

# Trabalho finanças Régis

*Fabio Michel*

*22 de agosto de 2016*

O presente trabalho visa implementar um modelo para a previsão de preços de imóveis para a cidade de Pelotas- RS.

A partir de agora, iremos iniciar a implementação e previsão do modelo.

Assim, iremos carregar e instalar alguns pacotes, utilizar o script anterior da análise preliminar como fonte e implementar o modelo de previsão.

## Instalar e carregar pacotes necessarios

```
if (!require("pacman")) install.packages("pacman") pacman::p_load(data.table, Metrics, caret)
```

## Separar conjuntos train e test

```
set.seed(4321) trainIndex <- createDataPartition(aptos$preco, p=.7, list=F) trainData <- aptos[trainIndex, ]  
testData <- aptos[-trainIndex, ]
```

## Criar variaveis do conjunto train e escalonar dados

```
dummiesTrain <- predict(dummyVars(~ bairro, data = trainData), newdata = trainData)
```

Aqui, criamos a variável cobertura que iremos utilizar na equação para o modelo de previsão.

```
... dataTrain <- data.frame(preco = trainData$preco, area = trainData$area, area2 = trainData$area^2, area3 =  
trainData$area^3, quartos = trainData$quartos, quartos2 = trainData$quartos^2, quartos3 = trainData$quartos^3, suites =  
trainData$suites, suites2 = trainData$suites^2, suites3 = trainData$suites^3, vagas = trainData$vagas, vagas2 =  
trainData$vagas^2, vagas3 = trainData$vagas^3, cobertura = trainData$cobertura, dummiesTrain[,2:ncol(dummiesTrain)])  
# Incluir novas variaveis aqui ... maxs <- apply(dataTrain, 2, max) mins <- apply(dataTrain, 2,  
min) scaledTrain <- as.data.frame(scale(dataTrain, center = mins, scale = maxs - mins)) adj1 <-  
(max(dataTrain$preco) - min(dataTrain$preco)) adj2 <- min(dataTrain$preco)
```

## Funcao para computar MAE utilizando train

```
maeSummary <- function (data, lev = NULL, model = NULL) { out <- mae(dataobs, datapred)
names(out) <- "MAE" out }
```

```
formula1 <- preco ~ area + area2 + area3 + bairro.PORTO + bairro.CENTRO + bairro.FRAGATA +
bairro.ZONA.NORTE + bairro.TRES.VENDAS + quartos + quartos2 + quartos3 + vagas + vagas2 +
vagas3 + suites + suites2 + suites3
```

```
formula2 <- preco ~ area + area2 + area3 + bairro.PORTO + bairro.CENTRO + bairro.FRAGATA +
bairro.ZONA.NORTE + bairro.TRES.VENDAS + quartos + quartos2 + quartos3 + vagas + vagas2 +
vagas3 + suites + suites2 + suites3 + cobertura
```

**Para melhorar e eficiência computacional o método da função cross validation foi alterado para “cv” e “boot632”, é possível que isso gere overfitting no modelo**

## Metodo de cross validacao da funcao train

```
fitControl <- trainControl(method = “cv”, number = 10, repeats = 10, summaryFunction = maeSummary,
savePredictions = “final”)
```

```
fitControl2 <- trainControl(method = “boot632”, number = 5, repeats = 10, summaryFunction = maeSummary,
savePredictions = “final”)
```

## Melhor modelo ate o momento

```
svmGrid <- expand.grid(sigma= 2^c(-10, -5, 0), C= 2^c(0:5)) set.seed(21) svm.radial2 <- train(formula1,
data = scaledTrain, method = “svmRadial”, trControl=fitControl, metric = “MAE”, maximize = FALSE,
tuneGrid = svmGrid) mse.svmradial2 <- mse(svm.radial2predobsadj1+adj2, svm.radial2predpredadj1+adj2)
mae.svmradial2 <- mae(svm.radial2predobsadj1+adj2, svm.radial2predpredadj1+adj2) mse.svmradial2
mae.svmradial2
```

## Modelo implementado

```
svmGrid2 <- expand.grid(sigma= 2^c(-10, -5, 0), C= 2^c(0:5)) set.seed(21) modelo <- train(formula2, data =
scaledTrain, method = “svmRadial”, trControl=fitControl2, metric = “MAE”, maximize = FALSE, tuneGrid
= svmGrid2)
```

```
mse.modelo <- mse(modelopredobsadj1+adj2, modelopredpredadj1+adj2) mae.modelo <- mae(modelopredobsadj1+adj2,
modelopredpredadj1+adj2) mse.modelo mae.modelo
```

## Criar variaveis do conjunto test e escalonar dados

```
dummiesTest <- predict(dummyVars(~ bairro, data = testData), newdata = testData)
```

```
dataTest <- data.frame(preco = testDatapreco, area = testDataarea, area2 = testDataarea^2, area3 =
testDataarea^3, quartos = testDataquartos, quartos2 = testDataquartos^2, quartos3 = testDataquartos^3, suites =
```

```
testDatasuites, suites2 = testDatasuites2, suites3 = testDatasuites3, vagas = testDatavagas, vagas2 =
testDatavagas2, vagas3 = testDatavagas3, cobertura = testDatacobertura, dummiesTest[,2:6]) #Incluir as
mesmas variaveis do conjunto train
```

```
scaledTest <- as.data.frame(scale(dataTest, center = mins, scale = maxs - mins))
```

## Avaliar resultado do melhor modelo no conjunto test

```
pred_test <- predict(modelo, newdata = scaledTest) pred_test <- pred_test*adj1+adj2 mse(pred_test,
testDatapreco)mae(predtest, testDatapreco)
```

```
## [1] 6252.174
```

```
## [1] 48.7491
```

```
View(cbind(mae.svmradial2, mae.modelo))
```