

#### Inteligência Artificial Aprendizado de Máquina com Python

Florianópolis, 22 de Maio de 2019





## Licença

O conteúdo desta oficina está licenciado sob Creative Commons

Atribuição-Compartilhalgual 4.0 Internacional





# **Agenda**

- Inteligência Artificial: definição e aplicações
- Aprendizado de máquina: supervisão, treinamento e classificadores.
- Ferramentas: Python, Scikit-Learn e Jupyter notebooks.
- Prática com classificadores: árvore de decisão.
- Pygame e TensorFlow Jogando com Aprendizado de Máquina
- Lógica fuzzy





#### Grupo de Pesquisa PECCE

- Computação Científica para Engenharia
- Desde 2011
- Engenharia Elétrica e Eletrônica
- Modelagem Eletromagnética, Instrumentação, Processamento de Sinais
- Prof. Sérgio Ávila, Prof. Cesar Penz, Prof. Fernando Pacheco
- Bolsistas: Heloiza Martins, Rafael Tominaga, Suzi Yousif





# Inteligência Artificial

?





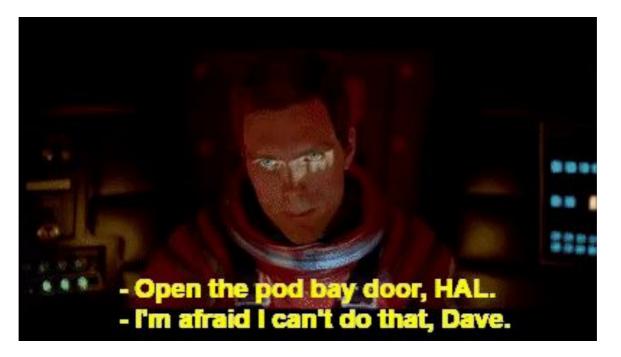
# Inteligência

- Faculdade de conhecer, compreender e aprender.
- Capacidade de compreender e resolver novos problemas e conflitos e de adaptar-se a novas situações.
- Conjunto de funções psíquicas e psicofisiológicas que contribuem para o conhecimento, para a compreensão da natureza das coisas e do significado dos fatos.





#### Inteligência Artificial



2001: Uma Odisseia no Espaço, 1968





## Inteligência Artificial

- Área de ciência da computação
- Autonomia
- Adaptatividade
- Inteligência humana exibida por máquinas
  - Não são necessariamente inteligentes, mas agem com inteligência





## Grupo de Pesquisa PECCE

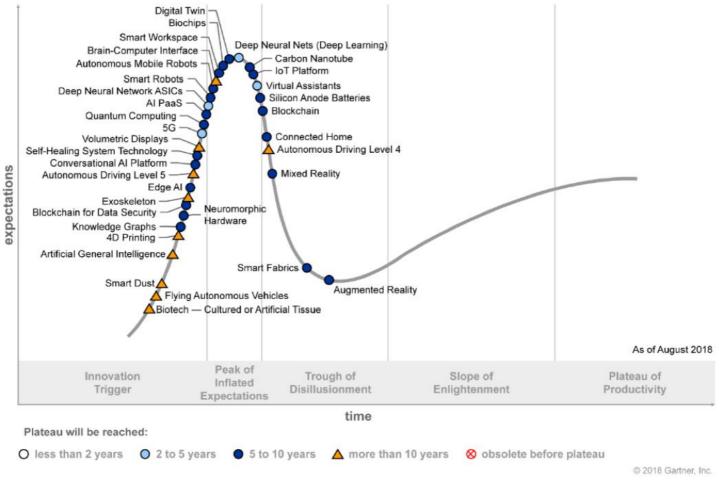
 TCCs com foco em identificação de desbalanceamento de motores elétricos





# INSTITUTO **FEDERAL** Santa Catarina

#### **Hype Cycle for Emerging Technologies (2018)**







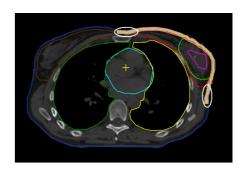
# **Aplicações**

- Recomendação de conteúdo
- Processamento de vídeo e Imagem
- Assistente virtual
- GPS

















#### Inteligência Artificial

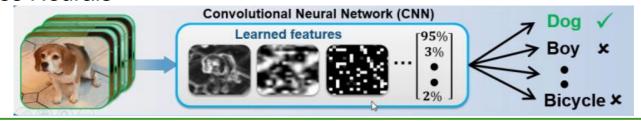
#### **Machine Learning:**

Permite que máquinas aprimorem suas previsões em determinadas tarefas, a partir de experiências (analisar dados e aprender com os mesmos)



