Classification with Naive-Bayes Algorithm in R

Prediction Analysis on Bank Loan Default

Domain – Banking & Finance
Sudipto Mitra
https://data.world/sudmitra

Problem Statement

A bank is interested in knowing the factors which identify customers who are likely to default on their loans. The bank utilises this analysis in future to avoid approving loans to applicants who are riskier to the business.

Dataset

Bank_Loan_Default.csv

Coding in R

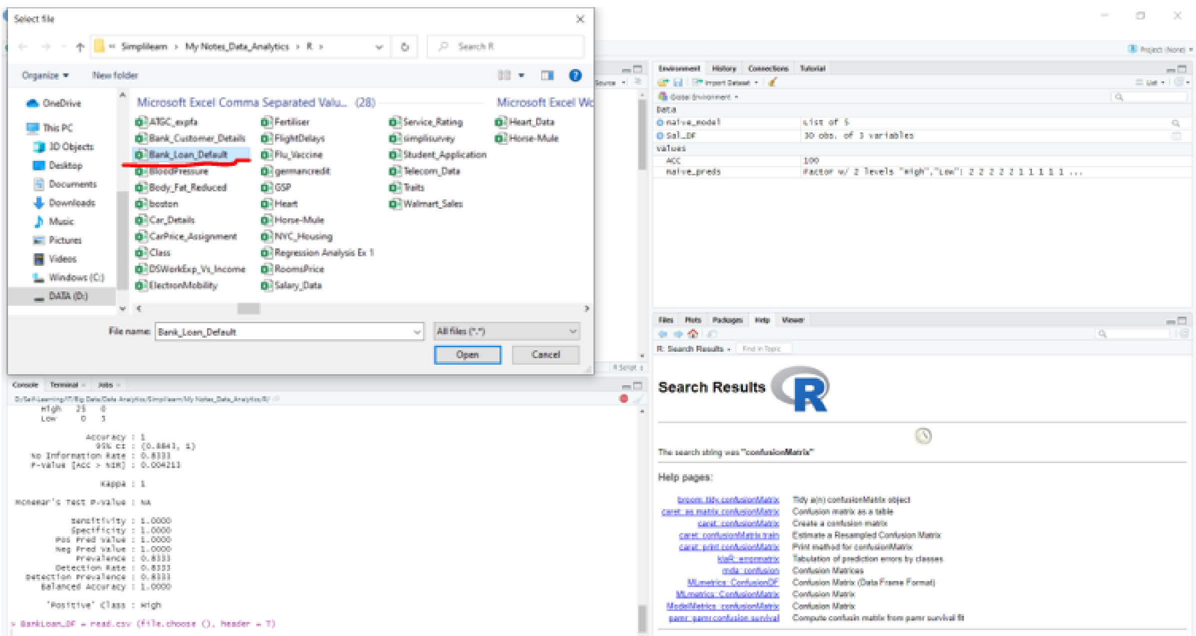
```
## NAIVE-BAYES CLASSIFIER: Prediction of Bank Loan Default ##

## Loading requisite packages ##

library (e1071)
library (plyr)
library (caret)
library (mlbench)
```

```
## Importing observational dataset ##

BankLoan_DF = read.csv (file.choose (), header = T)
```



Classification with Naive-Bayes Algorithm in R

Prediction Analysis on Bank Loan Default

Domain – Banking & Finance

Sudipto Mitra
https://data.world/sudmitra

```
View (BankLoan_DF)
```

Filter

	Default	checkingstatus1	duration	history	purpose	amount	savings	employ	installment	residence	age	otherplans	cards	te
1	0	A11	6	A34	A43	1169	A65	A75	4	4	67	A143	2	
2	1	A12	48	A32	A43	5951	A61	A73	2	2	22	A143	1	
3	0	A14	12	A34	A46	2096	A61	A74	2	3	49	A143	1	
4	0	A11	42	A32	A42	7882	A61	A74	2	4	45	A143	1	
5	1	A11	24	A33	A40	4870	A61	A73	3	4	53	A143	2	
6	0	A14	36	A32	A46	9055	A65	A73	2	4	35	A143	1	
7	0	A14	24	A32	A42	2835	A63	A75	3	4	53	A143	1	
8	0	A12	36	A32	A41	6948	A61	A73	2	2	35	A143	1	
9	0	A14	12	A32	A43	3059	A64	A74	2	4	61	A143	1	
10	1	A12	30	A34	A40	5234	A61	A71	4	2	28	A143	2	
11	1	A12	12	A32	A40	1295	A61	A72	3	1	25	A143	1	
12	1	A11	48	A32	A49	4308	A61	A72	3	4	24	A143	1	
13	0	A12	12	A32	A43	1567	A61	A73	1	1	22	A143	1	
14	1	A11	24	A34	A40	1199	A61	A75	4	4	60	A143	2	
15	0	A11	15	A32	A40	1403	A61	A73	2	4	28	A143	1	
16	1	A11	24	A32	A43	1282	A62	A73	4	2	32	A143	1	

