Métricas, ROC, AUC, features

Instruções (requisitos) 3-1

Fazer cada atividade no Jupyter (Collab). Poste apenas os links.

Siga a forma de entrega da atividade anterior.

Para cada parte da atividade, colocar como figuras os slides do comando do problema antes de cada solução.

Instruções (requisitos) 3-2

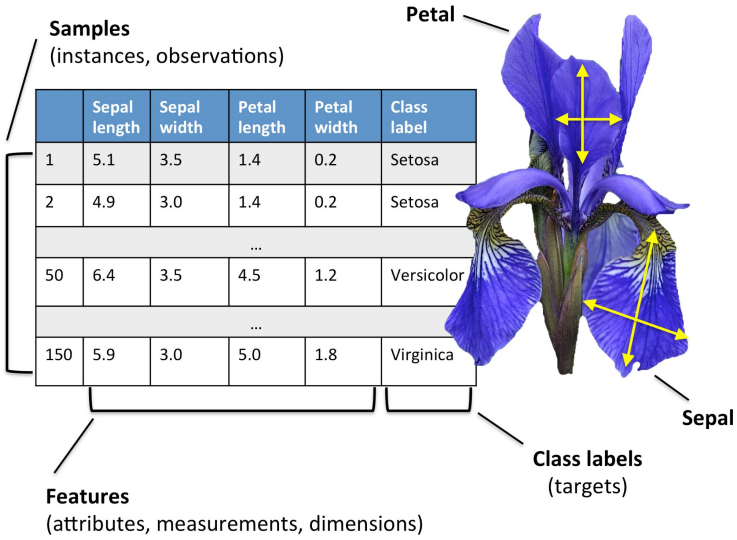
Quando eu falar “faça a análise os resultados”, a análise deve estar em um célula de texto. Sem a análise detalhada, a tarefa não será avaliada. Na verdade, eu não preciso falar nada: cada parte da atividade sempre tem de ter análises de resultados e conclusão.

Instruções (requisitos) 3-3

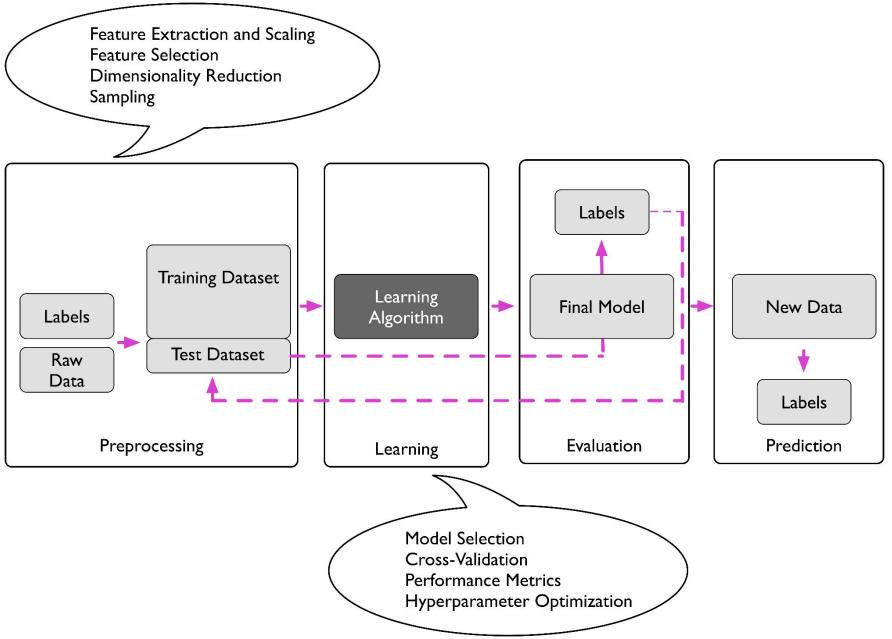
Caso algum requisito esteja faltando, sua tarefa não será analisada.

A

Poste no seu Jupyter as figuras a seguir e as explique em detalhes [ch01.ipynb]:



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

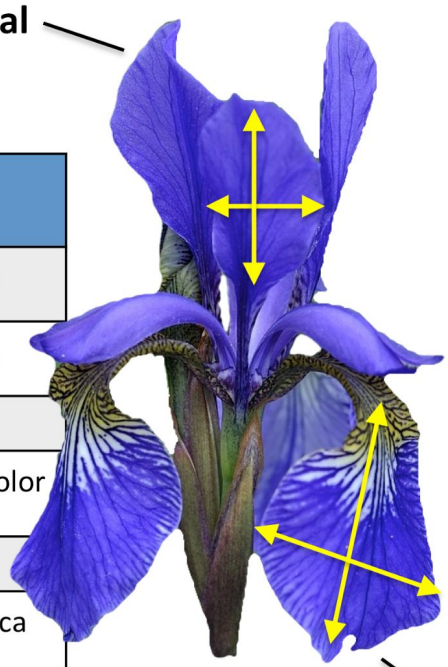


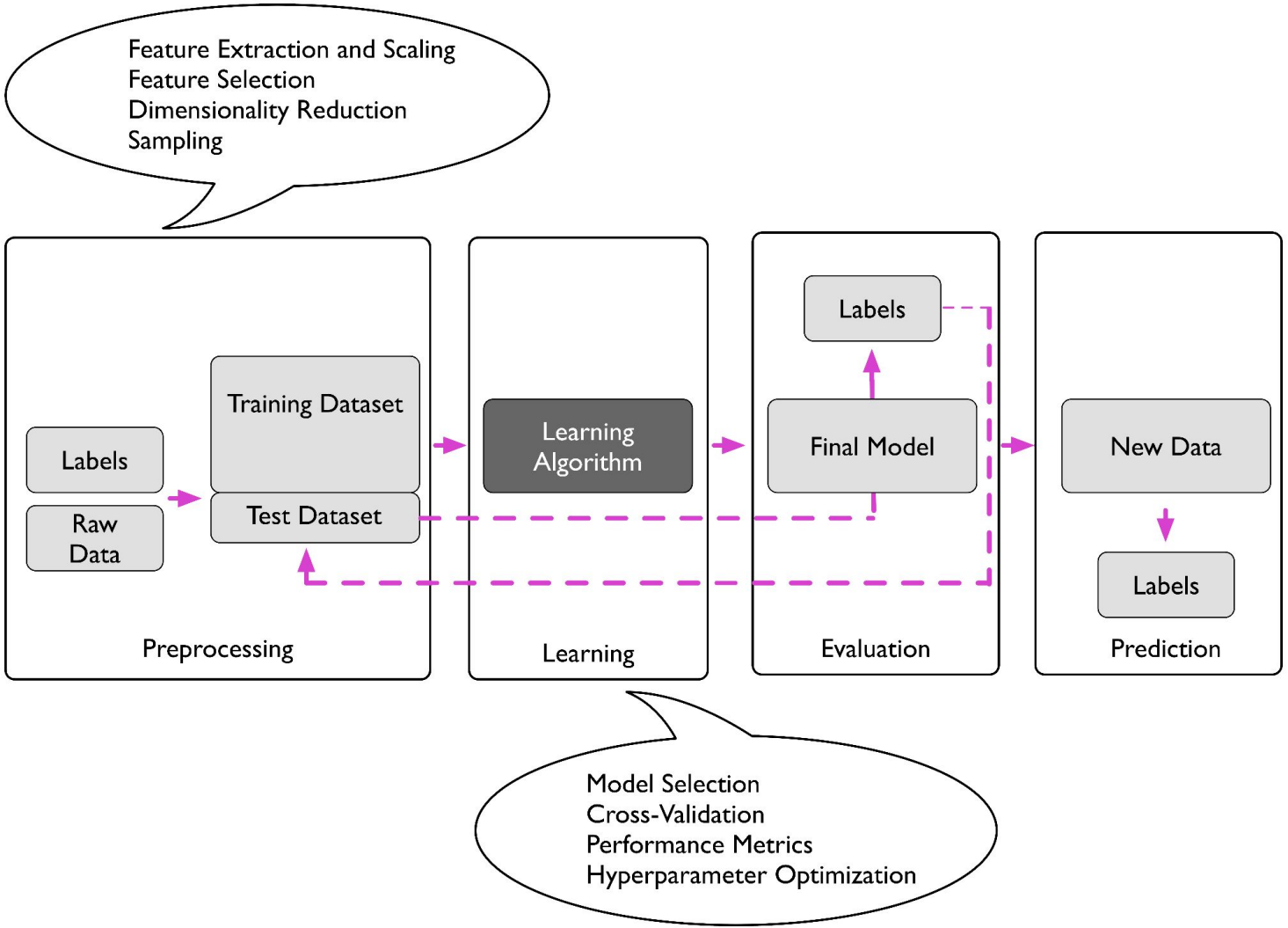
Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)




