PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 Topic: Liên kết dữ liệu ChEMBL với dữ liệu thực tế

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu Chembl 35 này. Với vai trò là một chuyên gia nghiên cứu và phát triển dược học, tôi hiểu rõ tầm quan trọng của việc phân tích dữ liệu Chembl để khám phá và phát triển thuốc mới.

Dựa trên thông tin bạn cung cấp, chúng ta sẽ cùng nhau xây dựng một quy trình phân tích hiệu quả, tuân thủ theo chuẩn AIMLOps Template và giải quyết các vấn đề kỹ thuật phát sinh.

1. Phân tích mô hình phân tích (Analysis Model)

Chúng ta sẽ tập trung vào chủ đề "Topic_CheMBL_35_60", tôi hiểu rằng bạn muốn phân tích một khía cạnh cụ thể của dữ liệu Chembl 35. Để đưa ra một mô hình phân tích chi tiết, tôi cần bạn cung cấp thêm thông tin về mục tiêu cụ thể của bạn. Ví dụ:

- Bạn muốn tìm hiểu mối liên hệ giữa cấu trúc hóa học và hoạt tính sinh học của một nhóm hợp chất cụ thể?
- Bạn muốn xây dựng mô hình dự đoán hoạt tính của các hợp chất mới dựa trên dữ liệu Chembl?
- Bạn muốn khám phá các "druggable targets" tiềm năng bằng cách phân tích dữ liệu về các protein và tương tác của chúng với các hợp chất?

Dựa trên mục tiêu cụ thể của bạn, chúng ta có thể lựa chọn các phương pháp phân tích phù hợp, chẳng hạn như:

- **Phân tích tương quan cấu trúc-hoạt tính (QSAR/SAR):** Sử dụng các thuật toán học máy để xây dựng mô hình dự đoán hoạt tính dựa trên các đặc điểm cấu trúc của hợp chất.
- **Phân tích cụm (Clustering):** Phân nhóm các hợp chất hoặc protein dựa trên sự tương đồng về cấu trúc hoặc hoạt tính.
- **Phân tích thành phần chính (PCA):** Giảm số chiều của dữ liệu và khám phá các yếu tố quan trọng ảnh hưởng đến hoạt tính.
- **Phân tích mạng lưới (Network analysis):** Xây dựng mạng lưới tương tác giữa các protein và hợp chất để xác định các "hub" quan trọng.

2. Hướng dẫn song ngữ (Bilingual Guidance)

Tôi sẽ cung cấp hướng dẫn chi tiết bằng cả tiếng Anh và tiếng Việt để bạn dễ dàng thực hiện theo.

3. Code SQL và Python (SQL and Python Code)

Tôi sẽ cung cấp code SQL để truy vấn dữ liệu từ cơ sở dữ liệu Chembl 35 và code Python để thực hiện các bước phân tích và tiền xử lý dữ liệu.

4. Ví dụ code mẫu (Code Examples)

Tôi sẽ cung cấp 5 ví dụ code SQL và Python mẫu để bạn có thể bắt đầu dự án của mình một cách dễ dàng.

Giải quyết lỗi (Error Handling)

- **Lỗi SQL:** `ERROR: operator does not exist: numeric ~ unknown, LINE 12: AND act.standard_value ~ '^[0-9\.]+$'`

Lỗi này xảy ra do bạn đang cố gắng sử dụng toán tử ~ (regular expression match) trên một cột kiểu số (numeric). Để khắc phục, bạn cần chuyển đổi cột `standard_value` sang kiểu text trước khi so sánh với regular expression.

SQL (English):

```sql
SELECT *
FROM activities act
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND act.standard_value::text ~ '^[0-9\.]+$'
LIMIT 100;
```

SQL (Tiếng Việt):

```sql
SELECT *
FROM activities act
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;
```

- **Lỗi Python: `squared=False` trong hàm `mean_squared_error`**

  Lỗi này xảy ra do phiên bản scikit-learn bạn đang sử dụng quá cũ và không hỗ trợ tham số `squared=False`. Để khắc phục, bạn có thể nâng cấp scikit-learn lên phiên bản mới nhất hoặc tính căn bậc hai của kết quả `mean_squared_error` để có được Root Mean Squared Error (RMSE).

