SVM

Adrian Jenkins

9/4/2021

The data contains 1470 observations and 35 attributes. The purpose of this model will be to predict 'Attrition' attribute, which refers to whether or not the employee leaves the company. For the SVM model we will be using as independent variables the followings: 'Monthly Income', 'Age', 'Overtime' and 'Job Level'.

This algorithm only accepts independent variables of numerical type, for this reason, the procedure known as "One Hot Encoding" must be performed. This technique consists of transforming each level of a factor into an independent column with values "1" or "0" depending on the presence or absence of the attribute.

The data set is partitioned with 75% of the observations for the training set and the remaining 25% for the test set. The model is then generated

```
dummy <- dummyVars( Attrition ~., data = attrition)
attrition_SVG <- data.frame(predict(dummy, newdata = attrition))
attrition_SVG %<>%
  select(-"OverTimeNo") %>%
  rename("OverTime" = "OverTimeYes")
attrition_SVG %<>%
  cbind(Attrition = attrition$Attrition)
attrition_SVG$Attrition <- factor(attrition_SVG$Attrition, ordered = TRUE,levels = c("No", "Yes"))

# Data Partioning
 set.seed(1)
  training.ids <- createDataPartition(attrition_SVG$Attrition, p = 0.75, list = F)
  mod <- svm(Attrition ~ ., data = attrition_SVG[training.ids, ])
```

```
##
## Call:
## svm(formula = Attrition ~ ., data = attrition_SVG[training.ids, ])
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  478
##
##  ( 170 308 )
##
##
## Number of Classes:  2
##
```

```
## Levels:
##  No Yes
```

Prediction is made:

```
##       Predicted
## Actual  No Yes
##    No  297  11
##    Yes  45  14
```

```
## [1] "Accuracy: 0.847411"
```

```
## [1] "Precision: 0.560000"
```

```
## [1] "Recall: 0.237288"
```

```
## [1] "Specificity: 0.964286"
```

```
## [1] "F1 Score: 0.333333"
```

According to **accuracy**, the model predicts correctly in 84.7% of the cases.

According to **precision**, of all the positive predictions that made the mode, only 56% did left the company.

According to **sensitivity**, of all the people who left the company, the model was correct in 23.7% of the cases.

According to **specificity**, of all the people who did not leave the company, the model correctly predicted 96.4% of the observations.

The **"F1 - Score"** has a value of 33.3%.

We proceed to optimize the model and find the best parameters for:

- 'gamma': defines how far the influence of a single observation in the training set extends. A low value translates into a large influence and vice versa.
- 'cost': compensates the correct classification of the training examples with the maximization of the margin of the decision function. For larger values of "C", a smaller margin will be accepted if the decision function is better at correctly classifying all training points.

```
tuned <- tune.svm(Attrition ~ ., data = attrition_SVG[training.ids, ],
                  gamma = 10^(-6: -1), cost = 10^(1:3))
  summary(tuned)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##    0.1  100
##
```

```
## - best performance: 0.1431941
##
## - Detailed performance results:
##    gamma cost     error dispersion
## 1  1e-06   10 0.1613841 0.02531139
## 2  1e-05   10 0.1613841 0.02531139
## 3  1e-04   10 0.1613841 0.02531139
## 4  1e-03   10 0.1613841 0.02531139
## 5  1e-02   10 0.1613841 0.02531139
## 6  1e-01   10 0.1522686 0.03849859
## 7  1e-06  100 0.1613841 0.02531139
## 8  1e-05  100 0.1613841 0.02531139
## 9  1e-04  100 0.1613841 0.02531139
## 10 1e-03  100 0.1613841 0.02531139
## 11 1e-02  100 0.1577396 0.02741303
## 12 1e-01  100 0.1431941 0.04382868
## 13 1e-06 1000 0.1613841 0.02531139
## 14 1e-05 1000 0.1613841 0.02531139
## 15 1e-04 1000 0.1613841 0.02531139
## 16 1e-03 1000 0.1613841 0.02531139
## 17 1e-02 1000 0.1568059 0.03660892
## 18 1e-01 1000 0.1468305 0.04568928
```

We then proceed to use this values for constructing the model and making the prediction once again:

```
##
## Call:
## svm(formula = Attrition ~ ., data = attrition_SVG[training.ids, ],
##     cost = 1000, gamma = 0.1, class.weights = c(No = 0.3, Yes = 0.7),
##     kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1000
##
## Number of Support Vectors:  504
##
##  ( 145 359 )
##
##
## Number of Classes:  2
##
## Levels:
##  No Yes

##       Predicted
## Actual  No Yes
##    No  283  25
##    Yes  38  21

## [1] "Accuracy: 0.828338"
```

```
## [1] "Precision: 0.456522"
```

```
## [1] "Recall: 0.355932"
```

```
## [1] "Specificity: 0.918831"
```

```
## [1] "F1 Score: 0.400000"
```

However, since the aim is to increase the recall, a "C" of 1000 will be used in order to allow for a greater number of error classifications.

In addition to using these values for the parameters, the "class.weights" parameter is also used as an attempt to mitigate the difference between those who did not drop out and those who did.

According to **accuracy**, the model predicts correctly in 82.8% of the cases.

According to **precision**, of all the positive predictions that made the model, only 45.6% did left the company.

According to **sensitivity**, of all the people who left the company, the model was correct in 35.5% of the cases.

According to **specificity**, of all the people who did not leave the company, the model correctly predicts 91.8% of the observations.

The **"F1 - Score"** has a value of 40%.

Comparing Models

Metric	First Model	Second Model
Accuracy	84.7%	82.3%
Precision	56%	45.6%
Sensibility	23.7%	35.5%
Specificity	96.4%	91.8%
F1-Score	33%	40%

Undoubtedly, the second model fulfills the proposed objective better. Not only does it achieve a better balance between sensitivity and specificity, exceeding the first model by 7%. In addition, of all the people who left the company, it correctly predicts 35.5% of the cases. This represents an improvement of 11.8%.Therefore, since the objective was to correctly predict those who left the company, the second model performs better.

You might say that the value is too low, but that could be for a number of reasons and it might be that the provided independent variables does not really explain the dependent variable. Nonetheless, a 35.5% is better than 0.

Hope you could learned something and feel free to take what you need. The code is in the file "script.R".