

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**

\*\*\*\*\*



**BÁO CÁO ĐỒ ÁN CUỐI KỲ**  
**MÔN HỌC: MÁY HỌC**

**GVHD :** PGS.TS. Lê Đình Duy  
ThS. Phạm Nguyễn Trường An

**LỚP :** CS114.P11

**Nhóm :** Trần Võ Lâm Trường - 22521586  
Ngô Nguyễn Nam Trung - 22521559

*Thành phố Hồ Chí Minh, tháng 01/2025*

# Mục lục

<b>MỤC LỤC</b>	<b>3</b>
<b>GITHUB REPOSITORY</b>	<b>5</b>
0.1 Link Github: . . . . .	5
0.2 Mô tả cấu trúc . . . . .	5
0.3 Cách sử dụng các code đã viết . . . . .	8
<b>I Đồ án dự đoán điểm từ dữ liệu wecode</b>	<b>9</b>
<b>1 GIỚI THIỆU BÀI TOÁN</b>	<b>10</b>
<b>2 CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN</b>	<b>11</b>
2.1 Bước 1: Thống kê dữ liệu. . . . .	11
2.2 Bước 2: Feature Engineering (Tạo Đặc Trưng) . . . . .	11
2.3 Bước 3: Xử lý dữ liệu . . . . .	12
2.4 Bước 4: Lựa chọn mô hình và tham số . . . . .	14
2.5 Bước 5: Huấn luyện và đánh giá mô hình . . . . .	15
2.6 Bước 6: Kiểm tra kết quả dự đoán thực tế . . . . .	16
<b>3 KẾT QUẢ NHẬN ĐƯỢC VÀ NHẬN XÉT</b>	<b>19</b>
<b>4 PHÂN TÍCH MỒ XẺ (ABLATION STUDY)</b>	<b>21</b>
<b>5 TÀI LIỆU THAM KHẢO</b>	<b>25</b>
<b>II Đồ án nhận diện hằng xe qua ảnh chụp</b>	<b>26</b>
<b>0 UPDATE SAU KHI VÂN ĐÁP</b>	<b>27</b>

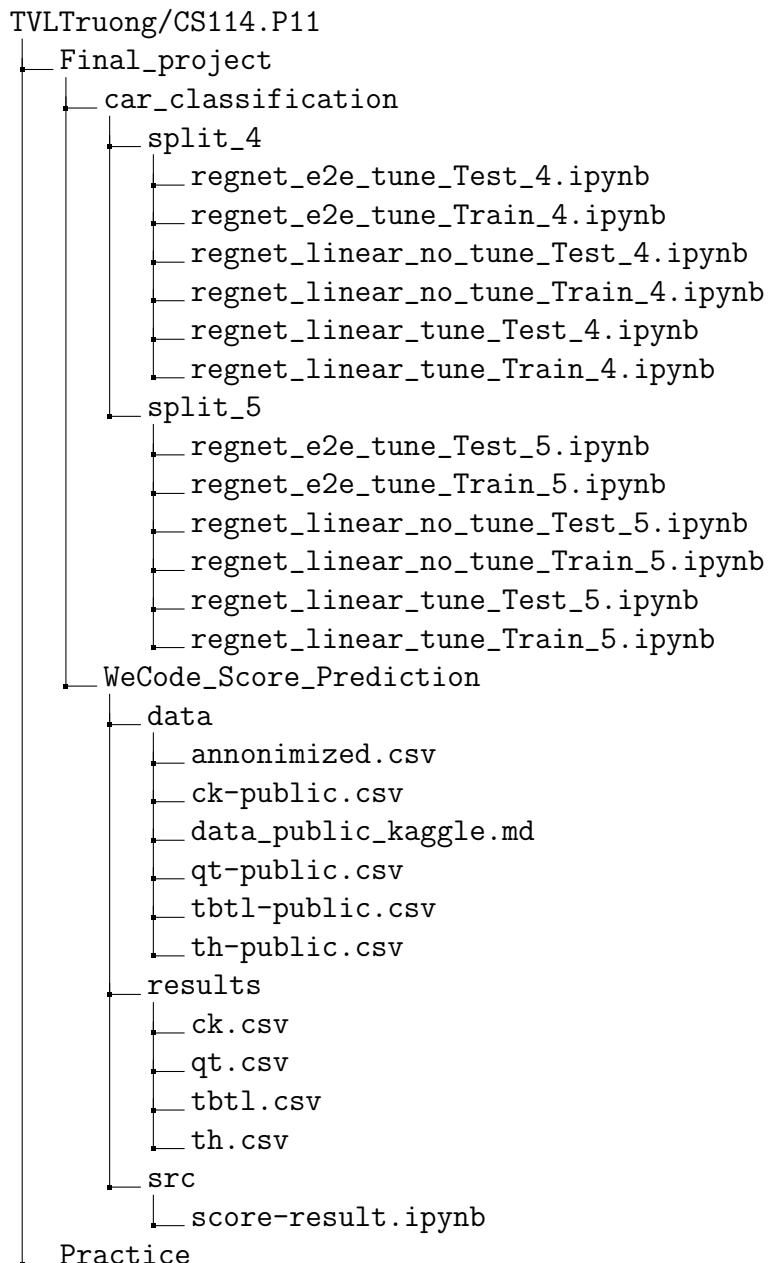
<b>1 GIỚI THIỆU BÀI TOÁN NHẬN DIỆN HÃNG XE</b>	<b>28</b>
<b>2 CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN</b>	<b>29</b>
Bước 1: Thực hiện các bài tập được giao . . . . .	29
Bước 2: Khảo sát dữ liệu . . . . .	34
Bước 3: Xử lý data . . . . .	41
Bước 4: Lựa chọn mô hình học sâu để trích xuất đặc trưng và huấn luyện . . . . .	43
Bước 5: Huấn luyện và đánh giá mô hình . . . . .	44
<b>3 KẾT QUẢ NHẬN ĐƯỢC VÀ NHẬN XÉT</b>	<b>47</b>
<b>4 ABLATION STUDY</b>	<b>48</b>
<b>5 TÀI LIỆU THAM KHẢO</b>	<b>50</b>

# GITHUB REPOSITORY

0.1 Link Github: <https://github.com/TVLTruong/CS114.P11.git>

## 0.2 Mô tả cấu trúc

```
TVLTruong/CS114.P11
├── README.md
└── Final_project
    ├── car_classification
    │   ├── Dataset.md
    │   ├── Model.md
    │   ├── processing
    │   │   ├── checkClustering.md
    │   │   ├── clustering.ipynb
    │   │   ├── survey_statistics_data.ipynb
    │   │   ├── tool-createsplit-car.ipynb
    │   │   ├── tool-datasetstat-car.ipynb
    │   │   └── tool-datasetviz.ipynb
    │   ├── split_1
    │   │   ├── regnet_e2e_tune_Test_1.ipynb
    │   │   ├── regnet_e2e_tune_Train_1.ipynb
    │   │   ├── regnet_linear_no_tune_Test_1.ipynb
    │   │   ├── regnet_linear_no_tune_Train_1.ipynb
    │   │   ├── regnet_linear_tune_Test_1.ipynb
    │   │   └── regnet_linear_tune_Train_1.ipynb
    │   ├── split_2
    │   │   ├── regnet_e2e_tune_Test_2.ipynb
    │   │   ├── regnet_e2e_tune_Train_2.ipynb
    │   │   ├── regnet_linear_no_tune_Test_2.ipynb
    │   │   ├── regnet_linear_no_tune_Train_2.ipynb
    │   │   ├── regnet_linear_tune_Test_2.ipynb
    │   │   └── regnet_linear_tune_Train_2.ipynb
    │   ├── split_3
    │   │   ├── regnet_e2e_tune_Test_3.ipynb
    │   │   ├── regnet_e2e_tune_Train_3.ipynb
    │   │   ├── regnet_linear_no_tune_Test_3.ipynb
    │   │   ├── regnet_linear_no_tune_Train_3.ipynb
    │   │   ├── regnet_linear_tune_Test_3.ipynb
    │   │   └── regnet_linear_tune_Train_3.ipynb
```



Trong thư mục Final\_project có 2 thư mục tương ứng với 2 project.

Trong folder car\_classification là folder đồ án nhận diện hằng xe qua ảnh chụp, bao gồm 5 thư mục khác:

- Dataset.md chứa đường dẫn đến link dataset kaggle được public và mô tả thông tin dataset. Tóm tắt mô tả: dataset kaggle bao gồm thư mục ảnh được tải từ dataset đóng góp chung, các file csv chứa thông tin ảnh trùng từ bài tập clustering, các file csv từ bài tập split, các file csv từ bài tập datasetstat-car, file csv ảnh lỗi không đọc được, và file thông tin các ảnh tên không hợp lệ.

- Model.md chứa đường dẫn đến các model nhóm đã train thuận lợi cho việc test.
- Thư mục Processing chứa các file xử lý data, thực hiện các bài tập được giao và các thống kê, khảo sát dữ liệu:
  - + clustering.ipynb thực hiện bài tập tìm kiếm ảnh trùng.
  - + checkClustering.md chứa link dẫn đến notebook đã được public trên kaggle thực hiện hiển thị các ảnh trùng từ kết quả file clustering.ipynb
  - + survey\_statistics\_data.ipynb thực hiện khảo sát, thống kê thông tin trong dataset (bước 2 chương 2).
  - + 3 file còn lại tool-createsplit-car.ipynb, tool-datasetstat-car.ipynb và tool-datasetviz.ipynb lần lượt là code và các output hiển thị của các bài tập còn lại trong 4 bài tập được giao.
- Các thư mục Split-1 đến split-5 chứa các code thực hiện việc xử lý data đến train và test với dataset tương ứng với split. Các file code trong thư mục được đặt tên mang đặc trưng của nội dung. Ví dụ: trong split-1 có regnet\_linear\_tune\_Train\_1.ipynb là file train mô hình RegNet\_Y\_128GF với Linear Head được finetune trên tập file CarDataset-Splits-1-Train.csv; regnet\_linear\_tune\_Test\_1.ipynb là file test mô hình RegNet\_Y\_128GF với Linear Head được finetune đã được train trên file CarDataset-Splits-1-Test.csv. Tương tự với các file khác.

Trong folder WeCode\_Score\_Prediction là folder đồ án dự đoán điểm từ dữ liệu wecode, bao gồm 3 thư mục khác:

- data là thư mục chứa dữ liệu, đặc biệt có file data\_public\_kaggle.md chứa đường dẫn đến dataset được public trên kaggle .
- results là các file kết quả nhóm đã thực thi và submit trên wecode.
- src chứa file code.

### 0.3 Cách sử dụng các code đã viết

Nhóm đã thực hiện đồ án trên kaggle nên phần này nhóm sẽ mô tả cách sử dụng trên kaggle.

**Bước 1:** clone github hoặc download file code về máy.

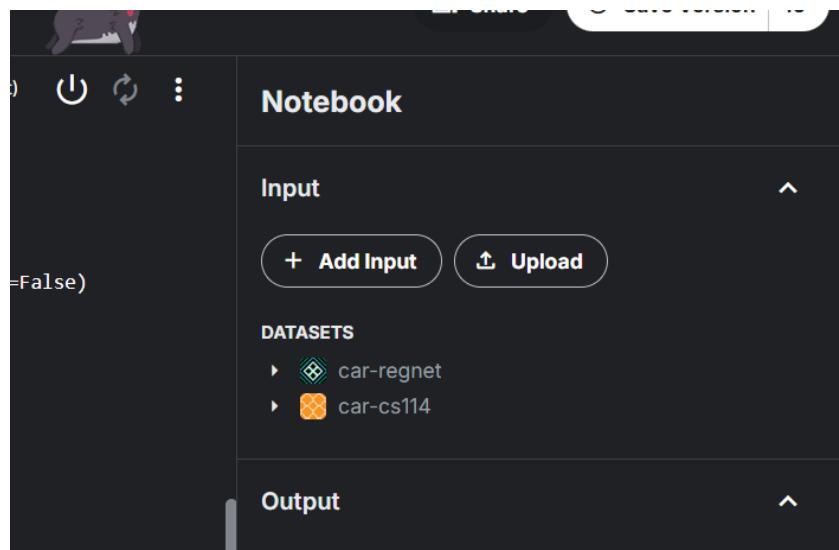
**Bước 2:** Mở kaggle.com

**Bước 3:** login(nếu đã có tài khoản)/signup(nếu chưa có tài khoản) **Bước 4:** vào setting -> Phone verification(nếu chưa xác thực)

**Bước 5:** Ra giao diện kaggle.com -> chọn create dưới logo kaggle -> Chọn New Notebook

**Bước 6:** Trong giao diện Notebook, chọn File -> import notebook -> tải/kéo thả file code ở bước 1.

**Bước 7:** kiểm tra xem có đầy đủ đường dẫn như trong code chưa, nếu chưa tiến hành "add input" các file cần thiết, ví dụ như trong ảnh dưới:



**Bước 8:** trong giao diện notebook vào setting -> accelerator -> chọn gpu nếu chưa bật -> start session.

**Bước 9:** Chạy all hoặc từng cell theo thứ tự từ trên xuống.

**Lưu ý:** trong quá trình chạy code gặp lỗi không tìm thấy đường dẫn. Lỗi đó có thể do chưa add input. Nếu đã có input tương ứng có thể do chỉ có 1 input. Khi có 1 input (chỉ 1 trong 2 như ảnh bước 7) thì add thêm một input bất kỳ sẽ chạy được.

**Bước 10:** chờ kết quả.

**Bước 11:** tải file nếu kết quả là file để xem vì khi kết thúc session file sẽ biến mất.

## **Phần I: Đồ án dự đoán điểm từ dữ liệu wecode**

# Chương 1: GIỚI THIỆU BÀI TOÁN

Bài toán dự đoán điểm số của sinh viên qua các bài tập trên trang WeCode dựa trên dữ liệu từ các tệp dữ liệu được cung cấp. Đây là bài toán thuộc nhóm *Regression*, với mục tiêu là xây dựng mô hình dự đoán chính xác điểm số dựa trên các đặc trưng đầu vào trích xuất từ dữ liệu nộp bài trên trang WeCode.

## Chương 2: CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN

### 2.1 Bước 1: Thống kê dữ liệu.

Nhóm thực hiện tổng hợp và thống kê dữ liệu từ các tệp anonymized.csv, ck-public.csv, th-public.csv, qt-public.csv, và tbtl-public.csv.

