Aprendizado de Máquinas
Pré-Processamento

Douglas Rodrigues

Universidade Federal Fluminense

Tratando NA's - Método `knn`

- Perder dados por falhas/erros no preenchimento nunca é bom. Em geral, o mais recomendável é descartar esses dados.
- Em alguns casos, podemos tentar "substituir" os NA's na amostra, por dados de outros indivíduos que possuam características parecidas.
- É um procedimento que deve ser feito com muito cuidado, apenas em situações de real necessidade.
- O método *k-Nearest Neighbors (knn)* consiste em procurar os k vizinhos mais próximos do indivíduo que possui o dado faltante, e substituir o NA pela média dos valores observamos dessa variável desses k vizinhos. O padrão da função é $k = 5$.

Outros métodos

- Paa outros métodos, visite a página do nosso projeto de pesquisa em Aprendizado de Máquinas:

http://cienciadedados.uff.br

Variável *dummy* (Variável Indicadora)

- As variáveis *dummies* ou variáveis indicadoras são formas de agregar informações qualitativas em modelos de regressão estatística.
- Ela atribui *1* se o indivíduo possui determinada característica, ou *0* caso contrário.
- É preciso definir qual evento/característica será atribuído o valor um e qual será atribuído o valor zero.

Criando Variável *dummy*

- Vamos utilizar o banco de dados `Wage`, do pacote `ISLR`. Ele apresenta dados pessoais e salariais de 300 trabalhadores do sexo masculino, de uma região dos EUA.

```
> library(ISLR)
> library(caret)
> library(epiDisplay)
> library(dplyr)

> data(Wage)
```

Criando Variáve *dummy*

- Vamos criar variáveis *dummies* utilizando duas variáveis do nosso banco de dados:
  jobclass: informação sobre o tipo de trabalho.

  ```
  > tab1(Wage$jobclass)
  ```

  race: indica a raça do trabalhador.

  ```
  > tab1(Wage$race)
  ```

Criando Variáve *dummy*

- Vamos criar variáveis *dummies* utilizando duas variáveis do nosso banco de dados:
  jobclass: informação sobre o tipo de trabalho.
  ```
  > tab1(Wage$jobclass)
  ```
  race: indica a raça do trabalhador.
  ```
  > tab1(Wage$race)
  ```
- Criando modelo para variáveis *dummies* p/ *jobclass* e *race*, com o comando `dummyVars()`:
  ```
  > dummies <- dummyVars(wage∼jobclass+race, data=Wage)
  ```

Criando Variáve *dummy*

- Agora, precisamos aplicar o modelo para criar as variáveis *dummies*, 0=não possui, 1=possui.
  ```
  > jobdummies <- predict(dummies,newdata=Wage)
  ```
- Observe que o novo objeto criado é uma matriz.
  ```
  > class(jobdummies)
  ```

Criando Variáve *dummy*

- Agora, precisamos aplicar o modelo para criar as variáveis *dummies*, 0=não possui, 1=possui.
  `> jobdummies <- predict(dummies,newdata=Wage)`
- Observe que o novo objeto criado é uma matriz.
  `> class(jobdummies)`
- Utilizando o comando `cbind()`, anexamos esse novo objeto nos dados.
  `> Wage_dummy <- cbind(Wage,jobdummies)`

Criando Variáve *dummy*

- Agora, precisamos aplicar o modelo para criar as variáveis *dummies*, 0=não possui, 1=possui.
  `> jobdummies <- predict(dummies,newdata=Wage)`
- Observe que o novo objeto criado é uma matriz.
  `> class(jobdummies)`
- Utilizando o comando `cbind()`, anexamos esse novo objeto nos dados.
  `> Wage_dummy <- cbind(Wage,jobdummies)`
- Se eu desejar, posso remover as antigas variáveis categóricas.
  `> Wage_dummy <- select(Wage_dummy, -c(jobclass,race))`

dummyVars()

- Para realizar modelos de regressão, não é necessário variáveis *dummies* para cada característica, pois estaríamos inserindo variáveis com colinearidade perfeita no modelo. Basta utilizar a opção `fullRank=T` na função `dummyVars()`.

```
> dummies <- dummyVars(wage∼jobclass+race, data=Wage, fullRank=T)
> jobdummies<-predict(dummies,newdata=Wage)
```