

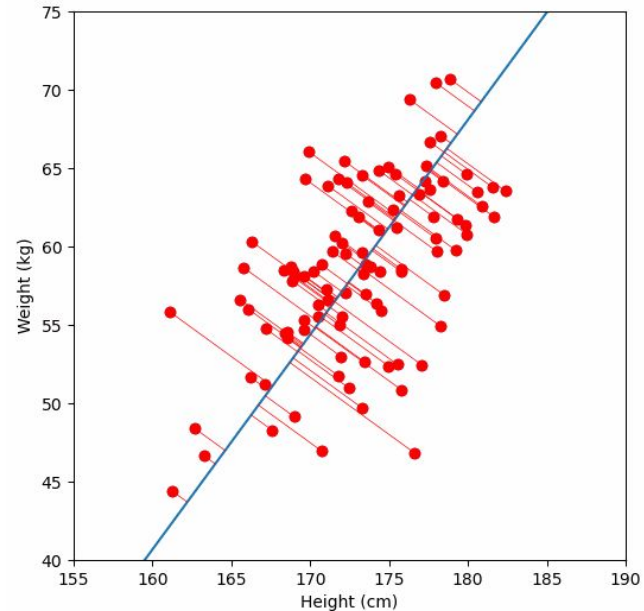
Redução de Dimensionalidade

PCA (Principal Component Analysis)



PCA (Principal Component Analysis)

O PCA simplifica dados complexos reduzindo suas dimensões, encontrando as características mais importantes que explicam a maior parte da variação nos dados. Ele identifica as direções de maior dispersão (componentes principais) e, ao manter apenas as principais, reduz a complexidade preservando informações relevantes. Isso auxilia na visualização, remoção de ruído e melhora o desempenho de algoritmos de aprendizado de máquina, sendo aplicado em áreas como compressão de imagem, reconhecimento facial e finanças.



PCA: Reduzir Dimensões

Reduzindo a dimensionalidade das colunas 'bmi', 'bloodpressure', 'diabetic' e 'smoker' para um único componente principal, pc1.

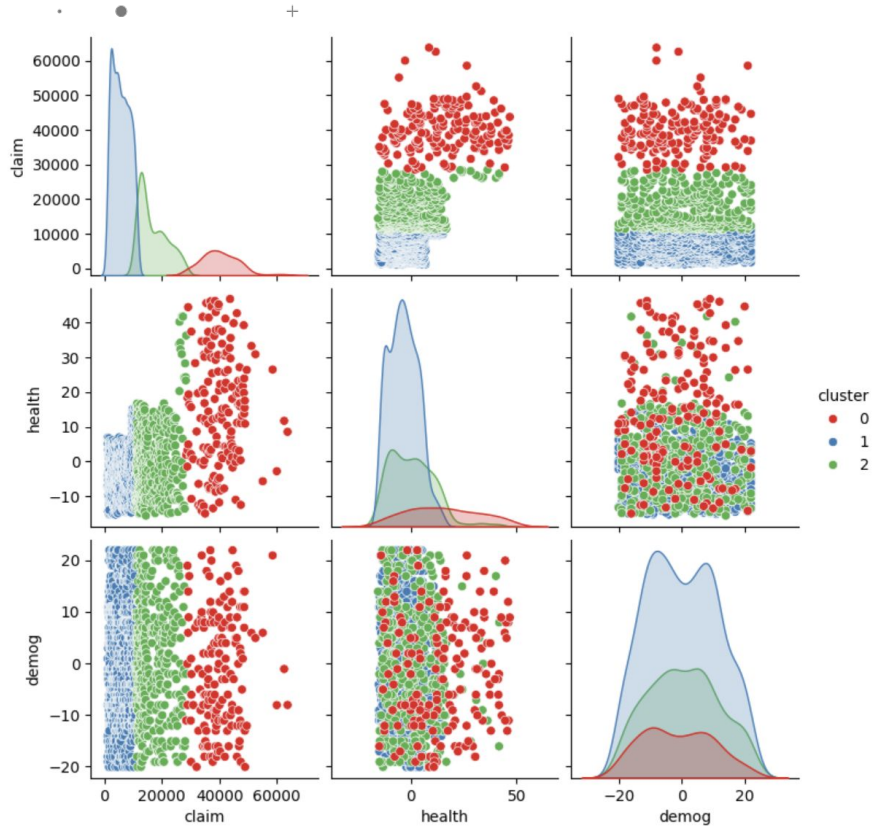
```
pca1 = PCA(n_components=1)
pca1 = pca1.fit_transform(df[['bmi', 'bloodpressure', 'diabetic', 'smoker']])
pca1

array([[ -3.96922345],
       [-7.20971262],
       [ 6.88770503],
       ...,
       [-2.74972123],
       [11.72650493],
       [ 8.56825067]])
```

