MÉTRICAS PARA AVALIAÇÃO DE MODELOS

Prof. Nielsen Rechia
nielsen.machado@uniritter.edu.br

Medidas de Avaliação

Principal objetivo de um modelo é classificar corretamente para novos exemplos

Existem problemas:

Em um conjunto de dados com 1000 instâncias. Temos 960 instâncias são da classe positiva e apenas 40 são da classe negativa. Qual é a acurácia de um modelo classificador que sempre prediz que as classes são positivas?

Matriz de Confusão

Pode ser utilizada com 2 ou mais classes

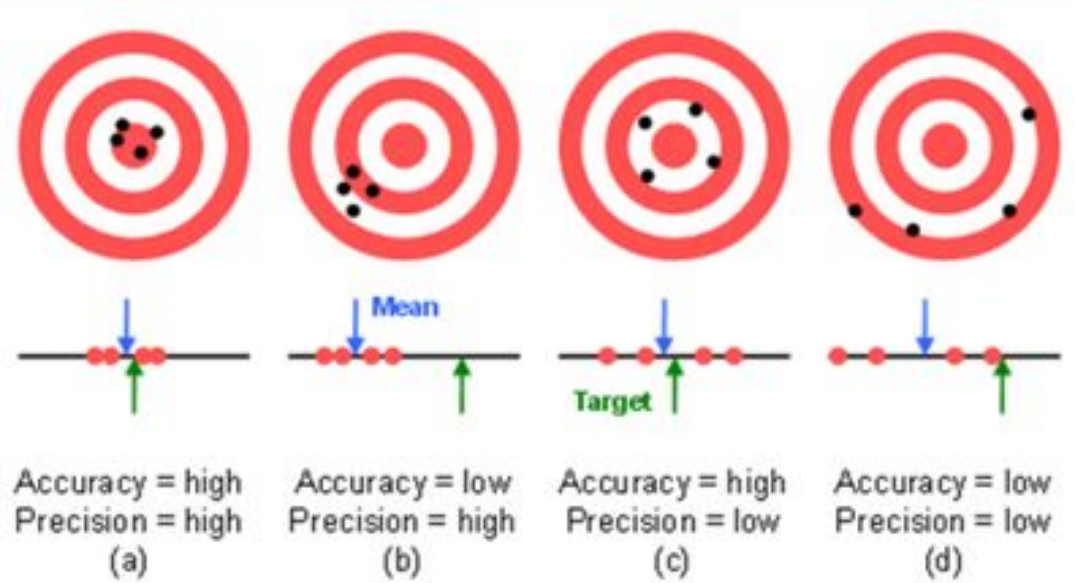
		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Acurácia = predições corretas / total de predições
Acurácia = (VP + VN) / (VP + VN + FP + FN)
Quão frequente está correto?

Matriz de Confusão



Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Taxa de Erro (Misclassification rate) = predições incorretas / total de predições

Taxa de Erro = (FP + FN) / (VP + VN + FP + FN)

Quão frequente está incorreto?

Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Taxa de Verdadeiros Positivos (recall or true positive rate) = VP / (VP + FN)
Probabilidade de classe positiva ser predita corretamente

Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Taxa de Falsos Positivos (False positive rate) = FP / (FP + VN)
Probabilidade de classe negativa ser predita incorretamente

Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Especificidade (Specificity) = VN / FP + VN
Probabilidade de predição da classe negativa ser correta

Quando prediz negativo, quão frequente está correto?
= 1 - false positive rate

Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Predição de Valores Positivos (precision) = VP / VP + FP
Probabilidade de predição da classe positiva ser correta
True positive rate

Quando prediz positivo, quão frequente está correto?

Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Prevalência (prevalence) = VP + FP / VP + FP+ FN + VN
Ocorrência da classe positiva na base

Matriz de Confusão

		Predição		
		positivo	negativo	**Total**
Real	positivo	**VP**	**FN**	**VP + FN**
	negativo	**FP**	**VN**	**FP + VN**
	Total	**VP + FP**	**FN + VN**	**VP + FP + FN + VN**

Escore F1 = 2 * (Precision * Recall) / (Precision + Recall)
Média harmônica

Matriz de Confusão

Dada a seguinte matriz de confusão, calcule para a classe gato

Acurácia, Precisão, Recall, F1 score

		Predito		
		Gato	**Cão**	**Coelho**
Real	**Gato**	5	3	0
	Cão	2	3	1
	Coelho	0	2	11

Medidas de Avaliação

Para uma lista extensiva de métricas, olhe a documentação do scikit-learn:

http://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics

Estimativa de Erro

Depende do problema:

