Processo com ML

- Um processo que utilize algoritmos de ML deve ter uma divisão entre:
  - Dados para treino:
    - São aqueles com os quais os algoritmos são calibrados;
    - Quanto mais treino, melhor a predição;
    - Geralmente, acima de 70%;
  - Dados de teste:
    - Uma quantidade geralmente menor que a de dados treino e servem pra medir a % de acerto do algoritmo;

Tópicos Especiais em Sistemas de Informação

Validação cruzada ou k-folding

Ely – elydasilvamiranda@gmail.com

Modelo simplificado

- Supondo um conjunto de dados com 100 registros :
  - Treino do registro 1 ao 75 (75%);
  - Teste do registro 76 a 100 (25%);

- Supondo ainda que um algoritmo foi executado sobre esse dados e obteve 60% de acerto.

Mudança de dados do treino e teste

- Supondo ainda que se detectou que o 3º conjunto de dados não ficaria bem no treino e sim no teste:

- O algoritmo foi novamente executado e o resultado foi 72% de acerto;

Mudança de dados do treino e teste

- Supondo agora a situação baixo:

- O algoritmo foi novamente executado e o resultado foi 50% de acerto.

Mudança de dados do treino e teste

- Eventos que alterem a ordem dos dados farão com que o resultado mude;
- Isso é uma dependência dos dados e devemos mitigar isso;
- Não necessariamente o melhor ou pior resultado vão definir o quão bom é um algoritmo;
- Devemos executar várias vezes o algoritmo com diferentes combinações de dados de um dataset.

K-folding ou validação cruzada

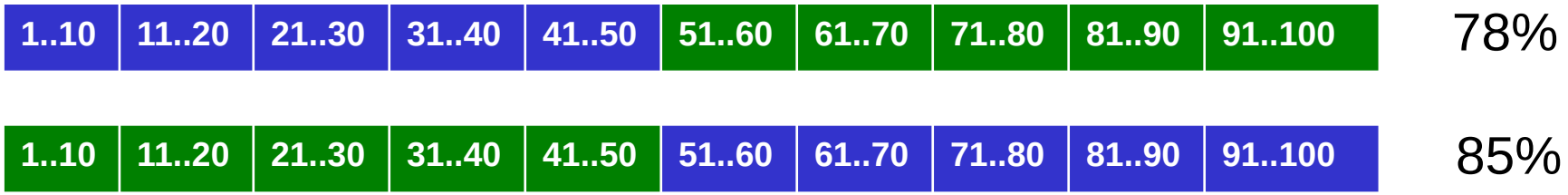
- Processo de executar o algoritmo várias vezes com diferentes conjuntos de treino e testes;
- Nesse processo de dividem-se os dados em de com uma constante **k**
- Executam-se várias combinações possíveis para um conjunto de dados;
- Ao final, deve-se calcular a média dos resultados e avaliar o desvio padrão.

K-folding ou validação cruzada

- Esse processo se chama de folding (dobrando);
- Nesse caso, chamamos de 2-fold (duas dobras) e genericamente k-fold;
- Podemos quebrar os dados em quantas dobras quisermos;
- Assim, com k-fold temos:
  - k divisões dos dados;
  - k execuções do algoritmo;

Número de foldings

- **K = 2:**
  - 50% pra treino e 50% para testes;
  - Duas execuções do algoritmo;



  - Média: 81,50%
  - Desvio padrão: 3,5%

Número de foldings

- **K = 5:**
  - 80% pra treino e 20% para testes;
  - 5 execuções do algoritmo;

1..10	11..20	21..30	31..40	41..50	51..60	61..70	71..80	81..90	91..100	78%
1..10	11..20	21..30	31..40	41..50	51..60	61..70	71..80	81..90	91..100	83%
1..10	11..20	21..30	31..40	41..50	51..60	61..70	71..80	81..90	91..100	79%
1..10	11..20	21..30	31..40	41..50	51..60	61..70	71..80	81..90	91..100	80%
1..10	11..20	21..30	31..40	41..50	51..60	61..70	71..80	81..90	91..100	81%

  - Média: 80,2%
  - Desvio padrão: 1,7205%
  - Intervalo de confiança 95%: 78,0637 a 82,33%

Implementando k-folding

- Como estudo de caso, usaremos o arquivo acesso_buscas2.csv;
- Usaremos a implementação presente na função skylearning.cross_val_score;

Implementando k-folding

- cross_val_score(modelo, dados_treino, marcacoes_treino, k), onde:
  - Modelo: modelo criado com o algoritmo de ML escolhido;
  - dados_treino: matriz de dados selecionados para treino/teste;
  - treino_marcacoes: matriz de dados selecionados como "resposta" para o treino/teste;
  - K: quantidade de divisões para folding.
- A função retorna um array de resultados, cada um equivalendo a uma execução do algoritmo.

Implementando k-folding

- Importando pacotes, lendo o arquivo e criando os dataframes:

```
import pandas as pd
import numpy as np

df = pd.read_csv('acessos_buscas2.csv')
x = df[['home', 'busca', 'logado']]
y = df['comprou']

x = pd.get_dummies(x)
```

Implementando k-folding

- Usando a implementação k-folding e exibindo os scores:

```
from sklearn.naive_bayes import MultinomialNB

from sklearn.cross_validation import cross_val_score
modelo = MultinomialNB()
k = 6
scores = cross_val_score(modelo, x, y, cv = k)
print(scores)
taxa_de_acerto = np.mean(scores)
print(taxa_de_acerto)

desvio = scores.std()

print(desvio)

from scipy.stats import bayes_mvs

print(scipy.stats.bayes_mvs(scores,0.95))
```

Intervalo de confiança

Intervalo de confiança

Resultados

- Scores:
  `[0.69230769 0.84615385 0.69230769 0.83333333 0.66666667 0.83333333]`
- Média: `0.7606837606837606`
- Intervalo de confiança:
  `0.6715733338441542 a 0.8497941875233671`

Tópicos Especiais em Sistemas de Informação

Validação cruzada ou k-folding

Ely – elydasilvamiranda@gmail.com