

Duas Etapas

- ◇ Primeiro
 - ◇ Ver se o modelo tem performance razoável
 - ◇ Aplicação em R “normal”, não Web
 - ◇ Se positivo, desenvolver a aplicação em Shiny

Primeira Etapa

- ♦ Dividir dados em treino e teste

- ♦ Criar modelo com Naive Bayes

```
modelo = naiveBayes(class ~ . ,carrostreino)
```

- ♦ Testar a performance do modelo

```
predicao = predict(modelo, novocarro)
```

```
confusao = table(carroteste$class,predicao)
```

Segunda Etapa

◈ Preparar dados único para preencher caixas de combinação:

```
buying = unique(carros$buying)
maint = unique(carros$maint)
doors = unique(carros$doors)
persons = unique(carros$persons)
lug_boot = unique(carros$lug_boot)
safety = unique(carros$safety)
```


Criar Modelo e Fazer Previsões

```
modelo <- naiveBayes(class ~ . ,carros)
```

```
novocarro =  
data.frame("buying"=input$buying, "maint"=input$maint, "doors"=input  
$doors, "persons"=input$persons, "lug_boot"=input$lug_boot, "safety"=  
input$safety)
```

```
predicao = predict(modelo, novocarro)
```

```
output$Resultado = renderText({as.character(predicao)})
```

Dados

- ◇ car.data
- ◇ 1728 Instâncias
- ◇ 7 atributos:
 - ◇ Class: unacc, acc, good, vgood
 - ◇ buying: vhigh, high, med, low.
 - ◇ maint: vhigh, high, med, low.
 - ◇ doors: 2, 3, 4, 5more.
 - ◇ persons: 2, 4, more.
 - ◇ lug_boot: small, med, big.
 - ◇ safety: low, med, high.

Referências:

Donors:

1. Marko Bohanec ([marko.bohanec '@' ijs.si](mailto:marko.bohanec@ijs.si))
2. Blaz Zupan ([blaz.zupan '@' ijs.si](mailto:blaz.zupan@ijs.si))

Aplicação

Previsão de Qualidade de Veículos

Preço:

vhigh

Manutenção:

vhigh

Portas:

2

Capacidade de Passageiros:

2

Porta Malas:

small

Segurança:

low

Processar

unacc