Số lượng bản ghi và cột trong từng tệp:

```
RangeIndex: 295198 entries, 0 to 295197
Data columns (total 11 columns):
 #   Column          Non-Null Count
 ---  -- 
 0   concat('it001','assignment_id')    2
 1   concat('it001','problem_id')       2
 2   concat('it001', 'username')        2
 3   is_final                         2
 4   status                           2
 5   pre_score                        2
 6   coefficient                     2
 7   concat('it001','language_id')     2
 8   created_at                       2
 9   updated_at                      2
 10  judgement                        2
```

(a) Tệp anonymized.csv

```
Index: 755 entries, 0 to 760
Data columns (total 2 columns):
 #   Column      Non-Null Count
 ---  -- 
 0   username    755 non-null
 1   CK          755 non-null
```

(b) Tệp ck-public.csv

```
Index: 755 entries, 0 to 760
Data columns (total 2 columns):
 #   Column      Non-Null Count
 ---  -- 
 0   username    755 non-null
 1   TH          755 non-null
```

(c) Tệp th-public.csv

```
Index: 755 entries, 0 to 760
Data columns (total 2 columns):
 #   Column      Non-Null Count
 ---  -- 
 0   username    755 non-null
 1   QT          755 non-null
```

(d) Tệp qt-public.csv

```
Data columns (total 2 columns):
 #   Column      Non-Null Count
 ---  -- 
 0   username    799 non-null
 1   TBTL        799 non-null
```

(e) Tệp tbtl-public.csv

### 2.2 Bước 2: Feature Engineering (Tạo Đặc Trưng)

Loại bỏ các bản ghi có `is_final = 0` hoặc `pre_score = 10000` (không hợp lệ).

Đổi tên các cột phức tạp để dễ sử dụng hơn, ví dụ:

`concat('it001', assignment_id) → assignment_id.`

`concat('it001', problem_id) → problem_id.`

`concat('it001', username) → username.`

Loại bỏ cột không cần thiết như `concat('it001', language_id)` và `updated_at`.

## 2.3 Bước 3: Xử lý dữ liệu

Nhóm đã thực hiện việc tạo ra các đặc trưng quan trọng từ dữ liệu WeCode, bao gồm:

- Tần suất nộp bài (`frequency_vector`): Biểu diễn tần suất nộp bài trong từng khung giờ của ngày.

	username	hour_0	hour_1	hour_2	hour_3	hour_4	hour_5	hour_6	hour_7	hour_8	...	hour_14	hour_15	hour_16	hour_17	hour_18	hour_19	hour_20	hour_21	hour_22
0	ed9eab6a707f50154024b24d7efcb874a9795dd	6.0	6.0	1.0	1.0	6.0	8.0	5.0	11.0	17.0	...	8.0	3.0	13.0	1.0	1.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0467e479c03507c71a805d291	1.0	3.0	7.0	10.0	25.0	5.0	16.0	28.0	36.0	...	25.0	30.0	29.0	26.0	9.0	2.0	0.0	0.0	0.0
2	0bd2037bf68a97753e5e67ab55dac026a649f279	0.0	11.0	1.0	0.0	0.0	0.0	0.0	6.0	12.0	...	17.0	7.0	52.0	25.0	8.0	1.0	0.0	0.0	0.0
3	b7298b0fe50443a623af9b56792b330c2d052845	0.0	24.0	28.0	39.0	22.0	2.0	4.0	7.0	...	37.0	25.0	16.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	c60be70309789b39355dc612f36e37090ccad5dc	0.0	4.0	20.0	13.0	9.0	4.0	5.0	10.0	11.0	...	14.0	8.0	5.0	4.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1484	b722e6209f2858fa0bf80947cadcbde586bb666	0.0	1.0	3.0	4.0	3.0	2.0	5.0	21.0	12.0	...	12.0	17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1485	02c88d535d7393d30ce3338174d0a17ee7e8b8cc	0.0	9.0	22.0	2.0	16.0	8.0	0.0	3.0	0.0	...	15.0	13.0	10.0	6.0	0.0	0.0	0.0	1.0	4.0
1486	b45e8d507778dab56f381a681c453cbfd3b4050	0.0	8.0	17.0	7.0	4.0	4.0	10.0	13.0	11.0	...	1.0	12.0	8.0	1.0	6.0	3.0	0.0	3.0	0.0
1487	ea385e57f5d3d6841a69977d7af680e135928bca	1.0	14.0	5.0	13.0	0.0	0.0	21.0	6.0	1.0	...	38.0	23.0	28.0	31.0	1.0	0.0	0.0	0.0	0.0
1488	232cce96362898f08e9150ba244adaf2d6583ab2	0.0	0.0	10.0	7.0	2.0	8.0	4.0	6.0	20.0	...	21.0	25.0	24.0	8.0	0.0	11.0	0.0	0.0	0.0

1489 rows × 25 columns

- Số lần nộp bài (`count_assignment_vector`): Tổng hợp số lần nộp bài của học viên cho từng bài tập.

	username	count_assignment_vector0	count_assignment_vector1	count_assignment_vector2	count_assignment_vector3	count_assignment_vector4	count_assignment_vector5	co
0	ed9eab6a707f50154024b24d7efcb874a9795dd	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0467e479c03507c71a805d291	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0bd2037bf68a97753e5e67ab55dac026a649f279	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	b7298b0fe50443a623af9b56792b330c2d052845	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	c60be70309789b39355dc612f36e37090ccad5dc	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
1484	b722e6209f2858fa0bf80947cadcbde586bb666	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1485	02c88d535d7393d30ce3338174d0a17ee7e8b8cc	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1486	b45e8d507778dab56f381a681c453cbfd3b4050	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1487	ea385e57f5d3d6841a69977d7af680e135928bca	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1488	232cce96362898f08e9150ba244adaf2d6583ab2	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1489 rows × 204 columns

- Số lượng trạng thái bài tập (`status_counts_vector`): Tính tần suất các trạng thái của bài tập để giúp mô hình hiểu thêm về quá trình làm bài.

	username	status_counts_vector0	status_counts_vector1	status_counts_vector2	status_counts_vector3	status_counts_vector4	status_counts_vector5	status_counts_vector6	status
0	ed9eae6a707f50154024b24d7efcb874a9795dd	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0467e479c03507c71a805d291	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0bd2037bf68a97753e5e67ab55dac026a649f279	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	b7298b0fe50443a623af9b56792b330c2d052845	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	c60be70309789b39355dc612f36e37090ccad5dc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
1484	b722e6209f2858fa0bf80947cadcbde586bb666	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1485	02c88d535d7393d30ce3338174d0a17ee7e8b8cc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1486	b45e8d507778dab56f381a681c453cb4d3b4050	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1487	ea385e57f5d3d6841a6997dd7af680e135928bca	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1488	232cce96362898f08e9150ba244adaf2d6583ab2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1489 rows × 204 columns

- Số lượng vấn đề đã giải quyết (**problem\_counts\_vector**): Số vấn đề đã giải quyết trong mỗi bài tập.

	username	problem_counts_vector0	problem_counts_vector1	problem_counts_vector2	problem_counts_vector3	problem_counts_vector4	problem_counts_vector5	problem_counts_vector6	problem_counts_vector7	problem_counts_vector8	...
0	ed9eae6a707f50154024b24d7efcb874a9795dd	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	ba12c0a2cb367af0467e479c03507c71a805d291	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
2	0bd2037bf68a97753e5e67ab55dac026a649f279	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
3	b7298b0fe50443a623af9b56792b330c2d052845	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
4	c60be70309789b39355dc612f36e37090ccad5dc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1484	b722e6209f2858fa0bf80947cadcbde586bb666	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1485	02c88d535d7393d30ce3338174d0a17ee7e8b8cc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1486	b45e8d507778dab56f381a681c453cb4d3b4050	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1487	ea385e57f5d3d6841a6997dd7af680e135928bca	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1488	232cce96362898f08e9150ba244adaf2d6583ab2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

1489 rows × 204 columns

- Vấn đề và trạng thái bài tập (**problem\_vector**): Biểu diễn sự xuất hiện của các vấn đề trong bài tập.

	username	problem_vector0	problem_vector1	problem_vector2	problem_vector3	problem_vector4	problem_vector5	problem_vector6	problem_vector7	problem_vector8	...
0	ed9eae6a707f50154024b24d7efcb874a9795dd	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	ba12c0a2cb367af0467e479c03507c71a805d291	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...
2	0bd2037bf68a97753e5e67ab55dac026a649f279	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
3	b7298b0fe50443a623af9b56792b330c2d052845	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...
4	c60be70309789b39355dc612f36e37090ccad5dc	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1484	b722e6209f2858fa0bf80947cadcbde586bb666	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1485	02c88d535d7393d30ce3338174d0a17ee7e8b8cc	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1486	b45e8d507778dab56f381a681c453cb4d3b4050	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1487	ea385e57f5d3d6841a6997dd7af680e135928bca	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1488	232cce96362898f08e9150ba244adaf2d6583ab2	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...

1489 rows × 469 columns

- Thời gian giải quyết vấn đề (**time\_problem\_vector**): Thời gian giữa lần nộp bài đầu tiên và cuối cùng cho mỗi vấn đề.

	username	time_problem_vector0	time_problem_vector1	time_problem_vector2	time_problem_vector3	time_problem_vector4	time_problem_vector5	time_problem_vector6	time
0	ed9eae6a707f50154024b24d7efcb874a9795dd	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0467e479c03507c71a805d291	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0bd2037bf68a97753e5e67ab55dac026a649f279	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	b7298b0fe50443a623af9b56792b330c2d052845	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	c60be70309789b3935dc612f36e37090ccad5dc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
1484	b722e6209f2858faf0bf80947cadcbde586bb666	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1485	02c88d535d7393d30ce3338174d0a17ee7e8bb8cc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1486	b45e8d507778dab56f381a681c453cbfd3b4050	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1487	ea385e57f5d3d6841a6997dd7af680e135928bca	0.0	0.0	975.0	0.0	0.0	0.0	0.0	0.0
1488	232cce96362898f08e9150ba244adaf2d6583ab2	43.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1489 rows × 469 columns

- Số lần không hoàn thành vấn đề (`count_problem_0_vector`): Số lần học viên không hoàn thành một vấn đề.

	username	count_problem_0_vector0	count_problem_0_vector1	count_problem_0_vector2	count_problem_0_vector3	count_problem_0_vector4	count_problem_0_vector5	count_p
0	ed9eae6a707f50154024b24d7efcb874a9795dd	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0467e479c03507c71a805d291	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0bd2037bf68a97753e5e67ab55dac026a649f279	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	b7298b0fe50443a623af9b56792b330c2d052845	0.0	0.0	2.0	0.0	0.0	0.0	0.0
4	c60be70309789b3935dc612f36e37090ccad5dc	0.0	0.0	3.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
1484	b722e6209f2858faf0bf80947cadcbde586bb666	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1485	02c88d535d7393d30ce3338174d0a17ee7e8bb8cc	2.0	0.0	0.0	0.0	0.0	0.0	0.0
1486	b45e8d507778dab56f381a681c453cbfd3b4050	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1487	ea385e57f5d3d6841a6997dd7af680e135928bca	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1488	232cce96362898f08e9150ba244adaf2d6583ab2	2.0	0.0	0.0	0.0	0.0	0.0	0.0

1489 rows × 469 columns

- Điểm trung bình trước khi làm bài (`mean_prescore_problem_vector`): Điểm số trung bình của học viên trước khi làm bài.

	username	mean_prescore_problem_vector0	mean_prescore_problem_vector1	mean_prescore_problem_vector2	mean_prescore_problem_vector3	mean_prescore_problem_vector4
0	ed9eae6a707f50154024b24d7efcb874a9795dd	9.210340	0.0	0.000000	0.0	
1	ba12c0a2cb367af0467e479c03507c71a805d291	9.210340	0.0	9.210340	0.0	
2	0bd2037bf68a97753e5e67ab55dac026a649f279	9.210340	0.0	0.000000	0.0	
3	b7298b0fe50443a623af9b56792b330c2d052845	9.210340	0.0	8.111728	0.0	
4	c60be70309789b3935dc612f36e37090ccad5dc	9.210340	0.0	8.817298	0.0	
...	...	...	...	...	...	
1484	b722e6209f2858faf0bf80947cadcbde586bb666	0.000000	0.0	0.000000	0.0	
1485	02c88d535d7393d30ce3338174d0a17ee7e8bb8cc	8.111728	0.0	9.210340	0.0	
1486	b45e8d507778dab56f381a681c453cbfd3b4050	0.000000	0.0	0.000000	0.0	
1487	ea385e57f5d3d6841a6997dd7af680e135928bca	9.210340	0.0	9.210340	0.0	
1488	232cce96362898f08e9150ba244adaf2d6583ab2	8.216989	0.0	9.210340	0.0	

1489 rows × 469 columns

## 2.4 Bước 4: Lựa chọn mô hình và tham số

Nhóm sử dụng hai mô hình Gradient Boosting là LightGBM và CatBoost cho bài toán dự đoán điểm. Các tham số của 2 mô hình:

```

def objective_lgb(trial, target):
    params = {
        "objective": "regression",
        "metric": "rmse",
        "n_estimators": 1000,
        "verbosity": -1,
        "bagging_freq": 1,
        "learning_rate": trial.suggest_float("learning_rate", 1e-3, 0.1, log=True),
        "num_leaves": trial.suggest_int("num_leaves", 2, 2**10),
        "subsample": trial.suggest_float("subsample", 0.05, 1.0),
        "colsample_bytree": trial.suggest_float("colsample_bytree", 0.05, 1.0),
        "min_data_in_leaf": trial.suggest_int("min_data_in_leaf", 1, 100)
    }

```

(a) Các tham số của mô hình LightGBM

```

def objective_cat(trial, target):
    # Cập nhật tham số cho model
    params = {
        'learning_rate': trial.suggest_loguniform('learning_rate', 0.01, 0.1),
        'depth': trial.suggest_int('depth', 3, 8),
        'l2_leaf_reg': trial.suggest_loguniform('l2_leaf_reg', 1e-3, 10),
        'iterations': trial.suggest_int('iterations', 100, 1000),
        'eval_metric': 'RMSE',
        'random_seed': 42,
        'verbose': False,
        'loss_function': 'RMSE'
    }

