PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

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

🧩 Topic: Phân cụm không giám sát các hợp chất ChEMBL

Tuyệt vời! 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 cho chủ đề Topic_CheMBL_35_68, tập trung vào việc khắc phục các lỗi bạn đã gặp và tuân thủ theo cấu trúc dự án AIMLOps.

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

Chủ đề Topic_CheMBL_35_68 có thể tập trung vào một số khía cạnh khác nhau của dữ liệu Chembl 35. Dựa trên thông tin bạn cung cấp, tôi giả định chủ đề này liên quan đến việc:

- **Lọc và chuẩn bị dữ liệu:** Loại bỏ các giá trị không hợp lệ hoặc không phù hợp để đảm bảo chất lượng dữ liệu.
- **Tính toán các đặc trưng hóa học:** Sử dụng rdkit để tạo ra các đặc trưng (descriptors) từ cấu trúc hóa học của các hợp chất.
- **Phân tích mối quan hệ cấu trúc-hoạt tính (SAR):** Tìm hiểu mối liên hệ giữa cấu trúc hóa học và hoạt tính sinh học của các hợp chất. Điều này có thể bao gồm việc xây dựng các mô hình dự đoán hoạt tính.

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

1. **Data Extraction and Filtering:** Extracting relevant data from the Chembl 35 database and filtering it based on specific criteria (e.g., activity type, target, assay).
2. **Feature Generation:** Generating molecular descriptors using RDKit based on the chemical structures of the compounds.
3. **Exploratory Data Analysis (EDA):** Performing EDA to understand the distribution of features and identify potential relationships between features and activity.
4. **Model Building:** Building machine learning models to predict activity based on the molecular descriptors.
5. **Model Evaluation:** Evaluating the performance of the models using appropriate metrics (e.g., R-squared, RMSE, AUC).
6. **Interpretation and Insights:** Interpreting the models to understand the key structural features that contribute to activity.

2. Hướng dẫn song ngữ (Bilingual Instructions)

Bước 1: Kết nối đến cơ sở dữ liệu và trích xuất dữ liệu (Connect to the database and extract data)

- **Tiếng Việt:** Sử dụng `psycopg2` trong Python để kết nối đến cơ sở dữ liệu PostgreSQL và viết các truy vấn SQL để lấy dữ liệu cần thiết.
- **English:** Use `psycopg2` in Python to connect to the PostgreSQL database and write SQL queries to retrieve the necessary data.

Bước 2: Khắc phục lỗi SQL (Fixing the SQL Error)

- **Tiếng Việt:** Lỗi `operator does not exist: numeric ~ unknown` xảy ra do bạn đang cố gắng so sánh một cột số (numeric) với một chuỗi (unknown). Bạn cần ép kiểu cột `standard_value` về kiểu số trước khi so sánh.

- **English:** The error `operator does not exist: numeric ~ unknown` occurs because you are trying to compare a numeric column with a string. You need to cast the `standard_value` column to a numeric type before comparison.

Bước 3: Tạo đặc trưng hóa học (Generating Chemical Features)

- **Tiếng Việt:** Sử dụng rdkit để đọc cấu trúc hóa học (SMILES) và tính toán các đặc trưng như MW (Molecular Weight), LogP, HBA (Hydrogen Bond Acceptors), HBD (Hydrogen Bond Donors), và TPSA (Topological Polar Surface Area).
- **English:** Use rdkit to read the chemical structures (SMILES) and calculate features such as MW (Molecular Weight), LogP, HBA (Hydrogen Bond Acceptors), HBD (Hydrogen Bond Donors), and TPSA (Topological Polar Surface Area).

Bước 4: Xây dựng mô hình (Building Models)

- **Tiếng Việt:** Sử dụng scikit-learn để xây dựng các mô hình hồi quy (ví dụ: Linear Regression, Random Forest) để dự đoán hoạt tính dựa trên các đặc trưng hóa học.
- **English:** Use scikit-learn to build regression models (e.g., Linear Regression, Random Forest) to predict activity based on the chemical features.

Bước 5: Đánh giá mô hình (Evaluating Models)

- **Tiếng Việt:** Sử dụng các độ đo như R-squared, RMSE (Root Mean Squared Error) để đánh giá hiệu suất của mô hình.
- **English:** Use metrics like R-squared, RMSE (Root Mean Squared Error) to evaluate the performance of the model.

