Aprendizado de Máquinas
Pré-processamento

Douglas Rodrigues

Universidade Federal Fluminense

Pré-Processamento

Ideia: realizar alterações em nossas variáveis, para melhorar/otimizar a nossa predição.

1. Remover variáveis que não auxiliam na predição.
   - Variância zero ou quase zero.
   - Alta correlação.
   - Dependência Linear.

Pré-Processamento

Ideia: realizar alterações em nossas variáveis, para melhorar/otimizar a nossa predição.

1. Remover variáveis que não auxiliam na predição.
   - Variância zero ou quase zero.
   - Alta correlação.
   - Dependência Linear.
2. Tornar variáveis quantitativas mais "amigáveis".
   - Padronização dos Dados.
   - Normalização dos Dados.

Pré-Processamento

Ideia: realizar alterações em nossas variáveis, para melhorar/otimizar a nossa predição.

1. Remover variáveis que não auxiliam na predição.
   - Variância zero ou quase zero.
   - Alta correlação.
   - Dependência Linear.
2. Tornar variáveis quantitativas mais "amigáveis".
   - Padronização dos Dados.
   - Normalização dos Dados.
3. Remover/Tratar dados faltantes (NA's).

Pré-Processamento

Ideia: realizar alterações em nossas variáveis, para melhorar/otimizar a nossa predição.

1. Remover variáveis que não auxiliam na predição.
   - Variância zero ou quase zero.
   - Alta correlação.
   - Dependência Linear.
2. Tornar variáveis quantitativas mais "amigáveis".
   - Padronização dos Dados.
   - Normalização dos Dados.
3. Remover/Tratar dados faltantes (NA's).
4. Transformar/Combinar variáveis.
   - Criar Variáveis Dummy.
   - Análise de Componentes Principais (PCA).

Variância Zero ou Quase-Zero

- A ideia é remover variáveis com um único valor (variância zero) ou com uma frequência muito alta de um único valor (variância quase zero).
- *near zero covariates* = variáveis que não auxiliam na predição, pois possuem o mesmo valor em muitos individuo.
- Muito cuidado ao utilizar em variáveis quantitativas.

nearZeroVar()

- Para detectar as *near zero covariates*, utilizamos o comando `nearZeroVar()`, do pacote `caret`.
- Ele utiliza dois critérios para identificar *near zero covariates*:
  1. Se o número de valores distintos em relação ao número de amostras é baixo.
  2. Se a proporção da frequência do valor mais comum para a frequência do segundo valor mais comum é grande.

nearZeroVar()

```
> nearZeroVar(x, freqCut = 95/5, uniqueCut = 10, saveMetrics = FALSE,
names = FALSE)
```

- x = data frame com os dados
- freqCut = ponto de corte para a freqRatio = $\frac{\text{freq } 1^{o} \text{ mais comum}}{\text{freq } 2^{o} \text{ mais comum}}$.
- uniqueCut = ponto de corte para a porcentagem de valores distintos em relação ao número total de amostras.
- saveMetrics = $\begin{cases} \text{T : mostra todos detalhes;} \\ \text{F : mostra apenas variaveis nearZeroVar.} \end{cases}$
- names = $\begin{cases} \text{T : retorna os nomes das variáveis ;} \\ \text{F : mostra a numeração da coluna.} \end{cases}$

- Se **freqRatio** $>$ **freqCut** e **percentUnique** $<$ **UniqueCut**, então a variável é classificada como `nearZeroVar`.

$$\text{freqRatio} = \frac{\text{freq 1º mais comum}}{\text{freq 2º mais comum}}$$

$$\text{percentUnique} = \frac{\text{nº de classes ou valores distintos}}{\text{numero de amostras}} \cdot 100\%$$

nearZeroVar()

```
> library(ISLR)
> library(caret)
> data(Wage)
#Para ver as métricas
> nearZeroVar(Wage,saveMetrics = T)

#Removendo nzv
>nzv<-nearZeroVar(Wage,saveMetrics = F)
>Wage_nzv<-Wage[,-nzv]

#Retorna nome da nzv e trocando ponto de corte
>nzv<-nearZeroVar(Wage,saveMetrics = F,names=T,freqCut = 8,
                      freqCut = 95/5)
>Wage_nzv<-dplyr::select(Wage,-nzv)
```

Correlação

- Os **coeficientes de correlação** são medidas que resumem a relação entre duas variáveis.
- Quando temos dados com diversas variáveis, construímos a matriz de correlação.

	X_1	X_2	X_3
X_1	1	Corr(X_1, X_2)	Corr(X_1, X_3)
X_2	Corr(X_1, X_2)	1	Corr(X_2, X_3)
X_3	Corr(X_1, X_3)	Corr(X_2, X_3)	1

- Remover variáveis com alta correlação ajuda a reduzir a complexidade do modelo.

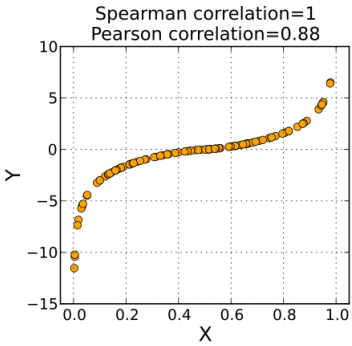
Correlação

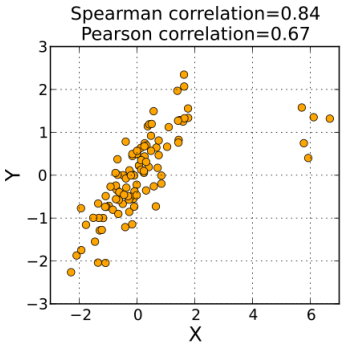
- Os **coeficientes de correlação** são medidas que resumem a relação entre duas variáveis.
- Quando temos dados com diversas variáveis, construímos a matriz de correlação.

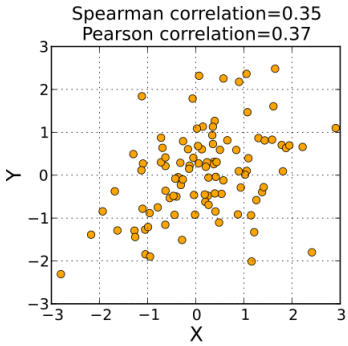
	X_1	X_2	X_3
X_1	1	$\text{Corr}(X_1, X_2)$	$\text{Corr}(X_1, X_3)$
X_2	$\text{Corr}(X_1, X_2)$	1	$\text{Corr}(X_2, X_3)$
X_3	$\text{Corr}(X_1, X_3)$	$\text{Corr}(X_2, X_3)$	1

- Remover variáveis com alta correlação ajuda a reduzir a complexidade do modelo.

Correlação: Pearson x Spearman







Fonte: https://pt.wikipedia.org/wiki/Coeficiente_de_correla%C3%A7%C3%A3o_de_postos_de_Spearman

Correlação

```
> library(kernlab)
> library(caret)
> data(spam)

#Calculamos matriz de correlação das var.  quantitativas
> descrCor <- cor(spam[1:57])
> summary(descrCor[upper.tri(descrCor)])

#Quais variáveis tem alta correlação?
findCorrelation(descrCor, cutoff = .75, verbose=T)

#Novo banco sem var.  com alta correlação
> highCor<-findCorrelation(descrCor, cutoff = .75, names=T)
> spam2<- dplyr::select(spam,-highCor)
```

Utilizando train()

```
>data(spam)
>set.seed(100)
>inTrain <- createDataPartition(y=spam$type,p=0.80,list=F)
#Separamos linhas para amostra treino/teste
>training <- spam[inTrain,]
>testing <- spam[-inTrain,]

>ctrl <- trainControl(preProcOptions = list(cutoff = 0.75,
                freqCut = 95/5, uniqueCut = 10))

>modelFit <- train(type∼ ·, data=training, method="glm",
                trControl=ctrl, preProcess=c("nzv","corr"))

>modelFit$preProcess$method$remove
#Aplicamos normamente na amostra teste
>pred_boot<-predict(modelFit,testing)
>confusionMatrix(pred_boot,testing$type)
```