Agora, vamos reduzir a dimensionalidade das colunas 'age', 'is_male' e 'children' para um único componente principal, armazenando, pca2

```
pca2 = PCA(n_components=1)
pca2 = pca2.fit_transform(df[['age', 'is_male', 'children']])
```

PCA: Recalculating KMeans



```
import seaborn as sns
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3)
kmeans.fit(df_new)
df_new['cluster'] = kmeans.labels_
sns.pairplot(df_new, hue='cluster',
```



MAD MEN

Intuición · Creatividad · Appeal



MATH MEN

Analytics · Resultados · Lógica



Atividade: PCA

- Escolha um conjunto de dados do Kaggle utilizando técnicas de clusterização e redução de dimensionalidade (PCA).
- O aluno deverá realizar o pré-processamento dos dados (tratamento de valores ausentes, codificação de variáveis categóricas e escalonamento de numéricas), aplicar o PCA para reduzir a dimensionalidade, implementar algoritmos de clusterização como K-Means,
- Analise e interpretar os clusters resultantes, criar visualizações para auxiliar a compreensão dos resultados e entregar um relatório conciso com o código Python comentado, descrevendo todas as etapas do projeto, desde a justificativa da escolha do dataset até as conclusões e insights obtidos

OBRIGADO

Luiz Barboza PhD



/luizbarboza



luiz@barboza.me.uk

FIAP

Copyright © 2019 | Professor (a) Nome do Professor

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.

Projeto Integrado - Quantum Finance

- Conforme analisado nas disciplinas anteriores, você possui acesso aos dados de clientes da Quantum Finance, composto por mais de uma centena de variáveis independentes. A esta altura você já tem uma percepção da importância de cada um delas. Partindo deste ponto, realize:
 - Selecione uma das opções de base de dados (ou se preferir trabalhe as duas)
 - Apresente uma análise descritiva destes dados
 - Entenda que tipo de análise preditiva você quer realizar, se classificação ou regressão, e qual algoritmo lhe traz o melhor resultado.
 - Realize um exercício para prever/inferir esta com base em combinação de variáveis diferentes. Experimente também com diversos tipos de algoritmos. Para todos eles, mensure a assertividade tanto nos dados de treino quanto de teste. Discorra sobre esses resultados
 - Não esqueça de apresentar suas análises/conclusões de negócio ao decorrer de todo trabalho
- Opção 1:**
 - **treino:** https://raw.githubusercontent.com/lcbjrrr/QuantumF/main/base_score%20-%20tr.csv
 - **teste:** https://raw.githubusercontent.com/lcbjrrr/QuantumF/main/base_score%20-%20ts.csv
- Opção 2:**
 - **treino:** <https://raw.githubusercontent.com/lcbjrrr/QuantumF/main/dataset%20-%20tr.csv>
 - **teste:** <https://raw.githubusercontent.com/lcbjrrr/QuantumF/main/dataset%20-%20ts.csv>

Projeto Integrado - Quantum Finance

Conforme analisado nas disciplinas anteriores, você possui acesso aos dados de clientes da Quantum Finance, composto por mais de uma centena de variáveis independentes. A esta altura você já tem uma percepção da importância de cada um delas. Partindo deste ponto, realize:

1A) Apresente uma análise descritiva (EDA) dos dados, não esqueça de trazer as conclusões de negócio

1B) Realize a limpeza dos dados, e revise sua EDA.

