R Notebook

Sai Gonuguntla

09/14/2022

In classification the target variable is qualitative, and we could have binary classification where its classified into two classes or it could be multiclass classification. For example,whether a tumor is benign or malignant. Strengths are, performs better than linear regression when there are non-linear relationships and is inexpensive. Weaknesses are, its prone to underfitting.

```
df <- read.csv("adult.csv", header=TRUE)
str(df)
```

```
## 'data.frame':    13713 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : chr  "?" "Private" "?" "Private" ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 4220
13 70037 ...
##  $ education     : chr  "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: chr  "Widowed" "Widowed" "Widowed" "Divorced" ...
##  $ occupation    : chr  "?" "Exec-managerial" "?" "Machine-op-inspct" ...
##  $ relationship  : chr  "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ..
.
##  $ race          : chr  "White" "White" "Black" "White" ...
##  $ sex           : chr  "Female" "Female" "Female" "Female" ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: chr  "United-States" "United-States" "United-States" "United-St
ates" ...
##  $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

a. divide into Train and test set

```
adult <-df[,c(1,5,15)]

set.seed(3)
i <- sample(1:nrow(adult), 0.80*nrow(adult), replace=FALSE)
train <- adult[i,]
test <- adult[-i,]
```

b. 5 R functions for data exploration using the training data.

```
attach(adult)
str(adult)
```

```
## 'data.frame':    13713 obs. of  3 variables:
##  $ age          : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ education.num: int  9 9 10 4 10 9 6 16 9 10 ...
##  $ income       : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

```
dim(adult)
```

```
## [1] 13713     3
```

```
head(adult,n=3)
```

	age	education.num	income
	<int>	<int>	<chr>
1	90	9	<=50K
2	82	9	<=50K
3	66	10	<=50K

3 rows

```
tail(adult,n=5)
```

	age	education.num	income
	<int>	<int>	<chr>
13709	28	9	<=50K
13710	46	9	<=50K
13711	45	9	<=50K
13712	23	10	<=50K
13713	33	12	<=50K