Feature Extraction and Scaling
Feature Selection
Dimensionality Reduction
Sampling
Labels
Raw Data
Training Dataset
Test Dataset
Preprocessing
Learning Algorithm
Learning
Labels
Final Model
Evaluation
New Data
Labels
Prediction
Model Selection
Cross-Validation
Performance Metrics
Hyperparameter Optimization

B

Faça uma apresentação no Jupyter sobre as principais métricas: accuracy, precision, recall e F1-score.

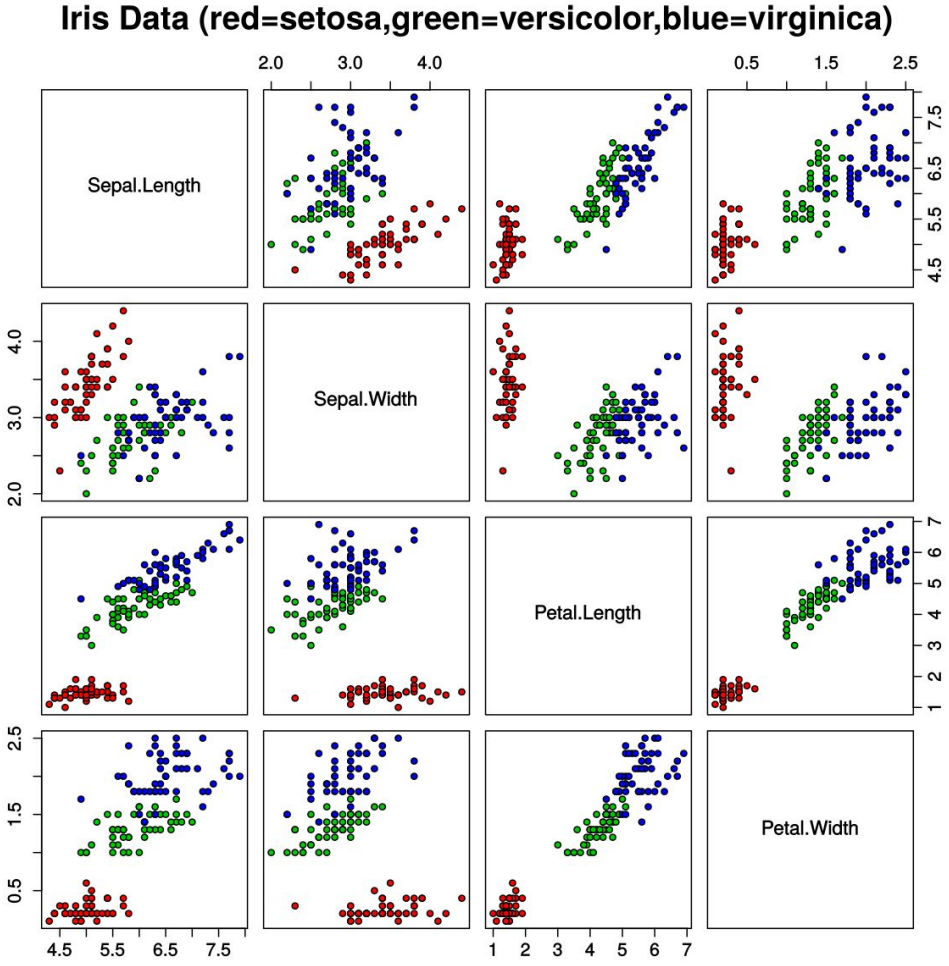
Para cada métrica crie células com figuras e textos explicando a métrica. Além disso, para cada uma delas crie, usando Pandas, duas matrizes de confusão e calcule a métrica. Uma dessas matrizes **deve ser tal que demonstre a desvantagem da métrica**. Discuta os resultados. Lembre da parte em negrito.

