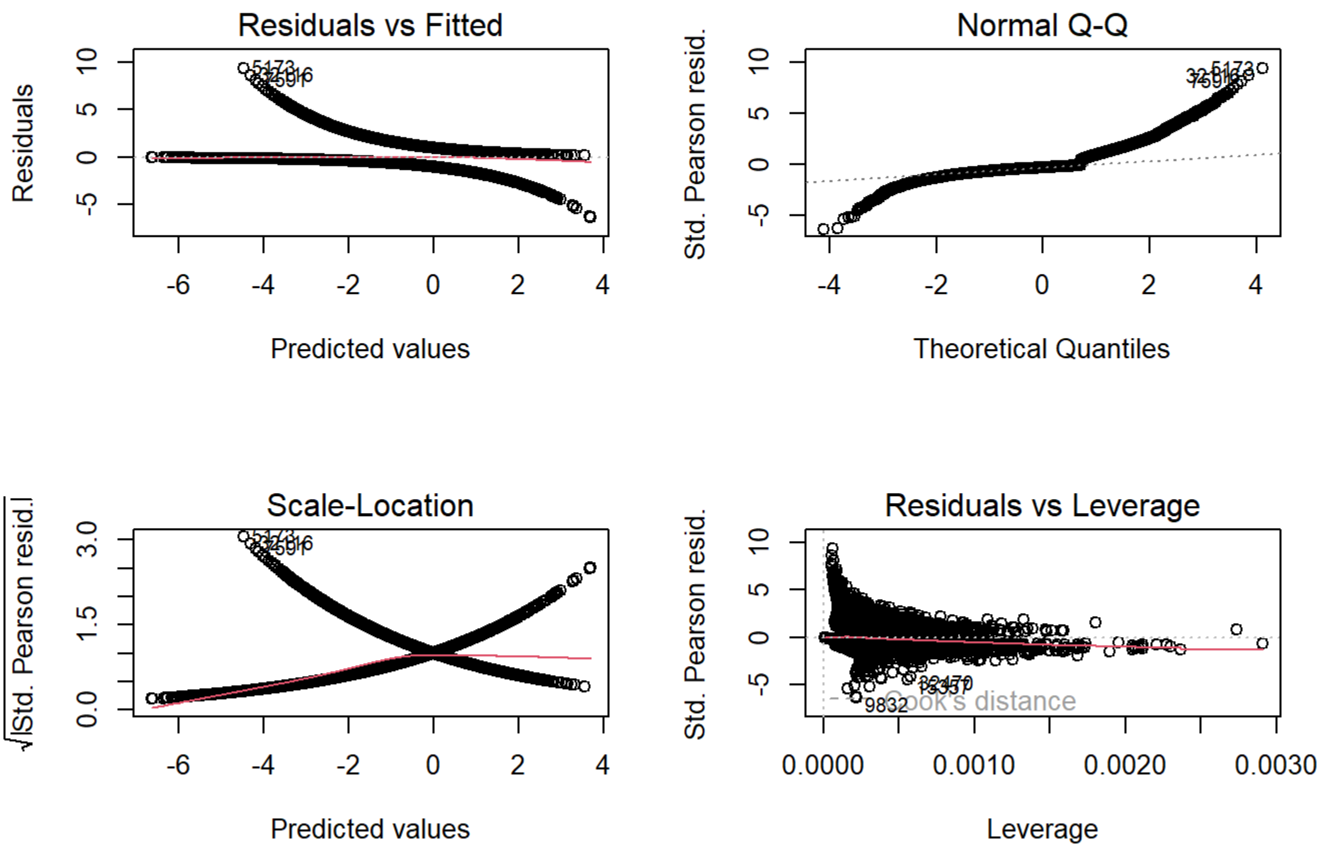


Build logistic regression model

```
glm1 <- glm(salary~., data=train, family="binomial")
summary(glm1)
```

```
##
## Call:
## glm(formula = salary ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7265  -0.6708  -0.4087  -0.1043   2.9950
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -9.219160   0.130206  -70.81   <2e-16 ***
## age            0.046302   0.001328   34.87   <2e-16 ***
## education.num  0.354422   0.007408   47.84   <2e-16 ***
## sex Male       1.181794   0.042127   28.05   <2e-16 ***
## hoursperweek   0.037027   0.001455   25.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28850  on 26047  degrees of freedom
## Residual deviance: 22319  on 26043  degrees of freedom
## AIC: 22329
##
## Number of Fisher Scoring iterations: 5
```

```
par(mfrow=c(2,2))
plot(glm1)
```




```
confint(glm1)
```