  Python (English):

```python
from sklearn.metrics import mean_squared_error
import numpy as np

# Option 1: Upgrade scikit-learn
# pip install -U scikit-learn

# Option 2: Calculate RMSE manually
y_true = [3, -0.5, 2, 7]
y_predicted = [2.5, 0.0, 2, 8]

mse = mean_squared_error(y_true, y_predicted)
rmse = np.sqrt(mse)

print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
```

  Python (Tiếng Việt):

```python
from sklearn.metrics import mean_squared_error
import numpy as np

# Cách 1: Nâng cấp scikit-learn
# pip install -U scikit-learn

# Cách 2: Tính RMSE thủ công
y_thuc_te = [3, -0.5, 2, 7]
y_du_doan = [2.5, 0.0, 2, 8]

mse = mean_squared_error(y_thuc_te, y_du_doan)
```

```
rmse = np.sqrt(mse)

print("Sai số bình phương trung bình (MSE):", mse)
print("Căn bậc hai của sai số bình phương trung bình (RMSE):", rmse)
```

Ví dụ code mẫu (Code Examples)

Dưới đây là 5 ví dụ code SQL và Python mẫu để bạn bắt đầu. Hãy nhớ rằng, những ví dụ này chỉ là điểm khởi đầu và bạn cần điều chỉnh chúng để phù hợp với mục tiêu cụ thể của bạn.

SQL Examples:

1. **Lấy thông tin về các hợp chất có hoạt tính IC50 dưới 100 nM:**

   SQL (English):

   ```
   SELECT md.chembl_id, act.standard_value, act.units
   FROM activities act
   JOIN molecule_dictionary md ON act.molregno = md.molregno
   WHERE act.standard_type = 'IC50'
     AND act.standard_relation = '='
     AND act.standard_value <= 100
     AND act.units = 'nM'
   LIMIT 100;
   ```

   SQL (Tiếng Việt):

   ```
   SELECT md.chembl_id, act.standard_value, act.units
   FROM activities act
   JOIN molecule_dictionary md ON act.molregno = md.molregno
   WHERE act.standard_type = 'IC50'
     AND act.standard_relation = '='
     AND act.standard_value <= 100
     AND act.units = 'nM'
   LIMIT 100;
   ```

2. **Lấy danh sách các protein (targets) liên quan đến một bệnh cụ thể (ví dụ: ung thư):**

   SQL (English):

   ```
   SELECT td.chembl_id, td.pref_name
   FROM target_dictionary td
   JOIN target_components tc ON td.tid = tc.tid
   JOIN component_sequences cs ON tc.component_id = cs.component_id
   WHERE td.target_type = 'SINGLE PROTEIN'
     AND td.organism = 'Homo sapiens'
     AND td.pref_name LIKE '%cancer%'
   LIMIT 100;
   ```

   SQL (Tiếng Việt):

   ```
   SELECT td.chembl_id, td.pref_name
   FROM target_dictionary td
   JOIN target_components tc ON td.tid = tc.tid
   JOIN component_sequences cs ON tc.component_id = cs.component_id
   WHERE td.target_type = 'SINGLE PROTEIN'
     AND td.organism = 'Homo sapiens'
     AND td.pref_name LIKE '%cancer%'
   LIMIT 100;
   ```

3. **Tìm kiếm các hợp chất có chứa một khung cấu trúc (scaffold) cụ thể:**

(Ví dụ này yêu cầu bạn biết SMILES của khung cấu trúc bạn quan tâm)

SQL (English):

```sql
-- This requires the 'chembl_id' of compounds containing a specific scaffold
-- Replace 'YOUR_SCAFFOLD_SMILES' with the actual SMILES string
SELECT md.chembl_id
FROM molecule_dictionary md
WHERE md.molecule_structures LIKE '%YOUR_SCAFFOLD_SMILES%'
LIMIT 100;
```

