Homework 3 – AI

1. **Danh sách nhóm 17**
    a. Nguyễn Thái An – 20020278
    b. Nguyễn Đức Anh – 20020074
    c. Lê Tuấn Anh – 20021286
2. **Giới thiệu bài toán**

- Trong bài này, bài toán nhóm sẽ thực hiện là bài toán dự đoán nhãn thu nhập của người lao động trên tập dữ liệu của cuộc thi UET Hackathon 2022 Data Science.
- Trong đó, các nhãn của dữ liệu được liệt kê trong bảng sau.

Nhãn	**Ý nghĩa**
7	Rất cao
6	Trung bình cao
5	Cao
4	Trung bình
3	Thấp
2	Trung bình thấp
1	Rất thấp

- Ở bài này, có 6 file dữ liệu chia thành 3 thể loại (info, work và label) cho hai tập (train và test). Nhưng trong bài này, nhóm sẽ chỉ sử dụng tập train để huấn luyện và kiểm thử.
- Ở tập info, có những trường sau đây:

Tên cột	**Ý nghĩa**
birthYear	Ngày sinh
gender	Giới tính
address	Địa chỉ
id_bh	Mã cá nhân

- Ở tập work, có những trường sau đây:

Tên cột	**Ý nghĩa**
id_bh	Mã cá nhân
id_management	Mã đơn vị quản lý
id_office	Mã văn phòng
company_type	Loại hình công ty
job/role	Vị trí, chức vụ của cá nhân
from_date	Công việc bắt đầu từ ngày nào
to_date	Công việc kết thúc vào ngày nào
employee_lv	Cấp độ của chức vụ

address	Địa chỉ

Ở tập label, có những trường như sau:

Tên cột	**Ý nghĩa**
id_bh	Mã cá nhân
label	Nhãn của thu nhập

Trong đó nhãn label là cột mà nhóm sẽ đi dự đoán.

3. Xử lý dữ liệu

a. Xử lý các file liên quan đến train đầu tiên

- Ta thấy ở cả 3 tập dữ liệu, đều có cùng một cột Mã khách hang, vì vậy ta cần ghép 3 tập dữ liệu lại với nhau để tiện cho quá trình xử lý.
- Sau khi ghép, ta sẽ có một tập dữ liệu lớn hơn với 247559 dòng và 14 cột

	id	id_bh	id_management	id_office	company_type	job/role	from_date	to_date	employee_lv	address_x	birthYear	gender	address_y	label
0	1	113039360	106	TF2212F	-1	Giám đốc	20130100	20151200	7.0	Hà Nội	1971	MALE	hà Nội	4
1	1	113039360	106	TF2212F	-1	Giám đốc	20160100	20220400	10.0	Hà Nội	1971	MALE	hà Nội	4
2	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20160600	20161200	7.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2
3	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20170100	20170300	8.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2
4	2	116074930	102	NaN	-1	NaN	20170400	20170700	-1.0	NaN	1993	FEMALE	Thành phố Hà Nội	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
247554	55008	2616117553	2600	YN0027Z	-1	Công nhân ĐM rôbốt	20100800	20100900	2.0	NaN	1984	MALE	NaN	2
247555	55008	2616117553	2600	NaN	-1	NaN	20101000	20161000	-1.0	NaN	1984	MALE	NaN	2
247556	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20161100	20171200	6.0	Vĩnh Phúc	1984	MALE	NaN	2
247557	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20180100	20191200	8.0	Vĩnh Phúc	1984	MALE	NaN	2
247558	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20200100	20210200	9.0	Vĩnh Phúc	1984	MALE	NaN	2

247559 rows × 14 columns

- Sau đó, gộp chung các thông tin address_x và address_y thành 1 cột chung là address

	id	id_bh	id_management	id_office	company_type	job/role	from_date	to_date	employee_lv	address_x	birthYear	gender	address_y	label	address
0	1	113039360	106	TF2212F	-1	Giám đốc	20130100	20151200	7.0	Hà Nội	1971	MALE	hà Nội	4	Hà Nội
1	1	113039360	106	TF2212F	-1	Giám đốc	20160100	20220400	10.0	Hà Nội	1971	MALE	hà Nội	4	Hà Nội
2	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20160600	20161200	7.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2	Hà Nội
3	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20170100	20170300	8.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2	Hà Nội
4	2	116074930	102	NaN	-1	NaN	20170400	20170700	-1.0	NaN	1993	FEMALE	Thành phố Hà Nội	2	Thành phố Hà Nội
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
247554	55008	2616117553	2600	YN0027Z	-1	Công nhân ĐM rôbốt	20100800	20100900	2.0	NaN	1984	MALE	NaN	2	NaN
247555	55008	2616117553	2600	NaN	-1	NaN	20101000	20161000	-1.0	NaN	1984	MALE	NaN	2	NaN
247556	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20161100	20171200	6.0	Vĩnh Phúc	1984	MALE	NaN	2	Vĩnh Phúc
247557	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20180100	20191200	8.0	Vĩnh Phúc	1984	MALE	NaN	2	Vĩnh Phúc
247558	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20200100	20210200	9.0	Vĩnh Phúc	1984	MALE	NaN	2	Vĩnh Phúc

- Lấy dữ liệu năm chính xác từ hàng to_date, ta lấy 4 số đầu, bỏ 4 số sau bởi nó không có ý nghĩa. Từ 4 số đầu tiên, ta ra được năm

	id	id_bh	id_management	id_office	company_type	job/role	from_date	to_date	employee_lv	address_x	bithYear	gender	address_y	label	address	lastest
0	1	113039360	106	TF2212F	-1	Giám đốc	20130100	20151200	7.0	Hà Nội	1971	MALE	hà Nội	4	Hà Nội	2015
1	1	113039360	106	TF2212F	-1	Giám đốc	20160100	20220400	10.0	Hà Nội	1971	MALE	hà Nội	4	Hà Nội	2022
2	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20160800	20161200	7.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2	Hà Nội	2016
3	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20170100	20170300	8.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2	Hà Nội	2017
4	2	116074930	102	NaN	-1	NaN	20170400	20170700	-1.0	NaN	1993	FEMALE	Thành phố Hà Nội	2	Thành phố Hà Nội	2017
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
247554	55008	2616117553	2600	YN0027Z	-1	Công nhân ĐM robốt	20100800	20100900	2.0	NaN	1984	MALE	NaN	2	NaN	2010
247555	55008	2616117553	2600	NaN	-1	NaN	20101000	20161000	-1.0	NaN	1984	MALE	NaN	2	NaN	2016
247556	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20161100	20171200	6.0	Vĩnh Phúc	1984	MALE	NaN	2	Vĩnh Phúc	2017
247557	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20180100	20191200	8.0	Vĩnh Phúc	1984	MALE	NaN	2	Vĩnh Phúc	2019
247558	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20200100	20210200	9.0	Vĩnh Phúc	1984	MALE	NaN	2	Vĩnh Phúc	2021

- Ta tiến hành kiểm tra tỉ lệ bị thiếu từ dữ liệu.

```
id              0.000000
id_bh           0.000000
id_management   0.000000
id_office       0.049152
company_type    0.000000
job/role        0.135148
from_date       0.000000
to_date         0.000000
employee_lv     0.000020
address_x       0.258649
bithYear        0.000000
gender          0.000000
address_y       0.422740
label           0.000000
dtype: float64
```

- Ta thấy được cột id_office bị thiếu khoảng 4%, cột job/role bị thiếu 13.5%, cột address_x bị thiếu 25%, cột address_y bị thiếu 42%. Các cột còn lại thì không bị thiếu.
- Trong 4 cột bị thiếu này, ta thấy cột job/role có ý nghĩa rất lớn đến việc xem xét thu nhập, vì điều kiện để biết được thu nhập của một người là phải xem chức vụ của họ là gì. Nên đối với những dòng bị thiếu job/role, thì ta tiến hành loại bỏ để có thể đạt được ý nghĩa cao hơn cho dữ liệu.

- Sau khi loại bỏ những hàng bị thiếu job/role, ta còn lại 214102 dòng và 14 cột.

	id	id_bh	id_management	id_office	company_type	job/role	from_date	to_date	employee_lv	address_x	bithYear	gender	address_y	label
0	1	113039360	106	TF2212F	-1	Giám đốc	20130100	20151200	7.0	Hà Nội	1971	MALE	hà Nội	4
1	1	113039360	106	TF2212F	-1	Giám đốc	20160100	20220400	10.0	Hà Nội	1971	MALE	hà Nội	4
2	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20160800	20161200	7.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2
3	2	116074930	102	TB16010	-1	Nhân viên lễ tân	20170100	20170300	8.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2
5	2	116074930	102	TB16010	-1	Nhân viên Sales Admin	20170800	20191200	8.0	Hà Nội	1993	FEMALE	Thành phố Hà Nội	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
247553	55008	2616117553	2600	YN0027Z	-1	Hàn X3	20100700	20100700	2.0	NaN	1984	MALE	NaN	2
247554	55008	2616117553	2600	YN0027Z	-1	Công nhân ĐM robốt	20100800	20100900	2.0	NaN	1984	MALE	NaN	2
247556	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20161100	20171200	6.0	Vĩnh Phúc	1984	MALE	NaN	2
247557	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20180100	20191200	8.0	Vĩnh Phúc	1984	MALE	NaN	2
247558	55008	2616117553	2600	TZ0085Z	1	Phụ kho	20200100	20210200	9.0	Vĩnh Phúc	1984	MALE	NaN	2

214102 rows × 14 columns

- Sau đó, ta tiếp tục lọc các giá trị bị trùng nhau. Ta lọc dựa trên 3 tiêu chí là 'id_bh', 'to_date', 'from_date'. Ta dùng code như sau:

```
df_filter = df.sort_values(
    by=['id_bh', 'to_date', 'from_date'],
    ascending=[True, False, False]
).drop_duplicates(
    subset=['id_bh'],
    keep='first',
    ignore_index=True
)
df_filter
```

