Escola Superior de Tecnologia e Gestão
Licenciatura em Engenharia Informática

Sistemas de Informação

Exploração e Desigualdade Laboral Global através de Dados Abertos

Data Mining / Machine Learning

João Augusto Costa Branco Marado Torres

Beja, dezembro de 2025

Instituto Politécnico de Beja

Escola Superior de Tecnologia e Gestão
Licenciatura em Engenharia Informática

Sistemas de Informação

Exploração e Desigualdade Laboral Global através de Dados Abertos

Data Mining / Machine Learning

João Augusto Costa Branco Marado Torres

Trabalho realizado no âmbito da unidade curricular de Sistemas de Informação

Orientação

Dr.ª Isabel Sofia Sousa Brito

Beja, dezembro de 2025

Júri

Responsável: Dr.ª Isabel Sofia Sousa Brito
Vogal: Dr. João Paulo Trindade
Vogal: Dr.ª Elsa da Piedade Chinita Soares Rodrigues

Conteúdo

Lista de Abreviaturas e Siglas

ETL *Extract, Transform, Load*

FAIR *Findable, Accessible, Interoperable, Reusable*

FLOSS *Free Libre and Open Source Software*

FMI Fundo Monetário Internacional

GUI *Graphical User Interface*

IED Investimento Estrangeiro Direto

ILO *International Labour Organization*

OLAP *Online Analytical Processing*

ONU Organização das Nações Unidas

PIB Produto Interno Bruto

POSIX *Portable Operating System Interface*

PPC Paridade do Poder de Compra

URI *Uniform Resource Identifier*

1 Introdução

A primeira fase deste projeto centrou-se na recolha, integração e modelação de dados abertos sobre trabalho, rendimento e condições económicas à escala global, Foi a construção de um *data warehouse* multidimensional orientado para análise *Online Analytical Processing* (OLAP). Esta infraestrutura constitui a base técnica necessária para análises sistemáticas e comparáveis entre países, regiões e períodos temporais, mitigando problemas comuns de fragmentação e inconsistência dos dados.

Uma vez assegurada essa base, torna-se possível avançar para uma segunda etapa analítica, orientada não apenas para a descrição dos dados, mas para a identificação de padrões e relações estruturais. É neste contexto que técnicas de *data mining* e *machine learning* assumem relevância, ao permitir explorar grandes volumes de dados multidimensionais de forma sistemática, indo além da análise univariada ou de correlações simples.

No domínio das ciências sociais, estas técnicas não devem ser entendidas como instrumentos de previsão determinista, mas como ferramentas exploratórias e analíticas que auxiliam a identificação de eventos recorrentes, assimetrias estruturais e relações complexas entre variáveis económicas e sociais, podendo elas contribuir para uma leitura empiríca das dinâmicas de desigualdade e exploração no capitalismo contemporâneo, se plicadas de forma crítica e acompanhadas de métricas de avaliação adequadas.

Apesar da disponibilidade crescente de indicadores socioeconómicos, a análise das desigualdades laborais globais enfrenta desafios importantes. A existência de cada vez mais variáveis relevantes, países e regiões com realidades diferentes, entre outras coisas, dificultam a identificação de padrões estruturais através de abordagens analíticas tradicionais.

O problema central desta fase do trabalho consiste, assim, em determinar de que forma técnicas de *data mining* e *machine learning* podem ser aplicadas a um *data warehouse* multidimensional para identificar padrões relevantes nas relações entre produtividade, *labour share*, dependência externa e desigualdades regionais. Coloca-se igualmente a questão de como avaliar empiricamente os resultados obtidos, garantindo que os modelos utilizados são interpretáveis, validados e metodologicamente justificados.

1.1 Objetivos

O objetivo geral desta segunda parte do projeto é aplicar técnicas de *data mining* e *machine learning* aos dados previamente integrados, de modo a identificar padrões estruturais e testar empiricamente hipóteses relacionadas com a desigualdade e a exploração do trabalho à escala global.

De forma mais específica, pretende-se:

- selecionar e preparar subconjuntos de dados adequados à aplicação de técnicas de *machine learning*;
- aplicar métodos de análise não supervisionada, como *clustering*, para identificar grupos de países ou períodos com características socioeconómicas semelhantes;
- aplicar modelos supervisionados de classificação e regressão para analisar a relação entre variáveis como produtividade, *labour share* e indicadores de dependência externa;
- avaliar o desempenho dos modelos através de métricas apropriadas, como *precision*, *recall*, *F1-score* e erro quadrático médio;
- interpretar criticamente os resultados.