```
## Waiting for profiling to be done...
```

```
##                      2.5 %      97.5 %
## (Intercept)   -9.47605375 -8.96563302
## age            0.04370552  0.04891039
## education.num  0.33996535  0.36900649
## sex Male       1.09964523  1.26479688
## hoursperweek   0.03418336  0.03988869
```

predict and evaluate on the test data for logistic regression model

```
probs <- predict(glm1, newdata=test, type="response")
test$salary <- as.factor(test$salary)
pred <- ifelse(probs>0.5, 1, 0)
acc <- mean(pred==test$salary)
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.803009365883617"
```

```
# confusion matrix
tb <- table(pred, test$salary)
tb
```

```
##
## pred  <=50K  >50K
##    0   4635   933
##    1    350   595
```

```
# confusion matrix
library(caret)
```

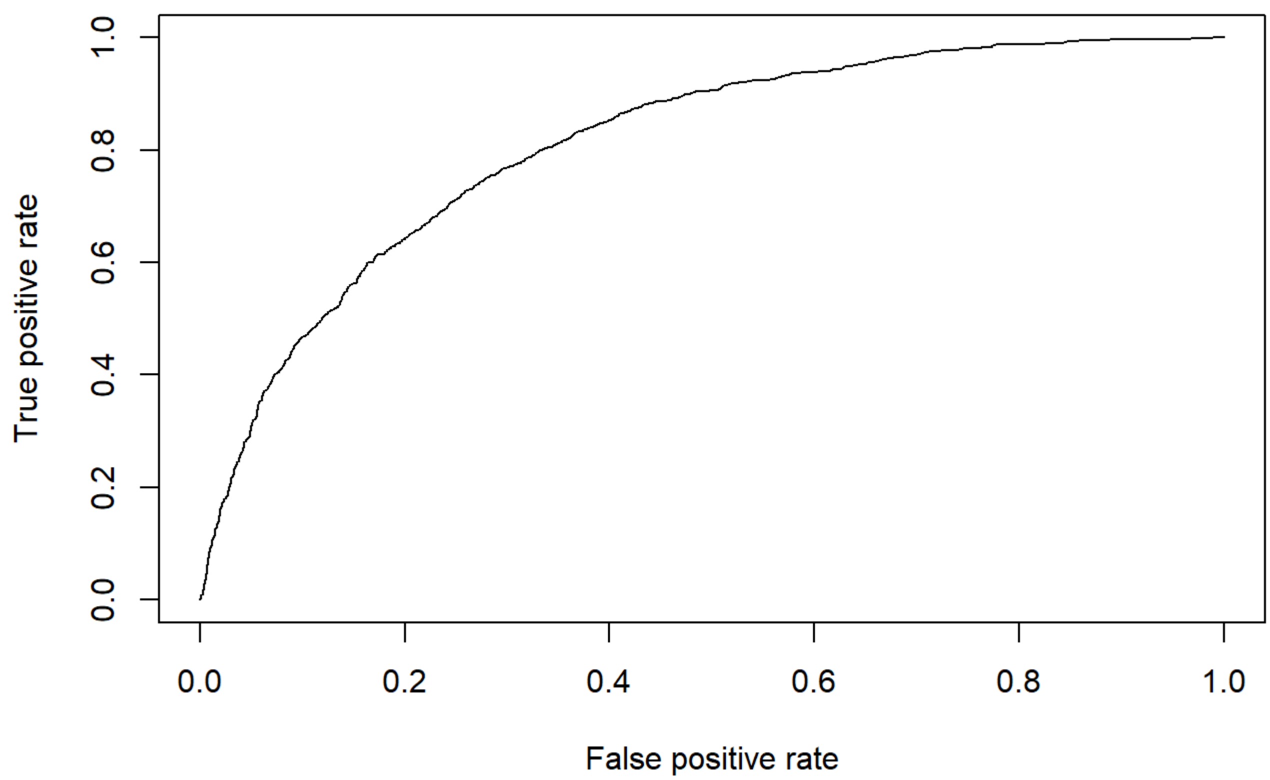
```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
confusionMatrix(as.factor(pred), reference=test$salary)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   <=50K >50K
##          0    4635 933
##          1     350 595
##
##                Accuracy : 0.8030
##                  95% CI : (0.7153, 0.8121)
##     No Information Rate : 0.6329
##     P-Value [Acc > NIR] : 2e-16
##
##                   Kappa : 0.5324
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8269
##             Specificity : 0.6460
##          Pos Pred Value : 0.8259
##          Neg Pred Value : 0.7105
##              Prevalence : 0.6389
##          Detection Rate : 0.5100
##    Detection Prevalence : 0.6239
##       Balanced Accuracy : 0.7319
##
##        'Positive' Class : 0
##
```

```
# Roc
library(ROCR)
p <- predict(glm1, newdata=test, type="response")
