Universidade de Trás-os-Montes e Alto Douro

Engenharia Informática
Introdução à Ciência de Dados

Trabalho Experimentar 1

Enunciado D2

Paulo Martins

Alunos:

Bernardo de Oliveira Almeida - al78403
David Gomes Fidalgo – al79881
Tiago Fernando Valente Sousa Carvalho – al78800
Vasco da Silva Macedo – al78798

Conteúdo

1 Introdução

A água doce é um recurso natural essencial e a sua gestão sustentável tornou-se crítica face ao crescimento populacional e às alterações climáticas. A análise de dados de consumo de água permite compreender padrões de utilização por país e setor (agrícola, industrial e doméstico), identificando tendências e possíveis ineficiências. Neste relatório técnico apresentam-se os resultados do Trabalho Experimental 1 da unidade curricular Introdução à Ciência dos Dados, que consistiu em explorar e modelar um conjunto de dados globais de consumo de água no período de 2000 a 2024. Seguindo o protocolo fornecido, procederam-se a diversas etapas de análise: (1) carregamento e preparação do conjunto de dados original, (2) visualização da evolução temporal do consumo total de água para países selecionados, (3) análise da distribuição setorial do uso da água num caso específico (Espanha, 2020), (4) implementação de uma função para detetar o ano de menor utilização de água na agricultura por país, (5) estudo da relação entre uso industrial da água e a taxa de esgotamento de aquíferos através de um gráfico de dispersão com regressão linear, e (6) construção de um modelo simples de machine learning para prever o consumo de água per capita com base em variáveis selecionadas. Cada etapa foi realizada em linguagem Python, recorrendo a bibliotecas comuns de ciência de dados (pandas, matplotlib, numpy, scikit-learn), e os resultados obtidos são discutidos de forma objetiva. O relatório está organizado em seções, começando por um enquadramento teórico sobre os conceitos de análise de dados, visualização, regressão linear e aprendizagem automática básica aplicados. Em seguida, descreve-se a implementação prática de cada tarefa, incluindo figuras ilustrativas (gráficos) e valores estatísticos relevantes extraídos diretamente do código. Por fim, é apresentado um balanço dos resultados obtidos e dos métodos utilizados, bem como as conclusões principais deste estudo.

2 Enquadramento Teórico

2.1 Análise de Dados e Visualização

No início de qualquer projeto de ciência de dados é fundamental efetuar uma análise exploratória de dados (EDA). Esta etapa envolve o carregamento do conjunto de dados bruto para uma estrutura apropriada (como um DataFrame em pandas) e a inspeção inicial do seu conteúdo. Procede-se à verificação do tamanho (número de linhas/observações e colunas/variáveis) e dos tipos de dados presentes, bem como à identificação de valores em falta (nulos) ou valores anómalos. Estatísticas descritivas básicas (mínimos, máximos, médias, quartis) são calculadas para compreender a distribuição de cada variável numérica. Esta exploração fornece uma compreensão inicial dos dados e pode guiar transformações ou filtragens necessárias antes da análise aprofundada. A visualização de dados é outra componente teórica importante, pois permite identificar padrões e relações de forma intuitiva. Gráficos de linhas são frequentemente utilizados para mostrar a evolução temporal de uma variável; por exemplo, a tendência do consumo total de água ao longo dos anos pode ser facilmente interpretada num gráfico temporal, comparando diferentes países através de linhas distintas. Gráficos de setores (pie charts) permitem analisar a composição percentual de um todo – no contexto deste trabalho, visualizar a percentagem de água consumida por setor (agrícola, industrial, doméstico) num determinado país e ano evidencia qual o setor predominante no consumo hídrico. Já os gráficos de dispersão (scatter plots) são indicados para avaliar a relação entre duas variáveis quantitativas: cada ponto representa uma observação (por exemplo, um país num ano), plotando um atributo no eixo X vs. outro no eixo Y. A adição de elementos como linhas de tendência ou regressão ajuda a quantificar e comunicar a eventual correlação entre as variáveis.

2.2 Regressão Linear

A regressão linear é um método estatístico fundamental para modelar a relação entre variáveis. No caso simples de regressão linear simples (uma variável explicativa), assume-se que a variável dependente Y pode ser expressa aproximadamente como uma combinação linear de uma variável independente X e um termo constante:

$$Y \approx \beta_0 + \beta_1 X,$$

onde os coeficientes β_0 (interceção) e β_1 (declive) são estimados de modo a minimizar o erro entre os valores observados e os valores previstos pelo modelo (tipicamente através do método dos mínimos quadrados). O declive β_1 indica a variação esperada em Y para cada aumento unitário em X: um valor positivo sugere uma relação direta (à medida que X aumenta, Y tende a aumentar), enquanto um valor negativo indica uma relação inversa.