#### **Deep Learning:**

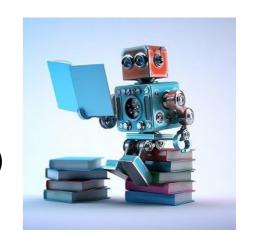
Técnicas de Machine Learning com várias camadas de processamento - Redes Neurais





#### Aprendizado de Máquina

- 1. Coleta de dados
- 2. Exploração e preparação dos dados
- 3. Treinar o modelo
- 4. Avaliar o modelo (exatidão das previsões)
- 5. Aperfeiçoamento do modelo

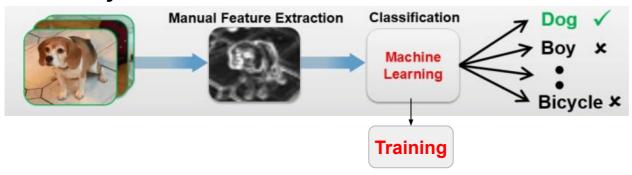






#### Aprendizado Supervisionado

- Tem as mesmas características anteriores
- Há conjunto de dados rotulados



Var. Independentes	Var. Dependentes
Anos de Carreira, Formação, Idade	Salário
Idade do Carro, Idade do Motorista	Risco de Acidente Automotivo
Texto de um livro	Escola Literária
Temperatura	Receita de venda de sorvete
Imagem da Rodovia	Ângulo da direção de um carro autônomo
Histórico escolar	nota no ENEM





#### Aprendizado não Supervisionado

Abordar problemas com pouca ou nenhuma ideia dos resultados

Dados	Forma Representativa	
Transações bancárias	Normalidade da transação	
Registros de Compras	Associação entre produtos	
Dados Multidimensionais	Dados com dimensão reduzida	
Registros de Compras	Perfil dos consumidores	
Palavras em um texto	Representação matemática das palavras	





Treino

**Teste** 

# Separação / Preparação dos dados

PassengerId	Survived	Pclass	Name	Sex	Age	Fare
1	0	3	Braund, Mr. Owen Harris	male	22	7.25
2	1	1	Cumings, Mrs. John Bradley	female	38	71.2833
3	1	3	Heikkinen, Miss. Laina	female	26	7.925
4	1	1	Futrelle, Mrs. Jacques	female	35	53.1
5	Θ	3	Allen, Mr. William Henry	male	35	8.05
6	Θ	3	Moran, Mr. James	male	nan	8.4583
7	Θ	1	McCarthy, Mr. Timothy J	male	54	51.8625
8	Θ	3	Palsson, Master. Gost	male	2	21.075
9	1	3	Johnson, Mrs. Oscar W (Eli	female	27	11.1333
10	1	2	Nasser, Mrs. Nicholas (Ad	female	14	30.0708
11	1	3	Sandstrom, Miss. Margue	female	4	16.7
12	1	1	Bonnell, Miss. Elizab	female	58	26.55

Classe





Características



# Algoritmos de aprendizagem supervisionados

Model	Learning task			
Supervised Learning Algorithms				
Nearest Neighbor	Classification			
Naive Bayes	Classification			
Decision Trees	Classification			
Classification Rule Learners	Classification			
Linear Regression	Numeric prediction			
Regression Trees	Numeric prediction			
Model Trees	Numeric prediction			
Neural Networks	Dual use			
Support Vector Machines	Dual use			



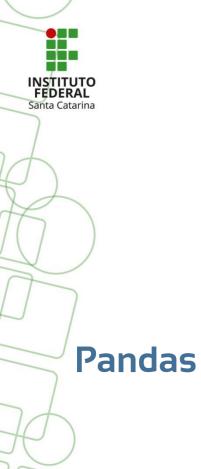


# **Python**

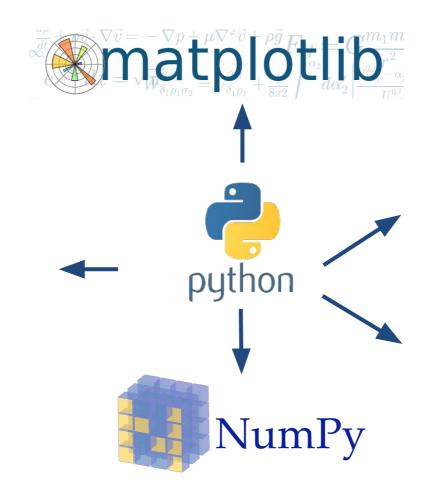


- 1991
- Linguagem de programação de alto nível
- Linguagem expressiva
- Legibilidade dos códigos
- Multiparadigma (orientação a objetos, imperativa, funcional)













