

FIA/P GRADUAÇÃO

ENGENHARIA DE COMPUTAÇÃO

Inteligência Artificial e Computacional

PROF. ANTONIO SELVATICI

SHORT BIO



É engenheiro eletrônico formado pelo Instituto Tecnológico de Aeronáutica (ITA), com mestrado e doutorado pela Escola Politécnica (USP), e passagem pela Georgia Institute of Technology em Atlanta (EUA). Desde 2002, atua na indústria em projetos nas áreas de robótica, visão computacional e internet das coisas, aliando teoria e prática no desenvolvimento de soluções baseadas em Machine Learning, processamento paralelo e modelos probabilísticos. Desenvolveu projetos para Avibrás, IPT, CESP e Systax.

PROF. ANTONIO SELVATICI

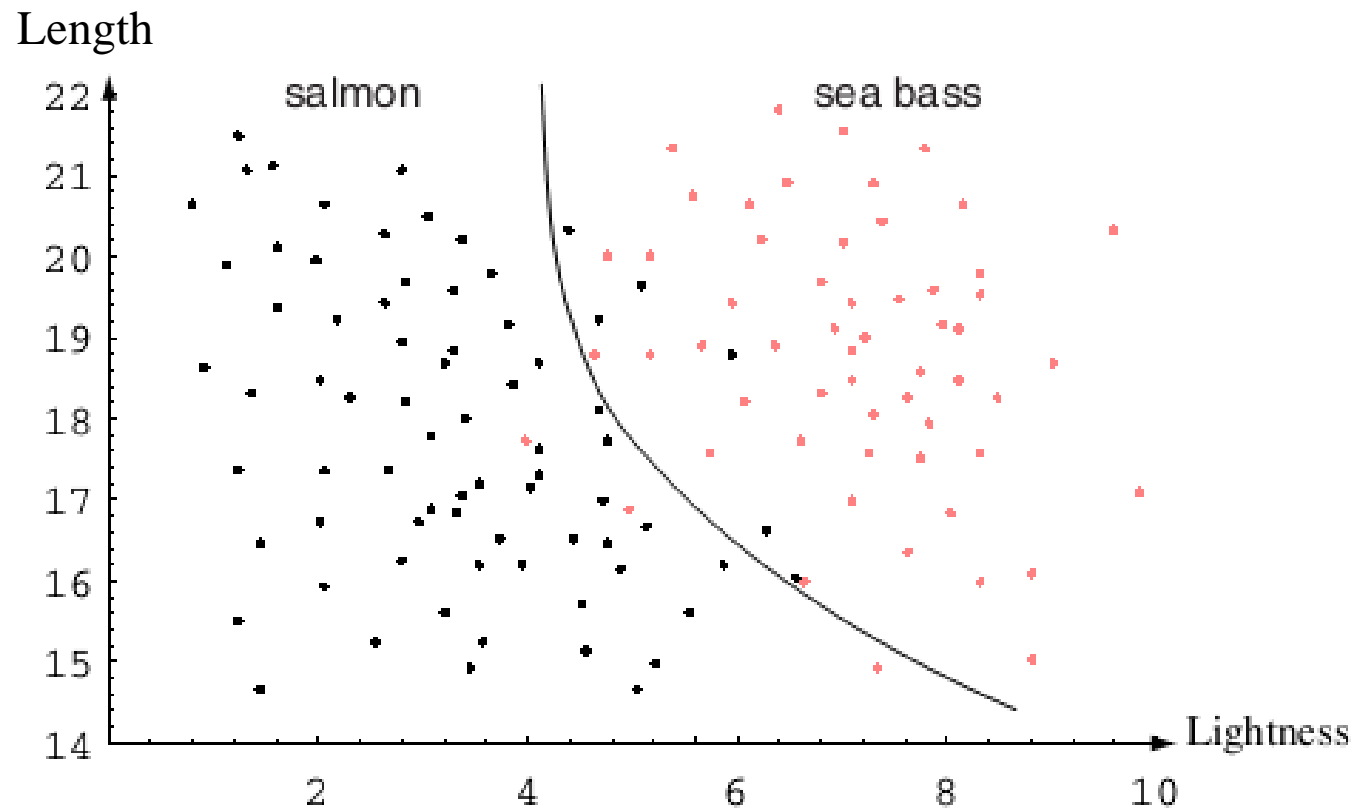
profantonio.selvatici@fiap.com.br

2. MACHINE LEARNING

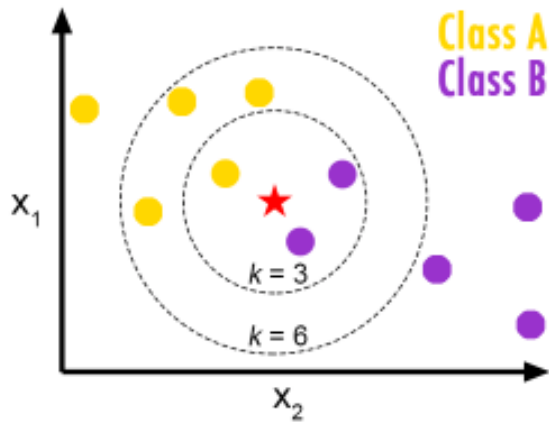
A execução de classificadores

- No aprendizado supervisionado, exemplares rotulados do vetor de atributos são fornecidos ao classificador para que este aprenda a identificar a classe corretamente. A esses exemplares denominamos **amostras de treinamento**.
- Por aprendizado entenda-se a construção de uma estrutura interna capaz de consolidar o conhecimento contido nos dados, que acaba por definir, explícita ou implicitamente, uma linha ou superfície de separação que melhor diferencia os vetores de atributos pertencentes a cada classe.