2A) Entenda que tipo de análise preditiva você quer realizar, se classificação ou regressão, e qual algoritmo lhe traz o melhor resultado. Caso necessário realize um trabalho de feature engineering.

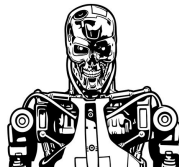
2B) Realize um exercício para prever/inferir esta com base em combinação de variáveis diferentes.

Experimente também com diversos tipos de algoritmos. Para todos eles, mensure a assertividade tanto nos dados de treino quanto de teste. Discorra sobre esses resultados. Não esqueça de realizar a separação dos dados em treino e teste.

dados: <https://raw.githubusercontent.com/lcbjrrr/QuantumF/main/credit%20-%20ok.csv>

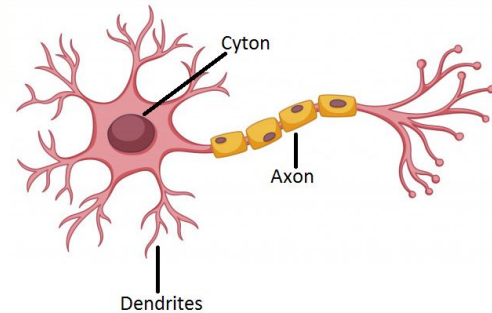
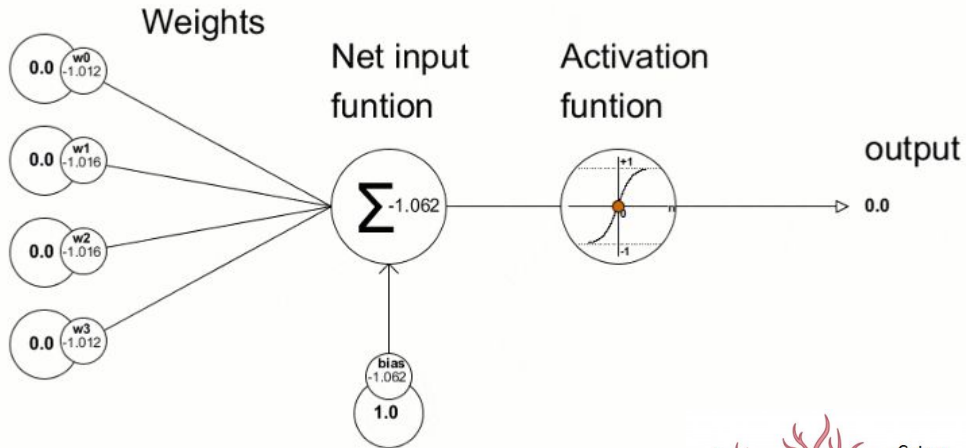
EXTRA:

Redes Neurais

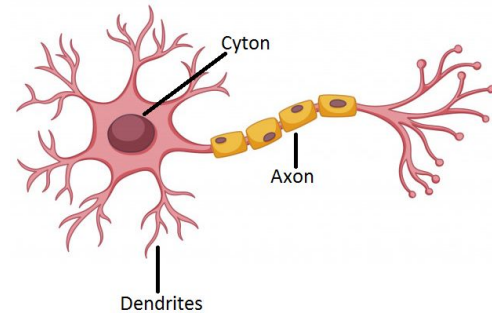
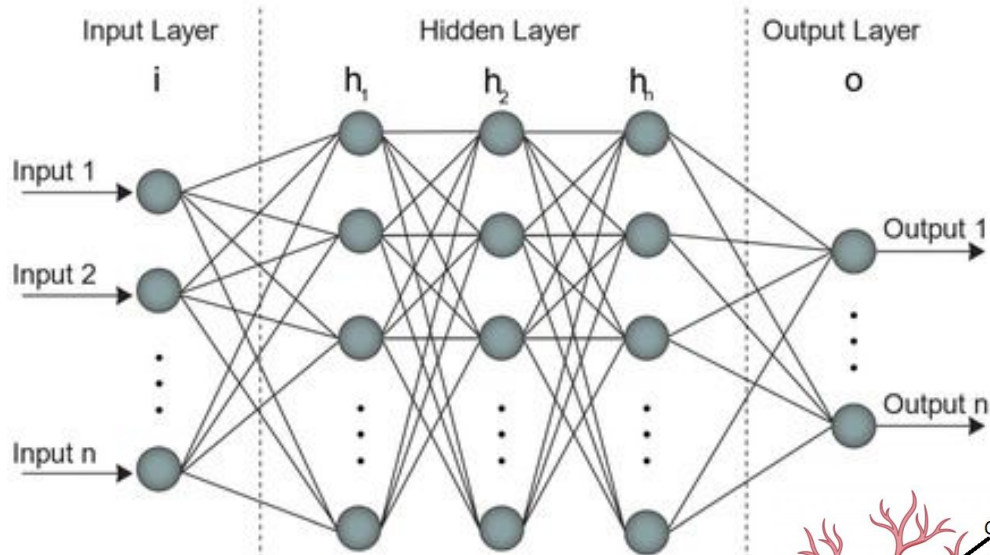


ML: Classificação - Redes Neurais

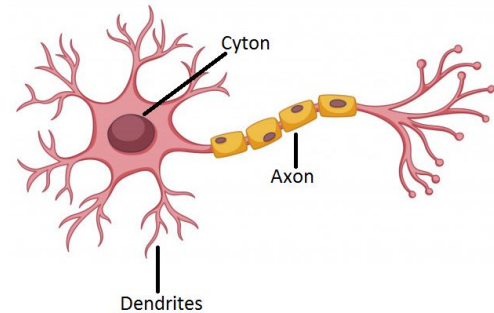
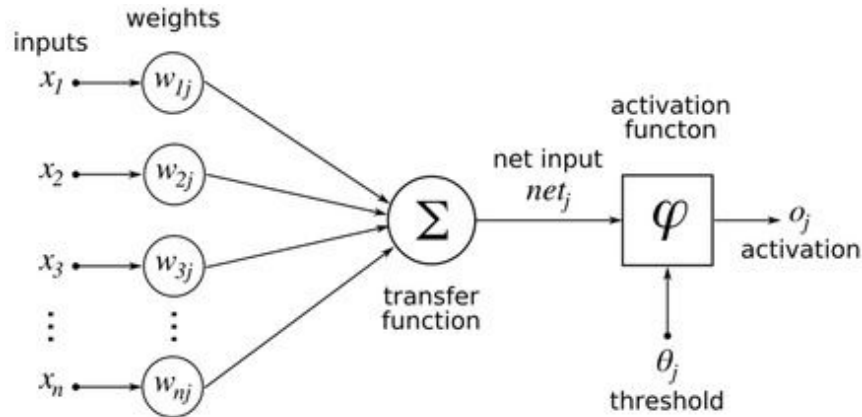
Inputs