C

Comparando Naive Bayes e KNN com Estratégias Diferentes de Normalização usando o Dataset Iris

Você irá trabalhar com o conjunto de dados Iris, que contém 150 amostras de 3 classes de flores (setosa, versicolor, virginica), cada uma descrita por 4 características numéricas, conforme figura do próximo slide.

Sua tarefa é aplicar dois classificadores — *Naive Bayes* e *K-Nearest Neighbors* (KNN) — sob três diferentes estratégias de pré-processamento, e depois comparar os resultados.


Iris Data (red=setosa,green=versicolor,blue=virginica)
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
2.0
3.0
4.0
0.5
1.5
2.5
4.5
5.5
6.5
7.5
1
2
3
4
5
6
7

Condições de Pré-processamento

A - Normalização Min-Max: Escalar todas as variáveis para o intervalo [0, 1].

B - Padronização (Z-score): Transformar as variáveis para que tenham média 0 e desvio padrão 1.

C - Ambos sequencialmente: Primeiro aplicar padronização (z-score), depois normalização Min-Max.

Você pode usar `MinMaxScaler` e `StandardScaler` do módulo `sklearn.preprocessing`.

1 - Utilize validação cruzada (5-fold cross-validation) para avaliar os modelos de forma mais robusta.

Para cada iteração da validação cruzada:

Ajuste o escalonador (normalizador ou padronizador) somente nos dados de treino do fold atual.

Aplique a transformação nos dados de treino e teste da fold.

Treine e avalie os classificadores com os dados transformados.

2 - Para cada uma das três estratégias de pré-processamento:

Aplique a validação cruzada nos dois classificadores:

Naive Bayes (GaussianNB)

KNN com k = 5

Armazene e calcule as métricas de desempenho.

2.1 Force para que ocorra vazamentos de dados e compare com o resultado acima.

3 - Avalie os modelos utilizando acurácia média, matriz de confusão média e métricas como precisão, recall e F1-score.

4 - Compare os resultados obtidos nas diferentes estratégias de pré-processamento.

Análise

Como o pré-processamento influenciou o desempenho dos dois algoritmos.

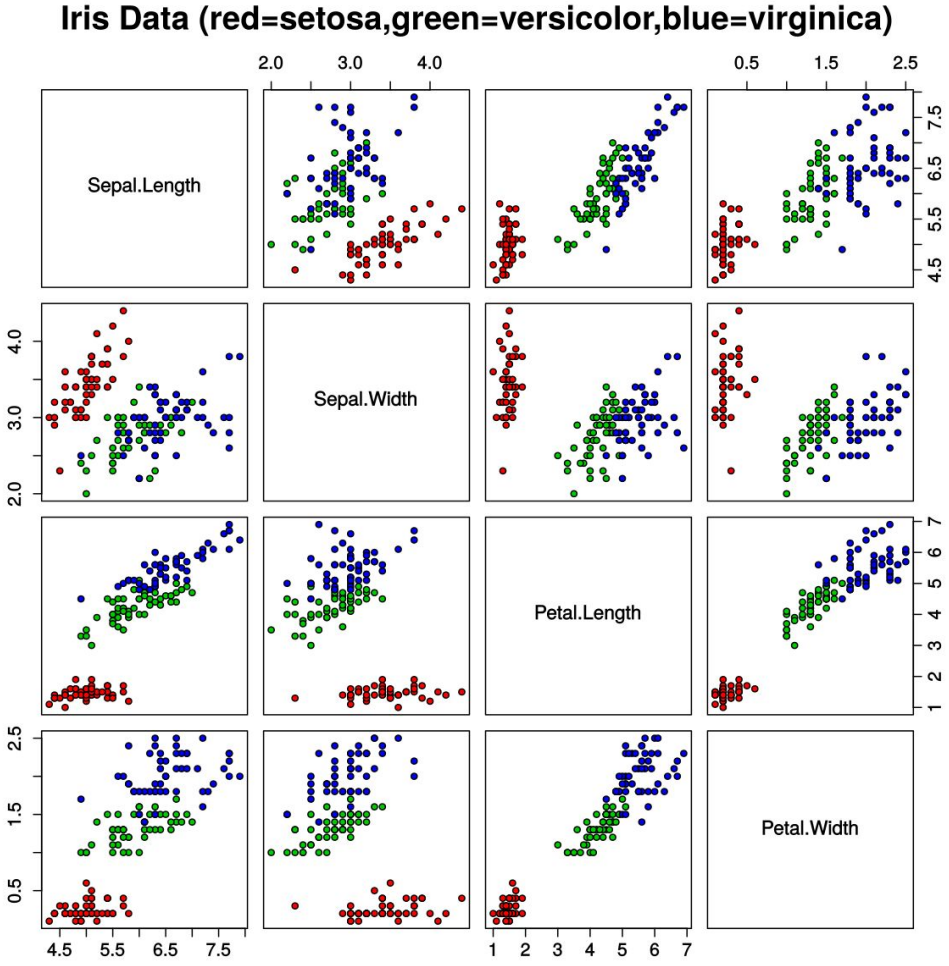
Qual classificador teve melhor desempenho em cada cenário.

Padrões ou conclusões observadas com base nas métricas.

Cuidado com vazamento de dados.

D

Classificador KNN no Conjunto de Dados Íris usando Curvas ROC


Iris Data (red=setosa,green=versicolor,blue=virginica)
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
2.0
3.0
4.0
0.5
1.5
2.5
4.5
5.5
6.5
7.5
1
2
3
4
5
6
7

Usando o conjunto de dados **Iris**, que contém 150 amostras de 3 espécies de flores do gênero *Iris*: *setosa*, *versicolor* e *virginica*. Cada amostra possui 4 *features*. **Use apenas duas features.**

Você irá utilizar o classificador **K-Nearest Neighbors (KNN)** para realizar uma tarefa de **classificação binária**, onde a variável alvo será se a amostra pertence ou não à classe *versicolor*.

1 - Converta o problema em uma classificação binária, onde a **classe positiva** é *versicolor* e a **classe negativa** é "não versicolor".

2 - Divida o conjunto de dados em conjunto de treino (70%) e conjunto de teste (30%) utilizando uma semente aleatória para reprodutibilidade.

3 - Para $k \in \{1,3,5,7,9\}$, faça o seguinte:

- Treine um classificador KNN com o valor de k correspondente.

- **Faça a predição das probabilidades** para o conjunto de teste (ou seja, a probabilidade de que uma amostra pertença à classe positiva).

- **Calcule a curva ROC** e a **AUC** para cada valor de k.

- **Plote todas as curvas ROC** em um único gráfico, uma para cada valor de k, com rótulos e legenda apropriados.
- **Comente** como o valor de k influencia o desempenho do classificador, com base nas curvas ROC e nos valores de AUC.
- **Compare** com o resultado do trabalho anterior que também usou o dataset Iris e avalie quanto ao ***bias*** e ***variance***

Resultado esperado:
Um gráfico com todas as curvas ROC (uma para cada k).

O valor da **AUC** para cada valor de k.

Uma **breve interpretação** dos resultados.

Dicas:

Use KNeighborsClassifier(..., probability=True) do módulo sklearn.neighbors.

Use as funções roc_curve e auc do módulo sklearn.metrics.

model.predict_proba(X_test)

E

Usando os seguintes dados:

Weight for obese rats (grams): 450, 455, 570, 660
Weight for obese rats (grams): 575, 665, 700, 710

1 - Obtenha a obese rat classification using logistic regression.
2 - Plote o resultado com a curva da **logistic regression** e as amostras com cores diferentes para cada classe

Usando os seguintes dados:

Weight for obese rats (grams): 450, 455, 570, 660
Weight for obese rats (grams): 575, 665, 700, 710

3 - Obtenha iterativamente a curva ROC, mostrando cada tabela verdade e cada novo ponto da curva ROC. Conforme procedimento deste slide Machine Learning Fundamentals: Sensitivity and Specificity.pptx