Classfication

Jimmy Harvin, Yixin Sun

9-25-2022

In ML, classification is like a True/False question. It would divide the target into two group. Either belongs to or not belong to. Through the training data, we may separate the data into two regions by a linear equation. One side is the True(which means belongs to) and other side is the False(which means not belongs to). The strengths of linear models are easy to be understanding and interpretations. It's very intuitive and can easily update the data model. The weaknesses of linear models are terrible at dealing with non-linear relationship. Not flexible to identify complex patterns. Very tricky and time-consuming to add the correct interaction terms.

The data set is from the 1994 Census database. It could be find at UCI machine learning repository. Target to be classification is individual's annual income exceeds $50,000.

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" ...
```

data cleaning and divide into train and test

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

```
## '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(adult), 0.8*nrow(adult), replace = F)
train <- adult[i,]
test <- adult[-i,]
```

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" ...
```

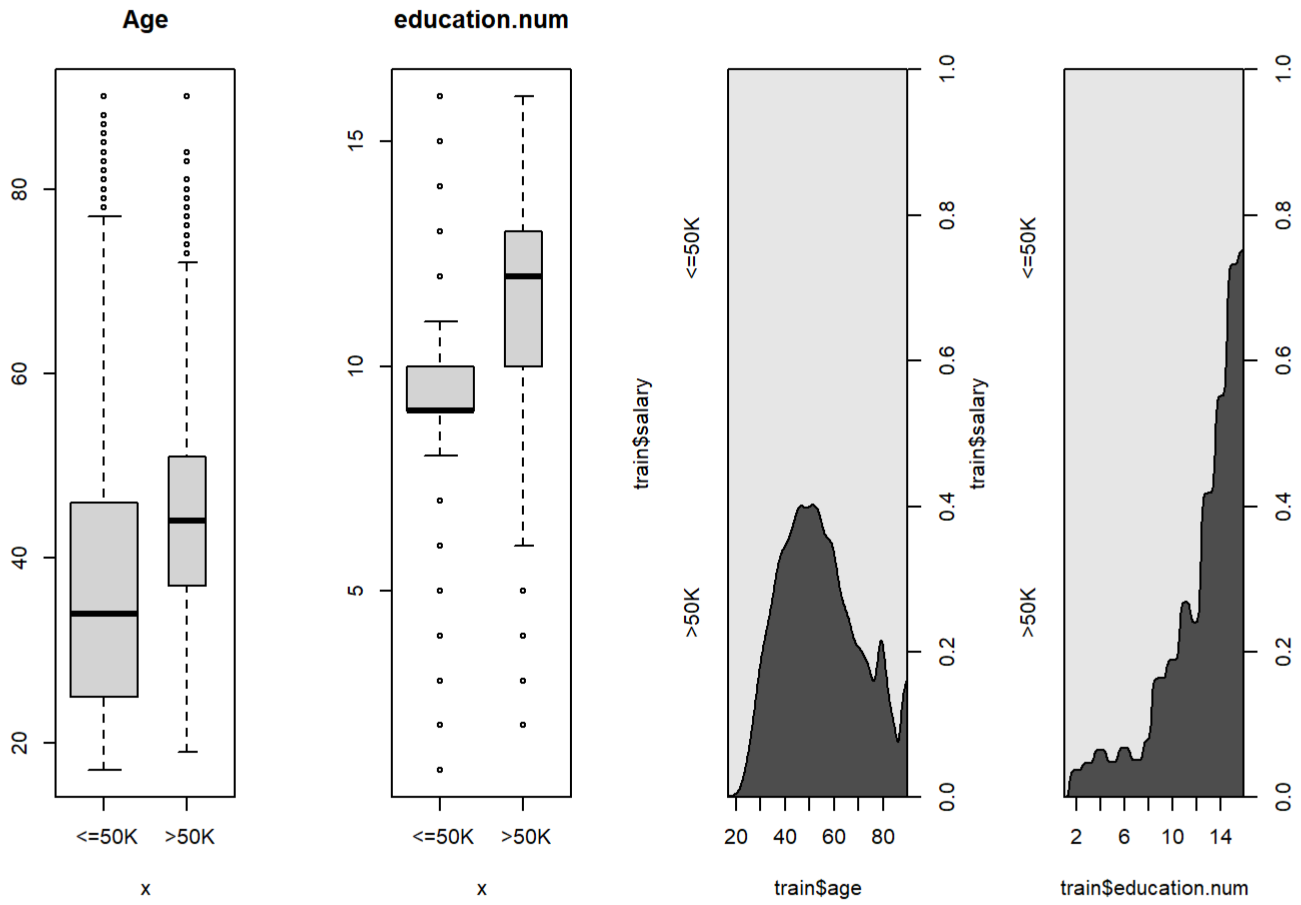
```
head(train, n = 15)
```

	age	education.num	sex	hoursperweek	salary
	<int>	<int>	<chr>	<int>	<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)
```



