SVMs For Banking Data

Ada Lazuli

2022-09-06

Contents

Libraries

```
library(ggplot2)
library(ggcorrplot)
library(dplyr)
library(tidyverse)
library(tidymodels)
library(e1071)
set.seed(1)
```

PreProcessing

```
df <- read.csv("credit card default data set.csv")

df$default <- as.factor(df$default.payment.next.month)
df$default.payment.next.month <- NULL

df$is_female <- as.factor(as.integer(df$SEX == 2))
df$SEX <- NULL
df$is_married <- as.factor(as.integer(df$MARRIAGE == 1))
df$is_single <- as.factor(as.integer(df$MARRIAGE == 2))
df$is_married_other <- as.factor(as.integer(df$MARRIAGE == 0 | df$MARRIAGE == 3))
df$MARRIAGE <- NULL

df$is_edu_other <- as.factor(as.integer(df$EDUCATION == 0 | df$EDUCATION > 3))
df$is_edu_grad <- as.factor(as.integer(df$EDUCATION == 1))
df$is_edu_uni <- as.factor(as.integer(df$EDUCATION == 2))
df$is_edu_hs <- as.factor(as.integer(df$EDUCATION == 3))
df$EDUCATION <- NULL


df <- df %>% pivot_longer(c(PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6), "Month") %>%
  rename(HIST = value) %>%
  select(LIMIT_BAL,
         is_edu_grad, is_edu_hs, is_edu_other, is_edu_uni,
         is_female, is_married, is_married_other, is_single,
          HIST, default)

df <- df %>% group_by(default) %>% slice_sample(prop = 0.3) %>% ungroup()
df$LIMIT_BAL <- scale(df$LIMIT_BAL)
df$HIST <- scale(df$HIST)
```

Train, Validation, and Test Split

```
first_split <- initial_split(df, prop = .7, strata = default)
trainset <- training(first_split)
second_split <- initial_split(testing(first_split), prop = .5, strata = default)
validationset <- training(second_split)
testset <- testing(second_split)
```

SVM 1 -Linear

Creation

```
svm1 <- svm(default ~ .,
            data = trainset,
            type = "C-classification",
            kernel = "linear",
            scale = FALSE)
```

Performance

```
train_accuracy1 <- mean(predict(svm1, trainset) == trainset$default)
val_accuracy1 <- mean(predict(svm1, validationset) == validationset$default)
```

```
predictions <- predict(svm1, validationset)
TP <-  sum( predictions == 1  & validationset$default == 1)
FP <-  sum( predictions == 0 &  validationset$default == 1)
FN <-  sum( predictions == 1 &  validationset$default == 0)
TN <-  sum( predictions == 0 &  validationset$default == 0)
precision <- (TP)/(TP + FP)
recall <- (TP)/(TP + FN)
```

	Test Real	Test Fake
Classified Real	0	1792
Classified Fake	0	6308

1. Accuracy: 0.7787654
2. Precision: 0
3. Recall: NaN
4. F1 Score: NaN

The accuracies against the training and validation sets is:

1. Train: 0.7788243
2. Validation: 0.7787654

SVM 2 -Polynomial Degree 3

Creation

```
svm2<- svm(default ~ .,
           data = trainset,
           type = "C-classification",
           kernel = "polynomial",
           degree = 3,
           scale = FALSE)
```

Performance

```
train_accuracy2 <- mean(predict(svm2, trainset) == trainset$default)
val_accuracy2 <- mean(predict(svm2, validationset) == validationset$default)
```

The accuracies against the training and validation sets is:

1. Train: 0.7954125
2. Validation: 0.7948148

```
predictions <- predict(svm2, validationset)
TP <-  sum( predictions == 1  & validationset$default == 1)
FP <-  sum( predictions == 0 &  validationset$default == 1)
FN <-  sum( predictions == 1 &  validationset$default == 0)
TN <-  sum( predictions == 0 &  validationset$default == 0)
precision <- (TP)/(TP + FP)
recall <- (TP)/(TP + FN)
```

	Test Real	Test Fake
Classified Real	423	1369
Classified Fake	293	6015

1. Accuracy: 0.7948148
2. Precision: 0.2360491
3. Recall: 0.5907821
4. F1 Score: 0.3373206

SVM 3 - Polynomial Degree 4

Creation

```
svm3 <- svm(default ~ .,
            data = trainset,
            type = "C-classification",
            kernel = "polynomial",
            degree = 4,
            scale = FALSE)
```

Performance

```
train_accuracy3 <- mean(predict(svm3, trainset) == trainset$default)
val_accuracy3 <- mean(predict(svm3, validationset) == validationset$default)
```

The accuracies against the training and validation sets is:

1. Train: 0.7945923
2. Validation: 0.794321

```
predictions <- predict(svm3, validationset)
TP <-  sum( predictions == 1  & validationset$default == 1)
FP <-  sum( predictions == 0 &  validationset$default == 1)
FN <-  sum( predictions == 1 &  validationset$default == 0)
TN <-  sum( predictions == 0 &  validationset$default == 0)
precision <- (TP)/(TP + FP)
recall <- (TP)/(TP + FN)
```