pr <- prediction(p, test$salary)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
# AUC
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8105806
```

kNN parts

data divide

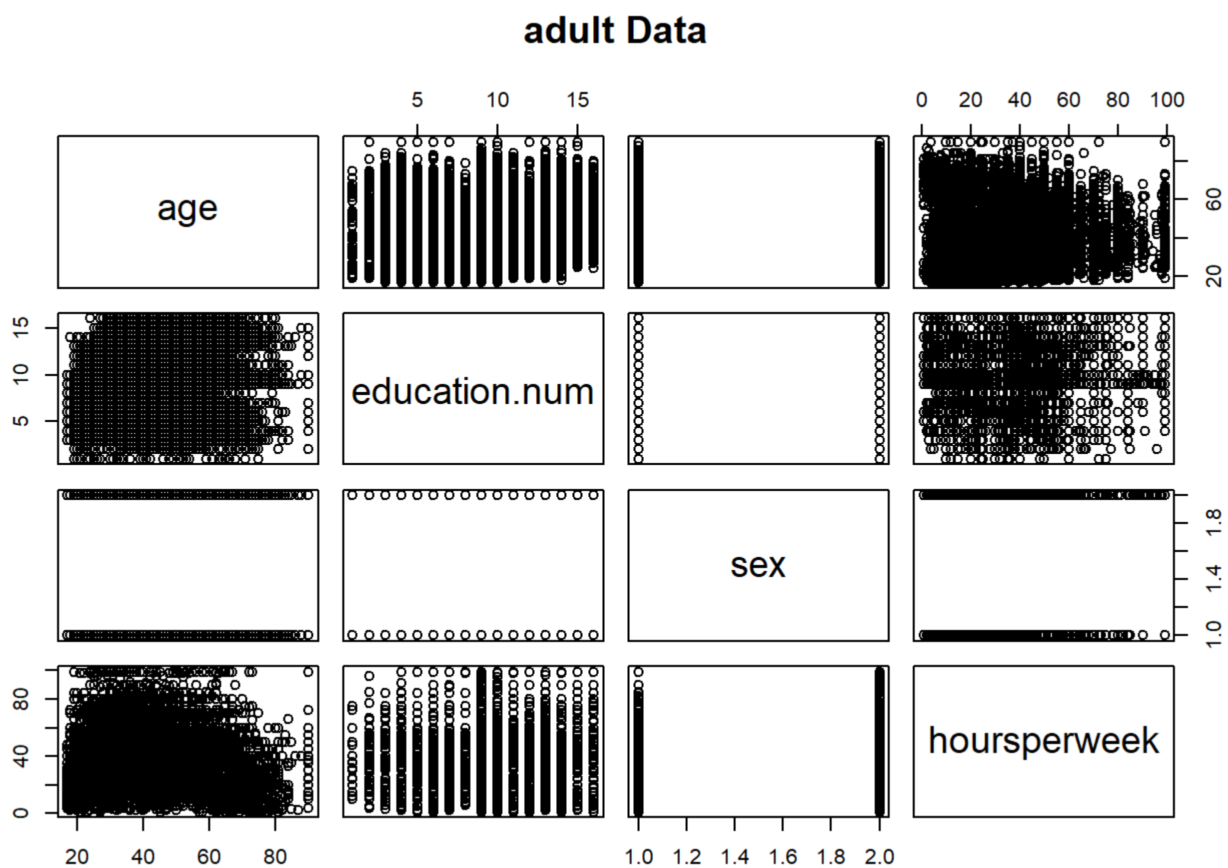
```
summary(adult2)
```

```
##       age        education.num       sex              hoursperweek
##  Min.   :17.00   Min.   : 1.00   Length:32561       Min.   : 1.00
##  1st Qu.:28.00   1st Qu.: 9.00   Class :character   1st Qu.:40.00
##  Median :37.00   Median :10.00   Mode  :character   Median :40.00
##  Mean   :38.58   Mean   :10.08                      Mean   :40.44
##  3rd Qu.:48.00   3rd Qu.:12.00                      3rd Qu.:45.00
##  Max.   :90.00   Max.   :16.00                      Max.   :99.00
##     salary
##  Length:32561
##  Class :character
##  Mode  :character
##
##
##
```

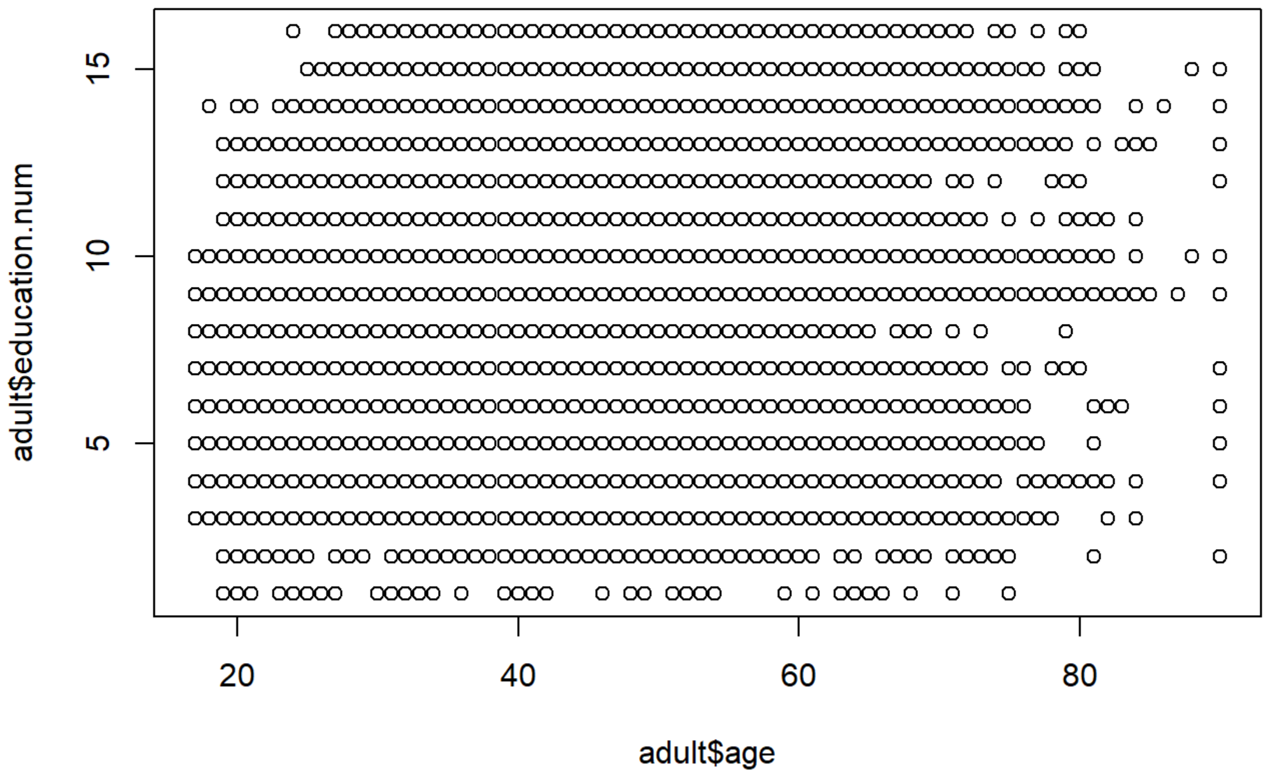
```
adult2$sex <- as.factor(adult2$sex)
adult2$salary <- as.factor(adult2$salary)
adult2$sex <- as.numeric(adult2$sex)
adult2$salary <- as.numeric(adult2$salary)
set.seed(2000)
ind <- sample(2, nrow(adult2), replace = T, prob=c(0.8, 0.2))
adult.train <- adult2[ind==1, 1:4]
adult.test <- adult2[ind==2, 1:4]
adult.trainLabels <- adult2[ind==1, 5]
adult.testLabels <- adult2[ind==2, 5]
summary(adult2)
```

```
##       age        education.num        sex         hoursperweek
##  Min.   :17.00   Min.   : 1.00   Min.   :1.000   Min.   : 1.00
##  1st Qu.:28.00   1st Qu.: 9.00   1st Qu.:1.000   1st Qu.:40.00
##  Median :37.00   Median :10.00   Median :2.000   Median :40.00
##  Mean   :38.58   Mean   :10.08   Mean   :1.669   Mean   :40.44
##  3rd Qu.:48.00   3rd Qu.:12.00   3rd Qu.:2.000   3rd Qu.:45.00
##  Max.   :90.00   Max.   :16.00   Max.   :2.000   Max.   :99.00
##      salary
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.241