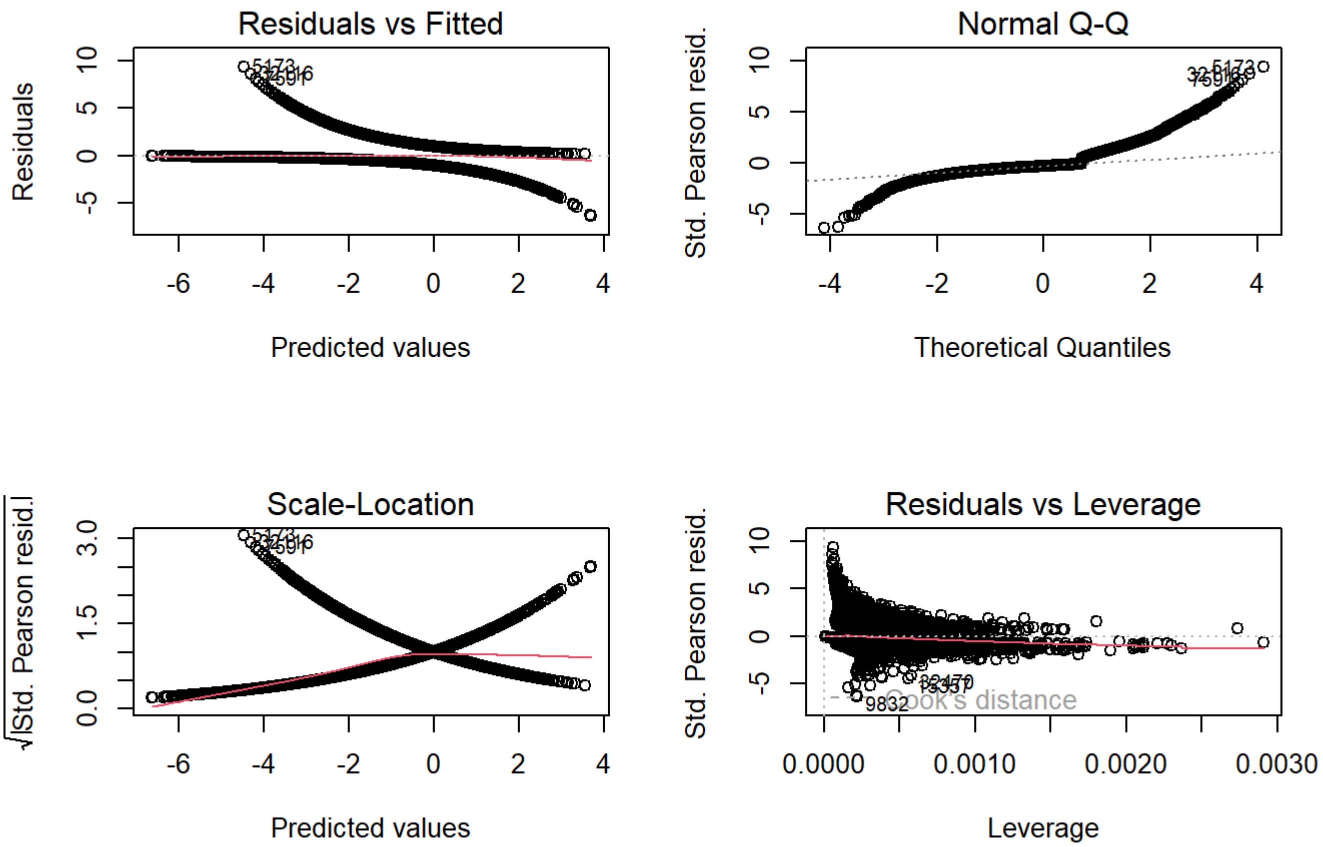
Build logistic regression model

In this model, we can mention that age, education.num, sex, and hours per week are best predictors.Every value has a very low p-value, indicates this is a good 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
```

Build naïve Bayes model

For the train data, 75.76% people's salary is lower than 50K and 24.23% are hight than 50k. For the people whose salary is lower than 50K, average age is 36, mean education number of years is 9.6, and mean work hours per week is 38.8 hours. and 39% of them are female and 60.8% of them are male. For the people whose salary is higher than 50K, average age is 44, mean education number of years is 11.6, and mean work hours per week is 45.57 hours. and 15% of them are female and 84.96% of them are male.

```
library(e1071)
nb1 <- naiveBayes(salary~., data = train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     <=50K      >50K
## 0.7576397 0.2423603
##
## Conditional probabilities:
##         age
## Y            [,1]     [,2]
##    <=50K 36.72207 13.99880
##    >50K  44.29178 10.48626
##
##         education.num
## Y             [,1]     [,2]
##    <=50K  9.602939 2.426069
##    >50K  11.598764 2.384754
##
##         sex
## Y           Female      Male
##    <=50K 0.3912845 0.6087155
##    >50K  0.1503247 0.8496753
##
##         hoursperweek
## Y            [,1]     [,2]
##    <=50K 38.80785 12.27280
##    >50K  45.57120 11.00593
```

