

# Card23\_Métricas e Validação de Modelos de Aprendizado de Máquina

Davi Bezerra Barros

## Comparando modelos de machine learning:

Para comparar a eficiência de diferentes modelos de machine learning são utilizadas as **métricas de validação**, indicadores de quão bem o modelo está performando com os dados de teste e validação, e da sua capacidade de generalização. As métricas escolhidas dependem do tipo de problema abordado, e as principais estão descritas a seguir.

### Métricas de ajuste:

#### 1. Classificação

- **Matriz de confusão:**
- **Acurácia:** Mede o nível de exatidão das predições do modelo.

$$\frac{Acertos}{Total} = \frac{VN + VP}{VN + FN + VP + FP}$$

- **Precisão:** Mede a proporção de predições positivas que realmente são positivas.

$$Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos}$$

- **Recall ou Sensibilidade:** Mede a proporção de positivos reais que foram corretamente classificados como positivos. É complementar à precisão

$$Recall = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos}$$

- **Especificidade:** É a Taxa de verdadeiros negativos. mede a proporção de classificações negativas que foram corretamente classificadas. É semelhante ao recall, mas em relação à classe negativa.

$$Especificidade = \frac{Verdadeiros\ Negativos}{Verdadeiros\ Negativos + Falsos\ Positivos}$$

- **Curva ROC:** É a curva produzida ao plotar a especificidade x recall para diferentes valores de corte de probabilidade. A curva ROC representa a capacidade de classificação do modelo.

#### 2. Regressão

- **Raiz do Erro Quadrático Médio (RMSE):** É a raiz quadrada do MSE, o que traz o erro de volta à mesma escala das variáveis originais.
- **Erro Absoluto Médio (MAE):** Calcula a média dos erros absolutos entre as previsões e os valores reais.
- **Erro Quadrático Médio (MSE):** Média dos quadrados dos erros (diferenças entre as previsões e os valores reais).

No entanto, as métricas abordadas na aula se referem apenas a problemas de classificação.

## Implementação da curva ROC

O cálculo da curva **roc** em um modelo de classificação treinado com a base de dados *Breast Cancer* foi feito nas seguintes etapas:

#### 1. Importação da base de dados

```
[86] import numpy as np
import pandas as pd
import sklearn
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

[81] #baixando e separando o dataset
data = load_breast_cancer()
X = data.data
y = data.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
```

#### 2. Treinamento do modelo

```
[77] #Treinando o modelo:
clf = LogisticRegression(solver="liblinear")
clf.fit(X_train, y_train)
```

LogisticRegression(solver='liblinear')

#### 3. Cálculo da curva ROC Para os datasets de teste e treino:

```
[78] from sklearn.metrics import roc_curve, auc, roc_auc_score
y_train_score = clf.predict_proba(X_train)[:, 1]
y_test_score = clf.predict_proba(X_test)[:, 1]
```

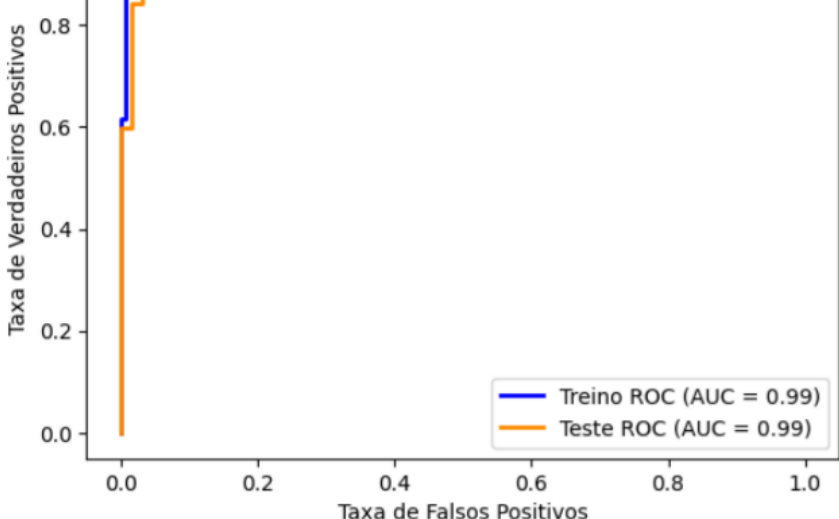
```
[79] # calculando curva roc
fpr_train, tpr_train, _ = roc_curve(y_train, y_train_score)
fpr_test, tpr_test, _ = roc_curve(y_test, y_test_score)

#área sob a curva
roc_auc_train = auc(fpr_train, tpr_train)
roc_auc_test = auc(fpr_test, tpr_test)
```

#### 4. Plotagem das curvas:

```
import matplotlib.pyplot as plt

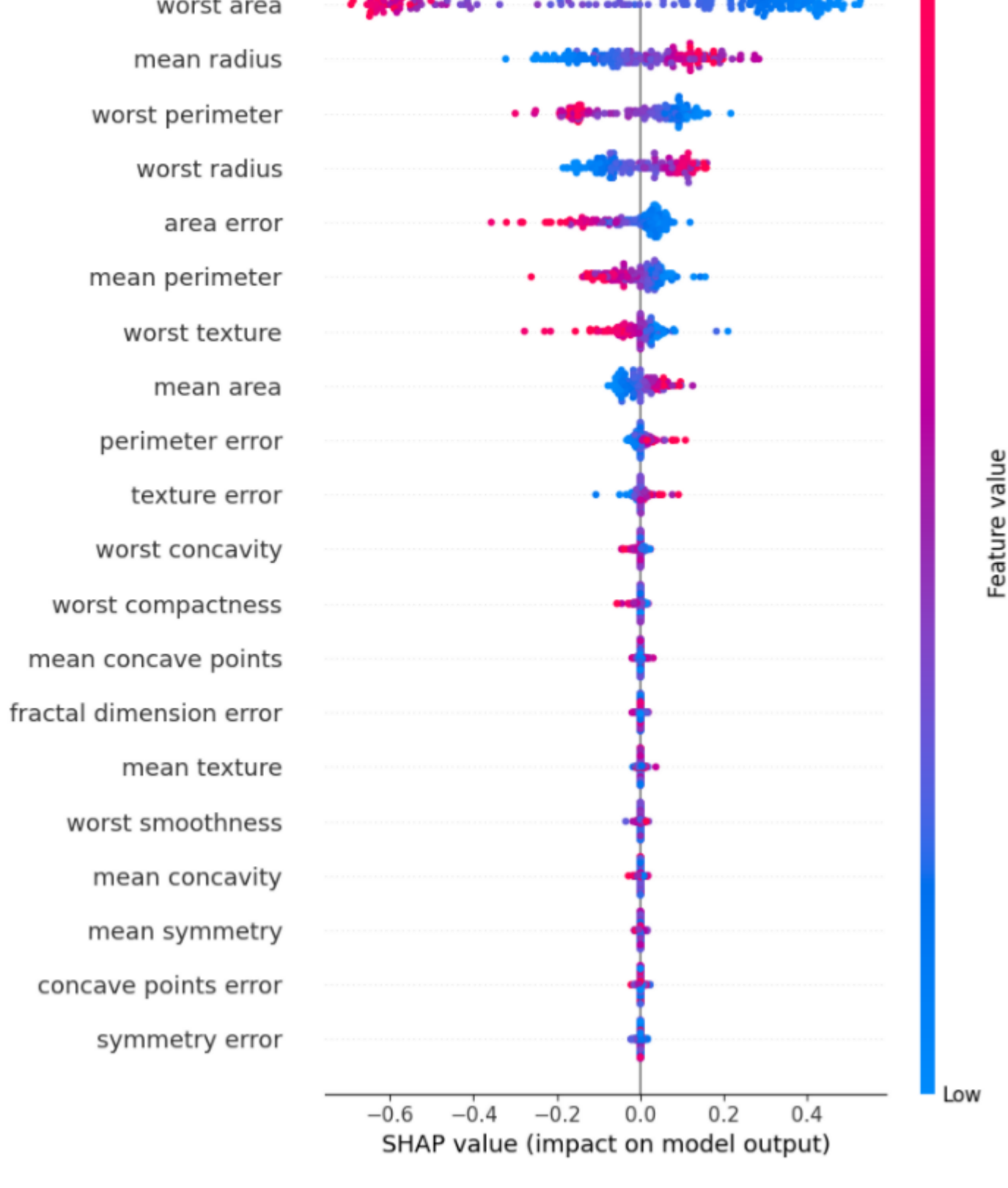
plt.figure()
plt.plot(fpr_train, tpr_train, color='blue', lw=2, label=f'Treino ROC (AUC = {roc_auc_train:.2f})')
plt.plot(fpr_test, tpr_test, color='darkorange', lw=2, label=f'Teste ROC (AUC = {roc_auc_test:.2f})')
plt.legend(loc="lower right")
plt.xlabel('Taxa de Verdadeiros Positivos')
plt.ylabel('Taxa de Falsos Positivos')
plt.title('CURVA ROC')
plt.show()
```



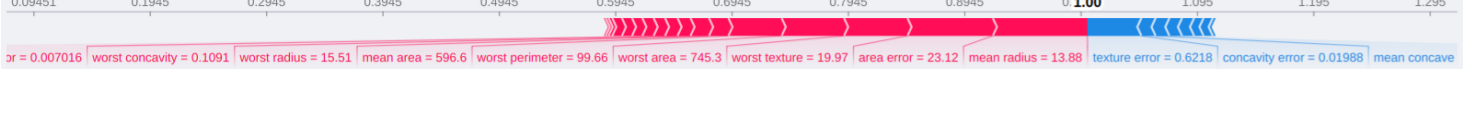
## Explicando predições com SHAP

Quando fazemos predições com um modelo de rede neural, geralmente obtemos apenas um valor de probabilidade para uma classificação, sem saber como o modelo chegou à sua conclusão. O **SHAP(Shapley Additive exPlanations)** é uma ferramenta baseada no conceito de *valores de Shapley* da teoria dos jogos, utilizada para detalhar as previsões feitas pelos modelos e mostrar como cada característica de um dado contribuiu para a classificação. O valor de Shapley representa a média de como uma característica afeta a previsão para todas as combinações possíveis de características, e quantifica a contribuição de cada uma.

Representação de várias previsões no modelo treinado anteriormente:



Representação de uma única previsão:



## Conclusão

A avaliação dos modelos de machine learning é fundamental para garantir a confiabilidade das predições. As métricas apresentadas, como a curva ROC e o método SHAP, possibilitam uma avaliação mais precisa e direcionada dos modelos treinados, facilitando a identificação de erros e a implementação de melhorias. A curva ROC permite visualizar a capacidade de separação linear de um modelo, enquanto o SHAP ilustra a importância que cada característica tem no treinamento, e a combinação destas e outras técnicas permite a construção de modelos mais confiáveis.