PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 Topic: Các thuốc tiềm năng tái định hướng

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu Chembl 35. Với kinh nghiệm trong lĩnh vực nghiên cứu và phát triển dược học, tôi sẽ cố gắng đưa ra những hướng dẫn và phân tích chi tiết nhất.

Tổng quan dự án

Dự án của bạn tập trung vào việc khai thác dữ liệu từ Chembl 35, một cơ sở dữ liệu lớn chứa thông tin về các phân tử sinh học và hoạt tính của chúng. Bạn sẽ sử dụng rdkit, một công cụ mạnh mẽ trong tin học hóa học, để xử lý và phân tích dữ liệu này. Mục tiêu cuối cùng là hỗ trợ quá trình nghiên cứu và phát triển thuốc.

Phân tích và hướng dẫn (Song ngữ)

1. Phân tích mô hình phân tích (Analysis of the analysis model):

- **Data Extraction (Trích xuất dữ liệu):** Bước đầu tiên là trích xuất dữ liệu cần thiết từ cơ sở dữ liệu Chembl 35. Bạn sẽ sử dụng SQL để truy vấn và lọc dữ liệu dựa trên các tiêu chí cụ thể liên quan đến hoạt tính sinh học (ví dụ: IC50, Ki) và các thuộc tính của phân tử.
- **Data Preprocessing (Tiền xử lý dữ liệu):** Dữ liệu thô thường chứa nhiều thông tin nhiễu và không nhất quán. Bước này bao gồm làm sạch dữ liệu (ví dụ: loại bỏ các giá trị không hợp lệ), chuẩn hóa dữ liệu (ví dụ: chuyển đổi đơn vị đo lường) và xử lý các giá trị thiếu.
- **Feature Engineering (Thiết kế đặc trưng):** Sử dụng rdkit để tạo ra các đặc trưng (features) từ cấu trúc hóa học của các phân tử. Các đặc trưng này có thể bao gồm các mô tả phân tử (molecular descriptors) như trọng lượng phân tử, hệ số phân vùng octanol-nước (logP), diện tích bề mặt phân cực (PSA), số lượng liên kết hydro cho và nhận, và các fingerprints (ví dụ: Morgan fingerprints, MACCS keys).
- **Model Building (Xây dựng mô hình):** Sử dụng các thuật toán học máy (machine learning) để xây dựng mô hình dự đoán hoạt tính sinh học của các phân tử dựa trên các đặc trưng đã được tạo ra. Các thuật toán phổ biến bao gồm hồi quy tuyến tính (linear regression), hồi quy logistic (logistic regression), máy vector hỗ trợ (support vector machines), rừng ngẫu nhiên (random forests), và mạng nơ-ron (neural networks).
- **Model Evaluation (Đánh giá mô hình):** Đánh giá hiệu suất của mô hình bằng cách sử dụng các chỉ số phù hợp, chẳng hạn như R-squared, RMSE (Root Mean Squared Error), AUC (Area Under the Curve), và độ chính xác (accuracy).
- **Model Interpretation (Giải thích mô hình):** Tìm hiểu các yếu tố quan trọng nhất ảnh hưởng đến hoạt tính sinh học của các phân tử. Điều này có thể giúp các nhà khoa học hiểu rõ hơn về cơ chế tác động của thuốc và thiết kế các phân tử mới có hoạt tính tốt hơn.

2. Hướng dẫn song ngữ (Bilingual Instructions):

Dưới đây là hướng dẫn chi tiết cho từng bước, kèm theo code ví dụ bằng cả SQL và Python:

a. Data Extraction (Trích xuất dữ liệu):

- **SQL:**

```sql
-- Lấy dữ liệu từ bảng activities và molecules, giới hạn 100 dòng
SELECT
    act.molregno,
    mol.chembl_id,
    act.standard_type,
```

```
    act.standard_value,
    act.standard_units
FROM
    activities act
JOIN
    molecule_dictionary mol ON act.molregno = mol.molregno
WHERE
    act.standard_type = 'IC50'  -- Lọc theo loại hoạt tính
    AND act.standard_units = 'nM' -- Lọc theo đơn vị
    AND act.standard_value IS NOT NULL
    AND act.standard_value != 0  -- Loại bỏ giá trị bằng 0
    AND act.standard_value::text ~ '^[0-9\.]+$'  -- Chỉ lấy giá trị số
LIMIT 100;
```

Lưu ý: Sửa lỗi `ERROR: operator does not exist: numeric ~ unknown:`

Thay vì sử dụng ~ (regex match) trực tiếp trên cột `standard_value` (numeric), bạn cần cast nó sang text trước khi so sánh với regex. Sử dụng `act.standard_value::text ~ '^[0-9\.]+$'` để giải quyết vấn đề này.

- **Translation:**
  - This SQL query retrieves data from the `activities` and `molecule_dictionary` tables in the ChEMBL database.
  - It filters for records where the `standard_type` is 'IC50', the `standard_units` is 'nM', and the `standard_value` is a non-null numeric value.
  - It also joins the two tables based on the `molregno` column.
  - The query returns the `molregno`, `chembl_id`, `standard_type`, `standard_value`, and `standard_units` for each matching record, limited to 100 rows.

b. Data Preprocessing (Tiền xử lý dữ liệu):

- **Python:**

```
import pandas as pd
import numpy as np

# Giả sử bạn đã tải dữ liệu từ file CSV vào DataFrame 'df'
# Assume you have loaded the data from a CSV file into a DataFrame 'df'

# Ví dụ: Xử lý giá trị NaN
# Example: Handling NaN values
df = df.dropna(subset=['standard_value'])

# Ví dụ: Chuyển đổi kiểu dữ liệu
# Example: Converting data types
df['standard_value'] = pd.to_numeric(df['standard_value'])

# Ví dụ: Chuyển đổi IC50 sang pIC50 (nếu cần)
# Example: Converting IC50 to pIC50 (if needed)
df['pIC50'] = -np.log10(df['standard_value'] * 1e-9)  # Chuyển đổi nM sang M
```

- **Translation:**
  - This Python code snippet demonstrates data preprocessing steps using the pandas library.
  - It first removes rows with missing values in the 'standard_value' column.
  - Then, it converts the 'standard_value' column to a numeric data type.
  - Finally, it calculates the pIC50 values from the IC50 values (in nM) using the formula pIC50 = -log10(IC50 * 1e-9).

c. Feature Engineering (Thiết kế đặc trưng):

- **Python:**

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

def calculate_morgan_fingerprint(smiles, radius=2, nBits=2048):
    """Tính toán Morgan fingerprint từ chuỗi SMILES."""
    """Calculates Morgan fingerprint from SMILES string."""
    try:
        mol = Chem.MolFromSmiles(smiles)
        if mol:
            fp = AllChem.GetMorganFingerprintAsBitVect(mol, radius, nBits=nBits)
            return fp
        else:
            return None
    except:
        return None

# Áp dụng hàm lên cột chứa SMILES (ví dụ: 'canonical_smiles')
# Apply the function to the column containing SMILES (e.g., 'canonical_smiles')
df['morgan_fp'] = df['canonical_smiles'].apply(calculate_morgan_fingerprint)

# Chuyển đổi fingerprint thành DataFrame (nếu cần)
# Convert fingerprint to DataFrame (if needed)
fp_df = pd.DataFrame([list(fp) if fp else [None]*2048 for fp in
df['morgan_fp'].tolist()])
```

- **Translation:**
    - This Python code snippet uses rdkit to calculate Morgan fingerprints from SMILES strings.
    - The `calculate_morgan_fingerprint` function takes a SMILES string as input and returns the corresponding Morgan fingerprint as a bit vector.
    - The code then applies this function to a column in the DataFrame containing SMILES strings (e.g., 'canonical_smiles') and stores the resulting fingerprints in a new column called 'morgan_fp'.
    - Finally, it converts the fingerprints into a DataFrame, where each column represents a bit in the fingerprint.

d. Model Building (Xây dựng mô hình):

- **Python:**

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Chuẩn bị dữ liệu
# Prepare the data
X = fp_df.fillna(0)  # Điền giá trị NaN bằng 0
y = df['pIC50']

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Xây dựng mô hình RandomForestRegressor
# Build a RandomForestRegressor model
```

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
# Predict on the test set
y_pred = model.predict(X_test)

# Đánh giá mô hình
# Evaluate the model
mse = mean_squared_error(y_test, y_pred, squared=False) #Sửa lỗi  squared=False
r2 = r2_score(y_test, y_pred)

print(f'RMSE: {mse}')
print(f'R-squared: {r2}')
```

Lưu ý: Sửa lỗi `TypeError: mean_squared_error() got an unexpected keyword argument 'squared'`:

Phiên bản scikit-learn cũ có thể không hỗ trợ `squared=False` trong `mean_squared_error`. Cập nhật scikit-learn lên phiên bản mới nhất hoặc tính RMSE thủ công bằng cách lấy căn bậc hai của MSE:

```
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print(f'RMSE: {rmse}')
```

- **Translation:**
  - This Python code snippet builds a RandomForestRegressor model to predict pIC50 values.
  - It first prepares the data by filling NaN values in the fingerprint DataFrame with 0 and separating the features (X) from the target variable (y).
  - Then, it splits the data into training and testing sets using `train_test_split`.
  - It builds a RandomForestRegressor model with 100 estimators and fits it to the training data.
  - Finally, it predicts pIC50 values on the test set and evaluates the model using Root Mean Squared Error (RMSE) and R-squared. **e. Model Interpretation (Giải thích mô hình):**
- **Python:**

```
import matplotlib.pyplot as plt

# Lấy độ quan trọng của các đặc trưng
# Get feature importances
importances = model.feature_importances_

# Sắp xếp độ quan trọng theo thứ tự giảm dần
# Sort feature importances in descending order
indices = np.argsort(importances)[::-1]

# Lấy tên của các đặc trưng quan trọng nhất (ví dụ: 10 đặc trưng đầu tiên)
# Get the names of the most important features (e.g., the first 10 features)
top_n = 10
top_indices = indices[:top_n]

# Vẽ biểu đồ độ quan trọng của các đặc trưng
# Plot feature importances
plt.figure(figsize=(10, 6))
plt.title("Feature Importances")
plt.bar(range(top_n), importances[top_indices], align="center")
plt.xticks(range(top_n), top_indices)  # Thay thế bằng tên đặc trưng nếu có
```

```
plt.xlim([-1, top_n])
plt.show()
```

- **Translation:**
  - This Python code snippet interprets the RandomForestRegressor model by identifying the most important features.
  - It first retrieves the feature importances from the trained model.
  - Then, it sorts the feature importances in descending order to identify the most influential features.
  - Finally, it plots a bar chart showing the importances of the top N features.

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

Dưới đây là 5 ví dụ về code SQL và Python mẫu cho các tác vụ khác nhau:

Ví dụ 1: Lọc dữ liệu theo khoảng giá trị (Filtering data by value range)

- **SQL:**

```sql
SELECT chembl_id, standard_value
FROM activities act
JOIN molecule_dictionary mol ON act.molregno = mol.molregno
WHERE standard_type = 'IC50'
AND standard_value BETWEEN 100 AND 1000
LIMIT 100;
```

- **Python:**

```
df_filtered = df[(df['standard_value'] >= 100) & (df['standard_value'] <= 1000)]
```

Ví dụ 2: Tính toán thống kê mô tả (Calculating descriptive statistics)

- **SQL:**

```sql
SELECT
    AVG(standard_value) AS average_ic50,
    MIN(standard_value) AS min_ic50,
    MAX(standard_value) AS max_ic50
FROM activities
WHERE standard_type = 'IC50';
```

- **Python:**

```
print(df['standard_value'].describe())
```

Ví dụ 3: Tạo fingerprint từ SMILES (Generating fingerprint from SMILES)

- **Python:**

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

def generate_fingerprint(smiles):
    mol = Chem.MolFromSmiles(smiles)
    fp = AllChem.GetMorganFingerprintAsBitVect(mol, 2, 1024)
    return fp

# Assuming you have a DataFrame with a 'smiles' column
df['fingerprint'] = df['canonical_smiles'].apply(generate_fingerprint)
```

Ví dụ 4: Xây dựng mô hình đơn giản (Building a simple model)

- **Python:**

```
from sklearn.linear_model import LinearRegression
```

```
# Assuming you have X (features) and y (target)
model = LinearRegression()
model.fit(X_train, y_train)
```

Ví dụ 5: Đánh giá mô hình (Evaluating a model)

- **Python:**

```
from sklearn.metrics import mean_squared_error, r2_score

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}")
```

Lưu ý quan trọng:

- **Đường dẫn:** Luôn sử dụng `os.path.join(base_path, ...)` để tạo đường dẫn một cách nhất quán và không phụ thuộc vào hệ điều hành.
- **Phiên bản thư viện:** Đảm bảo rằng bạn đang sử dụng các phiên bản thư viện tương thích với code. Nếu gặp lỗi, hãy thử cập nhật hoặc hạ cấp phiên bản của thư viện.
- **Xử lý lỗi:** Luôn kiểm tra và xử lý các lỗi có thể xảy ra trong quá trình chạy code, đặc biệt là khi làm việc với dữ liệu lớn và các công cụ phức tạp như rdkit.

Hy vọng những hướng dẫn và ví dụ này sẽ giúp bạn tiến xa hơn trong dự án của mình. Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi!