PharmApp Suite

🧠 AI for Drug Discovery and Development 🍼

🧩 Topic: Biểu đồ nhiệt tương quan các tính chất lý hóa

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu ChEMBL 35 kết hợp RDKit theo chuẩn AIMLOps. Với vai trò là một chuyên gia R&D dược học, tôi sẽ cung cấp cho bạn hướng dẫn chi tiết, mã nguồn mẫu, và các ví dụ cụ thể để giúp bạn triển khai dự án một cách hiệu quả.

Tổng quan dự án:

Dự án của bạn tập trung vào việc khai thác dữ liệu từ cơ sở dữ liệu ChEMBL 35, sử dụng RDKit để tính toán các thuộc tính phân tử, và áp dụng các mô hình học máy để khám phá các mối quan hệ giữa cấu trúc hóa học và hoạt tính sinh học. Việc tuân thủ theo AIMLOps Template sẽ giúp bạn quản lý dự án một cách có cấu trúc, dễ dàng theo dõi, tái sử dụng và mở rộng.

Phân tích mô hình phân tích:

Mô hình phân tích của bạn có thể bao gồm các bước sau:

1. **Thu thập và tiền xử lý dữ liệu:**
    - Trích xuất dữ liệu từ cơ sở dữ liệu ChEMBL 35 (sử dụng SQL).
    - Làm sạch dữ liệu: loại bỏ các giá trị trùng lặp, xử lý các giá trị bị thiếu, và chuẩn hóa dữ liệu.
    - Lọc dữ liệu: lựa chọn các hợp chất và hoạt tính sinh học phù hợp với mục tiêu nghiên cứu.
2. **Tính toán thuộc tính phân tử:**
    - Sử dụng RDKit để tính toán các thuộc tính phân tử (descriptors) từ cấu trúc hóa học của các hợp chất. Các thuộc tính này có thể bao gồm: trọng lượng phân tử, logP, số lượng liên kết hydro, diện tích bề mặt phân tử, v.v.
3. **Phân tích khám phá dữ liệu (EDA):**
    - Thống kê mô tả dữ liệu.
    - Trực quan hóa dữ liệu để tìm kiếm các xu hướng và mối quan hệ tiềm năng.
4. **Xây dựng mô hình học máy:**
    - Lựa chọn mô hình phù hợp với mục tiêu nghiên cứu (ví dụ: hồi quy tuyến tính, random forest, mạng nơ-ron).
    - Huấn luyện mô hình trên dữ liệu đã được chuẩn bị.
    - Đánh giá hiệu năng của mô hình sử dụng các chỉ số phù hợp (ví dụ: R-squared, RMSE, AUC).
5. **Diễn giải kết quả và đưa ra kết luận:**
    - Phân tích các thuộc tính phân tử quan trọng nhất ảnh hưởng đến hoạt tính sinh học.
    - Đề xuất các hợp chất tiềm năng để thử nghiệm thêm.

Hướng dẫn song ngữ và mã nguồn mẫu:

Dưới đây là một số ví dụ về mã nguồn SQL và Python mà bạn có thể sử dụng trong dự án của mình.

1. Trích xuất dữ liệu từ ChEMBL (SQL):

```sql
-- English
-- Extract 100 data rows of compounds and their activities from ChEMBL 35

SELECT
```

```
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50' AND act.standard_units = 'nM'
  AND act.standard_value IS NOT NULL
  AND act.standard_value > 0
LIMIT 100;

-- Vietnamese
-- Trích xuất 100 dòng dữ liệu về hợp chất và hoạt tính của chúng từ ChEMBL 35

SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50' AND act.standard_units = 'nM'
  AND act.standard_value IS NOT NULL
  AND act.standard_value > 0
LIMIT 100;
```

Lưu ý:

- Đảm bảo rằng bạn đã kết nối thành công đến cơ sở dữ liệu PostgreSQL bằng pgAdmin.
- Lưu kết quả truy vấn vào file CSV (ví dụ: `../data/chembl_ic50_data.csv`).

Sửa lỗi SQL:

Lỗi `ERROR: operator does not exist: numeric ~ unknown, LINE 12: AND act.standard_value ~ '^[0-9\.]+$'` xảy ra do bạn đang cố gắng sử dụng toán tử `~` (regex match) trên một cột kiểu số (`numeric`). Để khắc phục, bạn có thể ép kiểu cột `standard_value` sang kiểu `text` trước khi so sánh:

```
-- English
-- Corrected SQL query

SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
```

```
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50' AND act.standard_units = 'nM'
  AND act.standard_value IS NOT NULL
  AND act.standard_value > 0
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;

-- Vietnamese
-- Câu truy vấn SQL đã sửa

SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50' AND act.standard_units = 'nM'
  AND act.standard_value IS NOT NULL
  AND act.standard_value > 0
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;
```

2. Đọc dữ liệu và tính toán thuộc tính phân tử (Python):

```
# English
# Read data from CSV and calculate molecular properties using RDKit
import os
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors

base_path = "../data"  # Adjust the base path if needed
csv_file = "chembl_ic50_data.csv"
file_path = os.path.join(base_path, csv_file)

# Read the CSV file into a pandas DataFrame
df = pd.read_csv(file_path)

# Function to calculate molecular weight using RDKit
def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
    else:
        return None

# Apply the function to the 'canonical_smiles' column to create a new
'molecular_weight' column
df['molecular_weight'] = df['canonical_smiles'].apply(calculate_mw)

# Display the first few rows of the DataFrame with the new 'molecular_weight' column
print(df.head())

# Vietnamese
```

```python
# Đọc dữ liệu từ CSV và tính toán thuộc tính phân tử bằng RDKit
import os
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors

base_path = "../data"  # Điều chỉnh đường dẫn gốc nếu cần
csv_file = "chembl_ic50_data.csv"
file_path = os.path.join(base_path, csv_file)

# Đọc file CSV vào một DataFrame của pandas
df = pd.read_csv(file_path)

# Hàm tính toán trọng lượng phân tử sử dụng RDKit
def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
    else:
        return None

# Áp dụng hàm vào cột 'canonical_smiles' để tạo cột 'molecular_weight' mới
df['molecular_weight'] = df['canonical_smiles'].apply(calculate_mw)

# Hiển thị một vài dòng đầu tiên của DataFrame với cột 'molecular_weight' mới
print(df.head())
```

Sửa lỗi Python (scikit-learn):

Nếu bạn gặp lỗi liên quan đến tham số `squared=False` trong hàm `mean_squared_error` của scikit-learn, bạn có thể giải quyết bằng một trong hai cách sau:

1. **Nâng cấp scikit-learn:** Phiên bản mới nhất của scikit-learn hỗ trợ tham số `squared=False`.
2. **Sử dụng `squared=True` và lấy căn bậc hai:** Nếu bạn không thể nâng cấp scikit-learn, bạn có thể sử dụng `squared=True` (giá trị mặc định) và sau đó lấy căn bậc hai của kết quả để có được RMSE (Root Mean Squared Error).

```python
# English
# Example of calculating RMSE with older scikit-learn version
from sklearn.metrics import mean_squared_error
import numpy as np

# Assuming you have y_true and y_predicted
mse = mean_squared_error(y_true, y_predicted, squared=True)
rmse = np.sqrt(mse)
print("RMSE:", rmse)

# Vietnamese
# Ví dụ tính toán RMSE với phiên bản scikit-learn cũ
from sklearn.metrics import mean_squared_error
import numpy as np

# Giả sử bạn có y_true và y_predicted
mse = mean_squared_error(y_true, y_predicted, squared=True)
rmse = np.sqrt(mse)
print("RMSE:", rmse)
```

5 ví dụ code SQL và Python mẫu:

Dưới đây là 5 ví dụ khác nhau về cách bạn có thể sử dụng SQL và Python để phân tích dữ liệu ChEMBL 35:

Ví dụ 1: Tính số lượng hợp chất cho mỗi loại hoạt tính (SQL)

```sql
-- English
-- Count the number of compounds for each activity type

SELECT standard_type, COUNT(*) AS compound_count
FROM activities
GROUP BY standard_type
ORDER BY compound_count DESC
LIMIT 10;

-- Vietnamese
-- Đếm số lượng hợp chất cho mỗi loại hoạt tính

SELECT standard_type, COUNT(*) AS compound_count
FROM activities
GROUP BY standard_type
ORDER BY compound_count DESC
LIMIT 10;
```