Para avaliar a qualidade do ajuste de um modelo linear utiliza-se frequentemente

o coeficiente de determinação (R^2). Este coeficiente varia de 0 a 1 e representa a proporção da variância de Y que é explicada pelo modelo linear. Por exemplo, $R^2 = 0,8$ indica que 80% da variabilidade observada em Y é explicada pela relação linear com X (os restantes 20% permanecem sem explicação pelo modelo, podendo ser devidos a outros fatores ou ruído). Um R^2 próximo de 1 denota um excelente ajuste (pontos muito alinhados em torno da reta), enquanto valores baixos de R^2 indicam que a relação linear é fraca ou inexistente. Importa notar que correlação não implica causalidade: mesmo que X e Y exibam uma forte correlação linear, isso não significa necessariamente que variações em X causem variações em Y – pode haver outros fatores em jogo.

Outra métrica fundamental é o erro quadrático médio (MSE, *mean squared error*), que quantifica o erro médio ao quadrado das previsões do modelo em relação aos valores reais. O MSE é dado por:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

onde y_i é o valor real e $\hat{y}_i$ o valor previsto. Quanto menor for o MSE, melhor o modelo consegue ajustar os dados (erro médio mais pequeno). Muitas vezes utiliza-se a raiz quadrada do MSE (RMSE) para obter um erro médio na mesma unidade do problema original, o que facilita a interpretação. Em conjunto, o R^2 e o MSE permitem avaliar o compromisso entre ajuste do modelo e erro de previsão.

2.3 Machine Learning

No âmbito da aprendizagem automática (*machine learning*) supervisionada, a tarefa de previsão de uma variável numérica é frequentemente abordada com modelos de regressão. Quando existem múltiplas variáveis explicativas (*features*), pode-se recorrer à regressão linear múltipla, uma extensão do modelo linear para incluir diversos preditores. O modelo assume a forma:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p,$$

em que $X_1, X_2, \ldots, X_p$ são as variáveis de entrada (por exemplo, país, uso agrícola, precipitação, etc.) e $\beta_1, \ldots, \beta_p$ os respetivos coeficientes que medem a influência linear de cada variável em Y. A aprendizagem do modelo consiste em ajustar todos os coeficientes β para melhor prever Y a partir de X, minimizando o erro (usando critérios como o MSE mencionado).

Uma prática crucial em projetos de *machine learning* é dividir os dados em conjuntos de treino e teste. O modelo é ajustado usando o conjunto de treino (por exemplo, 80% dos dados) e posteriormente avaliado no conjunto de teste (os 20% restantes, não vistos pelo modelo durante o treino). Esta separação permite avaliar a capacidade de generalização do modelo a dados novos, prevenindo *sobreajuste* (*overfitting*) — situação em que o modelo se ajusta demasiado bem aos dados de treino (capturando

ruído específico), mas falha em generalizar para outros dados. No caso em estudo, a divisão estratificada por país ou aleatória dos anos assegura que o modelo de previsão do consumo per capita é validado em diferentes cenários.

Ao lidar com variáveis categóricas (como o nome do país), é necessário convertê-las em formato numérico para as usar no modelo. Uma técnica comum é o *one-hot encoding*, que cria colunas binárias (*dummies*) para cada categoria. Por exemplo, a variável categórica `Country` com valores {`Italy`, `Japan`, `Spain`, `UK`, `USA`} pode ser transformada em cinco colunas indicadoras (p. ex., `Country_Italy`, `Country_Japan`, etc.), onde o valor é 1 se o registo corresponde a esse país e 0 caso contrário. Para evitar redundâncias, costuma-se eliminar uma das colunas (*drop-first*), já que a última categoria é implicitamente representada pela ausência de todas as outras (isso evita problemas de multicolinearidade no modelo linear). Com as variáveis devidamente codificadas e normalizadas se necessário, procede-se ao treino do modelo de regressão nos dados de treino e à avaliação no conjunto de teste utilizando métricas como o MSE e o R^2.

No contexto deste trabalho, a aplicação de *machine learning* é relativamente básica — um modelo linear múltiplo — porém, o foco está em documentar bem o processo (escolha das variáveis, preparação dos dados, avaliação) e interpretar os resultados. Mesmo modelos lineares simples podem oferecer bons desempenhos e *insights* se as variáveis escolhidas forem relevantes, como se verá adiante na previsão do consumo de água per capita.

3 Implementação Prática das Tarefas

3.1 Carregamento e Filtragem de Dados

Para a realização das tarefas, foi utilizado o dataset "Global Water Consumption Dataset (2000–2024)", disponível na plataforma Kaggle, que compila indicadores de consumo de água para vários países ao longo do período de 25 anos (2000 a 2024). O conjunto de dados original foi carregado a partir de um ficheiro CSV para um `DataFrame pandas`, preservando as colunas originais com os campos relevantes: país (`Country`), ano (`Year`), consumo total de água (`Total Water Consumption (Billion Cubic Meters)`), percentagem de uso agrícola, industrial e doméstico da água (`Agricultural`, `Industrial`, `Household Water Use (%)`), taxa de esgotamento de águas subterrâneas (`Groundwater Depletion Rate (%)`), consumo de água per capita (`Per Capita Water Use (Liters per Day)`) e impacto da precipitação (`Rainfall Impact (Annual Precipitation in mm)`).