- Assim, o uso do classificador compreende duas fases:
 - **Fase de treinamento:** exemplos de vetores de atributos provenientes de diferentes classes (**amostras de treinamento**) são utilizados para definir, implícita ou explicitamente, a superfície de separação.
 - **Fase de teste:** uma amostra de vetor de atributo (**amostra de teste**) é classificada de acordo com a superfície de separação obtida na fase de treinamento

Exemplo de scatter plot para visualização do espaço de atributos

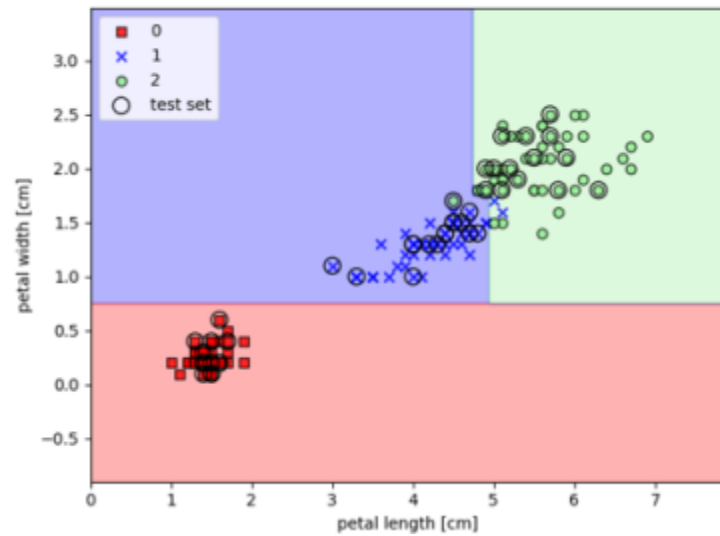


Tipos de classificadores

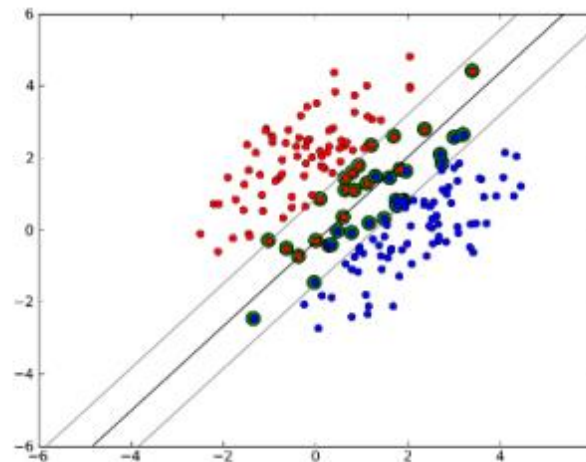


Baseado em instâncias
ou em vizinhança

Máquinas de vetores
de suporte (SVM)



Árvore de decisão



■ Como realizar a classificação dos atributos?

- Há vários tipos de classificadores
 - Classificadores com base em instâncias
 - Árvores de decisão
 - Classificadores bayesianos
 - Redes neurais
 - Máquinas de Vetores de suporte (SVM)
 - Etc.
- Diferenciam-se na forma como encontram a sua superfície de separação
- Brincando com redes neurais
 - <http://playground.tensorflow.org>

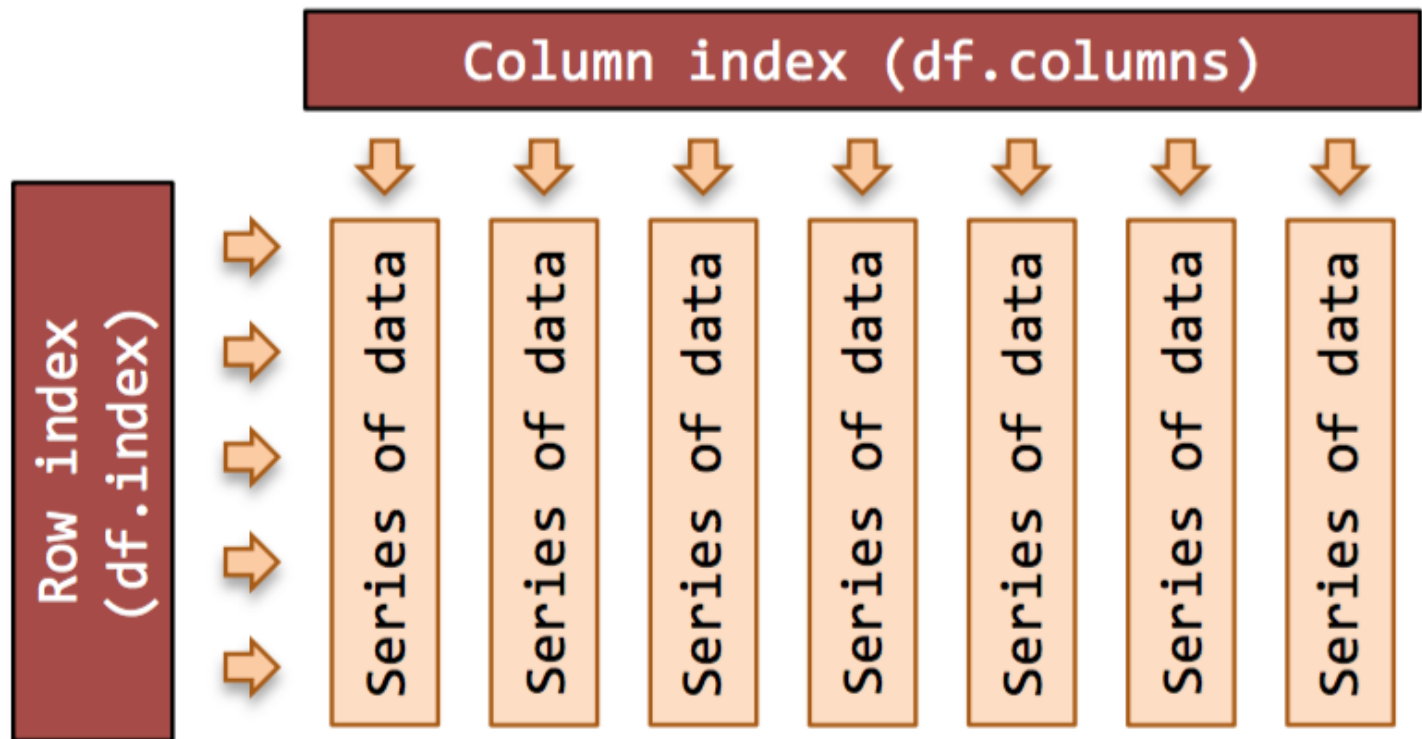
Ambiente de execução dos algoritmos de Machine Learning

- Aqui iremos usar dois módulos importantes do Python:
 - `scikit-learn`: módulo mais conhecido para a execução de algoritmos relacionados a ML
 - `pandas`: módulo contendo a estrutura para ler, armazenar e gravar uma tabela de dados
- No laboratório, esses pacotes estão instalados no ambiente Anaconda 3, cujo caminho deve ser configurado no novo projeto do PyCharm:
 - `C:\ProgramData\Anaconda3\python.exe`
- Para instalar esses módulos na versão padrão do Python 3, podemos usar o instalador de pacotes:
 - `pip3 install scikit-learn`
 - `pip3 install pandas`

■ O pacote **pandas**

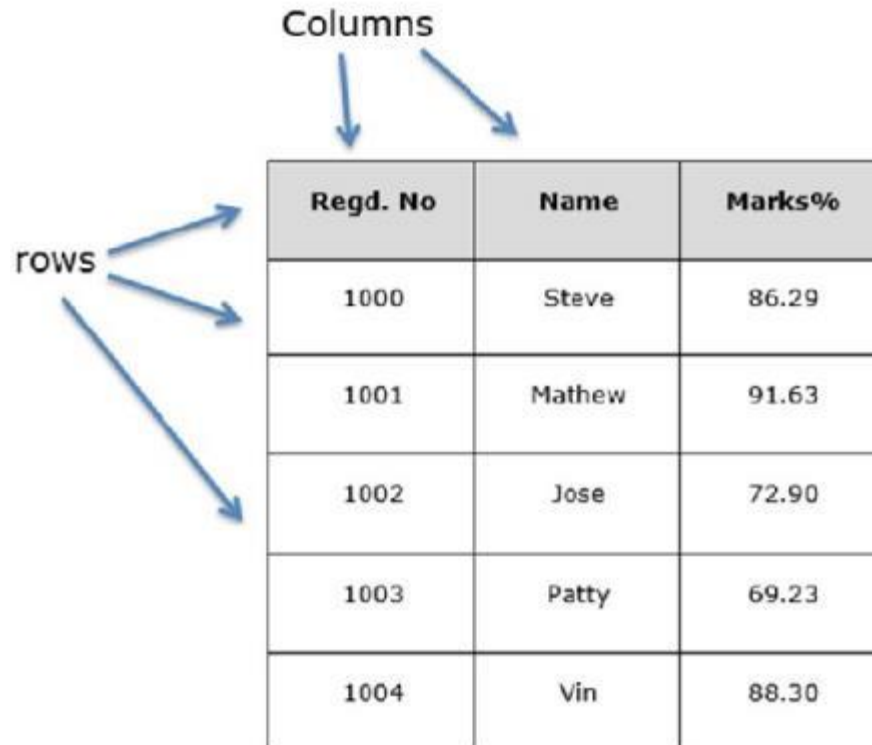
- `pandas` é uma ferramenta para manipulação de dados em alto nível.
- A estrutura de dados principal desse pacote é o `DataFrame`, que se assemelha a uma tabela de um banco de dados:
 - As colunas possuem dados de mesma natureza, e é implementada através da classe `Series` do `pandas`, que se assemelha ao array do `numpy`
 - Cada linha corresponde a uma entrada de dados
- Aprenderemos a manipular um `DataFrame` de acordo com as necessidades da disciplina
 - Vamos manipular as estruturas de dados apresentadas através do notebook `IntroPython2.ipynb`

