Credit risk rating classification with R

KLASIFIKASI merupakan salah satu metode dari *supervised learning,* yang dapat diartikan sebagai suatu algortima atau Teknik yang dapat digunakan untuk membuat suatu skema atau kategori data yang berlabel. Proses klasifikasi terdiri dari sua tahap, yaitu:

1. *LEARNING OF MAPPING* →proses pemetaan untuk membuat prediksi *class label* pada data menggunakan suatu *classification rules*
2. *CLASSIFIER* → suatu tahapan pengklasifikasian atau pelabelan data

Dalam melakukan pengklasifikasian diperlukan dua data, yaitu *data train* atau data yang digunakan dalam proses *learning of mapping, dan data test* atau data yang akan digunakan pada proses *classifier. Data train* dan *data test* ini bersifat independen terhadap satu sama lain. Terdapat beberapa algoritma terkenal yang sering digunakan dalam pengklasifikasian, antara lain *naïve bayes, k-nearest neighbor, decision tree, support vector machine, dan random forest.*

STUDI KASUS : PENENTUAN RISK RATING PADA KARYAWAN

Sebagai Data Scientist keuangan di perusahaan, perlu untuk mempertimbangkan pemberian pinjaman pada karyawan. Sehingga perlu untuk memberikan ***risk rating*** atau penilaian risiko yang dilihat dari riwayat finansial, kewajiban, dan asset yang dimiliki oleh individu tersebut.

Pada studi kasus ini, digunakan data yang dapat diperoleh disini. Pada data tersebut, terdapat 7 kolom, dengan 900 baris data yang akan digunakan dalam pengklasifikasian. Dalam prosesnya, hanya terdapat 3 kolom yang akan digunakan datanya untuk melakukan klasifikasi yaitu kolom pendapatan_setahun_juta, durasi_pinjaman_bulan, dan jumlah_tanggungan. Dalam data tersebut, terdapat suatu kolom "risk_rating" yang merupakan pelabelan yang telah dilakukan sebelumnya, yang kemudian akan digunakan oleh program untuk melakukan *learning mapping,* dan juga sebagai referensi saat melakukan *testing* pada model yang digunakan.

Pada studi kasus ini, digunakan *r programming* dan library

```
#library importing
> library(e1071)
> library(caret)
> library(dplyr)
```

creator:
LUTFIA HUSNA KHOIRUNNISA
linkedin.com/in/lutfiahusnakhoirunnisa | lutfiahusnakhoirunnisa@gmail.com | lynk.id/lutfiahusnak

Tahap 1. Data *preparation*

Tahapan ini bertujuan untuk mempersiapkan data sebelum dilakukan *modelling*. Pada studi kasus ini, data yang digunakan disimpan dalam suatu *variable "df"*

Data exploration

```
#mengetahui dimensi dari data
> dim(df)
[1] 900   7

#mengetahui struktur dari suatu data frame
> str(df)
 tibble [900 × 7] (S3: tbl_df/tbl/data.frame)
 $ kode_kontrak         : chr [1:900] "AGR-000001" "AGR-000011" "AGR-000030" "AGR-
000043" ...
 $ pendapatan_setahun_juta: num [1:900] 295 271 159 210 165 220 70 88 163 100 ...
 $ kpr_aktif            : chr [1:900] "YA" "YA" "TIDAK" "YA" ...
 $ durasi_pinjaman_bulan  : num [1:900] 48 36 12 12 36 24 36 48 48 36 ...
 $ jumlah_tanggungan      : num [1:900] 5 5 0 3 0 5 3 3 5 6 ...
 $ rata_rata_overdue      : chr [1:900] "61 - 90 days" "61 - 90 days" "0 - 30 days" "46 - 60 days"
...
 $ risk_rating          : num [1:900] 4 4 1 3 2 1 2 2 2 2 ...

#mengetahui nilai statistic deskriptif dari tiap variable data
> summary(df)
 kode_kontrak         pendapatan_setahun_juta   kpr_aktif
 Length:900           Min.   : 70.0           Length:900
 Class :character     1st Qu.:121.0           Class :character
 Mode  :character     Median :162.0           Mode  :character
                                              Mean   :163.3
                                              3rd Qu.:199.0
                                              Max.   :300.0
 durasi_pinjaman_bulan  jumlah_tanggungan  rata_rata_overdue   risk_rating
 Min.   :12.00          Min.   :0.000      Length:900          Min.   :1.000
 1st Qu.:12.00          1st Qu.:1.000      Class :character    1st Qu.:1.000
 Median :24.00          Median :3.000      Mode  :character    Median :3.000
 Mean   :29.93          Mean   :2.932                          Mean   :2.681
 3rd Qu.:48.00          3rd Qu.:5.000                          3rd Qu.:3.000
 Max.   :48.00          Max.   :6.000                          Max.   :5.000
```