Após o carregamento, verificou-se que o dataset possui dimensões de aproximadamente 500 linhas × 9 colunas, correspondendo a cerca de 20 países com registos anuais durante 25 anos. Numa inspeção inicial, constatou-se que não existem valores nulos significativos nas colunas de interesse — todos os países apresentam valores para as principais variáveis ao longo do período. As estatísticas descritivas confirmaram a diversidade nos padrões de consumo: por exemplo, o consumo total de água por país/ano varia entre valores muito baixos (na ordem de poucas dezenas de bilhões de m^3) até valores próximos de mil bilhões de m^3, refletindo diferenças de escala populacional e de atividades económicas entre países.

Também se observou que a percentagem de uso agrícola da água tende a ser elevada (muitas vezes acima de 50%) em vários países, enquanto o uso doméstico raramente ultrapassa 30% do total, embora haja variações conforme o nível de desenvolvimento do país. Em seguida, procedeu-se à filtragem dos dados conforme requerido: foi criado um novo `DataFrame` contendo apenas os registos dos países `Italy, Japan, Spain, UK` e `USA`, que são o foco de algumas análises específicas. Estes cinco países representam um conjunto diversificado em termos de localização geográfica e perfil de consumo hídrico (Europa do Sul, Ásia oriental, Europa ocidental e América do Norte).

O `DataFrame` filtrado resultante contém apenas as linhas correspondentes a esses países (5 países × 25 anos = 125 linhas) e foi gravado num novo ficheiro CSV para referência (`filtered_global_water_consumption.csv`). Esta filtragem facilita a visualização comparativa e a análise focada nestes países, sem a interferência de outros países no gráfico temporal, por exemplo. Antes de prosseguir, a coluna do ano foi convertida para formato numérico (inteiro) para assegurar o correto ordenamento nos eixos temporais dos gráficos.

3.2 Visualização da Evolução do Consumo Total de Água

Com os dados filtrados para Itália, Japão, Espanha, Reino Unido e Estados Unidos, construiu-se um gráfico de linhas que representa a evolução anual do consumo total de água em cada um destes países, de 2000 até 2024. Na Figura 1, cada país é representado por uma linha distinta com marcadores anuais, permitindo a comparação direta das tendências temporais.

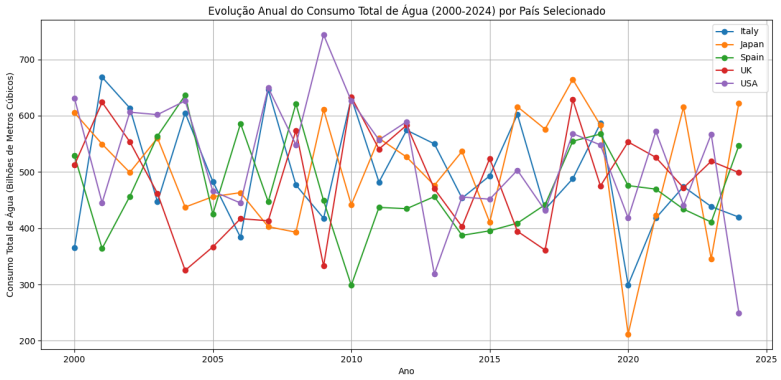

Figura 1: Evolução Anual do Consumo Total de Água (2000–2024)

3.3 Distribuição do Uso de Água em Espanha em 2020

Para analisar a repartição do consumo de água por setor, foi gerado um gráfico circular (pie chart) relativo às percentagens de uso agrícola, industrial e doméstico da água no ano de 2020, para o caso específico de Espanha. Este tipo de visualização ilustra de forma imediata qual o setor que mais contribui para o consumo total de água nesse país e ano, em proporção.

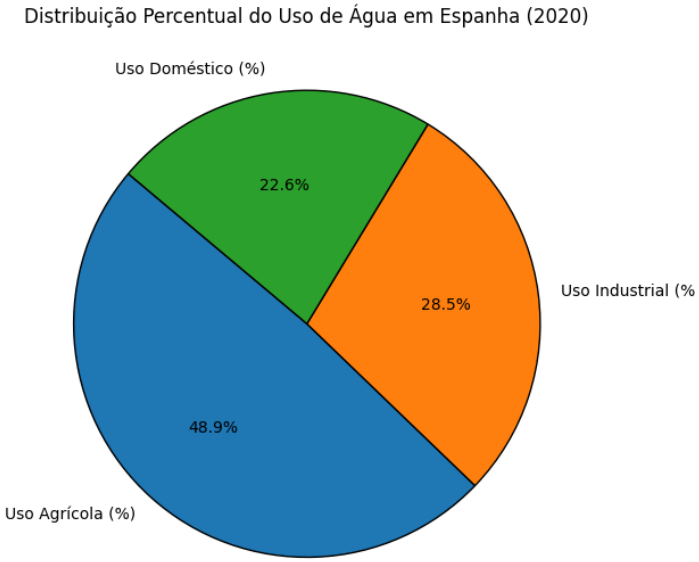

Figura 2: Gráfico circular da distribuição percentual do uso da água em Espanha no ano de 2020