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

SQL (lấy 100 dòng dữ liệu):

```sql
-- Extracting data from chembl_35
SELECT
    act.molregno,
    act.standard_value,
    act.standard_units,
    act.standard_type,
    md.chembl_id,
    md.smiles
FROM
    activities act
JOIN
    molecule_dictionary md ON act.molregno = md.molregno
WHERE
    act.standard_type = 'IC50'
    AND act.standard_units = 'nM'
    AND act.standard_value IS NOT NULL
    AND act.standard_value::TEXT ~ '^[0-9\.]+$' -- Ensuring standard_value is numeric
LIMIT 100;
```

Python (Jupyter Notebook):

```python
import os
import psycopg2
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors
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
```

```python
# Define project paths based on AIMLOps template
base_path = os.getcwd()  # Assuming you are in the project root
data_path = os.path.join(base_path, 'data')
notebook_path = os.path.join(base_path, 'notebooks')

# Database connection parameters
db_params = {
    'host': '192.168.206.136',
    'user': 'rd',
    'password': 'rd',
    'database': 'chembl_35'
}

# Function to connect to the database and execute query
def execute_query(query, db_params):
    conn = None
    try:
        conn = psycopg2.connect(**db_params)
        df = pd.read_sql_query(query, conn)
        return df
    except Exception as e:
        print(f"Error: {e}")
        return None
    finally:
        if conn:
            conn.close()

# SQL query to fetch data
sql_query = """
    SELECT
        act.molregno,
        act.standard_value,
        act.standard_units,
        act.standard_type,
        md.chembl_id,
        md.smiles
    FROM
        activities act
    JOIN
        molecule_dictionary md ON act.molregno = md.molregno
    WHERE
        act.standard_type = 'IC50'
        AND act.standard_units = 'nM'
        AND act.standard_value IS NOT NULL
        AND act.standard_value::TEXT ~ '^[0-9\.]+$'
    LIMIT 100;
"""

# Fetch data from the database
df = execute_query(sql_query, db_params)

# Save data to CSV (optional)
if df is not None:
    csv_file_path = os.path.join(data_path, 'chembl_data.csv')
    df.to_csv(csv_file_path, index=False)
    print(f"Data saved to {csv_file_path}")
else:
    print("Failed to retrieve data.")
    exit()
```

```
# Function to calculate molecular descriptors using RDKit
def calculate_descriptors(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is None:
        return None
    descriptors = {
        "MW": Descriptors.MolWt(mol),
        "LogP": Descriptors.MolLogP(mol),
        "HBA": Descriptors.NumHAcceptors(mol),
        "HBD": Descriptors.NumHDonors(mol),
        "TPSA": Descriptors.TPSA(mol)
    }
    return descriptors

# Apply descriptor calculation
df['descriptors'] = df['smiles'].apply(calculate_descriptors)
df = df.dropna(subset=['descriptors'])  # Drop rows where descriptor calculation
failed
df = pd.concat([df, df['descriptors'].apply(pd.Series)], axis=1)
df['pIC50'] = -np.log10(df['standard_value'].astype(float) * 1e-9) #convert IC50 to
pIC50
df = df.dropna(subset=['pIC50']) #remove null pIC50
# Prepare data for modeling
X = df[['MW', 'LogP', 'HBA', 'HBD', 'TPSA']]
y = df['pIC50']

# Split 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)

# Train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
rmse = np.sqrt(mean_squared_error(y_test, y_pred, squared=False))
r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse}")
print(f"R-squared: {r2}")

# Save the results or the model (optional)
# Example: Saving the model
# import joblib
# model_file_path = os.path.join(model_path, 'linear_regression_model.pkl')
# joblib.dump(model, model_file_path)
# print(f"Model saved to {model_file_path}")
```

Giải thích code (Code Explanation):

1. **Kết nối cơ sở dữ liệu (Database Connection):** Sử dụng `psycopg2` để kết nối đến cơ sở dữ liệu Chembl 35.
2. **Truy vấn SQL (SQL Query):** Lấy dữ liệu từ các bảng `activities` và `molecule_dictionary`, lọc theo `standard_type = 'IC50'` và `standard_units = 'nM'`.

3. **Lọc dữ liệu (Data Filtering):** Đảm bảo cột `standard_value` chỉ chứa các giá trị số.
4. **Tính toán đặc trưng (Descriptor Calculation):** Sử dụng rdkit để tính toán các đặc trưng hóa học từ cấu trúc SMILES.
5. **Xây dựng mô hình (Model Building):** Sử dụng `LinearRegression` từ scikit-learn để xây dựng mô hình dự đoán.
6. **Đánh giá mô hình (Model Evaluation):** Tính toán RMSE và R-squared để đánh giá hiệu suất của mô hình.
7. **Lưu trữ kết quả (Optional):** Lưu trữ kết quả hoặc mô hình đã huấn luyện để sử dụng sau này.

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

Ví dụ 1: Lọc theo một target cụ thể (Filtering by a specific target)

- **SQL:**

```sql
SELECT
    act.molregno,
    act.standard_value,
    act.standard_units,
    act.standard_type,
    md.chembl_id,
    md.smiles
FROM
    activities act
JOIN
    molecule_dictionary md ON act.molregno = md.molregno
WHERE
    act.standard_type = 'IC50'
    AND act.standard_units = 'nM'
    AND act.standard_value IS NOT NULL
    AND act.standard_value::TEXT ~ '^[0-9\.]+$'
    AND act.target_chembl_id = 'CHEMBL205'  -- Specific target
LIMIT 100;
```

- **Python:**

```python
# Modified SQL query
sql_query = """
    SELECT
        act.molregno,
        act.standard_value,
        act.standard_units,
        act.standard_type,
        md.chembl_id,
        md.smiles
    FROM
        activities act
    JOIN
        molecule_dictionary md ON act.molregno = md.molregno
    WHERE
        act.standard_type = 'IC50'
        AND act.standard_units = 'nM'
        AND act.standard_value IS NOT NULL
        AND act.standard_value::TEXT ~ '^[0-9\.]+$'
        AND act.target_chembl_id = 'CHEMBL205'  -- Specific target
    LIMIT 100;
"""
df = execute_query(sql_query, db_params)
```

Ví dụ 2: Tính toán nhiều đặc trưng hơn (Calculating More Descriptors)

- **Python:**

```
from rdkit.Chem import Lipinski

def calculate_more_descriptors(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is None:
        return None
    descriptors = {
        "MW": Descriptors.MolWt(mol),
        "LogP": Descriptors.MolLogP(mol),
        "HBA": Descriptors.NumHAcceptors(mol),
        "HBD": Descriptors.NumHDonors(mol),
        "TPSA": Descriptors.TPSA(mol),
        "RotatableBonds": Descriptors.NumRotatableBonds(mol),
        "LipinskiHBA": Lipinski.NumHAcceptors(mol),
        "LipinskiHBD": Lipinski.NumHDonors(mol)
    }
    return descriptors
```

Ví dụ 3: Sử dụng mô hình Random Forest (Using Random Forest Model)

- **Python:**

```
from sklearn.ensemble import RandomForestRegressor

# Initialize and train the Random Forest model
model = RandomForestRegressor(n_estimators=100, random_state=42)  # You can adjust
hyperparameters
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
rmse = np.sqrt(mean_squared_error(y_test, y_pred, squared=False))
r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse}")
print(f"R-squared: {r2}")
```

Ví dụ 4: Chuẩn hóa dữ liệu (Scaling Data)

- **Python:**

```
from sklearn.preprocessing import StandardScaler

# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train the model on scaled data
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = model.predict(X_test_scaled)
```

Ví dụ 5: Lưu trữ mô hình và kết quả (Saving Model and Results)

- **Python:**

```
import joblib
```

```
# Save the model
model_filename = os.path.join(base_path, 'models', 'linear_regression_model.pkl')
joblib.dump(model, model_filename)

# Save the scaler
scaler_filename = os.path.join(base_path, 'models', 'scaler.pkl')
joblib.dump(scaler, scaler_filename)


# Save the evaluation metrics to a text file
results_filename = os.path.join(base_path, 'results', 'evaluation_metrics.txt')
with open(results_filename, 'w') as f:
    f.write(f"RMSE: {rmse}\n")
    f.write(f"R-squared: {r2}\n")
```

Hy vọng những hướng dẫn và ví dụ này sẽ giúp bạn tiến xa hơn trong dự án của mình. Hãy cho tôi biết nếu bạn có bất kỳ câu hỏi nào khác!