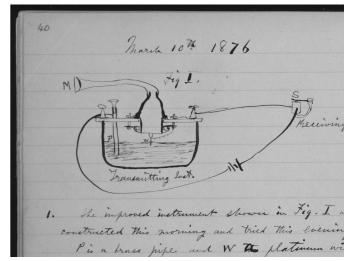


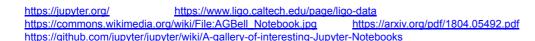


#### Jupyter notebook



- Aplicação web de código aberto
- Incorpora em um mesmo arquivo
  - equações
  - código-fonte e resultados
  - texto
  - gráficos
- Ciência aberta
  - LIGO (Prêmio Nobel de Física de 2017)
  - https://www.gw-openscience.org/tutorials/













- Pacote fundamental para computação científica com o Python.
- Ferramenta poderosa para trabalhar com vetores de N-dimensão
- Tipos de dados arbitrários podem ser definidos
- Operações com matrizes, incluindo manipulação matemática, lógica, de formas, classificação, seleção, I/O, transformadas discretas de Fourier, álgebra linear básica, operações estatísticas básicas, simulação aleatória, etc..





#### **Pandas**



- Rápido e eficiente para a manipulação de dados (DataFrame) com índices;
- Ler e escrever em dados com diferentes formatos: CSV, arquivos de texto, Microsoft Excel, SQL.
- Colunas podem ser inseridas ou/e excluídas;
- Permite a operações em certo conjuntos de dados;
- Alta performance para mesclar e juntar conjunto de dados.

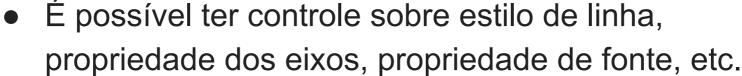


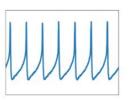


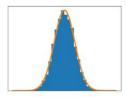
#### matplotlib

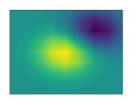


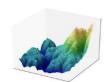
- Biblioteca de desenho
- Pode gerar:
  - Gráficos;
  - Histogramas;
  - Espectros de Potência;
  - Gráficos de Dispersão;
  - Gráficos de Barra...











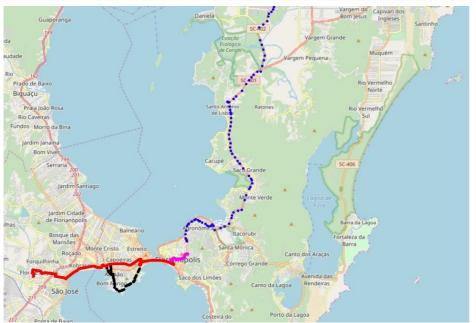




#### **Biblioteca Folium**



```
#Pontos do Abraão
track_points_Abraao = pd.read_csv('track_points_Abraao.csv')
for _,track_point in track_points_Abraao.iterrows():
        folium.Circle(radius = 1,
                     location = [track_point['Y'], track_point['X']],
                     color = '#000000',
                     fill = False
                     ).add_to(floripa)
```





Index	X	Y
0	-48.5971	-27.6052
1	-48.5976	-27.6043
2	-48.5983	-27.6027
3	-48.5985	-27.6024
4	-48.5971	-27.6041
5	-48.5959	-27.6056
6	-48.5951	-27.6067
7	-48.5953	-27.6048
8	-48.5954	-27.6035
9	-48.5955	-27.6026
10	-48.5956	-27.602
11	-48.5957	-27.6015
12	-48.5957	-27.6012
13	-48.5959	-27.6062





#### scikit-learn

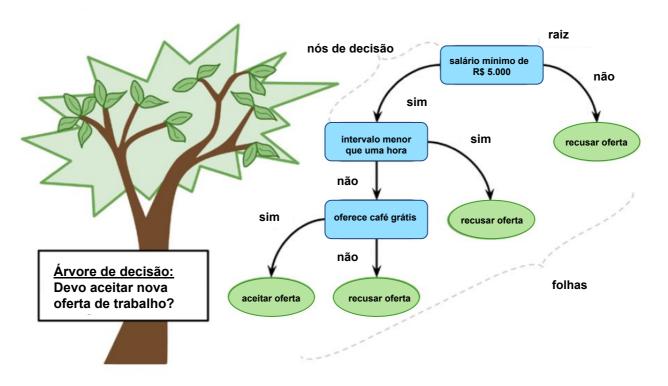


- Ferramentas simples e eficientes para mineração e análise de dados
- Acessível a todos e reutilizável em vários contextos
- Construído em NumPy, SciPy e matplotlib
- Código aberto, comercialmente utilizável
- Classificadores de aprendizagem supervisionada:
  - Decision Trees
  - Nearest Neighbors (kNN)
  - Gaussian Processes
  - Naive Bayes e muitos outros