```

(b) Các tham số của mô hình CatBoost

## 2.5 Bước 5: Huấn luyện và đánh giá mô hình

### Huấn Luyện Voting Regressor

Nhóm sử dụng *Voting Regressor* để kết hợp các mô hình *LightGBM* và *CatBoost*. Quy trình huấn luyện bao gồm:

- **Huấn luyện Voting Regressor:** Sau khi tối ưu tham số, mô hình *Voting Regressor* được huấn luyện và kết hợp các mô hình.
- **Đánh giá mô hình:** Sử dụng  $R^2$  để đánh giá hiệu quả và độ ổn định của mô hình. Kết quả đánh giá từng phần được minh họa dưới đây:

```

255: learn: 0.2766353      total: 8.21s      remaining: 64.1ms
256: learn: 0.2738284      total: 8.24s      remaining: 32.1ms
257: learn: 0.2730476      total: 8.27s      remaining: 0us
Voting Regressor R^2 for QT: 0.3583 ± 0.0975

```

(a) Dánh giá hiệu quả mô hình cho QT (Quá trình)

```

255: learn: 0.3745239      total: 7.89s      remaining: 61.7ms
256: learn: 0.3728235      total: 7.92s      remaining: 30.8ms
257: learn: 0.3713767      total: 7.95s      remaining: 0us
Voting Regressor R^2 for TH: 0.4957 ± 0.0564

```

(b) Dánh giá hiệu quả mô hình cho TH (Thực hành)

```

255: learn: 0.3823110      total: 8.08s      remaining: 63.1ms
256: learn: 0.3809083      total: 8.11s      remaining: 31.6ms
257: learn: 0.3796765      total: 8.14s      remaining: 0us
Voting Regressor R^2 for CK: 0.3850 ± 0.0277

```

(c) Dánh giá hiệu quả mô hình cho CK (Cuối kỳ)

```

255: learn: 0.1696203      total: 8.59s      remaining: 67.1ms
256: learn: 0.1684673      total: 8.62s      remaining: 33.5ms
257: learn: 0.1677978      total: 8.65s      remaining: 0us
Voting Regressor R^2 for TBTL: 0.2825 ± 0.0675

```

(d) Dánh giá hiệu quả mô hình cho TBTL (Trung bình tích lũy)

## 2.6 Bước 6: Kiểm tra kết quả dự đoán thực tế

**Chuẩn bị dữ liệu:**

- Với mỗi mục tiêu (*QT*, *TH*, *CK*, *TBTL*), chọn các cột chung giữa dữ liệu huấn luyện (`train_term`) và dữ liệu kiểm tra (`test_term`).
- Chỉ giữ lại những cột chung này từ cả hai bộ dữ liệu để đảm bảo tính nhất quán.
- So sánh dữ liệu giữa `train_term` và `test_term` để tìm các hàng không khớp nhau. Sau đó, chỉ giữ lại các hàng duy nhất (không trùng lặp) trong `different_rows`, đảm bảo mỗi học viên chỉ xuất hiện một lần.

Processing target: QT								
Number of rows in different_rows: 735								
	username	assignment_id_encoded	count_assignment_vector0	count_assignment_vector1	count_assignment_vector2	count_assignment_vector3	count_assignment_vector4	count
0	ed9eaebe6a707f50154024b24d7efcb874a9795dd	116	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0a467e479c03507c71a805d291	156	0.0	0.0	0.0	0.0	0.0	0.0
2	b7298b0fe50443a623a9b56792b330c2d052845	195	0.0	0.0	0.0	0.0	0.0	0.0
3	c60be70309789b39355dc612f6e37090ccad5dc	167	0.0	0.0	0.0	0.0	0.0	0.0
4	a22a58c5be8aa2c2700619e37f2b7a64ef7e6b	178	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
730	508e0627871ed482db8ab34544e43e5d856a889c	168	0.0	0.0	0.0	0.0	0.0	0.0
731	d525d6ed4b0c6362ddaa4814c3930a6a62bd99	167	0.0	0.0	0.0	0.0	0.0	0.0
732	b722e62092858faf0b80947cadcbde586bb666	12	0.0	0.0	0.0	0.0	0.0	0.0
733	b45e8d507778da56f381a681c453cbfd3d3b4050	90	0.0	0.0	0.0	0.0	0.0	0.0
734	ea385e57f5d3d6841a6997dd7af680e135928bca	38	0.0	0.0	0.0	0.0	0.0	0.0

735 rows × 612 columns

(a) Chuẩn bị dữ liệu cho QT

Processing target: TH								
Number of rows in different_rows: 736								
	username	assignment_id_encoded	count_assignment_vector0	count_assignment_vector1	count_assignment_vector2	count_assignment_vector3	count_assignment_vector4	count
0	ed9eaebe6a707f50154024b24d7efcb874a9795dd	116	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0a467e479c03507c71a805d291	156	0.0	0.0	0.0	0.0	0.0	0.0
2	b7298b0fe50443a623a9b56792b330c2d052845	195	0.0	0.0	0.0	0.0	0.0	0.0
3	c60be70309789b39355dc612f6e37090ccad5dc	167	0.0	0.0	0.0	0.0	0.0	0.0
4	a22a58c5be8aa2c2700619e37f2b7a64ef7e6b	178	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
731	508e0627871ed482db8ab34544e43e5d856a889c	168	0.0	0.0	0.0	0.0	0.0	0.0
732	d525d6ed4b0c6362ddaa4814c3930a6a62bd99	167	0.0	0.0	0.0	0.0	0.0	0.0
733	b722e62092858faf0b80947cadcbde586bb666	12	0.0	0.0	0.0	0.0	0.0	0.0
734	b45e8d507778da56f381a681c453cbfd3d3b4050	90	0.0	0.0	0.0	0.0	0.0	0.0
735	ea385e57f5d3d6841a6997dd7af680e135928bca	38	0.0	0.0	0.0	0.0	0.0	0.0

736 rows × 612 columns

(b) Chuẩn bị dữ liệu cho TH

Processing target: CK								
Number of rows in different_rows: 734								
	username	assignment_id_encoded	count_assignment_vector0	count_assignment_vector1	count_assignment_vector2	count_assignment_vector3	count_assignment_vector4	count
0	ed9eaebe6a707f50154024b24d7efcb874a9795dd	116	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0a467e479c03507c71a805d291	156	0.0	0.0	0.0	0.0	0.0	0.0
2	b7298b0fe50443a623a9b56792b330c2d052845	195	0.0	0.0	0.0	0.0	0.0	0.0
3	c60be70309789b39355dc612f6e37090ccad5dc	167	0.0	0.0	0.0	0.0	0.0	0.0
4	a22a58c5be8aa2c2700619e37f2b7a64ef7e6b	178	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
729	508e0627871ed482db8ab34544e43e5d856a889c	168	0.0	0.0	0.0	0.0	0.0	0.0
730	d525d6ed4b0c6362ddaa4814c3930a6a62bd99	167	0.0	0.0	0.0	0.0	0.0	0.0
731	b722e62092858faf0b80947cadcbde586bb666	12	0.0	0.0	0.0	0.0	0.0	0.0
732	b45e8d507778da56f381a681c453cbfd3d3b4050	90	0.0	0.0	0.0	0.0	0.0	0.0
733	ea385e57f5d3d6841a6997dd7af680e135928bca	38	0.0	0.0	0.0	0.0	0.0	0.0

734 rows × 612 columns

(c) Chuẩn bị dữ liệu cho CK

Processing target: TBTL								
Number of rows in different_rows: 690								
	username	assignment_id_encoded	count_assignment_vector0	count_assignment_vector1	count_assignment_vector2	count_assignment_vector3	count_assignment_vector4	count
0	ed9eaebe6a707f50154024b24d7efcb874a9795dd	116	0.0	0.0	0.0	0.0	0.0	0.0
1	ba12c0a2cb367af0a467e479c03507c71a805d291	156	0.0	0.0	0.0	0.0	0.0	0.0
2	b7298b0fe50443a623a9b56792b330c2d052845	195	0.0	0.0	0.0	0.0	0.0	0.0
3	c60be70309789b39355dc612f6e37090ccad5dc	167	0.0	0.0	0.0	0.0	0.0	0.0
4	a22a58c5be8aa2c2700619e37f2b7a64ef7e6b	178	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
685	f0d74f49e6479eae4f25634ff80dd51e60d60c	116	0.0	0.0	0.0	0.0	0.0	0.0
686	d525d6ed4b0c6362ddaa4814c3930a6a62bd99	167	0.0	0.0	0.0	0.0	0.0	0.0
687	b722e62092858faf0b80947cadcbde586bb666	12	0.0	0.0	0.0	0.0	0.0	0.0
688	b45e8d507778da56f381a681c453cbfd4d3b4050	90	0.0	0.0	0.0	0.0	0.0	0.0
689	ea385e57f5d3d6841a6997dd7af680e135928bca	38	0.0	0.0	0.0	0.0	0.0	0.0

690 rows × 612 columns

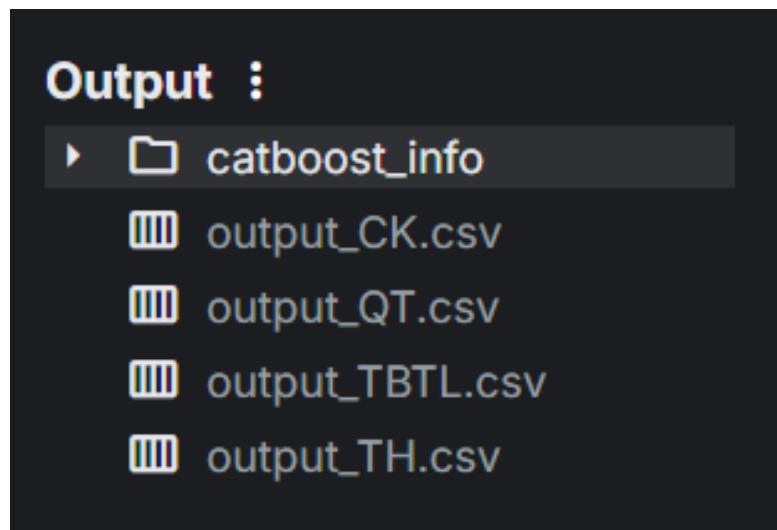
(d) Chuẩn bị dữ liệu cho TBTL

## Dự đoán:

- Chuyển dữ liệu  $X_{test}$  thành mảng NumPy ( $X_{pca}$ ) để chuẩn bị cho mô hình.
- Sử dụng mô hình *Voting Regressor* và *LightGBM* để dự đoán kết quả ( $y_{pre}$ ,  $y_{pre1}$ ).

## Lưu kết quả:

- Lấy tên người dùng từ cột `username` trong `different_rows`.
- Kết hợp với các giá trị dự đoán (`y_pre`) để tạo ra các cặp (`username`, `label`).
- Lưu kết quả này vào một tệp CSV với tên `output_<target>.csv`, nơi `<target>` là mục tiêu cụ thể.



Hình 5: Kết quả dự đoán và lưu vào tệp CSV

## Chương 3: KẾT QUẢ NHẬN ĐƯỢC VÀ NHẬN XÉT

### Kết quả đạt được

Nhóm đã áp dụng mô hình Voting Regressor để dự đoán điểm số trên từng loại bài tập. Kết quả đạt được được đánh giá thông qua hệ số xác định  $R^2$  (coefficient of determination), cho thấy mức độ mô hình có thể giải thích được sự biến động trong dữ liệu đầu vào. Kết quả cụ thể như sau:

**QT (Quá trình):**  $R^2 = 0.2168 \pm 0.0387$

→ Mô hình có khả năng giải thích 21.68% sự biến thiên của dữ liệu QT, với độ lệch chuẩn là 3.87%.

**TH (Thực hành):**  $R^2 = 0.2440 \pm 0.0304$

→ Mô hình đạt hiệu quả cao nhất đối với dữ liệu TH, giải thích được 24.40% sự biến thiên, với độ lệch chuẩn thấp (3.04%).

**CK (Cuối kỳ):**  $R^2 = 0.1782 \pm 0.0286$

→ Mô hình đạt hiệu quả thấp hơn so với QT và TH, giải thích được 17.82% sự biến thiên, với độ lệch chuẩn là 2.86%.

**TBTL (Trung bình tích lũy):**  $R^2 = 0.1204 \pm 0.0277$

→ Hiệu quả thấp nhất trong các mục tiêu, chỉ giải thích được 12.04% sự biến thiên, với độ lệch chuẩn là 2.77%.

### Đánh giá trên mô hình

#### Hiệu quả mô hình:

Kết quả cho thấy mô hình Voting Regressor hoạt động tốt nhất với dữ liệu TH (Thực hành), đạt  $R^2$  cao nhất (24.40%). Điều này có thể giải thích bởi sự đồng nhất và mối quan hệ rõ ràng hơn giữa các đặc trưng và điểm số thực hành, so với các loại điểm khác.

#### Hạn chế:

Kết quả thấp hơn đối với TBTL ( $R^2 = 0.1204$ ) và CK ( $R^2 = 0.1782$ ) cho thấy rằng các đặc

trưng hiện tại chưa đủ mạnh để giải thích sự biến động trong dữ liệu của các mục tiêu này. Điều này có thể do các yếu tố bên ngoài như khả năng tự học, động lực cá nhân hoặc tính ngẫu nhiên trong điểm cuối kỳ mà dữ liệu hiện tại chưa phản ánh được.

## Kết quả thực tế

### Kết quả thực tế:

Username	Name	Class	Total	Total accepted	Dự đoán điểm TBTL 1 năm sau khi làm bài tập NMLT trên wecode	Dự đoán điểm thực hành IT001 từ dữ liệu làm bài trên wecode	Dự đoán điểm quá trình IT001 từ dữ liệu làm bài trên wecode	Dự đoán điểm cuối kỳ IT001 từ dữ liệu làm bài trên wecode
22521586	Trần Võ Lâm Trường	CS114.P11	111	0	19	36	30	26

22521586	Trần Võ Lâm Trường	CS114.P11	111	0	19	36	30	26
				1s	0:4	3 - 574h 51s	1 - 573h 1m 43s	13 - 573h 3m 11s

22521559	Ngô Nguyễn Nam Trung	CS114.P11	96	0	19	36	30	11
				1s	0:4	12 - 541h 32m 49s	5 - 561h 1m 25s	3 - 562h 47m 31s

Mặc dù mô hình đã cho kết quả dự đoán trên các mục tiêu, nhưng điểm số thực tế vẫn còn khá thấp, cho thấy mô hình chưa đạt được độ chính xác mong muốn (TBTL: 19/100, TH: 36/100, QT: 30/100, CK: 26/100).

Nguyên nhân là do một số yếu tố như chất lượng dữ liệu chưa đủ tốt, các đặc trưng chưa được lựa chọn tối ưu, hoặc sự thiếu chính xác trong quá trình huấn luyện và tinh chỉnh mô hình. Việc sử dụng các mô hình như LightGBM và CatBoost vẫn cần cải tiến thêm về cấu hình tham số và cách kết hợp mô hình để đạt được kết quả chính xác hơn.

# Chương 4: PHÂN TÍCH MÔ XÉ (ABLATION STUDY)

## Bước 1: Thống Kê Dữ Liệu

Mục đích của Thống Kê Dữ Liệu là giúp phát hiện các giá trị thiếu, giá trị bất thường và các vấn đề trong dữ liệu để làm sạch và chuẩn bị dữ liệu chất lượng cho mô hình. Đồng thời, nó cũng giúp nhận diện các mối quan hệ và tương tác giữa các đặc trưng trong dữ liệu, từ đó hỗ trợ nhóm quyết định lựa chọn và xây dựng các đặc trưng phù hợp cho mô hình.

## Bước 2: Xử Lý Dữ Liệu

### Lọc dữ liệu không phù hợp:

- Loại bỏ các bản ghi có điều kiện:
  - + `is_final` bằng 0, tức là các bài nộp chưa hoàn thành.
  - + `pre_score` bằng 10000, biểu thị các bài nộp không hợp lệ.
- Mục tiêu là đảm bảo rằng chỉ các bài nộp hợp lệ và có ý nghĩa trong quá trình phân tích được giữ lại.

### Đổi tên cột: để dễ sử dụng hơn

- `concat('it001',assignment_id) → assignment_id.`
- `concat('it001',problem_id) → problem_id.`
- `concat('it001',username) → username.`

### Loại bỏ cột không cần thiết:

- `concat('it001',language_id).`
- `updated_at.`

## Bước 3: Feature Engineering (Tạo Đặc Trưng)

Trong quá trình tạo ra các đặc trưng từ dữ liệu, nhóm đã phân tích từng đặc trưng để đánh giá tác động đến mô hình:

- **Tần suất nộp bài (frequency\_vector):** Giúp mô hình hiểu thói quen học tập của học viên trong các khung giờ khác nhau.
- **Số lần nộp bài (count\_assignment\_vector):** Cung cấp thông tin về sự tương tác của học viên với các bài tập, thể hiện sự tích cực và mức độ hoàn thành công việc.
- **Số lượng trạng thái bài tập (status\_counts\_vector):** Tính toán tần suất các trạng thái bài tập giúp nhận diện xu hướng học tập của học viên.
- **Số lượng vấn đề đã giải quyết (problem\_counts\_vector):** Đo lường số lượng vấn đề đã giải quyết để đánh giá sự tiến bộ của học viên.
- **Vấn đề và trạng thái bài tập (problem\_vector):** Biểu diễn sự xuất hiện hoặc không xuất hiện của các vấn đề trong bài tập.
- **Thời gian giải quyết vấn đề (time\_problem\_vector):** Phản ánh thời gian học viên dành để giải quyết vấn đề trong bài tập.
- **Số lần học viên không hoàn thành vấn đề (count\_problem\_0\_vector):** Đánh giá mức độ không hoàn thành bài tập của học viên.
- **Điểm trung bình trước khi làm bài (mean\_prescore\_problem\_vector):** Giúp mô hình hiểu được sự cải thiện trong quá trình làm bài của học viên.

## Bước 4: Lựa Chọn Mô Hình và Tham Số

Các tham số được tối ưu hóa qua *Optuna* để cải thiện  $R^2$  cho *LightGBM* và giảm *RMSE* cho *CatBoost*.

### LightGBM

- **Learning rate:** Điều chỉnh tốc độ học để cân bằng giữa việc hội tụ nhanh và ổn định.

- **Num\_leaves:** Xác định độ phức tạp của cây, ảnh hưởng đến khả năng mô hình hóa dữ liệu, tránh overfitting hoặc underfitting.
- **Subsample và Colsample\_bytree:** Giảm overfitting và tăng khả năng tổng quát bằng cách chọn ngẫu nhiên các mẫu và đặc trưng.
- **Min\_data\_in\_leaf:** Kiểm soát độ phức tạp của cây, tránh overfitting khi số mẫu quá ít trong mỗi lá.

```
Best parameters for QT: {'learning_rate': 0.007193004196443358, 'num_leaves': 674, 'subsample': 0.8245983020617356, 'colsample_bytree': 0.2657318116001094, 'min_data_in_leaf': 15}
Best R2 score for QT: 0.3673109948992342
```

```
Best parameters for TH: {'learning_rate': 0.00875297529408041, 'num_leaves': 426, 'subsample': 0.4960391536371005, 'colsample_bytree': 0.8321293399456828, 'min_data_in_leaf': 15}
Best R2 score for TH: 0.49023683600581924
```

```
Best parameters for CK: {'learning_rate': 0.009543133805403274, 'num_leaves': 582, 'subsample': 0.5946229447992624, 'colsample_bytree': 0.8501876012592692, 'min_data_in_leaf': 20}
Best R2 score for CK: 0.41083340470444385
```

```
Best parameters for TBTL: {'learning_rate': 0.04598098385059911, 'num_leaves': 896, 'subsample': 0.8536529871371469, 'colsample_bytree': 0.22692399535797303, 'min_data_in_leaf': 17}
Best R2 score for TBTL: 0.27794176523325514
```

## CatBoost

- **Learning rate:** Điều chỉnh tốc độ cập nhật mô hình để cân bằng giữa hiệu quả và độ ổn định.
- **Depth:** Điều chỉnh độ sâu của cây để kiểm soát độ phức tạp, tránh overfitting.
- **L2\_leaf\_reg:** Điều chỉnh regularization L2 để giảm overfitting.
- **Iterations:** Điều chỉnh số vòng lặp huấn luyện để mô hình không bị underfitting hoặc overfitting.

```
Best parameters for QT: {'learning_rate': 0.07798603666523456, 'depth': 6, 'l2_leaf_reg': 0.00  
9999771916308091, 'iterations': 244}  
Best R2 score for QT: 0.3602626178110568
```

```
Best parameters for TH: {'learning_rate': 0.06505026898820336, 'depth': 5, 'l2_leaf_reg': 0.23  
685903382013157, 'iterations': 648}  
Best R2 score for TH: 0.48838398226953805
```

```
Best parameters for CK: {'learning_rate': 0.03523120480551389, 'depth': 7, 'l2_leaf_reg': 0.72  
38533452276372, 'iterations': 658}  
Best R2 score for CK: 0.379769598028054
```

```
Best parameters for TBTL: {'learning_rate': 0.06809335558402813, 'depth': 7, 'l2_leaf_reg': 0.  
006911740123969738, 'iterations': 258}  
Best R2 score for TBTL: 0.2582158479813418
```

## Bước 5: Huấn luyện và đánh giá

Voting Regressor được chọn vì nó kết hợp các dự đoán từ các mô hình cơ sở như LightGBM và CatBoost, giúp tận dụng điểm mạnh của từng mô hình riêng lẻ. Điều này không chỉ cải thiện độ chính xác mà còn làm giảm rủi ro overfitting, tạo ra một dự đoán tốt hơn.

Mục tiêu là kiểm tra và đánh giá kết quả dự đoán từ mô hình đối với các mục (QT, TH, CK, TBTL) so với dữ liệu thực tế. Kết quả sẽ giúp nhóm đánh giá khả năng của mô hình và cung cấp thông tin cải thiện.

## Chương 5: TÀI LIỆU THAM KHẢO

1. LightGBM - Light Gradient Boosting Machine: <https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/>

Tài liệu tham khảo này được sử dụng trong quá trình tìm hiểu các tham số của mô hình LGBM trong chương 2, Bước 5: Huấn luyện và đánh giá mô hình.

2. CatBoost - Categorical Boosting:

<https://www.geeksforgeeks.org/catboost-parameters-and-hyperparameters/>

Tài liệu tham khảo này được sử dụng trong quá trình tìm hiểu các tham số và cách sử dụng mô hình CatBoost trong quá trình tối ưu hóa mô hình ở Chương 2, Bước 5: Huấn luyện và đánh giá mô hình.

3. Xây dựng đặc trưng: [https://github.com/truong04/CS114.021-22520125\\_22521575\\_22521260\\_22521204/tree/main/%C4%91%E1%BB%93%20%C3%A1n%20cs114/task2\\_score\\_it001/Feature2](https://github.com/truong04/CS114.021-22520125_22521575_22521260_22521204/tree/main/%C4%91%E1%BB%93%20%C3%A1n%20cs114/task2_score_it001/Feature2)

Tài liệu tham khảo này được sử dụng như nguồn tham khảo cách xây dựng các đặc trưng mở rộng ở Chương 2 Bước 3: Tạo đặc trưng.

## **Phần II: Đồ án nhận diện hằng xe qua ảnh chụp**

## Chương 0: UPDATE SAU KHI VÂN ĐÁP

- Kiểm tra lại các ảnh trùng sau khi thực hiện bài tập clustering
- Chia lại dataset theo kfold chỉ xét các ảnh có tên theo cấu trúc:  
mssv1-mssv2-mssv3.{ten\_thu\_muc}.{stt}.{jpg/png/jpeg}
- Thống kê chi tiết hơn về dữ liệu có visualize.
- Train lại split 1 với RegNet\_Y\_128GF với Linear Head finetune và không finetune.
- Tiếp tục thử nghiệm với RegNet\_Y\_128GF với Linear Head không finetune trên split 2 đến 5.
- Sử dụng pretrained-model RegNet\_Y\_128GF với Linear Head được finetune để trích xuất đặc trưng và train trên tập train split 2 đến 5.
- Thử nghiệm pretrained-model RegNet\_Y\_128GF với E2E được finetune để trích xuất đặc trưng và train trên split 1 đến 5.

## **Chương 1: GIỚI THIỆU BÀI TOÁN**

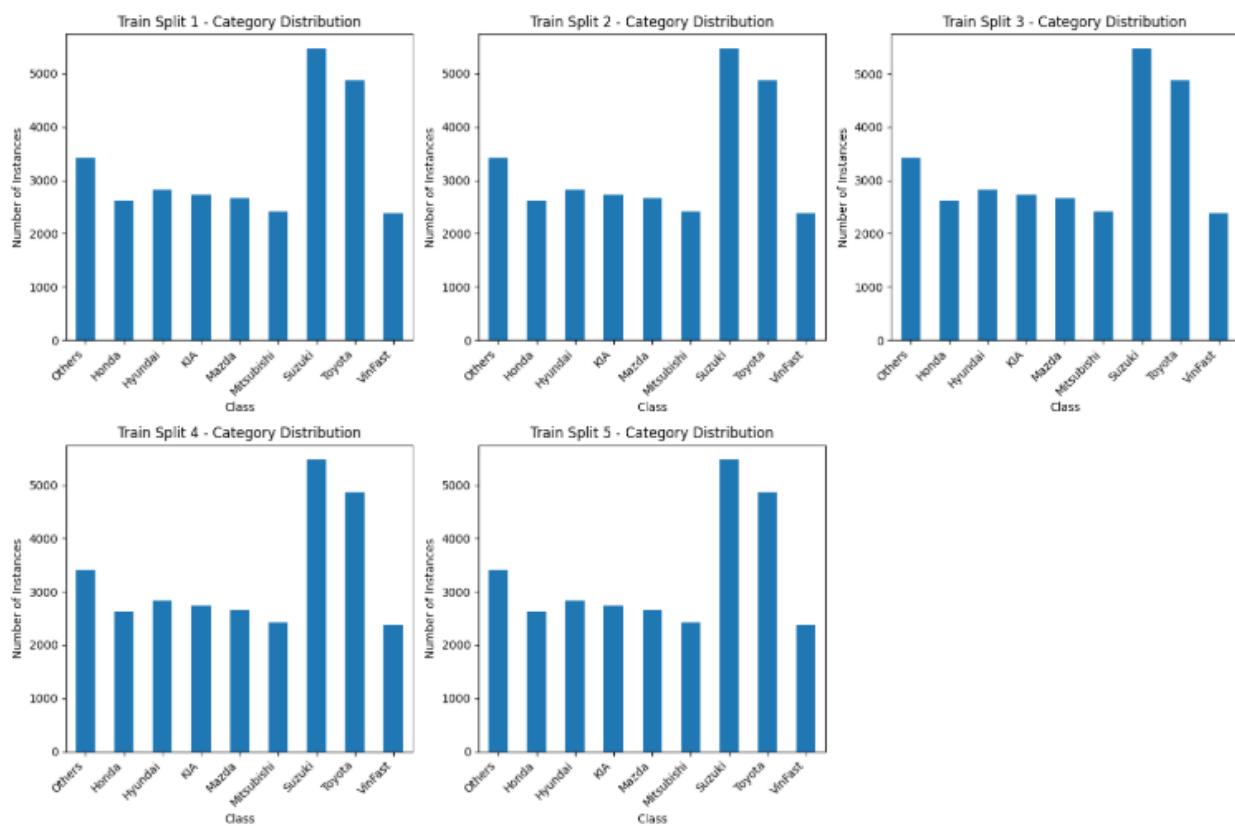
Đồ án "Nhận diện hãng xe qua ảnh chụp" tập trung xây dựng hệ thống phân loại hãng xe dựa trên hình ảnh. Hệ thống ứng dụng kỹ thuật xử lý ảnh và mạng học sâu để trích xuất đặc trưng và nhận diện đặc trưng của từng hãng xe. Hệ thống sẽ được huấn luyện trên tập dữ liệu hình ảnh đa dạng về các hãng xe để đảm bảo độ chính xác cao trong quá trình nhận diện. Kết quả đạt được có thể ứng dụng vào các lĩnh vực như giám sát giao thông, quản lý bãi đỗ xe thông minh và hỗ trợ dịch vụ chăm sóc khách hàng trong ngành ô tô.

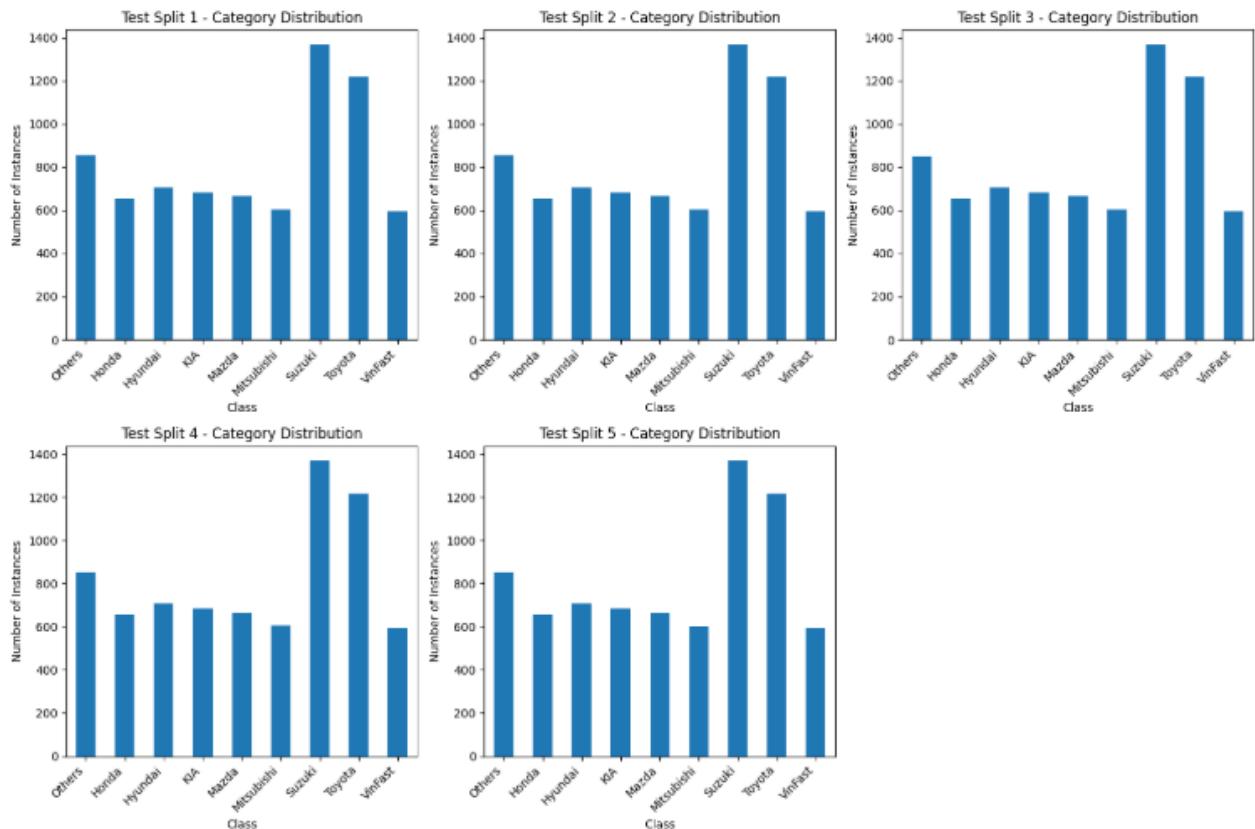
## Chương 2: CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN

Bước 1: Thực hiện các bài tập được giao.

Bài tập chia data

```
CS114.P11\Final_project\car_classification\processing\  
tool-createsplit-car.ipynb
```





## Bài tập datasetviz

CS114.P11\Final\_project\car\_classification\processing\  
tool-datasetviz.ipynb

Others (CategoryID: 0) - Total Images: 4262



Không thể đọc ảnh: /kaggle/input/car-cs114/dataset/Honda/22521560-22521614.Honda.37.jpg

Honda (CategoryID: 1) - Total Images: 3279



Hyundai (CategoryID: 2) - Total Images: 3530



KIA (CategoryID: 3) - Total Images: 3414



|

Mazda (CategoryID: 4) - Total Images: 3325



Mitsubishi (CategoryID: 5) - Total Images: 3019



Suzuki (CategoryID: 6) - Total Images: 6842



Toyota (CategoryID: 7) - Total Images: 6092



VinFast (CategoryID: 8) - Total Images: 2974

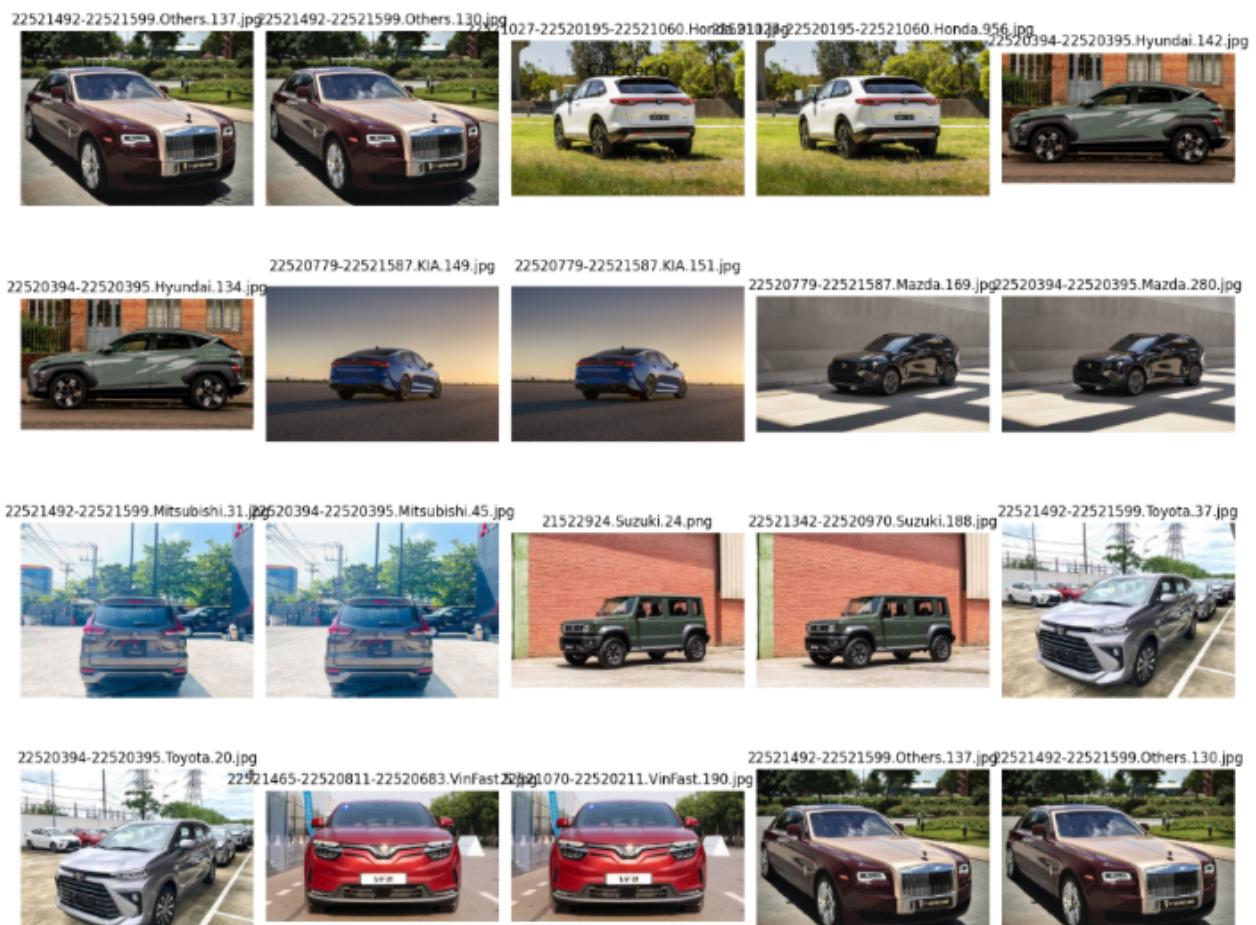


## Bài tập clustering

CS114.P11\Final\_project\car\_classification\processing\  
checkClustering.md

CS114.P11\Final\_project\car\_classification\processing\  
clustering.ipynb

Một số kết quả:



## Bài tập thống kê dữ liệu

CS114.P11\Final\_project\car\_classification\processing\  
tool-datasetstat-car.ipynb

	MSSV	All	Số lượng
0	22521027	All	11969
1	22520195	All	11969
2	22521060	All	11969
3	22520459	All	3779
4	22520507	All	3779
5	22520862	All	3779
6	22520394	All	3367
7	22520395	All	3367
8	22520779	All	2721
9	22521587	All	2721
10	22521070	All	2460
11	22520211	All	2460

	MSSV	Hiệu xe	Số lượng
0	22521027	Honda	1000
1	22520195	Honda	1000
2	22521060	Honda	1000
3	22520394	Honda	387
4	22520395	Honda	387
	...	...	...
416	22520223	VinFast	20
417	22520213	VinFast	20
418	22521463	VinFast	15
419	22521213	VinFast	15
420	22521259	VinFast	15

```
path_3 = "/kaggle/working/InvalidNames.csv"
df3 = pd.read_csv(path_3)
df3
```

	ImageFullPath	CategoryID
0	Others/22520348-22520530-22520837.MG.1.jpg	0
1	Others/22520223-22520213.Nissan.8.jpg	0
2	Others/21522373-21522499.LandRover.16.jpg	0
3	Others/21522373-21522499.Nissan.7.jpg	0
4	Others/22520223-22520213.Peugeot.6.jpg	0
	...	...
1302	VinFast/21522500-21522771.VinFast20.jpg	8
1303	VinFast/21520930-21522924.VinFast56-.png	8
1304	VinFast/21522500-21522771.VinFast25.jpg	8
1305	VinFast/21522500-21522771.VinFast18.jpg	8
1306	VinFast/21520930-21522924.VinFast31-.png	8

## Bước 2: Khảo sát dữ liệu

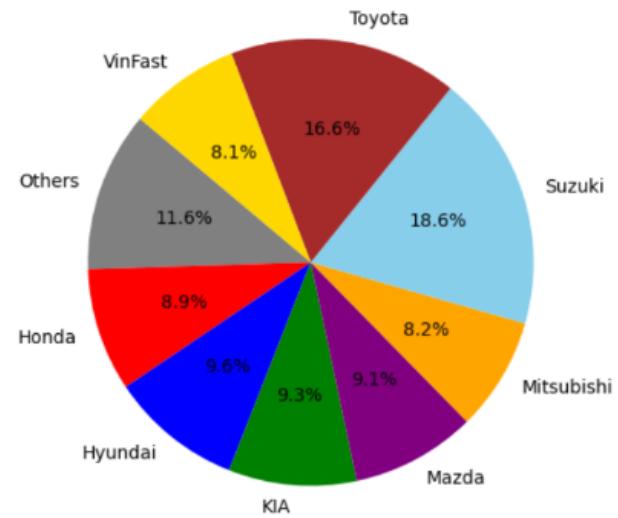
CS114.P11\Final\_project\car\_classification\processing\survey\_statistics\_data.ipynb

### Thống kê tổng số ảnh theo từng hiệu xe

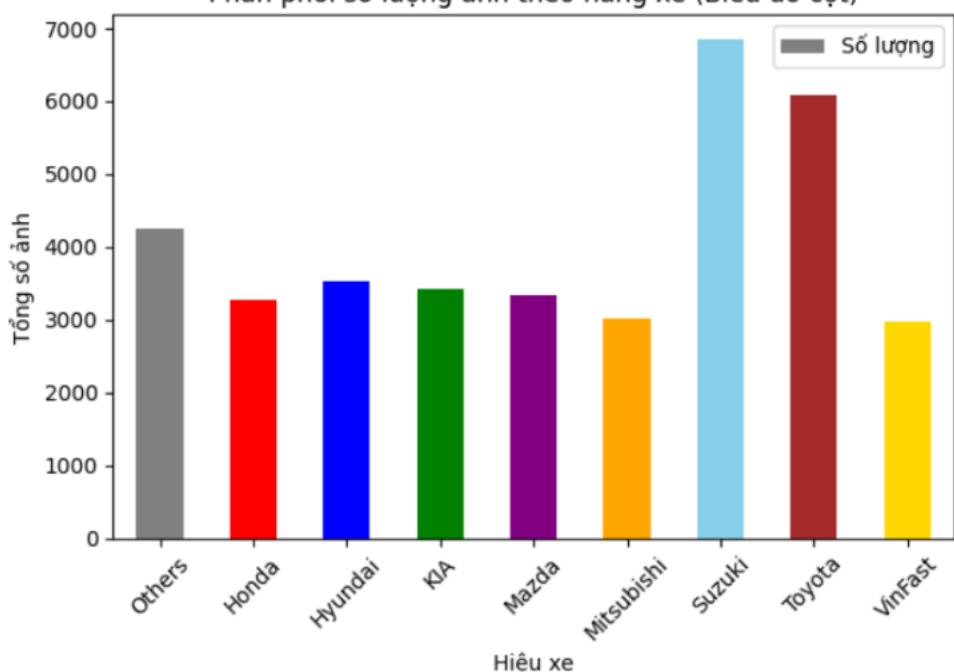
Tổng số ảnh theo từng hãng xe:

	Hiệu xe	Số lượng
0	Others	4262
1	Honda	3279
2	Hyundai	3530
3	KIA	3414
4	Mazda	3325
5	Mitsubishi	3019
6	Suzuki	6842
7	Toyota	6092
8	VinFast	2974

Phân phối số lượng ảnh theo hiệu xe (Biểu đồ tròn)