SQL (Tiếng Việt):

```sql
-- Điều này yêu cầu 'chembl_id' của các hợp chất chứa một khung cấu trúc cụ thể
-- Thay thế 'YOUR_SCAFFOLD_SMILES' bằng chuỗi SMILES thực tế
SELECT md.chembl_id
FROM molecule_dictionary md
WHERE md.molecule_structures LIKE '%YOUR_SCAFFOLD_SMILES%'
LIMIT 100;
```

4. **Lấy dữ liệu về các hoạt tính sinh học liên quan đến một protein cụ thể:**

   SQL (English):

```sql
SELECT act.standard_type, act.standard_value, act.units
FROM activities act
JOIN target_dictionary td ON act.tid = td.tid
WHERE td.chembl_id = 'CHEMBL203' -- Replace with the desired target CHEMBL_ID
LIMIT 100;
```

   SQL (Tiếng Việt):

```sql
SELECT act.standard_type, act.standard_value, act.units
FROM activities act
JOIN target_dictionary td ON act.tid = td.tid
WHERE td.chembl_id = 'CHEMBL203' -- Thay thế bằng CHEMBL_ID của target mong muốn
LIMIT 100;
```

5. **Thống kê số lượng hợp chất cho mỗi loại hoạt tính (IC50, Ki, EC50, etc.):**

   SQL (English):

```sql
SELECT act.standard_type, COUNT(*) AS compound_count
FROM activities act
GROUP BY act.standard_type
ORDER BY compound_count DESC
LIMIT 100;
```

   SQL (Tiếng Việt):

```sql
SELECT act.standard_type, COUNT(*) AS compound_count
FROM activities act
GROUP BY act.standard_type
ORDER BY compound_count DESC
LIMIT 100;
```

Python Examples:

1. **Đọc dữ liệu từ file CSV và hiển thị một vài dòng đầu tiên:**

   Python (English):

```python
import pandas as pd
import os

base_path = '../data'  # Adjust this to your actual base path
file_path = os.path.join(base_path, 'your_data.csv')  # Replace 'your_data.csv'

try:
    df = pd.read_csv(file_path)
    print(df.head())  # Show the first few rows
except FileNotFoundError:
    print(f"Error: File not found at {file_path}")
```

Python (Tiếng Việt):

```python
import pandas as pd
import os

base_path = '../data'  # Điều chỉnh đường dẫn gốc cho phù hợp
file_path = os.path.join(base_path, 'your_data.csv')  # Thay thế 'your_data.csv'

try:
    df = pd.read_csv(file_path)
    print(df.head())  # Hiển thị một vài dòng đầu tiên
except FileNotFoundError:
    print(f"Lỗi: Không tìm thấy file tại {file_path}")
```

2. **Tính toán các đặc tính lý hóa của hợp chất sử dụng RDKit:**

Python (English):

```python
from rdkit import Chem
from rdkit.Chem import Descriptors

smiles = 'CC(=O)Oc1ccccc1C(=O)O'  # Aspirin
mol = Chem.MolFromSmiles(smiles)

if mol is not None:
    mw = Descriptors.MolWt(mol)
    logp = Descriptors.MolLogP(mol)
    print(f"Molecular Weight: {mw}")
    print(f"LogP: {logp}")
else:
    print("Invalid SMILES string")
```

Python (Tiếng Việt):

```python
from rdkit import Chem
from rdkit.Chem import Descriptors

smiles = 'CC(=O)Oc1ccccc1C(=O)O'  # Aspirin
mol = Chem.MolFromSmiles(smiles)

if mol is not None:
    mw = Descriptors.MolWt(mol)
    logp = Descriptors.MolLogP(mol)
    print(f"Khối lượng phân tử: {mw}")
    print(f"LogP: {logp}")
else:
    print("Chuỗi SMILES không hợp lệ")
```