3.4 Menor percentagem de uso agrícola por país

Outra tarefa realizada foi a implementação de uma função genérica para identificar, para um dado país, em que ano ocorreu o menor valor da percentagem de uso de água na agricultura, bem como qual foi esse valor mínimo. Em termos práticos, a função percorre todos os registos do país especificado e procura a mínima percentagem registada na coluna do uso de água na Agricultura (%), retornando o ano correspondente e o valor.

```
--- Tarefa 4: Teste da Função ---
Digite um país:UK
Para o país 'UK', o menor valor da coluna do uso da água na agricultura foi (39.74%) ocorreu no ano 2004.
```

Figura 3: Menor percentagem do uso agrícola em UK no ano de 2004

3.5 Relação entre Uso Industrial e Esgotamento de Águas Subterrâneas

3.5.1 Distribuição dos Dados

No gráfico são representados pontos correspondentes a diferentes unidades (países ou regiões), onde o eixo horizontal indica a percentagem de utilização industrial de água

e o eixo vertical, a percentagem da taxa de esgotamento das águas subterrâneas. A dispersão dos pontos sugere uma variabilidade entre as observações, indicando que a relação entre as variáveis não é perfeitamente linear.

3.5.2 Regressão Linear

O modelo de regressão linear obteve a seguinte equação:

$$y = -0.00736x + 2.77774$$

O declive negativo (-0.00736) implica que, para cada incremento de 1% na utilização industrial de água, a taxa de esgotamento das águas subterrâneas diminui, em média, cerca de 0.00736 pontos percentuais. Este efeito, embora estatisticamente presente, é bastante pequeno, sugerindo que a utilização industrial explica apenas uma parte da variação na taxa de esgotamento.

3.5.3 Interpretação e Implicações

- **Relação Inversa:** A tendência inversa observada pode indicar que, em contextos com maior utilização industrial de água, as entidades responsáveis podem recorrer a práticas de gestão ou a fontes alternativas que contribuam para mitigar o esgotamento dos aquíferos.
- **Valor do Intercepto:** O valor do intercepto, aproximadamente 2.78, representa a taxa de esgotamento das águas subterrâneas prevista para um cenário com utilização industrial nula.

3.5.4 Conclusões

Em suma, o gráfico e o modelo de regressão evidenciam uma relação fraca, mas inversa, entre a utilização industrial de água e a taxa de esgotamento das águas subterrâneas. Estes resultados sugerem que outros fatores, para além do consumo industrial, podem influenciar de forma significativa o esgotamento dos aquíferos.

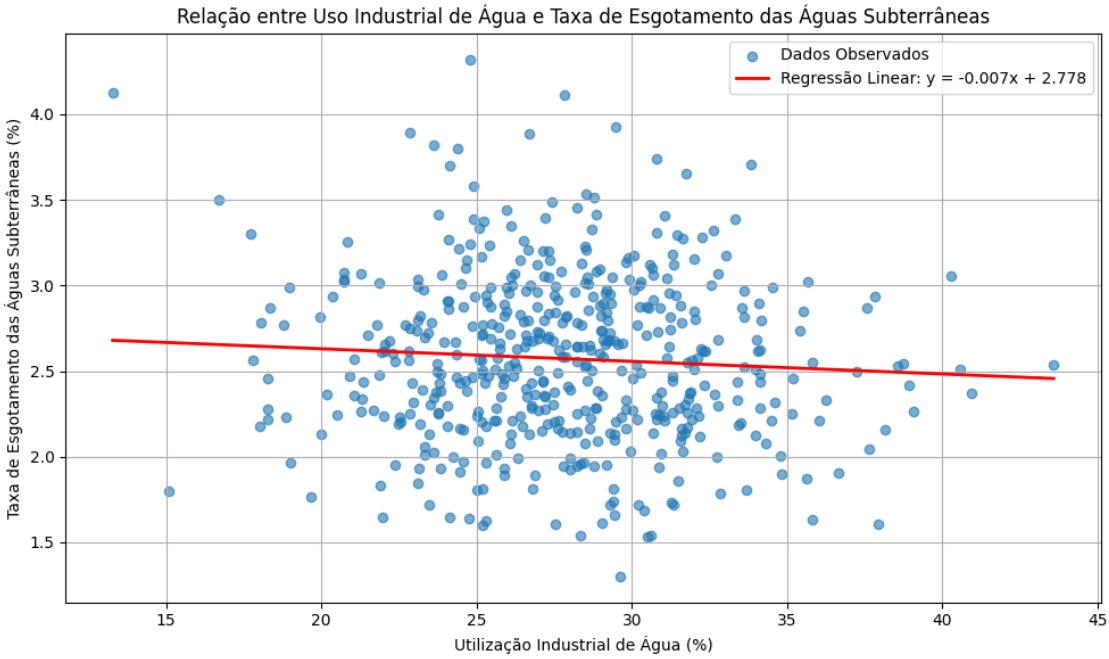

Figura 4: Gráfico de dispersão mostrando a relação entre a utilização industrial de água (%) e a taxa de esgotamento das águas subterrâneas (%) para vários países

3.6 Machine Learning para Previsão do Consumo Per Capita

Esta tarefa centrou-se na aplicação de técnicas de Machine Learning (ML) para o desenvolvimento de um modelo preditivo da variável "Per Capita Water Use (Liters per Day)". De acordo com as diretrizes do enunciado, a previsão deveria basear-se nas seguintes variáveis preditoras disponíveis no conjunto de dados: "Country"(País), "Agricultural Water Use (%)"(Percentagem de Utilização de Água na Agricultura) e "Rainfall Impact (Annual Precipitation in mm)"(Impacto da Precipitação Anual).