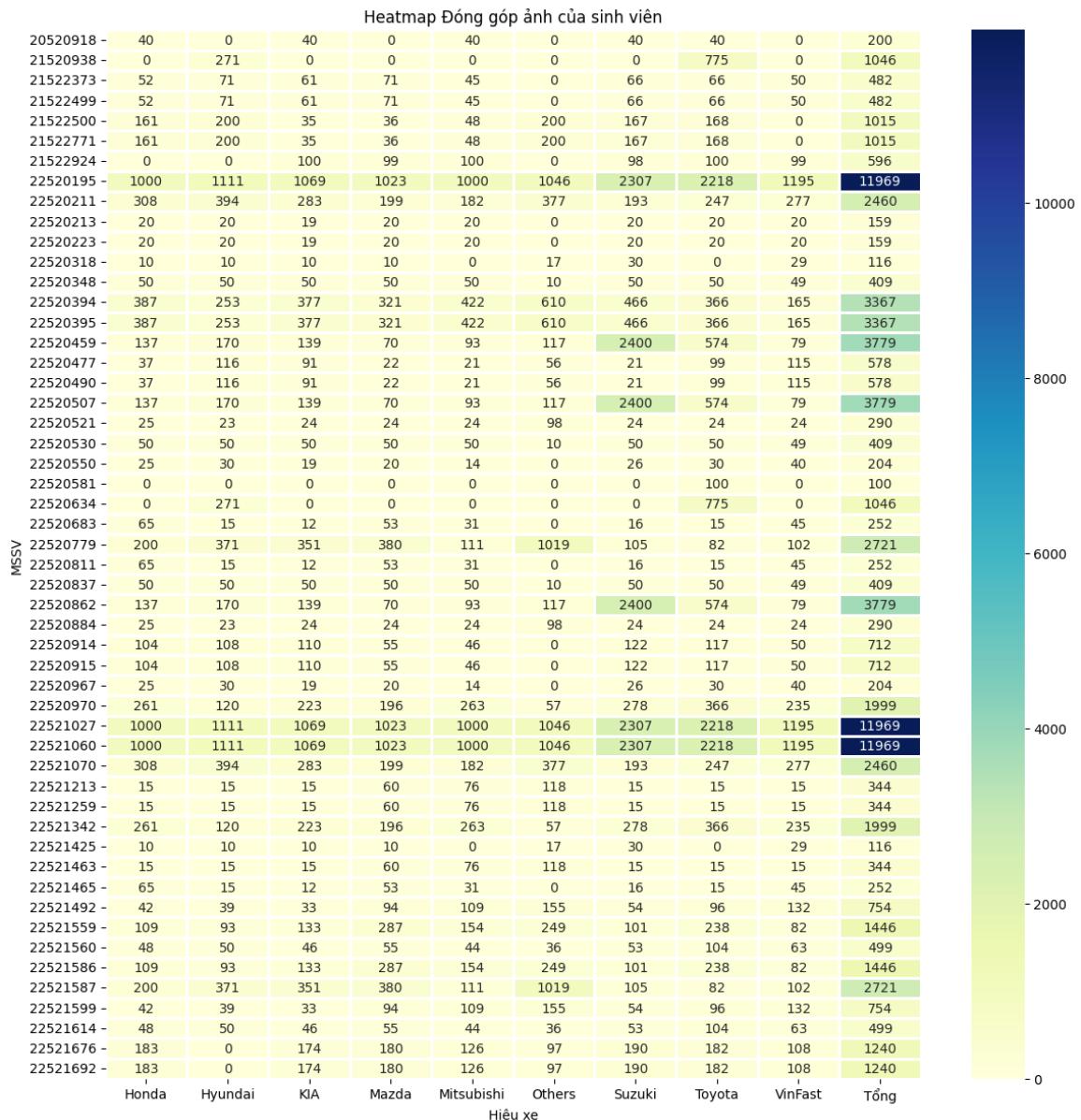


Phân phối số lượng ảnh theo hãng xe (Biểu đồ cột)



⇒ Số lượng ảnh của các hiệu xe không đồng đều. Có sự chênh lệch khá lớn giữa các hiệu xe, với Suzuki và Toyota có số lượng ảnh lớn hơn đáng kể so với các hiệu khác.

## Thống kê dữ liệu theo đóng góp của từng sinh viên



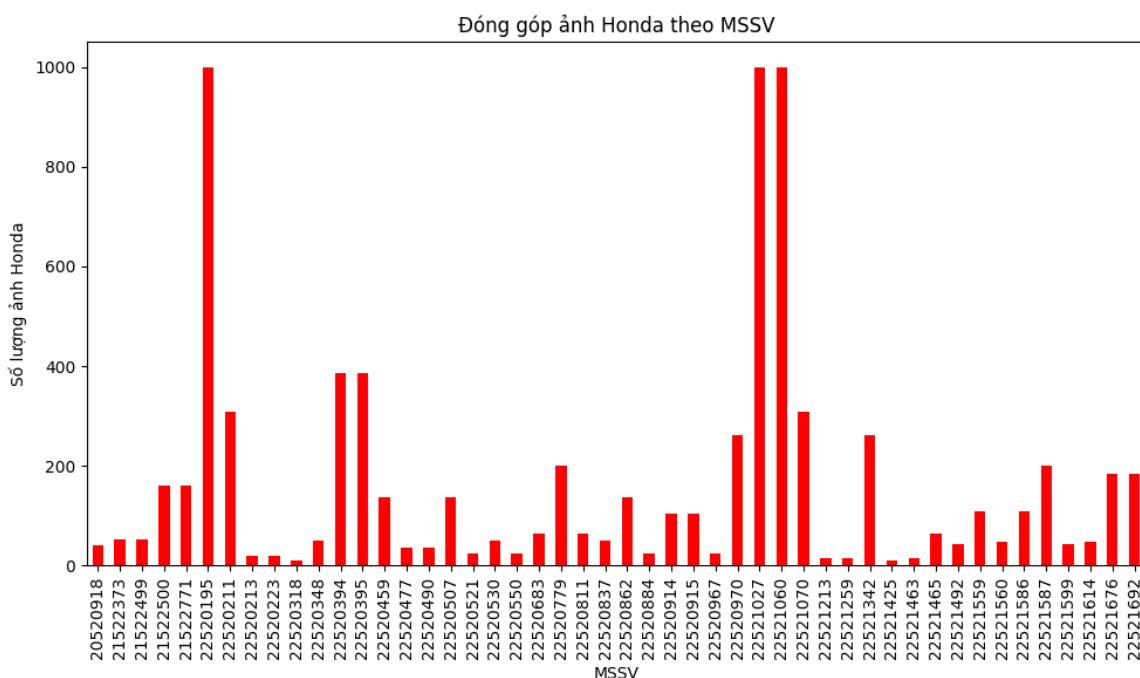
Nhìn vào heatmap, chúng ta có thể nhận thấy:

- Sự Biến Thiện Về Số Lượng Đóng Góp:

+ Theo sinh viên: Các sinh viên có mức độ đóng góp rất khác nhau. Một số đóng góp rất nhiều ảnh (màu xanh đậm), trong khi một số khác không đóng góp hoặc đóng

góp rất ít (màu trắng hoặc nhạt). Phần lớn sinh viên đóng góp ở mức trung bình, thể hiện qua màu vàng.

- + Theo hiệu xe: Tương tự, sự đóng góp ảnh cho từng hiệu xe cũng không đồng đều. Dòng "Others" có số lượng ảnh đóng góp cao hơn hẳn so với các hiệu xe cụ thể, thể hiện sự chú ý đặc biệt từ sinh viên.
- Tập Trung Vào Dòng "Others": Dòng "Others" nhận được một lượng đóng góp ảnh lớn, điều này cho thấy sự quan tâm đặc biệt từ sinh viên đối với nhóm xe này. So với các hiệu xe cụ thể khác, số lượng đóng góp cho "Others" thường cao hơn rất nhiều.
- Phân Bố Số Lượng Đóng Góp:
  - + Một số sinh viên đóng góp ảnh cho hầu hết các hiệu xe, trong khi những sinh viên khác chỉ đóng góp cho một số hiệu xe nhất định.
  - + Có những sinh viên chỉ đóng góp một lượng nhỏ hoặc thậm chí không đóng góp cho tất cả các hiệu xe.
- Cột "Tổng": Cột "Tổng" trong heatmap thể hiện tổng số ảnh đóng góp của mỗi sinh viên cho tất cả các hiệu xe. Đây là một chỉ số quan trọng, giúp so sánh trực tiếp số lượng đóng góp giữa các sinh viên.



- Sự phân bố không đồng đều:
  - + Có sự khác biệt rất lớn về số lượng ảnh đóng góp giữa các sinh viên.
  - + Một số sinh viên (ví dụ: 22520195, 22521027, 22521060) có số lượng ảnh đóng góp rất cao, thể hiện qua các cột dossier rất cao trên biểu đồ.
  - + Ngược lại, phần lớn các sinh viên có số lượng ảnh đóng góp thấp hoặc không có, thể hiện qua các cột ngắn hoặc không có cột.
- Các sinh viên đóng góp nổi bật:
  - + Nhóm ba sinh viên (22520195, 22521027, 22521060) có đóng góp vượt trội so với tất cả các sinh viên còn lại, với số lượng ảnh đóng góp gần như ngang nhau.
  - + Các sinh viên khác có cột thấp hơn nhiều hoặc không có, cho thấy đóng góp của họ là ít hoặc không có ảnh hưởng về Honda.
- Mô hình đóng góp:
  - + Biểu đồ cho thấy rõ một số ít sinh viên đóng góp phần lớn số lượng ảnh, trong khi phần lớn các sinh viên khác đóng góp rất ít.
  - + Sự phân bố này không đồng đều, không có nhiều sinh viên đóng góp ở mức trung bình.

Tương tự, với các hiệu xe khác được trình bày ở cell code số 6 trong

```
"CS114.P11\Final_project\car_classification\processing\
survey_statistics_data.ipynb"
```

---- ĐÓNG GÓP TOÀN BỘ ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	11969	22521027, 22520195, 22521060
ít nhất	100	22520581

---- ĐÓNG GÓP CHO HÃNG MITSUBISHI ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	1000	22521027, 22520195, 22521060
ít nhất	14	22520550, 22520967

---- ĐÓNG GÓP CHO HÃNG HONDA ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	1000	22521027, 22520195, 22521060
ít nhất	10	22521425, 22520318

---- ĐÓNG GÓP CHO HÃNG OTHERS ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	1046	22521027, 22520195, 22521060
ít nhất	10	22520348, 22520530, 22520837

---- ĐÓNG GÓP CHO HÃNG HYUNDAI ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	1111	22521027, 22520195, 22521060
ít nhất	10	22521425, 22520318

---- ĐÓNG GÓP CHO HÃNG SUZUKI ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	2400	22520459, 22520507, 22520862
ít nhất	15	22521463, 22521213, 22521259

---- ĐÓNG GÓP CHO HÃNG KIA ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	1069	22521027, 22520195, 22521060
ít nhất	10	22521425, 22520318

---- ĐÓNG GÓP CHO HÃNG TOYOTA ----

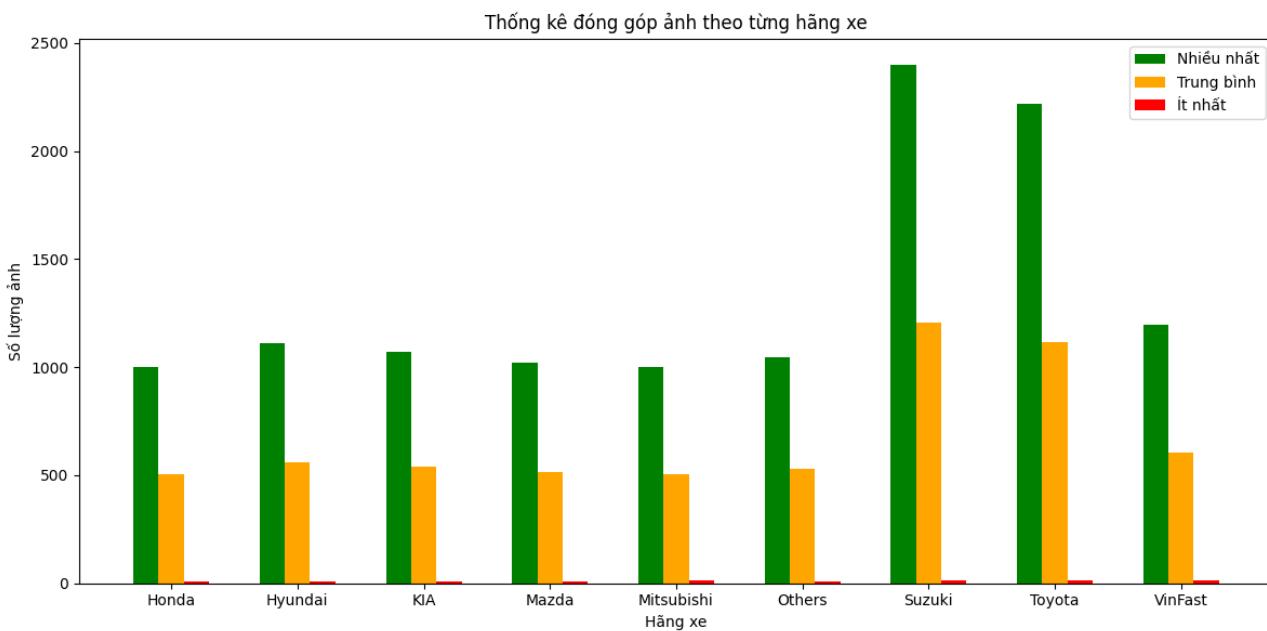
Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	2218	22521027, 22520195, 22521060
ít nhất	15	22521463, 22521213, 22521259, 22520683, 22520811, 22521465

---- ĐÓNG GÓP CHO HÃNG MAZDA ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	1023	22521027, 22520195, 22521060
ít nhất	10	22521425, 22520318

---- ĐÓNG GÓP CHO HÃNG VINFAST ----

Loại	Số lượng ảnh	Danh sách HSSV
Nhiều nhất	1195	22521027, 22520195, 22521060
ít nhất	15	22521463, 22521213, 22521259



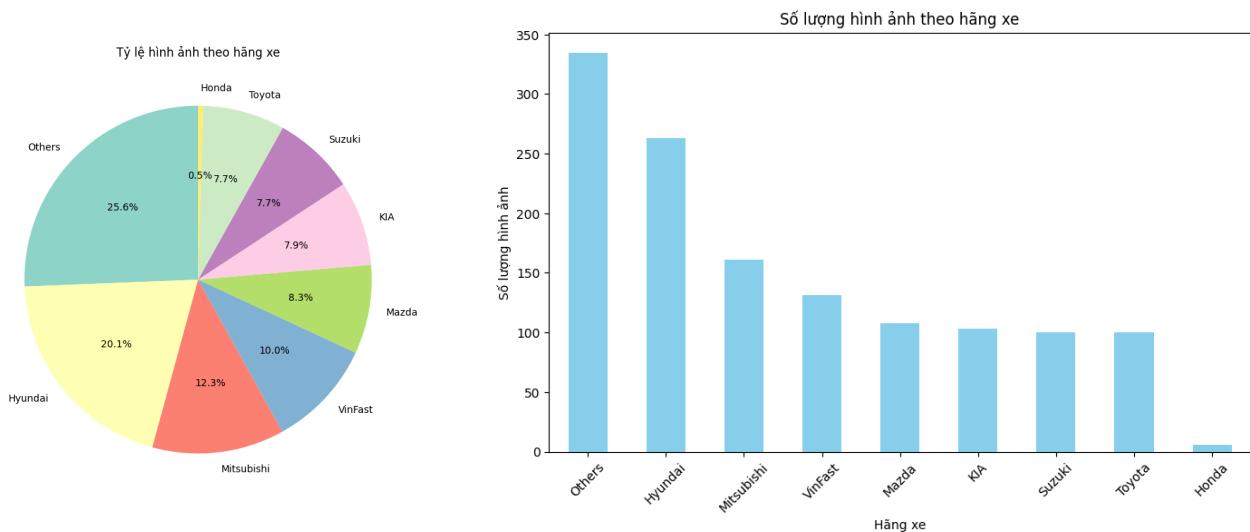
- 3 Sinh viên nổi bật: Các sinh viên 22521027, 22520195, và 22521060 có đóng góp lớn nhất cho hầu hết các hiệu xe và tổng số.

- Phân bố không đồng đều: Sự chênh lệch rất lớn giữa người đóng góp nhiều nhất và ít nhất cho thấy sự phân bố đóng góp không đồng đều.

## Thống kê ảnh có tên không hợp lệ

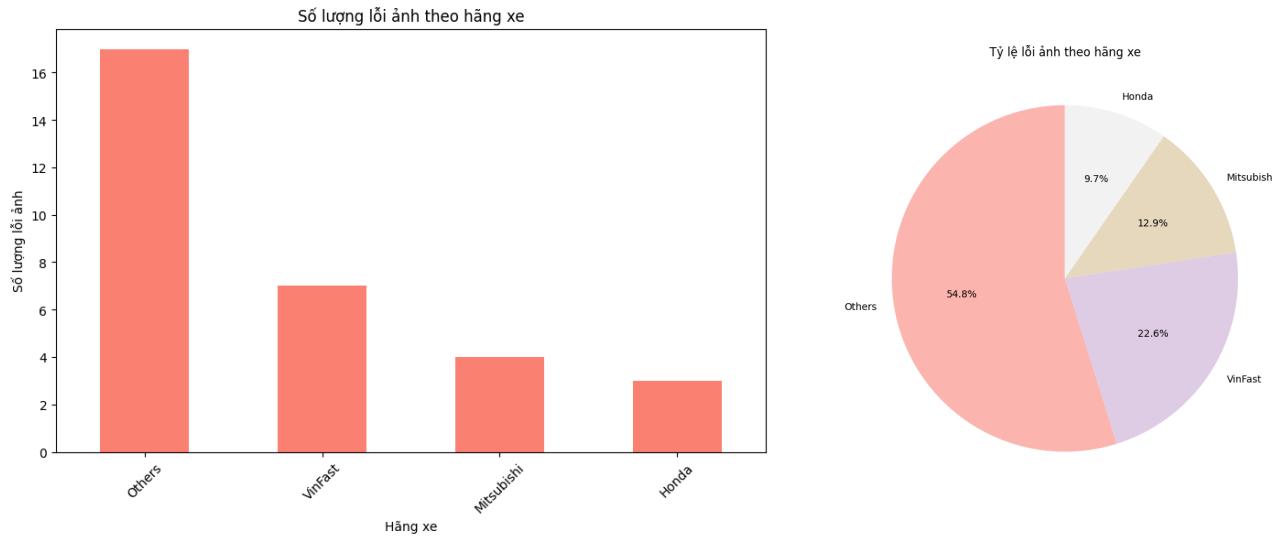
Tên hợp lệ là tên có cấu trúc:

mssv1-mssv2-mssv3.{Tên \_ thư \_ mục \_ hiệu \_ xe}.{stt}.{jpg/png/jpeg}



- "Others" có tỷ lệ ảnh không hợp lệ cao nhất (25.6%):
  - + Điều này cho thấy có một lượng lớn ảnh không có tên đúng cấu trúc được tìm thấy trong thư mục "Others".
  - + Có thể có nhiều nguyên nhân, ví dụ: tên file bị sai quá nhiều và không thể phân loại vào các hãng xe khác, hoặc một lỗi hệ thống nào đó.
- Hyundai (20.1%) cũng có tỷ lệ không hợp lệ đáng kể.
- Các thư mục còn lại cũng có các ảnh không hợp lệ:
  - + Các thư mục còn lại (Mitsubishi, VinFast, Mazda, KIA, Suzuki, Toyota, và Honda) cũng chứa các ảnh có tên không đúng cấu trúc, nhưng tỷ lệ thấp hơn so với "Others" và Hyundai.
- Honda có tỷ lệ không hợp lệ thấp nhất (0.5%).

## Thống kê ảnh lỗi không thể đọc được thông qua các thư viện như PIL hay cv2



- "Others" chiếm phần lớn số lượng ảnh lỗi:
  - + Thư mục "Others" có tỷ lệ lỗi cao nhất, chiếm đến 54.8% tổng số ảnh lỗi. Điều này cho thấy có một số lượng rất lớn ảnh không mở được nằm trong thư mục này.
  - + Nguyên nhân có thể do các lỗi định dạng, lỗi file, hoặc tên file không hợp lệ gây ra lỗi khi hệ thống cố gắng mở ảnh.
- VinFast đứng thứ hai về số lượng ảnh lỗi:
  - + VinFast có tỷ lệ lỗi là 22.6%, cho thấy cũng có một lượng đáng kể ảnh lỗi trong thư mục này.
- Mitsubishi có tỷ lệ lỗi thấp hơn:
  - + Mitsubishi có tỷ lệ ảnh lỗi là 12.9%, thấp hơn so với "Others" và VinFast.
- Honda có tỷ lệ lỗi thấp nhất:
  - + Honda có tỷ lệ ảnh lỗi là 9.7%, cho thấy thư mục này ít gặp phải vấn đề về ảnh lỗi nhất trong số các thư mục được thể hiện trên biểu đồ.

## Bước 3: Xử lý data

- Dữ liệu sử dụng là các tập *split* từ Bước 1.1.
- Loại bỏ ảnh không đọc được và ảnh trùng lặp (áp dụng cho tập *train*, kế thừa từ Bước 1.3).

	ImageFullPath	CategoryID
0	Others/22520394-22520395.Others.547.jpg	0
1	Others/22520394-22520395.Others.181.jpg	0
2	Others/22520459-22520507-22520862.Others.23.jpg	0
3	Others/22521027-22520195-22521060.Others.307.jpg	0
4	Others/22520394-22520395.Others.578.jpg	0
...	...	...
29382	VinFast/22521070-22520211.VinFast.167.jpg	8
29383	VinFast/21522373-21522499.VinFast.10.png	8
29384	VinFast/22521692-22521676.VinFast.7.jpg	8
29385	VinFast/22521259-22521213-22521463.VinFast.13.jpg	8
29386	VinFast/22521027-22520195-22521060.VinFast.092...	8

29387 rows × 2 columns

(a) Tập train ban đầu

	ImageFullPath	CategoryID
0	Others/22520394-22520395.Others.547.jpg	0
1	Others/22520394-22520395.Others.181.jpg	0
2	Others/22520459-22520507-22520862.Others.23.jpg	0
3	Others/22521027-22520195-22521060.Others.307.jpg	0
4	Others/22520394-22520395.Others.578.jpg	0
...	...	...
29382	VinFast/22521070-22520211.VinFast.167.jpg	8
29383	VinFast/21522373-21522499.VinFast.10.png	8
29384	VinFast/22521692-22521676.VinFast.7.jpg	8
29385	VinFast/22521259-22521213-22521463.VinFast.13.jpg	8
29386	VinFast/22521027-22520195-22521060.VinFast.092...	8

29369 rows × 2 columns

(b) tập train sau khi loại bỏ ảnh lỗi

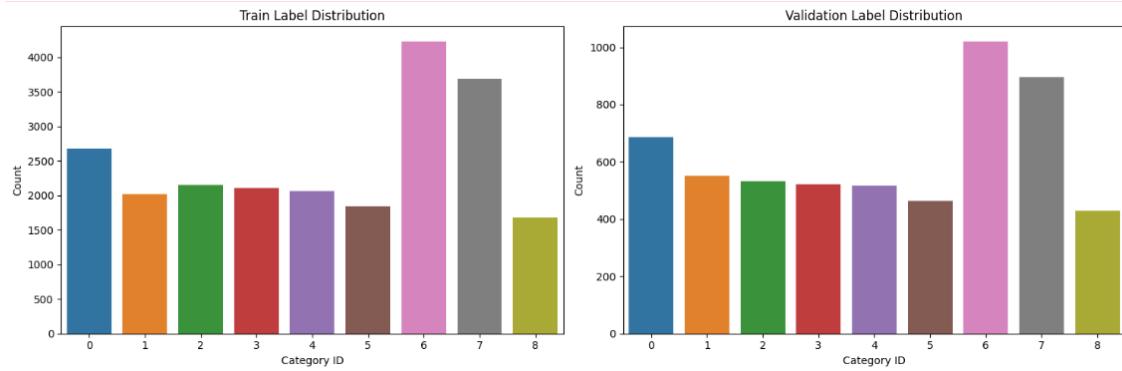
	ImageFullPath	CategoryID
0	Others/22520394-22520395.Others.547.jpg	0
1	Others/22520394-22520395.Others.181.jpg	0
2	Others/22520459-22520507-22520862.Others.23.jpg	0
3	Others/22521027-22520195-22521060.Others.307.jpg	0
4	Others/22520394-22520395.Others.578.jpg	0
...	...	...
29381	VinFast/22521560-22521614.VinFast.23.jpg	8
29382	VinFast/22521070-22520211.VinFast.167.jpg	8
29383	VinFast/21522373-21522499.VinFast.10.png	8
29385	VinFast/22521259-22521213-22521463.VinFast.13.jpg	8
29386	VinFast/22521027-22520195-22521060.VinFast.092...	8

28058 rows × 2 columns

(c) tập train sau khi loại bỏ ảnh trùng

- Chia tập *train* thành *train* mới và *validation* theo tỷ lệ 8:2.

```
Counter({6: 4231, 7: 3684, 0: 2682, 2: 2147, 3: 2111, 4: 2061, 1: 2016, 5: 1840, 8: 1674})
Counter({6: 1021, 7: 895, 0: 687, 1: 550, 2: 531, 3: 521, 4: 515, 5: 462, 8: 430})
```



- Tiền xử lý ảnh cho các tập *train*, *val*, *test* bằng cách áp dụng một khối transform lên ảnh:
  - + Chuyển ảnh sang RGB: Dảm bảo tất cả ảnh đầu vào có định dạng RGB đồng nhất.
  - + Thay đổi kích thước: Dưa chiều dài ngắn nhất của ảnh về 224 hoặc 384 pixel bằng nội suy bilinear hoặc bicubic.
  - + Cắt ảnh từ trung tâm: Cắt ảnh thành kích thước  $224 \times 224$  hoặc  $384 \times 384$  từ trung tâm.
  - + Chuyển sang tensor: Biến đổi ảnh thành tensor với giá trị pixel từ 0 đến 1.
  - + Chuẩn hóa ảnh: Theo giá trị trung bình và độ lệch chuẩn của tập ImageNet.
  - + Tăng cường dữ liệu: Áp dụng *Random Horizontal Flip* cho tập *train*.

Kết quả sau khi áp dụng lớp transform lên 1 ảnh ví dụ như sau:

- Với 1 tensor: ảnh đã được chuyển thành tensor.
- Số 4: nhãn.

```

train_set[0]

(tensor([[1.3413, 1.8379, 1.8037, ..., 2.2318, 2.1804, 2.2318],
       [1.1872, 1.9064, 1.7352, ..., 2.2318, 2.1804, 2.2318],
       [0.9817, 1.9235, 1.6838, ..., 2.2318, 2.1975, 2.2318],
       ...,
       [2.0948, 2.0777, 2.1290, ..., 2.0605, 2.0263, 2.0777],
       [2.0605, 2.0434, 2.1290, ..., 2.1290, 2.0777, 2.1462],
       [1.9920, 1.9749, 2.0777, ..., 2.1804, 2.1290, 2.1633]],

      [[1.6232, 2.1310, 2.0959, ..., 2.1835, 2.0784, 2.1660],
       [1.4657, 2.2010, 2.0259, ..., 2.1660, 2.0784, 2.1660],
       [1.2556, 2.2185, 1.9734, ..., 2.1485, 2.0959, 2.1485],
       ...,
       [2.1134, 2.0959, 2.1485, ..., 2.1485, 2.1134, 2.1660],
       [2.0784, 2.0609, 2.1485, ..., 2.2185, 2.1660, 2.2360],
       [2.0084, 1.9909, 2.0959, ..., 2.2885, 2.2185, 2.2535]],

      [[1.9777, 2.4831, 2.4483, ..., 1.7685, 1.6814, 1.7860],
       [1.8208, 2.5354, 2.3786, ..., 1.7511, 1.6814, 1.7860],
       [1.6117, 2.5354, 2.3263, ..., 1.7337, 1.6814, 1.7685],
       ...,
       [2.2391, 2.2217, 2.2740, ..., 2.3088, 2.2740, 2.3263],
       [2.2043, 2.1868, 2.2740, ..., 2.3786, 2.3263, 2.3960],
       [2.1346, 2.1171, 2.2217, ..., 2.4483, 2.3786, 2.4134]]]),

4)

```

## Bước 4: Lựa chọn mô hình học sâu để trích xuất đặc trưng và huấn luyện

Sử dụng mô hình RegNet\_Y\_128GF với weight = IMAGENET1K\_SWAG\_LINEAR\_V1:

- Đặc trưng trích xuất: Là một mô hình đã được huấn luyện trước trên tập dữ liệu ImageNet, RegNet\_Y\_128GF có khả năng trích xuất các đặc trưng cơ bản (cạnh, góc, kết cấu), trung gian (hình khối, cấu trúc phức tạp), và cao cấp (hình dạng, đối tượng).
- Tinh chỉnh (Fine-Tuning): Freeze toàn bộ mô hình ngoại trừ block4 và thay thế tầng fully connected bằng các lớp sau:

```

nn.Sequential(
    nn.Linear(num_ftrs, 1024),
    nn.ReLU(),
    nn.Dropout(0.5),
    nn.Linear(1024, 512),
    nn.ReLU(),

```

```

        nn.Dropout(0.5),
        nn.Linear(512, 9)

    )

(block4): AnyStage(
    (block4-0): ResBottleneckBlock(
        (proj): Conv2dNormActivation(
            (0): Conv2d(2904, 7392, kernel_size=(1, 1), stride=(2, 2), bias=False)
            (1): BatchNorm2d(7392, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (f): BottleneckTransform(
            (a): Conv2dNormActivation(
                (0): Conv2d(2904, 7392, kernel_size=(1, 1), stride=(1, 1), bias=False)
                (1): BatchNorm2d(7392, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (2): ReLU(inplace=True)
            )
            (b): Conv2dNormActivation(
                (0): Conv2d(7392, 7392, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=28, bias=False)
                (1): BatchNorm2d(7392, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (2): ReLU(inplace=True)
            )
            (se): SqueezeExcitation(
                (avgpool): AdaptiveAvgPool2d(output_size=1)
                (fc1): Conv2d(7392, 726, kernel_size=(1, 1), stride=(1, 1))
                (fc2): Conv2d(726, 7392, kernel_size=(1, 1), stride=(1, 1))
                (activation): ReLU()
                (scale_activation): Sigmoid()
            )
            (c): Conv2dNormActivation(
                (0): Conv2d(7392, 7392, kernel_size=(1, 1), stride=(1, 1), bias=False)
                (1): BatchNorm2d(7392, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            )
        )
        (activation): ReLU(inplace=True)
    )
)
)
(avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
(fc): Sequential(
    (0): Linear(in_features=7392, out_features=1024, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=1024, out_features=512, bias=True)
    (4): ReLU()
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=512, out_features=9, bias=True)
)
)

```

---

## Bước 5: Huấn luyện và đánh giá mô hình

- Huấn luyện mô hình (train\_model)

+ Tham số:

\* **Loss Function:** Sử dụng `CrossEntropyLoss` cho bài toán phân loại.

\* **Mixed Precision Training:** Dùng `torch.amp.GradScaler` để tối ưu hiệu suất trên GPU.

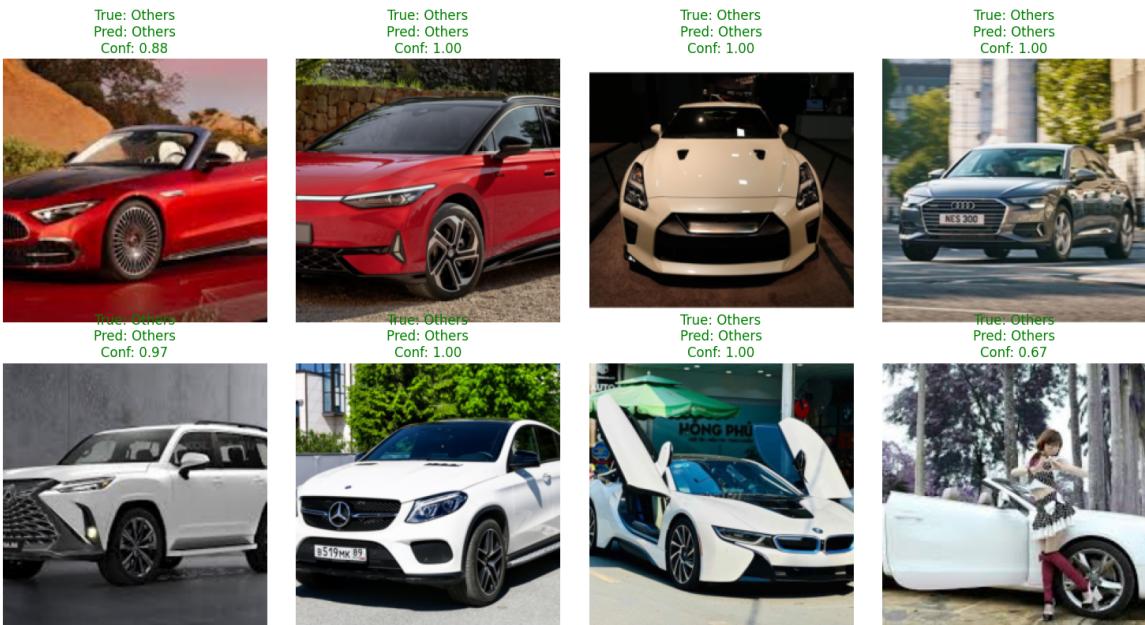
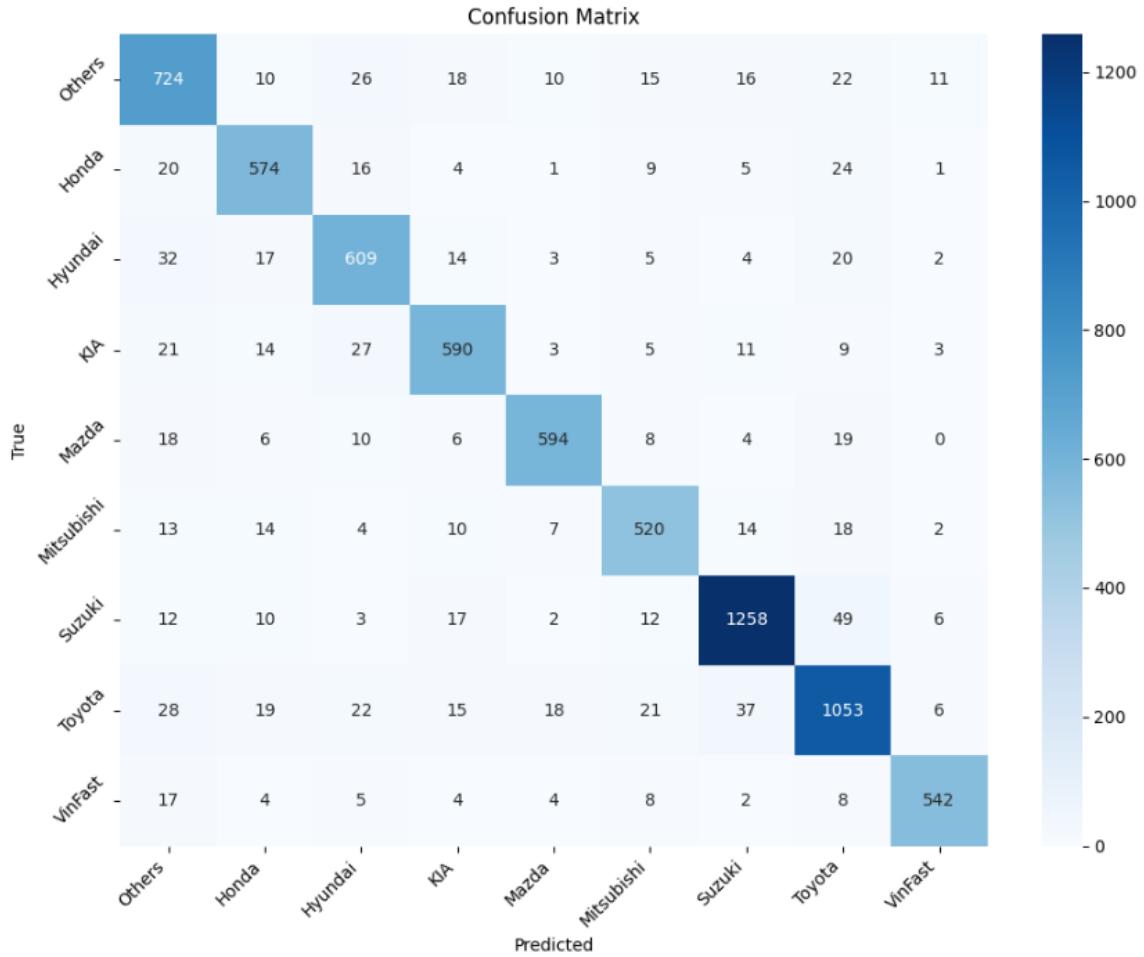
\* **Optimizer:** Adam.

- \* Learning Rate Scheduler: StepLR.
- + Các bước thực hiện:
  - \* Forward pass dữ liệu qua mô hình.
  - \* Tính loss và backward propagation.
  - \* Cập nhật trọng số với optimizer và scheduler.
  - \* Lưu mô hình tốt nhất khi độ chính xác tăng.



- Đánh giá mô hình (evaluate\_and\_visualize)
  - + Accuracy: Tính chính xác trên tập test.
  - + Confusion Matrix: Hiển thị nhầm lẫn giữa các lớp.
  - + Dự đoán trực quan: Hiển thị ảnh với nhãn thật và dự đoán.

Test Accuracy: 0.8802



### Chương 3: KẾT QUẢ NHẬN ĐƯỢC VÀ NHẬN XÉT

Mô hình	Split 1	Split 2	Split 3	Split 4	Split 5
RegNet_Y_128GF	0,8802	0,8755	0,8852	0,8872	0,8909
Linear finetune	7h55p (30)	5h46p (20)	4h42p (20)	5h44p (20)	4h41p (20)

Trong đó: dòng 1 là accuracy trên test, dòng 2 là thời gian training và số epoch.

- Acc trên tập test của mô hình qua các lần split dao động từ 0,8755 đến 0,8909. Hiệu suất giữa các lần split có sự thay đổi nhẹ, cho thấy mô hình tương đối ổn định.
- Dữ liệu cho thấy mặc dù thời gian finetune giảm qua các split, điểm số của mô hình vẫn duy trì hoặc cải thiện nhẹ (từ 0,8802 lên 0,8909). Điều này có thể chứng minh rằng việc finetune không cần kéo dài vẫn đạt hiệu quả tốt nếu quy trình được tối ưu.

## Chương 4: ABLATION STUDY

**Bước 1: Thực hiện các bài tập được giao:** Vì đây là nền tảng để thực hiện các bước tiếp theo

**Bước 2: Khảo sát dữ liệu** Để hiểu rõ hơn về data

**Bước 3: Xử lý data**

- Xử lý ảnh lỗi để tránh code thêm xử lý ảnh lỗi phía sau.
- Chia thành train và val để mục đích theo dõi mô hình trong quá trình huấn luyện.
- Khối transform được áp dụng vì khi tìm kiếm mô hình tại [torchvision.models.regnet\\_y\\_128gf](#) có mô tả phần tham số như trong ảnh phía dưới:

### **RegNet\_Y\_128GF\_Weights.IMAGENET1K\_SWAG\_LINEAR\_V1:**

These weights are composed of the original frozen **SWAG** trunk weights and a linear classifier learnt on top of them trained on ImageNet-1K data.

acc@1 (on ImageNet-1K)	86.068
acc@5 (on ImageNet-1K)	97.844
min_size	height=1, width=1
categories	tench, goldfish, great white shark, ... (997 omitted)
recipe	<a href="#">link</a>
license	<a href="#">link</a>
num_params	644812894
GFLOPS	127.52
File size	2461.6 MB

The inference transforms are available at `RegNet_Y_128GF_Weights.IMAGENET1K_SWAG_LINEAR_V1.transforms` and perform the following preprocessing operations: Accepts `PIL.Image`, `batched (B, C, H, W)` and `single (C, H, W) image torch.Tensor objects`. The images are resized to `resize_size=[224]` using `interpolation=InterpolationMode.BICUBIC`, followed by a central crop of `crop_size=[224]`. Finally the values are first rescaled to `[0.0, 1.0]` and then normalized using `mean=[0.485, 0.456, 0.406]` and `std=[0.229, 0.224, 0.225]`.

### **Bước 4 và 5 chương 2**

Nhóm thử nghiệm 3 mô hình: RegNet\_Y\_128GF với Weight = IMAGENET1K\_SWAG\_LINEAR\_V1 không finetune, RegNet\_Y\_128GF với Weight = IMAGENET1K\_SWAG\_LINEAR\_V1 finetune

block4 với thay đổi lớp fc và RegNet\_Y\_128GF với Weight = IMAGENET1K\_SWAG\_E2E\_V1, thu được kết quả: :

Mô hình	Split 1	Split 2	Split 3	Split 4	Split 5
RegNet_Y_128GF	0,7188	0,7020	0,7110	0,7213	0,7154
Linear	7h22p (30)	7h21p (30)	7h03p (30)	4h05p (20)	4h55p (20)
RegNet_Y_128GF	0,8802	0,8755	0,8852	0,8872	0,8909
Linear finetune	7h55p (30)	5h46p (20)	4h42p (20)	5h44p (20)	4h41p (20)
RegNet_Y_128GF	0,8806	0,8714	0,8822	0,8790	0,8814
E2E Finetune	11h32p (15)	11h31p (15)	9h01p (15)	11h32p (15)	11h4p (20)

Trong đó: dòng 1 là accuracy trên test, dòng 2 là thời gian training và số epoch.

⇒ Có thể thấy mô hình RegNet\_Y\_128GF với weight = IMAGENET1K\_SWAG\_LINEAR\_V1 được finetune Linear Finetune cho hiệu suất cao nhất về độ chính xác trên test với thời gian huấn luyện hợp lý.

Do đó: nhóm đã chọn mô hình RegNet\_Y\_128GF với weight = IMAGENET1K\_SWAG\_LINEAR\_V1 được finetune.

## Chương 5: TÀI LIỆU THAM KHẢO

[1 ] [https://pytorch.org/vision/main/models/generated/torchvision.models.regnet\\_y\\_128gf.html](https://pytorch.org/vision/main/models/generated/torchvision.models.regnet_y_128gf.html)

được sử dụng ở bước 3, 4 chương 2 và chương 4.

[2 ] <https://www.kaggle.com/code/satishgunjal/tutorial-k-fold-cross-validation> được sử dụng

ở bước 1 chương 2.

[3 ] <https://pytorch.org/vision/main/generated/torchvision.transforms.RandomHorizontalFlip.html>

được sử dụng ở bước 3 chương 2.

[4 ] <https://seaborn.pydata.org/generated/seaborn.heatmap.html> được sử dụng ở bước 2

chương 2.