1.2 Abordagem e estrutura do trabalho

Metodologicamente, o trabalho segue o processo de *Knowledge Discovery in Databases* (KDD), compreendendo as etapas de seleção, pré-processamento, transformação, aplicação de técnicas de *data mining* e interpretação/avaliação dos resultados. Esta abordagem permite estruturar de forma clara e reprodutível o percurso analítico desde os dados brutos até à extração de conhecimento.

Do ponto de vista técnico, a implementação das análises é realizada em *Python*, para variar com a escolha da etapa anterior, *R*, recorrendo a bibliotecas livres amplamente utilizadas na área da ciência de dados, como `Pandas`, `NumPy`, `Matplotlib`, `Seaborn` e `Scikit-learn`. Esta escolha visa garantir a reprodutibilidade do trabalho, a transparência metodológica e a coerência com a opção por *software* livre adotada ao longo de todo o projeto.

2 Enquadramento e Ferramentas

2.1 Dados abertos e desigualdade laboral

Tal como no relatório anterior, o foco é novamente a exploração e a desigualdade laboral à escala global. Os dados abertos, já os temos numa *data warehouse* (aquela construída na etapa anterior) a partir de um processo ETL. A diferença está na forma de como vamos analisar analisar os dados. Na etapa anterior, analisamos os dados com base na sua evolução ao longo do tempo, respondendo à pergunta "o que foi que aconteceu?". Agora, a pergunta é mais "o que está a acontecer (agora) e o que vai acontecer". *Data mining* é um processo de descoberta de padrões e outras informações valiosas, a partir de grandes conjuntos de dados, como é o exemplo da nossa *data warehouse*.

Na etapa anterior, foi feita uma análise comparativa entre regiões e grupos de rendimento, onde se verificava uma diferença clara entre os países centrais e os periféricos no que toca a níveis de *labour share* e de produtividade. Mas isso apenas acontecia quando olhávamos para os dados a nível global. Era um paradoxo de Simpson, onde vendo os dados todos agregados, dava a ideia de que países mais produtivos garantem um melhor salário ao trabalhador, mas quando faziamos a mesma análise regionalmente, viamos uma história diferente: o aumento da produtividade não se traduzia num aumento igual do *labour share*, que ou continuava estagnado, ou até descia um pouco! Isto em todo o globo. Esta análise não seria possível sem a possiblidade de fazer OLAP, mostrando a importância de um modelo multidimensional. E isto apenas usando alguns indicadores macroeconómicos.

Também mostrou-se a capacidade dos dados abertos e do software livre de fazer todas as tarefas que um software proprietário consegue fazer, e para além disso, conseguimos ter um projeto reproduzivél em qualquer máquina — desde que tenha o software livre instalado obviamente, mas isso ia ser um requesito para qualquer software que eu escolhe-se usar) —, e que partilha uma posição ética sobre o papel do software livre no ensino.

2.2 *Data mining* e descoberta de conhecimento

Como foi dito brevemente em cima, uma análise, usando OLAP, é uma análise descritiva. Usando dados que já existem das diferentes dimensões da *data warehouse*, tu crias um sumário sobre o que foi que aconteceu, onde e porquê. Não é automáticamente que tu vais ter a razão para a resposta para estas perguntas, e para isso, tens que recorrer ao *data mining* e *machine learning*.

O objetivo do *data mining* é aprender sobre informações escondidas, relações causais e preditivas, correlações, entre os dados. reponde a perguntas como o que é mais provável, o que vai ou pode acontecer ou porquê que aconteceu.

Apesar de serem coisas diferentes, eu acho que ambas se complementam muito bem. O processo de *data mining* não seria possível sem alguém primeiro ter feito o processo da engenharia de dados, ou pelo menos seria mais complicado. Por eu ter feito o processo de análise OLAP, isso deu-me a capacidade de eu saber o quais são os padrões que eu quero descobrir com o *data mining*, até para ver se esses resultados suportam a minha análise. Também seria possível concerteza exportar os resultados dos modelos de *data mining* para a *data warehouse* através do processo ETL, criando uma espécie de *loop* onde o conhecimento, cria mais conhecimento, mas sem nunca substituir o valor dos dados originais.