3.6.1 Preparação dos Dados

Inicialmente, foi realizada a seleção do subconjunto de colunas relevantes para a modelagem preditiva. Em seguida, procedeu-se ao tratamento de valores ausentes, removendo todas as linhas que continham pelo menos um valor em falta nas colunas selecionadas. Esta abordagem resultou num conjunto final de 500 amostras completas para a modelagem.

Um dos desafios críticos foi o tratamento da variável categórica "Country". Como os modelos de regressão linear requerem variáveis numéricas, foi aplicada a técnica de One-Hot Encoding por meio da função pd.get_dummies(), com a opção drop_first=True, a fim de mitigar problemas de multicolinearidade, eliminando uma das colunas binárias geradas para os países.

O conjunto de dados final foi posteriormente dividido em dois subconjuntos: um para treino e outro para teste. Foi adotada uma divisão de 80% dos dados (400 amostras) para treino e 20% (100 amostras) para teste, utilizando a função train_test_split da biblioteca scikit-learn, com random_state = 42 para garantir a reprodutibilidade da divisão.

3.6.2 Modelagem e Avaliação

Foram testados três modelos de regressão linear:

Regressão Linear Simples (LinearRegression): Utilizada como modelo base para estabelecer um referencial de desempenho[6][4].

Lasso (Lasso): Modelo regularizado que aplica uma penalização L1 aos coeficientes, promovendo seleção de variáveis e reduzindo overfitting. Foi configurado com alpha=1.0. A penalização L1 impõe um custo proporcional à soma absoluta dos coeficientes, forçando alguns deles a se tornarem exatamente zero, o que favorece a simplificação do modelo[5][2].

ElasticNet (ElasticNet): Modelo que combina penalizações L1 e L2, oferecendo um equilíbrio entre Lasso e Ridge Regression. A penalização L2 reduz a magnitude dos coeficientes sem anulá-los completamente, ajudando a evitar overfitting. Para este estudo, foi configurado com alpha=1.0 e l1_ratio=0.50, aplicando uma combinação equilibrada de ambas as penalizações[3][1].

Cada modelo foi treinado utilizando os dados de treino (X_train, y_train) e posteriormente utilizado para gerar previsões (predict) no conjunto de teste (X_test). O desempenho foi avaliado por meio das seguintes métricas:

Erro Quadrático Médio (Mean Squared Error - MSE): Mede o erro médio quadrático das previsões. Valores mais baixos indicam melhor desempenho.

Coeficiente de Determinação (R-squared - R^2): Indica a proporção da variabilidade da variável alvo explicada pelo modelo, variando de -Inf a 1.

Os resultados obtidos foram os seguintes:

Modelo	MSE	R^2
Regressão Linear	2079.86	-0.097
Lasso (alpha=1.0)	1859.50	0.019
ElasticNet (alpha=1.0, l1_ratio=0.50)	1862.02	0.018

3.6.3 Análise dos Resultados

Os resultados revelam um desempenho preditivo insatisfatório. O R^2 negativo (-0.097) da regressão linear sugere que o modelo performa pior do que uma simples

previsão baseada na média dos valores da amostra. Esse fenômeno ocorre quando a soma dos quadrados dos erros do modelo (SS_res) é maior do que a soma total dos quadrados (SS_tot), evidenciando um ajuste inadequado aos dados de teste. Os modelos Lasso e ElasticNet apresentaram um MSE menor e um R^2 ligeiramente positivo (0.019 e 0.018, respetivamente). No entanto, esses valores indicam que os modelos conseguem explicar apenas uma fração mínima da variabilidade do consumo de água per capita, o que é considerado extremamente baixo em termos práticos.

3.6.4 Considerações Finais e Perspectivas Futuras

Os resultados obtidos sugerem fortemente que as variáveis preditoras selecionadas ("Country", "Agricultural Water Use (%)", "Rainfall Impact") possuem baixo poder preditivo sobre a variável alvo ("Per Capita Water Use"). Tal limitação pode decorrer dos seguintes fatores:

• A percentagem de água usada na agricultura pode não se correlacionar diretamente com o consumo individual diário de água.

• A precipitação anual pode ser uma métrica demasiado agregada para refletir a disponibilidade efetiva de água para consumo pessoal.

• A variável "Country", mesmo após o One-Hot Encoding, pode ser demasiado genérica e introduzir esparsidade no modelo sem fornecer informação significativa.

• A relação entre as variáveis preditoras e o consumo per capita pode ser altamente não linear, reduzindo a eficácia dos modelos lineares utilizados.

Para superar essas limitações, algumas melhorias são recomendadas para futuras abordagens:

1. **Exploração de Novas Variáveis:** Incluir atributos como PIB per capita, densidade populacional, tipo de clima e infraestrutura hídrica pode aumentar a capacidade preditiva do modelo.

2. **Engenharia de Features:** Criar variáveis derivadas que capturem melhor a dinâmica entre as variáveis preditoras e a variável alvo.

3. **Utilização de Modelos Mais Sofisticados:** Testar algoritmos não lineares, como árvores de decisão, Random Forests e Gradient Boosting, pode melhorar a capacidade de captura de padrões mais complexos nos dados. Desta forma, a revisão das variáveis utilizadas e a adoção de métodos mais robustos são passos fundamentais para aprimorar a precisão da modelagem preditiva do consumo de água per capita.

4 Conclusão

Neste trabalho, realizámos uma análise do consumo global de água através de várias etapas importantes. Começámos por carregar, limpar e filtrar os dados com a biblioteca `pandas`, de forma a garantir que apenas os registos completos e relevantes fossem usados. Esta preparação foi essencial para que as análises fossem feitas com dados de boa qualidade.

Para visualizar os dados, usamos o `matplotlib` para criar gráficos que mostraram a evolução do consumo de água ao longo dos anos, a distribuição do uso de água por setor (agrícola, industrial e doméstico) e a relação entre a utilização industrial de água e o esgotamento dos aquíferos. Estas representações ajudaram-nos a perceber melhor as diferenças entre os países e as tendências ao longo do tempo.

No âmbito da modelação, aplicámos técnicas simples de Machine Learning com a biblioteca `scikit-learn`, nomeadamente a Regressão Linear, Lasso e ElasticNet, para tentar prever o consumo de água per capita. Os resultados indicaram que as variáveis selecionadas explicam apenas uma pequena parte da variabilidade do consumo, sugerindo que seria necessário incluir mais fatores ou usar modelos mais sofisticados para melhorar as previsões.

Em resumo, aprendemos que a preparação e limpeza dos dados são passos fundamentais e que a visualização dos resultados é essencial para identificar padrões. Apesar das limitações dos modelos lineares utilizados, este estudo permitiu-nos perceber melhor os desafios na previsão do consumo de água e mostrou a importância de explorar novas variáveis e técnicas para obter resultados mais precisos.

Referências

[1] Elasticnet — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html. Accessed: 2025-03-29.

[2] Lasso — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html. Accessed: 2025-03-29.

[3] Linear elastic net regression. https://www.ibm.com/docs/en/spss-statistics/saas?topic=features-linear-elastic-net-regression. Accessed: 2025-03-30.

[4] Linearregression — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. Accessed: 2025-03-29.

[5] What is lasso regression? — ibm. https://www.ibm.com/think/topics/lasso-regression. Accessed: 2025-03-28.

[6] What is linear regression? — ibm. https://www.ibm.com/think/topics/linear-regression. Accessed: 2025-03-27.

Anexos

A Código Python

```
"""

Bernardo Almeida, al78403
David Fidalgo, al79881
Tiago Valente al78800
Vasco Macedo al78798

Trabalho Experimental 1, "Global Water Consumption Dataset (2000-2024)":
  1. Carrega o ficheiro CSV completo, exibe informações iniciais e cria um novo DataFrame
     filtrado por países específicos (Italy, Japan, Spain, UK, USA), guardando-o num novo
  2. Plota a evolução do "Total Water Consumption" ao longo dos anos para os países selec
  3. Cria um gráfico circular (pie chart) mostrando a distribuição percentual dos usos de
     (agrícola, industrial e doméstico) para Spain no ano de 2020.
  4. Define uma função que retorna o ano e o valor da menor percentagem de "Agricultural
     para um país fornecido como entrada.
  5. Plota um gráfico de dispersão com uma linha de regressão linear entre "Industrial Wa
     "Groundwater Depletion Rate", utilizando o dataset completo.
  6. Utiliza técnicas de Machine Learning (Regressão Linear) para prever "Per Capita Wate
     com base nas variáveis "Country", "Agricultural Water Use" e "Rainfall Impact".

"""
import pandas as pd
import matplotlib.pyplot as matplt
import numpy as np
from sklearn.model_selection import train_test_split  # Para dividir dados em treino e te
from sklearn.metrics import mean_squared_error, r2_score  # Para avaliar o modelo de ML
from sklearn.linear_model import LinearRegression  # Para criar o modelo de Regressão Lin
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet

# --- Tarefa 1: Carregar, Filtrar e Guardar Dados ---
df = pd.read_csv("cleaned_global_water_consumption.csv")


# - df.shape: Retorna um tuple com o número de linhas e colunas.
# - df.columns.tolist(): Lista os nomes de todas as colunas.
# - df.head(): Mostra as primeiras 5 linhas do DataFrame por defeito.
print("--- Análise Exploratória Inicial do Dataset Completo ---")
print("Dimensões (linhas, colunas):", df.shape)
print("Nomes das colunas:", df.columns.tolist())
print("\nPrimeiras 5 linhas:")
print(df.head())
```

```
43
44  # Exibir estatísticas descritivas para as colunas numéricas (contagem, média, desvio padr
45  print("\nEstatísticas Descritivas:")
46  print(df.describe())
47
48  # Verificar a existência de valores nulos (ausentes) em cada coluna.
49  # isnull() retorna um DataFrame booleano (True onde há nulo) e sum() conta os True por co
50  print("\nContagem de Valores Nulos por Coluna:")
51  print(df.isnull().sum())
52  print("-----------------------------------------------------")
53
54  # Definir a lista de países de interesse.
55  paises = ['Italy', 'Japan', 'Spain', 'UK', 'USA']
56
57  # Filtrar o DataFrame original ('df') para manter apenas as linhas onde a coluna 'Country
58  # corresponde a um dos países na lista 'paises'.
59  df_filtrado = df[df['Country'].isin(paises)].copy() # Usar .copy() para evitar SettingWit
60
61  # Guardar o DataFrame filtrado num novo ficheiro CSV.
62  # index=False evita que o índice do DataFrame seja escrito como uma coluna no CSV.
63  df_filtrado.to_csv("filtered_global_water_consumption.csv", index=False)
64  print(f"--- Tarefa 1 Concluída ---")
65  print(f"Dados filtrados para os países {', '.join(paises)} guardados em 'filtered_global_
66  print("-----------------------------------------------------")
67
68
69  # --- Tarefa 2: Gráfico da Evolução do Consumo Total de Água ---
70
71  # É boa prática garantir que a coluna 'Year' é do tipo numérico para o plot.
72  # errors='coerce' transforma valores que não podem ser convertidos em NaN (Not a Number).
73  # A linha seguinte pode gerar um SettingWithCopyWarning se .copy() não foi usado acima.
74  # Como usamos .copy() na criação de df_filtrado, este aviso é evitado.
75  df_filtrado['Year'] = pd.to_numeric(df_filtrado['Year'], errors='coerce')
76
77  # Criar a figura e os eixos para o gráfico com um tamanho específico (largura 10, altura
78  matplt.figure(figsize=(12, 7))
79  # Iterar sobre cada país na lista 'paises'.
80  for pais in paises:
81      # Filtrar o DataFrame 'df_filtrado' para obter os dados apenas do país atual.
82      df_pais = df_filtrado[df_filtrado['Country'] == pais]
83      # Plotar a evolução: 'Year' no eixo X, 'Total Water Consumption' no eixo Y.
84      # marker='o' adiciona um marcador circular em cada ponto de dados.
85      # label=pais define o nome que aparecerá na legenda para esta linha.
86      matplt.plot(df_pais['Year'], df_pais['Total Water Consumption (Billion Cubic Meters)'
87
88  # Adicionar rótulos aos eixos X e Y e um título ao gráfico.
89  matplt.xlabel("Ano")
```

```
matplt.ylabel("Consumo Total de Água (Bilhões de Metros Cúbicos)")
matplt.title("Evolução Anual do Consumo Total de Água (2000-2024) por País Selecionado")
matplt.legend()
matplt.grid(True)

matplt.tight_layout()

print(f"--- Tarefa 2 Concluída ---")
print("A exibir o gráfico da evolução do consumo total de água...")
matplt.show()
print("-----------------------------------------------------")


# --- Tarefa 3: Gráfico Circular (Pie Chart) para Spain em 2020 ---

# Utiliza-se um bloco try-except para lidar com o caso de não existirem dados para Spain
try:
    dados_spain_2020 = df[(df['Country'] == 'Spain') & (df['Year'] == 2020)].iloc[0]
    labels = ['Uso Agrícola (%)', 'Uso Industrial (%)', 'Uso Doméstico (%)']
    valores = [
        dados_spain_2020['Agricultural Water Use (%)'],
        dados_spain_2020['Industrial Water Use (%)'],
        dados_spain_2020['Household Water Use (%)']
    ]

    matplt.figure(figsize=(7, 7))
    matplt.pie(valores, labels=labels, autopct='%1.1f%%', startangle=140, wedgeprops={'ed
    matplt.title("Distribuição Percentual do Uso de Água em Espanha (2020)")

