# O que é AUC e ROC nos modelos de Machine Learning

Adriano Avelar

21 mai. 2019

Fonte: <https://medium.com/@eam.avelar/o-que-é-auc-e-roc-nos-modelos-de-machine-learning-2e2c4112033d>

Logotipo

Descrição gerada automaticamente

*Em aprendizagem de máquina, a medição de desempenho é uma tarefa crucial. As métricas ROC (Receiver Operating Characteristics) e AUC (Area Under The Curve) são as mais utilizadas para medir o desempenho de modelos de classificação.*

## O que é AUC ?

ROC (Receiver Operating Characteristics)é uma curva de probabilidade. Ela é criada traçando a taxa verdadeiro-positivo contra a taxa de falsos-positivos. Ou seja, o número de vezes que o classificador acertou a predição conta o número de vezes que o classificador errou a predição.

O AUC (Area Under the Curve)representa o grau ou medida de separabilidade. Quanto maior o AUC, melhor o modelo está em prever 0s como 0s e 1s como 1s. Por exemplo, quanto maior a AUC, melhor o modelo está em distinguir entre pacientes com doença e pacientes sem doença

O **ROC** possui dois parâmetros:

* **Taxa de verdadeiro positivo** *(True Positive Rate)*, que é dado por *true positives / (true positives + false negatives).* Essa taxa também é conhecida como sensibilidade, recordação ou probabilidade de detecção (*sensitivity*, *recall* ou probability of detection*)*
* **Taxa de falso positivo** *(False Positive Rate)*,que é dado por *false positives / (false positives + true negatives).* A taxa de falsos positivos também é conhecida como probabilidade de alarme falso ([fall-out](https://en.wikipedia.org/wiki/Information_retrieval" \l "Fall-out" \t "_blank) or *probability of false alarm)* e pode ser calculada como (1 — **Specificity**). A **Specificity (**especificidade) também é conhecida como **true negative rate.** Por exemplo, quantas pessoas sem uma doença (**true negative**) foi classificadas como sadias.

Assim, para simplificar a curva **ROC,** foi criada a **AUC.** A **AUC** resume a curva **ROC** num único valor, calculando a *“área sob a curva”*.

Figura 1. Área sob a curva (AUC).

Gráfico

Descrição gerada automaticamente com confiança baixa<https://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-9863-7_209>

Na Figura 1 é mostrado duas AUC (A e B). A área de A é maior que a área de B, com isso, o desempenho de A é maior que o desempenho de B.

Um modelo excelente tem AUC próximo ao 1, o que significa que tem boa medida de separabilidade. Um modelo pobre tem AUC próximo do 0, o que significa que tem a pior medida de separabilidade, ou seja, está prevendo 0s como 1s e 1s como 0s. E quando a AUC é 0,5, significa que o modelo não tem capacidade de separação de classe.

## Como plotar uma curva AUC?

Vamos plotar a curva AUC para o dataset **Breast Cancer**.

Este dataset pode ser encontrado em <https://archive.ics.uci.edu/ml/datasets.php>.

Para nossa sorte, a biblioteca *sklearn* possui vários datasets famosos que podemos utilizar para teste. A linha 5 do código abaixo carrega justamente esse dataset.

from sklearn import metrics # Metricas para calcular accuracy score

from sklearn.linear\_model import LogisticRegression # Modelo utilizado

from sklearn.model\_selection import train\_test\_split # Separa dados de treinamento e teste

from sklearn.datasets import load\_breast\_cancer # Carrega o dataset Breast Cancer

import matplotlib.pyplot as plt # Plotagem de gráficos

O próximo passo é separar os dados (teste e treinamento) e criar o modelo. Estou utilizando o **LogisticRegression** como modelo, mas você pode utilizar qualquer um. Na verdade, mais adiante vamos comparar com outros modelos.

# Carregando Breast Cancer Dataset

breast\_cancer = load\_breast\_cancer()

X = breast\_cancer.data

y = breast\_cancer.target

# Separando o Dataset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.33, random\_state=44)

# Criando um modelo

clf = LogisticRegression(penalty='l2', C=0.1)

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

O último passo é calcular a precisão do modelo e plotar o gráfico AUC

# Accuracy

print("Accuracy", metrics.accuracy\_score(y\_test, y\_pred))

#AUC Curve

y\_pred\_probability = clf.predict\_proba(X\_test)[::,1]

fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_probability)

auc = metrics.roc\_auc\_score(y\_test, y\_pred\_probability)

plt.plot(fpr,tpr,label="data 1, auc="+str(auc))

plt.legend(loc=4)

plt.show()

O resultado desse código será o gráfico da Figura 2.

Figura 2. Gráfico AUC para o dataset Breat Cancer.

Gráfico

Descrição gerada automaticamente

Vamos fazer uma comparação entre o LogisticRegression e um modelo de Árvore de Decisão Simples.

Antes de adicionar o código abaixo, remova a linha “plt.show()” do final do código anterior.

Ele cria um novo modelo baseado em árvore e plot a AUC junto com a AUC do modelo anterior.

from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

y\_pred\_probability = clf.predict\_proba(X\_test)[::,1]

fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_probability)

auc = metrics.roc\_auc\_score(y\_test, y\_pred\_probability)

plt.plot(fpr,tpr,label="DecisionTree, auc="+str(auc))

plt.legend(loc=4)

plt.show()

O resultado é a Figura 3:

Figura 3. Comparação entre modelo LogisticRegression e um modelo de Árvore de Decisão Simples.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Percebe-se que o modelo com DecisionTree possui um desempenho inferior ou do LogisticRegression, pois a área sob a curva é menor no primeiro modelo.

Outro modelo bastante utilizado em classificação é o RandomForest. Vamos fazer a comparação dos três. O código para adicionar o RandomForest é semelhante ao DesicionTree.

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier()

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

y\_pred\_probability = clf.predict\_proba(X\_test)[::,1]

fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_probability)

auc = metrics.roc\_auc\_score(y\_test, y\_pred\_probability)

plt.plot(fpr,tpr,label="RandomForest, auc="+str(auc))

plt.legend(loc=4)

plt.show()

A saída é a Figura 4.

Figura 4. Comparação entre LogistcRegression, DecisionTree e RandomForest

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Percebe-se que o desempenho do LogistRegression ainda é superior que os demais.

Espero que você tenha gostado deste artigo. Coloque nos comentários o que você achou. Grande Abraço.

Você pode me encontrar em: <https://www.linkedin.com/in/adriano-avelar-26b10624/>