



FIAP

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISRUPTIVE ARCHITECTURES: IOT, IOB & IA

04 – Modelos da Aprendizado Supervisionado – Classificação

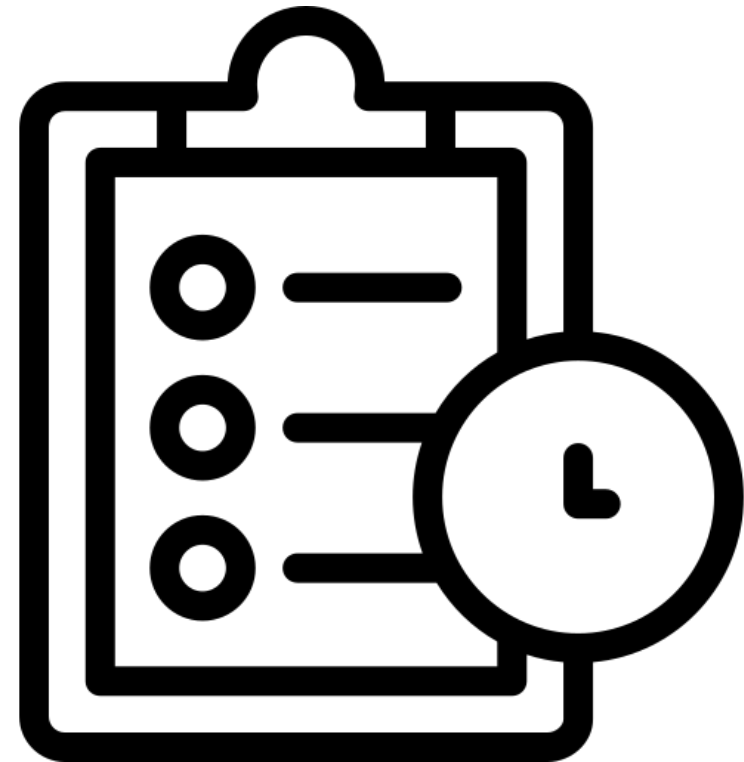


Prof. Airton Y. C. Toyofuku



profairton.toyofuku@fiap.com.br

- Revisão
- O que é Aprendizado Supervisionado
- Tipos de Problemas de Aprendizado Supervisionado
- Algoritmos de Classificação
- Métricas de Avaliação para Problemas de Classificação
- Exercícios



❖ O que é Machine Learning?

- Técnica da Inteligência Artificial que permite que o computador aprenda uma tarefa específica.

❖ Quais os tipos de aprendizado?

- Supervisionado;
- Não Supervisionado;
- Por reforço;

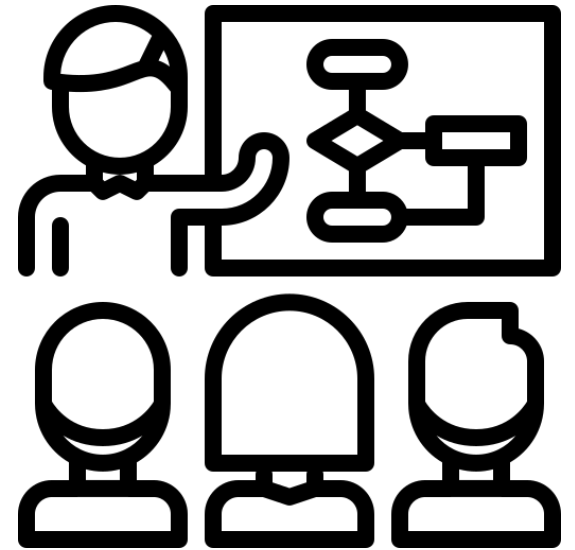
❖ O que precisamos para implementar uma ML?

- Dados!!!!

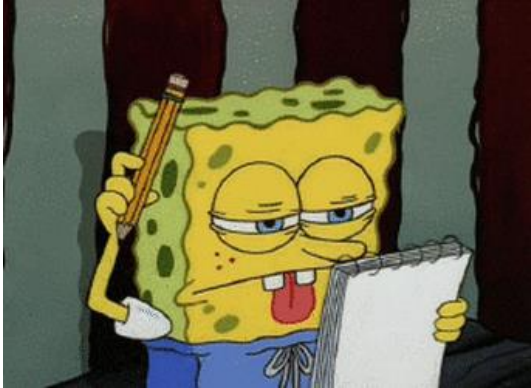


O que é Aprendizado Supervisionado

- ❖ Consiste em treinar um modelo com um grupo de dados já rotulados;
- ❖ Ou seja, nós já sabemos qual é a resposta ou correlação entre os dados;
- ❖ O Objetivo é fazer o modelo entender essa correlação e generalizar para novos dados que ele não conhece;