Existem vários modelos de *data mining*. Uns são mais adequados para dados discretos (valores finitos) e outros para dados contínuos (valores infinitos). Esses modelos também se podem separar em modelos de *supervised* e *unsupervised* textitlearning. Os modelos *supervised* recebem variáveis independentes associadas com um rótulo representante da tupla dessas variáveis independentes (variável dependente), que pode ser uma variável discreta ou contínua. No caso dos modelos *unsupervised*, ao modelo é apenas lhe dados as variáveis independentes, e tem que achar qual seria um valor que seria a substituição da variável dependente em falta.

O tipo das variável independente vai influenciar o tipo do modelo também. Por exemplo, no caso dos modelos de *supervised learning*, quando as variáveis dependentes são discretas, estamos perante um modelo de classificação, e quando são contínuas, de um modelo de regressão.

No ano passado fiz um projeto (Costa Branco Marado Torres, 2024) onde aprendi os básicos dos modelos *supervised*, implementando do zero um sistema de *neural network* sem o uso de bibliotecas, já que elas funcionam como uma caixa fechada que, de certa forma, fazem com que tu não a queiras abrir para aprender como é que ela realmente funciona por dentro, e que por isso, apesar de essas bibliotecas facilitarem a tua vida, elas abstraem o conhecimento e não te deixam entender de verdade as coisas. Esse projeto certamente ajudou-me neste agora. Foi um projeto onde aprendi os fundamentos, que são necessários para aprender tudo o resto se um dia quiser.

2.3 Ferramentas e bibliotecas utilizadas

Nesta etapa, decidi variar e usar o Python como linguagem de programação. Não foi apenas para variar. É também porque sei que na área de *data mining* e *machine learning*, o que não faltam são bibliotecas.

Como o *data warehouse* está numa base de dados DuckDB, usei a API que eles têm para Python apenas na fase de seleção de dados do KDD **??**.

Na decomentação dessa API, encontrei 4 formas de trabalhar com os dados retirados da *data warehouse* que fossem substituir os `data.frames` do R, tirando os objetos do Python:

- Pandas — uma ferramenta para manipulação e análise de dados;
- Polars — também uma ferramenta para manipulação de dados, mas que promete mais eficiência que as alternativas;
- Apache Arrow — ferramenta para guardar e manipular dados num formato colunar, na memória, que depois pode ser acessado por outros processos. Não parece ser bem o que precisamos;
- NumPy — segundo eles «a biblioteca fundamental para computação científica».

As únicas opções que realmente pareceram uma alternativa aos `data.frames` foram as primeiras duas, sendo a primeira a única de que já tinha ouvido falar das duas, e por isso optei por usar essa.

Não desvalorizando o NumPy, que no futuro poderá ter outras funcionalidades que venham a ser necessárias. Mas por enquanto, fica na gaveta.

Fui procurar ao sítio web roadmap.sh mais informação que me poderia ajudar, porque sabia que cada vez exitem mais *roadmaps* e que alguns deles eram relacionados a esta tarefa. No *roadmap Machine Learning* encontrei refrências à biblioteca `scikit-learn`: «Scikit-learn é uma biblioteca Python gratuita e de código aberto que fornece ferramentas simples e eficientes para análise de dados e aprendizagem automática. Possui vários algoritmos para classificação, regressão, agrupamento, redução de dimensionalidade, seleção de modelos e pré-processamento. É construída sobre NumPy, SciPy e matplotlib, facilitando a integração com outras bibliotecas científicas Python.»

3 Metodologia

Existem muitas tarefas que involvem *data mining* que eu consigo fazer usando os dados da minha *data warehouse*. Fazemos *clustering* para separar países ou regiões em como sendo do centro ou da períferia. Prevemos qual vai será o país com a maior descida do *labour share* nos próximos anos. Conseguimos identificar períodos de crise e austeridade. Identificamos *outliers*.

Este trabalho segue o processo KDD. Começamos com a preparação de dados. Na seleção, tentamos remover de imediato dados que podem projudicar os resultados do nosso modelo. Bons dados produzem bons resultados. Depois de os dados a utilizar estárem defenidos, passamos por uma etapa de pré-processamento onde tentamos polir os dados da seleção, que podem vir incompletos ou inconsistentes, ou alguma parte deles pode até ser redundante, e por isso descartados. Por fim, e para não modificar mais os dados daqui para a frente, normalizamos, derivamos valores a partir de outros, discretizamos (transformando variáveis contínuas em variáveis discretas). Neste momento estamos prontos para aprender. Aplicamos o algoritmo e avaliamos as respostas, e com base nelas tomamos desisões que podem servir para reformar os passos anteriores de forma a obter uma melhor resposta da próxima vez.

3.1 Modelo de dados e Seleção

O modelo de dados é o resultado da etapa de engenharia de dados deste projeto. Consiste num modelo multidimensional de esquema floco de neve para *data warehouse*.

Para esta etapa agora, vou decidir usar as seguintes informações desse modelo:

- Informações geográficas acerca de países e regiões;
- Indicadores macroeconómicos como o *labour share*, a ”produtividade”, os fluxos IDE, o índice Gini, salários;
- O tempo.

A razão de escolhermos esses indicadores específicos são porque foram os utilizados também na etapa de análise da etapa anterior do projeto, o que permite no final ligar os resultados do data mining com essa análise.

Os dados do tempo começam desde o ano 1960, mas com bastante falta de dados nos países periféricos da altura. Então vou diminuir um pouco a janela em 10 anos, que coincide com:

- Uma época de choques petrolíferos — a 16 de outubro de 1973, delegados da Organização dos Países Exportadores de Petróleo (OPEP) decidem aumentar a 70% o preço do petróleo, e no dia seguinte diferenciam os fornecimentos com base na posição dos países consumidores e relação à guerra do Yom Kippur. A 5 de março de 1979, no Irão a exportação de petróleo é retomada, em quantidade reduzida à metade em relação ao nível normal antes da crise;
- Uma época de estagflação — inflação alta e um alto nível de desemprego ao mesmo tempo;
- Foi também nessa altura que a Europa e os EUA acabaram com a conversibilidade do ouro, acelerando as suas financeirizações.

Todos estes dados e informações são comparáveis e de relevância.

Mesmo assim, ainda é muito provável que vá existir muita falta de dados mesmo passado 10 anos, especialmente nos países periféricos, como tinha sido notado durante o ETL. Mas talvez a falta de dados faça com que o modelo entenda alguma diferença entre diferentes unidades de análise, e que por isso permita responder com informações mais interessantes. Mas também pela relevância histórica do século de 1970.

O que nós pretendemos analisar são países num determinado ano civil.

3.2 Pré-processamento

Apesar de já termos diminuido a janela do tempo, ainda vão faltar dados, por diversas razões. Os dados podem não ser disponibilizados de forma consistente, nem que seja anualmente. Na maioria das vezes a ausência dos dados estão estruturalmente relacionados com posições periféricas.

Os valores em falta foram tratados utilizando interpolação linear específica por país, aplicada apenas a lacunas internas curtas. Não foi realizada qualquer extrapolação, e as lacunas longas foram deixadas em falta. Esta abordagem preserva a continuidade temporal, evitando simultaneamente a fabricação de ausência de dados estruturais.

Também é importante pensar sobre os *outliers*, e como tratar dos mesmos. Mas muito desse trabalho consegue ser feito da etapa **??** já asseguir.

3.3 Transformação dos dados

Alguns dados fará sentido serem normalizados. Um uso da normalização na primeira etapa do projeto foi na produtividade que variava demasiado entre regiões, e que por isso desviava as atenções do que realmente eu queria analisar. Ao normalizar a produtividade, "acabou-se" a variadade.

Um exemplo de uma possível transformação está nos fluxos IED, onde tu encontras regiões com fluxos positivos de milhares de milhões, talvez até bilhões de IED, e outras regiões com fluxo negativo de alguns milhões de IED. As desparidades são enormes.

Talvez seja necessário ocorrer ao uso a discretização para transformar variáveis contínuas (com valores infinitos) em variáveis discretas (onde tu consegues contar a quantidade de valor que a variável consegue ter). É uma transformação importante para modelos de classificação por exemplo.

É nesta etapa onde começamos a usufruir das bibliotecas Python já mencionadas.

3.4 Processo de *data mining* (KDD)

The analytical workflow combines interactive notebooks with text-based representations using Jupytext, enabling reproducible execution and version control. Analytical results are exported as LaTeX fragments and incorporated directly into the final report, ensuring consistency between code, results and documentation.

O fluxo de trabalho nesta etapa combina *notebooks*, outra vez usando R Markdown, permitindo a execução reproduzível e o controlo de versão. Os resultados analíticos são exportados como fragmentos LaTeX e incorporados diretamente neste relatório final, garantindo a consistência entre o código, os resultados e a documentação.

