# PharmApp Suite

## 🧠 AI for Drug Discovery and Development 🧪

Copyright 2025 | Nghiên Cứu Thuốc | www.nghiencuuthuoc.com | Zalo: +84888999311 |

## 🧩 Topic: Dữ liệu tín hiệu cảnh báo dược lý

Tuyệt vời! Với vai trò là một chuyên gia nghiên cứu và phát triển dược học, tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu ChEMBL 35 sử dụng RDKit. Dưới đây là phân tích, hướng dẫn và code mẫu, tập trung vào việc giải quyết các vấn đề bạn đã nêu và tuân thủ theo quy trình làm việc bạn đã thiết lập.

**1. Phân tích mô hình phân tích (Analysis of the Analysis Model)**

Dựa trên thông tin bạn cung cấp, mô hình phân tích của bạn có vẻ như tập trung vào việc trích xuất, xử lý và phân tích dữ liệu từ cơ sở dữ liệu ChEMBL 35 để hỗ trợ nghiên cứu và phát triển thuốc. Cụ thể, bạn đang sử dụng PostgreSQL (psql) để truy vấn dữ liệu, RDKit để xử lý thông tin hóa học, và Jupyter Notebook để viết và chạy code phân tích.

* **SQL (Structured Query Language):** Được sử dụng để truy vấn và trích xuất dữ liệu từ cơ sở dữ liệu ChEMBL 35. Các truy vấn SQL sẽ được viết để lọc dữ liệu liên quan đến các hoạt tính sinh học, cấu trúc hóa học, và các thông tin khác cần thiết cho nghiên cứu.
* **RDKit:** Một thư viện mã nguồn mở cho tin học hóa học và tin sinh học. RDKit sẽ được sử dụng để xử lý cấu trúc hóa học (ví dụ: chuyển đổi SMILES thành fingerprint), tính toán các thuộc tính hóa lý, và thực hiện các phân tích cấu trúc.
* **Jupyter Notebook:** Một môi trường tương tác cho phép bạn viết và chạy code Python, hiển thị kết quả, và ghi lại quá trình phân tích. Jupyter Notebook rất phù hợp cho việc khám phá dữ liệu, phát triển mô hình, và trình bày kết quả nghiên cứu.

**2. Hướng dẫn song ngữ (Bilingual Guidance)**

**2.1. Vấn đề và giải pháp (Problems and Solutions)**

* **Lỗi SQL: ERROR: operator does not exist: numeric ~ unknown LINE 12: AND act.standard\_value ~ '^[0-9\.]+$'**
  + **Tiếng Việt:** Lỗi này xảy ra do bạn đang cố gắng sử dụng toán tử ~ (regex match) trên một cột kiểu số (numeric). PostgreSQL không hỗ trợ toán tử này cho kiểu số.
  + **Tiếng Anh:** This error occurs because you are trying to use the ~ (regex match) operator on a numeric column (numeric). PostgreSQL does not support this operator for numeric types.
  + **Giải pháp:** Chuyển đổi cột standard\_value sang kiểu text trước khi sử dụng toán tử ~, hoặc sử dụng các hàm số học để lọc dữ liệu.
    - **Tiếng Việt:** Chuyển đổi kiểu dữ liệu bằng CAST(act.standard\_value AS TEXT)
    - **Tiếng Anh:** Cast the data type using CAST(act.standard\_value AS TEXT)
* **Lỗi Python: phiên bản scikit-learn cũ không hỗ trợ tham số squared=False trong hàm mean\_squared\_error**
  + **Tiếng Việt:** Phiên bản scikit-learn bạn đang sử dụng quá cũ và không hỗ trợ tham số squared=False trong hàm mean\_squared\_error. Tham số này được thêm vào để trả về RMSE (Root Mean Squared Error) trực tiếp.
  + **Tiếng Anh:** The scikit-learn version you are using is too old and does not support the squared=False parameter in the mean\_squared\_error function. This parameter was added to directly return RMSE (Root Mean Squared Error).
  + **Giải pháp:**
    1. Nâng cấp scikit-learn lên phiên bản mới nhất: pip install -U scikit-learn
    2. Nếu không thể nâng cấp, tính RMSE thủ công: rmse = np.sqrt(mean\_squared\_error(y\_true, y\_pred))
       - **Tiếng Việt:** Sử dụng np.sqrt từ thư viện NumPy để tính căn bậc hai của MSE.
       - **Tiếng Anh:** Use np.sqrt from the NumPy library to calculate the square root of MSE.

**2.2. Cấu trúc thư mục (Folder Structure)**

Bạn đã sử dụng cấu trúc thư mục theo chuẩn AIMLOps Template. Điều này rất tốt vì nó giúp bạn tổ chức code, dữ liệu, và các thành phần khác của dự án một cách có hệ thống.

**2.3. Quy trình làm việc (Workflow)**

1. **SQL trên pgAdmin:** Viết và chạy các truy vấn SQL trên pgAdmin để trích xuất dữ liệu từ cơ sở dữ liệu ChEMBL 35. Lưu kết quả vào các file CSV trong thư mục ../data/.
2. **Jupyter Notebook:** Sử dụng Jupyter Notebook để đọc dữ liệu từ các file CSV, xử lý dữ liệu bằng RDKit, xây dựng mô hình, và phân tích kết quả.
3. **Tên file Notebook:** Đặt tên file Notebook theo định dạng "Topic\_CheMBL\_35\_57\_1\_\*" và "Topic\_CheMBL\_35\_57\_2\_\*".
4. **Đường dẫn:** Sử dụng os.path.join(base\_path, ...) để kết nối các đường dẫn một cách an toàn và linh hoạt.

**3. Code mẫu (Code Examples)**

**3.1. SQL (English & Vietnamese Comments)**

-- English: This query extracts 100 compounds with IC50 values for a specific target.  
-- Vietnamese: Truy vấn này trích xuất 100 hợp chất với giá trị IC50 cho một mục tiêu cụ thể.  
SELECT  
 md.molregno, -- English: Molecule registry number, Vietnamese: Số đăng ký phân tử  
 cs.canonical\_smiles, -- English: Canonical SMILES string, Vietnamese: Chuỗi SMILES chuẩn tắc  
 act.standard\_value, -- English: Standard value of the activity, Vietnamese: Giá trị tiêu chuẩn của hoạt tính  
 act.standard\_units -- English: Standard units of the activity, Vietnamese: Đơn vị tiêu chuẩn của hoạt tính  
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' -- English: Filter for IC50 values, Vietnamese: Lọc theo giá trị IC50  
 AND act.standard\_relation = '=' -- English: Filter for exact IC50 values, Vietnamese: Lọc theo giá trị IC50 chính xác  
 AND act.standard\_value IS NOT NULL -- English: Exclude null values, Vietnamese: Loại bỏ giá trị null  
 AND CAST(act.standard\_value AS TEXT) ~ '^[0-9\.]+$' -- English: Filter for numeric values, Vietnamese: Lọc theo giá trị số  
LIMIT 100;

**3.2. Python (English & Vietnamese Comments)**

# English: Import necessary libraries  
# Vietnamese: Nhập các thư viện cần thiết  
import pandas as pd  
import numpy as np  
import os  
from rdkit import Chem  
from rdkit.Chem import AllChem  
from sklearn.model\_selection import train\_test\_split  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.metrics import mean\_squared\_error, r2\_score  
  
# English: Define the base path for the project  
# Vietnamese: Định nghĩa đường dẫn gốc cho dự án  
base\_path = "path/to/your/project" # Replace with your actual path  
  
# English: Load the data from the CSV file  
# Vietnamese: Tải dữ liệu từ file CSV  
data\_path = os.path.join(base\_path, "data", "your\_data.csv") # Replace with your actual file name  
df = pd.read\_csv(data\_path)  
  
# English: Function to convert SMILES to Morgan Fingerprints  
# Vietnamese: Hàm chuyển đổi SMILES thành Morgan Fingerprints  
def smiles\_to\_fingerprint(smiles, radius=2, nBits=2048):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is not None:  
 fp = AllChem.GetMorganFingerprintAsBitVect(mol, radius, nBits=nBits)  
 return np.array(fp)  
 else:  
 return None  
  
# English: Apply the function to the SMILES column  
# Vietnamese: Áp dụng hàm cho cột SMILES  
df['fingerprint'] = df['canonical\_smiles'].apply(smiles\_to\_fingerprint)  
  
# English: Drop rows with missing fingerprints  
# Vietnamese: Loại bỏ các hàng có fingerprint bị thiếu  
df = df.dropna(subset=['fingerprint'])  
  