Ví dụ 2: Lọc các hợp chất có trọng lượng phân tử nằm trong khoảng nhất định (Python)

```python
# English
# Filter compounds with molecular weight within a specific range

import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors

def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
    else:
        return None

# Sample DataFrame (replace with your actual DataFrame)
data = {'chembl_id': ['CHEMBL1', 'CHEMBL2', 'CHEMBL3'],
        'canonical_smiles': ['CC(=O)Oc1ccccc1C(=O)O', 'c1ccccc1', 'C[C@H]
(O)c1ccccc1']}
df = pd.DataFrame(data)

df['molecular_weight'] = df['canonical_smiles'].apply(calculate_mw)

# Filter compounds with molecular weight between 200 and 400
filtered_df = df[(df['molecular_weight'] >= 200) & (df['molecular_weight'] <= 400)]

print(filtered_df)

# Vietnamese
# Lọc các hợp chất có trọng lượng phân tử nằm trong khoảng nhất định

import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors

def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
```

```
    else:
        return None

# DataFrame mẫu (thay thế bằng DataFrame thực tế của bạn)
data = {'chembl_id': ['CHEMBL1', 'CHEMBL2', 'CHEMBL3'],
        'canonical_smiles': ['CC(=O)Oc1ccccc1C(=O)O', 'c1ccccc1', 'C[C@H]
(O)c1ccccc1']}
df = pd.DataFrame(data)

df['molecular_weight'] = df['canonical_smiles'].apply(calculate_mw)

# Lọc các hợp chất có trọng lượng phân tử từ 200 đến 400
filtered_df = df[(df['molecular_weight'] >= 200) & (df['molecular_weight'] <= 400)]

print(filtered_df)
```

Ví dụ 3: Tìm các hợp chất có hoạt tính IC50 thấp nhất (SQL)

```
-- English
-- Find compounds with the lowest IC50 values

SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_value
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50' AND act.standard_units = 'nM'
ORDER BY act.standard_value ASC
LIMIT 10;

-- Vietnamese
-- Tìm các hợp chất có giá trị IC50 thấp nhất

SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_value
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50' AND act.standard_units = 'nM'
ORDER BY act.standard_value ASC
LIMIT 10;
```

Ví dụ 4: Tính logP cho các hợp chất (Python)

```
# English
# Calculate LogP for compounds using RDKit

import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors
```

```python
def calculate_logp(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolLogP(mol)
    else:
        return None

# Sample DataFrame (replace with your actual DataFrame)
data = {'chembl_id': ['CHEMBL1', 'CHEMBL2', 'CHEMBL3'],
        'canonical_smiles': ['CC(=O)Oc1ccccc1C(=O)O', 'c1ccccc1', 'C[C@H]
(O)c1ccccc1']}
df = pd.DataFrame(data)

df['logp'] = df['canonical_smiles'].apply(calculate_logp)

print(df)

# Vietnamese
# Tính LogP cho các hợp chất sử dụng RDKit

import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors

def calculate_logp(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolLogP(mol)
    else:
        return None

# DataFrame mẫu (thay thế bằng DataFrame thực tế của bạn)
data = {'chembl_id': ['CHEMBL1', 'CHEMBL2', 'CHEMBL3'],
        'canonical_smiles': ['CC(=O)Oc1ccccc1C(=O)O', 'c1ccccc1', 'C[C@H]
(O)c1ccccc1']}
df = pd.DataFrame(data)

df['logp'] = df['canonical_smiles'].apply(calculate_logp)

print(df)
```

Ví dụ 5: Kết hợp dữ liệu từ nhiều bảng (SQL)

```sql
-- English
-- Combine data from molecule_dictionary, compound_structures, and activities tables

SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type IN ('IC50', 'Ki') AND act.standard_units = 'nM'
```

```
LIMIT 100;

-- Vietnamese
-- Kết hợp dữ liệu từ các bảng molecule_dictionary, compound_structures, và activities

SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type IN ('IC50', 'Ki') AND act.standard_units = 'nM'
LIMIT 100;
```

Lưu ý quan trọng:

- **Đường dẫn:** Hãy chắc chắn rằng bạn đã điều chỉnh đường dẫn `base_path` cho phù hợp với cấu trúc thư mục AIMLOps Template của bạn.
- **Mã chủ đề và tên file:** Tuân thủ nghiêm ngặt quy ước đặt tên file mà bạn đã đề ra (ví dụ: `Topic_CheMBL_35_77_1_data_extraction.ipynb`).
- **Phạm vi dữ liệu:** Bắt đầu với một tập dữ liệu nhỏ (100 dòng) để thử nghiệm và gỡ lỗi. Sau khi bạn đã xác minh rằng mọi thứ hoạt động tốt, bạn có thể tăng kích thước tập dữ liệu.
- **Hiệu năng:** Tính toán thuộc tính phân tử có thể tốn thời gian đối với các tập dữ liệu lớn. Cân nhắc sử dụng các kỹ thuật tối ưu hóa như vector hóa hoặc song song hóa để cải thiện hiệu năng.

Chúc bạn thành công với dự án của mình! Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi.