Santander Coders 2023-2

Projeto de conclusão do módulo:
Lógica de Programação II - Ada Tech.

Implementação do algoritmo de machine learning **K-Nearest Neighbors** em Python

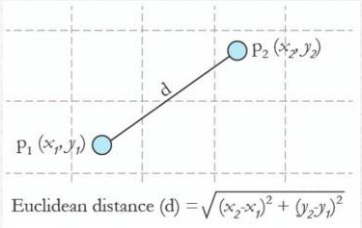
Participantes: Anderson, André, Artur, João, Juliana.

Algoritmo Supervisionado **KNN**

Modelo de Machine Learning

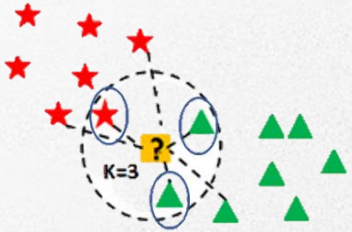
1 Escolher o valor de K.

2 Escolher um método de distância.



3 Calcular as distâncias entre o ponto de interesse e os pontos conhecidos.

4 Descobrir a classe majoritária entre os K-vizinhos mais próximos.
Atribuir o valor da classe ao ponto de interesse.



https://www.kaggle.com/datasets/adityakadiwal/water-potability/data

Water Quality

Drinking water potability

Data Card | Code (490) | Discussion (22)

About Dataset

Context

Access to safe drinking-water is essential to health, a basic human right and a component of effective policy for health protection. This is important as a health and development issue at a national, regional and local level. In some regions, it has been shown that investments in water supply and sanitation can yield a net economic benefit, since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the interventions.

Usability ⓘ
10.00

License
CC0: Public Domain

Expected update frequency
Annually

Tags

Earth and Nature
Beginner
Energy
Public Health

Base de Dados – **Qualidade da água**

pH value – pH da água (0 a 14);

Hardness – Capacidade da água de precipitar sabão em mg/L;

Solids – Total de sólidos dissolvidos em ppm;

Chloramines – Quantidade de Cloraminas em ppm;

Sulfate – Quantidade de Sulfatos dissolvidos em mg/L;

Conductivity – Condutividade elétrica da água em µS/cm;

Organic_carbon – Quantidade de carbono orgânico em ppm;

Trihalomethanes – Quantidade de Trihalometanos em µg/L;

Turbidity – Medida da propriedade de emissão de luz da água em NTU;

Potability – Indica se a água é segura para consumo humano. Potável (1) e Não potável (0).

# ph pH of water	# Hardness Capacity of water to precipitate soap in mg/L	# Solids Total dissolved solids in ppm	# Chloramines Amount of Chloramines in ppm	# Sulfate Amount of Sulfates dissolved in mg/L
0 – 14	47.4 – 323	321 – 61.2k	0.35 – 13.1	129 – 481
	204.8904554713363	20791.318980747026	7.300211873184757	368.51644134980336
3.71608007538699	129.42292051494425	18630.057857970347	6.635245883862	
8.099124189298397	224.23625939355776	19909.541732292393	9.275883602694089	
8.316765884214679	214.37339408562252	22018.417440775294	8.05933237743854	356.88613564305666
9.092223456290965	181.10158923612525	17978.98633892625	6.546599974207941	310.13737524284444
5.584086638456089	188.3133237696164	28748.68773904612	7.544868778877965	326.6783629116736
10.223862164528773	248.07173527013992	28749.716543528233	7.513408465831302 5	393.66339551509645

# Conductivity Electrical conductivity of water in µS/cm	# Organic_carbon Amount of organic carbon in ppm	# Trihalomethanes Amount of Trihalomethanes in µg/L	# Turbidity Measure of light emiting property of water in NTU (Nephelometric Turbidity Units)	# Potability Indicates if water is safe for human consumption
181 – 753	2.2 – 28.3	0.74 – 124	1.45 – 6.74	0 – 1
564.3086541722439	10.3797830788847	86.9909704615088	2.9631353806316407	0
592.8853591348523	15.180013116357259	56.32907628451764	4.500656274942408	0
418.6062130644815	16.868636929550973	66.42009251176368	3.0559337496641685	0
363.2665161642437	18.436524495493302	100.34167436508008	4.628770536837084	0
398.41081338184466	11.558279443446395	31.997992727424737	4.075075425430034	0
280.4679159334877	8.399734648152758	54.917861841994466	2.5597082275565217	0

fonte: https://www.kaggle.com/datasets/adityakadiwal/water-potability/data

Funções **Python**

Leitura

```
create_csv()
read_csv()
convert_value()

data_info()
data_describe()
data_row()
data_col()
data_value_counts()
data_serie()
data_view()
```

Tratamento

```
dict_to_list()
list_to_dict()

data_drop_attribute()
data_drop_na()
data_fill_na()
data_mean()
data_median()
```

Modelo

```
ml_normalize()
ml_standardize()
ml_train_test_split()
ml_grid_search()
ml_fit()
ml_predict()
ml_score_knn()
metric_classification_report()
metric_accuracy()
euclidean_distance()
manhattan_distance()
minkowski_distance()
```

Total: 29 funções

Funções **Python**

Leitura

create_csv()	recebe um dicionário ou uma lista de listas e escreve em um arquivo CSV (nome do arquivo e delimitador podem ser especificados).
read_csv()	recebe o nome de um arquivo CSV, um delimitador e um tipo opcional; lê o arquivo CSV e retorna os dados no tipo especificado.
convert_value()	recebe um valor e o converte para um tipo específico com base no próprio valor.
data_info()	analisa e retorna informações sobre um conjunto de dados fornecido como um dicionário (número de colunas, o número de linhas, os nomes das colunas e informações adicionais sobre cada coluna, como o tipo de dados e a quantidade de valores nulos)
data_describe()	recebe um dicionário como entrada e retorna uma string que descreve os dados no dicionário (tipo, quantidade, máximo, mínimo, soma, intervalo, média e mediana de cada coluna no dicionário)
data_row()	recebe um dicionário e um índice de linha opcional e retorna uma lista de valores da linha especificada.
data_col()	recebe um dicionário, um nome de coluna (str) e um tipo opcional e retorna os dados de uma coluna especificada no formato desejado (lista, tupla ou dicionário)
data_value_counts()	recebe um dicionário ou uma lista e retorna um dicionário que conta a ocorrência de cada valor único na coluna ou lista especificada.
data_serie()	recebe um dicionário e um índice de linha opcional e retorna um dicionário que representa uma linha de dados do dicionário de entrada.
data_view()	recebe um dicionário e um limite opcional e imprime uma visão formatada dos dados até o limite especificado.

Funções **Python**

Tratamento

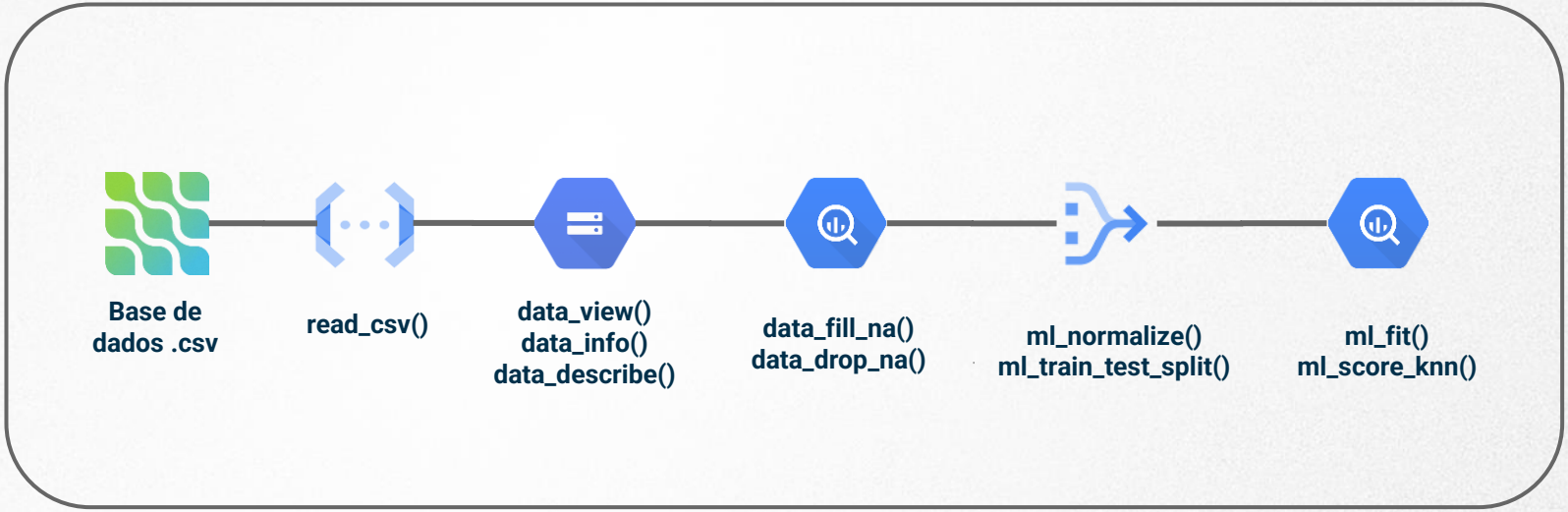
dict_to_list()	recebe um dicionário ou uma lista e converte em uma lista de listas.
list_to_dict()	recebe uma lista ou um dicionário como entrada e converte em um dicionário.
data_drop_attribute()	recebe um dicionário e uma string e remove o par chave-valor do dicionário cuja chave corresponde à string fornecida.
data_drop_na()	recebe um dicionário e remove quaisquer linhas que contenham valores ausentes ou nulos.
data_fill_na()	recebe um dicionário, um nome de coluna e um valor; substitui quaisquer valores ausentes ou nulos na coluna especificada do dicionário pelo valor fornecido.
data_mean()	recebe uma lista e retorna a média dos valores na lista.
data_median()	recebe uma lista e retorna a mediana dos valores na lista.

Funções **Python**

Modelo

ml_normalize()	normaliza os valores de um dicionário, ajustando-os para uma escala de 0 a 1, e retorna um dicionário com os valores normalizados.
ms_standardize()	padroniza os valores de um dicionário, subtraindo a média e dividindo pelo desvio padrão de cada coluna, e retorna um dicionário com os valores padronizados.
ml_train_test_split()	divide os dados em conjuntos de treinamento e teste com base na taxa de teste fornecida. Retorna os conjuntos de treinamento e teste para os recursos e alvos.
ml_grid_search()	realiza uma busca em grade para encontrar os melhores parâmetros para o algoritmo KNN. Ela recebe o conjunto de treinamento, a coluna alvo e uma lista de parâmetros para testar como entrada e retorna os melhores parâmetros encontrados durante a busca em grade.
ml_fit()	ajusta um modelo KNN aos dados usando um número específico de vizinhos e uma métrica de distância. Retorna um dicionário representando o ajuste do modelo.
ml_predict()	usa o modelo KNN treinado para fazer previsões em um novo conjunto de dados. Ela recebe o modelo e o novo conjunto de dados como parâmetros e retorna os valores previstos. é útil quando o modelo já foi treinado e deseja-se usá-lo para prever resultados em novos dados em que o modelo nunca foi usado.
ml_socre_knn()	calcula a precisão do modelo KNN. Ela recebe o modelo, o conjunto de teste e a coluna alvo como parâmetros e retorna a precisão do modelo no conjunto de teste. A precisão é uma métrica que indica o quão bem o modelo foi capaz de fazer previsões corretas no conjunto de teste.
metric_classification_report()	gera um relatório de classificação para o modelo de ajuste fornecido. O relatório inclui métricas como precisão, recall e pontuação F1 para cada classe.
metric_accuracy()	calcula a precisão do modelo de ajuste fornecido. A precisão é a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões.
euclidean_distance()	recebe dois pontos e calcula a distância euclidiana entre eles.
manhattan_distance()	recebe dois pontos e calcula a distância de Manhattan entre eles.
minkowski_distance()	recebe dois pontos e um parâmetro de potência e calcula a distância de Minkowski entre eles.

Metodologia do **Projeto**



* Ícones com fins ilustrativos.

Resultados

K=10

```
[94]: knn = ml_fit(X_train, y_train, 10)
      print('Score:', ml_score_knn(knn)*100, "%")
      for n, val in knn.items():
          print(n, val)
```

```
Score: 62.27 %
n_vizinhos 10
erros 927
acertos 1530
tx_erros 0.5659340659340664
tx_acertos 0.4340659340659355
rodadas 6036849
num_rows 2457
```

K=20

```
[93]: knn = ml_fit(X_train, y_train, 20)
      print('Score:', ml_score_knn(knn)*100, "%")
      for n, val in knn.items():
          print(n, val)
```

```
Score: 63.0 %
n_vizinhos 20
erros 909
acertos 1548
tx_erros 0.57956857956858
tx_acertos 0.420431420431421
rodadas 6036849
num_rows 2457
```

K=30

```
[89]: knn = ml_fit(X_train, y_train, 30)
      print('Score:', ml_score_knn(knn)*100, "%")
      for n, val in knn.items():
          print(n, val)
```

```
Score: 64.96 %
n_vizinhos 30
erros 861
acertos 1596
tx_erros 0.5785782119115431
tx_acertos 0.42142178808845615
rodadas 6036849
num_rows 2457
```

K=40

```
[90]: knn = ml_fit(X_train, y_train, 40)
      print('Score:', ml_score_knn(knn)*100, "%")
      for n, val in knn.items():
          print(n, val)
```

```
Score: 65.08 %
n_vizinhos 40
erros 858
acertos 1599
tx_erros 0.5829873829873821
tx_acertos 0.4170126170126162
rodadas 6036849
num_rows 2457
```

K=50

```
[91]: knn = ml_fit(X_train, y_train, 50)
      print('Score:', ml_score_knn(knn)*100, "%")
      for n, val in knn.items():
          print(n, val)
```

```
Score: 65.2 %
n_vizinhos 50
erros 855
acertos 1602
tx_erros 0.5846560846560873
tx_acertos 0.41534391534391385
rodadas 6036849
num_rows 2457
```

K=60

```
[92]: knn = ml_fit(X_train, y_train, 70)
      print('Score:', ml_score_knn(knn)*100, "%")
      for n, val in knn.items():
          print(n, val)
```

```
Score: 64.14 %
n_vizinhos 70
erros 881
acertos 1576
tx_erros 0.5934472934472942
tx_acertos 0.4065527065527063
rodadas 6036849
num_rows 2457
```

AGRADECEMOS PELA ATENÇÃO.

REFERÊNCIAS:

Artificial Intelligence. (2024, 3 de Janeiro). *What are the most effective distance metrics for optimizing k-nearest neighbors algorithms?* Linkedin.com; www.linkedin.com. https://www.linkedin.com/advice/3/what-most-effective-distance-metrics-optimizing-xndwc.

Bruce, P., & Bruce, A. (2019). *Estatística prática para cientistas de dados: 50 conceitos essenciais*. Alta Books.

de Maquina, A. [@aprendizagemdemaquina9452]. (2021, March 4). *O que é o KNN e como implementar do zero*. Youtube. Acesso em 20 Jan 2024 de https://www.youtube.com/watch?v=E7R6O4Aqw-M.

Comunidade Ada. (n.d.). Ada.Tech. Acesso em 18 Jan 2024 de https://lms.ada.tech/student.

Fávero, L. P., Lopes E, B., & Prado, P. (2017). *Manual de análise de dados: estatística e modelagem multivariada com Excel, SPSS e Stata*. Elsevier.

Kadiwal, A. (2021). Water Quality [Data set]. In *Drinking Water Potability*. Acesso em 12 Jan 2024 de https://www.kaggle.com/datasets/adityakadiwal/water-potability/data.

Kaggle: Your machine learning and data science community. (n.d.). Kaggle.com. Acesso em 20 Jan 2024 de https://www.kaggle.com.

REFERÊNCIAS:

Kunumi. (2020, 10 Junho). *Métricas de Avaliação em Machine Learning: Classificação*. Kunumi Blog. Acesso em 17 Jan 2024 de https://medium.com/kunumi/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-em-machine-learning-classifica%C3%A7%C3%A3o-49340dcdb198.

Matos, G. (2023, December 5). *K-Nearest Neighbors(KNN): Entendendo o seu funcionamento e o construindo do zero*. Share! Por Ateliê de Software. Acesso em 18 Jan 2024 de https://share.atelie.software/k-nearest-neighbors-knn-entendo-o-seu-funcionamento-e-o-construindo-do-zero-a21b022acd6f.

PEP 257 – docstring conventions. (n.d.). Python.org. Acesso em 23 Jan 2024 de vhttps://peps.python.org/pep-0257.

Srivastava, T. (2018, 25 março). *A complete guide to K-Nearest Neighbors (updated 2024)*. Analytics Vidhya. Acesso em 24 Jan 2024 de https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering.

Tavares, C. (2019, 26 Março). *KNN sem caixa preta*. Medium. Acesso em 22 Jan 2024 de https://medium.com/@caroli.agro/aplicando-knn-em-iris-dataset-d594b79652d1.

Yu, C., Ooi, B. C., Tan, K., & Jagadish, H. V. (2001). Indexing the Distance: An Efficient Method to KNN Processing. In *Very Large Data Bases Conference*.