PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 Topic: Mối liên hệ giữa target và bệnh học

Tuyệt vời! Với vai trò là chuyên gia R&D dược học, tôi rất sẵn lòng hỗ trợ bạn trong dự án phân tích dữ liệu ChEMBL 35 sử dụng RDKit. Dưới đây là phân tích chi tiết, hướng dẫn song ngữ, code mẫu SQL và Python, cùng với các ví dụ, dựa trên các yêu cầu và thông tin bạn cung cấp.

Tổng quan dự án (Project Overview)

Dự án của bạn tập trung vào việc khai thác dữ liệu từ cơ sở dữ liệu ChEMBL 35 để hỗ trợ quá trình nghiên cứu và phát triển thuốc. Bạn sẽ sử dụng PostgreSQL (với thông tin kết nối bạn đã cung cấp), RDKit (một thư viện Python mạnh mẽ cho tin học hóa học), và tuân theo cấu trúc thư mục chuẩn AIMLOps.

1. Phân tích và Mô hình phân tích (Analysis and Analysis Model)

1.1. Mục tiêu (Objectives):

- **Trích xuất dữ liệu liên quan:** Lấy thông tin về các hợp chất hóa học và hoạt tính sinh học của chúng từ ChEMBL 35.
- **Tiền xử lý dữ liệu:** Làm sạch và chuyển đổi dữ liệu để phù hợp với các thuật toán học máy.
- **Phân tích hoạt tính (Activity Analysis):** Xác định các yếu tố cấu trúc liên quan đến hoạt tính sinh học.
- **Xây dựng mô hình dự đoán (Predictive Model Building):** Tạo ra các mô hình có khả năng dự đoán hoạt tính của các hợp chất mới.

1.2. Mô hình phân tích (Analysis Model):

Chúng ta có thể sử dụng một số mô hình sau, tùy thuộc vào mục tiêu cụ thể:

- **Phân tích tương quan (Correlation Analysis):** Tìm mối liên hệ giữa các thuộc tính hóa học (ví dụ: trọng lượng phân tử, logP) và hoạt tính sinh học.
- **Hồi quy (Regression):** Xây dựng mô hình dự đoán giá trị hoạt tính liên tục (ví dụ: IC50) dựa trên các thuộc tính hóa học. Các thuật toán có thể sử dụng bao gồm hồi quy tuyến tính, hồi quy đa thức, Support Vector Regression (SVR), và Random Forest Regression.
- **Phân loại (Classification):** Xây dựng mô hình dự đoán một hợp chất có hoạt tính hay không (dựa trên ngưỡng hoạt tính). Các thuật toán có thể sử dụng bao gồm Logistic Regression, Support Vector Machines (SVM), và Random Forest.
- **Phân cụm (Clustering):** Phân nhóm các hợp chất dựa trên tính tương đồng về cấu trúc hoặc hoạt tính.

1.3. Kế hoạch thực hiện (Execution Plan):

1. **Trích xuất dữ liệu từ ChEMBL 35 (Data Extraction from ChEMBL 35):** Sử dụng SQL để lấy dữ liệu cần thiết và lưu vào file CSV.
2. **Tiền xử lý dữ liệu (Data Preprocessing):**
    - Đọc dữ liệu từ file CSV bằng Python và Pandas.
    - Sử dụng RDKit để tạo ra các descriptor hóa học (ví dụ: fingerprints, descriptors phân tử).
    - Xử lý các giá trị thiếu và chuẩn hóa dữ liệu.
3. **Phân tích và mô hình hóa (Analysis and Modeling):**
    - Chọn các thuộc tính phù hợp cho mô hình.

- Chia dữ liệu thành tập huấn luyện và tập kiểm tra.
- Huấn luyện mô hình trên tập huấn luyện.
- Đánh giá mô hình trên tập kiểm tra.

4. **Đánh giá và cải tiến (Evaluation and Improvement):**
   - Đánh giá hiệu suất của mô hình bằng các độ đo phù hợp (ví dụ: R-squared, RMSE, AUC).
   - Tinh chỉnh mô hình để cải thiện hiệu suất.

2. Hướng dẫn song ngữ (Bilingual Instructions)

2.1. Kết nối đến cơ sở dữ liệu ChEMBL 35 (Connecting to the ChEMBL 35 Database)

- **English:** Use the following credentials to connect to the ChEMBL 35 database:
  - IP Address: 192.168.206.136
  - User: rd
  - Password: rd
  - Database: chembl_35
- **Tiếng Việt:** Sử dụng thông tin sau để kết nối đến cơ sở dữ liệu ChEMBL 35:
  - Địa chỉ IP: 192.168.206.136
  - Người dùng: rd
  - Mật khẩu: rd
  - Cơ sở dữ liệu: chembl_35

2.2. Tạo descriptor hóa học bằng RDKit (Generating Chemical Descriptors with RDKit)

- **English:** Use RDKit to generate chemical descriptors from SMILES strings. These descriptors will be used as features in your machine learning models.
- **Tiếng Việt:** Sử dụng RDKit để tạo ra các descriptor hóa học từ chuỗi SMILES. Các descriptor này sẽ được sử dụng làm đặc trưng trong các mô hình học máy của bạn.

2.3. Xử lý lỗi (Error Handling)

- **English:**
  - **Error:** `ERROR: operator does not exist: numeric ~ unknown, LINE 12: AND act.standard_value ~ '^[0-9\.]+$'`
    - **Solution:** This error occurs because the `~` operator is used for regular expression matching, but the `standard_value` column might not be of text type. You can cast the column to text using `CAST(act.standard_value AS TEXT)`.
  - **Error:** `scikit-learn version is old and does not support squared=False in mean_squared_error`
    - **Solution:** Update scikit-learn to the latest version using `pip install --upgrade scikit-learn`. If you cannot update, remove the `squared=False` argument.
- **Tiếng Việt:**
  - **Lỗi:** `ERROR: operator does not exist: numeric ~ unknown, LINE 12: AND act.standard_value ~ '^[0-9\.]+$'`
    - **Giải pháp:** Lỗi này xảy ra do toán tử `~` được sử dụng để so khớp biểu thức chính quy, nhưng cột `standard_value` có thể không phải là kiểu văn bản. Bạn có thể chuyển đổi cột thành kiểu văn bản bằng cách sử dụng `CAST(act.standard_value AS TEXT)`.
  - **Lỗi:** `phiên bản scikit-learn cũ không hỗ trợ tham số squared=False trong hàm mean_squared_error`
    - **Giải pháp:** Cập nhật scikit-learn lên phiên bản mới nhất bằng cách sử dụng `pip install --upgrade scikit-learn`. Nếu bạn không thể cập nhật, hãy xóa đối số `squared=False`.

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

3.1. SQL (Trích xuất dữ liệu - Data Extraction)

```sql
-- English: Extracting 100 rows of data for a specific target (e.g., CHEMBL205)
-- Tiếng Việt: Trích xuất 100 dòng dữ liệu cho một mục tiêu cụ thể (ví dụ: CHEMBL205)
SELECT
    cmp.chembl_id,
    cmp.pref_name,
    act.standard_type,
    act.standard_value,
    act.standard_units,
    mol.molfile
FROM
    compound_structures AS mol
JOIN
    activities AS act ON mol.molregno = act.molregno
JOIN
    target_dictionary AS tgt ON act.tid = tgt.tid
JOIN
    component_sequences AS cs ON tgt.component_id = cs.component_id
JOIN
    chembl_id_lookup AS cmp ON mol.molregno = cmp.molregno
WHERE
    tgt.chembl_id = 'CHEMBL205' -- Replace with your target of interest / Thay thế
bằng mục tiêu bạn quan tâm
    AND act.standard_type = 'IC50'
    AND act.standard_units = 'nM'
    AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$' --Fix for the error
LIMIT 100;

--English: save the result to ../data/chembl_205_ic50_100.csv
--Tiếng Việt: Lưu kết quả vào ../data/chembl_205_ic50_100.csv
```

3.2. Python (Tiền xử lý và phân tích - Preprocessing and Analysis)

```python
# English: Import necessary libraries
# Tiếng Việt: Nhập các thư viện cần thiết
import pandas as pd
from rdkit import Chem
from rdkit.Chem import AllChem
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import os

# English: Define the base path for the project
# Tiếng Việt: Định nghĩa đường dẫn gốc cho dự án
base_path = "." # Assuming the notebook is in the root directory

# English: Load the data from the CSV file
# Tiếng Việt: Tải dữ liệu từ file CSV
data_path = os.path.join(base_path, "data", "chembl_205_ic50_100.csv") #Correct file
name
df = pd.read_csv(data_path)

# English: Convert SMILES to RDKit Mol objects
# Tiếng Việt: Chuyển đổi chuỗi SMILES thành đối tượng Mol của RDKit
df['mol'] = df['molfile'].apply(lambda x: Chem.MolFromMolBlock(x))
df = df.dropna(subset=['mol'])
```

```python
# English: Generate Morgan fingerprints
# Tiếng Việt: Tạo fingerprint Morgan
df['fingerprint'] = df['mol'].apply(lambda x: AllChem.GetMorganFingerprintAsBitVect(x,
2, nBits=2048))

# English: Convert fingerprints to numpy arrays
# Tiếng Việt: Chuyển đổi fingerprint thành mảng numpy
df['fingerprint_array'] = df['fingerprint'].apply(lambda x: np.array(list(x)))

# English: Prepare data for the model
# Tiếng Việt: Chuẩn bị dữ liệu cho mô hình
X = np.stack(df['fingerprint_array'].values)
y = df['standard_value'].astype(float).values

# English: Split data into training and testing sets
# Tiếng Việt: 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 linear regression model
# Tiếng Việt: Huấn luyện mô hình hồi quy tuyến tính
model = LinearRegression()
model.fit(X_train, y_train)

# English: Make predictions on the test set
# Tiếng Việt: Dự đoán trên tập kiểm tra
y_pred = model.predict(X_test)

# English: Evaluate the model
# Tiếng Việt: Đánh giá mô hình
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

4. Ví dụ code SQL và Python mẫu (Sample SQL and Python Code Examples)

Dưới đây là 5 ví dụ code SQL và Python mẫu, minh họa các tác vụ khác nhau trong dự án của bạn.

Ví dụ 1: Trích xuất dữ liệu cơ bản (Basic Data Extraction)

SQL:

```sql
-- English: Extract CHEMBL_ID, preferred name, and molecular weight for the first 100
compounds
-- Tiếng Việt: Trích xuất CHEMBL_ID, tên ưu tiên và trọng lượng phân tử cho 100 hợp
chất đầu tiên
SELECT
    cmp.chembl_id,
    cmp.pref_name,
    mol.mol_weight
FROM
    compound_structures AS mol
JOIN
    chembl_id_lookup AS cmp ON mol.molregno = cmp.molregno
LIMIT 100;
```

Python:

```python
# English: Load data from CSV and display the first 5 rows
# Tiếng Việt: Tải dữ liệu từ CSV và hiển thị 5 dòng đầu tiên
import pandas as pd
import os

base_path = "."
data_path = os.path.join(base_path, "data", "basic_compounds.csv") # Assuming you saved the SQL result to this file
df = pd.read_csv(data_path)
print(df.head())
```

Ví dụ 2: Lọc dữ liệu theo hoạt tính (Filtering by Activity)

SQL:

```sql
-- English: Extract data for compounds with IC50 values less than 100 nM for target CHEMBL205
-- Tiếng Việt: Trích xuất dữ liệu cho các hợp chất có giá trị IC50 nhỏ hơn 100 nM cho mục tiêu CHEMBL205
SELECT
    cmp.chembl_id,
    act.standard_value
FROM
    compound_structures AS mol
JOIN
    activities AS act ON mol.molregno = act.molregno
JOIN
    target_dictionary AS tgt ON act.tid = tgt.tid
JOIN
    chembl_id_lookup AS cmp ON mol.molregno = cmp.molregno
WHERE
    tgt.chembl_id = 'CHEMBL205'
    AND act.standard_type = 'IC50'
    AND act.standard_units = 'nM'
    AND act.standard_value < 100
    AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;
```

Python:

```python
# English: Convert IC50 values to pIC50 values
# Tiếng Việt: Chuyển đổi giá trị IC50 thành giá trị pIC50
import pandas as pd
import numpy as np
import os

base_path = "."
data_path = os.path.join(base_path, "data", "ic50_compounds.csv") # Assuming you saved the SQL result to this file
df = pd.read_csv(data_path)

df['pIC50'] = -np.log10(df['standard_value'] / 1e9)  # Convert nM to M and then to pIC50
print(df.head())
```

Ví dụ 3: Tạo descriptor hóa học (Generating Chemical Descriptors)

Python:

```python
# English: Generate molecular weight and logP descriptors using RDKit
# Tiếng Việt: Tạo descriptor trọng lượng phân tử và logP sử dụng RDKit
import pandas as pd
```

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

base_path = "."
data_path = os.path.join(base_path, "data", "basic_compounds.csv") # Assuming you
saved the SQL result to this file
df = pd.read_csv(data_path)

df['mol'] = df['molfile'].apply(lambda x: Chem.MolFromMolBlock(x))
df = df.dropna(subset=['mol'])

df['mol_weight'] = df['mol'].apply(lambda x: Descriptors.MolWt(x))
df['logP'] = df['mol'].apply(lambda x: Descriptors.MolLogP(x))

print(df[['chembl_id', 'mol_weight', 'logP']].head())
```

Ví dụ 4: Huấn luyện mô hình hồi quy (Training a Regression Model)

Python:

```python
# English: Train a linear regression model to predict IC50 values
# Tiếng Việt: Huấn luyện mô hình hồi quy tuyến tính để dự đoán giá trị IC50
import pandas as pd
import numpy as np
from rdkit import Chem
from rdkit.Chem import AllChem
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import os

base_path = "."
data_path = os.path.join(base_path, "data", "ic50_compounds.csv") # Assuming you saved
the SQL result to this file
df = pd.read_csv(data_path)

df['mol'] = df['molfile'].apply(lambda x: Chem.MolFromMolBlock(x))
df = df.dropna(subset=['mol'])
df['fingerprint'] = df['mol'].apply(lambda x: AllChem.GetMorganFingerprintAsBitVect(x,
2, nBits=2048))
df['fingerprint_array'] = df['fingerprint'].apply(lambda x: np.array(list(x)))

X = np.stack(df['fingerprint_array'].values)
y = df['standard_value'].astype(float).values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

Ví dụ 5: Phân tích tương quan (Correlation Analysis)

Python:

```python
# English: Calculate the correlation between molecular weight, logP, and pIC50
# Tiếng Việt: Tính toán tương quan giữa trọng lượng phân tử, logP và pIC50
import pandas as pd
import numpy as np
from rdkit import Chem
from rdkit.Chem import Descriptors
import os

base_path = "."
data_path = os.path.join(base_path, "data", "ic50_compounds.csv") # Assuming you saved the SQL result to this file
df = pd.read_csv(data_path)

df['mol'] = df['molfile'].apply(lambda x: Chem.MolFromMolBlock(x))
df = df.dropna(subset=['mol'])
df['mol_weight'] = df['mol'].apply(lambda x: Descriptors.MolWt(x))
df['logP'] = df['mol'].apply(lambda x: Descriptors.MolLogP(x))
df['pIC50'] = -np.log10(df['standard_value'] / 1e9)

correlation_matrix = df[['mol_weight', 'logP', 'pIC50']].corr()
print(correlation_matrix)
```

Lưu ý quan trọng (Important Notes):

- **Đường dẫn (Paths):** Hãy chắc chắn rằng bạn đã thay đổi các đường dẫn file CSV cho phù hợp với cấu trúc thư mục AIMLOps của bạn.
- **Phiên bản thư viện (Library Versions):** Đảm bảo rằng bạn đã cài đặt các phiên bản mới nhất của RDKit, Pandas, Scikit-learn và các thư viện khác.
- **Xử lý lỗi (Error Handling):** Luôn luôn kiểm tra và xử lý các lỗi có thể xảy ra trong quá trình thực thi code.
- **Hiệu suất (Performance):** Với bộ dữ liệu lớn, hãy xem xét sử dụng các kỹ thuật tối ưu hóa để cải thiện hiệu suất.
- **Chọn mô hình phù hợp (Choose the Right Model):** Lựa chọn mô hình phân tích phù hợp phụ thuộc vào mục tiêu cụ thể của bạn.

Chúc bạn thành công với dự án phân tích dữ liệu ChEMBL 35! Hãy cho tôi biết nếu bạn có bất kỳ câu hỏi nào khác.