



**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**BÁO CÁO CUỐI KÌ**  
**KHOA HỌC DỮ LIỆU ỨNG DỤNG**

| Chủ đề |

**Dự đoán tiến triển của bệnh Parkinson**

| Sinh viên |

<b>Nguyễn Hữu Phương</b>	<b>19120625</b>
<b>Phạm Hoàng Nam Anh</b>	<b>18120278</b>
<b>Phạm Quốc Hưng</b>	<b>19120522</b>
<b>Hoàng Mạnh Khiêm</b>	<b>19120543</b>

Thành phố Hồ Chí Minh – Tháng 4 năm 2023

---

## MỤC LỤC

---

<b>GIỚI THIỆU.....</b>	<b>4</b>
Về đề tài.....	4
Ứng dụng, ý nghĩa khoa học và thực tiễn.....	4
<b>CHUẨN BỊ DỮ LIỆU.....</b>	<b>5</b>
Tải và đọc các tập dữ liệu.....	5
Kaggle Notebook.....	5
Colab Notebook.....	6
Tập dữ liệu Peptides.....	7
Tập dữ liệu Proteins.....	8
Tập dữ liệu khám lâm sàng.....	9
<b>KHÁM PHÁ DỮ LIỆU.....</b>	<b>11</b>
Tập dữ liệu Peptides.....	11
Tập dữ liệu Proteins.....	14
Tập dữ liệu khám lâm sàng.....	16
<b>TRỰC QUAN HÓA DỮ LIỆU.....</b>	<b>18</b>
Tập dữ liệu peptide.....	18
Phân bố của các cột dữ liệu kiểu số.....	18
Top 20 mã Uniprot protein xuất hiện nhiều nhất?.....	19
Top 20 Peptide xuất hiện nhiều nhất?.....	19
Hệ số tương quan giữa các cột dữ liệu.....	20
Mối quan hệ giữa các cột dữ liệu.....	21
Những Amino Acid nào xuất hiện nhiều nhất?.....	21
Tập dữ liệu protein.....	23
Phân bố của các cột dữ liệu kiểu số.....	23
Top 20 mã Uniprot Protein xuất hiện nhiều nhất?.....	24
Tương quan giữa các cột dữ liệu.....	25
Mối quan hệ giữa các cột dữ liệu.....	26
Sự thay đổi của tần suất xuất hiện của Protein trong mẫu (NPX) như thế nào?.....	27

Tập dữ liệu khám lâm sàng.....	28
Phân bố của các cột dữ liệu kiểu số.....	28
Tỉ lệ phần trăm bệnh nhân có sử dụng thuốc hay không trong lúc đánh giá?.....	30
Tương quan giữa các cột dữ liệu.....	31
Mối quan hệ giữa các cột dữ liệu.....	32
Sự thay đổi của tình trạng bệnh theo thời gian như thế nào?.....	33
Mối quan hệ giữa tần suất xuất hiện protein(NPX) và các cột điểm đánh giá....	38
<b>RÚT TRÍCH ĐẶC TRƯNG &amp; CHUẨN BỊ DỮ LIỆU HUẤN LUYỆN.....</b>	<b>39</b>
Xử lý dữ liệu bị thiếu.....	39
Chuẩn bị dữ liệu huấn luyện.....	40
Chuẩn hóa dữ liệu.....	41
Rút trích đặc trưng.....	41
<b>XÂY DỰNG MÔ HÌNH HUẤN LUYỆN.....</b>	<b>42</b>
<b>ĐÁNH GIÁ.....</b>	<b>43</b>
UPDRS_1.....	43
UPDRS_2.....	45
UPDRS_3.....	48
UPDRS_4.....	50
<b>NHẬN XÉT CHUNG.....</b>	<b>54</b>
<b>KẾT QUẢ SMAPE TRÊN KAGGLE.....</b>	<b>57</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>60</b>

---

## GIỚI THIỆU

---

### Về đề tài

Đề tài mà nhóm lựa chọn là **Dự đoán tiến triển của bệnh Parkinson (Parkinson's Disease Progression Prediction)**.

Bệnh Parkinson là một bệnh gây rối loạn não làm ảnh hưởng đến khả năng đi lại, nhận thức, giấc ngủ và các chức năng bình thường khác. Hiện nay vẫn chưa có phương pháp nào có thể chữa trị hiệu quả căn bệnh này và bệnh sẽ ngày càng trầm trọng hơn theo thời gian. Các nghiên cứu cho thấy các đột biến protein và peptide đóng một vai trò quan trọng trong sự khởi phát và trầm trọng hóa của căn bệnh này.

**Bài toán:** Mục tiêu của đề tài này là dự đoán điểm **MDS-UPDRS** thông qua dữ liệu đo lường protein và peptide để xác định tiến triển của bệnh Parkinson. Thang điểm này gồm có 4 phần:

- **Phần 1:** Đánh giá các trải nghiệm không chuyển động trong cuộc sống hàng ngày của bệnh nhân Parkinson. Gồm 13 câu hỏi.
- **Phần 2:** Đánh giá các trải nghiệm chuyển động trong cuộc sống hàng ngày của bệnh nhân Parkinson. Gồm 13 câu hỏi.
- **Phần 3:** Đánh giá các khía cạnh chuyển động của bệnh Parkinson trong khi thực hiện các hoạt động hàng ngày. Gồm 18 câu hỏi.
- **Phần 4:** Đánh giá các biến chứng chuyển động của bệnh Parkinson. Gồm 6 câu hỏi.

Mỗi câu hỏi được đánh giá trên thang điểm 0 - 4: **0 (bình thường - normal)**, **1 (nhẹ - slight)**, **2 (trung bình - mild)**, **3 (nặng - moderate)** và **4 (rất nặng - severe)**.

### Ứng dụng, ý nghĩa khoa học và thực tiễn

Việc thực hiện đề tài này sẽ giúp dự đoán tiến triển của bệnh Parkinson thông qua dữ liệu đo lường protein và peptide. Điều này sẽ hỗ trợ các bác sĩ và các nhà nghiên cứu trong việc tìm hiểu, nghiên cứu chuyên sâu hơn về căn bệnh này.

Bên cạnh đó, có thể tìm được các giải pháp để làm chậm tiến triển, hạn chế ảnh hưởng của căn bệnh tới người mắc phải. Thậm chí chữa trị dứt điểm bệnh Parkinson.

---

## CHUẨN BỊ DỮ LIỆU

---

### Tải và đọc các tập dữ liệu

#### Kaggle Notebook

Các tập dữ liệu sẽ được lấy từ thư mục *.kaggle/input*.

```
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/amp-parkinsons-disease-progression-prediction/train_proteins.csv
/kaggle/input/amp-parkinsons-disease-progression-prediction/train_clinical_data.csv
/kaggle/input/amp-parkinsons-disease-progression-prediction/public_timeseries_testing_util.py
/kaggle/input/amp-parkinsons-disease-progression-prediction/supplemental_clinical_data.csv
/kaggle/input/amp-parkinsons-disease-progression-prediction/train_peptides.csv
/kaggle/input/amp-parkinsons-disease-progression-prediction/amp_pd_peptide/competition.cpython-37m-x86_64-linux-gnu.so
/kaggle/input/amp-parkinsons-disease-progression-prediction/amp_pd_peptide/__init__.py
/kaggle/input/amp-parkinsons-disease-progression-prediction/amp_pd_peptide_310/competition.cpython-310-x86_64-linux-gnu.so
```

Sau đó tiến hành đọc dữ liệu lên để xử lý.

```
peptides_df = pd.read_csv('/kaggle/input/amp-parkinsons-disease-progression-prediction/train_peptides.csv')
proteins_df = pd.read_csv('/kaggle/input/amp-parkinsons-disease-progression-prediction/train_proteins.csv')
clinical_df = pd.read_csv('/kaggle/input/amp-parkinsons-disease-progression-prediction/train_clinical_data.csv')
```

---

## CHUẨN BỊ DỮ LIỆU

---

### Colab Notebook

Các tập dữ liệu được tải về từ trên [Kaggle](#) lưu trữ trong thư mục [Drive](#) của nhóm đã tạo trước.

Sử dụng thư viện hỗ trợ [gdown](#) để tải xuống các tập dữ liệu và *pandas* để đọc các tập dữ liệu lên phục vụ cho các phân tích, khám phá tiếp theo.

```
!gdown https://drive.google.com/drive/folders/1OL53up22m7ykFMvE9XVGMBsq-v0Mg-RA?usp=sharing -O /content/data --folder

Retrieving folder list
Processing file 1QtL8mTFjokTqvAeVpBbjgFEZ16YpwB6X train_clinical_data.csv
Processing file 18lMzGXyWxkzRiJcydYC2vG4s7W7LLnNL train_peptides.csv
Processing file 1uJ-N328rCDEsMh0k5_xySRXwvAzT40mJ train_proteins.csv
Retrieving folder list completed
Building directory structure
Building directory structure completed
Downloading...
From: https://drive.google.com/uc?id=1QtL8mTFjokTqvAeVpBbjgFEZ16YpwB6X
To: /content/data/train_clinical_data.csv
100% 74.1k/74.1k [00:00<00:00, 42.1MB/s]
Downloading...
From: https://drive.google.com/uc?id=18lMzGXyWxkzRiJcydYC2vG4s7W7LLnNL
To: /content/data/train_peptides.csv
100% 51.4M/51.4M [00:00<00:00, 217MB/s]
Downloading...
From: https://drive.google.com/uc?id=1uJ-N328rCDEsMh0k5_xySRXwvAzT40mJ
To: /content/data/train_proteins.csv
100% 7.66M/7.66M [00:00<00:00, 161MB/s]
Download completed

peptides_df = pd.read_csv('/content/data/train_peptides.csv')
proteins_df = pd.read_csv('/content/data/train_proteins.csv')
clinical_df = pd.read_csv('/content/data/train_clinical_data.csv')
```

Như đã đề cập ở đề cương đồ án trước đó, dữ liệu mà nhóm sử dụng sẽ gồm 3 tập dữ liệu:

- train\_peptides.csv
- train\_proteins.csv
- train\_clinical\_data.csv

Các bước chuẩn bị dữ liệu, khám phá và trực quan hóa dữ liệu sẽ được trình bày bên dưới.

---

## CHUẨN BỊ DỮ LIỆU

---

Các bước xử lý tiếp theo trong phần chuẩn bị dữ liệu là tương đồng nhau ở cả Kaggle Notebook và Colab Notebook.

### Tập dữ liệu Peptides

Tập dữ liệu này không có bất cứ cột dữ liệu nào bị thiếu hay bị lặp dòng dữ liệu.

#### Dữ liệu có bị thiếu hay không?

```
[ ] peptides_df.isna().mean().sort_values(ascending = False).round(4) * 100
```

visit_id	0.0
visit_month	0.0
patient_id	0.0
UniProt	0.0
Peptide	0.0
PeptideAbundance	0.0
dtype: float64	

**Nhận xét:** Không có cột thuộc tính nào bị thiếu dữ liệu.

#### Dữ liệu có bị lặp hay không?

```
[ ] peptides_df.duplicated().sum()
```

0

**Nhận xét:** Không có dòng dữ liệu nào bị lặp.

Do đó không cần thực hiện trước bước tiền xử lý dữ liệu để chuẩn bị cho bước khám phá, trực quan tiếp theo.

---

## CHUẨN BỊ DỮ LIỆU

---

### Tập dữ liệu Proteins

Tập dữ liệu này cũng không có bất cứ cột dữ liệu nào bị thiếu hay bị lặp dòng dữ liệu.

#### Dữ liệu có bị thiếu hay không?

```
[ ] proteins_df.isna().mean().sort_values(ascending = False).round(4) * 100  
  
visit_id      0.0  
visit_month   0.0  
patient_id    0.0  
UniProt       0.0  
NPX           0.0  
dtype: float64
```

**Nhận xét:** Không có cột thuộc tính nào bị thiếu dữ liệu.

#### Dữ liệu có bị lặp hay không?

```
[ ] proteins_df.duplicated().sum()  
  
0
```

**Nhận xét:** Không có dòng dữ liệu nào bị lặp.

Do đó cũng không cần thực hiện trước bước tiền xử lý dữ liệu để chuẩn bị cho bước khám phá, trực quan tiếp theo.



---

## CHUẨN BỊ DỮ LIỆU

---

### Tập dữ liệu khám lâm sàng

Dữ liệu có bị thiếu hay không?

```
[ ] clinical_df.isna().mean().sort_values(ascending = False).round(4) * 100

upd23b_clinical_state_on_medication    50.75
updrs_4                                39.69
updrs_3                                 0.96
updrs_2                                 0.08
updrs_1                                 0.04
visit_id                               0.00
patient_id                             0.00
visit_month                            0.00
dtype: float64
```

**Nhận xét:** Tập dữ liệu có một số cột bị thiếu dữ liệu, trong đó có 2 cột có tỉ lệ dữ liệu thiếu cao nhất lần lượt là 50.75% và 39.69%.

Tập dữ liệu này có các cột `upd23b_clinical_state_on_medication`, `updrs_4`, `updrs_3`, `updrs_2` và `updrs_1` là bị thiếu dữ liệu do đó sẽ tiến hành tiền xử lý một chút trước khi khám phá, trực quan.

Các tập dữ liệu `updrs_3`, `updrs_2` và `updrs_1` có tỉ lệ thiếu nhỏ và không đáng kể nên sẽ tiến hành xóa các dòng có dữ liệu bị thiếu ở cột này đi.

```
clinical_df.dropna(subset = ['updrs_1', 'updrs_2', 'updrs_3'], inplace = True)
clinical_df.isna().mean().sort_values(ascending = False).round(4) * 100

upd23b_clinical_state_on_medication    50.54
updrs_4                                39.64
visit_id                               0.00
patient_id                             0.00
visit_month                            0.00
updrs_1                                0.00
updrs_2                                0.00
updrs_3                                0.00
dtype: float64
```

Cột dữ liệu `upd23b_clinical_state_on_medication` thiếu tới 50.54% nếu loại bỏ các hàng thiếu dữ liệu sẽ gây mất mát lượng lớn dữ liệu. Do đó sẽ cần điền vào dữ liệu bị thiếu ở đây.

---

## CHUẨN BỊ DỮ LIỆU

---

Theo nhóm tìm hiểu được, dữ liệu bị thiếu ở cột này là do không xác định được liệu bệnh nhân có sử dụng thuốc hay không trong quá trình đánh giá. Vì vậy, những giá trị bị thiếu nhóm sẽ điền bằng giá trị 'Unknown'

```
clinical_df['upd23b_clinical_state_on_medication'].fillna(value = 'Unknow', inplace = True)
clinical_df.isna().mean().sort_values(ascending = False).round(4) * 100
```

```
updrs_4          39.64
visit_id         0.00
patient_id       0.00
visit_month      0.00
updrs_1          0.00
updrs_2          0.00
updrs_3          0.00
upd23b_clinical_state_on_medication  0.00
dtype: float64
```

Cột cuối cùng bị thiếu là `updrs_4`. Tuy nhiên đây lại là cột dữ liệu mục tiêu mà ta cần phải dự đoán nên việc điền dữ liệu ở cột này có thể rất nguy hiểm, ảnh hưởng đến kết quả mô hình dự đoán sẽ thực hiện ở giai đoạn tiếp theo. Do đó nhóm quyết định tạm thời giữ nguyên cột dữ liệu này.

Tập dữ liệu này không có hàng dữ liệu nào bị lặp lại.

### Dữ liệu có bị lặp hay không?

```
[ ] clinical_df.duplicated().sum()
```

0

**Nhận xét:** Không có dòng dữ liệu nào bị lặp.

Các tập dữ liệu đã sẵn sàng cho phần khám phá và trực quan hóa dữ liệu tiếp theo.

## KHÁM PHÁ DỮ LIỆU

Các bước xử lý trong phần khám phá dữ liệu là tương đồng nhau ở cả Kaggle Notebooj và Colab Notebook

### Tập dữ liệu Peptides

Tập dữ liệu này chứa dữ liệu phổ khối lượng ở mức peptide, là các đơn vị thành phần của protein. Dữ liệu này giúp nắm bắt sự thay đổi về cấu trúc của protein trong quá trình tiến triển của bệnh Parkinson.

```
peptides_df.head(5)
```

	visit_id	visit_month	patient_id	UniProt	Peptide	PeptideAbundance
0	55_0	0	55	O00391	NEQEQLGQWHLS	11254.3
1	55_0	0	55	O00533	GNPEPTFSWTK	102060.0
2	55_0	0	55	O00533	IEIPSSVQQVPTIIK	174185.0
3	55_0	0	55	O00533	KPQSAVYSTGSGILLC(UniMod_4)EAEQEPQPTIK	27278.9
4	55_0	0	55	O00533	SMEQNGPGLEYR	30838.7

Tập dữ liệu này 981834 hàng và 6 cột thuộc tính.

```
[ ] peptides_df.shape
```

```
(981834, 6)
```

**Nhận xét:** Tập dữ liệu có 981834 hàng và 6 cột thuộc tính.

Mô tả về các cột dữ liệu như sau:

Tên cột	Ý nghĩa
visit_id	Mã id của lần khám
month_id	Tháng khám, tính từ thời điểm lần đầu bệnh nhân đến khám
patient_id	Mã bệnh nhân
UniProt	Mã UniProt của protein dùng để phân biệt các loại protein. Mỗi protein thường có nhiều loại peptide
Peptide	Chuỗi amino acids có trong peptide. Được kí hiệu bằng các mã chữ cái
PeptideAbundance	Tần suất xuất hiện của amino acids trong tập mẫu

## KHÁM PHÁ DỮ LIỆU

Tập dữ liệu này có 3 cột dữ liệu kiểu số (`int64`, `float64`) và 3 cột kiểu `object`

```
peptides_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 981834 entries, 0 to 981833
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   visit_id              981834 non-null  object  
1   visit_month           981834 non-null  int64   
2   patient_id            981834 non-null  int64   
3   UniProt               981834 non-null  object  
4   Peptide               981834 non-null  object  
5   PeptideAbundance      981834 non-null  float64  
dtypes: float64(1), int64(2), object(3)
memory usage: 44.9+ MB
```

Mô tả thống kê cho dữ liệu kiểu số:

	count	mean	std	min	25%	50%	75%	max
visit_month	981834.0	26.11	22.91	0.0	6.00	24.0	48.00	108.0
patient_id	981834.0	32603.47	18605.93	55.0	16566.00	29313.0	49995.00	65043.0
PeptideAbundance	981834.0	642890.25	3377989.09	11.0	28174.25	74308.3	221338.75	178752000.0

Ta thấy cột `PeptideAbundance` có giá trị min nhỏ trong khi giá trị max rất lớn nên cần chuẩn hóa lại dữ liệu trước khi đưa vào mô hình.

Mô tả thống kê cho dữ liệu kiểu `object`:

	visit_id	UniProt	Peptide
count	981834	981834	981834
unique	1113	227	968
unique_vals	[55_0, 1517_0, 1923_0, 2660_0, 3636_0, 3863_0,...	[000391, 000533, 000584, 014498, 014773, 01479...	[NEQEQLGQWHL, GNPEPTFSWTK, IEIPSSVQVPTIIK, ...
top	47171_6	P02787	AIGYLNTGYQR

---

## KHÁM PHÁ DỮ LIỆU

---

Một số **Peptide** có thêm mã **UniMod\_X**. Trong dữ liệu có 2 mã là **UniMod\_4** và **UniMod\_25**. Theo nhóm tìm hiểu 2 mã này có nghĩa là:

- **UniMod\_4** trong mã Uniprot là mã của một biến thể bổ sung của axit amin Methionine (Met) gọi là Oxidation, nó được sử dụng để chỉ ra rằng Met đã bị oxy hóa trong chuỗi polypeptide.
- **UniMod\_35** trong mã Uniprot là mã của một biến thể bổ sung của axit amin Serine (Ser) gọi là Phosphorylation, nó được sử dụng để chỉ ra rằng Ser đã được phosphory hóa trong chuỗi polypeptide.

	visit_id	visit_month	patient_id	Peptide
3	55_0	0	55	KPQSAVYSTGSNGILLC(UniMod_4)EAEGEPQPTIK
10	55_0	0	55	HGTC(UniMod_4)AAQVDALNSQKK
19	55_0	0	55	NIINSDGGPYVC(UniMod_4)R
24	55_0	0	55	SEGLLAC(UniMod_4)GTNAR
30	55_0	0	55	EVGPTNADPVC(UniMod_4)LAK
35	55_0	0	55	MYYSAVDPTKDIFTGLIGPM(UniMod_35)K
39	55_0	0	55	DKLAAC(UniMod_4)LEGNC(UniMod_4)AEGLGTNYR
41	55_0	0	55	LAVTTHGLPC(UniMod_4)LAWASAQAK
42	55_0	0	55	RQEC(UniMod_4)SIPVC(UniMod_4)GQDQVTVAMTPR
43	55_0	0	55	SEGSSVNLSPPLEQC(UniMod_4)VPDRGQQYQGR

## KHÁM PHÁ DỮ LIỆU

### Tập dữ liệu Proteins

Tập dữ liệu chứa tần suất biểu hiện protein được tổng hợp từ tập dữ liệu mức peptide. Dữ liệu này giúp nắm bắt sự thay đổi về mức độ biểu hiện của các protein liên quan đến bệnh Parkinson.

	visit_id	visit_month	patient_id	UniProt	NPX
0	55_0	0	55	O00391	11254.3
1	55_0	0	55	O00533	732430.0
2	55_0	0	55	O00584	39585.8
3	55_0	0	55	O14498	41526.9
4	55_0	0	55	O14773	31238.0

Tập dữ liệu này 232741 hàng và 5 cột thuộc tính.

```
[ ] proteins_df.shape  
  
(232741, 5)
```

**Nhận xét:** Tập dữ liệu có 232741 hàng và 5 cột thuộc tính.

Mô tả về các cột dữ liệu như sau:

Tên cột	Ý nghĩa
visit_id	Mã id của lần khám
month_id	Tháng khám, tính từ thời điểm lần đầu bệnh nhân đến khám
patient_id	Mã bệnh nhân
UniProt	Mã UniProt của protein dùng để phân biệt các loại protein. Mỗi protein thường có nhiều loại peptide
NPX	Tần suất xuất hiện của protein trong mẫu.

## KHÁM PHÁ DỮ LIỆU

Tập dữ liệu này có 3 cột dữ liệu kiểu số (`int64`, `float64`) và 2 cột kiểu `object`

```
[ ] proteins_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 232741 entries, 0 to 232740
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   visit_id    232741 non-null object
1   visit_month 232741 non-null int64
2   patient_id  232741 non-null int64
3   UniProt     232741 non-null object
4   NPX         232741 non-null float64
dtypes: float64(1), int64(2), object(2)
memory usage: 8.9+ MB
```

Mô tả thống kê cho dữ liệu kiểu số:

```
numerical_features = ['visit_month', 'patient_id', 'NPX']
round(proteins_df[numerical_features].describe().T, 2)
```

	count	mean	std	min	25%	50%	75%	max
<b>visit_month</b>	232741.0	26.10	22.87	0.00	6.0	24.0	48.0	108.0
<b>patient_id</b>	232741.0	32593.88	18608.48	55.00	16566.0	29313.0	49995.0	65043.0
<b>NPX</b>	232741.0	2712076.94	22241547.32	84.61	29464.4	113556.0	563894.0	613851000.0

Ta thấy cột `NPX` có giá trị min nhỏ trong khi giá trị max rất lớn nên cần chuẩn hóa lại dữ liệu trước khi đưa vào mô hình.

Mô tả thống kê cho dữ liệu kiểu `object`:

	visit_id	UniProt
<b>count</b>	232741	232741
<b>unique</b>	1113	227
<b>unique_vals</b>	[55_0, 1517_0, 1923_0, 2660_0, 3636_0, 3863_0,...	[O00391, O00533, O00584, O14498, O14773, O1479...
<b>top</b>	27715_36	O15240

## KHÁM PHÁ DỮ LIỆU

### Tập dữ liệu khám lâm sàng

Tập dữ liệu chứa thông tin khám lâm sàng của các bệnh nhân, bao gồm các chỉ số đánh giá tiến triển của bệnh qua thời gian. Thông tin này giúp nắm bắt sự thay đổi về chức năng và các biến chứng của bệnh nhân trong quá trình tiến triển của bệnh Parkinson.

	visit_id	patient_id	visit_month	updrs_1	updrs_2	updrs_3	updrs_4	upd23b_clinical_state_on_medication
0	55_0	55	0	10.0	6.0	15.0	NaN	Unknow
1	55_3	55	3	10.0	7.0	25.0	NaN	Unknow
2	55_6	55	6	8.0	10.0	34.0	NaN	Unknow
3	55_9	55	9	8.0	9.0	30.0	0.0	On
4	55_12	55	12	10.0	10.0	41.0	0.0	On

Tập dữ liệu này 2588 hàng và 8 cột thuộc tính.

```
[ ] clinical_df.shape  
  
(2588, 8)
```

**Nhận xét:** Tập dữ liệu có 2588 hàng và 8 cột thuộc tính.

Mô tả về các cột dữ liệu như sau:

Tên cột	Ý nghĩa
visit_id	Mã id của lần khám
month_id	Tháng khám, tính từ thời điểm lần đầu bệnh nhân đến khám
patient_id	Mã bệnh nhân
updrs_n	Điểm của bệnh nhân tương ứng với 4 phần theo MDS-UPDRS ( <a href="#">Unified Parkinson's Disease Rating Scale</a> ). Điểm càng cao thì triệu chứng bệnh càng nghiêm trọng.
upd23b_clinical_state_on_medication	Việc bệnh nhân có sử dụng thuốc (VD: Levodopa) trong quá trình đánh giá UDPRS không?



## KHÁM PHÁ DỮ LIỆU

Tập dữ liệu này có 6 cột dữ liệu kiểu số (`int64`, `float64`) và 2 cột kiểu `object`

```
clinical_df.info();

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2588 entries, 0 to 2614
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   visit_id                             2588 non-null   object
1   patient_id                           2588 non-null   int64
2   visit_month                           2588 non-null   int64
3   updrs_1                               2588 non-null   float64
4   updrs_2                               2588 non-null   float64
5   updrs_3                               2588 non-null   float64
6   updrs_4                               1562 non-null   float64
7   upd23b_clinical_state_on_medication  2588 non-null   object
dtypes: float64(4), int64(2), object(2)
memory usage: 182.0+ KB
```

Mô tả thống kê cho dữ liệu kiểu số:

	count	mean	std	min	25%	50%	75%	max
patient_id	2588.0	32609.27	18541.03	55.0	16572.0	29313.0	50611.0	65043.0
visit_month	2588.0	31.04	25.10	0.0	9.0	24.0	48.0	108.0
updrs_1	2588.0	7.11	5.52	0.0	3.0	6.0	10.0	33.0
updrs_2	2588.0	6.73	6.34	0.0	1.0	5.0	10.0	40.0
updrs_3	2588.0	19.43	15.00	0.0	6.0	19.0	29.0	86.0
updrs_4	1562.0	1.87	3.02	0.0	0.0	0.0	3.0	20.0

Mô tả thống kê cho dữ liệu kiểu object:

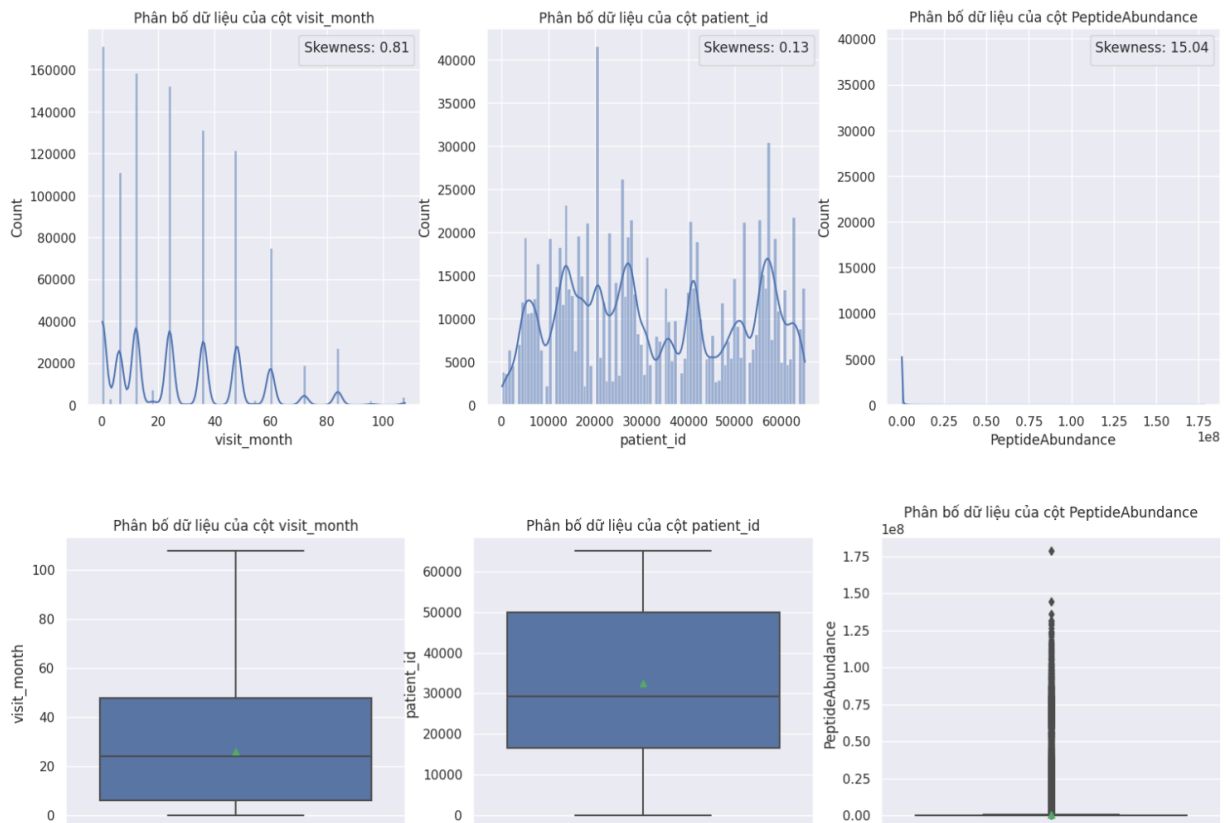
	visit_id	upd23b_clinical_state_on_medication
count	2588	2588
unique	2588	3
unique_vals	[55_0, 55_3, 55_6, 55_9, 55_12, 55_18, 55_24, ...]	[Unknow, On, Off]
top	10053_0	Unknow

## TRỰC QUAN HÓA DỮ LIỆU

Các bước xử lý trong phân khám phá dữ liệu là tương đồng nhau ở cả Kaggle Notebooj và Colab Notebook

### Tập dữ liệu peptide

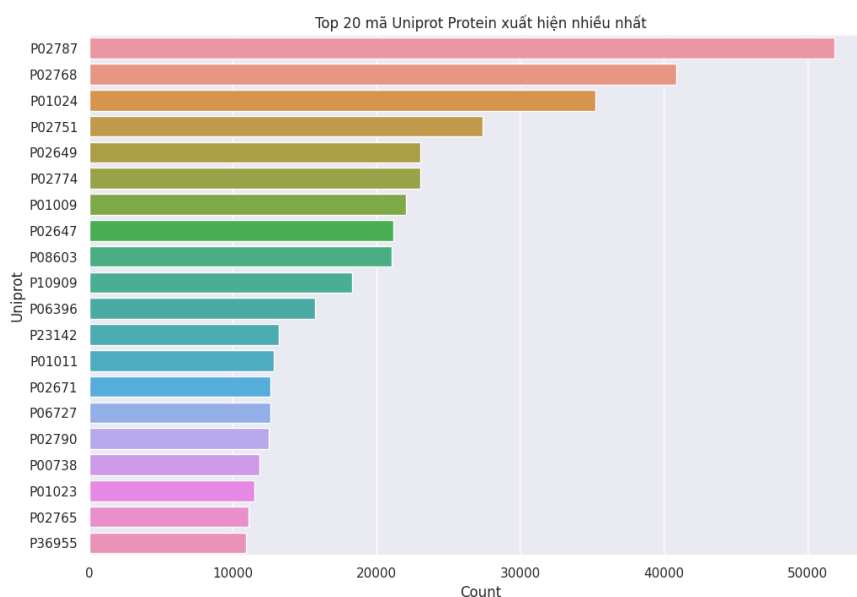
#### Phân bố của các cột dữ liệu kiểu số



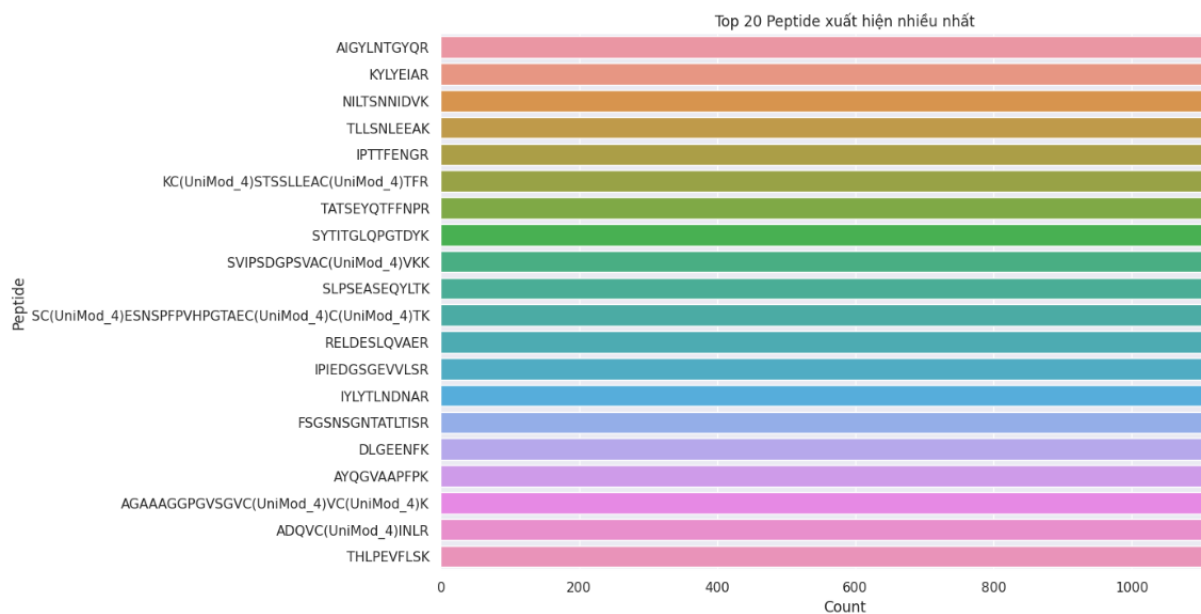
Phân bố của cột dữ liệu `PeptideAbundance` bị lệch và có outlier nên cần chuẩn hóa lại trước khi đưa vào mô hình.

## TRỰC QUAN HÓA DỮ LIỆU

### Top 20 mã Uniprot protein xuất hiện nhiều nhất?

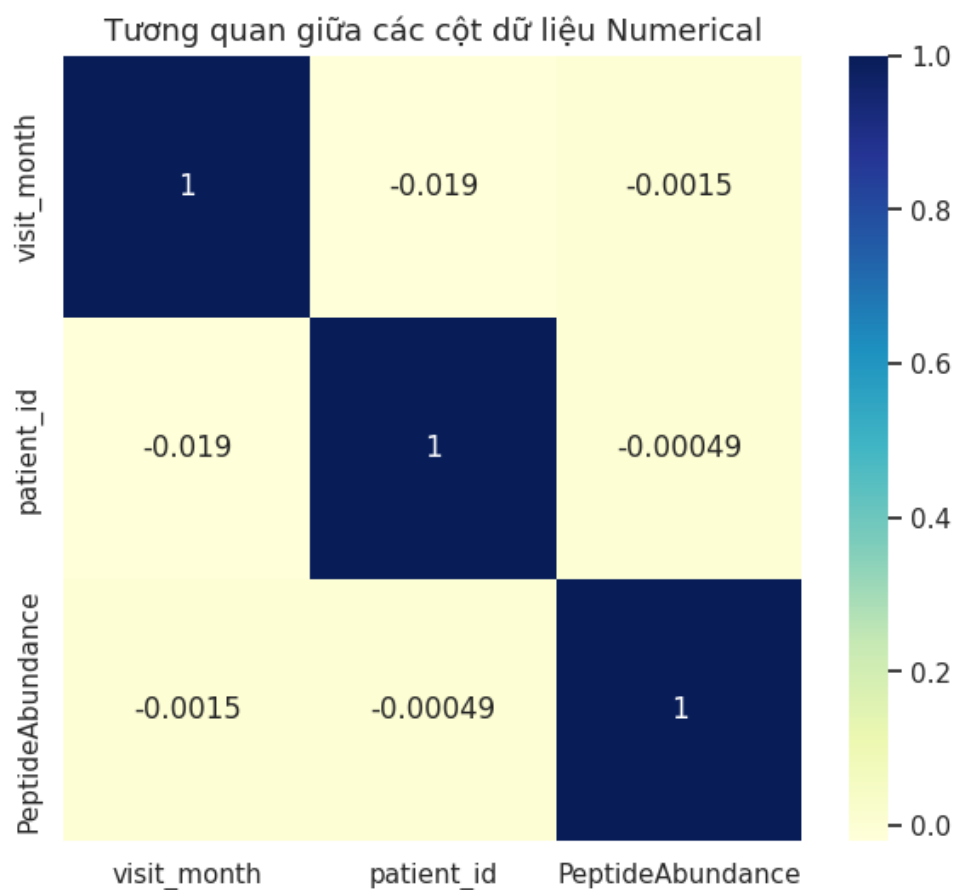


### Top 20 Peptide xuất hiện nhiều nhất?



## TRỰC QUAN HÓA DỮ LIỆU

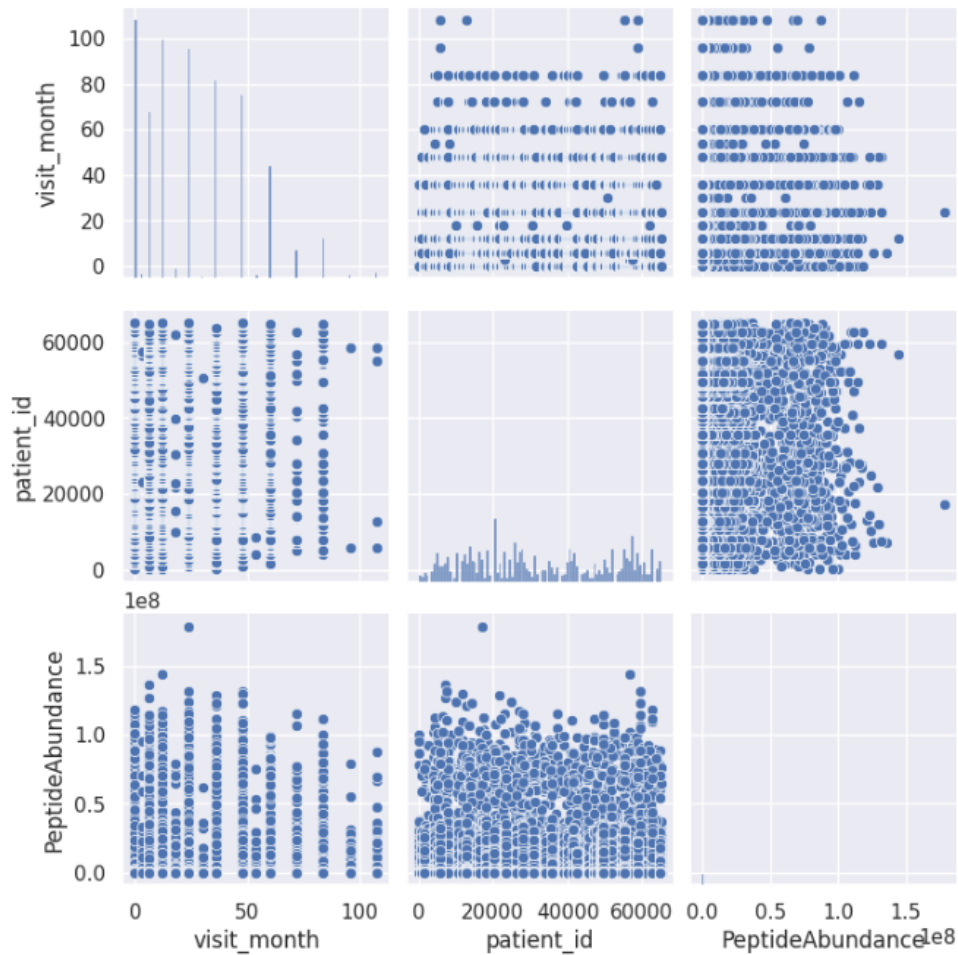
Hệ số tương quan giữa các cột dữ liệu



Không có mối tương quan đặc biệt nào giữa các cột dữ liệu kiểu số.

## TRỰC QUAN HÓA DỮ LIỆU

Mối quan hệ giữa các cột dữ liệu



Không có mối quan hệ rõ ràng nào giữa các cột dữ liệu kiểu số.

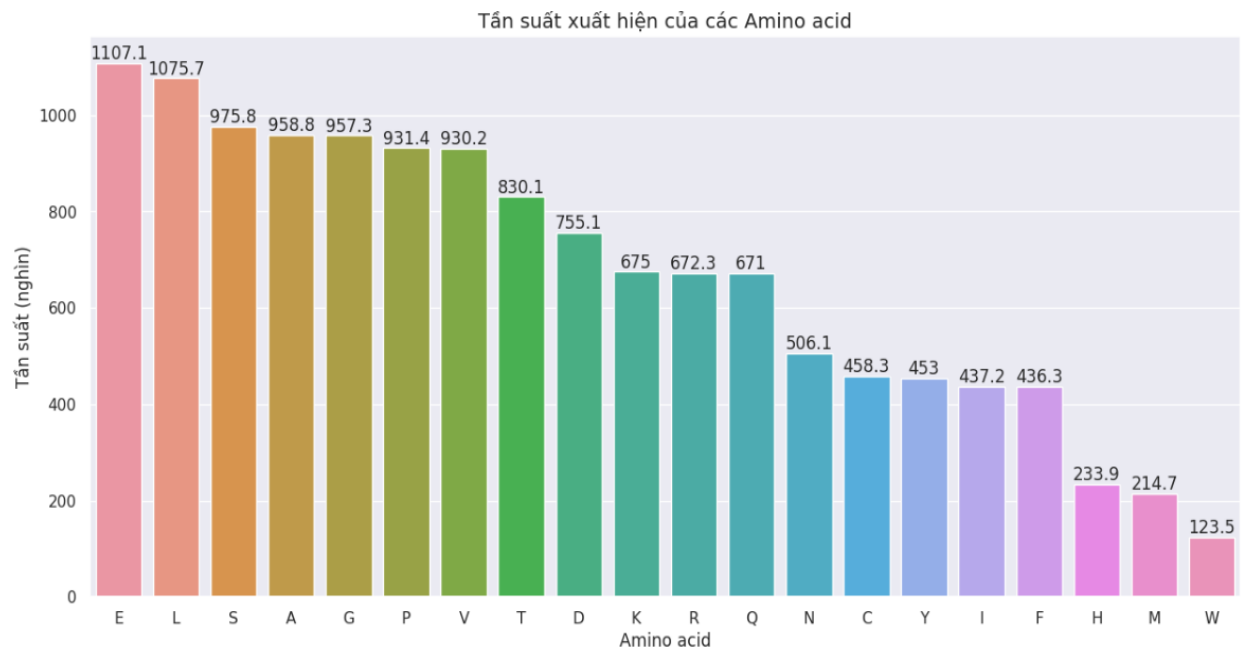
**Những Amino Acid nào xuất hiện nhiều nhất?**

Amino acid là thành phần của Peptide. Peptide là những hợp chất chứa từ 2 đến 50 gốc  $\alpha$  - amino acid liên kết với nhau bằng các liên kết peptide.

## TRỰC QUAN HÓA DỮ LIỆU

Dưới đây là 20 nhóm amino acid phổ biến:

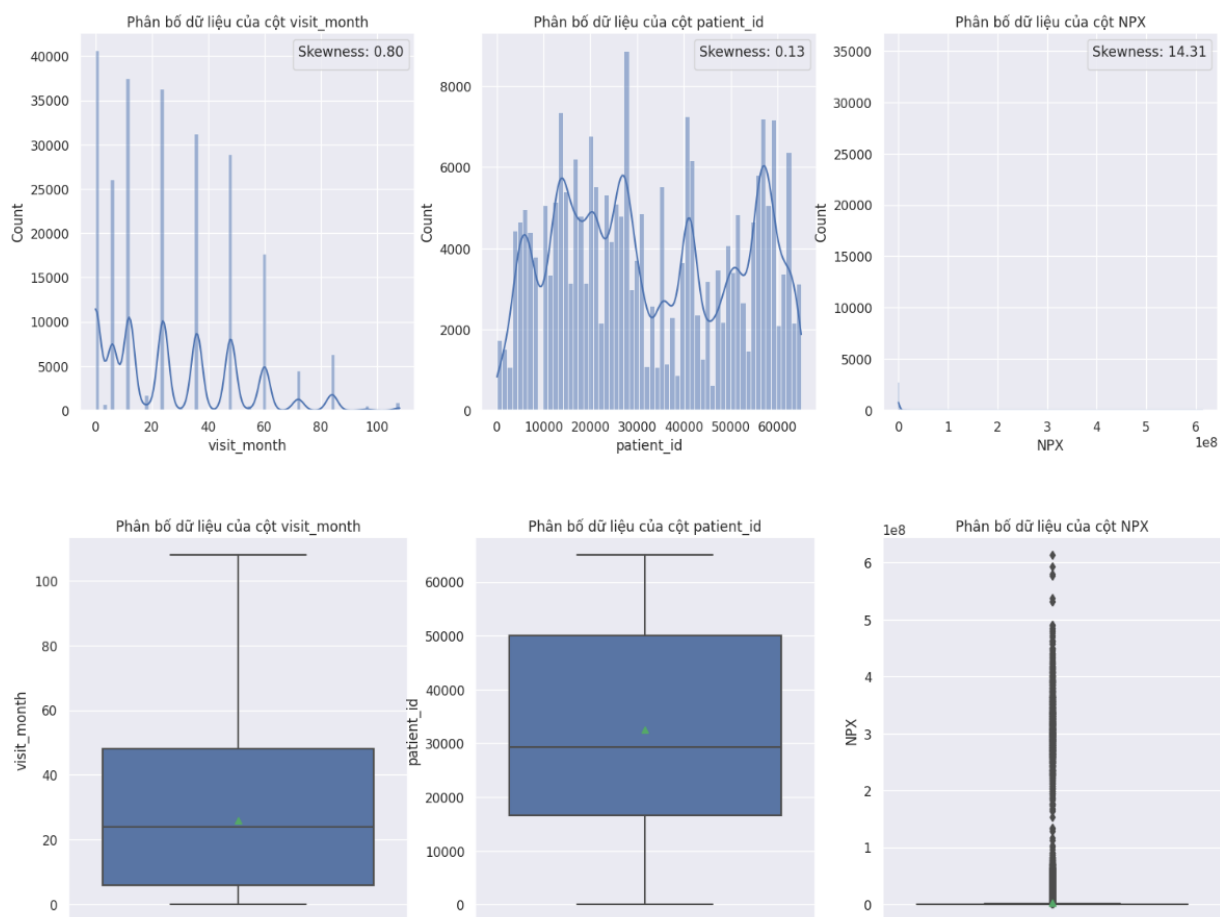
- Alanine (Ala): A
- Arginine (Arg): R
- Asparagine (Asn): N
- Aspartic acid (Asp): D
- Cysteine (Cys): C
- Glutamine (Gln): Q
- Glutamic acid (Glu): E
- Glycine (Gly): G
- Histidine (His): H
- Isoleucine (Ile): I
- Leucine (Leu): L
- Lysine (Lys): K
- Methionine (Met): M
- Phenylalanine (Phe): F
- Proline (Pro): P
- Serine (Ser): S
- Threonine (Thr): T
- Tryptophan (Trp): W
- Tyrosine (Tyr): Y
- Valine (Val): V



## TRỰC QUAN HÓA DỮ LIỆU

### Tập dữ liệu protein

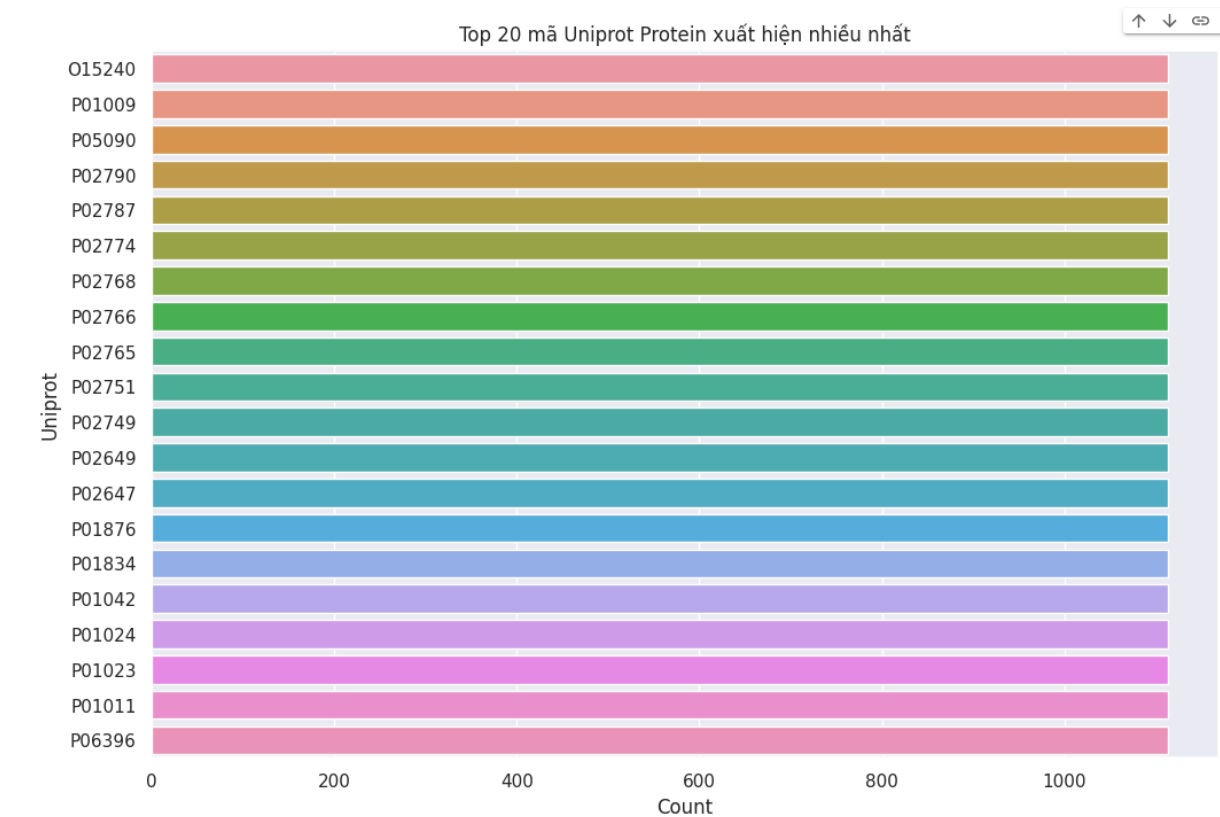
#### Phân bố của các cột dữ liệu kiểu số



Phân bố của cột dữ liệu `NPX` bị lệch và có outlier nên cần chuẩn hóa lại trước khi đưa vào mô hình.

# TRỰC QUAN HÓA DỮ LIỆU

## Top 20 mã Uniprot Protein xuất hiện nhiều nhất?



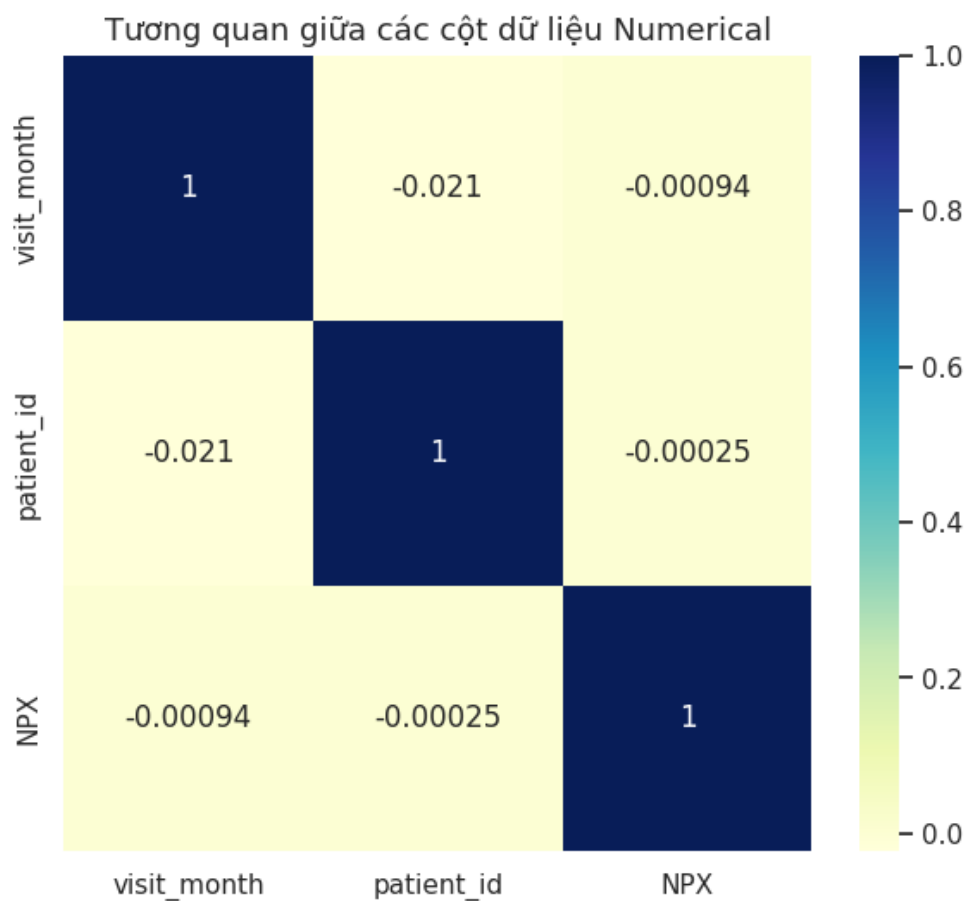


---

## TRỰC QUAN HÓA DỮ LIỆU

---

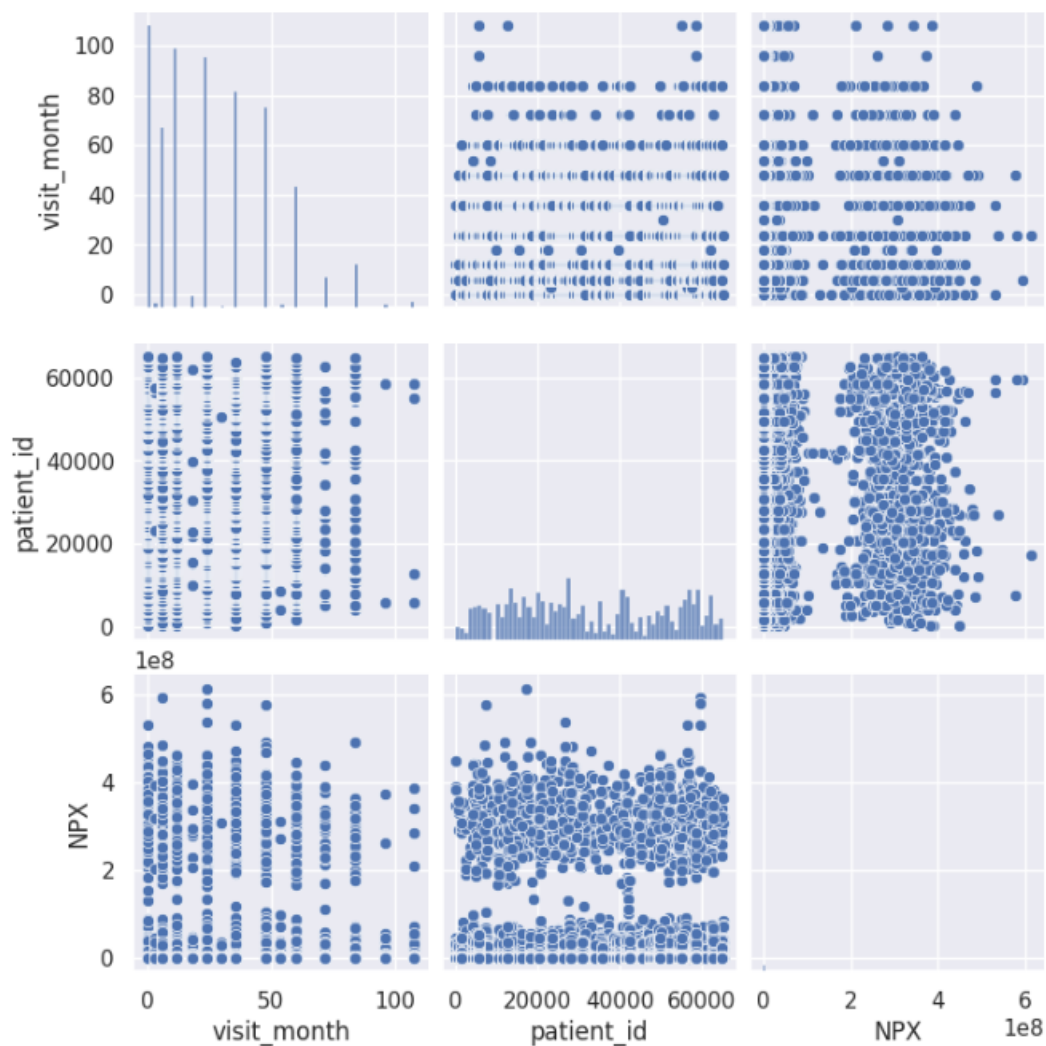
Tương quan giữa các cột dữ liệu



Không có mối tương quan đặc biệt nào giữa các cột dữ liệu kiểu số.

## TRỰC QUAN HÓA DỮ LIỆU

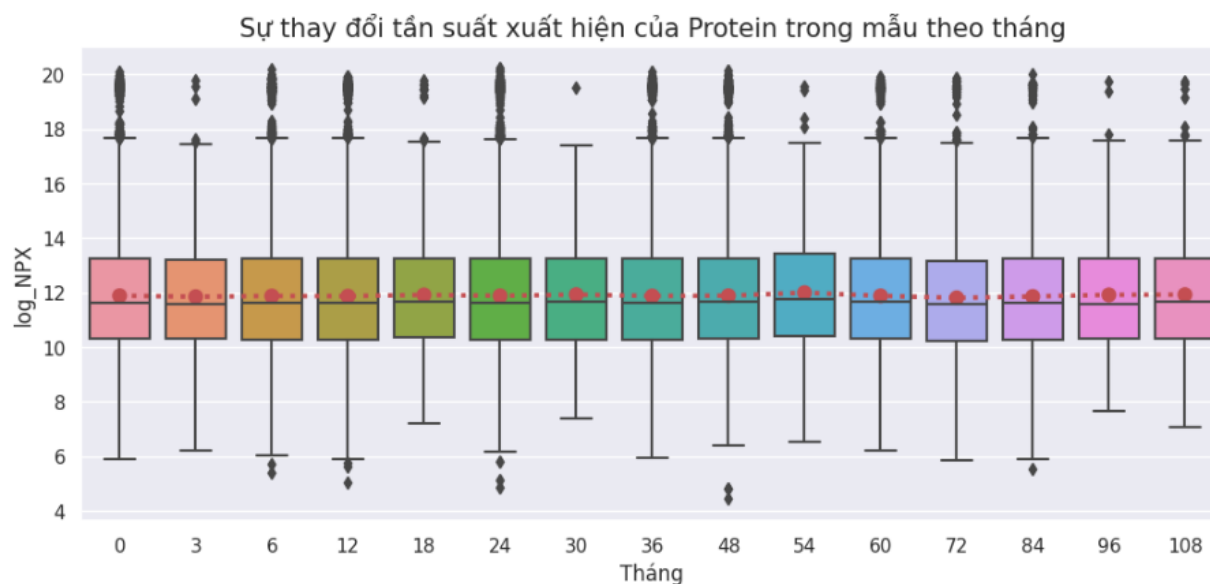
Mối quan hệ giữa các cột dữ liệu



Không có mối tương quan rõ ràng nào giữa các cột dữ liệu kiểu số.

## TRỰC QUAN HÓA DỮ LIỆU

Sự thay đổi của tần suất xuất hiện của Protein trong mẫu (NPX) như thế nào?

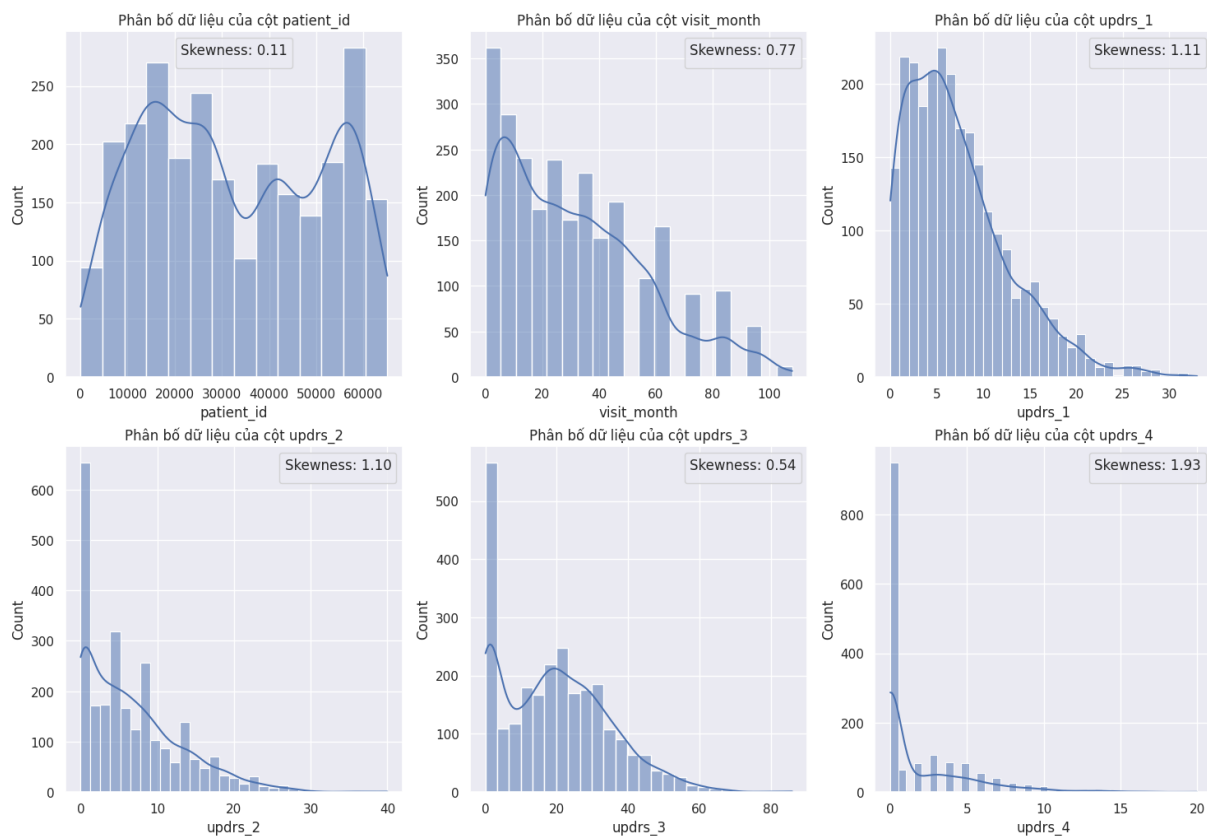


Theo biểu đồ bên trên, tần suất xuất hiện của Protein trong mẫu đánh giá không đổi theo thời gian.

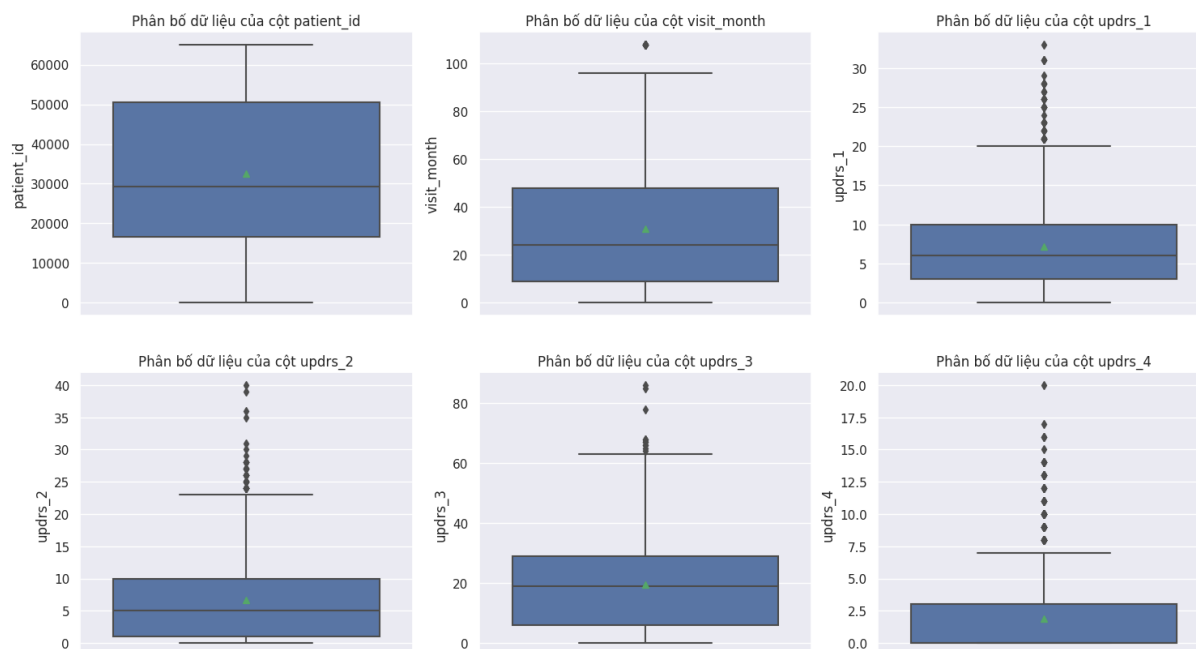
## TRỰC QUAN HÓA DỮ LIỆU

### Tập dữ liệu khám lâm sàng

#### Phân bố của các cột dữ liệu kiểu số



## TRỰC QUAN HÓA DỮ LIỆU



Phân phối của các cột dữ liệu `updrs_1` và `updrs_4` lệch phải.

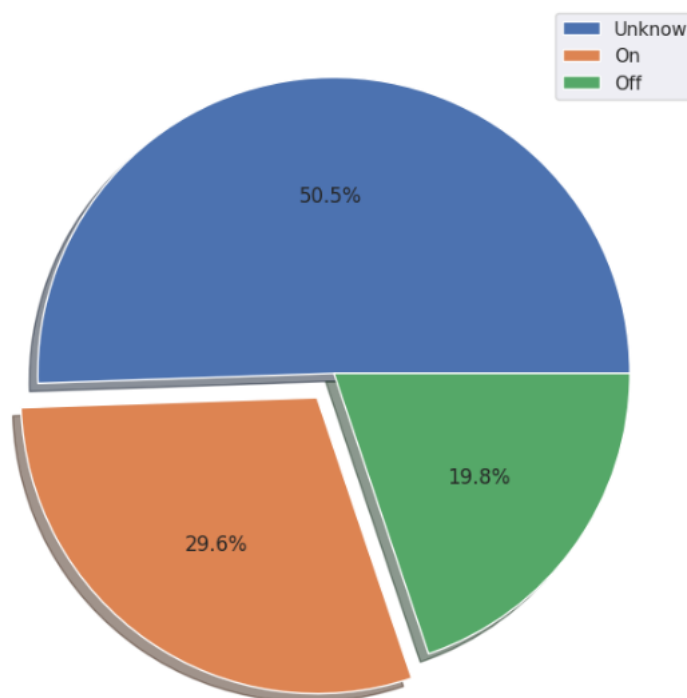
---

## TRỰC QUAN HÓA DỮ LIỆU

---

**Tỉ lệ phần trăm bệnh nhân có sử dụng thuốc hay không trong lúc đánh giá?**

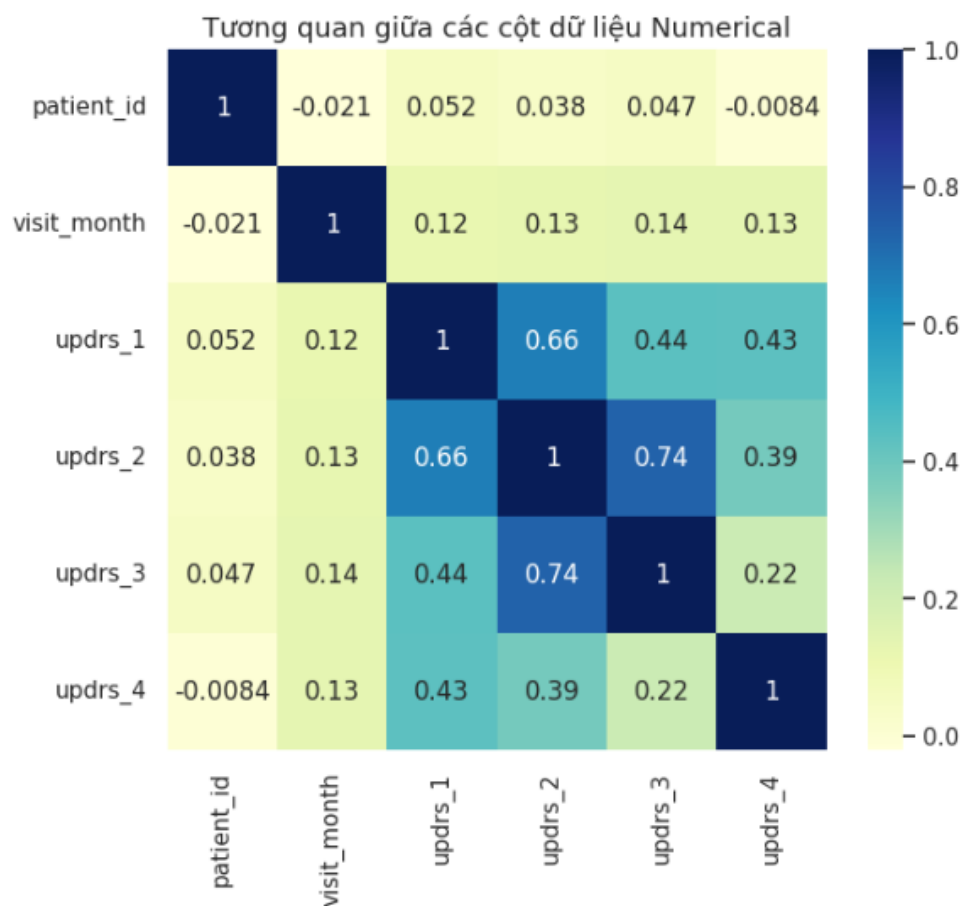
Tỉ lệ phần trăm bệnh nhân có sử dụng thuốc hay không trong lúc đánh giá



Tỉ lệ không xác định được bệnh nhân có sử dụng thuốc trong quá trình đánh giá hay không chiếm phần lớn. Kế đến là có sử dụng thuốc trong quá trình đánh giá với 29.6% và không sử dụng thuốc là 19.8%

## TRỰC QUAN HÓA DỮ LIỆU

Tương quan giữa các cột dữ liệu

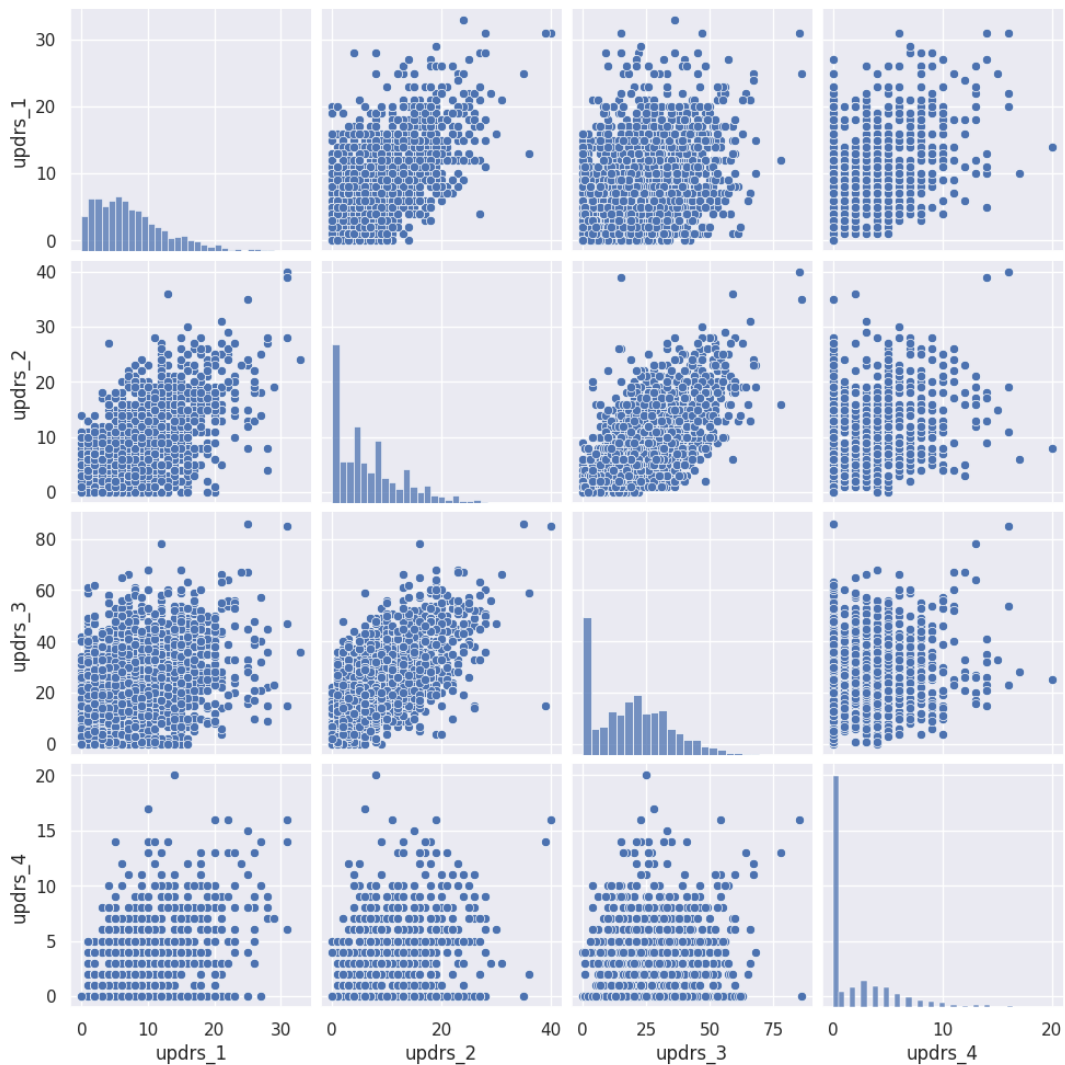


Có mối quan hệ tương quan giữa các cặp dữ liệu điểm UPDRS. Đặc biệt 2 cặp thuộc tính  $updrs\_1 - updrs\_2$  và  $updrs\_2 - updrs\_3$  có hệ số tương quan lớn.

## TRỰC QUAN HÓA DỮ LIỆU

### Mối quan hệ giữa các cột dữ liệu

Ta thấy được có mối quan hệ tương quan giữa các cặp biến đích. Ta sẽ tiến hành trực quan hóa để thấy rõ hơn những mối quan hệ này.

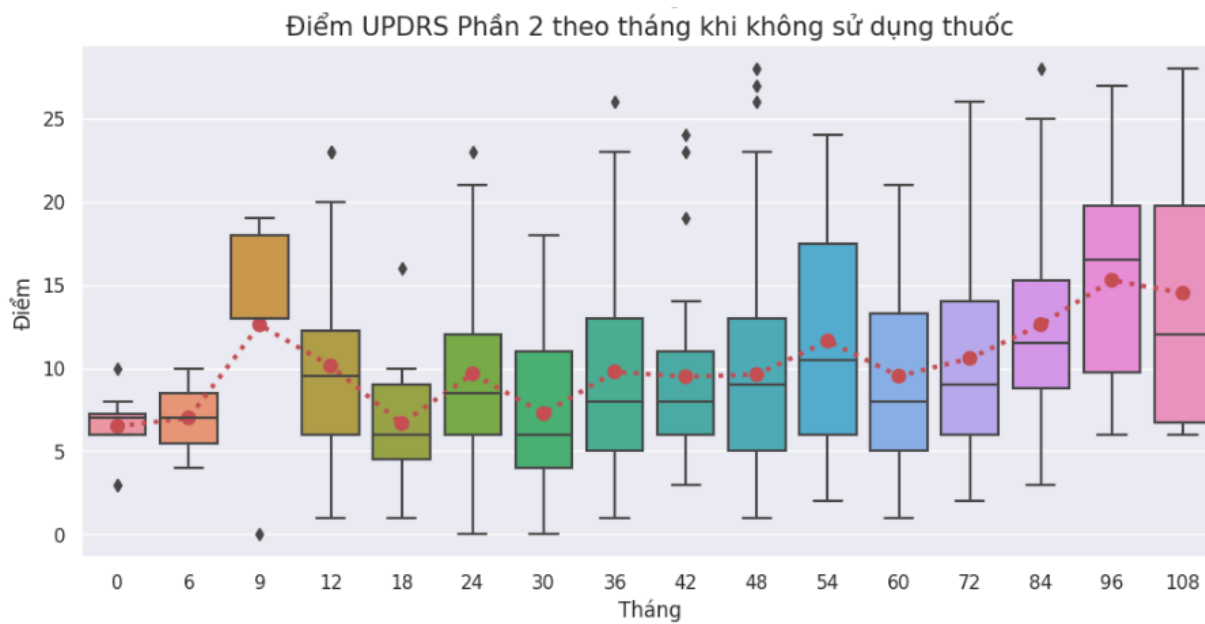
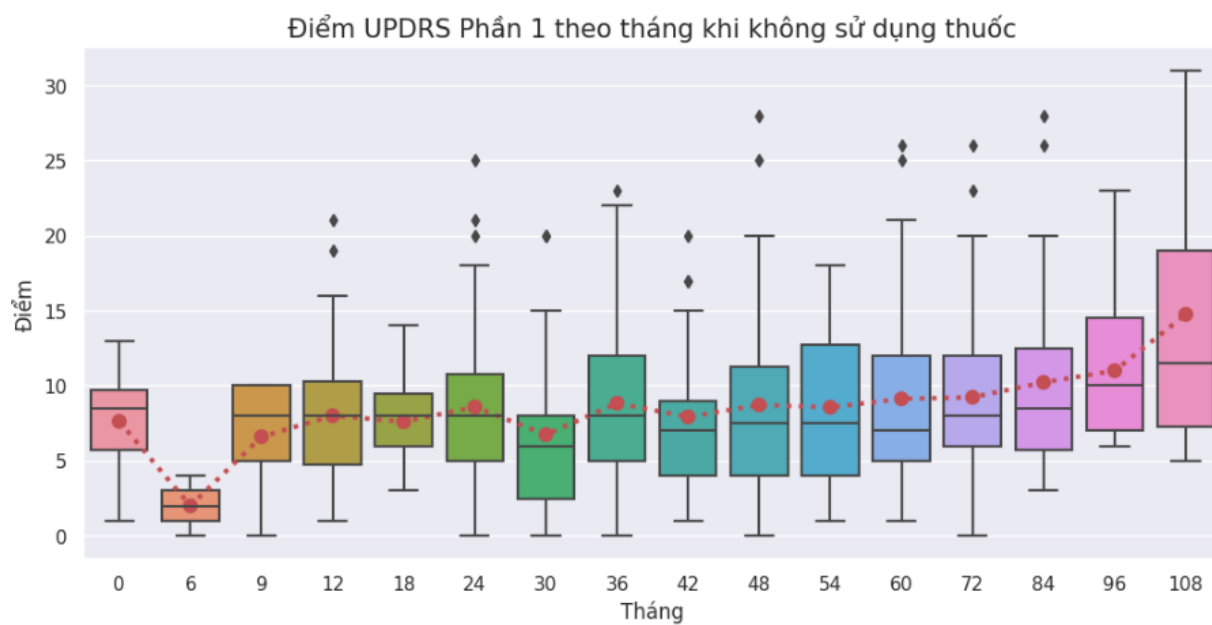


2 cặp thuộc tính  $\text{updrs}_1 - \text{updrs}_2$  và  $\text{updrs}_2 - \text{updrs}_3$  có mối quan hệ cùng tăng.

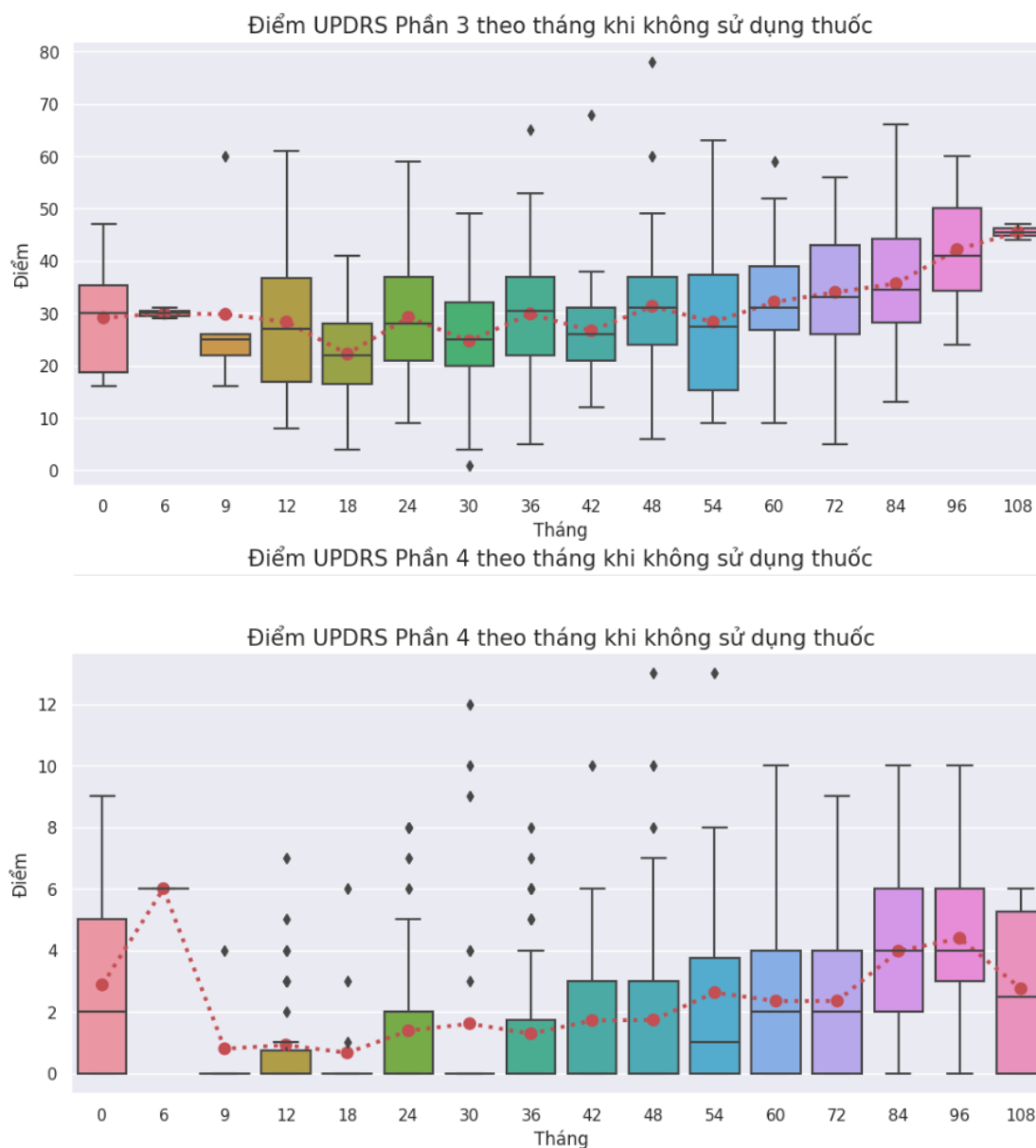


## TRỰC QUAN HÓA DỮ LIỆU

Sự thay đổi của tình trạng bệnh theo thời gian như thế nào?



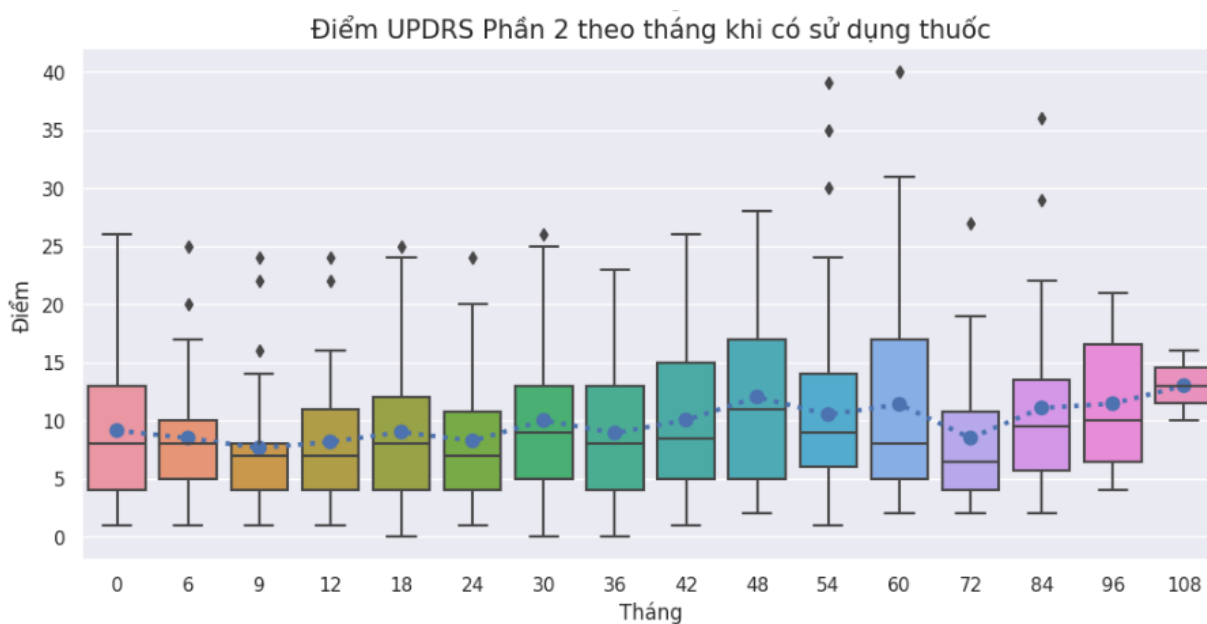
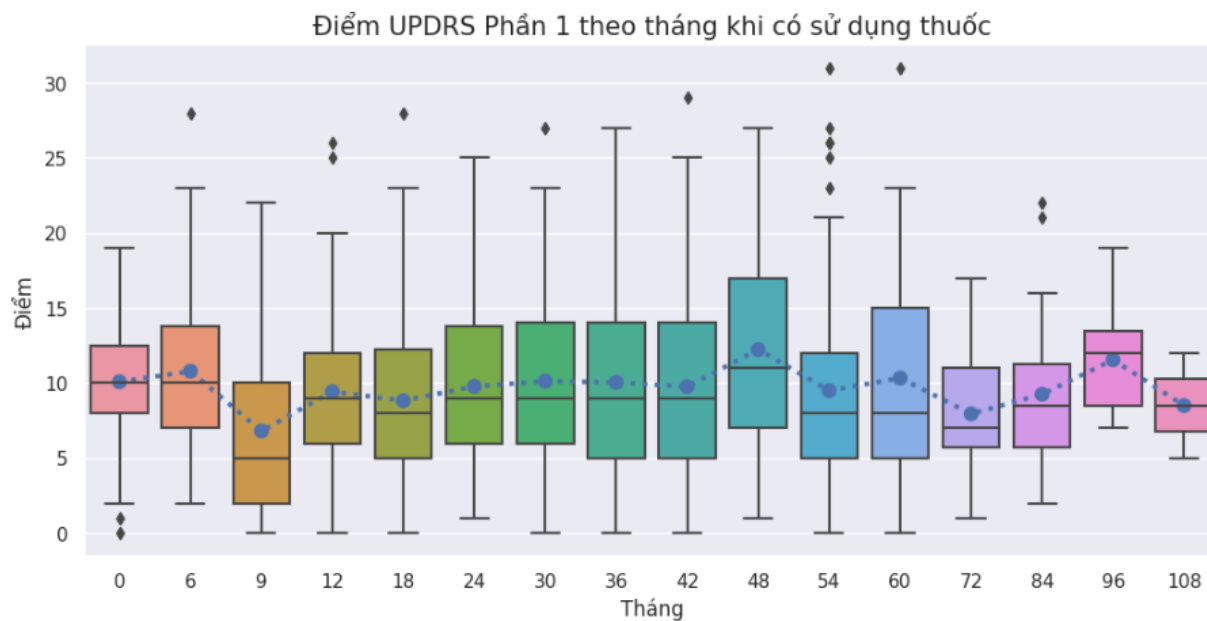
## TRỰC QUAN HÓA DỮ LIỆU



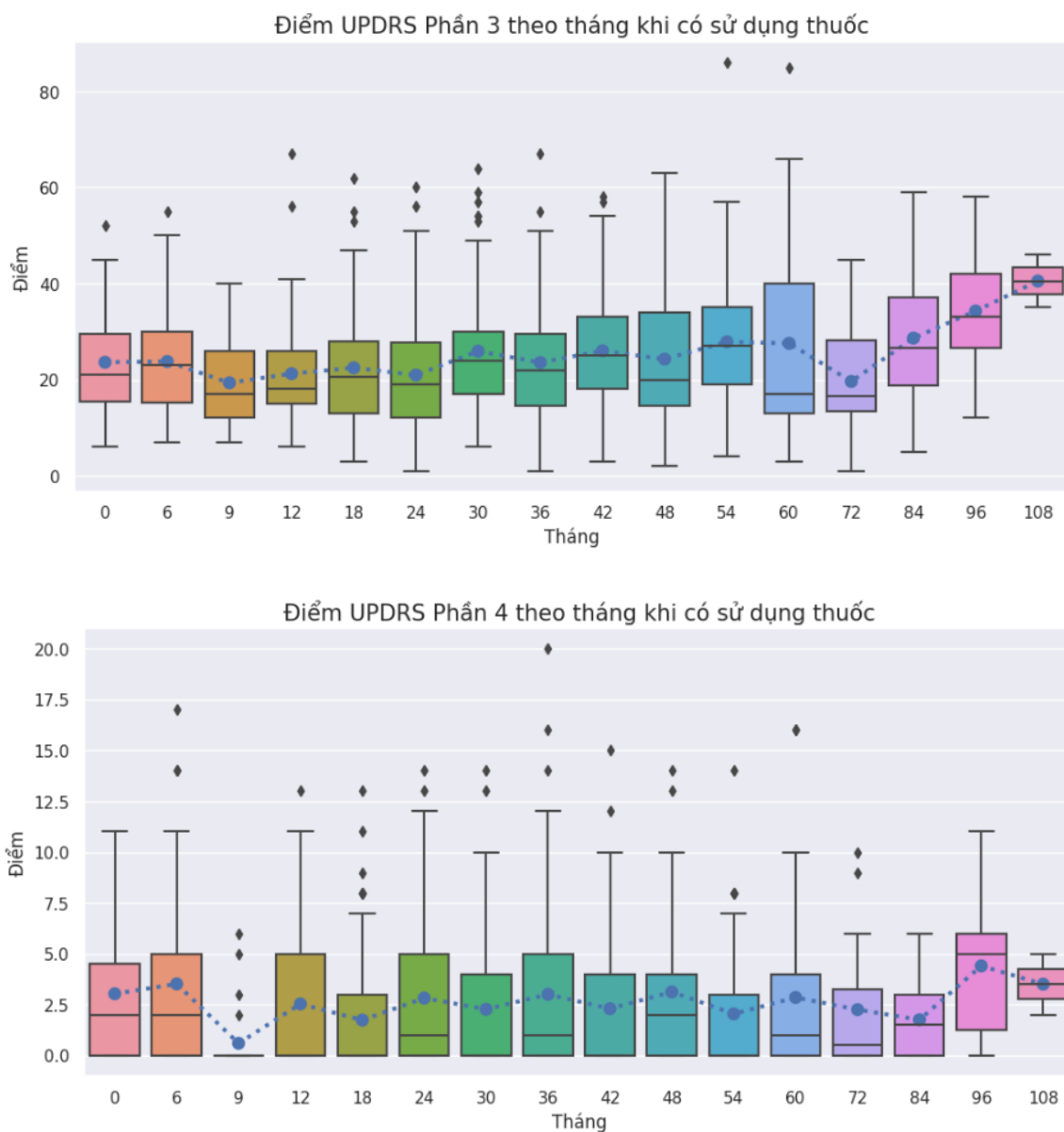
Từ các biểu đồ trên rút ra được những nhận xét sau về tình trạng bệnh theo thời gian khi không sử dụng thuốc trong quá trình đánh giá:

- Điểm UDPRS phần 1, 2, 3 có xu hướng tăng nhẹ theo thời gian.
- Điểm UDPRS phần 4 tăng sau 6 tháng sau đó giảm, sau 9 tháng có xu hướng tăng nhẹ theo thời gian.

## TRỰC QUAN HÓA DỮ LIỆU



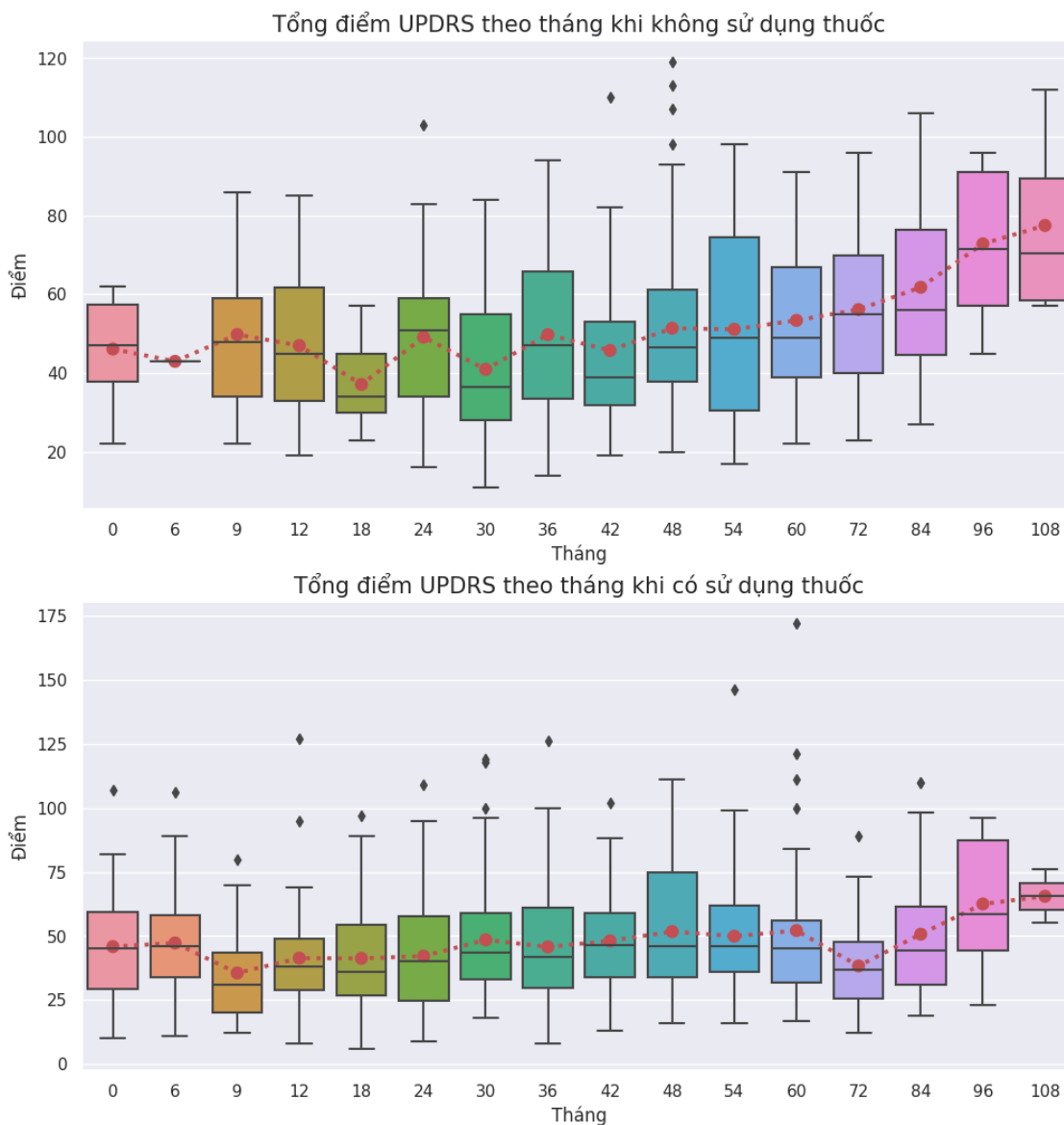
## TRỰC QUAN HÓA DỮ LIỆU



Từ các biểu đồ trên rút ra được những nhận xét sau về tình trạng bệnh theo thời gian khi có sử dụng thuốc trong quá trình đánh giá:

- Điểm UDPRS phần 1, 2, 4 gần như không đổi theo thời gian.
- Điểm UDPRS phần 3 có xu hướng tăng nhẹ theo thời gian.

## TRỰC QUAN HÓA DỮ LIỆU

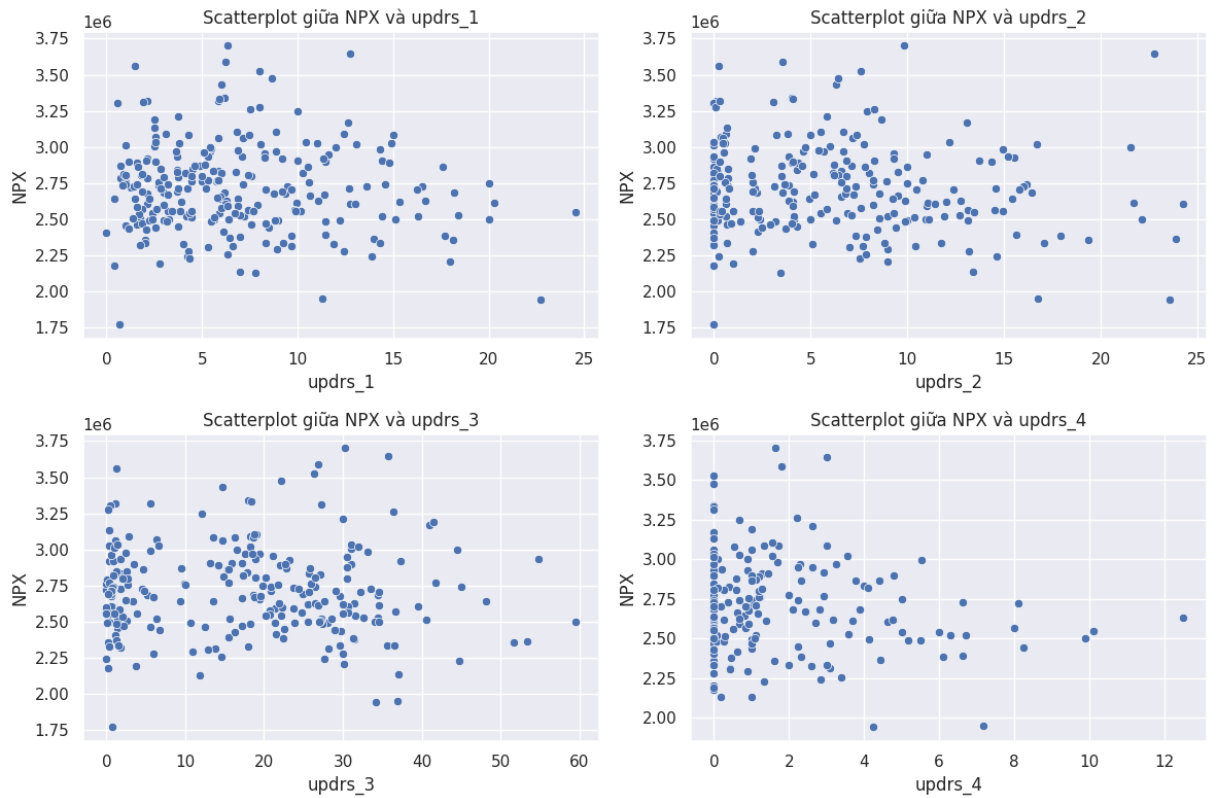


Xét về tổng điểm cả 4 phần:

- Tổng điểm UPDRS khi không sử dụng thuốc trong quá trình đánh giá tăng theo thời gian.
- Tổng điểm UPDRS khi có sử dụng thuốc trong quá trình đánh giá gần như không đổi theo thời gian.

## TRỰC QUAN HÓA DỮ LIỆU

Mối quan hệ giữa tần suất xuất hiện protein(NPX) và các cột điểm đánh giá



Giữa số điểm đánh giá các phần và tần suất xuất hiện của protein không thực sự có mối liên hệ, sự tương quan nào rõ ràng.

## RÚT TRÍCH ĐẶC TRƯNG & CHUẨN BỊ DỮ LIỆU HUẤN LUYỆN

---

Các bước xử lý trong phần **Rút trích đặc trưng và chuẩn bị dữ liệu huấn luyện** là tương đồng nhau ở cả Kaggle Notebook và Colab Notebook.

### Xử lý dữ liệu bị thiếu

Như đã đề cập ở phần **Chuẩn bị dữ liệu bên trên**, dữ liệu bị thiếu hiện tại là cột `updrs_4` của tập dữ liệu khám lâm sàng `clinical_df`.

Theo như phân bố của cột dữ liệu `updrs_4` trong phần **Trực quan hóa dữ liệu** bên trên, phần lớn các giá trị của cột dữ liệu này là **0**. Do đó, nhóm quyết định điền những giá trị bị thiếu bằng giá trị 0.

```
clinical_df1 = clinical_df.copy()
clinical_df1.isna().sum()
```

```
visit_id          0
patient_id        0
visit_month       0
updrs_1           0
updrs_2           0
updrs_3           0
updrs_4          1026
upd23b_clinical_state_on_medication  0
dtype: int64
```

```
clinical_df1['updrs_4'] = clinical_df1['updrs_4'].fillna(0)
clinical_df1.isna().sum()
```

```
visit_id          0
patient_id        0
visit_month       0
updrs_1           0
updrs_2           0
updrs_3           0
updrs_4           0
upd23b_clinical_state_on_medication  0
dtype: int64
```

## RÚT TRÍCH ĐẶC TRƯNG & CHUẨN BỊ DỮ LIỆU HUẤN LUYỆN

### Chuẩn bị dữ liệu huấn luyện

Nhóm sẽ tiến hành kết hợp 2 bảng dữ liệu protein (`proteins_df`) và peptide (`peptides_df`) lại với nhau. Sau đó trộn với tập dữ liệu khám lâm sàng (`clinical_df`) để đưa các biến đích cần dự đoán vào tập dữ liệu

Các cột dữ liệu `visit_id`, `patient_id` (không cần thiết) và `upd23b_clinical_state_on_medication` (thiếu nhiều dữ liệu ~50.57%) sẽ được loại bỏ khỏi tập dữ liệu.

```
def prepare_train_data(proteins_df, peptides_df):
    # Gộp nhóm 2 bảng dữ liệu protein và peptide theo 'visit_id' và 'UniProt'
    proteins_by_uniprot_df = proteins_df.groupby(['visit_id', 'UniProt'])['NPX'].mean().reset_index()
    peptides_by_peptide_df = peptides_df.groupby(['visit_id', 'Peptide'])['PeptideAbundance'].mean().reset_index()

    # Tạo bảng pivot từ 2 bảng dữ liệu thu được bên trên
    proteins_by_uniprot_pivot_df = proteins_by_uniprot_df.pivot(index='visit_id', columns='UniProt', values='NPX').rename_axis(columns=None).reset_index()
    peptides_by_peptide_pivot_df = peptides_by_peptide_df.pivot(index='visit_id', columns='Peptide', values='PeptideAbundance').rename_axis(columns=None).reset_index()

    # Merge 2 bảng dữ liệu lại làm 1
    proteins_peptides_df = proteins_by_uniprot_pivot_df.merge(peptides_by_peptide_pivot_df, on='visit_id', how='left')
    proteins_peptides_df = proteins_peptides_df.fillna(0)

    return proteins_peptides_df

proteins_peptides_df = prepare_train_data(proteins_df, peptides_df)

# Merge bảng trên với đặc trưng đích (updrs_[1-4])
# Loại bỏ cột visit_id, patient_id (do không cần thiết) và upd23b_clinical_state_on_medication (do thiếu dữ liệu ~50.57%)
train_df = proteins_peptides_df.merge(clinical_df1, on='visit_id', how='left').drop(columns=['visit_id', 'patient_id', 'upd23b_clinical_state_on_medication'])
train_df
```

Kết quả thu được như sau:

5394	043505	060888	...	YVNKEIQNAVNGVK	YWGVASFLQK	YYC(UniMod_4)FQGNQFLR	YYTYLIMNK	YYWGGQYTWDMAK	visit_month	updrs_1	updrs_2	updrs_3	updrs_4
13.6	167327.0	129048.0	...	76705.7	104260.0	530223.0	0.0	7207.30	0.0	3.0	0.0	13.0	0.0
99.1	164268.0	108114.0	...	92078.1	123254.0	453883.0	49281.9	25332.80	12.0	4.0	2.0	8.0	0.0
89.6	168107.0	163776.0	...	63203.6	128336.0	447505.0	52389.1	21235.70	18.0	2.0	2.0	0.0	0.0
46.4	204013.0	56725.0	...	89822.1	129964.0	552232.0	65657.8	9876.98	12.0	3.0	6.0	31.0	0.0
38.1	240892.0	85767.1	...	80919.3	111799.0	0.0	56977.6	4903.09	24.0	4.0	7.0	19.0	10.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
35.3	254247.0	138910.0	...	118897.0	133682.0	571879.0	80268.3	54889.70	24.0	11.0	10.0	13.0	2.0
41.3	212132.0	100519.0	...	65966.9	77976.8	486239.0	45032.7	0.00	12.0	5.0	2.0	25.0	0.0
95.7	185836.0	99183.5	...	68505.7	74483.1	561398.0	52916.4	21847.60	24.0	2.0	3.0	23.0	0.0
46.6	167633.0	84875.1	...	63265.8	64601.8	632782.0	51123.7	20700.30	48.0	2.0	6.0	35.0	0.0
23.1	165524.0	85272.7	...	80001.2	79661.9	573300.0	48005.8	15674.10	6.0	8.0	2.0	21.0	0.0



## RÚT TRÍCH ĐẶC TRƯNG & CHUẨN BỊ DỮ LIỆU HUẤN LUYỆN

### Chuẩn hóa dữ liệu

Các giá trị hiện tại khá lớn và để mô hình có thể huấn luyện được tốt hơn ta sẽ tiến hành chuẩn hóa lại dữ liệu trước khi đưa vào mô hình. Phương pháp chuẩn hóa sẽ là [Z-score](#).

Tất nhiên sẽ không chuẩn hóa các cột `visit_month` (giữ nguyên tháng) và `updrs_[1-4]` (biến đích cần dự đoán).

```
std_features = list(train_df.drop(columns=['visit_month', 'updrs_1', 'updrs_2', 'updrs_3', 'updrs_4']).columns)
train_std_df = train_df.copy()
std_scaler = StandardScaler()
train_std_df[std_features] = std_scaler.fit_transform(train_df[std_features])
train_std_df
```

Kết quả thu được như sau:

	000391	000533	000584	014498	014773	014791	015240	015394	043505	060888	...	YVNKEIQNAVNGVK	YWGVASFLLQK	YYC(UniMod_4)FQGNQFLR	YYTYL
0	9104.27	402321.0	0.00	0.0	7150.57	2497.84	83002.9	15113.6	167327.0	129048.0	...	76705.7	104260.0	530223.0	
1	10464.20	435586.0	0.00	0.0	0.00	0.00	197117.0	15099.1	164268.0	108114.0	...	92078.1	123254.0	453883.0	49
2	13235.70	507386.0	7126.96	24525.7	0.00	2372.71	126506.0	16289.6	168107.0	163776.0	...	63203.6	128336.0	447505.0	52
3	12600.20	494581.0	9165.06	27193.5	22506.10	6015.90	156313.0	54546.4	204013.0	56725.0	...	89822.1	129964.0	552232.0	65
4	12003.20	522138.0	4498.51	17189.8	29112.40	2665.15	151169.0	52338.1	240892.0	85767.1	...	80919.3	111799.0	0.0	56
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1108	9983.00	400290.0	24240.10	0.0	16943.50	6303.17	77493.6	46435.3	254247.0	138910.0	...	118897.0	133682.0	571879.0	80
1109	6757.32	360858.0	18367.60	14760.7	18603.40	1722.77	86847.4	37741.3	212132.0	100519.0	...	65966.9	77976.8	486239.0	45
1110	0.00	352722.0	22834.90	23393.1	16693.50	1487.91	114772.0	36095.7	185836.0	99183.5	...	68505.7	74483.1	561398.0	52
1111	11627.80	251820.0	22046.50	26360.5	22440.20	2117.43	82241.9	30146.6	167633.0	84875.1	...	63265.8	64601.8	632782.0	51
1112	11218.70	399518.0	20581.00	31290.9	6173.58	2564.37	160526.0	43423.1	165524.0	85272.7	...	80001.2	79661.9	573300.0	48

### Rút trích đặc trưng

Với mục đích ban đầu của bài toán là dự đoán điểm MDS-UPDRS thông qua dữ liệu đo lường protein và peptide để xác định tiến triển của bệnh Parkinson. Do đó, sẽ giữ lại tất cả các đặc trưng protein và peptide kết hợp với `visit_month` để đưa vào mô hình huấn luyện. Dù số lượng đặc trưng rất lớn nhưng cũng đảm bảo tập luật được đầy đủ nhất có thể.

```
# Tách các protein, peptide và visit_month phục vụ cho việc huấn luyện mô hình
features = list(train_std_df.drop(columns= ['updrs_1', 'updrs_2', 'updrs_3', 'updrs_4']).columns)
len(features)
```

1196

---

## XÂY DỰNG MÔ HÌNH HUẤN LUYỆN

---

Xác định các biến đích cần dự đoán.

```
targets = ['updrs_1', 'updrs_2', 'updrs_3', 'updrs_4']
```

Xây dựng hàm tính giá trị [SMAPE](#). Đây là độ đo đánh giá được sử dụng trong cuộc thi trên [Kaggle](#).

```
def smape(y_true, y_pred):
    smap = np.zeros(len(y_true))

    num = np.abs(y_true - y_pred)
    dem = ((np.abs(y_true) + np.abs(y_pred)) / 2)

    pos_ind = (y_true != 0) | (y_pred != 0)
    smap[pos_ind] = num[pos_ind] / dem[pos_ind]

    return 100 * np.mean(smap)
```

Hai mô hình mà nhóm sẽ sử dụng là [Random Forest Regressor](#) và [LightGBM Regressor](#). Lý do lựa chọn 2 mô hình này là:

- Cả 2 mô hình đều có khả năng xử lý dữ liệu có kích thước lớn hoặc có số đặc trưng lớn.
- Kết quả dự đoán ổn định và tốc độ dự đoán, huấn luyện khá nhanh.
- Xử lý được các trường hợp thiếu dữ liệu
- Cả 2 mô hình đều phù hợp với bài toán hồi quy này.
- Có thể xác định được độ quan trọng của các thuộc tính.

Tập dữ liệu huấn luyện sẽ được chia ra thành tập train và tập validation.

Một vòng lặp sẽ chạy qua từng biến đích để xây dựng mô hình Random Forest Regressor và LightGBM Regressor tương ứng với từng biến đích.

Mô hình huấn luyện, kết quả SMAPE trên tập validation và độ quan trọng của các đặc trưng sẽ được lưu lại để phục vụ cho phần đánh giá và nhận xét.

## ĐÁNH GIÁ

Đánh giá 2 mô hình trên từng biến đích. Sau đó lựa chọn mô hình với kết quả tốt hợp để làm mô hình dự đoán cuối cùng.

Do phương thức đánh giá riêng của 2 mô hình khác nhau nên kết quả có thể không giống nhau. Do đó sẽ sử dụng [SHAP](#) làm nền tảng chung để xem xét mức độ ảnh hưởng của các đặc trưng tới biến đích tương ứng với từng mô hình.

### UPDRS\_1

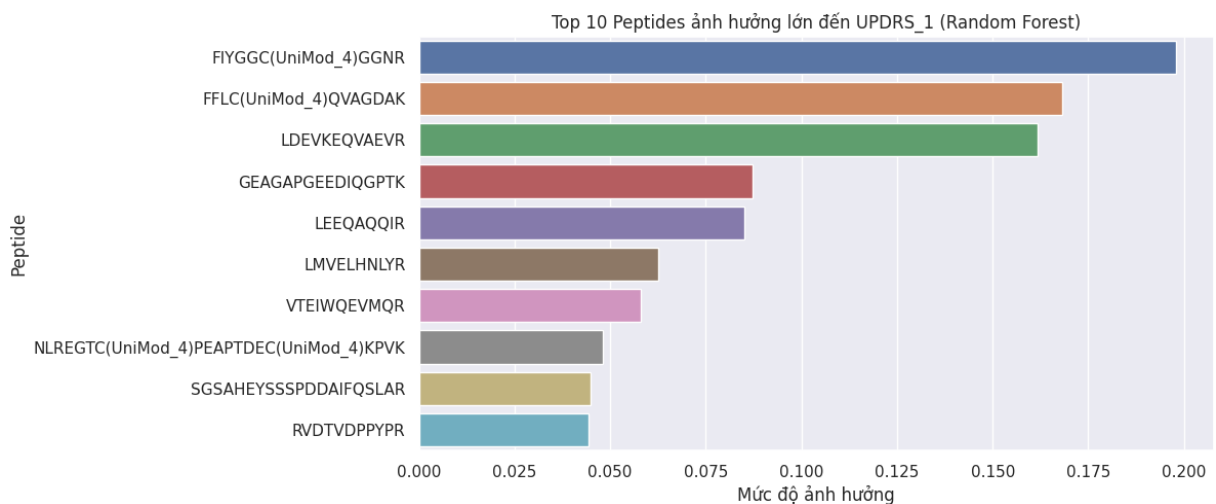
Kết quả SMAPE trên tập validation của 2 mô hình như sau:

```
-----  
Random Forest model UPDRS_1:  
sMAPE test: 67.6423  
-----
```

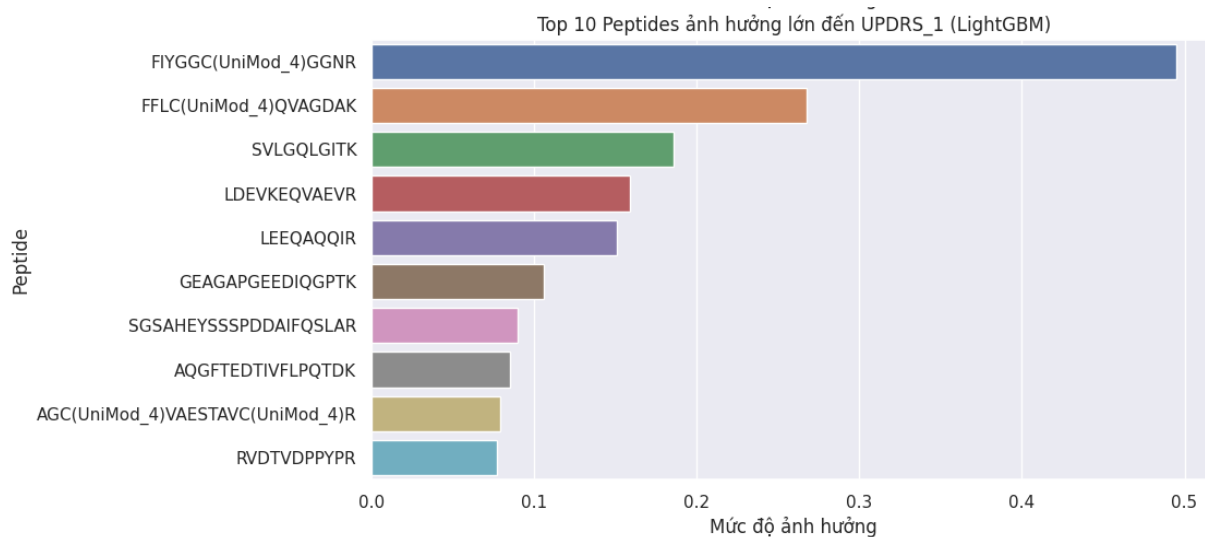
```
LightGBM model UPDRS_1:  
sMAPE test: 65.4288  
-----
```

**Nhận xét:** Mô hình LightGBM Regressor cho kết quả SMAPE trên tập validation tốt hơn.

Xem xét ảnh hưởng của các peptide tới điểm **UPDRS\_1** từ 2 mô hình đã huấn luyện.

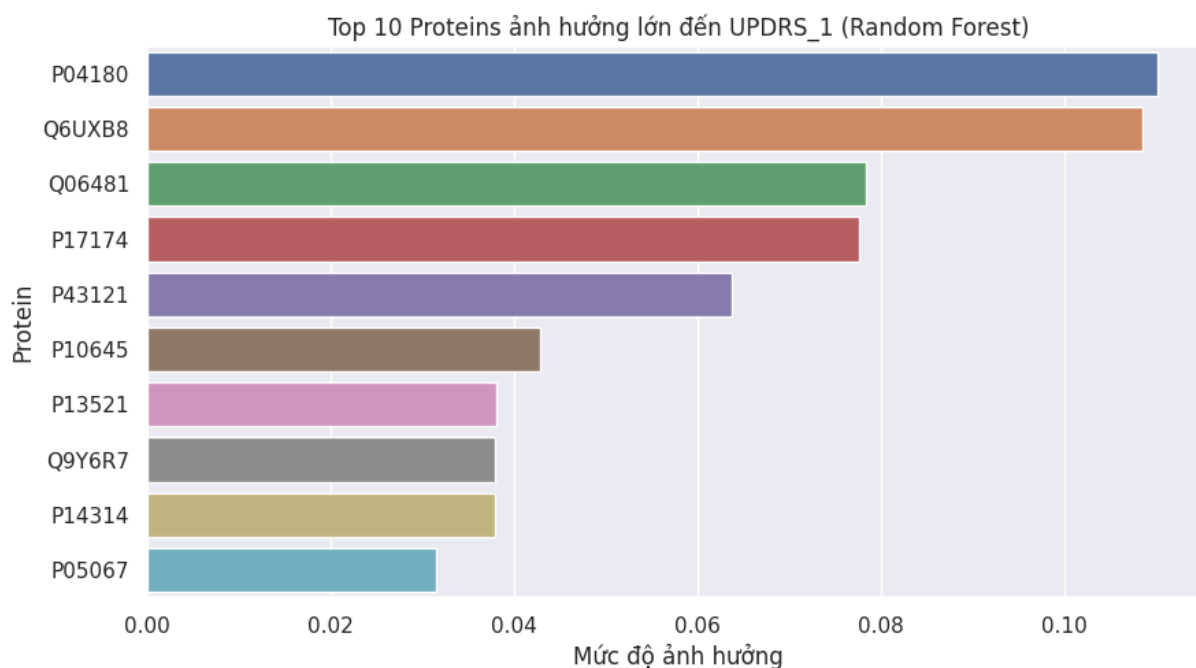


## ĐÁNH GIÁ

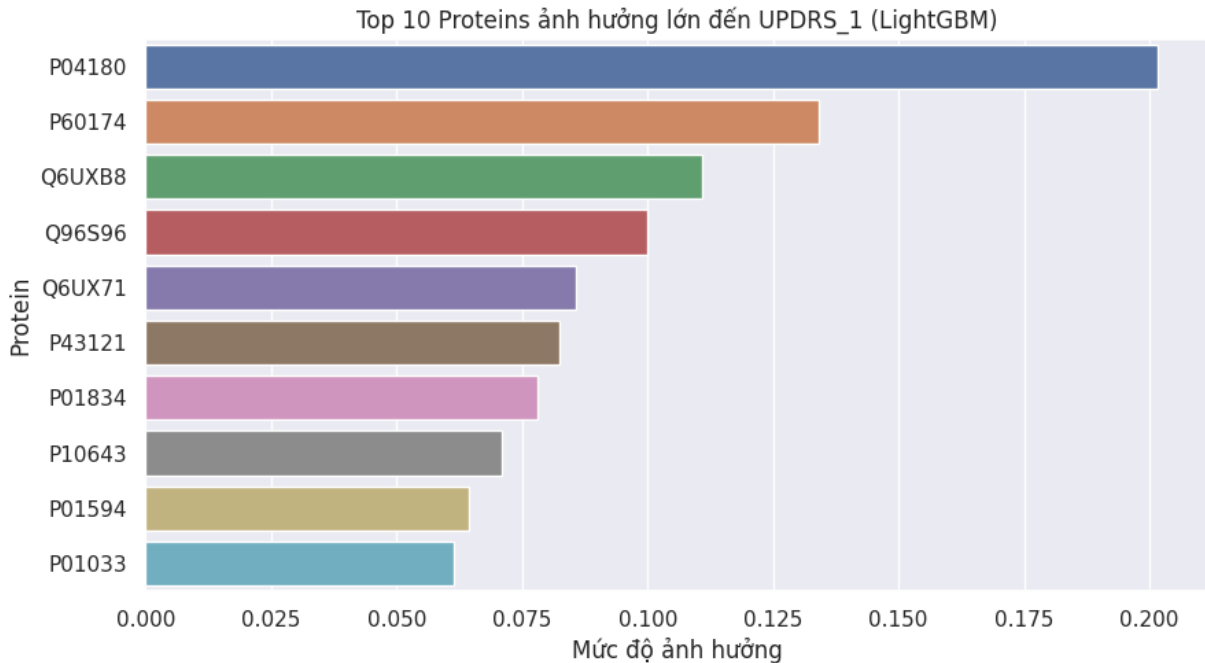


**Nhận xét:** Từ kết quả trực quan ta dễ dàng thấy được peptide **FIYGGC(UniMod\_4)GGNR** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_1** ở cả 2 mô hình.

Xem xét ảnh hưởng của các protein tới điểm **UPDRS\_1** từ 2 mô hình đã huấn luyện.



## ĐÁNH GIÁ



**Nhận xét:** Từ kết quả trực quan ta dễ dàng thấy được protein **P04180** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_1** ở cả 2 mô hình.

## UPDRS\_2

Kết quả SMAPE trên tập validation của 2 mô hình như sau:

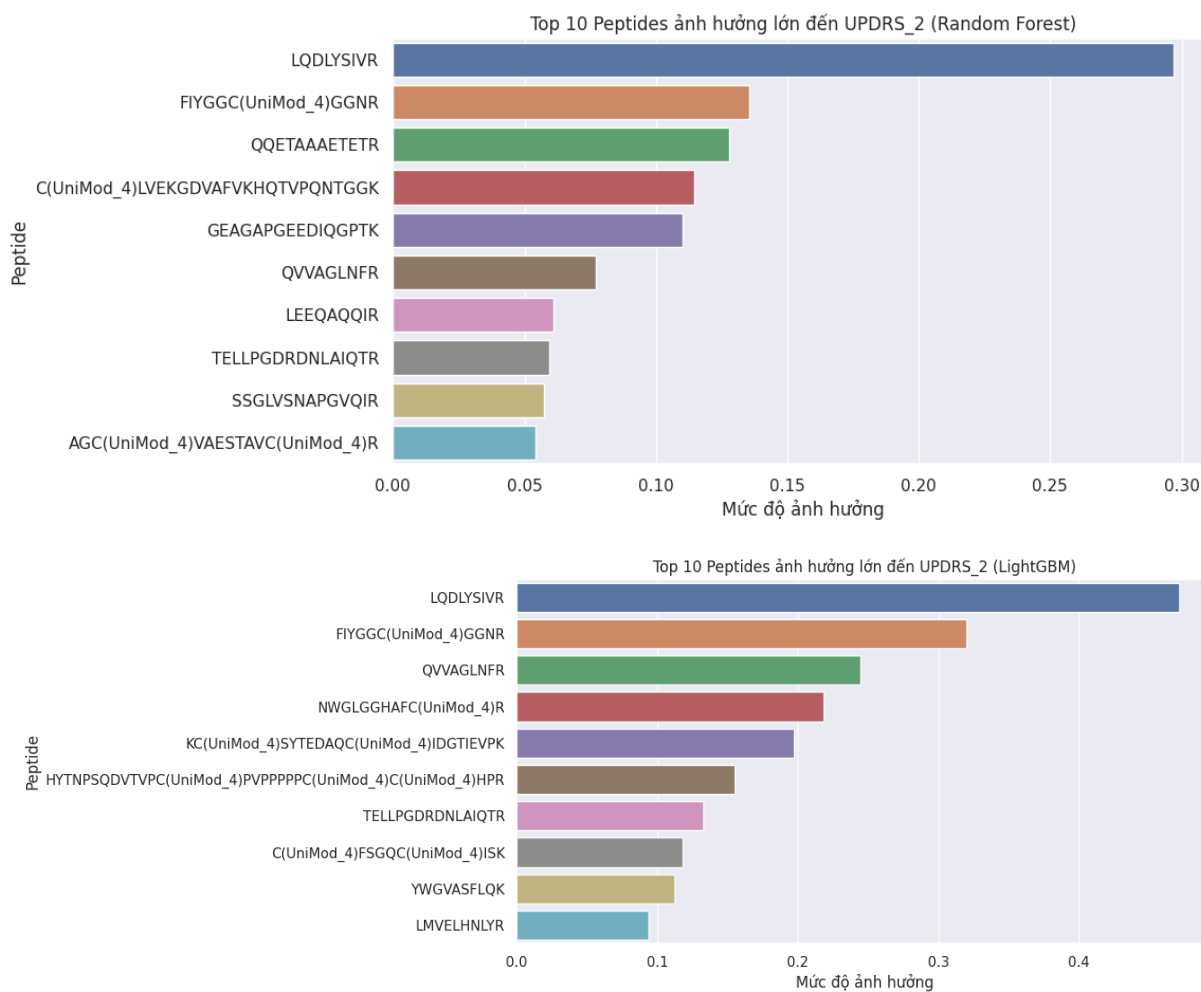
```
-----  
Random Forest model UPDRS_2:  
sMAPE test: 93.6242  
-----
```

```
LightGBM model UPDRS_2:  
sMAPE test: 89.9186  
-----
```

**Nhận xét:** Mô hình LightGBM Regressor cho kết quả SMAPE trên tập validation tốt hơn.

## ĐÁNH GIÁ

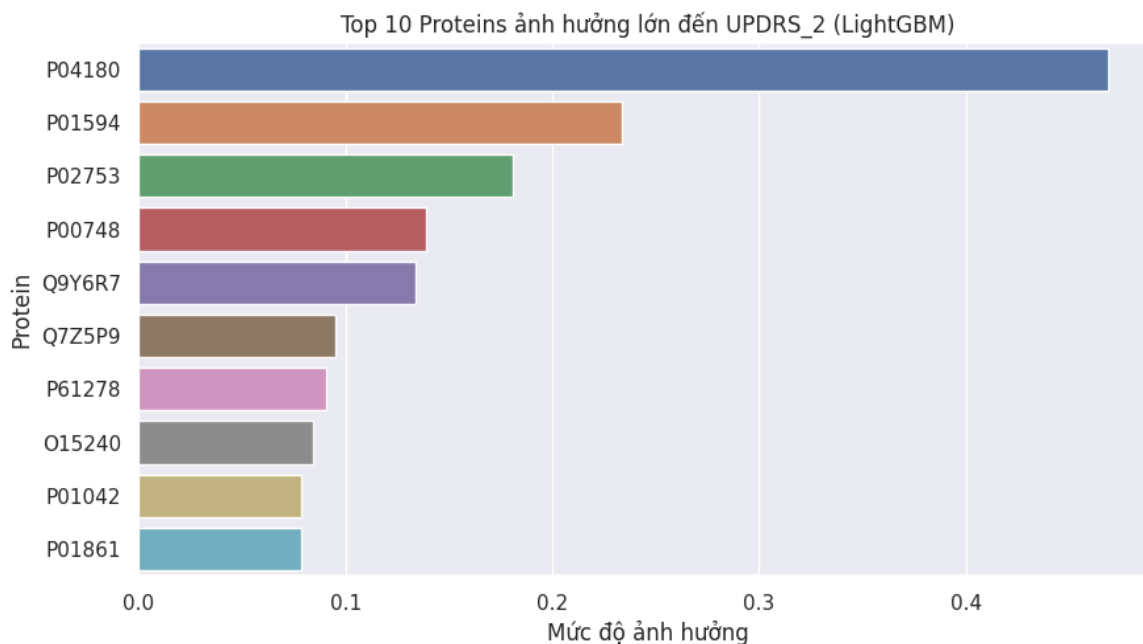
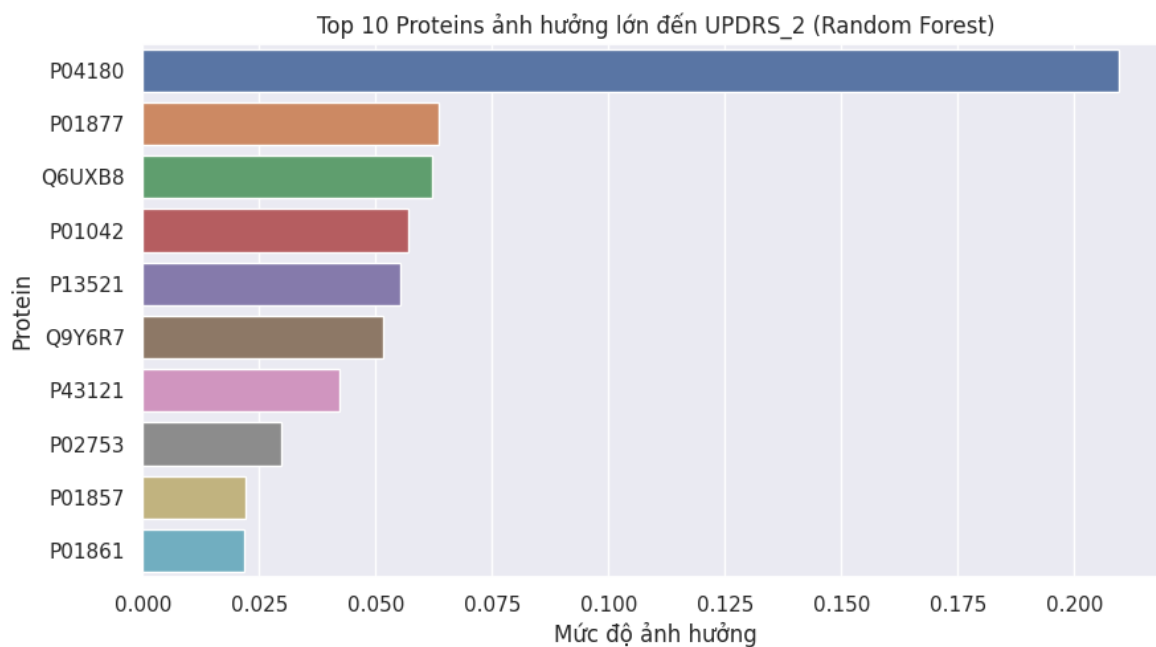
Xem xét ảnh hưởng của các peptide tới điểm **UPDRS\_2** từ 2 mô hình đã huấn luyện.



**Nhận xét:** Từ kết quả trực quan ta dễ dàng thấy được peptide **LQDLYSIVR** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_2** ở cả 2 mô hình.

Xem xét ảnh hưởng của các protein tới điểm **UPDRS\_2** từ 2 mô hình đã huấn luyện.

## ĐÁNH GIÁ



**Nhận xét:** Từ kết quả trực quan ta dễ dàng thấy được protein **P04180** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_2** ở cả 2 mô hình.

## ĐÁNH GIÁ

### UPDRS\_3

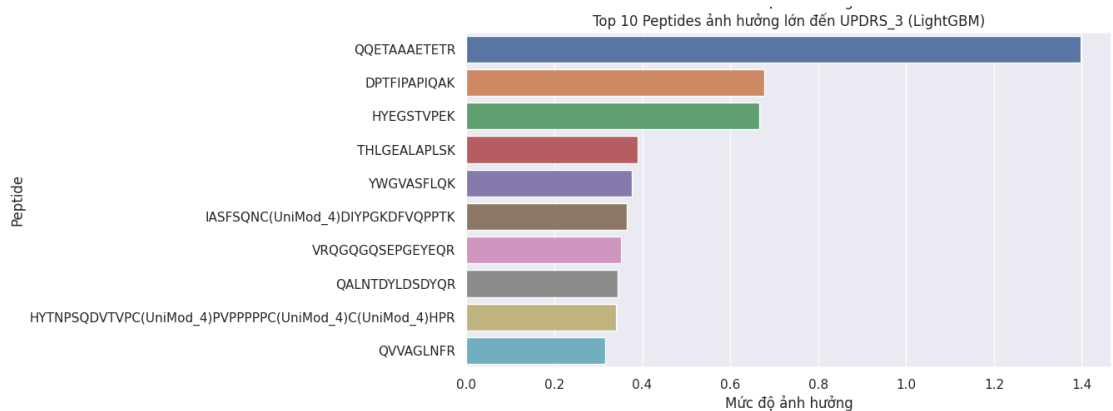
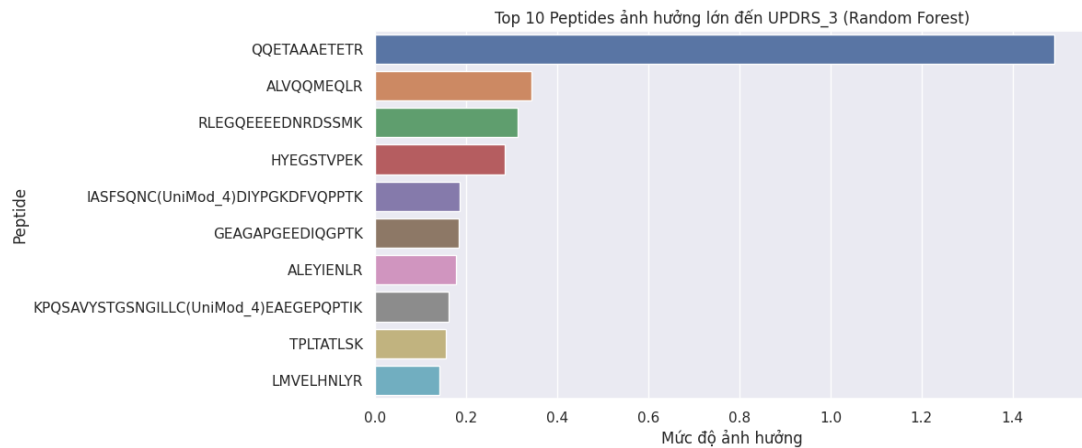
Kết quả SMAPE trên tập validation của 2 mô hình như sau:

```
-----  
Random Forest model UPDRS_3:  
sMAPE test: 90.7962  
-----
```

```
LightGBM model UPDRS_3:  
sMAPE test: 88.9188  
-----
```

**Nhận xét:** Mô hình LightGBM Regressor cho kết quả SMAPE trên tập validation tốt hơn.

Xem xét ảnh hưởng của các peptide tới điểm **UPDRS\_3** từ 2 mô hình đã huấn luyện.

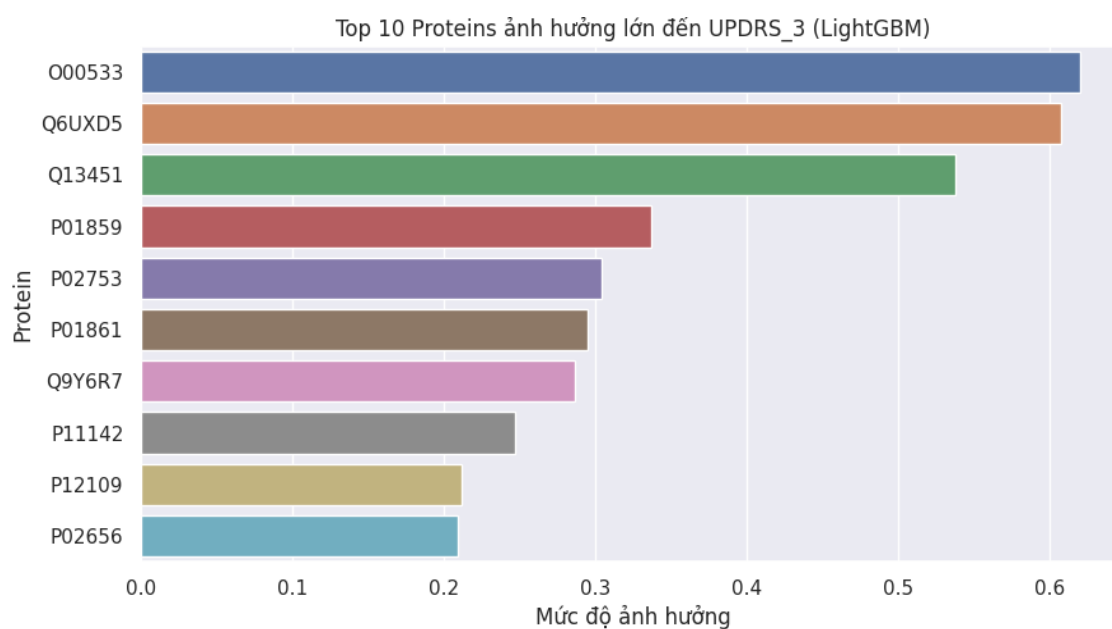
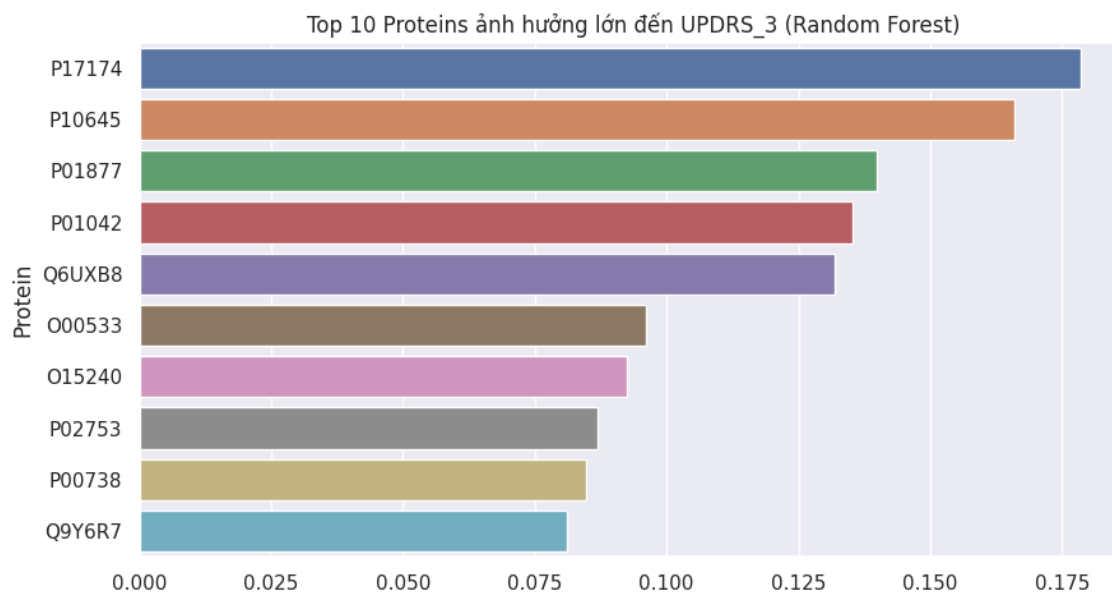




## ĐÁNH GIÁ

**Nhận xét:** Từ kết quả trực quan ta thấy được peptide **QQETAAAETETR** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_3** ở cả 2 mô hình.

Xem xét ảnh hưởng của các protein tới điểm **UPDRS\_3** từ 2 mô hình đã huấn luyện.



---

## ĐÁNH GIÁ

---

**Nhận xét:** Từ kết quả trực quan ta thấy sự ảnh hưởng của protein đến giá trị **UPDRS\_3** khác nhau ở 2 mô hình.

- Protein **P17174** có ảnh hưởng lớn nhất tới giá trị **UPDRS\_3** của mô hình Random Forest Regressor.
- Protein **O00533** có ảnh hưởng lớn nhất tới giá trị **UPDRS\_3** của mô hình LightGBM Regressor. **O00533** cũng có ảnh hưởng khá lớn tới **UPDRS\_3** của mô hình Random Forest Regressor.

### UPDRS\_4

Kết quả SMAPE trên tập validation của 2 mô hình như sau:

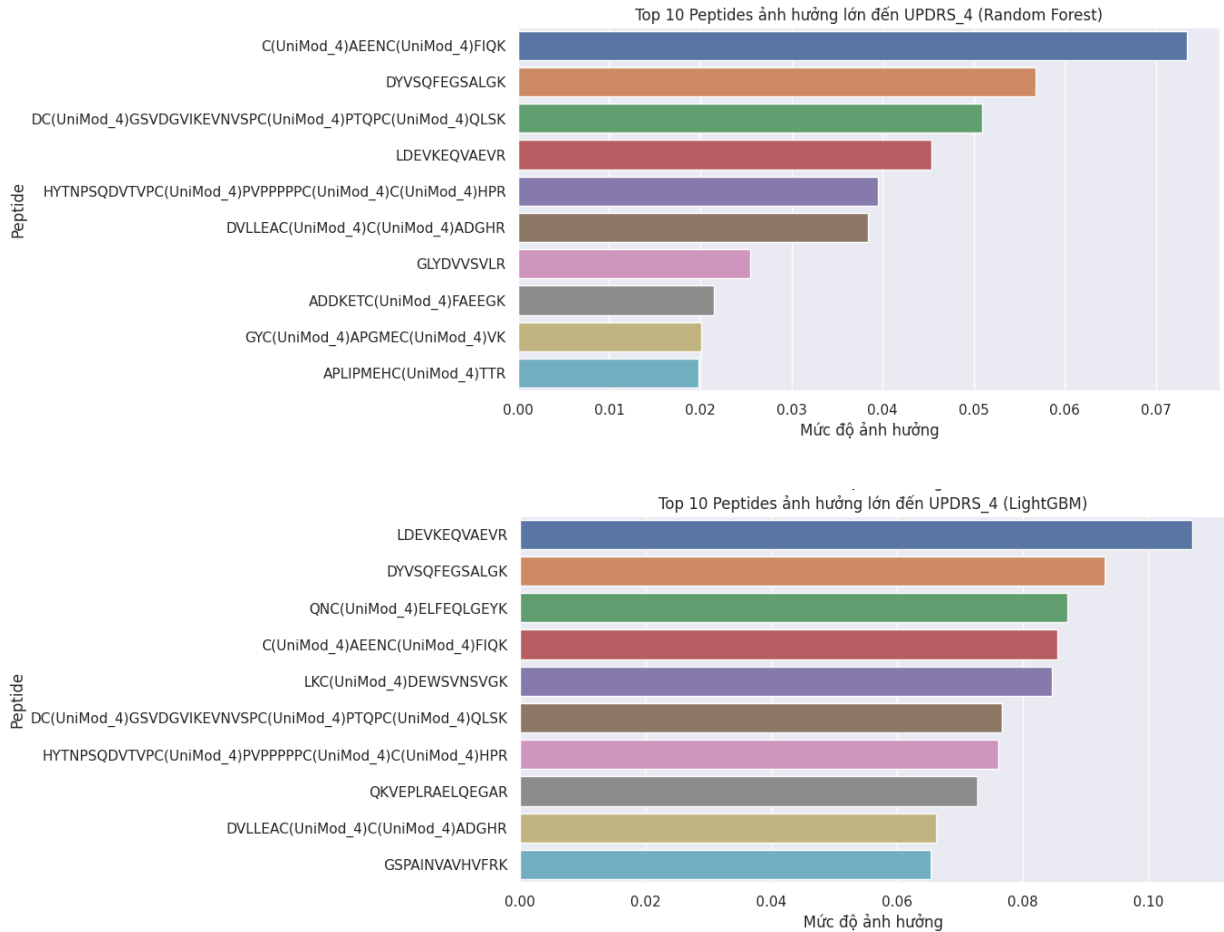
```
-----  
Random Forest model UPDRS_4:  
sMAPE test: 124.5383
```

```
-----  
LightGBM model UPDRS41:  
sMAPE test: 119.5429  
-----
```

**Nhận xét:** Mô hình LightGBM Regressor cho kết quả SMAPE trên tập validation tốt hơn.

Xem xét ảnh hưởng của các peptide tới điểm **UPDRS\_4** từ 2 mô hình đã huấn luyện.

## ĐÁNH GIÁ

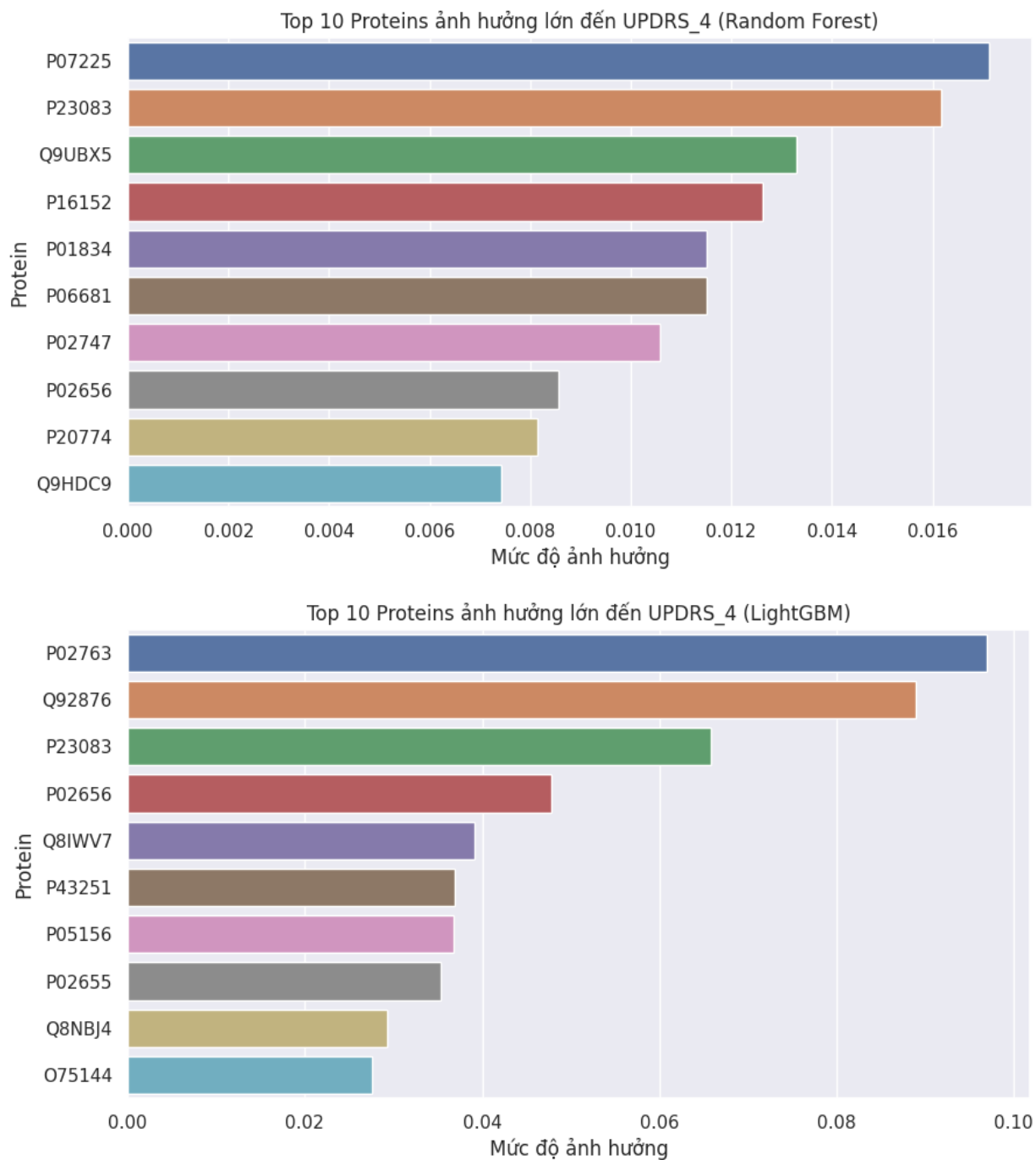


**Nhận xét:** Có sự khác biệt về sự ảnh hưởng của các peptide đến giá trị **UPDRS\_4** ở cả 2 mô hình.

- Peptide **C(UniMod\_4)AEENC(UniMod\_4)FIQK** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_4** của mô hình Random Forest Regressor. **C(UniMod\_4)AEENC(UniMod\_4)FIQK** cũng có ảnh hưởng lớn tới giá trị **UPDRS\_4** của mô hình LightGBM Regressor.
- Peptide **LDEVKEQVAEVR** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_4** của mô hình LightGBM Regressor. **LDEVKEQVAEVR** cũng có ảnh hưởng lớn tới giá trị **UPDRS\_4** của mô hình Random Forest Regressor.

## ĐÁNH GIÁ

Xem xét ảnh hưởng của các protein tới điểm **UPDRS\_4** từ 2 mô hình đã huấn luyện.



---

## ĐÁNH GIÁ

---

**Nhận xét:** Có sự khác biệt về sự ảnh hưởng của các protein đến giá trị **UPDRS\_4** ở cả 2 mô hình.

- Protein **P07225** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_4** của mô hình Random Forest Regressor.
- Protein **P02763** có ảnh hưởng lớn nhất đến giá trị **UPDRS\_4** của mô hình LightGBM Regressor.

---

## NHẬN XÉT CHUNG

---

### Về SMAPE của Random Forest Regressor và LightGBM Regressor

Trên cả 4 giá trị **UPDRS\_1**, **UPDRS\_2**, **UPDRS\_3** và **UPDRS\_4** LightGBM Regressor đều cho giá trị SMAPE tốt hơn trên tập validation.

Tuy vậy, Giá trị SMAPE của 2 mô hình chênh lệch nhau không lớn.

Nhóm sẽ lựa chọn LightGBM Regressor làm mô hình đánh giá cuối cùng.

### Về mức độ ảnh hưởng của các protein và peptide đến điểm UPDRS của 2 mô hình Random Forest Regressor và LightGBM Regressor

Dưới đây là một vài nhận xét chung về ảnh hưởng của protein và peptide mà nhóm đã tìm ra.

Ảnh hưởng của các peptide

- Peptide **FIYGGC(UniMod\_4)GGNR** ảnh hưởng đến điểm **UPDRS\_1** và **UPDRS\_2** của cả 2 mô hình.
- Peptide **LDEVKEQVAEVR** ảnh hưởng lớn đến điểm **UPDRS\_1** và **UPDRS\_4** của cả 2 mô hình.
- Peptide **HYTNPSQDVTVP(UniMod\_4)PVPPPPPC(UniMod\_4)C(UniMod\_4)HP R** ảnh hưởng lớn đến điểm **UPDRS\_2**, **UPDRS\_3** và **UPDRS\_4** của mô hình LightGBM Regressor.
- Peptide **GEAGAPGEEDIQGPTK** có ảnh lớn đến điểm **UPDRS\_2**, **UPDRS\_3** và **UPDRS\_4** của mô hình Random Forest Regressor.

---

## NHẬN XÉT CHUNG

---

### Ảnh hưởng của các protein

- Protein **P04180** ảnh hưởng lớn đến điểm **UPDRS\_1** và **UPDRS\_2** của cả 2 mô hình.
- Protein **Q6UXB8** ảnh hưởng lớn đến điểm **UPDRS\_1**, **UPDRS\_2** và **UPDRS\_3** của mô hình Random Forest Regressor.
- Protein **Q9Y6R7** ảnh hưởng lớn đến điểm **UPDRS\_2** và **UPDRS\_3** của cả 2 mô hình.
- Protein **P02753** ảnh hưởng lớn đến điểm **UPDRS\_2** và **UPDRS\_3** của cả 2 mô hình.
- Protein **P02656** ảnh hưởng lớn đến điểm **UPDRS\_3** và **UPDRS\_4** của mô hình LightGBM Regressor.




Độ quan trọng của các đặc trưng bên trên chỉ được dựa theo kết quả của việc huấn luyện 2 mô hình Random Forest Regressor và LightGBM Regressor. Những protein, peptide ảnh hưởng lớn là những protein, peptide nằm trong **TOP 10**.

Do dữ liệu cho điểm **UPDRS\_4** quá ít dẫn đến mô hình thiếu độ chính xác do đó sẽ không được sử dụng để dự đoán. Thay vào đó các giá trị **UPDRS\_4** khi dự đoán Hidden Test của Kaggle sẽ được gán giá trị **0** (do **UPDRS\_4** từ tập dữ liệu huấn luyện chủ yếu là các giá trị 0).

### Về lựa chọn đặc trưng để giảm số đặc trưng của mô hình

Nhóm đã tiến hành lựa chọn đặc trưng dựa trên tương quan của các đặc trưng với biến đích với sự hỗ trợ của hàm [SelectKBest](#) phương thức đánh giá là [f-regression](#).

Tuy nhiên kết quả không được như mong đợi. Kết quả tốt nhất sẽ được đề cập sau.

	<b>Parkinson's Disease Progression Prediction - Version 44</b> <small>Succeeded (after deadline) · 17h ago · Notebook Parkinson's Disease Progression Prediction   Version 44   Feature selection 120</small>	<b>78.095</b>	<b>69.437</b>	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 43</b> <small>Succeeded (after deadline) · 17h ago · Notebook Parkinson's Disease Progression Prediction   Version 43   Feature selection 100</small>	<b>75.453</b>	<b>66.558</b>	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 42</b> <small>Succeeded (after deadline) · 17h ago · Notebook Parkinson's Disease Progression Prediction   Version 42   Feature selection 40</small>	<b>77.456</b>	<b>69.513</b>	<input type="checkbox"/>

Bên cạnh đó nhóm cũng cho rằng giữ lại toàn bộ protein và peptide sẽ tránh được thiếu sót cho tập luật.

---

## NHẬN XÉT CHUNG








---

### Về việc điều chỉnh các tham số để cải thiện mô hình

Nhóm đã tiến hành khởi tạo không gian tìm kiếm cho các siêu tham số và sử dụng [RandomizedSearchCV](#) để tìm kiếm tham số tối ưu.

Tuy nhiên, thời gian thực thi quá lâu trong khi kết quả không được như mong đợi.

Có một số lần thử cho kết quả tốt hơn tuy nhiên khi sử dụng bộ tham số đó cho mô hình kết quả lại không giống như trước đó. Kết quả tốt nhất sẽ được đề cập sau.

	<b>Parkinson's Disease Progression Prediction - Version 39</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 39   rf params, lgbm no params	73.574	65.145	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 38</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 38   rf_params	72.792	63.656	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 37</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 37   rf params	73.032	63.403	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 36</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 36   rf params	72.849	63.508	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 35</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 35   lgbm params	72.424	62.561	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 34</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 34   lgbm params	72.203	61.957	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 33</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 33   rf params	73.119	64.243	<input type="checkbox"/>
	<b>Parkinson's Disease Progression Prediction - Version 32</b> Succeeded (after deadline) · 2d ago · Notebook Parkinson's Disease Progression Prediction   Version 32   lgbm params	72.515	63.09	<input type="checkbox"/>

Nguyên nhân có thể là do không gian tìm kiếm không phù hợp. Mà việc chạy kiểm tra lại tốn kém rất nhiều thời gian (khoảng 5 - 6 tiếng với Hidden Test của Kaggle) trong khi mô hình với các tham số mặc định cho kết quả tốt và thời gian thực thi nhanh.

Do đó nhóm đã quyết định sử dụng các tham số mặc định của mô hình để huấn luyện.



---

## KẾT QUẢ SMAPE TRÊN KAGGLE

---

Sử dụng API có sẵn của Kaggle trên trang cuộc thi để đánh giá mô hình LighGBM Regressor với Hidden Test của cuộc thi.

```
import sys
sys.path.append('/kaggle/input/amp-parkinsons-disease-progression-prediction')
```

```
import amp_pd_peptide_310
amp_pd_peptide_310.make_env.func_dict['__called__'] = False
env = amp_pd_peptide_310.make_env()
iter_test = env.iter_test()
```

Xây dựng hàm dự đoán để dự đoán điểm **UPDRS** và tạo tập kết quả dự đoán theo đúng định dạng mà cuộc thi yêu cầu. Điểm **UPDRS\_4** sẽ được gán giá trị 0.

```
def get_predictions(test_df, model):
    targets = ['updrs_1', 'updrs_2', 'updrs_3', 'updrs_4']
    test_ds = test_df[features].copy()
    test_ds = test_ds.fillna(0)

    for updrs in targets:
        test_df['result_' + str(updrs)] = 0
        if updrs != 'updrs_4':
            test_df['result_' + str(updrs)] = np.round(model[updrs].predict(test_ds))

    result = pd.DataFrame()
```

---

## KẾT QUẢ SMAPE TRÊN KAGGLE

---

```
for month in [0, 6, 12, 24]:
    for updrs in [1, 2, 3, 4]:
        temp = test_df[['visit_id', 'result_updrs_' + str(updrs)]].copy()
        temp['prediction_id'] = temp['visit_id'] + '_updrs_' + str(updrs) + '_plus_' + str(month) + '_months'
        temp['rating'] = temp['result_updrs_' + str(updrs)]
        temp = temp[['prediction_id', 'rating']]

        result = result.append(temp)

result = result.drop_duplicates(subset=['prediction_id', 'rating'])
result = result.reset_index().drop(columns=['index'])

return result
```

Xử lý dữ liệu Test tương tự như đã đề cập bên trên sau đó chạy thử để kiểm tra.

```
for (test, test_peptides, test_proteins, sample_submission) in iter_test:
    # Chuẩn bị dữ liệu
    test_proteins_peptides_df = prepare_train_data(test_proteins, test_peptides)

    std_features = list(test_proteins_peptides_df.drop(columns=['visit_id']).columns)
    test_proteins_peptides_std_df = test_proteins_peptides_df.copy()
    std_scaler = StandardScaler()
    test_proteins_peptides_std_df[std_features] = std_scaler.fit_transform(test_proteins_peptides_std_df[std_features])

    test_df = test.merge(test_proteins_peptides_std_df, on=['visit_id'], how='left')

    # Trong trường hợp có cột bị thiếu, điền tất cả các cột thiếu bằng giá trị 0
    for col in features:
        if col not in test_df.columns:
            test_df[col] = 0

    test_df = test_df[['visit_id'] + features]

    result = get_predictions(test_df, lightgbm_model)
    print(result)

env.predict(result)
```

## KẾT QUẢ SMAPE TRÊN KAGGLE

Kết quả chạy thử như sau:

This version of the API is not optimized and should not be used to estimate the runtime of your code on the hidden test set.

	prediction_id	rating
0	3342_0_updrs_1_plus_0_months	4.0
1	50423_0_updrs_1_plus_0_months	4.0
2	3342_0_updrs_2_plus_0_months	4.0
3	50423_0_updrs_2_plus_0_months	4.0
4	3342_0_updrs_3_plus_0_months	13.0
5	50423_0_updrs_3_plus_0_months	13.0
6	3342_0_updrs_4_plus_0_months	0.0
7	50423_0_updrs_4_plus_0_months	0.0
8	3342_0_updrs_1_plus_6_months	4.0
9	50423_0_updrs_1_plus_6_months	4.0
10	3342_0_updrs_2_plus_6_months	4.0
11	50423_0_updrs_2_plus_6_months	4.0
12	3342_0_updrs_3_plus_6_months	13.0
13	50423_0_updrs_3_plus_6_months	13.0
14	3342_0_updrs_4_plus_6_months	0.0
15	50423_0_updrs_4_plus_6_months	0.0
16	3342_0_updrs_1_plus_12_months	4.0
17	50423_0_updrs_1_plus_12_months	4.0
18	3342_0_updrs_2_plus_12_months	4.0
19	50423_0_updrs_2_plus_12_months	4.0

Sau khi chạy thử thành công Submit notebook để đánh giá bằng Hidden Test với SMAPE. Kết quả tốt nhất mà nhóm nhận được là **Private Score: 72.246** và **Public Score: 62.287** sau khoảng 1 giờ chạy.



Parkinson's Disease Progression Prediction - Version 45

Succeeded (after deadline) · 15h ago · Notebook Parkinson's Disease Progression Prediction | Version 45 | Add SHAP

72.246

62.287



---

## TÀI LIỆU THAM KHẢO

---

[1] Trang web của cuộc thi

<https://www.kaggle.com/competitions/amp-parkinsons-disease-progression-prediction>

[2] International Parkinson and Movement Disorder Society

[MDS-UPDRS The MDS-sponsored Revision of the Unified Parkinson's Disease Rating Scale](#)

[3] Slide bài giảng lý thuyết

<https://drive.google.com/drive/u/0/folders/1AJjAjoWg9yiT954iD-iPnSrfDvVWtM4N>

[4] Về việc xử lý dữ liệu

<https://www.kaggle.com/code/kimtaehun/simple-preprocessing-for-time-series-prediction>

[5] SMAPE và Feature Selection

<https://www.kaggle.com/code/gokifujiya/parkinson-s-disease-mds-updrs-features-selection>

[6] Điều chỉnh các siêu tham số

[https://scikit-learn.org/stable/modules/grid\\_search.html#randomized-parameter-search](https://scikit-learn.org/stable/modules/grid_search.html#randomized-parameter-search)