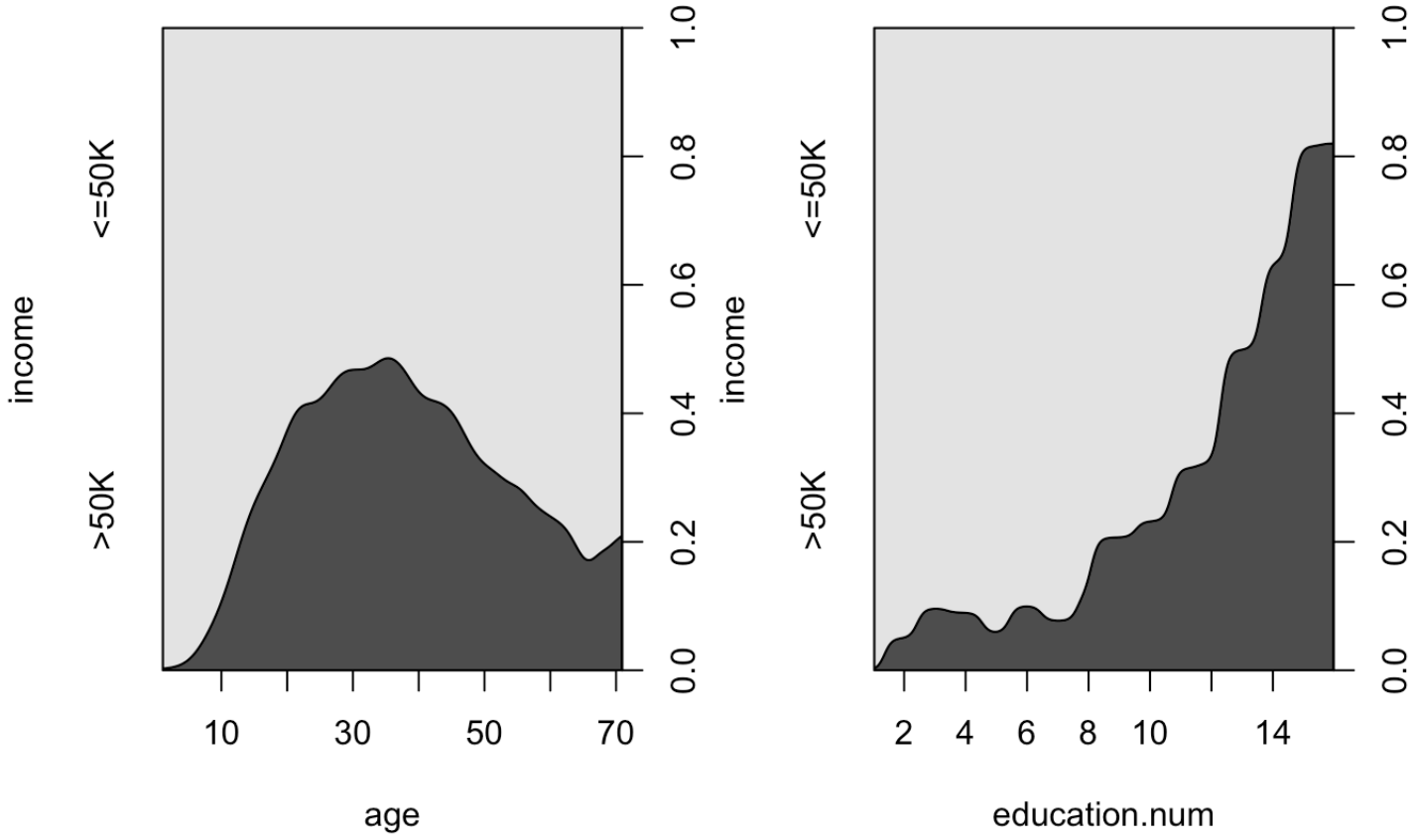
5 rows

c. 2 informative graphs using the training data

The second set of pots are conditional density plots. The total probability space is the rectangle, with the darker grey indicating income >50k. We can see that as the years of education increases salary increases but this is harder to see with the age cd plot

```
income<-factor(income)
age <- factor(age)

par(mfrow=c(1,2))
cdplot(income~age)
cdplot(income~education.num)
```



d. Build a logistic regression model

The model uses only age as a predictor.

deviance residuals are measures of deviance contributed from each observation, in our case min was -1.6841 and max was 1.9107. Age has a very low p-value which is good. There is decrease from null deviance to residual deviance from 13587 to 12935 this shows that the predictor has value. The AIC is 12939 and it is better if it is lower.

```
education.num <- factor(education.num)
glm1 <- glm(as.factor(income)~age, data=train, family=binomial)
summary(glm1)
```

```
##
## Call:
## glm(formula = as.factor(income) ~ age, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6841  -0.8669  -0.6834   1.3131   1.9107
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.396571   0.069513  -34.48   <2e-16 ***
## age          0.039305   0.001588   24.75   <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: 13587  on 10969  degrees of freedom
## Residual deviance: 12935  on 10968  degrees of freedom
## AIC: 12939
##
## Number of Fisher Scoring iterations: 4
```

e. build a naive bayes model and outputs what the model learned

I'm predicting income as a function of age and years of education. The prior probabiliy of earning <= 50k is 0.69 and earning > 50k is 0.31. For quantitative predictors the mean and standard deviation are given. The mean age for earning less than 50k is 37.2 and the mean age for earning more than 50k is 44.36. The mean education number for earning less than 50k is 9.65, and for earning more than 50k is 11.71.

```
library(e1071)
nb1 <- naiveBayes(income~age+education.num, data=train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     <=50K      >50K
## 0.6897903 0.3102097
##
## Conditional probabilities:
##        age
## Y           [,1]     [,2]
##   <=50K 37.20021 14.11790
##   >50K  44.36321 10.74748
##
##        education.num
## Y            [,1]     [,2]
##   <=50K  9.645831 2.422601
##   >50K  11.710843 2.427035
```

f. Evaluate on the test data naive bayes

mean Accuracy is 0.76, higher than what we got for logistic rgression. The table shows that the values which were predicted to be less than 50, and actually were less than 50 are 1717. There were 472 values predicted to be less than 50 but were actually greater than 50 . There were 187 values prediced to be greater than 50 but were acually less than 50, and 367 values that were predicted to be greater than 50 and actually were greater than 50.

```
p1 <- predict(nb1, newdata=test, type="class")
table(p1, test$income)
```

```
##
## p1      <=50K >50K
##   <=50K  1717  472
##   >50K    187  367
```

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

```
## [1] 0.7597521
```

f. Evaluate on the test data logisic regression

Probability greater than .5 is classified as 2 and probability less than or equal to .5 is classified as 1. The first model uses age as a predictor and there's about 66% accuracy. The table shows that the values which were predicted to be one, and actually were one are 1749. There were 775 values predicted to be one but were actually two. There were 155 values predicted to be 2 but were actually 1, and 64 values that were predicted to be 2 and actually were 2. The ROCR plot had slight curve but not that much which shows that the classifier didn't have as much predictive value. AUC is .7 which is okay but it should be closer 1.