##  3rd Qu.:1.000
##  Max.   :2.000
```

```
plot(adult[1:4], main = "adult Data", pch = 21, bg = c("red", "green3", "blue")[unclass(adult$sa
lary)])
```




```
plot(adult$age, adult$education.num, pch = 21, bg = c("red", "green3", "blue")[unclass(adult$sal
ary)] )
```




```
library(class)
adult_pred <- knn(train=adult.train, test=adult.test, cl=adult.trainLabels, k=3)
kNNresults <- adult_pred == adult.testLabels
acc <- length(which(kNNresults==T))/length(kNNresults)
table(kNNresults,adult_pred)
```

```
##           adult_pred
## kNNresults    1    2
##      FALSE  906  493
##      TRUE  4415  694
```

```
acc
```

```
## [1] 0.7850338
```

Dicision Tree parts

```
library(rpart)
tree_adult <- rpart(salary~., data=adult3, method="class")
tree_adult
```

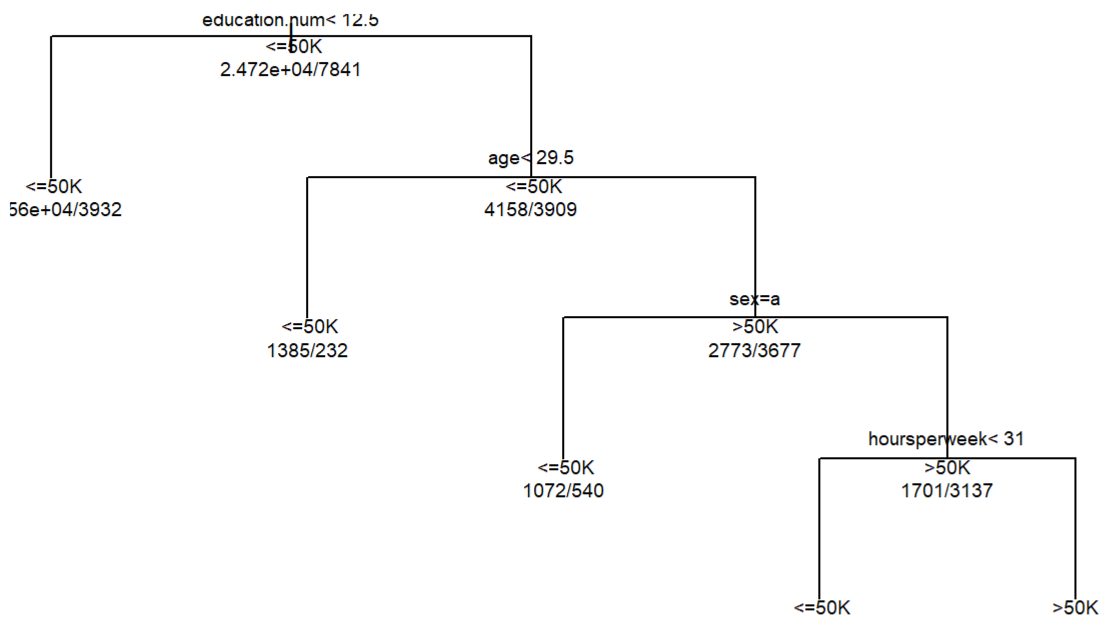
```
## n= 32561
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 32561 7841  <=50K (0.7591904 0.2408096)
##    2) education.num< 12.5 24494 3932  <=50K (0.8394709 0.1605291) *
##    3) education.num>=12.5 8067 3909  <=50K (0.5154332 0.4845668)
##      6) age< 29.5 1617  232  <=50K (0.8565244 0.1434756) *
##      7) age>=29.5 6450 2773  >50K (0.4299225 0.5700775)
##       14) sex= Female 1612  540  <=50K (0.6650124 0.3349876) *
##       15) sex= Male 4838 1701  >50K (0.3515916 0.6484084)
##         30) hoursperweek< 31 327  105  <=50K (0.6788991 0.3211009) *
##         31) hoursperweek>=31 4511 1479  >50K (0.3278652 0.6721348) *
```

```
summary(tree_adult)
```

```