	Test Real	Test Fake
Classified Real	411	1381
Classified Fake	285	6023

1. Accuracy: 0.794321
2. Precision: 0.2293527
3. Recall: 0.5905172
4. F1 Score: 0.3303859

SVM 4 - Polynomial Degree 2

Creation

```
svm4 <- svm(default ~ .,
            data = trainset,
            type = "C-classification",
            kernel = "polynomial",
            degree = 2,
            scale = FALSE)
```

Performance

```
train_accuracy4 <- mean(predict(svm4, trainset) == trainset$default)
val_accuracy4 <- mean(predict(svm4, validationset) == validationset$default)
```

The accuracies against the training and validation sets is:

1. Train: 0.7954654
2. Validation: 0.7945679

```
predictions <- predict(svm4, validationset)
TP <-  sum( predictions == 1  & validationset$default == 1)
FP <-  sum( predictions == 0 &  validationset$default == 1)
FN <-  sum( predictions == 1 &  validationset$default == 0)
TN <-  sum( predictions == 0 &  validationset$default == 0)
precision <- (TP)/(TP + FP)
recall <- (TP)/(TP + FN)
```

	Test Real	Test Fake
Classified Real	524	1268
Classified Fake	396	5912

1. Accuracy: 0.7945679
2. Precision: 0.2924107
3. Recall: 0.5695652

4. F1 Score: 0.3864307

```
predictions <- predict(svm4, testset)
TP <-  sum( predictions == 1  & validationset$default == 1)
```

```
## Warning in predictions == 1 & validationset$default == 1: longer object length
## is not a multiple of shorter object length
```

```
FP <-  sum( predictions == 0 &  validationset$default == 1)
```

```
## Warning in predictions == 0 & validationset$default == 1: longer object length
## is not a multiple of shorter object length
```

```
FN <-  sum( predictions == 1 &  validationset$default == 0)
```

```
## Warning in predictions == 1 & validationset$default == 0: longer object length
## is not a multiple of shorter object length
```

```
TN <-  sum( predictions == 0 &  validationset$default == 0)
```

```
## Warning in predictions == 0 & validationset$default == 0: longer object length
## is not a multiple of shorter object length
```

```
precision <- (TP)/(TP + FP)
recall <- (TP)/(TP + FN)
```

	Test Real	Test Fake
Classified Real	542	1250
Classified Fake	396	5913

1. Accuracy: 0.7968152
2. Precision: 0.3024554
3. Recall: 0.5778252
4. F1 Score: 0.3970696

SVM 5 - Sigmoid

Creation

```
svm5 <- svm(default ~ .,
            data = trainset,
            type = "C-classification",
            kernel = "sigmoid",
            scale = FALSE)
```

Performance

```
train_accuracy5 <- mean(predict(svm5, trainset) == trainset$default)
val_accuracy5 <- mean(predict(svm5, validationset) == validationset$default)
```

The accuracies against the training and validation sets is:

1. Train: 0.6877613
2. Validation: 0.6964198

```
predictions <- predict(svm5, validationset)
TP <-  sum( predictions == 1  & validationset$default == 1)
FP <-  sum( predictions == 0 &  validationset$default == 1)
FN <-  sum( predictions == 1 &  validationset$default == 0)
TN <-  sum( predictions == 0 &  validationset$default == 0)
precision <- (TP)/(TP + FP)
recall <- (TP)/(TP + FN)
```

	Test Real	Test Fake
Classified Real	478	1314
Classified Fake	1145	5163

1. Accuracy: 0.6964198
2. Precision: 0.2667411
3. Recall: 0.2945163
4. F1 Score: 0.2799414

SVM 6 - Radial

Creation

```
svm6<- svm(default ~ .,
           data = trainset,
           type = "C-classification",
           kernel = "radial",
           scale = FALSE)
```

Performance

```
train_accuracy6 <- mean(predict(svm6, trainset) == trainset$default)
val_accuracy6 <- mean(predict(svm6, validationset) == validationset$default)
```

The accuracies against the training and validation sets is:

1. Train: 0.7962855
2. Validation: 0.7946914

```
predictions <- predict(svm6, validationset)
TP <-  sum( predictions == 1  & validationset$default == 1)
FP <-  sum( predictions == 0 &  validationset$default == 1)
FN <-  sum( predictions == 1 &  validationset$default == 0)
TN <-  sum( predictions == 0 &  validationset$default == 0)
precision <- (TP)/(TP + FP)
recall <- (TP)/(TP + FN)
```

	Test Real	Test Fake
Classified Real	507	1285
Classified Fake	378	5930

1. Accuracy: 0.7946914
2. Precision: 0.2829241
3. Recall: 0.5728814
4. F1 Score: 0.3787822