Showing 1 to 17 of 1,000 entries, 14 total columns

```
> head (BankLoan_DF)
  Default checkingstatus1 duration history purpose amount savings employ installment residence age otherplans cards tele
1       0             A11        6     A34     A43   1169     A65    A75           4         4  67       A143     2 A192
2       1             A12       48     A32     A43   5951     A61    A73           2         2  22       A143     1 A191
3       0             A14       12     A34     A46   2096     A61    A74           2         3  49       A143     1 A191
4       0             A11       42     A32     A42   7882     A61    A74           2         4  45       A143     1 A191
5       1             A11       24     A33     A40   4870     A61    A73           3         4  53       A143     2 A191
6       0             A14       36     A32     A46   9055     A65    A73           2         4  35       A143     1 A192
```

```
## Checking data structures of variables ##
```

```
> str (BankLoan_DF)
'data.frame':   1000 obs. of  14 variables:
 $ Default        : int  0 1 0 0 1 0 0 0 0 1 ...
 $ checkingstatus1: chr  "A11" "A12" "A14" "A11" ...
 $ duration       : int  6 48 12 42 24 36 24 36 12 30 ...
 $ history        : chr  "A34" "A32" "A34" "A32" ...
 $ purpose        : chr  "A43" "A43" "A46" "A42" ...
 $ amount         : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
 $ savings        : chr  "A65" "A61" "A61" "A61" ...
 $ employ         : chr  "A75" "A73" "A74" "A74" ...
 $ installment    : int  4 2 2 2 3 2 3 2 2 4 ...
 $ residence      : int  4 2 3 4 4 4 4 2 4 2 ...
 $ age            : int  67 22 49 45 53 35 53 35 61 28 ...
 $ otherplans     : chr  "A143" "A143" "A143" "A143" ...
 $ cards          : int  2 1 1 1 2 1 1 1 1 2 ...
 $ tele           : chr  "A192" "A191" "A191" "A191" ...
```

```
## Checking missing values ##
```

```
> sum (is.na (BankLoan_DF))
[1] 0
```

```
## Checking duplicated values ##
```

```
> sum (duplicated (BankLoan_DF))
[1] 0
```

```
## Checking distribution of values in variables ##
```

Sudipto Mitra
https://data.world/sudmitra


```
> table (BankLoan_DF $checkingstatus1)

A11 A12 A13 A14
274 269  63 394
>
> table (BankLoan_DF $history)

A30 A31 A32 A33 A34
 40  49 530  88 293
>
> table (BankLoan_DF $purpose)

 A40  A41 A410  A42  A43  A44  A45  A46  A48  A49
 234  103   12  181  280   12   22   50    9   97
>
> table (BankLoan_DF $savings)

A61 A62 A63 A64 A65
603 103  63  48 183
```

```
> table (BankLoan_DF $employ)

A71 A72 A73 A74 A75
 62 172 339 174 253
>
> table (BankLoan_DF $otherplans)

A141 A142 A143
 139   47  814
>
> table (BankLoan_DF $tele)

A191 A192
 596  404
```

```
## Changing data types to factor ##
```

```
> BankLoan_DF $checkingstatus1 = as.factor (BankLoan_DF $checkingstatus1)
>
> BankLoan_DF $history = as.factor (BankLoan_DF $history)
>
> BankLoan_DF $purpose = as.factor (BankLoan_DF $purpose)
>
> BankLoan_DF $savings = as.factor (BankLoan_DF $savings)
>
> BankLoan_DF $employ = as.factor (BankLoan_DF $employ)
>
> BankLoan_DF $otherplans = as.factor (BankLoan_DF $otherplans)
>
> BankLoan_DF $tele = as.factor (BankLoan_DF $tele)
>
> BankLoan_DF $Default = as.factor (BankLoan_DF $Default)

## Re-checking data structures of the variables ##
```

```
> str (BankLoan_DF)
'data.frame':	1000 obs. of  14 variables:
 $ Default        : Factor w/ 2 levels "0","1": 1 2 1 1 2 1 1 1 1 2 ...
 $ checkingstatus1: Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...
 $ duration       : int  6 48 12 42 24 36 24 36 12 30 ...
 $ history        : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 ...
 $ purpose        : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...
 $ amount         : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
 $ savings        : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...
 $ employ         : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...
 $ installment    : int  4 2 2 2 3 2 3 2 2 4 ...
 $ residence      : int  4 2 3 4 4 4 4 2 4 2 ...
 $ age            : int  67 22 49 45 53 35 53 35 61 28 ...
 $ otherplans     : Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ cards          : int  2 1 1 1 2 1 1 1 1 2 ...
 $ tele           : Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...
```

```
## Splitting into training/testing subset ##

set.seed (123)

Sampl_sz = floor (0.8 * nrow (BankLoan_DF))
```

```
> Sampl_sz
[1] 800
```

```
train_ind = sample (seq_len (nrow (BankLoan_DF)), size = Sampl_sz)

train_data = BankLoan_DF [train_ind, ]

test_data = BankLoan_DF [ - train_ind, ]
```

```
> dim (train_data)
[1] 800  14
> dim (test_data)
[1] 200  14
```

```
## Model Building ##

naive_model = naiveBayes (Default ~ ., data = train_data)
```

Sudipto Mitra

https://data.world/sudmitra


```
> naive_model

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
     0      1
0.7125 0.2875

Conditional probabilities:
   checkingstatus1
Y          A11        A12        A13        A14
  0 0.20000000 0.24385965 0.07017544 0.48596491
  1 0.44782609 0.34347826 0.05652174 0.15217391
```

```
   duration
Y       [,1]     [,2]
  0 19.18421 11.37766
  1 24.58696 13.43881

   history
Y          A30        A31        A32        A33        A34
  0 0.01929825 0.02982456 0.52807018 0.07543860 0.34736842
  1 0.07391304 0.10000000 0.56086957 0.08695652 0.17826087

   purpose
Y           A40         A41        A410         A42         A43         A44         A45         A46         A48
  0 0.205263158 0.126315789 0.010526316 0.173684211 0.315789474 0.008771930 0.015789474 0.043859649 0.007017544
  1 0.291304348 0.056521739 0.013043478 0.191304348 0.226086957 0.017391304 0.030434783 0.069565217 0.004347826
   purpose
Y           A49
  0 0.092982456
  1 0.100000000
```

```
   amount
Y       [,1]     [,2]
  0 3010.340 2462.639
  1 3894.574 3566.695

   savings
Y          A61        A62        A63        A64        A65
  0 0.55614035 0.10175439 0.07719298 0.05789474 0.20701754
  1 0.73913043 0.10869565 0.03043478 0.02608696 0.09565217

   employ
Y          A71        A72        A73        A74        A75
  0 0.05964912 0.14035088 0.33508772 0.20000000 0.26491228
  1 0.07391304 0.23478261 0.35652174 0.13043478 0.20434783

   installment
Y       [,1]     [,2]
  0 2.905263 1.136458
  1 3.104348 1.100805
```

Domain – Banking & Finance

Sudipto Mitra

https://data.world/sudmitra


```
   residence
Y       [,1]     [,2]
  0 2.836842 1.120057
  1 2.830435 1.078546

   age
Y       [,1]     [,2]
  0 36.18421 11.41467
  1 34.56957 11.50249

   otherplans
Y         A141       A142       A143
  0 0.12982456 0.04385965 0.82631579
  1 0.18695652 0.05217391 0.76086957

   cards
Y       [,1]      [,2]
  0 1.426316 0.5890108
  1 1.339130 0.5431102

   tele
Y        A191      A192
  0 0.5736842 0.4263158
  1 0.6260870 0.3739130
```

```
### Observation - The model creates conditional probability for each feature
### separately. It also returns Apriori probability indicating data
### distribution in which the model shows that 71.25% are loan defaulters,
### as per training dataset. The model also shows that the average in loan
### default is 20 with variability/standard deviation of 11.38. Likewise, it
### also depicts average duration in repayment is 25 with standard deviation of
### 13.44. It is also observed that the average amount of loan default is
### 3010.34 with std of 2463. The average amount of loan repayment is 3895 with
### std of 3567. The average age of loan defaulters is 36 yrs with std of 11.41.
### The average age of loan repayers is 35 yrs with std of 12.
### The probability of loan default for "A32" category (history) is
### highest with 53% & the probability of repayment is highest for "A31"
### (history) with 100%.
```

```
> summary (naive_model)
          Length Class  Mode
apriori    2     table  numeric
tables    13     -none- list
levels     2     -none- character
isnumeric 13     -none- logical
call       4     -none- call
```

```
## Predictions ##

naive_preds = predict (naive_model, test_data)
```

Domain – Banking & Finance

Sudipto Mitra

https://data.world/sudmitra


```
> head (naive_preds, 20)
 [1] 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0
Levels: 0 1
> table (naive_preds, test_data $Default)

naive_preds   0   1
          0 113  36
          1  17  34
```

```
### Observation - The above tabular display shows that the model predicts
### 113 nos. of correct defaulters & 36 nos. of misprediction as loan
### repayer which are actually recorded as defaulters in the observational
### dataset. Similarly, it predicts 34 nos. of repayers correctly but 17 nos.
### as defaulters wrongly who repaid their loans actually. ###

## Accuracy Metric ##

ACC = 100 * sum (test_data [, 1] == naive_preds) / NROW (test_data [, 1])
```

```
> ACC
[1] 73.5

> confusionMatrix (naive_preds, test_data [, 1])
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 113  36
         1  17  34

               Accuracy : 0.735
                 95% CI : (0.6681, 0.7948)
    No Information Rate : 0.65
    P-Value [Acc > NIR] : 0.006371

                  Kappa : 0.3787

 Mcnemar's Test P-Value : 0.013418

            Sensitivity : 0.8692
            Specificity : 0.4857
         Pos Pred Value : 0.7584
         Neg Pred Value : 0.6667
             Prevalence : 0.6500
         Detection Rate : 0.5650
   Detection Prevalence : 0.7450
      Balanced Accuracy : 0.6775

       'Positive' Class : 0
```

```
> confusionMatrix (table (naive_preds, test_data $Default))
Confusion Matrix and Statistics


naive_preds   0   1
          0 113  36
          1  17  34

               Accuracy : 0.735
                 95% CI : (0.6681, 0.7948)
    No Information Rate : 0.65
    P-Value [Acc > NIR] : 0.006371

                  Kappa : 0.3787

 Mcnemar's Test P-Value : 0.013418

            Sensitivity : 0.8692
            Specificity : 0.4857
         Pos Pred Value : 0.7584
         Neg Pred Value : 0.6667
             Prevalence : 0.6500
         Detection Rate : 0.5650
   Detection Prevalence : 0.7450
      Balanced Accuracy : 0.6775

       'Positive' Class : 0
```

```
### Observation - Accuracy of the model in predictions is 73.5%.
### The above tabular display shows that the model predicts
### 113 nos. of correct defaulters & 36 nos. of misprediction as loan
### repayer which are actually recorded as defaulters in the observational
### dataset. Similarly, it predicts 34 nos. of repayers correctly but 17 nos.
### as defaulters wrongly who repaid their loans actually.
### It returns 95% CI indicating that the accuracy of predictions would range
### between 66.81% & 79.48%. This confirms that there's only 5% probability
### that the accuracy rate of predictions would fall beyond the above
### mentioned range of 95% CI. In short, the model would return predictions
### with accuracy rate of 66.81%, at least, for 95% of times.
### As the P-value < 0.05, the model is statistically significant.
### The Kappa metric denotes that, say, if a layman is asked to predict with
### this model, there'd be least 38% accuracy rate. In general, higher Kappa
### value close to 1 is a good metric for model.
### A concerning observation is that there's much difference in the values of
### Sensitivity & Specificity indicating high probability of mispredictions
### for the model developed. Specificity & Sensitivity values close to equal
### to each other is an ideal metric for the model. ###
```

```
## Model Tuning ##

Loan_classifier = naiveBayes (train_data, train_data [, 1], laplace = 1)
```