## Call:
## rpart(formula = salary ~ ., data = adult3, method = "class")
##   n= 32561
##
##           CP nsplit rel error    xerror        xstd
## 1 0.05764571      0 1.0000000 1.0000000 0.009839876
## 2 0.01492157      3 0.8168601 0.8177528 0.009151746
## 3 0.01000000      4 0.8019385 0.8062747 0.009102917
##
## Variable importance
## education.num           age           sex  hoursperweek
##            62            23            12             4
##
## Node number 1: 32561 observations,    complexity param=0.05764571
##   predicted class= <=50K  expected loss=0.2408096  P(node) =1
##     class counts: 24720  7841
##    probabilities: 0.759 0.241
##   left son=2 (24494 obs) right son=3 (8067 obs)
##   Primary splits:
##       education.num < 12.5 to the left,  improve=1274.3680, (0 missing)
##       age           < 29.5 to the left,  improve= 980.1513, (0 missing)
##       hoursperweek  < 41.5 to the left,  improve= 712.6073, (0 missing)
##       sex           splits as  LR,       improve= 555.3667, (0 missing)
##
## Node number 2: 24494 observations
##   predicted class= <=50K  expected loss=0.1605291  P(node) =0.7522496
##     class counts: 20562  3932
##    probabilities: 0.839 0.161
##
## Node number 3: 8067 observations,    complexity param=0.05764571
##   predicted class= <=50K  expected loss=0.4845668  P(node) =0.2477504
##     class counts:  4158  3909
##    probabilities: 0.515 0.485
##   left son=6 (1617 obs) right son=7 (6450 obs)
##   Primary splits:
##       age           < 29.5 to the left,  improve=470.5799, (0 missing)
##       sex           splits as  LR,       improve=326.7394, (0 missing)
##       hoursperweek  < 43.5 to the left,  