    print(f"--- Tarefa 3 Concluída ---")
    print("A exibir o gráfico circular da distribuição do uso de água em Espanha (2020)..
    matplt.show()
except IndexError:
    print(f"--- Tarefa 3 Falhou ---")
    print("Não foram encontrados dados para 'Spain' no ano 2020 no dataset.")
print("-----------------------------------------------------")


# --- Tarefa 4: Função para Encontrar o Menor Uso Agrícola por País ---

def menor_uso_agricola(df_completo, nome_pais):
    """
    Encontra e retorna o ano e o valor da menor percentagem de 'Agricultural Water Use'
    para um país específico no DataFrame fornecido.

    Parâmetros:
      df_completo (pd.DataFrame): O DataFrame completo contendo os dados.
```

```
        nome_pais (str): O nome do país a ser pesquisado (sensível a maiúsculas/minúsculas)

    Retorna:
        tuple: (ano_min, valor_min) - O ano e o valor mínimo encontrado.
               Retorna (None, None) se o país não for encontrado no DataFrame.
    """

    df_country = df_completo[df_completo['Country'] == nome_pais]
    if df_country.empty:
        print(f"Aviso: Não foram encontrados dados para o país '{nome_pais}'.")
        return None, None

    idx_min = df_country['Agricultural Water Use (%)'].idxmin()

    linha_min = df_country.loc[idx_min]

    ano_min = linha_min['Year']
    valor_min = linha_min['Agricultural Water Use (%)']

    return int(ano_min), valor_min

print(f"--- Tarefa 4: Teste da Função ---")
pais_selecionado = input("Digite um país:")

ano, valor = menor_uso_agricola(df, pais_selecionado)

if ano is not None:
    print(f"Para o país '{pais_selecionado}', o menor valor da coluna do uso da água na a
else:
    print(f"País selecionado não encontrado")
print("-----------------------------------------------------")


# --- Tarefa 5: Gráfico de Dispersão e Regressão Linear ---

# Selecionar as colunas relevantes ('Industrial Water Use (%)', 'Groundwater Depletion Ra
# .dropna() remove as linhas onde qualquer um destes valores seja nulo, para evitar erros
df_scatter = df[['Industrial Water Use (%)', 'Groundwater Depletion Rate (%)']].dropna()

# Criar a figura e os eixos para o gráfico de dispersão.
matplt.figure(figsize=(10, 6))

# Criar o gráfico de dispersão:
# - Eixo X: 'Industrial Water Use (%)'.
# - Eixo Y: 'Groundwater Depletion Rate (%)'.
# - alpha=0.6: Define a transparência dos pontos (útil se houver sobreposição).
# - label='Dados': Rótulo para a legenda.
```

```
matplt.scatter(df_scatter['Industrial Water Use (%)'], df_scatter['Groundwater Depletion

# --- Cálculo e Plot da Regressão Linear ---
# Extrair os valores das colunas como arrays NumPy para a função polyfit.
x = df_scatter['Industrial Water Use (%)'].values
y = df_scatter['Groundwater Depletion Rate (%)'].values

# Calcular os coeficientes da regressão linear (polinómio de grau 1).
# coef[0] será o declive (slope) e coef[1] será a intercepção (intercept) da linha y = mx
coef = np.polyfit(x, y, 1)

# Criar um objeto de função polinomial a partir dos coeficientes calculados.
# Isto permite calcular facilmente os valores y da linha de regressão para quaisquer valo
linha_regressao = np.poly1d(coef)

# Gerar valores de x uniformemente espaçados entre o mínimo e o máximo de 'Industrial Wat
# Estes valores serão usados para desenhar a linha de regressão de forma suave.
x_vals = np.linspace(x.min(), x.max(), 100)

# Plotar a linha de regressão usando os x_vals gerados e a função linha_regressao(x_vals)
# color='red': Define a cor da linha.
# label=...: Cria um rótulo para a legenda que inclui a equação da linha formatada.
matplt.plot(x_vals, linha_regressao(x_vals), color='red', linewidth=2,
         label=f'Regressão Linear: y = {coef[0]:.3f}x + {coef[1]:.3f}') # Mais casas deci

# Adicionar rótulos aos eixos e título ao gráfico.
matplt.xlabel("Utilização Industrial de Água (%)")
matplt.ylabel("Taxa de Esgotamento das Águas Subterrâneas (%)")
matplt.title("Relação entre Uso Industrial de Água e Taxa de Esgotamento das Águas Subter

# Adicionar legenda e grelha.
matplt.legend()
matplt.grid(True)
matplt.tight_layout()

# Exibir o gráfico.
print(f"--- Tarefa 5 Concluída ---")
print("A exibir o gráfico de dispersão com regressão linear...")
matplt.show()

print("-----------------------------------------------------")


# --- Tarefa 6: Previsão com Machine Learning (Regressão Linear) ---
features = ['Country', 'Agricultural Water Use (%)', 'Rainfall Impact (Annual Precipitati
target = 'Per Capita Water Use (Liters per Day)'
df_ml = df[features + [target]].dropna()
```

```

# 2. Converter a variável categórica "Country" para numérica via One-Hot Encoding.
df_ml_encoded = pd.get_dummies(df_ml, columns=['Country'], drop_first=True)

# 3. Dividir os dados em conjuntos de treino (80%) e teste (20%).
X = df_ml_encoded.drop(target, axis=1)
y = df_ml_encoded[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 4. Treinar o modelo escolhido. Aqui encontram-se 3 diferentes: Regressão Linear, Lasso
#model = LinearRegression()
#model = Lasso(alpha=1.0)
model = ElasticNet(alpha=1.0, l1_ratio=0.50)
model.fit(X_train, y_train)

# 5. Avaliar o desempenho do modelo.
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Exibir resultados e interpretações.
print("Previsão de 'Per Capita Water Use'")
print(f"Número total de amostras: {len(df_ml)}")
print(f"Treino: {X_train.shape[0]} amostras | Teste: {X_test.shape[0]} amostras")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R-squared (R²): {r2:.3f}")


print("--- Tarefa 6 Concluída ---")
print("-------------------------------------------------------")
```