Representação de um DataFrame



<https://www.kaggle.com/timolee/a-home-for-pandas-and-sklearn-beginner-how-tos>

Exemplo de um DataFrame



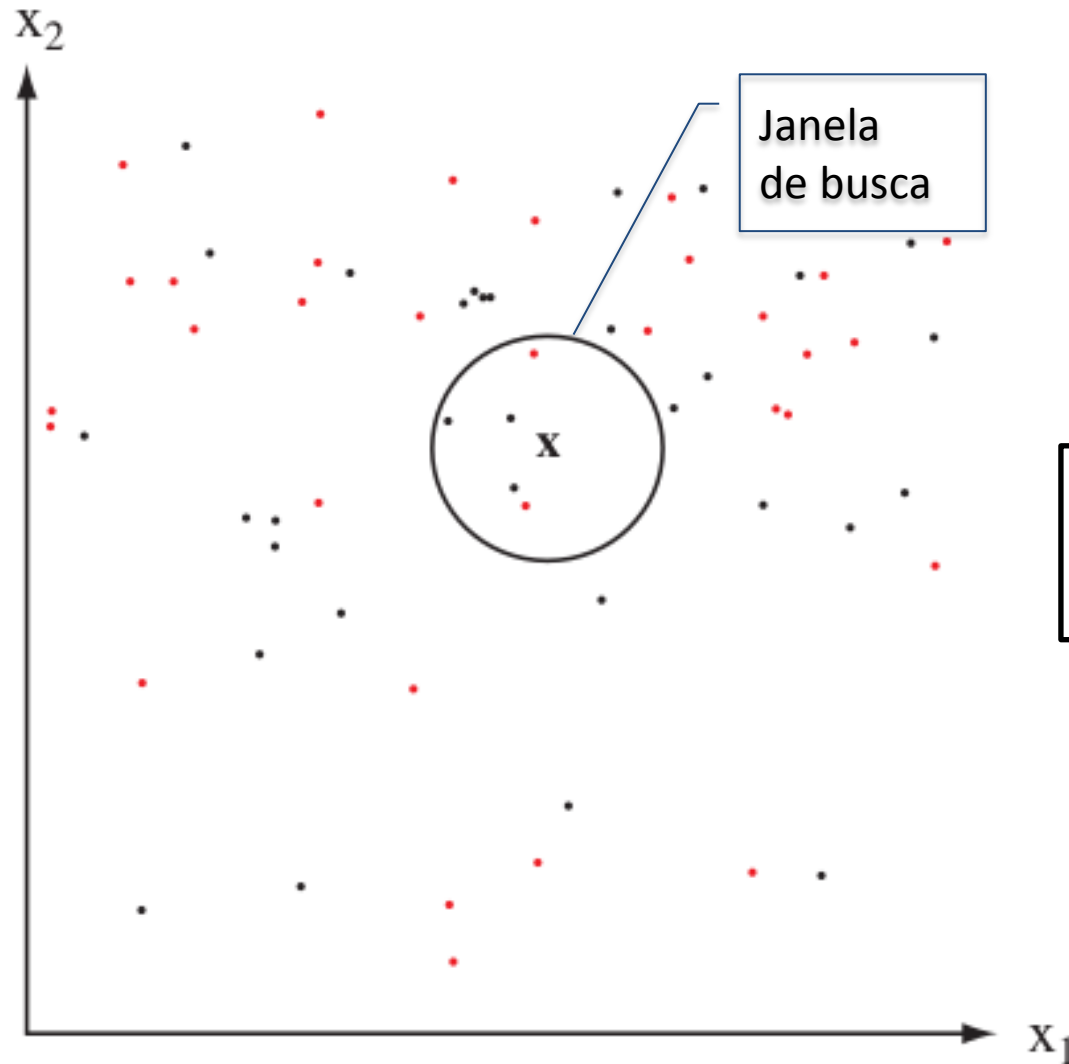
The diagram shows a table representing a DataFrame. The columns are labeled 'Regd. No', 'Name', and 'Marks%'. The rows are labeled with student IDs: 1000, 1001, 1002, 1003, and 1004. Blue arrows point from the labels 'Columns' and 'ROWS' to their respective parts of the table.

Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

■ Classificadores simples: classificação com base em instâncias

- Uma forma simples de realizar a classificação de padrões é utilizar diretamente as amostras de treinamento rotuladas para definir a superfície de separação e decidir a classe com base em uma comparação direta
- Esse tipo de classificação é denominada **classificação com base em instâncias**, cujas técnicas se diferenciam pela estratégia de comparação entre o exemplo de teste e as amostras de treinamento
- O classificador baseado em instâncias mais conhecido é o chamado **k-vizinhos-mais-próximos** (*k-nearest-neighbors*, ou kNN)
 - O vetor de atributos é comparado com um certo número (k) de amostras de treinamento que mais se assemelham a ele
 - A classe que for representada em mais rótulos nas amostras de treinamento selecionada é escolhida para o vetor de atributos

Classificação por 5-vizinhos-mais-próximos

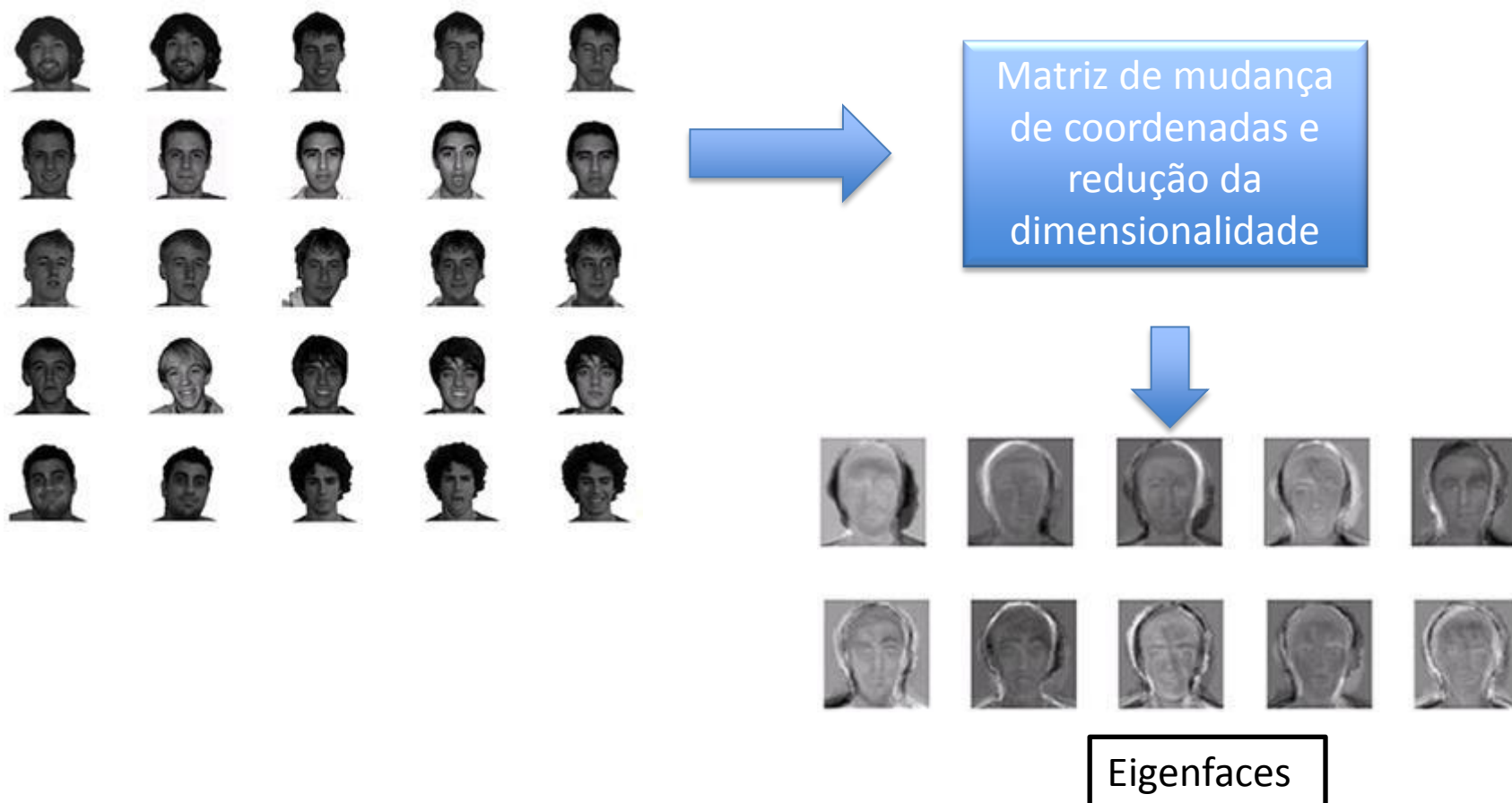


Executar o KNN
no notebook
knn.ipynb

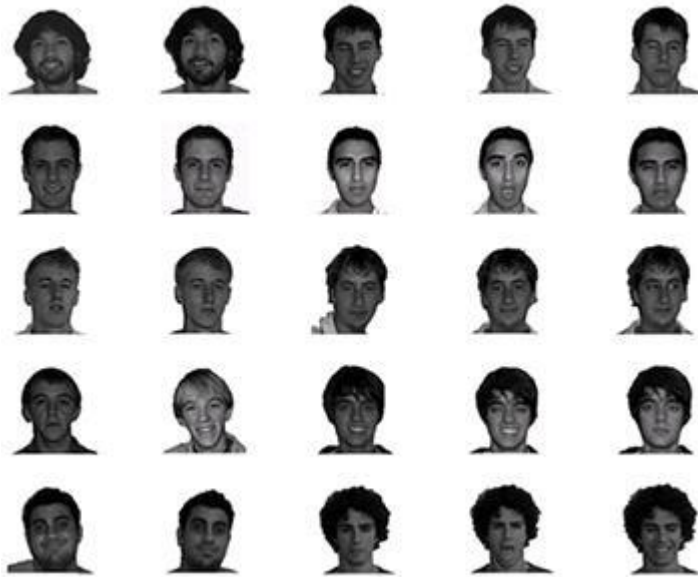
Caso de uso: Eigenfaces

- A técnica de Eigenfaces se baseia na técnica de kNN para o reconhecimento facial
 - Em vez de detectar a posição de pontos de controle no rosto da pessoa os dados brutos constituem os próprios pixels da imagem
 - Para evitar que o vetor de atributos seja muito grande, e aproveitando para ressaltar as características mais importantes da imagem, que passam por um processo de redução da dimensionalidade
 - Nesse processo, os atributos que serão classificados passam a ser os pesos que multiplicam as chamadas Eigenfaces, que são imagens geradas no processo de treinamento por redução de dimensionalidade

Pré-processamento: o processo de geração de Eigenfaces



Fase de treinamento: gerando os atributos



Matriz de mudança
de coordenadas e
redução da
dimensionalidade

Amostra 0: $[a_{00}, a_{01}, a_{02}, \dots, a_{0N}]$

Amostra 1: $[a_{10}, a_{11}, a_{12}, \dots, a_{1N}]$

Amostra 2: $[a_{20}, a_{21}, a_{22}, \dots, a_{2N}]$

...

Amostra M: $[a_{M0}, a_{M1}, a_{M2}, \dots, a_{MN}]$



Para treinar o kNN

- $M + 1$ é o número de amostras de treinamento
- $N + 1$ é o número de Eigenfaces geradas

Quando usar o kNN?

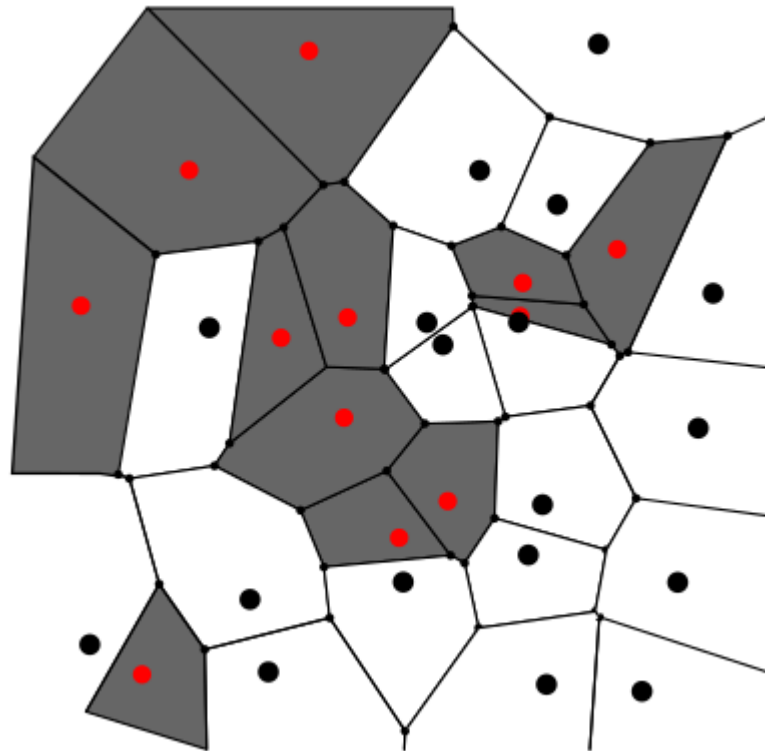
■ Vantagens:

- O treinamento é muito simples, bastando registrar as amostras de treinamento, possivelmente em alguma estrutura de dados que agilize a recuperação de pontos por proximidade (indexação espacial)
- As amostras de classes diferentes induzem uma superfície de separação bastante complexa, que poderia ser muito difícil de obter com outros classificadores
 - É o que ocorre com Eigenfaces: uma vez que a mesma pessoa pode aparecer nas imagens de treinamentos em ângulos diferentes, regiões bastante distintas do espaço de atributos deverão ser atribuídas à mesma pessoa, tornando a superfície de separação bem irregular

■ Desvantagens:

- O classificador pode ocupar muita memória de o número de amostras de treinamento for muito grande
- Encontrar o valor de k que apresente os melhores resultados é um processo trabalhoso, já que o número de opções é bem grande

Superfície de separação para $k = 1$



Considerações sobre kNN

- O número de vizinhos a serem pesquisados (k) é, em geral, ímpar
- A comparação entre a amostra de teste e as amostras de treinamento é, em geral, baseada na distância euclidiana.
 - Se o vetor de atributos for $v = (x, y, z)$, a distância euclidiana entre as amostras v_1 e v_2 será dada por:
 - $||v_1 - v_2|| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$
- Se houver muitas amostras de treinamento, pode ser inviável ter que mantê-las todas na memória para realizar a classificação
- Além do mais, para encontrar os k vizinhos mais próximos é necessário percorrer todas as amostras de aprendizado e medir a distância até a amostra de teste, o que pode demandar muito tempo computacional
 - Para melhorar o desempenho da classificação, as amostras de treinamento são, em geral, armazenadas em uma estrutura denominada **kD-tree**, onde **kD** é o número de atributos empregados na classificação, ou em uma **ball-tree**, que performa melhor para um número grande de dimensões.

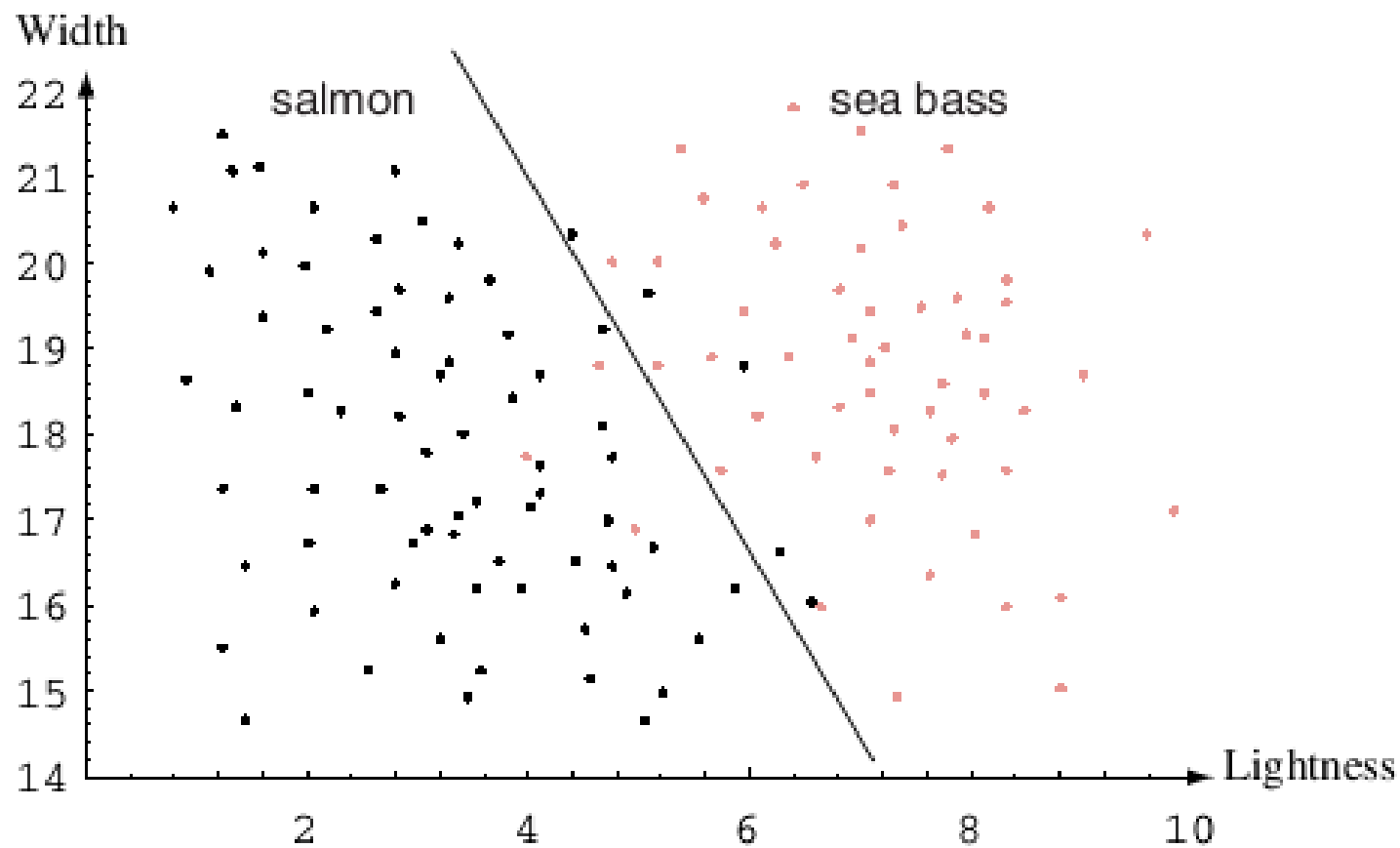
■ Tipos de classificadores

- Com base no tipo de superfície de separação entre as regiões correspondentes às diferentes classes, o classificador pode ser:
 - **Linear:** define separadores lineares entre as regiões, como retas ou planos
 - **Não-linear:** define separadores não-lineares, como superfícies quadráticas, regiões fechadas e outras formas genéricas
- Como você classificaria:
 - KNN
 - Árvores de decisão
 - SVM

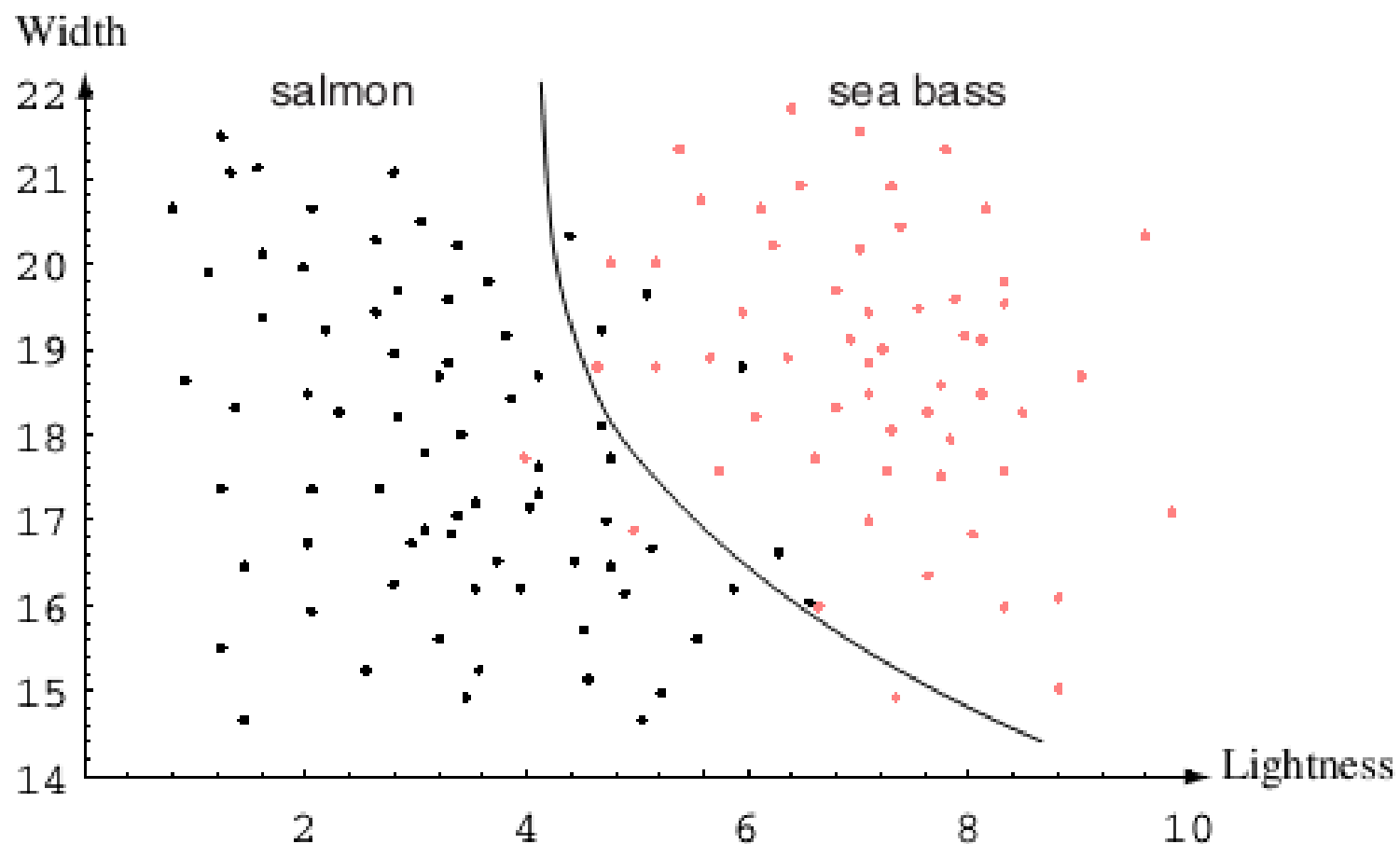
Exercício

- Escolha outro conjunto de dados do site de Machine Learning da UCI (<https://archive.ics.uci.edu/ml/datasets.html>) para refazer a análise por kNN. Use a interface do PyCharm
- Tente manter pelo menos 3 atributos para fazer a classificação de novas amostras de teste
- Para visualizar os resultados, escolha os dois eixos que você acha mais importante

Exemplo de classificador linear



Exemplo de classificador não-linear



■ Condensando o conhecimento

- O que é Machine Learning?
- Quais as três modalidades de aprendizado segundo o livro texto?
- O que é reconhecimento de padrões?
- Quais são os três passos do reconhecimento de padrões?
- O que são os atributos?
- O que é um classificador?
- Que tipos de classificadores nós vimos nesta aula?

REFERÊNCIAS

- Stuart Russel & Peter Norvig. Inteligência Artificial – tradução da 2ª ed. Editora Campus, 2004, capítulo 18
- Duda, Hart & Stork. Pattern Classification, 2nd. Ed., 2000, capítulo 4





Copyright © 2018 Prof. Antonio Selvatici

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).