improve=227.0248, (0 missing)
##       education.num < 13.5 to the left,  improve=155.2742, (0 missing)
##
## Node number 6: 1617 observations
##   predicted class= <=50K  expected loss=0.1434756  P(node) =0.04966064
##     class counts:  1385   232
##    probabilities: 0.857 0.143
##
## Node number 7: 6450 observations,    complexity param=0.05764571
##   predicted class= >50K   expected loss=0.4299225  P(node) =0.1980897
##     class counts:  2773  3677
##    probabilities: 0.430 0.570
##   left son=14 (1612 obs) right son=15 (4838 obs)
##   Primary splits:
##       sex           splits as  LR,       improve=237.55100, (0 missing)
```

```
##       hoursperweek  < 42.5 to the left,  improve=135.11260, (0 missing)
##       education.num < 14.5 to the left,  improve= 85.80394, (0 missing)
##       age           < 36.5 to the left,  improve= 42.61073, (0 missing)
## 
## Node number 14: 1612 observations
##   predicted class= <=50K  expected loss=0.3349876  P(node) =0.04950708
##     class counts:  1072   540
##    probabilities: 0.665 0.335 
## 
## Node number 15: 4838 observations,    complexity param=0.01492157
##   predicted class= >50K   expected loss=0.3515916  P(node) =0.1485827
##     class counts:  1701  3137
##    probabilities: 0.352 0.648 
##   left son=30 (327 obs) right son=31 (4511 obs)
##   Primary splits:
##       hoursperweek  < 31   to the left,  improve=75.14200, (0 missing)
##       education.num < 13.5 to the left,  improve=55.32434, (0 missing)
##       age           < 36.5 to the left,  improve=32.51499, (0 missing)
##   Surrogate splits:
##       age < 70.5 to the right, agree=0.934, adj=0.024, (0 split)
## 
## Node number 30: 