3.4.1 *Clustering* de países por posição estrutural

Um dos objetivos exploratórios deste trabalho consiste na identificação de grupos de países com dinâmicas semelhantes em termos da relação trabalho–capital, isto é, à forma como produtividade, salários, *labour share* e desigualdade evoluem de forma conjunta ao longo do tempo. A técnica de *clustering* permite abordar este problema sem impor categorias *a priori*, sendo, por isso, adequada a uma análise exploratória de natureza estrutural.

Cada país é descrito por um conjunto de indicadores macroeconómicos e laborais derivados do modelo multidimensional construído na etapa de engenharia de dados **??**. Todos os dados utilizados nesta etapa correspondem à versão final pré-processada, após tratamento de valores em falta e transformações necessárias, conforme descrito na secção metodológica anterior.

O recurso ao *clustering* apresenta uma vantagem adicional: os países encontram-se já classificados, na *data warehouse*, por região geográfica e níveis de rendimento. Tal permite comparar os agrupamentos obtidos por métodos de *data mining* com categorias amplamente utilizadas na literatura económica, avaliando em que medida estas correspondem — ou não — a padrões empíricos observáveis nos dados.

Foram considerados dois algoritmos de *clustering*: o método dos k-médias (*k-means*) e o *Ward's method*. O primeiro permite identificar partições compactas baseadas na minimização da variância intra-grupo, enquanto o segundo fornece uma representação hierárquica das semelhanças entre países, sendo particularmente útil para a análise exploratória de estruturas aninhadas.

É um bom desafio apenas para começar, e talvez, como foi para mim, a primeira idea de como aplicar *data mining* num modelo multidimensional como o deste projeto, e a contextualização da **??**, que te vêm à cabeça. Por ser também um problema de *unsupervised learning*, tu sabes os dados que tens para oferecer ao modelo, mas não sabes ainda o que queres tirar de lá, e talvez por essa natureza, seja o primeiro tipo de problema que me veio à cabeça, quase como se tivesse que pensar menos.

A escolha do *clustering* como primeira técnica de *data mining* aplicada neste projeto pode ser que por ele se tratar de um problema de *unsupervised learning*, parte-se do pressuposto de que os dados disponíveis descrevem padrões estruturais ainda não totalmente conhecidas. O objetivo não é revelar esses padrões, regimes e trajetórias possíveis que possam, posteriormente, ser interpretados com base no enquadramento teórico apresentado na introdução **??**.

Para que o k*-means* calcule cada distância entre os pontos, cada valor de da tupla de dados da unidade de análise precisa existir.

Excluir os valores ausentes aqui parece ser a melhor opção, uma vez que a fase de pré-processamento e transformação já agiu sobre os valores ausentes. E caso mais transformações aos dados tenhamm que ser feitas, isso é algo para a etapa anterior.

	feature	Valores ausentes	%
labor_share_final	labor_share_final	6498	61.05995
productivity_final	productivity_final	4694	44.10825
gini_final	gini_final	7817	73.45424
fdi_net_gdp_final	fdi_net_gdp_final	2450	23.02199
avg_hourly_wage_final	avg_hourly_wage_final	9698	91.12949

```
## 10067 linhas eliminadas
```

Esta escolha fez-nos perder 10 mil linhas.

O k-means usa o número de clusters como parâmetros e a distância entre pontos como métrica.

O *Ward's method* pode usar o número de clusters ou limites de distância e usa a mesma métrica que o k-means.

Vamos ler o que o `scikit-learn` tem para nos ensinar acerca do k-means:

> O algoritmo k-means agrupa dados tentando separar amostras em n grupos de variância igual.
>
> O algoritmo k-means divide um conjunto de N amostras X em K *clusters* disjuntos C, cada um descrito pela média das amostras no *cluster*. As médias são comumente chamadas de «centróides» do *cluster*;
>
> O k-means é frequentemente referido como algoritmo de Lloyd. Em termos básicos, o algoritmo tem três etapas. A primeira etapa escolhe os centróides iniciais, sendo o método mais básico escolher amostras do conjunto de dados. Após a inicialização, o k-means consiste em um *loop* entre as duas outras etapas. A primeira etapa atribui cada amostra ao seu centroide mais próximo. A segunda etapa cria novos centroides, tomando o valor médio de todas as amostras

> atribuídas a cada centroide anterior. A diferença entre os centroides antigos e novos é calculada e o algoritmo repete estas duas últimas etapas até que este valor seja inferior a um limite. Em outras palavras, ele repete até que os centroides não se movam significativamente.

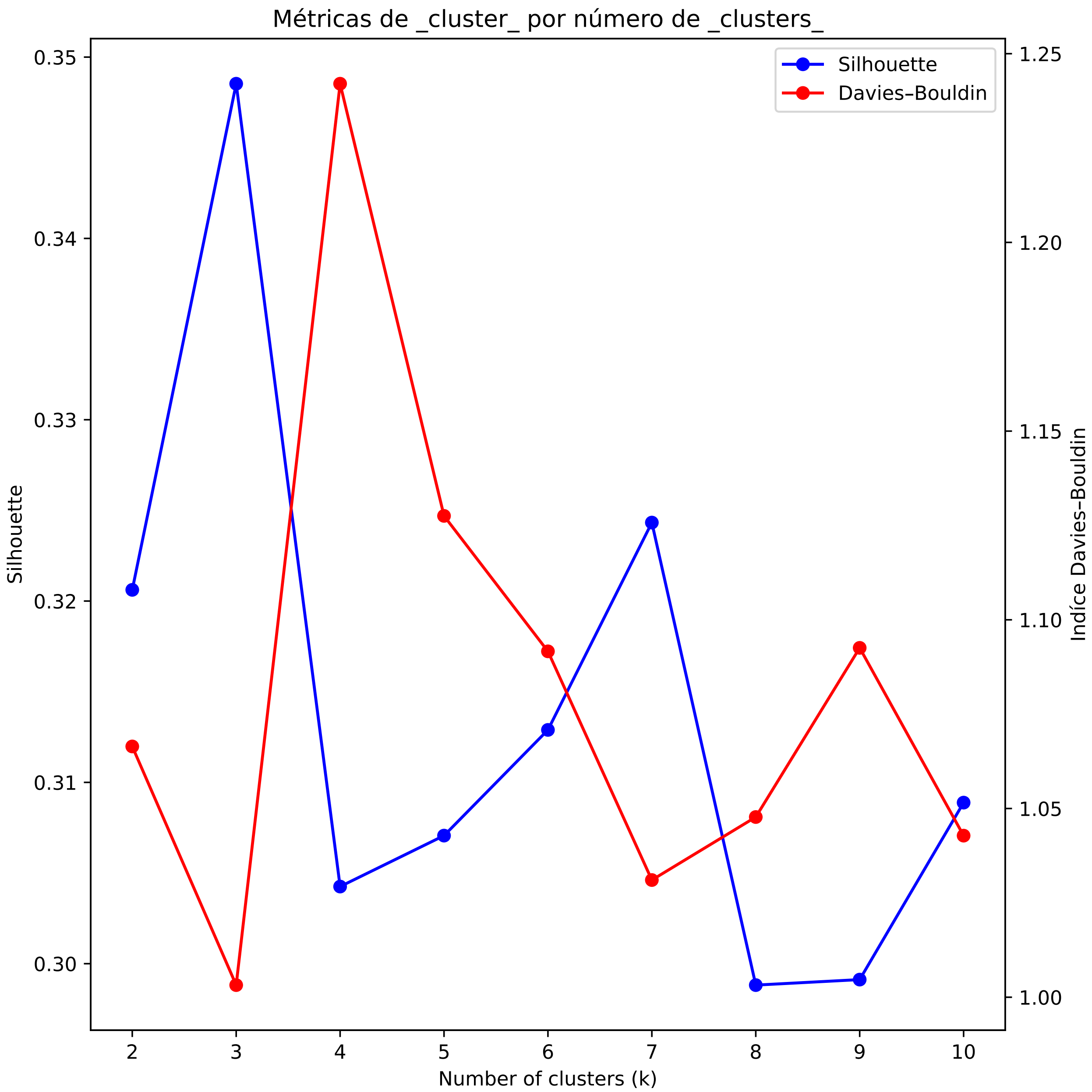
Também é possível encontrar exemplos de como usar a biblioteca para esta tarefa específica.

Todas as variáveis foram normalizadas antes de criar os *clusters* usando o `StandardScaler.fit_transform`. Isso porque queremos que os centróides estejam próximos da média de um *cluster*. Também podemos ter algumas *features* da unidade de análise com escalas muito diferentes umas das outras.

Vamos começar a usar a biblioteca `scikit-learn` para termos acesso ao algoritmo k-means sem o ter que implementar.

Mas antes, temos que decidir qual é o número de *clusters* que nós queremos ter no resultado da análise. Existem uns números que são melhores que outros, e existem algoritmos, como o *silhouette* e o de indíce de *Davies–Bouldin* que nos dizem quais são.

Dentro das respostas que o *silhouette* nos dá, devemos escolher o maior valor, ou um dos máximos locais, e evitar números muito grandes sem uma justificação teorica. Para o índice de *Davies–Bouldin*, menor o valor, melhor.



Reparamos que a melhor escolha para número de clusters com certeza é o 3, apesar de que escolher 7 não parece uma opção tão má assim, aliás, na *data warehouse* os países estão dívididos em 7 regiões.

Aplicando o algoritmo, conseguimos ver quantos países (juntamente com o ano) é que ficaram em cada um dos clusters. O objetivo é não ficar com clusters pequenos, ou com um enorme que engloba quase tudo.

	x
0	410
1	132
2	33

Conseguimos ver o valor de cada *feature* para cada *centroid*.

labor_share_final	productivity_final	gini_final	fdi_net_gdp_final	avg_hourly_wage_final	cluster
47.60750	35899.81	42.31411	0.0283984	406.34766	0
60.43545	111838.21	35.01023	-0.0062086	24.12754	1
49.35324	23538.45	38.65000	0.0253379	7720.74122	2

Também é possível analisar com que grupo de rendimento e região do globo é que cada centroid se identifica mais, ou de outra forma, como é que um país, dependendo da sua região geográfica e nível de rendimento, fica agrupado com outros.

	High income	Low income	Lower middle income	Upper middle income
0	0.1466993	0.0488998	0.2640587	0.5403423
1	0.9621212	0.0000000	0.0075758	0.0303030
2	0.0000000	0.0303030	0.4242424	0.5454545

	East Asia & Pacific	Europe & Central Asia	Latin America & Caribbean	Middle East & North Africa	North America	South Asia	Sub--Saharan Africa
0	0.1243902	0.1317073	0.5341463	0.0682927	0.0000000	0.0341463	0.1073171
1	0.0000000	0.6742424	0.0303030	0.0000000	0.2878788	0.0000000	0.0075758
2	0.6969697	0.0000000	0.2727273	0.0000000	0.0000000	0.0000000	0.0303030

Vamos criar uma visualização, usando como base o exemplo "Visualize the results on PCA-reduced data".

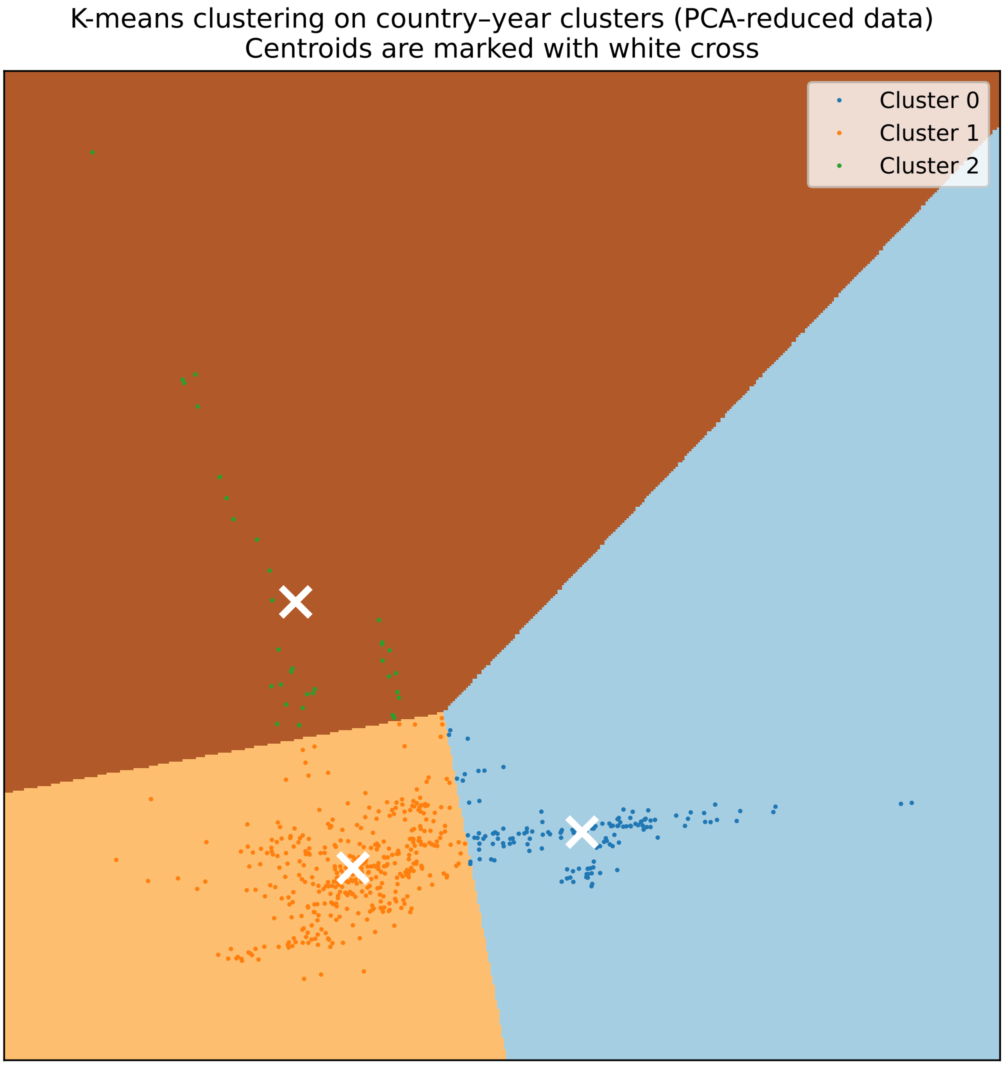
```
## KMeans(n_clusters=np.int64(3), n_init=20)
```

```
## (-4.563880357651497, 6.710042047616408)
```

```
## (-2.673281992085398, 9.48497381266387)
```

```
## ([], [])
```

```
## ([], [])
```