⇨ Ta chỉ còn 27490 dòng và 14 cột

- Sau đó, ta tiến hành tính số tuổi của các nhân viên trong danh sách, tạo them 1 cột là ‘age’ cho dễ nhìn. Từ đó, ta bắt đầu lọc những người bị delay thời gian dựa trên thời gian tham gia, sau đó có dữ liệu để dễ dàng làm việc hơn

	id	id_bh	id_management	id_office	company_type	job/role	from_date	to_date	employee_lv	address_x	birthYear	gender	address_y	label	address	lastest	age	delay
0	4686	100000725	100	HW01180	6	CVC, TP	20210600	20220400	17.0	HN	1963	MALE	Hà Nội	5	HN	2022	59	0
1	29905	100007067	2400	TA0002A	1	Nhân viên bán xăng dầu	20190100	20220400	9.0	TP Bắc Giang	1971	MALE	Bắc Giang	4	TP Bắc Giang	2022	51	0
2	4503	100007555	109	TI33611	-1	Nhân viên	20211000	20220400	9.0	Việt Nam	1970	MALE	NaN	2	Việt Nam	2022	52	0
3	2939	100008102	100	HW0013Z	-1	Chuyên viên chính, Ủy viên Thường trực Ủy ban...	20200800	20220400	18.0	Hà Nội	1970	MALE	HN	5	Hà Nội	2022	52	0
4	8967	100008777	114	HN05380	-1	PCT UBND huyện	20201000	20220400	17.0	HÀ NỘI	1965	MALE	NaN	6	HÀ NỘI	2022	57	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
27485	32655	9719632107	129	TS02775	-1	Kỹ sư công nghệ thông tin	20210700	20211200	14.0	NaN	1995	MALE	NaN	6	NaN	2021	27	1

- Tiếp theo, t dùng hàm fillna để điền vào các vị trí bị trống của office và address (trong bảng cũ hiện là NaN)

```
df_filter['id_office'].fillna('Unknow', inplace=True)
df_filter['address_x'].fillna('Việt Nam', inplace=True)
df_filter['address_y'].fillna('Việt Nam', inplace=True)
df_filter
```

⇨ Đây là kết quả

	id	id_bh	id_management	id_office	company_type	job/role	from_date	to_date	employee_lv	address_x	birthYear	gender	address_y	label	address	lastest	age	delay
0	4686	100000725	100	HW01180	0	CVC, TP	20210600	20220400	17.0	HN	1983	MALE	Hà Nội	5	HN	2022	39	0
1	29905	100007067	2400	TA0002A	1	Nhân viên bán xăng dầu	20190100	20220400	9.0	TP Bắc Giang	1971	MALE	Bắc Giang	4	TP Bắc Giang	2022	51	0
2	4503	100007535	109	TI3361I	-1	Nhân viên	20211000	20220400	9.0	Việt Nam	1970	MALE	Việt Nam	2	Việt Nam	2022	52	0
3	2939	100008102	100	HW0013Z	-1	Chuyên viên chính, Ủy viên Thường trực Ủy ban...	20200800	20220400	18.0	Hà Nội	1970	MALE	HN	5	Hà Nội	2022	52	0
4	8967	100008777	114	HN05360	-1	PCT UBND huyện	20201000	20220400	17.0	HÀ NỘI	1965	MALE	Việt Nam	6	HÀ NỘI	2022	57	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
27485	32655	9719632107	129	TI02775	-1	Kỹ sư công nghệ thông tin	20210700	20211200	14.0	Việt Nam	1995	MALE	Việt Nam	6	NaN	2021	27	1
27486	32685	9809227388	4009	TI11934	-1	Quản lý bán hàng	20210900	20220400	7.0	Tỉnh Nghệ An	1984	MALE	Việt Nam	2	Tỉnh Nghệ An	2022	38	0

- Sử dụng hàm thuộc Catboost, ta chia các dữ liệu đã ghi nhận được ra thành 4 file

b. Xử lý dữ liệu liên quan đến test (làm tương tự 3a)

⇨ Sau khi hoàn tất việc xử lý dữ liệu, ta gộp kết quả với các file đã được chia, sử dụng hàm để lưu các dữ liệu thành 1 file csv

```
submit = pd.read_csv('../input/uet-hackathon-2022-data-science/label_test.csv')
```

+ Code + Markdown

```
res = df_filter_test[['id_bh', 'label']]
```

```
submit_df = pd.merge(submit, res, 'inner', 'id_bh')
submit_df.to_csv('submission.csv', index=False)
```

4. Xây dựng mô hình

- Bài toán sử dụng mô hình Catboost, mô hình này triển khai cây đối xứng (symmetric trees) giúp giảm thời gian dự đoán và nó cũng có độ sâu của cây nông hơn theo mặc định (sáu)
- CatBoost tận dụng các hoán vị ngẫu nhiên

5. Kết quả