predict and evaluate on the test data

For logistic regression, the accuracy is about 80.3%.And for naive Bayes, the accuracy is about 80.7%. The reason that naive Bayes is better than logistic regression model may because of each columns are independent of each other. This may increase the accuracy of the naive Bayes. Also there may some specific large error data in the dataset. This may influence the 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"
```

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

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

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

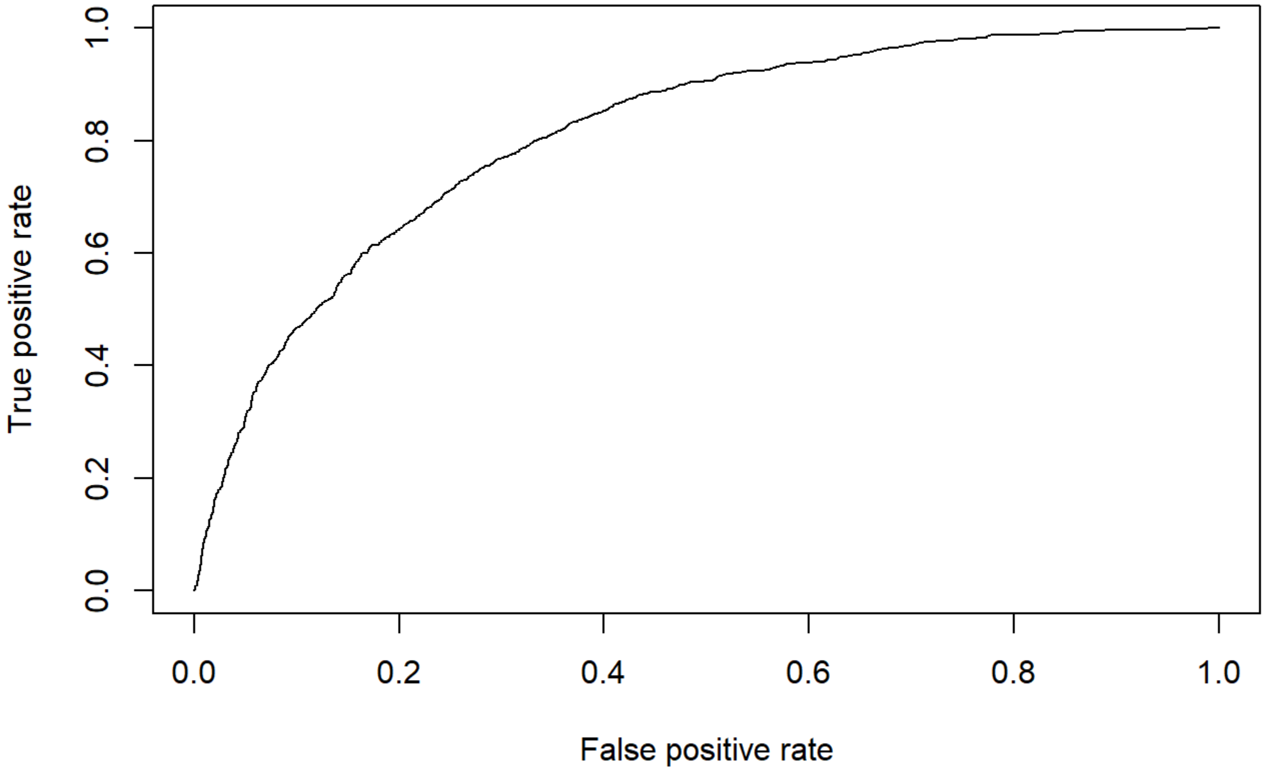
```
## Loading required package: ggplot2
```

```
## 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
```

```
# naive Bayes
p1 <- predict(nb1, newdata=test, type="class")
table(p1, test$salary)
```

```
##
## p1        <=50K  >50K
##    <=50K   4612   884
##    >50K     373   644
```

```
mean(p1==test$salary)
```

```
## [1] 0.8070014
```

the strengths and weaknesses of Naïve Bayes and Logistic Regression.

For naive Bayes, the algorithm logic is simple and easy to implement and has small space-time overhead in the classification process. But it assume that the sample features are independent of each other. This assumption basically does not exist in reality.When the number of attributes is large or the correlation between attributes is large, The classification effect is not good. For logistic regression, it suitable for binary classification problems, the interpret ability of the model is very good, can see the influence of different features on the final result. But it cannot be used to solve nonlinear problems and difficult to deal with the problem of data imbalance. And the accuracy is not very high, because the form is similar to a linear model, and it is difficult to fit the real distribution of the data.

the benefits, drawbacks of each of the classification metric

The confusion matrix contains a large number of parameters that can help us interpret the data, including matrix, accuracy, p-value, kappa, etc. The advantages of the ROC curve include that the ROC curve can remain unchanged when the distribution of positive and negative samples in the test set changes. In actual datasets, class imbalance often occurs, that is, there are many more negative samples than positive samples (or vice versa). AUC can judge the pros and cons of a prediction model. AUC = 1, it is a perfect classifier, when using this prediction model, there is at least one threshold to get a perfect prediction. 0.5 < AUC < 1, better than random guessing. This model can have predictive value if the threshold is properly set. AUC = 0.5, the model has no predictive value as the machine guesses. AUC < 0.5, worse than random guessing