# Árvore de Decisão (Decision Tree Algorithm)







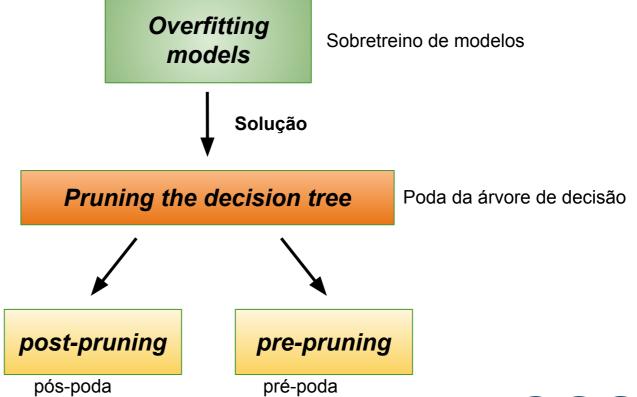
# Árvore de Decisão

Vantagens	Desvantagens
<ul> <li>Fácil de interpretar, entender e visualizar.</li> <li>Possibilidade de visualizar as decisões do algoritmo para determinado dado.</li> <li>Pode ser aplicado em pequenos e grandes conjuntos de dados.</li> <li>Mais eficiente que outros modelos mais complexos.</li> </ul>	<ul> <li>Pequenas mudanças nos dados de treinamento podem resultar em grandes mudanças na lógica de decisão.</li> <li>Pode ter problema para modelar alguns relacionamentos devido à dependência das divisões dos dados.</li> <li>Árvores grandes podem ser difíceis de serem interpretadas e as decisões que serão feitas podem parecer contraintuitivas</li> <li>É fácil de apresentar problemas de overfitting ou underfitting</li> </ul>





#### Árvore de Decisão







#### scikit-learn - Decision Tree



#### Importar a classe que contém o método de classificação

from sklearn.tree import DecisionTreeClassifier

#### Crie uma instância da classe

DTC = DecisionTreeClassifier ( max\_features=10, max\_depth=5)

#### Ajustar a instância nos dados e, em seguida, prever o valor esperado

DTC = DTC.fit (x\_train, y\_train)

y\_predict = DTC.predict (x\_test)

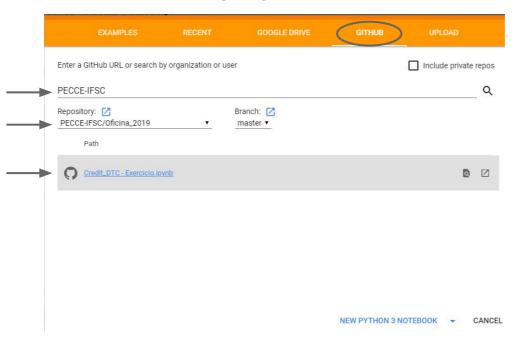




#### scikit-learn - Decision Tree



- Exercício: Aplicação do classificador Decision Tree
  - o link:
    - https://colab.research.google.com/notebooks/welcome.ipynb#recent=true





#### Avaliação da Oficina

Pedimos a gentileza de avaliar a oficina em

https://forms.gle/tTGQ3XWamWMPkX16A



#### Referências

LANTZ, Brett. **Machine Learning with R.** 2. ed. Birmingham: Packt Publishing Ltd., 2015. TROTTER, Cate. **The unknown technologies retail needs to be aware of.** 2017. Disponível em: <a href="https://www.insider-trends.com/the-unknown-technologies-shaping-our-future/">https://www.insider-trends.com/the-unknown-technologies-shaping-our-future/</a>>. Acesso em: 27 set. 2018.

#### Recomendação de Conteúdo

- <a href="https://www.datacamp.com/courses/intro-to-python-for-data-science">https://www.datacamp.com/courses/intro-to-python-for-data-science</a>
- https://www.datascienceacademy.com.br/
- https://course.elementsofai.com/
- <a href="https://software.intel.com/pt-br/ai-academy/students/kits/machine-learning-501">https://software.intel.com/pt-br/ai-academy/students/kits/machine-learning-501</a>