Tipos de Problemas de Aprendizado Supervisionado



❖ Existem dois tipos de problemas em que Podemos aplicar Aprendizado Supervisionado:

❖ De Classificação:

- Sim ou Não;
- Vai comprar ou Não vai comprar;
- É do tipo A, do tipo B ou do tipo C;

❖ De Regressão:

- Prever um determinado número;
- Quanto custa uma casa;
- Qual vai ser a quilometragem;

❖ Regressão Logística:

- Técnica da estatística que procura correlacionar dados e produzir uma saída binária do tipo 0 ou 1 (Sim ou Não).

❖ Árvore de Decisão:

- Algoritmo que divide repetidamente os dados em subconjuntos com base nas suas características até que chegar a um subconjunto único.

❖ K-Nearest Neighbors (K-NN)

- Algoritmo que classifica o dado em um conjunto conforme a semelhança entre os seus vizinhos mais próximos.

❖ Naive Bayes

- Usa o teorema de Bayes para estimar a probabilidade de um dado pertencer a um conjunto com base em suas características independentes.

❖ Support Vector Machines (SVM)

- Separa os dados em espaços multimensionais e procura encontrar o melhor hiperplano que separa as classes.

Métricas de Avaliação para Problemas de Classificação

❖ Matriz de Confusão

- Mostra a contagem de previsões corretas e incorretas em uma matriz de Quatro Quadrantes: Verdadeiro Positivo (TP), Falso Positivo (FP), Verdadeiro Negativo (TN) e Falso Negativo (FN)

❖ Acurácia

- Diz o quanto o modelo acertou das previsões possíveis: $\text{Accuracy} = (TP+TN) / (TP + FP + TN + FN) = \text{Corretos} / \text{Total}$

❖ Recall

- O quão bom o modelo é em prever classes positivas: $\text{Recall} = TP / (TP + FN)$

❖ Precisão

- O quão correto o modelo foi em prever classes positivas: $\text{Precision} = TP / (TP + FP)$

❖ F-Score

- Mostra o balanço entre a precisão e o recall: $\text{F-Score} = 2 * ((\text{precision} * \text{recall}) / (\text{precision} + \text{recall}))$

Exercícios

O conjunto de dados Iris é um conjunto de dados clássico de aprendizado de máquina que contém informações sobre três espécies diferentes de flores iris. Cada flor iris é caracterizada por quatro atributos: comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala. O objetivo deste exercício é comparar o desempenho de diferentes algoritmos de classificação para classificar as flores iris com base nesses atributos.



1. Carregue o conjunto de dados Iris do pacote scikit-learn.
2. Divida o conjunto de dados em conjuntos de treinamento e teste usando a função `train_test_split` do pacote scikit-learn.
3. Padronize os dados de treinamento e de teste usando a classe `StandardScaler` do pacote scikit-learn.
4. Crie objetos para cada um dos seguintes algoritmos de classificação: Regressão Logística, Árvore de Decisão, Naive Bayes e SVM.
5. Ajuste cada um dos modelos aos dados de treinamento.
6. Use cada modelo treinado para prever as classes das flores iris no conjunto de teste.
7. Avalie a precisão de cada modelo usando a função `accuracy_score` do pacote scikit-learn.
8. Compare a precisão dos diferentes modelos e tire conclusões sobre qual algoritmo de classificação teve o melhor desempenho.

Exercícios - Dicas

1. Consulte a documentação do pacote scikit-learn para obter informações sobre como carregar o conjunto de dados Iris e como usar a função `train_test_split`.
2. Use o parâmetro `random_state` ao chamar a função `train_test_split` para garantir que o conjunto de dados seja dividido de maneira consistente.
3. Certifique-se de padronizar os dados de treinamento e teste antes de ajustar cada modelo. Use a classe `StandardScaler` do pacote scikit-learn para fazer isso.
4. Para avaliar a precisão de cada modelo, compare as classes previstas com as classes reais usando a função `accuracy_score`. Esta função retorna a proporção de amostras classificadas corretamente.
5. Para comparar a precisão dos diferentes modelos, você pode usar gráficos de barras ou tabelas para visualizar as pontuações de precisão para cada modelo.



Copyright © 2023 Prof. Airton Y. C. Toyofuku

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