3. **Vẽ cấu trúc 2D của một hợp chất sử dụng RDKit:**

Python (English):

```python
from rdkit import Chem
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem import Draw

smiles = 'CC(=O)Oc1ccccc1C(=O)O'  # Aspirin
mol = Chem.MolFromSmiles(smiles)

if mol is not None:
    img = Draw.MolToImage(mol)
    img.save("aspirin.png") # save image
    img  # Display the image in the notebook
else:
    print("Invalid SMILES string")
```

Python (Tiếng Việt):

```python
from rdkit import Chem
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem import Draw

smiles = 'CC(=O)Oc1ccccc1C(=O)O'  # Aspirin
mol = Chem.MolFromSmiles(smiles)

if mol is not None:
    img = Draw.MolToImage(mol)
    img.save("aspirin.png") # Lưu ảnh
    img  # Hiển thị ảnh trong notebook
else:
    print("Chuỗi SMILES không hợp lệ")
```

4. **Tính toán fingerprint của hợp chất:**

Python (English):

```python
from rdkit import Chem
from rdkit.Chem import AllChem

smiles = 'CC(=O)Oc1ccccc1C(=O)O'  # Aspirin
mol = Chem.MolFromSmiles(smiles)

if mol is not None:
    fp = AllChem.GetMorganFingerprintAsBitVect(mol, 2, nBits=2048)
    print(fp.ToBitString())
else:
    print("Invalid SMILES string")
```

Python (Tiếng Việt):

```python
from rdkit import Chem
from rdkit.Chem import AllChem

smiles = 'CC(=O)Oc1ccccc1C(=O)O'  # Aspirin
mol = Chem.MolFromSmiles(smiles)

if mol is not None:
    fp = AllChem.GetMorganFingerprintAsBitVect(mol, 2, nBits=2048)
    print(fp.ToBitString())
else:
    print("Chuỗi SMILES không hợp lệ")
```

5. **Sử dụng Pandas để lọc dữ liệu dựa trên điều kiện:**

   Python (English):

```python
import pandas as pd
import os

base_path = '../data'  # Adjust this to your actual base path
file_path = os.path.join(base_path, 'your_data.csv')  # Replace 'your_data.csv'

try:
    df = pd.read_csv(file_path)
    # lọc các dòng có standard_value < 100
    filtered_df = df[df['standard_value'] < 100]
    print(filtered_df.head())
except FileNotFoundError:
    print(f"Error: File not found at {file_path}")
```

   Python (Tiếng Việt):

```python
import pandas as pd
import os

base_path = '../data'  # Điều chỉnh đường dẫn gốc cho phù hợp
file_path = os.path.join(base_path, 'your_data.csv')  # Thay thế 'your_data.csv'

try:
    df = pd.read_csv(file_path)
    # Lọc các dòng có standard_value < 100
    filtered_df = df[df['standard_value'] < 100]
    print(filtered_df.head())
except FileNotFoundError:
    print(f"Lỗi: Không tìm thấy file tại {file_path}")
```

Lưu ý quan trọng:

- Hãy thay thế các đường dẫn file, tên bảng, tên cột và các giá trị tham số trong các ví dụ trên bằng các giá trị phù hợp với dữ liệu và mục tiêu của bạn.
- Đảm bảo rằng bạn đã cài đặt đầy đủ các thư viện cần thiết (RDKit, Pandas, Scikit-learn) trước khi chạy code Python. Bạn có thể sử dụng `pip install rdkit pandas scikit-learn` để cài đặt.
- Khi làm việc với dữ liệu lớn, hãy cân nhắc sử dụng các kỹ thuật tối ưu hóa hiệu suất như sử dụng indexing trong SQL, sử dụng vectorized operations trong Pandas, và sử dụng các thuật toán học máy hiệu quả.

Để tiếp tục, vui lòng cho tôi biết mục tiêu cụ thể của bạn cho chủ đề "Topic_CheMBL_35_60". Với thông tin này, tôi có thể cung cấp hướng dẫn và code chi tiết hơn để giúp bạn đạt được kết quả tốt nhất.