Classification

Sunni Magan

02/14/2023

Introduction

Linear Classifiers are models which attempt to find a line which can linearly separate two classes of data. This is a binary classification techique where one side of the line contains observations of one class, and the other side of the line contains observations belonging to the other. The two methods of linear classification explored in the notebook are Logistic Regression, and Naive Bayes. Logistic Regression maximizes the log-likelihood to estimate the probability of an event occurring. Naive Bayes instead uses Bayes Theorem to to estimate the probability. Logistic Regression is a fast, probabilistic method which works well on linearly separable data. However, it is susceptible to underfitting the data is not perfectly linearly separable. Naive Bayes is powerful even on smaller datasets and is relatively simple to implement, However, on larger datasets, it may under perform. This could be due to the fact that Naive Bayes “naively” assumes that the features are independent.

Logistic regression is what is known as a discriminative classifier and Naive Bayes is a generative classifier. A generative classifier means you can “generate” new data from the result because you have directly calculated the posterior from prior probabilities. However in a discriminative classifier, you are learning the posterior probability directly form the data, which mean you cannot create new data from it.

This notebook analyzes the information of credit card users in Taiwan from April 2005 to September 2005. Using linear classifiers we will predict whether a credit card user would default payment.

Source: https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset (https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset)

About the data

- ID: ID of each client
- LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
- PAY_2: Repayment status in August, 2005 (scale same as above)
- PAY_3: Repayment status in July, 2005 (scale same as above)
- PAY_4: Repayment status in June, 2005 (scale same as above)
- PAY_5: Repayment status in May, 2005 (scale same as above)
- PAY_6: Repayment status in April, 2005 (scale same as above)
- BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
- BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
- BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
- BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
- BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)

- BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
- PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
- PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
- PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
- PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
- PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
- PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
- default.payment.next.month: Default payment (1=yes, 0=no)

```
df <- read.csv("UCI_Credit_Card.csv")
str(df)
```

```
## 'data.frame':    30000 obs. of  25 variables:
##  $ ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ LIMIT_BAL: num  20000 120000 90000 50000 50000 50000 500000 100000 140000 20000 ...
##  $ SEX      : int  2 2 2 2 1 1 1 2 2 1 ...
##  $ EDUCATION: int  2 2 2 2 2 1 1 2 3 3 ...
##  $ MARRIAGE : int  1 2 2 1 1 2 2 2 1 2 ...
##  $ AGE      : int  24 26 34 37 57 37 29 23 28 35 ...
##  $ PAY_0    : int  2 -1 0 0 -1 0 0 0 0 -2 ...
##  $ PAY_2    : int  2 2 0 0 0 0 0 -1 0 -2 ...
##  $ PAY_3    : int  -1 0 0 0 -1 0 0 -1 2 -2 ...
##  $ PAY_4    : int  -1 0 0 0 0 0 0 0 0 -2 ...
##  $ PAY_5    : int  -2 0 0 0 0 0 0 0 0 -1 ...
##  $ PAY_6    : int  -2 2 0 0 0 0 0 -1 0 -1 ...
##  $ BILL_AMT1: num  3913 2682 29239 46990 8617 ...
##  $ BILL_AMT2: num  3102 1725 14027 48233 5670 ...
##  $ BILL_AMT3: num  689 2682 13559 49291 35835 ...
##  $ BILL_AMT4: num  0 3272 14331 28314 20940 ...
##  $ BILL_AMT5: num  0 3455 14948 28959 19146 ...
##  $ BILL_AMT6: num  0 3261 15549 29547 19131 ...
##  $ PAY_AMT1 : num  0 0 1518 2000 2000 ...
##  $ PAY_AMT2 : num  689 1000 1500 2019 36681 ...
##  $ PAY_AMT3 : num  0 1000 1000 1200 10000 657 38000 0 432 0 ...
##  $ PAY_AMT4 : num  0 1000 1000 1100 9000 ...
##  $ PAY_AMT5 : num  0 0 1000 1069 689 ...
##  $ PAY_AMT6 : num  0 2000 5000 1000 679 ...
##  $ default  : int  1 1 0 0 0 0 0 0 0 0 ...
```

Data Cleaning

```
df$SEX <- as.factor(df$SEX)
df$EDUCATION <- as.factor(df$EDUCATION)
df$MARRIAGE <- as.factor(df$MARRIAGE)
df$default <- as.factor(df$default)
```

Train/Test Split

```
set.seed(1234)
i <- sample(1:nrow(df), 0.8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Performing an 80/20 split on the data to create training and testing sets

Data Exploration

```
str(train)
```

```
## 'data.frame':    24000 obs. of  25 variables:
##  $ ID       : int  7452 8016 7162 8086 23653 9196 623 15241 10885 934 ...
##  $ LIMIT_BAL: num  360000 80000 100000 500000 20000 30000 90000 180000 450000 30000 ...
##  $ SEX      : Factor w/ 2 levels "1","2": 2 2 2 1 2 1 2 2 1 1 ...
##  $ EDUCATION: Factor w/ 7 levels "0","1","2","3",..: 3 2 3 3 2 4 2 3 2 3 ...
##  $ MARRIAGE : Factor w/ 4 levels "0","1","2","3": 3 3 3 3 2 3 3 2 2 3 ...
##  $ AGE      : int  26 26 37 35 37 31 27 42 53 32 ...
##  $ PAY_0    : int  1 0 0 0 1 0 0 1 -1 2 ...
##  $ PAY_2    : int  -2 0 0 0 -2 0 0 -2 -1 0 ...
##  $ PAY_3    : int  -2 2 0 0 -1 0 -2 -2 -1 0 ...
##  $ PAY_4    : int  -2 2 0 0 2 0 -2 -2 -1 2 ...
##  $ PAY_5    : int  -2 2 0 0 2 0 -2 -2 -1 2 ...
##  $ PAY_6    : int  -2 2 0 0 -2 -2 -2 -2 0 2 ...
##  $ BILL_AMT1: num  0 38174 177961 207237 -113 ...
##  $ BILL_AMT2: num  0 40550 108173 224007 -113 ...
##  $ BILL_AMT3: num  0 41577 15697 275615 10887 ...
##  $ BILL_AMT4: num  0 41595 11353 220088 10413 ...
##  $ BILL_AMT5: num  0 43264 9306 216482 -245 ...
##  $ BILL_AMT6: num  0 43402 9693 136086 -245 ...
##  $ PAY_AMT1 : num  0 3000 3082 20001 1575 ...
##  $ PAY_AMT2 : num  0 2000 2022 30168 11000 ...
##  $ PAY_AMT3 : num  0 1000 1000 6022 0 ...
##  $ PAY_AMT4 : num  0 2500 1000 6375 0 ...
##  $ PAY_AMT5 : num  0 1000 500 5005 0 ...
##  $ PAY_AMT6 : num  0 2000 300 5000 5100 ...
##  $ default  : Factor w/ 2 levels "0","1": 1 2 2 1 2 2 1 1 1 2 ...
```

```
summary(train)
```

```
##        ID          LIMIT_BAL       SEX       EDUCATION MARRIAGE
##  Min.   :    1   Min.   :  10000   1: 9463   0:   14   0:   41
##  1st Qu.: 7522   1st Qu.:  50000   2:14537   1: 8462   1:10915
##  Median :14966   Median : 140000             2:11243   2:12781
##  Mean   :14990   Mean   : 167201             3: 3916   3:  263
##  3rd Qu.:22473   3rd Qu.: 240000             4:  103
##  Max.   :30000   Max.   :1000000             5:  218
##                                              6:   44
##       AGE            PAY_0              PAY_2              PAY_3
##  Min.   :21.00   Min.   :-2.00000   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:28.00   1st Qu.:-1.00000   1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median :34.00   Median : 0.00000   Median : 0.0000   Median : 0.0000
##  Mean   :35.45   Mean   :-0.01217   Mean   :-0.1333   Mean   :-0.1652
##  3rd Qu.:41.00   3rd Qu.: 0.00000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   :79.00   Max.   : 8.00000   Max.   : 8.0000   Max.   : 8.0000
##
##      PAY_4              PAY_5              PAY_6           BILL_AMT1
##  Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-165580
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:   3556
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median :  22593
##  Mean   :-0.2188   Mean   :-0.2645   Mean   :-0.2898   Mean   :  51171
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:  67572
##  Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 964511
##
##    BILL_AMT2         BILL_AMT3          BILL_AMT4          BILL_AMT5
##  Min.   :-69777   Min.   :-157264   Min.   :-170000   Min.   :-81334
##  1st Qu.:  3000   1st Qu.:   2635   1st Qu.:   2304   1st Qu.:  1730
##  Median : 21336   Median :  20197   Median :  19106   Median : 18107
##  Mean   : 49139   Mean   :  46987   Mean   :  43251   Mean   : 40313
##  3rd Qu.: 64251   3rd Qu.:  60585   3rd Qu.:  54814   3rd Qu.: 50297
##  Max.   :983931   Max.   :1664089   Max.   : 891586   Max.   :927171
##
##    BILL_AMT6           PAY_AMT1          PAY_AMT2           PAY_AMT3
##  Min.   :-339603   Min.   :     0   Min.   :      0   Min.   :     0
##  1st Qu.:   1245   1st Qu.:  1000   1st Qu.:    833   1st Qu.:   390
##  Median :  17084   Median :  2110   Median :   2008   Median :  1800
##  Mean   :  38874   Mean   :  5625   Mean   :   5927   Mean   :  5206
##  3rd Qu.:  49497   3rd Qu.:  5007   3rd Qu.:   5000   3rd Qu.:  4500
##  Max.   : 961664   Max.   :873552   Max.   :1684259   Max.   :896040
##
##     PAY_AMT4           PAY_AMT5           PAY_AMT6      default
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0   0:18664
##  1st Qu.:   298.8   1st Qu.:   225.8   1st Qu.:   100   1: 5336
##  Median :  1500.0   Median :  1500.0   Median :  1500
##  Mean   :  4788.3   Mean   :  4792.3   Mean   :  5155
##  3rd Qu.:  4000.0   3rd Qu.:  4001.2   3rd Qu.:  4000
##  Max.   :621000.0   Max.   :417990.0   Max.   :528666
##
```

```
head(train)
```

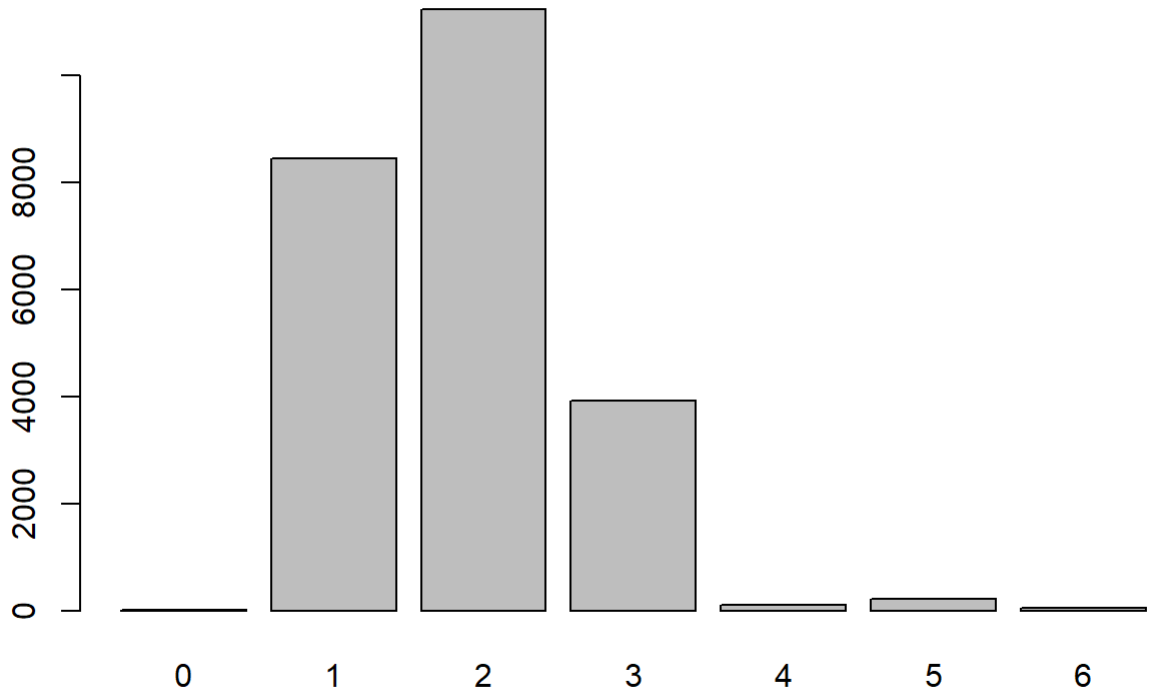
```
##          ID LIMIT_BAL SEX EDUCATION MARRIAGE AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5
## 7452   7452    360000   2         2        2  26     1    -2    -2    -2    -2
## 8016   8016     80000   2         1        2  26     0     0     2     2     2
## 7162   7162    100000   2         2        2  37     0     0     0     0     0
## 8086   8086    500000   1         2        2  35     0     0     0     0     0
## 23653 23653     20000   2         1        1  37     1    -2    -1     2     2
## 9196   9196     30000   1         3        2  31     0     0     0     0     0
##       PAY_6 BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6
## 7452     -2         0         0         0         0         0         0
## 8016      2     38174     40550     41577     41595     43264     43402
## 7162      0    177961    108173     15697     11353      9306      9693
## 8086      0    207237    224007    275615    220088    216482    136086
## 23653    -2      -113      -113     10887     10413      -245      -245
## 9196     -2     27838     28791     27788     29784         0         0
##       PAY_AMT1 PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default
## 7452         0        0        0        0        0        0       0
## 8016      3000     2000     1000     2500     1000     2000       1
## 7162      3082     2022     1000     1000      500      300       1
## 8086     20001    30168     6022     6375     5005     5000       0
## 23653     1575    11000        0        0        0     5100       1
## 9196      1703     1200     2196     2500        0        0       1
```

```
cor(train$LIMIT_BAL, train$AGE)
```

```
## [1] 0.146431
```

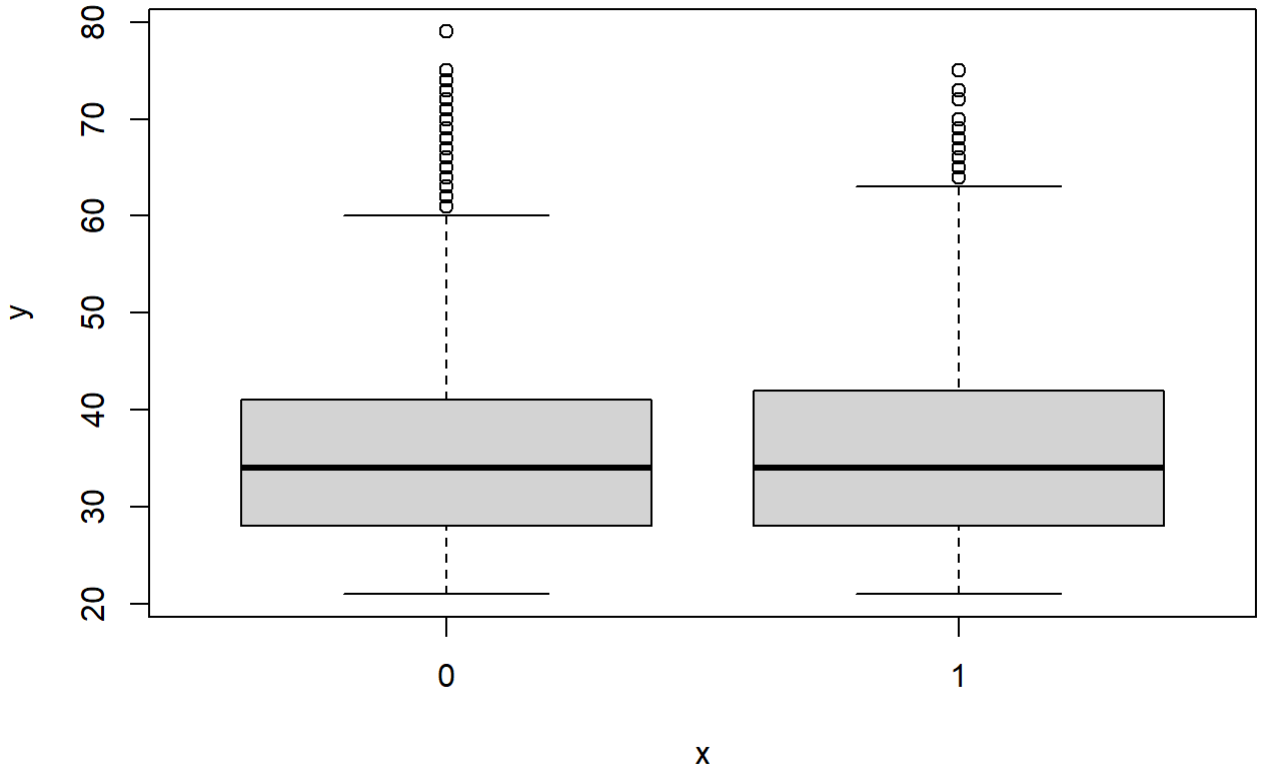
Interestingly, there is a slight correlation between age and limit balance.

```
barplot(table(train$EDUCATION))
```



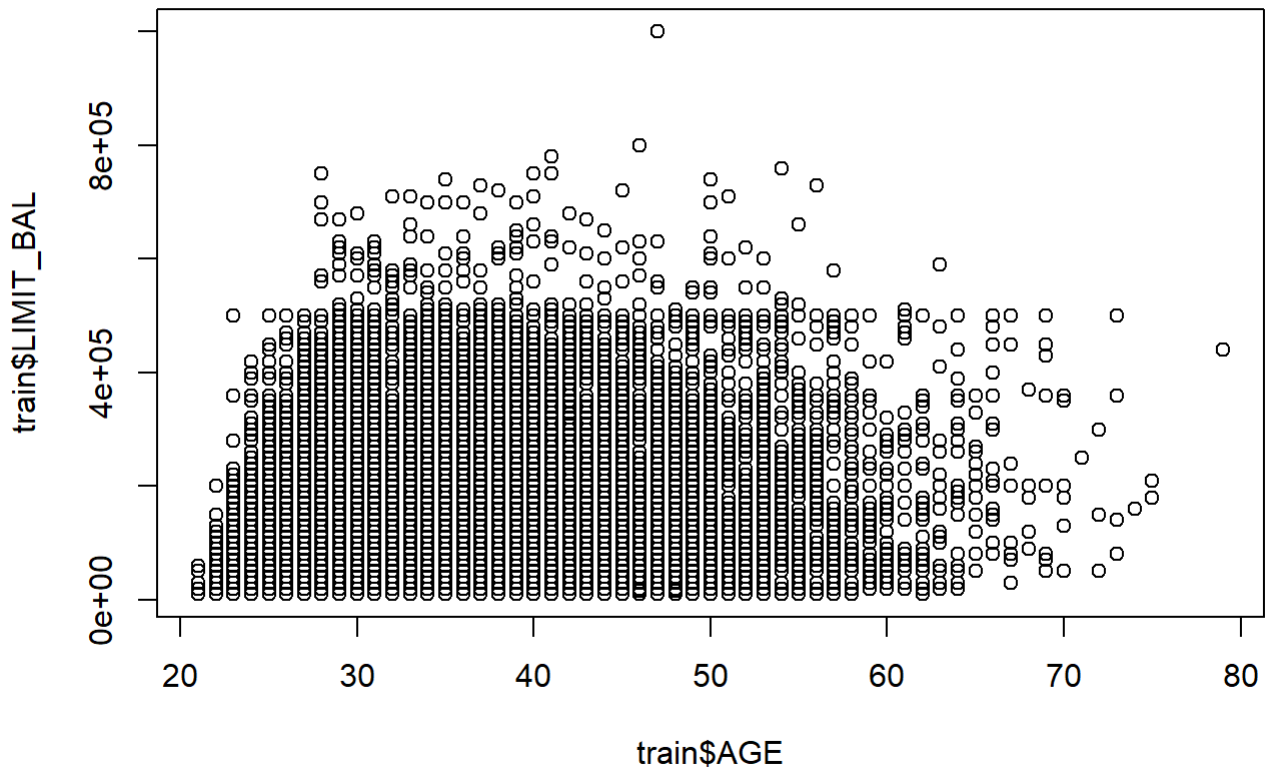
As we can see fro the histogram, more than half of the observations are of people who have recieve some form of further education past highschool. This could be an import metric in determining weather people default on credit card payments.

```
plot(train$default, train$AGE)
```



It seems like age alone is irrelevant to whether a person default on their credit card

```
plot(train$AGE, train$LIMIT_BAL)
```



Here it seems as age is not relevant to credit limit either ## Training Models ### Logistic Regression

```
model_1 <- glm(default~., data=train, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model_1)
```

```
##
## Call:
## glm(formula = default ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1213  -0.7035  -0.5459  -0.2787   3.2581
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.321e+01  8.251e+01  -0.160 0.872779
## ID          -2.010e-06  1.957e-06  -1.027 0.304231
## LIMIT_BAL   -6.835e-07  1.764e-07  -3.874 0.000107 ***
## SEX2        -1.009e-01  3.437e-02  -2.937 0.003313 **
## EDUCATION1   1.079e+01  8.251e+01   0.131 0.895959
## EDUCATION2   1.071e+01  8.251e+01   0.130 0.896720
## EDUCATION3   1.071e+01  8.251e+01   0.130 0.896722
## EDUCATION4   9.239e+00  8.251e+01   0.112 0.910845
## EDUCATION5   9.158e+00  8.251e+01   0.111 0.911617
## EDUCATION6   1.047e+01  8.251e+01   0.127 0.898979
## MARRIAGE1    1.486e+00  5.826e-01   2.550 0.010776 *
## MARRIAGE2    1.275e+00  5.828e-01   2.187 0.028729 *
## MARRIAGE3    1.422e+00  6.009e-01   2.366 0.017989 *
## AGE          4.897e-03  2.084e-03   2.350 0.018799 *
## PAY_0        5.709e-01  1.979e-02  28.849  < 2e-16 ***
## PAY_2        8.126e-02  2.270e-02   3.580 0.000343 ***
## PAY_3        6.177e-02  2.554e-02   2.419 0.015575 *
## PAY_4        2.205e-02  2.809e-02   0.785 0.432431
## PAY_5        3.672e-02  2.997e-02   1.225 0.220439
## PAY_6        2.265e-02  2.465e-02   0.919 0.358074
## BILL_AMT1   -6.807e-06  1.301e-06  -5.232 1.68e-07 ***
## BILL_AMT2    3.589e-06  1.667e-06   2.153 0.031355 *
## BILL_AMT3    1.825e-06  1.471e-06   1.240 0.214818
## BILL_AMT4   -3.487e-07  1.506e-06  -0.232 0.816823
## BILL_AMT5    7.774e-08  1.704e-06   0.046 0.963602
## BILL_AMT6    7.314e-07  1.331e-06   0.550 0.582525
## PAY_AMT1    -1.644e-05  2.731e-06  -6.018 1.76e-09 ***
## PAY_AMT2    -7.896e-06  2.199e-06  -3.591 0.000329 ***
## PAY_AMT3    -2.760e-06  1.977e-06  -1.396 0.162734
## PAY_AMT4    -4.690e-06  2.037e-06  -2.302 0.021332 *
## PAY_AMT5    -1.465e-06  1.839e-06  -0.797 0.425680
## PAY_AMT6    -1.541e-06  1.449e-06  -1.064 0.287382
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25433  on 23999  degrees of freedom
## Residual deviance: 22291  on 23968  degrees of freedom
## AIC: 22355
```

```
##
## Number of Fisher Scoring iterations: 11
```

Naive Bayes

```
library(e1071)
model_2 <- naiveBayes(default~., data=train)
summary(model_2)
```

```
##           Length Class  Mode
## apriori    2     table  numeric
## tables    24     -none- list
## levels     2     -none- character
## isnumeric 24     -none- logical
## call       4     -none- call
```

Results

Logistic Regression

```
prob <- predict(model_1, newdata=test)
p1 <- ifelse(prob >0.5, 1, 0)
pred <- table(p1, test$default)
acc <- mean(p1==test$default)
error_rate <- 1 - acc
sensitivity <- pred[1,1]/(pred[1,1]+pred[2,1])
specificity <- pred[2,2]/(pred[2,2]+pred[1,2])
pred
```

```
##
## p1     0    1
##   0 4666 1193
##   1   34  107
```

```
paste("acc: ", acc)
```

```
## [1] "acc:  0.7955"
```

```
paste("error_rate: ", error_rate)
```

```
## [1] "error_rate:  0.2045"
```

```
paste("sensitivity: ", sensitivity)
```

```
## [1] "sensitivity:  0.992765957446809"
```

```
paste("specificity: ", specificity)
```

```
## [1] "specificity:  0.0823076923076923"
```

Naive Bayes

```
p2 <- predict(model_2, newdata=test)
pred <- table(p2, test$default)
acc <- mean(p2==test$default)
error_rate <- 1 - acc
sensitivity <- pred[1,1]/(pred[1,1]+pred[2,1])
specificity <- pred[2,2]/(pred[2,2]+pred[1,2])
pred
```

```
##
## p2     0    1
##   0 3359  449
##   1 1341  851
```

```
paste("acc: ", acc)
```

```
## [1] "acc:  0.701666666666667"
```

```
paste("error_rate: ", error_rate)
```

```
## [1] "error_rate:  0.298333333333333"
```

```
paste("sensitivity: ", sensitivity)
```

```
## [1] "sensitivity:  0.71468085106383"
```

```
paste("specificity: ", specificity)
```

```
## [1] "specificity:  0.654615384615385"
```

Of the two, Logistic Regression as a higher accuracy which means it classified 79.55% of the observations corectly as opposed to NB only classifying 70.17% correctly. This is useful, but it could be misleading if our data is skewed. To determine if the Logistic Regression model is actually classifying generally, we must look at other metrics. Logistic Regression also has a higher sensitivity which is the rate that it classifies ‘true’ observations

correctly. However, the sensitivity rate (true negative rate) is very low for Logistic Regression. Overall, it seems that the Logistic Regression Model has not generaliesd well and is eager to classify obervations as true. For this reason, Naive Bayes seems to be the better model of the two.