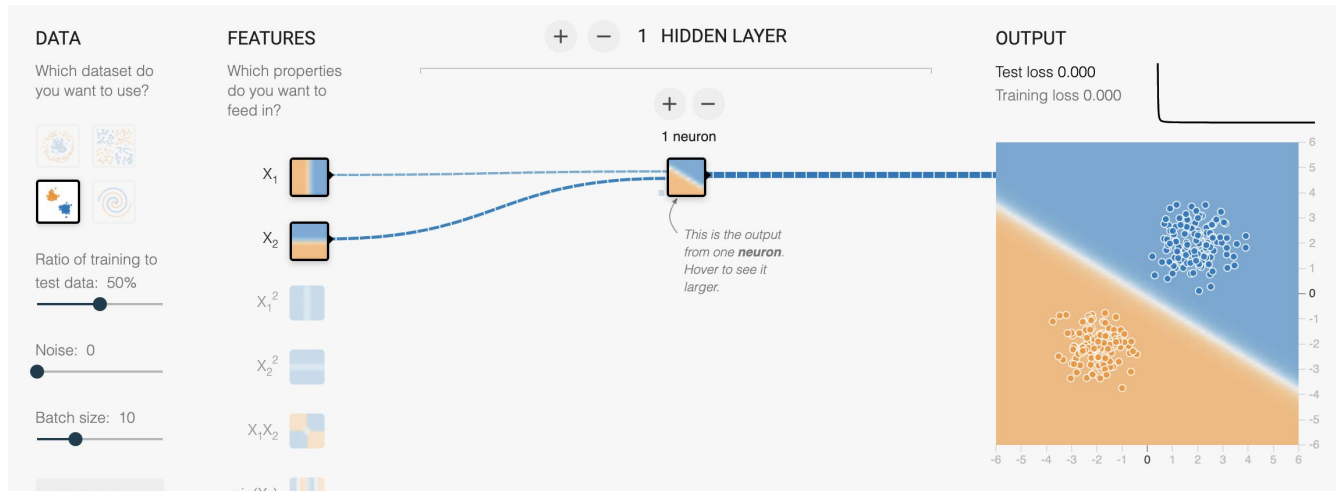
ML: Classificação - Redes Neurais



ML: Classificação - Redes Neurais



ML: Classificação - Redes Neurais



Try: <https://playground.tensorflow.org>

ML: Classificação - Redes Neurais

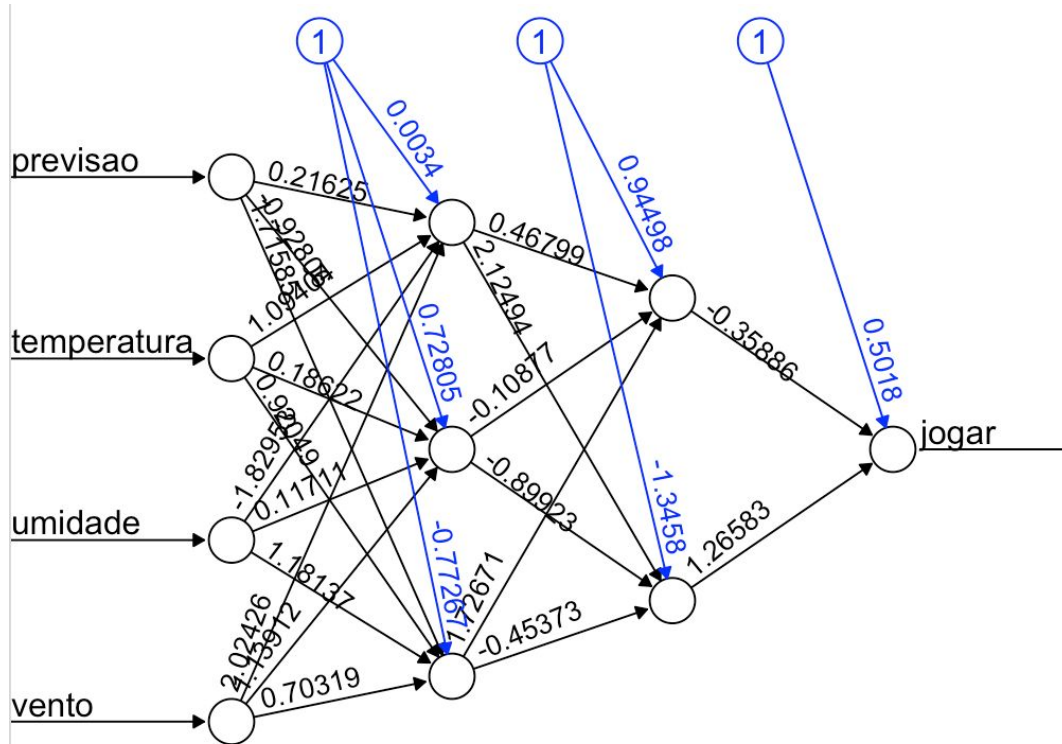
```
import pandas as pd
import tensorflow
from tensorflow import keras
```

```
volei = pd.read_csv('volei - ok.csv')
v_prev = pd.read_csv('volei - prd.csv')
```