Comparing the results naive bayes seems to have done better as it has a higher accuracy rate at 0.7598. And it did a better job predicting the true positive rate. I think this results happened because naive bayes works well on high dmensions

```
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, 2, 1)
acc1 <- mean(pred==(as.integer(as.factor(test$income))))
print(paste("accuracy = ", acc1))
```

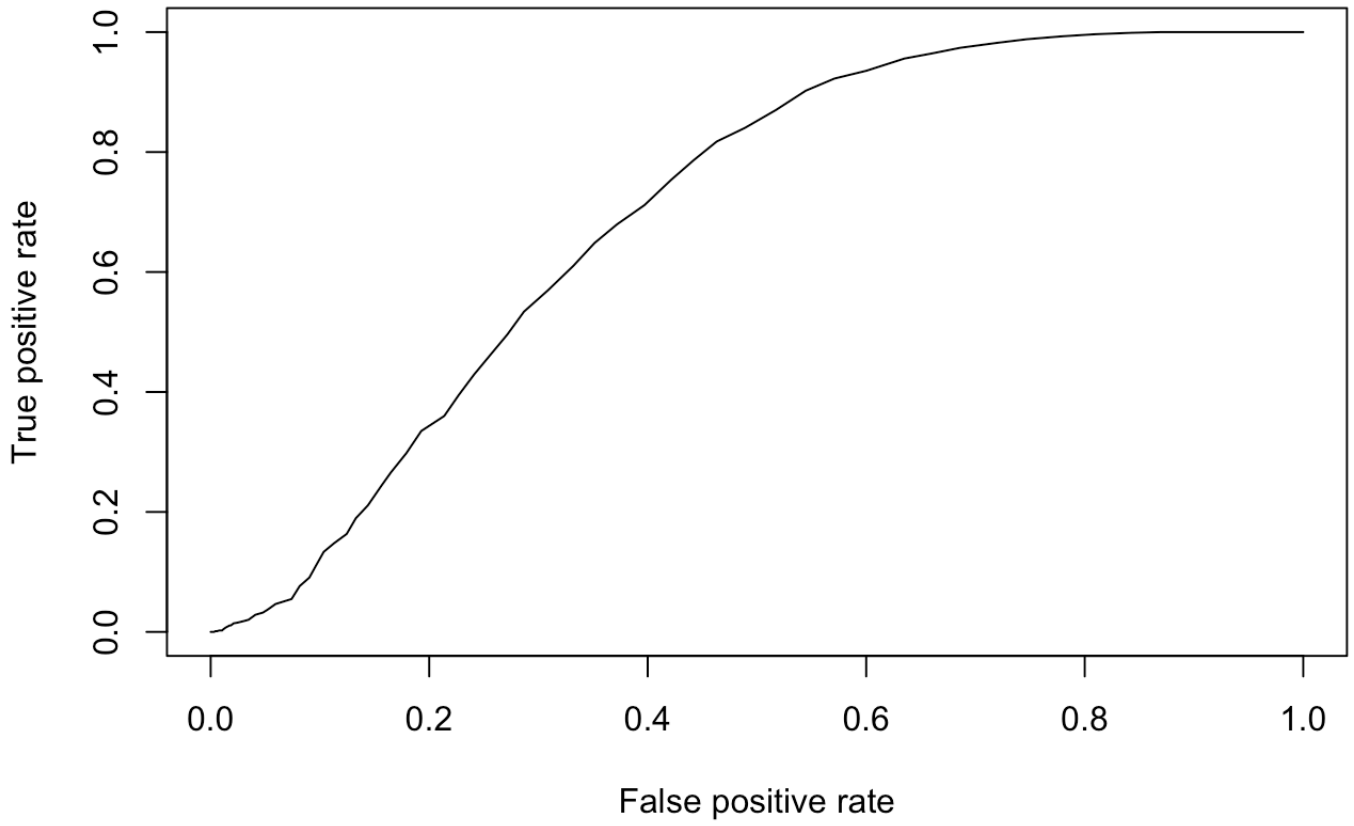
```
## [1] "accuracy =  0.66095515858549"
```

```
table(pred,  as.integer(as.factor(test$income)))
```

```
##
## pred    1    2
##    1 1749  775
##    2  155   64
```

```
pred<-as.factor(pred)
income<-as.factor(income)


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




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

```
## [1] 0.7000255
```

###g. The strengths of naive bayes is that it's easy to implement and interpret, works well with high dimensions and small datasets. The weakness is other classifiers may outperform it for larger data sets and if the predictors aren't independent he performance of the algorithm may be worse. The strengths of logistic regression are it's computationally inexpensive, and outputs of good probability. A weakness is that it's prone to underfitting.

###h. The matrix tells us the false positive, false negative, true positive, true negative values which are helpful to know so we know how many values are predicting correctly and how many weren't. The accuracy tells us how well the predict function estimated values which is important to know. The ROC plot tells us the trade off for predicting true positive and avoiding false positives, and tells us how good of a classifier it is. The AUC is the area under the curve and it should be closer to 1 for it to be a good classifier.