327 observations
##   predicted class= <=50K  expected loss=0.3211009  P(node) =0.01004269
##     class counts:   222   105
##    probabilities: 0.679 0.321 
## 
## Node number 31: 4511 observations
##   predicted class= >50K   expected loss=0.3278652  P(node) =0.13854
##     class counts:  1479  3032
##    probabilities: 0.328 0.672
```

```
plot(tree_adult, uniform=TRUE)
text(tree_adult, use.n=TRUE, all=TRUE, cex=.6)
```



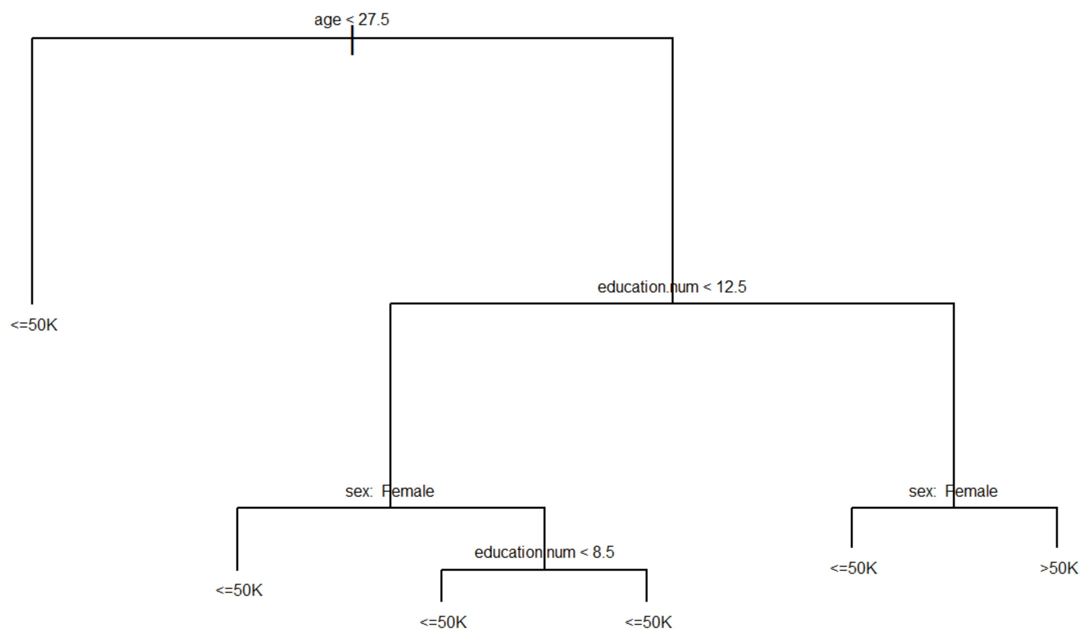

```r
library(tree)
adult3$sex <- as.factor(adult3$sex)
adult3$salary <- as.factor(adult3$salary)
tree_adult2 <- tree(salary~., data=adult3, method="class")
tree_adult2
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 32561 35950  <=50K ( 0.75919 0.24081 )
##    2) age < 27.5 8031  2282  <=50K ( 0.96787 0.03213 ) *
##    3) age > 27.5 24530 30340  <=50K ( 0.69087 0.30913 )
##      6) education.num < 12.5 17654 18320  <=50K ( 0.78617 0.21383 )
##       12) sex:  Female 5605  3517  <=50K ( 0.90508 0.09492 ) *
##       13) sex:  Male 12049 14030  <=50K ( 0.73085 0.26915 )
##         26) education.num < 8.5 2017  1343  <=50K ( 0.89638 0.10362 ) *
##         27) education.num > 8.5 10032 12300  <=50K ( 0.69757 0.30243 ) *
##      7) education.num > 12.5 6876  9452  >50K ( 0.44619 0.55381 )
##       14) sex:  Female 1761  2232  <=50K ( 0.67064 0.32936 ) *
##       15) sex:  Male 5115  6735  >50K ( 0.36891 0.63109 ) *
```

```r
summary(tree_adult2)
```

```
##
## Classification tree:
## tree(formula = salary ~ ., data = adult3, method = "class")
## Variables actually used in tree construction:
## [1] "age"           "education.num" "sex"
## Number of terminal nodes:  6
## Residual mean deviance:  0.8726 = 28410 / 32560
## Misclassification error rate: 0.1996 = 6500 / 32561
```