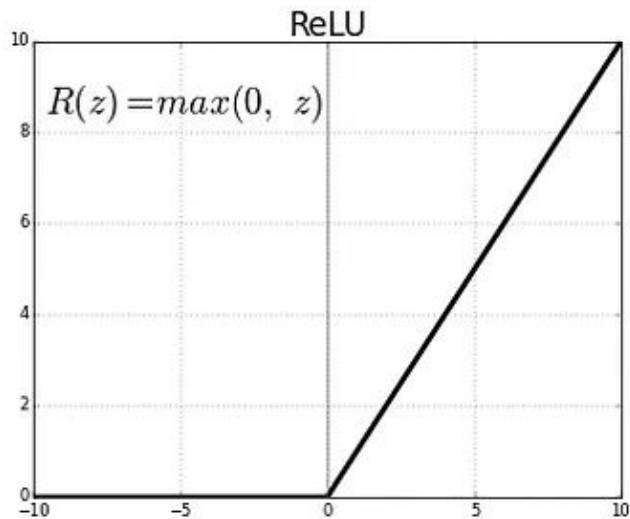
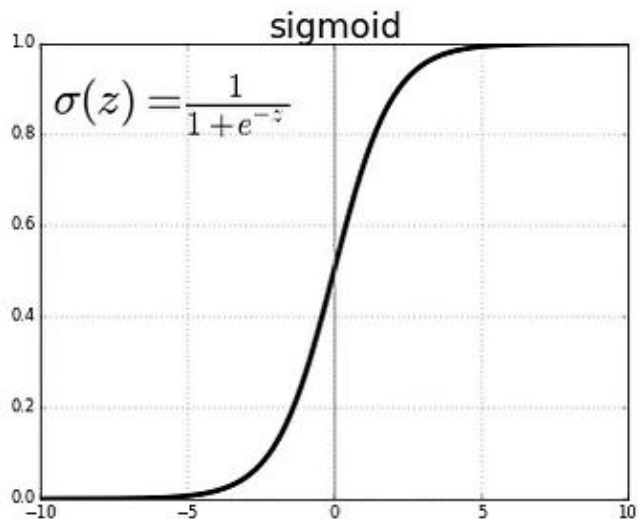
```
nn = keras.Sequential([
    keras.layers.Input(len(volei.columns)-1),
    keras.layers.Dense(3, activation=tensorflow.nn.relu),
    keras.layers.Dense(2, activation=tensorflow.nn.relu),
    keras.layers.Dense(1, activation=tensorflow.nn.sigmoid)
])
```

previsao	temperatura	umidade	vento
1	1	1	2
1	1	1	1
2	1	1	2
3	2	1	2
3	3	2	2
3	3	2	1
2	3	2	1
1	2	1	2
1	3	2	2
3	2	2	2
1	2	2	1
2	2	1	1
2	1	2	2
3	2	1	1

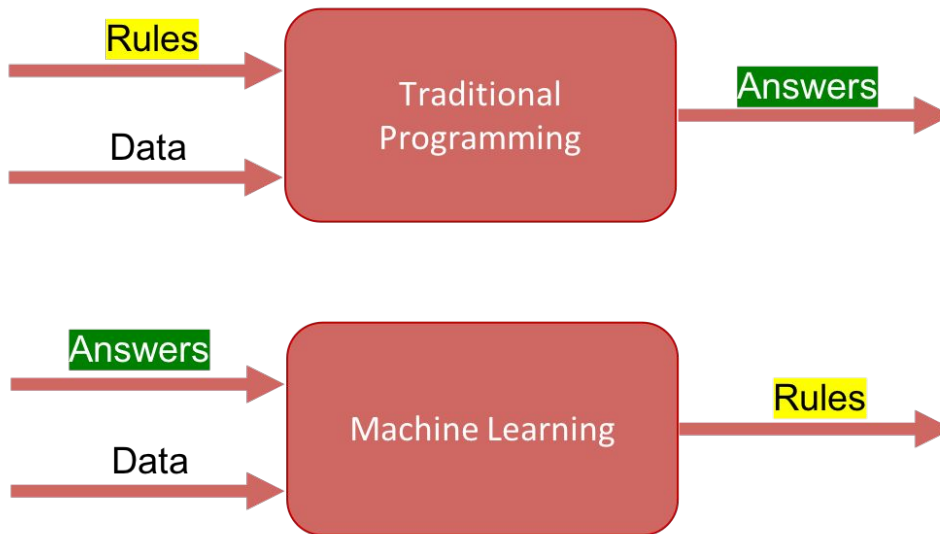
ML: Classificação - Redes Neurais



ML: Classificação - Redes Neurais



ML: Classificação - Redes Neurais



ML: Classificação - Redes Neurais

```
nn.compile(optimizer='adam', loss='binary_crossentropy')
nn.fit(volei[['previsao', 'temperatura', 'umidade', 'vento']], volei['jogar'])
previsao = nn.predict(v_prev[['previsao', 'temperatura', 'umidade', 'vento']])
print(previsao)
```

previsao	temperatura	umidade	vento	-
2	1	1	2	[[0.01834392]
1	1	1	2	[0.03572804]
1	1	1	1	[0.08500457]
2	1	1	2	[0.01834392]
3	2	1	2	[0.00275669]
3	3	2	2	[0.0012719]]

ML: Classificação - Redes Neurais

- Se definirmos nesse exemplo um limiar de ativação=0.08, temos as seguintes previsões

```
limiar = 0.08
def ativacao(prb):
    if(prb>=limiar):
        return 1
    else:
        return 0

for prev in previsao:
    print(ativacao(prev))
```

previsao	temperatura	umidade	vento	PRV	
2	1	1	2	0	[0.01834392]
1	1	1	2	0	[0.03572804]
1	1	1	1	1	[0.08500457]
2	1	1	2	1	[0.01834392]
3	2	1	2	0	[0.00275669]
3	3	2	2	0	[0.0012719]]

Exercício: Rede Neural

- Crie sua rede neural para o conjunto de dados do histórico de crédito.
Atenção na hora de definir o limiar da sua função de ativação.
 - Compare os resultados com os algoritmos de classificação dos exercícios anteriores
 - Experimente sua rede neural com distintas configurações (mais ou menos camadas como mais ou menos perceptrons por camada).

treino: <https://raw.githubusercontent.com/lcbjrrr/data/main/RiscoCredito%20-%20okk.csv>

teste: <https://raw.githubusercontent.com/lcbjrrr/data/main/RiscoCredito%20-%20prever2.csv>

ATIVIDADE 4: Redes Neurais

- Escolha uma base de dados no <https://www.kaggle.com/datasets>, e se familiarize com sua base
- Procure realizar a previsão (inferência) de uma variável categórica através de uma Rede Neural. Se certifique de medir seus níveis de assertividade.
- Compare os resultados com os algoritmos de classificação dos exercícios anteriores (KNN e/ou Árvore de Decisão)
- Não esqueça de junto com seus códigos realizar suas análises/conclusões (use o botão de +Texto).



MAD MEN

Intuición · Creatividad · Appeal



MATH MEN

Analytics · Resultados · Lógica