Um modelo de agrupamento k-means foi aplicado às observações por país e ano utilizando indicadores macroeconómicos e laborais padronizados. O número de agrupamentos foi selecionado através de uma análise de *silhouette*, e os agrupamentos resultantes foram interpretados comparando a composição dos agrupamentos com as classificações regionais e de rendimento existentes.

4 Conclusão

Este trabalho teve como objetivo explorar a aplicação de técnicas de *data mining* e *machine learning* a um *data warehouse* multidimensional construído a partir de dados abertos sobre trabalho e economia à escala global. Partindo de uma base sólida de engenharia de dados e análise OLAP desenvolvida na etapa anterior do projeto, esta fase procurou avançar para uma análise exploratória mais profunda, orientada para a identificação de padrões estruturais nas desigualdades laborais.

A utilização de métodos de aprendizagem não supervisionada, em particular técnicas de *clustering*, permitiu agrupar países–ano com base em indicadores macroeconómicos e laborais relevantes, sem impor categorias pré-definidas. Os resultados obtidos evidenciam que as dinâmicas observadas não se reduzem a simples níveis de desenvolvimento, mas refletem posições estruturais distintas na divisão internacional do trabalho, coerentes com a literatura crítica sobre desigualdade e dependência económica.

Do ponto de vista metodológico, o trabalho demonstrou a importância do processo KDD enquanto enquadramento estruturado para a descoberta de conhecimento, bem como o papel central do pré-processamento, transformação e avaliação dos modelos na obtenção de resultados interpretáveis e consistentes. A comparação entre métricas como o *silhouette score* e o índice de *Davies–Bouldin* revelou-se essencial para fundamentar a escolha dos modelos e parâmetros utilizados.

Em termos técnicos, a opção por ferramentas de *software* livre e dados abertos revelou-se adequada, permitindo garantir a reprodutibilidade, transparência e extensibilidade do trabalho. A integração entre análise programática e documentação científica reforça igualmente a coerência entre código, resultados e interpretação.

Por fim, importa salientar que os resultados apresentados devem ser entendidos como exploratórios. As limitações inerentes à disponibilidade e qualidade dos dados, bem como às escolhas metodológicas realizadas, apontam para a necessidade de análises futuras, nomeadamente através da aplicação de modelos supervisionados e da incorporação de novas variáveis. Ainda assim, o trabalho contribui para uma leitura empírica crítica das desigualdades laborais globais e demonstra o potencial das técnicas de *data mining* como ferramentas de análise no domínio das ciências sociais.

Referências Bibliográficas

Costa Branco Marado Torres, J. A. (2024, dezembro). *LANGuage IDentification* (Versão 0.1.0). https://doi.org/10.5281/zenodo.17502601

Licença