Ridge e Lasso

Pré-Processamento

Na terceira parte do Lab 3 foi, novamente, utilizado o data frame dos alunos, iniciando a atividade com um pré-processamento dos dados, para que fossem agrupadas as notas por matrícula e, assim, fosse possível a leitura do rendimento de um determinado aluno nas diferentes disciplinas. Além disso, foram selecionadas somente as variáveis de interesse: média dos alunos nas disciplinas de primeiro e segundo período. Por fim, foi feito o cálculo do coeficiente de rendimento do aluno. De forma que o data frame ficou no seguinte formato:

```
##    X Cálculo.Diferencial.e.Integral.I
## 1 26                              8.7
## 2 28                              7.0
## 3 30                              8.6
## 4 35                              7.8
## 5 41                              5.2
## 6 46                              6.1
##   Álgebra.Vetorial.e.Geometria.Analítica Leitura.e.Produção.de.Textos
## 1                                    8.6                        10.0
## 2                                    5.6                         7.0
## 3                                   10.0                         9.8
## 4                                    6.1                         8.3
## 5                                    8.8                         9.3
## 6                                    9.4                         9.2
##   Programação.I Introdução.à.Computação Laboratório.de.Programação.II
## 1           9.0                     9.1                           9.4
## 2           7.7                     7.0                           8.9
## 3           7.9                     9.6                           9.7
## 4           6.8                     8.2                           9.0
## 5           5.0                     8.5                           8.2
## 6           9.1                     9.3                           9.6
##   Cálculo.Diferencial.e.Integral.II Matemática.Discreta Programação.II
## 1                               8.4                 8.3            8.8
## 2                               6.2                 7.3            8.2
## 3                               8.7                 8.8            9.5
## 4                               8.0                 6.3            8.9
## 5                               5.0                 5.8            7.1
## 6                               5.6                 8.2            9.0
##   Teoria.dos.Grafos Fundamentos.de.Física.Clássica      cra
## 1               8.2                            7.9 8.477647
## 2               5.4                            7.7 6.851724
## 3               9.2                            8.6 9.090588
## 4               7.0                            8.5 7.283516
## 5               5.4                            8.7 7.205747
## 6               8.5                            7.3 7.808235
```

O mesmo tratamento foi feito com os dados de validação, para que depois fosse possível fazer a previsão corretamente e o medir o erro obtido.

Primeiro treino com Ridge e Lasso

O primeiro treino com os algoritmos de regressão foram feitos utilizando todas as variáveis. Utilizando a biblioteca Carrot, foi criada uma tabela de possíveis lambdas a serem utilizados e, com eles, foram feitos os primeiros treinos

```
library(ISLR)
library(caret)

set.seed(825)

fitControl <- trainControl(method = "cv",
                           number = 10)

lambdaGrid <- expand.grid(lambda = 10^seq(-3, -4, length=200))

ridge <- train(cra~., data = graduados.selected,
               method='ridge',
               trControl = fitControl,
               tuneGrid = lambdaGrid,
               preProcess=c('center', 'scale')
)

ridge.pred <- predict(ridge, validacao.cra)
```

Através da previsão que foi obtida é possível, então achar a média dos erros ao quadrado:

```
sqrt(mean(ridge.pred - validacao.cra$cra)^2)
```

```
## [1] 0.06367807
```

O processo então foi repetido com o algoritmo Lasso:

```
lasso <- train(cra ~., graduados.selected,method='lasso',preProc=c('scale','center'),trControl=fitContr
predict.enet(lasso$finalModel, type='coefficients', s=lasso$bestTune$fraction, mode='fraction')
lasso.pred <- predict(lasso, validacao.cra)
sqrt(mean(lasso.pred - validacao.cra$cra)^2)
```

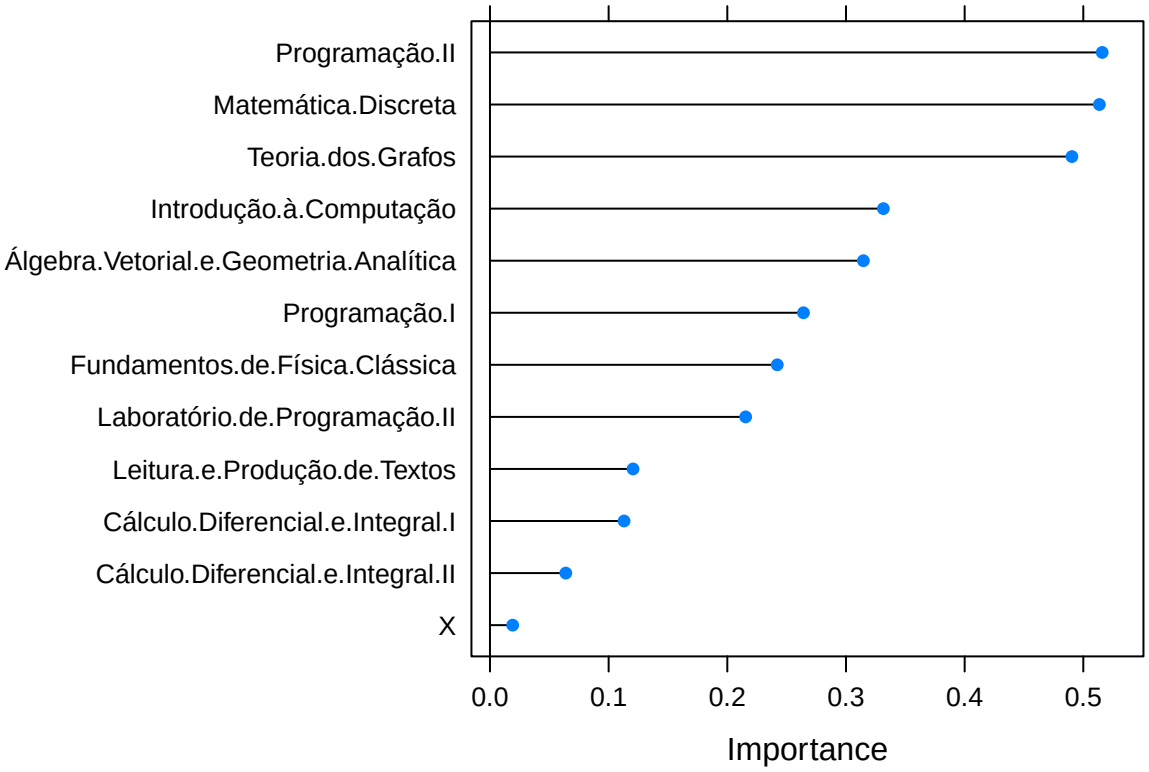
```
## [1] 0.0428245
```

Assim, é possível observar que, em primeira instância, o Lasso obteve melhor desempenho.

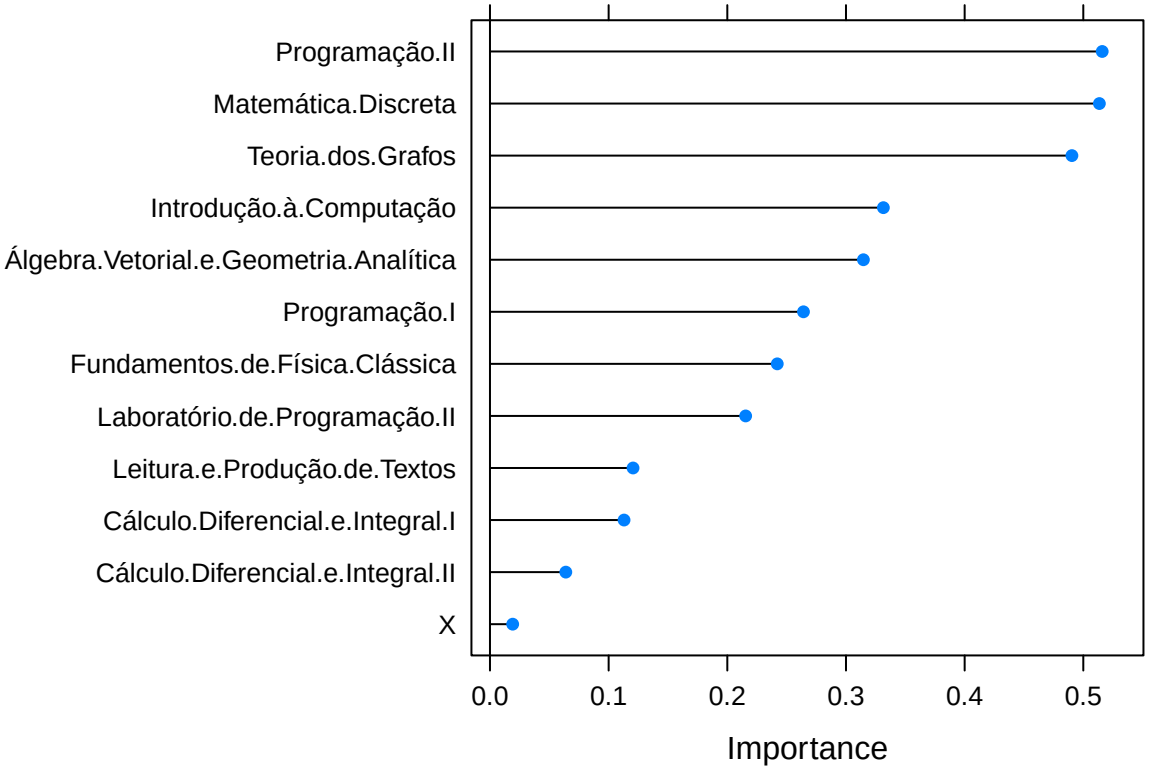
Melhorando o modelo

A fim de melhorar o modelo, foi feito um estudo acerca da importância que cada variável estava exercendo. Obtiveram-se então os seguintes resultados, nos algoritmos Ridge e Lasso, respectivamente:

```r
plot(varImp(ridge, scale = FALSE))
```




```r
plot(varImp(lasso, scale = FALSE))
```



Verificamos então, que algumas variáveis podem ser descartadas, pois podem estar atrapalhando o modelo.

```
graduados.improved <- graduados.selected %>%
select(Matemática.Discreta,
       Programação.II,
       Teoria.dos.Grafos,
       Fundamentos.de.Física.Clássica,
       cra)
```

Repetimos então o treino para conferir se houveram melhoras

```
ridgeImproved <- train(cra~., data = graduados.improved,
               method='ridge',
               trControl = fitControl,
               tuneGrid = lambdaGrid,
               preProcess=c('center', 'scale')
)

ridge.improvedPred <- predict(ridgeImproved, validacao.cra)
sqrt(mean(ridge.improvedPred - validacao.cra$cra)^2)
```

```
## [1] 0.01792701
```

E agora com o Lasso:

```
lasso <- train(cra ~., graduados.improved,method='lasso',preProc=c('scale','center'),trControl=fitContro
predict.enet(lasso$finalModel, type='coefficients', s=lasso$bestTune$fraction, mode='fraction')
lasso.improvedPred <- predict(lasso, validacao.cra)
```

```
## [1] 0.00763592
```

Últimas melhorias

É possível ir além. Então, olhando outra vez para o gráfico é visível que Programação II é a variável mais importante, juntamente com Matemática Discreta. E ainda, Física Clássica é consideravelmente menos importante comparada com as outras três variáveis selecionadas. Sendo assim, serão adicionadas três colunas ao modelo, na tentativa de melhorá-lo ainda mais: uma que pega o produto das notas de Matemática Discreta e de Programação II; uma que eleva ao quadrado as notas de Programação II; uma que tira a raiz quadrada das notas de Física Clássica. Fazendo o processamento, tem-se:

```
graduados.improved.more <- transform(graduados.improved, discreta.prog2 = Matemática.Discreta*Programaç
graduados.improved.more <- transform(graduados.improved, prog2 = Programação.II^2)
graduados.improved.more <- transform (graduados.improved, fisicasqrt = Fundamentos.de.Física.Clássica^(
```

Confirimos então, mais uma vez, os resultados obtidos:

```
ridge.improved.more <- train(cra~., data = graduados.improved.more,
               method='ridge',
               trControl = fitControl,
               tuneGrid = lambdaGrid,
               preProcess=c('center', 'scale')
)

ridge.improved.more.pred <- predict(ridge.improved.more, validacao.cra.transformed)
sqrt(mean(ridge.improved.more.pred - validacao.cra.transformed$cra)^2)
```

```
## [1] 0.01551424
```

Novamente, com o Lasso:

```
lasso.improved.more <- train(cra ~., graduados.improved.more,method='lasso',preProc=c('scale','center')
predict.enet(lasso.improved.more$finalModel, type='coefficients', s=lasso.improved.more$bestTune$fracti
lasso.improved.more.pred <- predict(lasso.improved.more, validacao.cra.transformed)
sqrt(mean(lasso.improved.more.pred - validacao.cra.transformed$cra)^2)
```

```
## 0.006297658
```

Conclusão

Conclui-se então que o melhor modelo encontrado foi selecionando as variáveis mais importantes do primeiro e segundo períodos e criando novas colunas que destaram ainda mais a importância de algumas variáveis, utilizando do algoritmo Lasso.