Selanjutnya, karena pada klasifikasi ini hanya akan digunakan data dari *variable "pendapatan_setahun_juta", "durasi_pinjaman_bulan", "jumlah_tanggungan",* dan *"risk_rating",* maka kita akan memisahkan data yang digunakan dan disimpan dalam *variable "datarisk".*

```
#mengambil index variable yang diperlukan
> index<-c("pendapatan_setahun_juta", "durasi_pinjaman_bulan", "jumlah_tanggungan",
"risk_rating")

#menyimpan data yang akan digunakan dalam variable baru
> datarisk<-df[index]

#mengubah tipe data pada variable risk_rating menjadi bentuk factor
> datarisk$risk_rating<-as.factor(datarisk$risk_rating)

# mengetahui struktur dari suatu data frame
> str(datarisk)
tibble [900 × 4] (S3: tbl_df/tbl/data.frame)
 $ pendapatan_setahun_juta: num [1:900] 295 271 159 210 165 220 70 88 163 100 ...
 $ durasi_pinjaman_bulan  : num [1:900] 48 36 12 12 36 24 36 48 48 36 ...
 $ jumlah_tanggungan      : num [1:900] 5 5 0 3 0 5 3 3 5 6 ...
 $ risk_rating            : Factor w/ 5 levels "1","2","3","4",..: 4 4 1 3 2 1 2 2 2 2 ...

#melihat data teratas dari dataframe datarisk
> head(datarisk)
# A tibble: 6 × 4
  pendapatan_setahun_juta durasi_pinjaman_bulan jumlah_tanggungan risk_rating
                    <dbl>                 <dbl>              <dbl> <fct>
1                     295                    48                  5 4
2                     271                    36                  5 4
3                     159                    12                  0 1
4                     210                    12                  3 3
5                     165                    36                  0 2
6                     220                    24                  5 1
```

Dalam melakukan pemodelan klasifikasi, perlu untuk menyiapkan data *train* dan data *test.* Pada studi kasus ini, data yang telah dipersiapkan sebelumnya akan dibagi menjadi 80% data *train,* dan 20% data test.

```
#datasplitting
> set.seed(123)
> index<-sample(1:nrow(datarisk), size = 0.8*nrow(datarisk)) #80%data digunakan sbg data
train
> datatrain<-datarisk[index,]
> datatest<-datarisk[-index,]

#mengetahui dimensi dari data train dan data test
> dim(datatrain)
[1] 720   4
> dim(datatest)
[1] 180   4
```

Tahap 2. *Classification*

Setelah *data train* dan *data test* siap, kemudian akan dibentuk suatu model untuk melakukan klasifikasi data. Pada studi kasus ini, akan digunakan beberapa metode pengklasifikasian, antara lain *naïve bayes, random forest, decision tree, k-nearest neighbor* dan *support vector machine.*

```
#classification modeling
##using naive bayes method
> modelnaive<-naiveBayes(x=datatrain %>% select(-risk_rating), y=datatrain$risk_rating)

> ###testing the model
> predictnaive<-predict(modelnaive, datatest, type="class")
> tbpredictnaive <- data.frame(datatest,predictnaive)
> head(tbpredictnaive)
  pendapatan_setahun_juta durasi_pinjaman_bulan jumlah_tanggungan risk_rating
1                     295                    48                  5           4
2                     159                    12                  0           1
3                      70                    36                  3           2
4                     163                    36                  0           2
5                     208                    36                  0           1
6                      84                    12                  3           2
  predictnaive
1            4
2            1
3            3
4            1
5            1
6            3
```

Pada data frame "*tbpredictnaive*", dapat dilihat klasifikasi awal dari *data test* yang dalam kolom "*risk_rating*", dan klasifikasi hasil dari prediksi dalam kolom "*predictnaive*". dari hasil prediksi yang diperoleh, dapat dilakukan test dengan *confussionMatrix* untuk menilai keakuratan hasil prediksi klasifikasi.

```
> naiveeval<- confusionMatrix(data=predictnaive, reference=datatest$risk_rating)
> naiveeval
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 33  8  2  0  0
         2  1  7  3  0  0
         3  2  8 52  0  0
         4  1  0  0 32  1
         5  1  2  0  3 24

Overall Statistics

               Accuracy : 0.8222
                 95% CI : (0.7584, 0.8751)
    No Information Rate : 0.3167
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7698

 Mcnemar's Test P-Value : NA
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity          0.8684  0.28000   0.9123   0.9143   0.9600
Specificity          0.9296  0.97419   0.9187   0.9862   0.9613
Pos Pred Value       0.7674  0.63636   0.8387   0.9412   0.8000
Neg Pred Value       0.9635  0.89349   0.9576   0.9795   0.9933
Prevalence           0.2111  0.13889   0.3167   0.1944   0.1389
Detection Rate       0.1833  0.03889   0.2889   0.1778   0.1333
Detection Prevalence 0.2389  0.06111   0.3444   0.1889   0.1667
Balanced Accuracy    0.8990  0.62710   0.9155   0.9502   0.9606
```

Dari hasil *confussionMatrix* tersebut diketahui bahwa keakuratan hasil prediksi yang diperoleh sebesar 0.8222, dengan data yang salah prediksi disajikan dalam matrix:

confussionMatrix

	Reference				
Prediction	1	2	3	4	5
1	33	8	2	0	0
2	1	7	3	0	0
3	2	8	52	0	0
4	1	0	0	32	1
5	1	2	0	3	24

Dari *confussionMatrix* tersebut dapat diketahui bahwa terdapat 1 data yang seharusnya dikasifikasikan sebagai kelas "1", tetapi diprediksikan berada pada kelas "2". Terdapat 2 data yang yang seharusnya dikasifikasikan sebagai kelas "1", tetapi diprediksikan berada pada kelas "3", dst.

Untuk selanjutnya, digunakan metode klasifikasi *random forest, decision tree,* dan *support vector machine.* Akan disajikan hanya hasil confusion untuk membandingkan hasil prediksi dari semua metode:

random forest:

```
> rfeval<- confusionMatrix(data=predictrf, reference=datatest$risk_rating)
> rfeval
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 33  7  0  0  0
         2  3 12  2  0  3
         3  0  5 55  0  0
         4  1  0  0 34  0
         5  1  1  0  1 22

Overall Statistics

               Accuracy : 0.8667
                 95% CI : (0.8081, 0.9127)
    No Information Rate : 0.3167
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.828

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            0.8684  0.48000   0.9649   0.9714   0.8800
Specificity            0.9507  0.94839   0.9593   0.9931   0.9806
Pos Pred Value         0.8250  0.60000   0.9167   0.9714   0.8800
Neg Pred Value         0.9643  0.91875   0.9833   0.9931   0.9806
Prevalence             0.2111  0.13889   0.3167   0.1944   0.1389
Detection Rate         0.1833  0.06667   0.3056   0.1889   0.1222
Detection Prevalence   0.2222  0.11111   0.3333   0.1944   0.1389
Balanced Accuracy      0.9096  0.71419   0.9621   0.9823   0.9303
```

decision tree:

```
> confusionMatrix(data = predictdt, reference=datatest$risk_rating)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 32  9  0  0  0
         2  4  9  2  0  0
         3  0  5 55  0  0
         4  1  0  0 35  0
         5  1  2  0  0 25

Overall Statistics

               Accuracy : 0.8667
                 95% CI : (0.8081, 0.9127)
    No Information Rate : 0.3167
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8278

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity          0.8421  0.36000   0.9649   1.0000   1.0000
Specificity          0.9366  0.96129   0.9593   0.9931   0.9806
Pos Pred Value       0.7805  0.60000   0.9167   0.9722   0.8929
Neg Pred Value       0.9568  0.90303   0.9833   1.0000   1.0000
Prevalence           0.2111  0.13889   0.3167   0.1944   0.1389
Detection Rate       0.1778  0.05000   0.3056   0.1944   0.1389
Detection Prevalence   0.2278  0.08333   0.3333   0.2000   0.1556
Balanced Accuracy      0.8894  0.66065   0.9621   0.9966   0.9903
```

support vector machine:

```
> svmeval<- confusionMatrix(data=predictsvm, reference=datatest$risk_rating)
> svmeval
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 33  6  0  0  0
         2  3 13  2  0  0
         3  0  5 55  0  0
         4  1  0  0 35  0
         5  1  1  0  0 25

Overall Statistics

               Accuracy : 0.8944
                 95% CI : (0.8401, 0.9352)
    No Information Rate : 0.3167
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8638

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity          0.8684  0.52000   0.9649   1.0000   1.0000
Specificity          0.9577  0.96774   0.9593   0.9931   0.9871
Pos Pred Value       0.8462  0.72222   0.9167   0.9722   0.9259
Neg Pred Value       0.9645  0.92593   0.9833   1.0000   1.0000
Prevalence           0.2111  0.13889   0.3167   0.1944   0.1389
Detection Rate       0.1833  0.07222   0.3056   0.1944   0.1389
Detection Prevalence   0.2167  0.10000   0.3333   0.2000   0.1500
Balanced Accuracy      0.9131  0.74387   0.9621   0.9966   0.9935
```

k-nearest neighbor:

```
> confusionMatrix(data = predictknn, reference=datatest$risk_rating)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 32  8  0  0  0
         2  4 10  2  1  1
         3  0  6 55  0  0
         4  1  0  0 31  1
         5  1  1  0  3 23

Overall Statistics

               Accuracy : 0.8389
                 95% CI : (0.7769, 0.8894)
    No Information Rate : 0.3167
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.792

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            0.8421  0.40000   0.9649   0.8857   0.9200
Specificity            0.9437  0.94839   0.9512   0.9862   0.9677
Pos Pred Value         0.8000  0.55556   0.9016   0.9394   0.8214
Neg Pred Value         0.9571  0.90741   0.9832   0.9728   0.9868
Prevalence             0.2111  0.13889   0.3167   0.1944   0.1389
Detection Rate         0.1778  0.05556   0.3056   0.1722   0.1278
Detection Prevalence   0.2222  0.10000   0.3389   0.1833   0.1556
Balanced Accuracy      0.8929  0.67419   0.9581   0.9360   0.9439
```

Dari model-model yang telah dibuat, diperoleh nilai keakuratan prediksi sebagai berikut:

1. *Naïve bayes* : 0.8222
2. *Support vector machine* : 0.8944
3. *Random forest :* 0.8667
4. *K-nearest neighbor* : 0.8389
5. *Decision tree* : 0.8667

Sehingga, dapat diperoleh kesimpulan bahwa metode klasifikasi yang paling tepat untuk digunakan dalam pengklasifikasian data pemberian pinjaman karyawan ini adalah dengan metode *Support vector machine* dengan nilai keakuratan 0.8944.