```
plot(tree_adult2)
text(tree_adult2, cex=0.5, pretty=0)
```




```
set.seed(2000)
i <- sample(1:nrow(adult3), 0.8*nrow(adult3), replace = F)
DT_train <- adult3[i,]
DT_test <- adult3[-i,]
tree_adult3 <- tree(salary~., data=DT_train)
DT_pred <- predict(tree_adult3, newdata=DT_test, type="class")
table(DT_pred, DT_test$salary)
```

```
##
## DT_pred   <=50K  >50K
##    <=50K   4585   927
##    >50K     367   634
```

```
mean(DT_pred==DT_test$salary)
```

```
## [1] 0.8013204
```

Narrative parts

In conclusion, DT is better than kNN for classification. The accurancy of DT is .80 and the accurancy of kNN is 0.785.

The KNN algorithm has a relatively high degree of adaptation to numerical data, and has less preprocessing. Generally, a single type of data can be normalized, and the formula for selecting distance can also be carried out based on the actual situation. One advantage of the KNN algorithm is that it is not sensitive to outliers, but during preprocessing, if the extreme data can be removed and then normalized, the classification effect will be better.

The decision tree algorithm has higher requirements for data preprocessing and requires pre-classification. The classification process will be better if it is oriented to the problem itself. If it is used to make the actual algorithm and put it into use, it will be better for a specific user to perform a scaling effect by the user. However, when there is too much data, more consideration should be given to the pruning of the decision tree, but it will inevitably reduce the accuracy.