Domain – Banking & Finance

Sudipto Mitra

https://data.world/sudmitra


```
> Loan_classifier

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = train_data, y = train_data[, 1], laplace = 1)

A-priori probabilities:
train_data[, 1]
     0      1
0.7125 0.2875

Conditional probabilities:
               Default
train_data[, 1]           0           1
              0 0.998251748 0.001748252
              1 0.004310345 0.995689655
```

```
               checkingstatus1
train_data[, 1]        A11        A12        A13        A14
              0 0.20034843 0.24390244 0.07142857 0.48432056
              1 0.44444444 0.34188034 0.05982906 0.15384615

               duration
train_data[, 1]     [,1]     [,2]
              0 19.18421 11.37766
              1 24.58696 13.43881

               history
train_data[, 1]        A30        A31        A32        A33        A34
              0 0.02086957 0.03130435 0.52521739 0.07652174 0.34608696
              1 0.07659574 0.10212766 0.55319149 0.08936170 0.17872340
```

```
               purpose
train_data[, 1]         A40         A41        A410         A42         A43         A44         A45         A46
              0 0.203448276 0.125862069 0.012068966 0.172413793 0.312068966 0.010344828 0.017241379 0.044827586
              1 0.283333333 0.058333333 0.016666667 0.187500000 0.220833333 0.020833333 0.033333333 0.070833333
               purpose
train_data[, 1]         A48         A49
              0 0.008620690 0.093103448
              1 0.008333333 0.100000000

               amount
train_data[, 1]     [,1]     [,2]
              0 3010.340 2462.639
              1 3894.574 3566.695
```

```
               savings
train_data[, 1]        A61        A62        A63        A64        A65
              0 0.55304348 0.10260870 0.07826087 0.05913043 0.20695652
              1 0.72765957 0.11063830 0.03404255 0.02978723 0.09787234

               employ
train_data[, 1]        A71        A72        A73        A74        A75
              0 0.06086957 0.14086957 0.33391304 0.20000000 0.26434783
              1 0.07659574 0.23404255 0.35319149 0.13191489 0.20425532

               installment
train_data[, 1]     [,1]     [,2]
              0 2.905263 1.136458
              1 3.104348 1.100805
```

Sudipto Mitra

https://data.world/sudmitra


```
                 residence
train_data[, 1]     [,1]      [,2]
                0 2.836842 1.120057
                1 2.830435 1.078546

                 age
train_data[, 1]     [,1]      [,2]
                0 36.18421 11.41467
                1 34.56957 11.50249

                 otherplans
train_data[, 1]        A141        A142        A143
                0 0.13089005 0.04537522 0.82373473
                1 0.18884120 0.05579399 0.75536481

                 cards
train_data[, 1]     [,1]       [,2]
                0 1.426316 0.5890108
                1 1.339130 0.5431102

                 tele
train_data[, 1]      A191      A192
                0 0.5734266 0.4265734
                1 0.6250000 0.3750000
```

```
preds_tune = predict (Loan_classifier, test_data)
```

```
## Re-checking Model Accuracy Metrics ##

> table (preds_tune, test_data $Default)

preds_tune   0   1
         0 129   0
         1   1  70
```

```
### Observation - The resulting table shows only 1 no. of misclassification
### on the test data by the model. ###
```

Sudipto Mitra
https://data.world/sudmitra


```
> confusionMatrix (preds_tune, test_data [, 1])
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 129   0
         1   1  70

               Accuracy : 0.995
                 95% CI : (0.9725, 0.9999)
    No Information Rate : 0.65
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.989

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9923
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9859
             Prevalence : 0.6500
         Detection Rate : 0.6450
   Detection Prevalence : 0.6450
      Balanced Accuracy : 0.9962

       'Positive' Class : 0
```

```
### Observation - Upon tuning of parameters, there's remarkable improvement in
### all metrics of the model. The accuracy rate is increased to 99.5%.
### The resulting table shows only 1 no. of misclassification on the test data
### by the model. ###
```