Classificação: considera taxa de exemplos incorretamente classificados

Acurácia

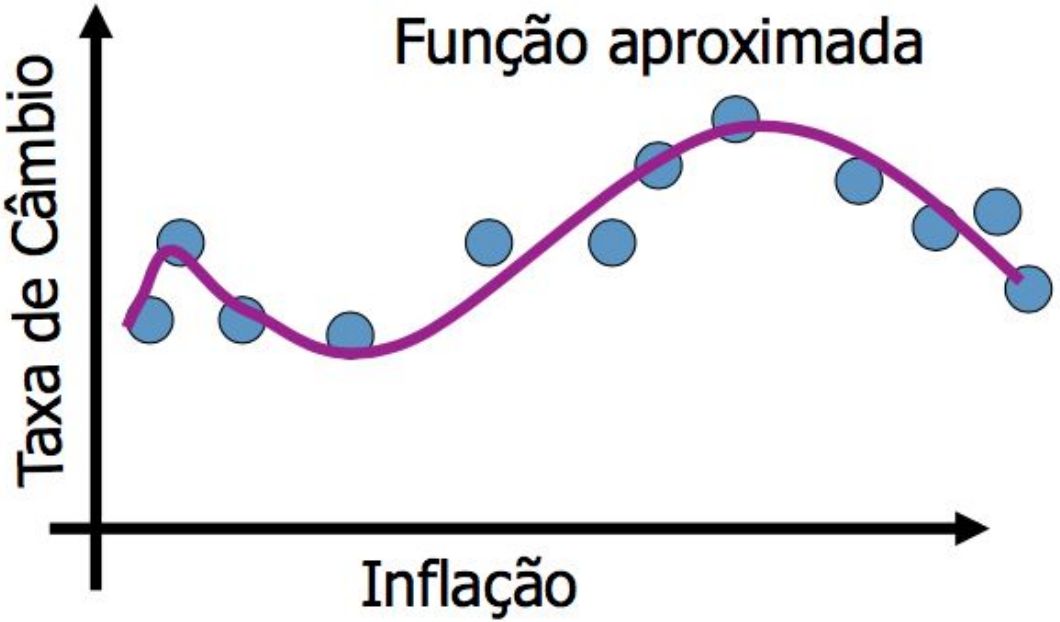
Regressão: considera diferença entre valor o produzido e valor esperado

Agrupamento: diferentes critérios

Média dos erros obtidos em diferentes execuções de um experimento

Problemas

Problema:



Erros em problemas

Os erros cometidos em uma classificação ou regressão podem ser de tipos:

Erro de treino: sobre os dados de treino

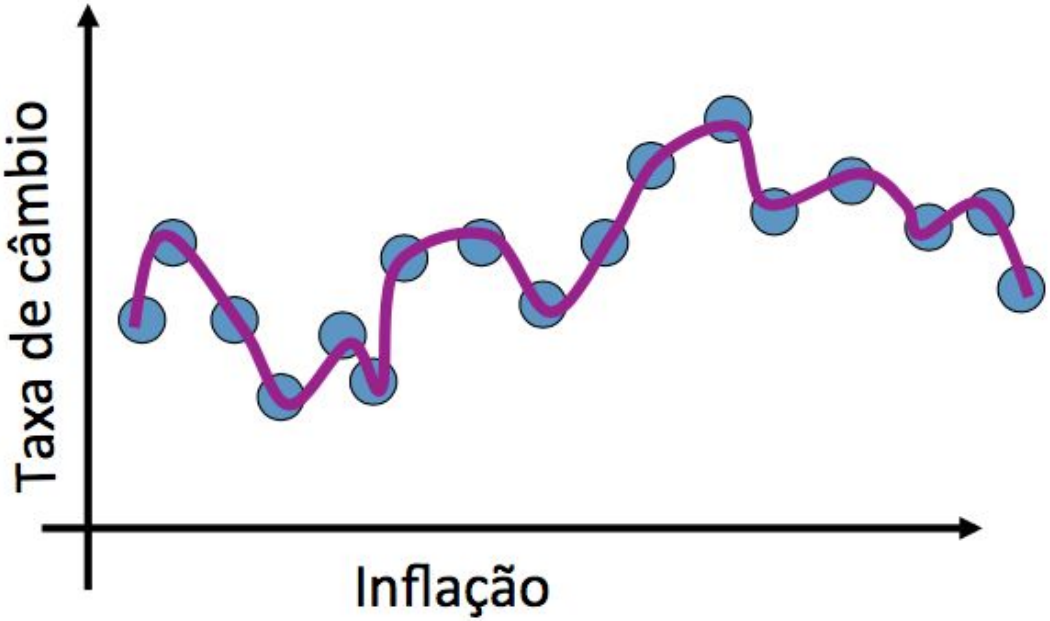
Erro de generalização: sobre os dados de teste

Erros em problemas

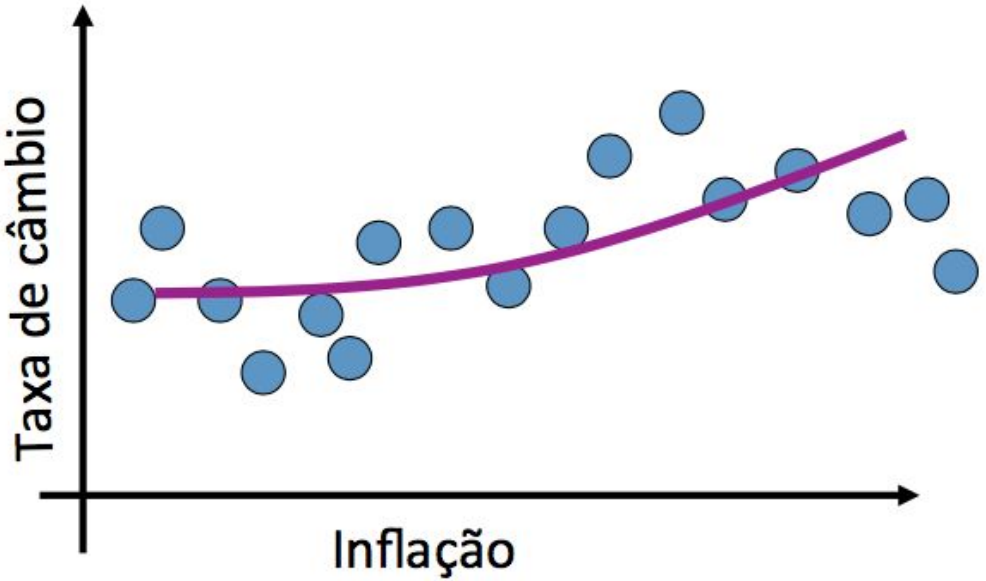
Uma classificação ou regressão deve aprender bem os dados de treino e generalizar bem para dados de teste

- Bom desempenho no treino e um desempenho ruim nos testes: **overfitting**
- Desempenho ruim nos testes e no treino: **underfitting**

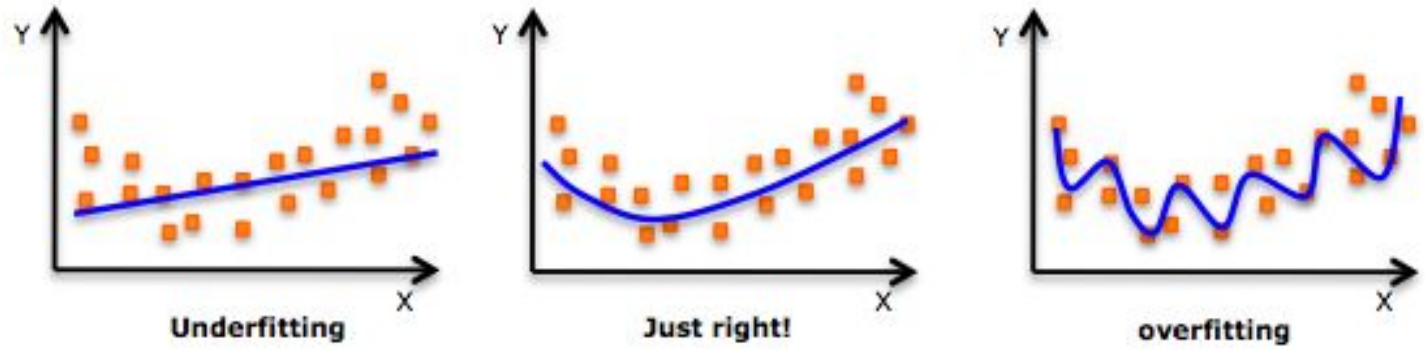
Overfitting



Underfitting



Overfitting e Underfitting



Overfitting

Overfitting ocorre:

- Por causa de dados de ruído
- Por causa da falta de instâncias que representam o conjunto de dados
- Por que o conjunto de treino é pequeno para aprender

Exercício

Variar o tamanho do conjunto de treino e teste para exemplo

O que acontece quando variamos?

Resumo

Baixo índice de erro no treino nem sempre é bom

- Um certo nível de erro é aceitável
- Evite Overfitting

A realização de testes é necessária

- Não existe uma receita de bolo para o ajuste do modelo, os dados são muito importantes
- **No Free Launch**

Métodos de Amostragem

Utilizados para avaliar desempenho de um classificador

- É comum utilizar um conjunto adicional de validação
- Algoritmos evolutivos, redes neurais
- Uma divisão comum é utilizar 60/20/20, ou 80/10/10 para treino/teste/validação

Avaliação de Desempenho

Tipos:

- Holdout
- Validação cruzada
- Amostragem randômica

Treino - Teste - Validação

Modelos iterativos:

- Em cada iteração um modelo intermediário é salvo
- É Verificado a qualidade deste modelo utilizando o conjunto de validação
- Ao final do treinamento é utilizado o modelo com a maior acurácia
- Então, é verificada a qualidade deste modelo no conjunto de testes

Holdout

Mais simples e indicado para grandes quantidades de instâncias

Divisão do dataset em diferentes conjuntos

- Treino e teste
- Ou treino, teste e validação

Conjunto de treino é o maior

- Por quê?

Holdout

In [6]:
```
# from sklearn.model_selection import train_test_split  # OU
from sklearn.cross_validation import train_test_split

X_HOLD, X_test, y_HOLD, y_test = train_test_split(X, y, test_size=0.2, stratify=y)
X_train, X_val, y_train, y_val = train_test_split(X_HOLD, y_HOLD, test_size=0.2, stratify=y_HOLD)

print 'Número de instâncias e distribuição de classes no conjunto de    treino:', X_train.shape[0], Counter(y_train)
print 'Número de instâncias e distribuição de classes no conjunto de     teste:', X_test.shape[0], Counter(y_test)
print 'Número de instâncias e distribuição de classes no conjunto de validação:', X_val.shape[0], Counter(y_val)
```

```
Número de instâncias e distribuição de classes no conjunto de    treino: 64 Counter({1: 33, 0: 31})
Número de instâncias e distribuição de classes no conjunto de     teste: 20 Counter({0: 10, 1: 10})
Número de instâncias e distribuição de classes no conjunto de validação: 16 Counter({0: 8, 1: 8})
```

Validação Cruzada

K-fold cross-validation (validação cruzada de k partes)

- Dataset é dividido em partes (folds)
- Cada instância participa o mesmo número de vezes do treinamento e uma vez do teste

Leave-one-out and test (teste com apenas uma instância)

Se as classes forem desbalanceadas podemos utilizar partes estratificadas

Amostragem Randômica

Acontece quando repetimos o holdout N vezes

Instâncias podem fazer parte do treino e do test

Validação Cruzada Estratificada

In [38]:
```
from sklearn.cross_validation import StratifiedKFold  # OU
# from sklearn.model_selection import StratifiedKFold

n_folds = 5

skf = StratifiedKFold(y, n_folds)
for i, (train_index, test_index) in enumerate(skf):
    print train_index
    print test_index

    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    print 'Número de instâncias e distribuição de classes no treino do fold %d:' % i, X_train.s
    print 'Número de instâncias e distribuição de classes no  teste do fold %d:' % i, X_test.sh
```

```
[19 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
 95 96 97 98 99]
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 20]
Número de instâncias e distribuição de classes no treino do fold 0: 80 Counter({0: 40, 1: 40})
Número de instâncias e distribuição de classes no  teste do fold 0: 20 Counter({0: 10, 1: 10})
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 20 36 38 39 40 43
 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
 95 96 97 98 99]
[19 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 37 41 42 44]
Número de instâncias e distribuição de classes no treino do fold 1: 80 Counter({0: 40, 1: 40})
Número de instâncias e distribuição de classes no  teste do fold 1: 20 Counter({0: 10, 1: 10})
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35 37 41 42 44 54 58 59 60 61 65 66 67 68 69
 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
 95 96 97 98 99]
[36 38 39 40 43 45 46 47 48 49 50 51 52 53 55 56 57 62 63 64]
Número de instâncias e distribuição de classes no treino do fold 2: 80 Counter({0: 40, 1: 40})
Número de instâncias e distribuição de classes no  teste do fold 2: 20 Counter({0: 10, 1: 10})
```

Resumo

Estratificação é sempre melhor

A estratificação deixa as classes nos conjuntos de treino e teste balanceadas

Isso faz com que melhore a generalização

Exercício

O dataset RHs.csv é uma coleção de informações para determinar se um funcionário deixará a empresa em que trabalha

Treine um modelo k-NN sobre este conjunto de dados

Aplique uma validação cruzada de 5 passos

Verifique a matriz de confusão para cada um dos folds

O dataset RH.csv é o mesmo dataset com dois atributos nominais que devem ser pré-processados antes.

Veja o que muda!

Metricas

Para saber mais sobre as possíveis métricas, verifique a documentação do scikit-learn:
http://scikit-learn.org/stable/modules/classes.html

Conclusão

Leitura recomendada:

Capítulo 4 e 5 de Introduction to Data Mining