Sistema de Gestão de Dados para a Bioeconomia da Resina

Ana Carolina Menoita

Objetivo do projeto

Este projeto tem como objetivo implementar um sistema de gestão de dados (SGD) para analisar a bioeconomia da resina em florestas portuguesas de *Pinus pinaster*. O sistema utiliza dados sobre área florestal, custos de extração, impacto ambiental, mão de obra e produção de resina para fornecer insights sobre a indústria da resina. O sistema foi pensado para facilitar a tomada de decisões informadas que promovam otimizar a produção de resina, com boas praticas.

Estrutura do projeto

Bioeconomy_resin/

|---- original_data/ #Arquivos de CSV, baixados diretamente do INE

|---- processed_data/ #Um código em python para limpar os dados gerados pela IA, os dados provenientes do INE foram trabalhados no OpenRefine

|---- sql_scrits/ #Scripts para a criação e implementação dos dados

|---- data_use_scripts/ #Scripts SQL para análise de dados

|---- documents/ #Relatorio

Recolha e preparação dos dados

Os dados forma recolhidos a partir do Instituto Nacional de Estatística (INE), como gerados pela IA (inteligência artificial) e organizados em cinco conjuntos principais, representados em arquivos CSV:

1. **forest_areas.csv**

   ID_region – Identificador único da região (PK)

   hectares – Área florestal em hectares

2. **extraction_costs.csv**

   ID_region – Identificador único da região (PK, mas também é uma chave estrangeira que da referência a forest_areas.)

   Region – Nome da região

   Extraction_Cost – Custo de extração por Kg

Production – Custo de produção €/kg

3. **workforce.csv**

   ID_region – Identificador único da região (PK, mas também é uma chave estrangeira que da referência a forest_areas.)

   Region – Nome da região

   Workers – Número de trabalhadores por região

4. **environmental_impact.csv**

   ID_region – Identificador único da região (PK, mas também é uma chave estrangeira que da referência a forest_areas.)

   CO2_Avoided – Quantidade de CO2 é evitado

5. **resin_production.csv**

   ID_region – Identificador único da região (PK, mas também é uma chave estrangeira que da referência a forest_areas.)

   Production_kg – Produçao em Kg

   Production_Euros – Produçao em Euros

Os dados provenientes do INE foram limpos utilizando o OpenRefine e um script Python que se encontra na pasta processed_data, para garantir que estivesse tudo pronto para a importação no banco de dados.

Chaves Primárias e Estrangeiras

As chaves primárias garantem que cada registro nas tabelas seja único.

O campo ID_region foi utilizado como chave primária nas tabelas correspondentes, garantindo que não exista duplicações.

As chaves estrangeiras estabelecem o relacionamento entre as tabelas, permitindo que os dados se interrelacionem.

O campo ID_region nas tabelas extraction_cost, environmental_impact, workforce, resin_production serve como chave estrangeira, dando referencia a chave primária na tabela forest_areas, o que garante a integridade referencial e permite consultas mais complexas que envolvem múltiplas tabelas.

Estrutura do banco de dados e implementação

O banco de dados foi criado e projetado em 5 tabelas principais, cada uma reflete um conjunto específico de dados.

A implementação do banco de dados foi realizada utilizando MariaDB.

Criação da base de dados a utilizar e o comando para se usar esta base de dados para criar as tabelas.

```
#Criar a base de dados
CREATE DATABASE IF NOT EXISTS bioeconomy_resin;
USE bioeconomy_resin;
```

Criação das tabelas e importação dos dados

```
#Importar os dados da florest_areas
CREATE TABLE florest_areas (
    ID_region INT PRIMARY KEY,
    region VARCHAR(100),
    hectares FLOAT
);
LOAD DATA LOCAL INFILE 'C:\Users\anaca\DMSPROJECT\bioeconomy_resin\processed_data\forest_areas.csv'
INTO TABLE florest_areas
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

#Importar os dados da extraction_costs
CREATE TABLE extraction_costs (
    ID_region INT PRIMARY KEY,
    region VARCHAR(100),
    Extraction_Cost FLOAT,
    Production_Cost FLOAT
);
LOAD DATA LOCAL INFILE 'C:\Users\anaca\DMSPROJECT\bioeconomy_resin\processed_data\extraction_costs.csv'
INTO TABLE extraction_costs
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(ID_region, region, Extraction_Cost, @Production_Cost)
SET Production_Cost = @Production_Cost;

#Importar os dados da environmental_impact
CREATE TABLE environmental_impact (
    ID_region INT PRIMARY KEY,
    region VARCHAR(100),
    CO2_Avoided FLOAT
);
LOAD DATA LOCAL INFILE 'C:\Users\anaca\DMSPROJECT\bioeconomy_resin\processed_data\environmental_impact.csv'
INTO TABLE environmental_impact
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```sql
#Importar os dados da workforce
CREATE TABLE workforce (
    ID_region INT PRIMARY KEY,
    region VARCHAR(100),
    Workers INT
);
LOAD DATA LOCAL INFILE 'C:\Users\anaca\DMSPROJECT\bioeconomy_resin\processed_data\workforce.csv'
INTO TABLE workforce
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

#Importar os dados da resin_production
CREATE TABLE resin_production (
    ID_region INT PRIMARY KEY,
    Production_T FLOAT,
    Production_Euros FLOAT
);
LOAD DATA LOCAL INFILE 'C:\Users\anaca\DMSPROJECT\bioeconomy_resin\processed_data\resin_production.csv'
INTO TABLE resin_production
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(@ID_region, @Production_T, @Production_Euros)
SET ID_region = @ID_region,
    Production_T = @Production_T,
    Production_Euros = @Production_Euros;
```

A importação dos dados foi deita após a criação de cada tabela, quis importar os dados dos arquivos CSV para o banco de dados utilizando primeiramente o comando LOAD DATA INFILE, mas como me estava a dar um erro pedindo ajuda a IA foi aconselhado escrever LOAD DATA LOCAL INFILE, este comando permite carregar os dados diretamente dos arquivos de CSV para as tabelas correspondentes.

Eu não sei se foi o comando utilizado, o caminho mal escrito ou os lines terminted que me causaram o erro de não conseguir colocar os dados na tabela.

Os @ presentes, são de variáveis temporais que estão a armazenar valores lidos dos arquivos CSV, a instrução SET é usada para atribuir esses valores às colunas correspondentes na tabela após a importação.

Análise dos Dados

Após a importação “bem-sucedida” dos dados, desenvolvi outro script SQL dataanalysis.sql, que realiza consultas para responder a 10 perguntas especificas sobre os dados.

1. **Produção total de resina por região e a relação com a área florestal**

   Vai fornecer informação sobre a área florestal e a produção total em Kg e euros por região, além da eficiência da produção em relação à área disponível.

```sql
USE bioeconomy_resin;

#Produção total de resina por região e relação com área florestal
SELECT fa.region,
       fa.hectares AS florest_areas,
       rp.Production_Kg AS resin_production_kg,
       rp.Production_Euros AS resin_production_euros,
       rp.Production_Kg / af.hectares AS production_hectare
FROM florest_areas fa
INNER JOIN resin_production rp ON fa.ID_region = rp.ID_region;
```

2. **Eficiência da produção de resina por trabalhador**
   Calcula quantos kg de resina são extraídas por "trabalhador" em cada região, permitindo identificar quais são as regiões mais eficientes.

```sql
#Eficiência da produção de resina por trabalhador
SELECT wf.region,
       rp.Production_T / wf.Workers AS efficiency_kg_per_worker
FROM workforce wf
INNER JOIN resin_production rp ON wf.ID_region = rp.ID_region
ORDER BY efficiency_kg_per_worker DESC;
```

3. **Custo de extração e produção por região**
   Apresenta os custos médios da extração e produção por região

```sql
#Custos de extração e produção por região
SELECT region,
       Extraction_Cost AS extraction_cost,
       Production_Cost AS production_cost
FROM extraction_cost;
```

4. **CO2 evitado por região**
   Quanto cada região evita em termos de CO2 em relação a quantidade produzida, fornecendo uma métrica sobre o impacto ambiental da produção.

```sql
#CO2 evitado por regiao
SELECT ei.region,
       ei.CO2_Avoided AS co2_evitado,
       rp.Production_Kg,
       ei.CO2_Avoided / rp.Production_kg AS co2_avoided_per_Kg
FROM environmental_impact ei
INNER JOIN resin_production rp ON ei.ID_region = rp.ID_region;
```

5. **Relação entre o número de trabalhadores com a produção de resina**
   Como o número de trabalhadores se relaciona com a quantidade total produzida em cada região

```sql
#Relaçao entre o numero de trabalhadores com a producao de resina
SELECT wf.region,
       wf.Workers,
       rp.Production_Kg,
       rp.Production_Kg / wf.Workers AS production_per_worker
FROM workforce wf
INNER JOIN resin_production rp ON wf.ID_region = rp.ID_region;
```

6. **Relação custo benefício da produção de resina**
   Calcula a relação custo-benefício da produção, permitindo identificar quais regiões têm melhor retorno financeiro em relação aos custos.

```sql
#Relaçao custo-beenficio da producao de resina
SELECT ec.region,
       rp.Production_Euros / (ec.Extraction_Cost + ec.Production_Cost) AS cost_benefit_ratio
FROM extraction_costs ec
INNER JOIN resin_production rp ON ec.ID_region = rp.ID_region
ORDER BY cost_benefit_ratio DESC;
```

7. **Comparação entre euros e kg**
   Compara o valor gerado pela venda da resina em euros com a quantidade produzida em kg, auxiliando a entender o preço médio por kg.

```sql
#Comparaçao do euros e de kg
SELECT region,
       Production_Kg,
       Production_Euros,
       Production_Euros / Production_Kg AS value_per_kg
FROM resin_production
ORDER BY value_per_kg DESC;
```

8. **Relação entre a área florestal e a eficiência da produção**
   Analisa como a eficiência da produção se relaciona com a área florestal disponível em cada região

```sql
#Relaçao entre a area florestal e a eficiencia da produçao
SELECT fa.region,
       fa.hectares,
       rp.Production_Kg,
       rp.Production_Kg / fa.hectares AS efficiency_per_hectare
FROM florest_areas fa
INNER JOIN resin_production rp ON fa.ID_region = rp.ID_region
ORDER BY efficiency_per_hectare DESC;
```

9. **Potencial de expansão (considerando a área florestal e a eficiência atual)**
   Avalia o potencial para expansão da produção com base na área florestal disponível e na eficiência atual.

```sql
Potencial de expansao (considerando a area florestal e a efeciencia atual)
SELECT fa.region,
       fa.hectares,
       rp.Production_Kg,
       rp.Production_Kg / fa.hectares AS current_efficiency,
       fa.hectares * (SELECT MAX(Production_Kg / hectares) FROM resin_production JOIN florest_areas ON resin_production.ID_region = florest_areas.ID_region) AS potential_output
FROM florest_areas fa
INNER JOIN resin_production rp ON fa.ID_region = rp.ID_region
ORDER BY potential_output - rp.Production_Kg DESC;
```

10. **Relação entre o impacto ambiental e a escala de produção**
    Analisa como o impacto ambiental se relaciona com a quantidade produzida em cada região.

```sql
#Relaçao entre o imapcto ambiental e a escala de produçao
SELECT ei.region,
       ei.CO2_Avoided,
       rp.Production_Kg,
       ei.CO2_Avoided / rp.Production_Kg AS co2_avoided_per_Kg
FROM environmental_impact ei
INNER JOIN resin_production rp ON ei.ID_region = rp.ID_region
ORDER BY co2_avoided_per_Kg DESC;
```

Problemas encontrados e tentativa de resolução

Durante a execução do script SQL para importar os dados, enfrentei alguns erros que impediram a geração bem-sucedida dos dados no banco de dados, dentro deles, penso que tenha sido:

- Caminho dos arquivos -> o comando LOAD DATA INFILE requer que o caminho do csv esteja correto, verifiquei algumas vezes o caminho, tendo utilizado o método de copiar o

caminho do próprio ficheiro, mudei o comando para LOAD DATA LOCAL INFILE e o erro continuou.

- Formato dos arquivos em csv -> não observei inconsistência nos delimitadores, ou na formatação do arquivo.

- Permissões do MariaDB -> Utilizei uma ferramenta data pela IA que conseguisse observar se estava “on” ou “off” esta configuração, no qual me deparei que estava on e reinicie o programa, podia ser algum bugg. Também verifiquei se tinha acesso ao diretório correto, onde todos os arquivos estavam.

- Mensagem de Erro -> Ao tentar executar os scipts SQL, apareceu-me diversas mensagens de erro que não consegui resolver.

Conclusão

Embora tenha tentado implementar este DMS e importar os dados necessários para a analise, tive como obstáculos algumas dificuldades técnicas que me impediram uma conclusão bem-sucedida do projeto.

Gostaria de ter um feedback, sobre como poderia proceder a resolução destes tipo de problemas, devido a não observar nenhuma falha de Syntax, como também poderia melhorar este trabalho.

Os dados que se encontram disponíveis são muito escassos, incluindo os estatísticos (só estavam disponíveis dos anos 2013 e 2024), o que me dificultou a encontrar dados para desenvolver mais sobre o assunto, que tipo de sites, numa próxima devo procurar para conseguir obter uma melhor qualidade de dados?