# English: Convert IC50 values to pIC50  
# Vietnamese: Chuyển đổi giá trị IC50 sang pIC50  
df['pIC50'] = -np.log10(df['standard\_value'] / 1e9)  
  
# English: Prepare the data for machine learning  
# Vietnamese: Chuẩn bị dữ liệu cho máy học  
X = np.array(list(df['fingerprint']))  
y = df['pIC50']  
  
# English: Split the data into training and testing sets  
# Vietnamese: Chia dữ liệu thành tập huấn luyện và tập kiểm tra  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# English: Train a Random Forest Regressor model  
# Vietnamese: Huấn luyện mô hình Random Forest Regressor  
model = RandomForestRegressor(n\_estimators=100, random\_state=42)  
model.fit(X\_train, y\_train)  
  
# English: Make predictions on the test set  
# Vietnamese: Dự đoán trên tập kiểm tra  
y\_pred = model.predict(X\_test)  
  
# English: Evaluate the model  
# Vietnamese: Đánh giá mô hình  
mse = mean\_squared\_error(y\_test, y\_pred)  
rmse = np.sqrt(mse) # Calculate RMSE manually if necessary  
r2 = r2\_score(y\_test, y\_pred)  
  
print(f"Mean Squared Error: {mse}")  
print(f"Root Mean Squared Error: {rmse}")  
print(f"R-squared: {r2}")

**4. Ví dụ code mẫu (Example Code Snippets)**

Dưới đây là 5 ví dụ code SQL và Python mẫu, tập trung vào các tác vụ phổ biến trong phân tích dữ liệu ChEMBL.

**4.1. SQL: Lọc theo khoảng giá trị IC50 (Filtering by IC50 Range)**

-- English: Select compounds with IC50 values between 100 and 1000 nM  
-- Vietnamese: Chọn các hợp chất có giá trị IC50 nằm trong khoảng từ 100 đến 1000 nM  
SELECT  
 md.molregno,  
 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\_value >= 100  
 AND act.standard\_value <= 1000  
LIMIT 100;

**4.2. Python: Tính toán các thuộc tính hóa lý (Calculating Physicochemical Properties)**

# English: Calculate LogP and Molecular Weight using RDKit  
# Vietnamese: Tính LogP và Trọng lượng phân tử sử dụng RDKit  
from rdkit.Chem import Descriptors  
  
def calculate\_properties(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is not None:  
 logp = Descriptors.MolLogP(mol)  
 mw = Descriptors.MolWt(mol)  
 return logp, mw  
 else:  
 return None, None  
  
df[['logp', 'mw']] = df['canonical\_smiles'].apply(lambda x: pd.Series(calculate\_properties(x)))

**4.3. SQL: Tìm kiếm các hợp chất tương tự (Finding Similar Compounds)**

(Ví dụ này yêu cầu tiện ích mở rộng pg\_trgm cho PostgreSQL để so sánh chuỗi)

-- English: Find compounds with similar SMILES strings  
-- Vietnamese: Tìm các hợp chất có chuỗi SMILES tương tự  
CREATE EXTENSION IF NOT EXISTS pg\_trgm;  
  
SELECT  
 md.molregno,  
 cs.canonical\_smiles  
FROM  
 molecule\_dictionary md  
JOIN  
 compound\_structures cs ON md.molregno = cs.molregno  
WHERE  
 cs.canonical\_smiles % 'CCO' -- English: Find compounds similar to ethanol, Vietnamese: Tìm các hợp chất tương tự ethanol  
LIMIT 100;

**4.4. Python: Trực quan hóa dữ liệu (Data Visualization)**

# English: Create a scatter plot of LogP vs. pIC50  
# Vietnamese: Tạo biểu đồ phân tán của LogP so với pIC50  
import matplotlib.pyplot as plt  
  
plt.scatter(df['logp'], df['pIC50'])  
plt.xlabel('LogP')  
plt.ylabel('pIC50')  
plt.title('LogP vs. pIC50')  
plt.show()

**4.5. SQL: Thống kê số lượng hoạt chất theo loại (Counting Activities by Type)**

-- English: Count the number of activities for each standard type  
-- Vietnamese: Đếm số lượng hoạt động cho mỗi loại tiêu chuẩn  
SELECT  
 act.standard\_type,  
 COUNT(\*)  
FROM  
 activities act  
GROUP BY  
 act.standard\_type;

Đây chỉ là một vài ví dụ để bạn bắt đầu. Chúc bạn thành công với dự án của mình! Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi.